

LS-DYNA[®]
KEYWORD USER'S MANUAL
VOLUME I

May 2007
Version 971

LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC)

Corporate Address

Livermore Software Technology Corporation
P. O. Box 712
Livermore, California 94551-0712

Support Addresses

Livermore Software Technology Corporation
7374 Las Positas Road
Livermore, California 94551
Tel: 925-449-2500 ♦ Fax: 925-449-2507

Email: sales@lstc.com

Website: www.lstc.com

Livermore Software Technology Corporation
1740 West Big Beaver Road
Suite 100
Troy, Michigan 48084
Tel: 248-649-4728 ♦ Fax: 248-649-6328

Disclaimer

Copyright © 1992-2007 Livermore Software Technology Corporation. All Rights Reserved.

LS-DYNA[®], LS-OPT[®] and LS-PrePost[®] are registered trademarks of Livermore Software Technology Corporation in the United States. All other trademarks, product names and brand names belong to their respective owners.

LSTC reserves the right to modify the material contained within this manual without prior notice.

The information and examples included herein are for illustrative purposes only and are not intended to be exhaustive or all-inclusive. LSTC assumes no liability or responsibility whatsoever for any direct or indirect damages or inaccuracies of any type or nature that could be deemed to have resulted from the use of this manual.

Any reproduction, in whole or in part, of this manual is prohibited without the prior written approval of LSTC. All requests to reproduce the contents hereof should be sent to sales@lstc.com.

ISBN 0-9778540-2-7

AES

Copyright © 2001, Dr Brian Gladman <brg@gladman.uk.net>, Worcester, UK. All rights reserved.

LICENSE TERMS

The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:

1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holder's name is not used to endorse products built using this software without specific written permission.

DISCLAIMER

This software is provided 'as is' with no explicit or implied warranties in respect of any properties, including, but not limited to, correctness and fitness for purpose.

Issue Date: 21/01/2002

This file contains the code for implementing the key schedule for AES (Rijndael) for block and key sizes of 16, 24, and 32 bytes.

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION.....	I.1
CHRONOLOGICAL HISTORY	I.1
MATERIAL MODELS.....	I.21
SPATIAL DISCRETIZATION	I.22
CONTACT-IMPACT INTERFACES	I.26
INTERFACE DEFINITIONS FOR COMPONENT ANALYSIS	I.27
CAPACITY.....	I.27
PRECISION.....	I.27
GETTING STARTED.....	GS.1
DESCRIPTION OF KEYWORD INPUT	GS.1
SUMMARY OF COMMONLY USED OPTIONS	GS.9
EXECUTION SYNTAX.....	GS.10
SENSE SWITCH CONTROLS	GS.12
PROCEDURE FOR LS-DYNA/MPP.....	GS.13
FILES: INPUT AND OUTPUT	GS.14
RESTART ANALYSIS	GS.16
VDA/IGES DATABASES.....	GS.17
LS-PrePost®	GS.18
EXECUTION SPEEDS	GS.20
UNITS.....	GS.21
GENERAL CARD FORMAT.....	GS.21
*AIRBAG.....	1.1
*AIRBAG_OPTION1_{OPTION2}_{OPTION3}_{OPTION4}.....	1.2
SIMPLE_PRESSURE_VOLUME.....	1.8
SIMPLE_AIRBAG_MODEL	1.9
ADIABATIC_GAS_MODEL.....	1.11
WANG_NEFSKE.....	1.13
LOAD_CURVE.....	1.28
LINEAR_FLUID	1.29
HYBRID	1.31
HYBRID_CHEMKIN.....	1.38
*AIRBAG_ADVANCED_ALE.....	1.43
*AIRBAG_ALE.....	1.51
*AIRBAG_INTERACTION.....	1.65

TABLE OF CONTENTS

*AIRBAG_REFERENCE_GEOMETRY_{OPTION}_{OPTION}.....	1.67
*AIRBAG_SHELL_REFERENCE_GEOMETRY	1.69
*ALE.....	2.1
*ALE_FSI_PROJECTION	2.2
*ALE_FSI_SWITCH_MMG_{OPTION}	2.5
*ALE_MULTI-MATERIAL_GROUP	2.9
*ALE_REFERENCE_SYSTEM_CURVE	2.13
*ALE_REFERENCE_SYSTEM_GROUP	2.15
*ALE_REFERENCE_SYSTEM_NODE.....	2.22
*ALE_REFERENCE_SYSTEM_SWITCH.....	2.24
*ALE_SMOOTHING	2.26
*ALE_TANK_TEST	2.27
*ALE_UP_SWITCH.....	2.31
*BOUNDARY	3.1
*BOUNDARY_ACOUSTIC_COUPLING.....	3.2
*BOUNDARY_AMBIENT_EOS.....	3.3
*BOUNDARY_CONVECTION_OPTION	3.5
*BOUNDARY_CYCLIC.....	3.7
*BOUNDARY_ELEMENT_METHOD_OPTION	3.9
*BOUNDARY_FLUX_OPTION.....	3.23
*BOUNDARY_MCOL.....	3.26
*BOUNDARY_NON_REFLECTING.....	3.28
*BOUNDARY_NON_REFLECTING_2D.....	3.30
*BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID	3.32
*BOUNDARY_PRESCRIBED_MOTION_OPTION1_{OPTION2}.....	3.34
*BOUNDARY_PRESCRIBED_ORIENTATION_RIGID_OPTION	3.39
*BOUNDARY_PRESSURE_OUTFLOW_OPTION	3.44
*BOUNDARY_RADIATION_OPTION1_{OPTION2}	3.45
*BOUNDARY_SLIDING_PLANE.....	3.53
*BOUNDARY_SPC_OPTION1_{OPTION2}.....	3.54
*BOUNDARY_SPH_FLOW	3.56
*BOUNDARY_SPH_SYMMETRY_PLANE.....	3.58
*BOUNDARY_SYMMETRY_FAILURE	3.59
*BOUNDARY_TEMPERATURE_OPTION	3.60
*BOUNDARY_THERMAL_WELD.....	3.61
*BOUNDARY_USA_SURFACE.....	3.64

TABLE OF CONTENTS

*CASE	4.1
*CASE_{ <i>OPTION</i> }	4.1
*COMPONENT	5.1
*COMPONENT_GEBOD_ <i>OPTION</i>	5.2
*COMPONENT_GEBOD_JOINT_ <i>OPTION</i>	5.4
*COMPONENT_HYBRIDIII	5.8
*COMPONENT_HYBRIDIII_JOINT_ <i>OPTION</i>	5.10
*CONSTRAINED	6.1
*CONSTRAINED_ADAPTIVITY	6.2
*CONSTRAINED_BUTT_WELD	6.3
*CONSTRAINED_EULER_IN_EULER	6.6
*CONSTRAINED_EXTRA_NODES_ <i>OPTION</i>	6.8
*CONSTRAINED_GENERALIZED_WELD_ <i>OPTION</i> _{ <i>OPTION</i> }	6.10
*CONSTRAINED_GLOBAL	6.23
*CONSTRAINED_INTERPOLATION_{ <i>OPTION</i> }	6.25
*CONSTRAINED_JOINT_ <i>OPTION</i> _{ <i>OPTION</i> }_{ <i>OPTION</i> }_{ <i>OPTION</i> }	6.30
*CONSTRAINED_JOINT_STIFFNESS_ <i>OPTION</i>	6.42
*CONSTRAINED_LAGRANGE_IN_SOLID_{ <i>OPTION1</i> }_{ <i>OPTION2</i> }	6.55
*CONSTRAINED_LINEAR_GLOBAL	6.71
*CONSTRAINED_LINEAR_LOCAL	6.74
*CONSTRAINED_LOCAL	6.76
*CONSTRAINED_NODAL_RIGID_BODY_{ <i>OPTION</i> }_{ <i>OPTION</i> }_{ <i>OPTION</i> }	6.78
*CONSTRAINED_NODE_SET_{ <i>OPTION</i> }	6.86
*CONSTRAINED_POINTS	6.89
*CONSTRAINED_RIGID_BODIES	6.91
*CONSTRAINED_RIGID_BODY_STOPPERS	6.93
*CONSTRAINED_RIVET_{ <i>OPTION</i> }	6.96
*CONSTRAINED_SHELL_TO_SOLID	6.98
*CONSTRAINED_SPLINE	6.100
*CONSTRAINED_SPOTWELD_{ <i>OPTION</i> }_{ <i>OPTION</i> }	6.102
*CONSTRAINED_TIE-BREAK	6.106
*CONSTRAINED_TIED_NODES_FAILURE	6.107
*CONTACT	7.1
*CONTACT_ <i>OPTION1</i> _{ <i>OPTION2</i> }_{ <i>OPTION3</i> }_{ <i>OPTION4</i> }_{ <i>OPTION5</i> }	7.2
*CONTACT_AUTO_MOVE	7.51
*CONTACT_COUPLING	7.52

TABLE OF CONTENTS

*CONTACT_ENTITY.....	7.54
*CONTACT_GEBOD_OPTION.....	7.63
*CONTACT_GUIDED_CABLE_{ OPTION1}_{ OPTION2}.....	7.66
*CONTACT_INTERIOR	7.68
*CONTACT_RIGID_SURFACE	7.70
*CONTACT_1D.....	7.74
*CONTACT_2D_OPTION1_{ OPTION2}_{ OPTION3}.....	7.75
*CONTROL.....	8.1
*CONTROL_ACCURACY	8.4
*CONTROL_ADAPSTEP.....	8.6
*CONTROL_ADAPTIVE.....	8.7
*CONTROL_ADAPTIVE_CURVE.....	8.14
*CONTROL_ALE	8.15
*CONTROL_BULK_VISCOSITY.....	8.19
*CONTROL_CHECK_{ OPTION}.....	8.20
*CONTROL_COARSEN.....	8.22
*CONTROL_CONTACT	8.24
*CONTROL_COUPLING.....	8.32
*CONTROL_CPU	8.34
*CONTROL_DYNAMIC_RELAXATION.....	8.35
*CONTROL_EFG.....	8.38
*CONTROL_ENERGY	8.40
*CONTROL_EXPLOSIVE_SHADOW	8.41
*CONTROL_FORMING_POSITION.....	8.42
*CONTROL_FORMING_PROJECTION.....	8.43
*CONTROL_FORMING_TEMPLATE.....	8.44
*CONTROL_FORMING_TRAVEL	8.50
*CONTROL_FORMING_USER.....	8.51
*CONTROL_HOURLASS_{ OPTION}	8.54
*CONTROL_IMPLICIT_AUTO.....	8.56
*CONTROL_IMPLICIT_BUCKLE	8.59
*CONTROL_IMPLICIT_DYNAMICS.....	8.60
*CONTROL_IMPLICIT_EIGENVALUE.....	8.63
*CONTROL_IMPLICIT_GENERAL	8.66
*CONTROL_IMPLICIT_INERTIA_RELIEF.....	8.69
*CONTROL_IMPLICIT_JOINTS	8.70

TABLE OF CONTENTS

*CONTROL_IMPLICIT_MODES.....	8.71
*CONTROL_IMPLICIT_SOLUTION.....	8.73
*CONTROL_IMPLICIT_SOLVER.....	8.80
*CONTROL_IMPLICIT_STABILIZATION.....	8.84
*CONTROL_IMPLICIT_TERMINATION.....	8.86
*CONTROL_MPP_DECOMPOSITION_AUTOMATIC.....	8.87
*CONTROL_MPP_DECOMPOSITION_CHECK_SPEED.....	8.88
*CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE.....	8.89
*CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE.....	8.90
*CONTROL_MPP_DECOMPOSITION_FILE.....	8.91
*CONTROL_MPP_DECOMPOSITION_METHOD.....	8.92
*CONTROL_MPP_DECOMPOSITION_NUMPROC.....	8.93
*CONTROL_MPP_DECOMPOSITION_SHOW.....	8.94
*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION.....	8.95
*CONTROL_MPP_IO_NOD3DUMP.....	8.97
*CONTROL_MPP_IO_NODUMP.....	8.98
*CONTROL_MPP_IO_NOFULL.....	8.99
*CONTROL_MPP_IO_SWAPBYTES.....	8.100
*CONTROL_NONLOCAL.....	8.101
*CONTROL_OUTPUT.....	8.102
*CONTROL_PARALLEL.....	8.105
*CONTROL_REMESHING.....	8.107
*CONTROL_RIGID.....	8.108
*CONTROL_SHELL.....	8.110
*CONTROL_SOLID.....	8.116
*CONTROL_SOLUTION.....	8.118
*CONTROL_SPH.....	8.119
*CONTROL_SPOTWELD_BEAM.....	8.121
*CONTROL_STAGED_CONSTRUCTION.....	8.124
*CONTROL_STRUCTURED_{OPTION}.....	8.126
*CONTROL_SUBCYCLE.....	8.127
*CONTROL_TERMINATION.....	8.128
*CONTROL_THERMAL_NONLINEAR.....	8.129
*CONTROL_THERMAL_SOLVER.....	8.130
*CONTROL_THERMAL_TIMESTEP.....	8.133
*CONTROL_TIMESTEP.....	8.134

TABLE OF CONTENTS

*DAMPING	9.1
*DAMPING_FREQUENCY_RANGE	9.2
*DAMPING_GLOBAL	9.4
*DAMPING_PART_MASS	9.6
*DAMPING_PART_STIFFNESS	9.8
*DAMPING_RELATIVE	9.10
*DATABASE	10.1
*DATABASE_OPTION	10.2
*DATABASE_ADAMS	10.8
*DATABASE_BINARY_OPTION	10.9
*DATABASE_CROSS_SECTION_OPTION1_{OPTION2}	10.13
*DATABASE_EXTENT_OPTION	10.17
*DATABASE_FORMAT	10.27
*DATABASE_FSI	10.28
*DATABASE_FSI_SENSOR	10.32
*DATABASE_HISTORY_OPTION	10.34
*DATABASE_NODAL_FORCE_GROUP	10.38
*DATABASE_SPRING_FORWARD	10.39
*DATABASE_SUPERPLASTIC_FORMING	10.40
*DATABASE_TRACER	10.41
*DEFINE	11.1
*DEFINE_ALEBAG_BAG	11.3
*DEFINE_ALEBAG_HOLE	11.8
*DEFINE_ALEBAG_INFLATOR	11.10
*DEFINE_BOX	11.13
*DEFINE_BOX_ADAPTIVE	11.14
*DEFINE_BOX_COARSEN	11.15
*DEFINE_BOX_DRAWBEAD	11.16
*DEFINE_BOX_SPH	11.18
*DEFINE_CONNECTION_PROPERTIES_{OPTION}	11.20
*DEFINE_CONSTRUCTION_STAGES	11.26
*DEFINE_CONTACT_VOLUME	11.27
*DEFINE_COORDINATE_NODES	11.30
*DEFINE_COORDINATE_SYSTEM	11.32
*DEFINE_COORDINATE_VECTOR	11.34
*DEFINE_CURVE_{OPTION}	11.36

TABLE OF CONTENTS

*DEFINE_CURVE_COMPENSATION	11.39
*DEFINE_CURVE_DRAWBEAD	11.40
*DEFINE_CURVE_ENTITY	11.41
*DEFINE_CURVE_FEEDBACK	11.43
*DEFINE_CURVE_FUNCTION.....	11.45
*DEFINE_CURVE_SMOOTH	11.54
*DEFINE_CURVE_TRIM_{OPTION}.....	11.56
*DEFINE_DEATH_TIMES_OPTION.....	11.60
*DEFINE_FRICTION	11.63
*DEFINE_HEX_SPOTWELD_ASSEMBLY_{OPTION}.....	11.65
*DEFINE_SD_ORIENTATION	11.67
*DEFINE_SET_ADAPTIVE	11.69
*DEFINE_SPOTWELD_FAILURE_RESULTANTS	11.70
*DEFINE_SPOTWELD RUPTURE_PARAMETER	11.72
*DEFINE_SPOTWELD RUPTURE_STRESS.....	11.75
*DEFINE_STAGED_CONSTRUCTION_PART	11.77
*DEFINE_TABLE.....	11.78
*DEFINE_TRANSFORMATION.....	11.80
*DEFINE_VECTOR	11.83
EXAMPLES	11.84
*DEFORMABLE_TO_RIGID.....	12.1
*DEFORMABLE_TO_RIGID	12.2
*DEFORMABLE_TO_RIGID_AUTOMATIC.....	12.3
*DEFORMABLE_TO_RIGID_INERTIA	12.7
*EF.....	13.1
*EF_CONTROL	13.2
*EF_GRID	13.4
*EF_MATERIAL	13.5
*EF_TOGGLES.....	13.8
*ELEMENT.....	14.1
*ELEMENT_BEAM_{OPTION}_{OPTION}	14.2
*ELEMENT_DIRECT_MATRIX_INPUT	14.12
*ELEMENT_DISCRETE.....	14.14
*ELEMENT_INERTIA_{OPTION}.....	14.16
*ELEMENT_MASS_{OPTION}.....	14.19
*ELEMENT_MASS_PART_{OPTION}.....	14.20

TABLE OF CONTENTS

*ELEMENT_PLOTEL	14.21
*ELEMENT_SEATBELT	14.22
*ELEMENT_SEATBELT_ACCELEROMETER	14.24
*ELEMENT_SEATBELT_PRETENSIONER	14.26
*ELEMENT_SEATBELT_RETRACTOR.....	14.31
*ELEMENT_SEATBELT_SENSOR	14.37
*ELEMENT_SEATBELT_SLIPRING	14.41
*ELEMENT_SHELL_{ <i>OPTION</i> }.....	14.43
*ELEMENT_SHELL_SOURCE_SINK	14.49
*ELEMENT_SOLID_{ <i>OPTION</i> }	14.50
*ELEMENT_SPH.....	14.56
*ELEMENT_TRIM	14.57
*ELEMENT_TSHELL	14.58
*EOS.....	15.1
*EOS_LINEAR_POLYNOMIAL.....	15.7
*EOS_JWL	15.9
*EOS_SACK_TUESDAY	15.10
*EOS_GRUNEISEN.....	15.11
*EOS_RATIO_OF_POLYNOMIALS.....	15.13
*EOS_LINEAR_POLYNOMIAL_WITH_ENERGY_LEAK.....	15.17
*EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE.....	15.19
*EOS_TABULATED_COMPACTION	15.23
*EOS_TABULATED	15.25
*EOS_PROPELLANT_DEFLAGRATION	15.27
*EOS_TENSOR_PORE_COLLAPSE.....	15.32
*EOS_IDEAL_GAS.....	15.35
*EOS_JWLB.....	15.37
*EOS_GASKET.....	15.41
*EOS_USER_DEFINED	15.43
*HOURLASS.....	16.1
*HOURLASS	16.1
*INCLUDE.....	17.1
*INCLUDE_{ <i>OPTION</i> }	17.2
*INCLUDE_COMPENSATION_OPTION	17.10

TABLE OF CONTENTS

*INITIAL	18.1
*INITIAL_DETONATION	18.2
*INITIAL_FOAM_REFERENCE_GEOMETRY	18.5
*INITIAL_GAS_MIXTURE	18.6
*INITIAL_MOMENTUM	18.8
*INITIAL_STRAIN_SHELL	18.9
*INITIAL_STRAIN_SOLID	18.10
*INITIAL_STRESS_BEAM	18.11
*INITIAL_STRESS_DEPTH	18.13
*INITIAL_STRESS_SECTION	18.14
*INITIAL_STRESS_SHELL_{ <i>OPTION</i> }	18.16
*INITIAL_STRESS_SOLID	18.18
*INITIAL_STRESS_TSHELL	18.21
*INITIAL_TEMPERATURE_ <i>OPTION</i>	18.23
*INITIAL_VEHICLE_KINEMATICS	18.24
*INITIAL_VELOCITY	18.27
*INITIAL_VELOCITY_NODE	18.30
*INITIAL_VELOCITY_RIGID_BODY	18.31
*INITIAL_VELOCITY_GENERATION	18.32
*INITIAL_VOID_ <i>OPTION</i>	18.34
*INITIAL_VOLUME_FRACTION	18.35
*INITIAL_VOLUME_FRACTION_GEOMETRY	18.36
*INTEGRATION	19.1
*INTEGRATION_BEAM	19.2
*INTEGRATION_SHELL	19.15
*INTERFACE	20.1
*INTERFACE_COMPENSATION_NEW	20.2
*INTERFACE_COMPONENT_ <i>OPTION</i>	20.5
*INTERFACE_JOY	20.6
*INTERFACE_LINKING_DISCRETE_NODE_ <i>OPTION</i>	20.7
*INTERFACE_LINKING_EDGE	20.8
*INTERFACE_LINKING_SEGMENT	20.9
*INTERFACE_SPRINGBACK_ <i>OPTION1</i> _ <i>OPTION2</i>	20.10
*KEYWORD	21.1

TABLE OF CONTENTS

*LOAD	22.1
*LOAD_ALE_CONVECTION_{OPTION}	22.3
*LOAD_BEAM_OPTION	22.5
*LOAD_BLAST	22.7
*LOAD_BODY_OPTION	22.9
*LOAD_BODY_GENERALIZED	22.13
*LOAD_BODY_POROUS	22.15
*LOAD_BRODE	22.17
*LOAD_DENSITY_DEPTH	22.19
*LOAD_GRAVITY_PART	22.21
*LOAD_HEAT_CONTROLLER	22.23
*LOAD_HEAT_GENERATION_OPTION	22.24
*LOAD_MASK	22.25
*LOAD_MOTION_NODE	22.27
*LOAD_MOVING_PRESSURE	22.29
*LOAD_NODE_OPTION	22.31
*LOAD_REMOVE_PART	22.34
*LOAD_RIGID_BODY	22.35
*LOAD_SEGMENT_{OPTION}	22.37
*LOAD_SEGMENT_NONUNIFORM_{OPTION}	22.41
*LOAD_SEGMENT_SET_{OPTION}	22.44
*LOAD_SEGMENT_SET_NONUNIFORM_{OPTION}	22.46
*LOAD_SHELL_OPTION1_{OPTION2}	22.48
*LOAD_SSA	22.51
*LOAD_STIFFEN_PART	22.55
*LOAD_SUPERPLASTIC_FORMING	22.57
*LOAD_SURFACE_STRESS_{OPTION}	22.59
*LOAD_THERMAL_OPTION	22.60
*LOAD_THERMAL_VARIABLE_SHELL_{OPTION}	22.68
*LOAD_VOLUME_LOSS	22.70
*NODE	23.1
*NODE	23.2
*NODE_RIGID_SURFACE	23.4
*NODE_SCALAR_{OPTION}	23.5
*NODE_TRANSFORM	23.7

TABLE OF CONTENTS

*PARAMETER	24.1
*PARAMETER	24.2
*PARAMETER_EXPRESSION	24.3
*PART	25.1
*PART_{ <i>OPTION1</i> }_{ <i>OPTION2</i> }_{ <i>OPTION3</i> }_{ <i>OPTION4</i> }.....	25.2
*PART_ADAPTIVE_FAILURE.....	25.11
*PART_COMPOSITE_{ <i>OPTION</i> }.....	25.12
*PART_MODES	25.16
*PART_SENSOR	25.20
*PART_MOVE.....	25.21
*PERTURBATION	26.1
*PERTURBATION_NODE	26.2
*PERTURBATION_SHELL_THICKNESS	26.5
*RAIL	27.1
*RAIL_TRACK	27.2
*RAIL_TRAIN	27.7
*RIGIDWALL	28.1
*RIGIDWALL_GEOMETRIC_OPTION_{ <i>OPTION</i> }_{ <i>OPTION</i> }	28.2
*RIGIDWALL_PLANAR_{ <i>OPTION</i> }_{ <i>OPTION</i> }_{ <i>OPTION</i> }	28.12
*SECTION	29.1
*SECTION_BEAM	29.2
*SECTION_DISCRETE.....	29.16
*SECTION_POINT_SOURCE	29.19
*SECTION_POINT_SOURCE_MIXTURE	29.21
*SECTION_SEATBELT	29.27
*SECTION_SHELL_{ <i>OPTION</i> }	29.28
*SECTION_SOLID_{ <i>OPTION</i> }.....	29.40
*SECTION_SPH_{ <i>OPTION</i> }	29.49
*SECTION_TSHELL.....	29.52
*SENSOR	30.1
*SENSOR_CONTROL.....	30.3
*SENSOR_DEFINE_CALC-MATH.....	30.4
*SENSOR_DEFINE_ELEMENT.....	30.6
*SENSOR_DEFINE_FORCE	30.7
*SENSOR_DEFINE_NODE	30.9

TABLE OF CONTENTS

*SENSOR_SWITCH	30.10
*SENSOR_SWITCH_CALC-LOGIC.....	30.11
*SET	31.1
*SET_BEAM_{ <i>OPTION</i> }.....	31.2
*SET_BEAM_ADD.....	31.5
*SET_DISCRETE_{ <i>OPTION</i> }.....	31.6
*SET_DISCRETE_ADD	31.9
*SET_MULTI-MATERIAL_GROUP_LIST.....	31.10
*SET_NODE_{ <i>OPTION</i> }	31.11
*SET_NODE_ADD_{ <i>OPTION</i> }.....	31.15
*SET_PART_{ <i>OPTION</i> }	31.17
*SET_PART_ADD.....	31.20
*SET_SEGMENT_{ <i>OPTION</i> }.....	31.21
*SET_2D_SEGMENT_{ <i>OPTION</i> }.....	31.25
*SET_SHELL_{ <i>OPTION</i> }	31.27
*SET_SHELL_ADD.....	31.31
*SET_SOLID_{ <i>OPTION</i> }.....	31.32
*SET_SOLID_ADD	31.35
*SET_TSHELL_{ <i>OPTION</i> }.....	31.36
*TERMINATION.....	32.1
*TERMINATION_BODY.....	32.2
*TERMINATION_CONTACT	32.3
*TERMINATION_CURVE.....	32.4
*TERMINATION_DELETED_SHELLS_{ <i>OPTION</i> }.....	32.5
*TERMINATION_NODE	32.6
*TITLE.....	33.1
*USER	34.1
*USER_INTERFACE_ <i>OPTION</i>	34.1
*USER_LOADING.....	34.3
RESTART INPUT DATA.....	35.1
*CHANGE_ <i>OPTION</i>	35.3
*CONTROL_DYNAMIC_RELAXATION.....	35.17
*CONTROL_SHELL.....	35.19
*CONTROL_TERMINATION	35.21
*CONTROL_TIMESTEP	35.22

TABLE OF CONTENTS

*DAMPING_GLOBAL.....	35.23
*DATABASE_OPTION.....	35.24
*DATABASE_BINARY_OPTION.....	35.26
*DELETE_OPTION.....	35.27
*INTERFACE_SPRINGBACK.....	35.29
*RIGID_DEFORMABLE_OPTION.....	35.31
*STRESS_INITIALIZATION_{OPTION}.....	35.35
*TERMINATION_OPTION.....	35.37
*TITLE.....	35.39
REFERENCES.....	REF.1
APPENDIX A	
USER DEFINED MATERIALS.....	A.1
APPENDIX B	
USER DEFINED EQUATION OF STATE.....	B.1
APPENDIX C	
USER DEFINED ELEMENT INTERFACE FOR SOLIDS AND SHELLS.....	C.1
APPENDIX D	
USER DEFINED AIRBAG SENSOR.....	D.1
APPENDIX E	
USER DEFINED SOLUTION CONTROL.....	E.1
APPENDIX F	
USER DEFINED INTERFACE CONTROL.....	F.1
APPENDIX G	
USER DEFINED INTERFACE FRICTION.....	G.1
APPENDIX H	
USER DEFINED THERMAL MATERIAL MODEL.....	H.1
APPENDIX I	
OCCUPANT SIMULATION INCLUDING COUPLING TO CAL3D AND MADYMO.....	I.1
INTRODUCTION.....	I.1
THE LS-DYNA/OCCUPANT SIMULATION PROGRAM LINK.....	I.1
DUMMY MODELING.....	I.3
AIRBAG MODELING.....	I.4
KNEE BOLSTER.....	I.5
COMMON ERRORS.....	I.5

TABLE OF CONTENTS

APPENDIX J

INTERACTIVE GRAPHICS COMMANDS.....	J.1
------------------------------------	-----

APPENDIX K

INTERACTIVE MATERIAL MODEL DRIVER	K.1
INTRODUCTION.....	K.1
INPUT DEFINITION	K.1
INTERACTIVE DRIVER COMMANDS	K.3

APPENDIX L

VDA DATABASE	L.1
--------------------	-----

APPENDIX M

COMMANDS FOR TWO-DIMENSIONAL REZONING.....	M.1
REZONING COMMANDS BY FUNCTION.....	M.2

APPENDIX N

RIGID BODY DUMMIES.....	N.1
-------------------------	-----

APPENDIX O

LS-DYNA MPP USER GUIDE.....	O.1
-----------------------------	-----

APPENDIX P

IMPLICIT SOLVER.....	P.1
INTRODUCTION.....	P.1
SETTING UP AN IMPLICIT SIMULATION	P.1
LINEAR EQUATION SOLVER	P.2
NONLINEAR EQUATION SOLVER.....	P.3
ELEMENT FORMULATIONS FOR IMPLICIT ANALYSIS.....	P.4
APPLYING LOADS DURING IMPLICIT ANALYSIS.....	P.5
AUTOMATIC TIME STEP SIZE CONTROL.....	P.5
IMPLICIT STRESS INITIALIZATION.....	P.6
TROUBLESHOOTING CONVERGENCE PROBLEMS.....	P.6

APPENDIX Q

USER DEFINED WELD FAILURE	Q.1
---------------------------------	-----

APPENDIX R

USER DEFINED COHESIVE MODEL.....	R.1
----------------------------------	-----

APPENDIX S

USER DEFINED BOUNDARY FLUX.....	S.1
---------------------------------	-----

LS-DYNA USER'S MANUAL

INTRODUCTION

CHRONOLOGICAL HISTORY

DYNA3D originated at the Lawrence Livermore National Laboratory [Hallquist 1976]. The early applications were primarily for the stress analysis of structures subjected to a variety of impact loading. These applications required what was then significant computer resources, and the need for a much faster version was immediately obvious. Part of the speed problem was related to the inefficient implementation of the element technology which was further aggravated by the fact that supercomputers in 1976 were much slower than today's PC. Furthermore, the primitive sliding interface treatment could only treat logically regular interfaces that are uncommon in most finite element discretizations of complicated three-dimensional geometries; consequently, defining a suitable mesh for handling contact was often very difficult. The first version contained trusses, membranes, and a choice of solid elements. The solid elements ranged from a one-point quadrature eight-noded element with hourglass control to a twenty-noded element with eight integration points. Due to the high cost of the twenty node solid, the zero energy modes related to the reduced 8-point integration, and the high frequency content which drove the time step size down, higher order elements were all but abandoned in later versions of DYNA3D. A two-dimensional version, DYNA2D, was developed concurrently.

A new version of DYNA3D was released in 1979 that was programmed to provide near optimal speed on the CRAY-1 supercomputers, contained an improved sliding interface treatment that permitted triangular segments and was an order of magnitude faster than the previous contact treatment. The 1979 version eliminated structural and higher order solid elements and some of the material models of the first version. This version also included an optional element-wise implementation of the integral difference method developed by Wilkins et al. [1974].

The 1981 version [Hallquist 1981a] evolved from the 1979 version. Nine additional material models were added to allow a much broader range of problems to be modeled including explosive-structure and soil-structure interactions. Body force loads were implemented for angular velocities and base accelerations. A link was also established from the 3D Eulerian code, JOY [Couch, et. al., 1983] for studying the structural response to impacts by penetrating projectiles. An option was provided for storing element data on disk thereby doubling the capacity of DYNA3D.

The 1982 version of DYNA3D [Hallquist 1982] accepted DYNA2D [Hallquist 1980] material input directly. The new organization was such that equations of state and constitutive models of any complexity could be easily added. Complete vectorization of the material models had been nearly achieved with about a 10 percent increase in execution speed over the 1981 version.

In the 1986 version of DYNA3D [Hallquist and Benson 1986], many new features were added, including beams, shells, rigid bodies, single surface contact, interface friction, discrete springs and dampers, optional hourglass treatments, optional exact volume integration, and VAX/ VMS, IBM, UNIX, COS operating systems compatibility, that greatly expanded its range of applications. DYNA3D thus became the first code to have a general single surface contact algorithm.

In the 1987 version of DYNA3D [Hallquist and Benson 1987] metal forming simulations and composite analysis became a reality. This version included shell thickness changes, the Belytschko-Tsay shell element [Belytschko and Tsay, 1981], and dynamic relaxation. Also

INTRODUCTION

included were non-reflecting boundaries, user specified integration rules for shell and beam elements, a layered composite damage model, and single point constraints.

New capabilities added in the 1988 DYNA3D [Hallquist 1988] version included a cost effective resultant beam element, a truss element, a C^0 triangular shell, the BCIZ triangular shell [Bazeley et al. 1965], mixing of element formulations in calculations, composite failure modeling for solids, noniterative plane stress plasticity, contact surfaces with spot welds, tie break sliding surfaces, beam surface contact, finite stonewalls, stonewall reaction forces, energy calculations for all elements, a crushable foam constitutive model, comment cards in the input, and one-dimensional slidelines.

By the end of 1988 it was obvious that a much more concentrated effort would be required in the development of this software if problems in crashworthiness were to be properly solved; therefore, Livermore Software Technology Corporation was founded to continue the development of DYNA3D as a commercial version called LS-DYNA3D which was later shortened to LS-DYNA. The 1989 release introduced many enhanced capabilities including a one-way treatment of slide surfaces with voids and friction; cross-sectional forces for structural elements; an optional user specified minimum time step size for shell elements using elastic and elastoplastic material models; nodal accelerations in the time history database; a compressible Mooney-Rivlin material model; a closed-form update shell plasticity model; a general rubber material model; unique penalty specifications for each slide surface; external work tracking; optional time step criterion for 4-node shell elements; and internal element sorting to allow full vectorization of right-hand-side force assembly.

During the last ten years, considerable progress has been made as may be seen in the chronology of the developments which follows.

Capabilities added in 1989-1990:

- arbitrary node and element numbers,
- fabric model for seat belts and airbags,
- composite glass model,
- vectorized type 3 contact and single surface contact,
- many more I/O options,
- all shell materials available for 8 node thick shell,
- strain rate dependent plasticity for beams,
- fully vectorized iterative plasticity,
- interactive graphics on some computers,
- nodal damping,
- shell thickness taken into account in shell type 3 contact,
- shell thinning accounted for in type 3 and type 4 contact,
- soft stonewalls,
- print suppression option for node and element data,
- massless truss elements, rivets – based on equations of rigid body dynamics,
- massless beam elements, spot welds – based on equations of rigid body dynamics,
- expanded databases with more history variables and integration points,
- force limited resultant beam,
- rotational spring and dampers, local coordinate systems for discrete elements,
- resultant plasticity for C^0 triangular element,
- energy dissipation calculations for stonewalls,
- hourglass energy calculations for solid and shell elements,
- viscous and Coulomb friction with arbitrary variation over surface,
- distributed loads on beam elements,
- Cowper and Symonds strain rate model,
- segmented stonewalls,
- stonewall Coulomb friction,
- stonewall energy dissipation,
- airbags (1990),

- nodal rigid bodies,
- automatic sorting of triangular shells into C^0 groups,
- mass scaling for quasi static analyses,
- user defined subroutines,
- warpage checks on shell elements,
- thickness consideration in all contact types,
- automatic orientation of contact segments,
- sliding interface energy dissipation calculations,
- nodal force and energy database for applied boundary conditions,
- defined stonewall velocity with input energy calculations,

Capabilities added in 1991-1992:

- rigid/deformable material switching,
- rigid bodies impacting rigid walls,
- strain-rate effects in metallic honeycomb model 26,
- shells and beams interfaces included for subsequent component analyses,
- external work computed for prescribed displacement/velocity/accelerations,
- linear constraint equations,
- MPGS database,
- MOVIE database,
- Slideline interface file,
- automated contact input for all input types,
- automatic single surface contact without element orientation,
- constraint technique for contact,
- cut planes for resultant forces,
- crushable cellular foams,
- urethane foam model with hysteresis,
- subcycling,
- friction in the contact entities,
- strains computed and written for the 8 node thick shells,
- “good” 4 node tetrahedron solid element with nodal rotations,
- 8 node solid element with nodal rotations,
- 2x2 integration for the membrane element,
- Belytschko-Schwer integrated beam,
- thin-walled Belytschko-Schwer integrated beam,
- improved TAURUS database control,
- null material for beams to display springs and seatbelts in TAURUS,
- parallel implementation on Crays and SGI computers,
- coupling to rigid body codes,
- seat belt capability.

Capabilities added in 1993-1994:

- Arbitrary Lagrangian Eulerian brick elements,
- Belytschko-Wong-Chiang quadrilateral shell element,
- Warping stiffness in the Belytschko-Tsay shell element,
- Fast Hughes-Liu shell element,
- Fully integrated thick shell element,
- Discrete 3D beam element,
- Generalized dampers,
- Cable modeling,
- Airbag reference geometry,
- Multiple jet model,
- Generalized joint stiffnesses,
- Enhanced rigid body to rigid body contact,
- Orthotropic rigid walls,

INTRODUCTION

- Time zero mass scaling,
- Coupling with USA (Underwater Shock Analysis),
- Layered spot welds with failure based on resultants or plastic strain,
- Fillet welds with failure,
- Butt welds with failure,
- Automatic eroding contact,
- Edge-to-edge contact,
- Automatic mesh generation with contact entities,
- Drawbead modeling,
- Shells constrained inside brick elements,
- NIKE3D coupling for springback,
- Barlat's anisotropic plasticity,
- Superplastic forming option,
- Rigid body stoppers,
- Keyword input,
- Adaptivity,
- First MPP (Massively Parallel) version with limited capabilities.
- Built in least squares fit for rubber model constitutive constants,
- Large hysteresis in hyperelastic foam,
- Bilhku/Dubois foam model,
- Generalized rubber model,

Capabilities added in 1995:

- Belytschko - Leviathan Shell
- Automatic switching between rigid and deformable bodies.
- Accuracy on SMP machines to give identical answers on one, two or more processors.
- Local coordinate systems for cross-section output can be specified.
- Null material for shell elements.
- Global body force loads now may be applied to a subset of materials.
- User defined loading subroutine.
- Improved interactive graphics.
- New initial velocity options for specifying rotational velocities.
- Geometry changes after dynamic relaxation can be considered for initial velocities..
- Velocities may also be specified by using material or part ID's.
- Improved speed of brick element hourglass force and energy calculations.
- Pressure outflow boundary conditions have been added for the ALE options.
- More user control for hourglass control constants for shell elements.
- Full vectorization in constitutive models for foam, models 57 and 63.
- Damage mechanics plasticity model, material 81,
- General linear viscoelasticity with 6 term prony series.
- Least squares fit for viscoelastic material constants.
- Table definitions for strain rate effects in material type 24.
- Improved treatment of free flying nodes after element failure.
- Automatic projection of nodes in CONTACT_TIED to eliminate gaps in the surface.
- More user control over contact defaults.
- Improved interpenetration warnings printed in automatic contact.
- Flag for using actual shell thickness in single surface contact logic rather than the default.
- Definition by exempted part ID's.
- Airbag to Airbag venting/segmented airbags are now supported.
- Airbag reference geometry speed improvements by using the reference geometry for the time step size calculation.
- Isotropic airbag material may now be directly for cost efficiency.
- Airbag fabric material damping is specified as the ratio of critical damping.

INTRODUCTION

- Ability to attach jets to the structure so the airbag, jets, and structure to move together.
- PVM 5.1 Madymo coupling is available.
- Meshes are generated within LS-DYNA3D for all standard contact entities.
- Joint damping for translational motion.
- Angular displacements, rates of displacements, damping forces, etc. in JNTFORC file.
- Link between LS-NIKE3D to LS-DYNA3D via *INITIAL_STRESS keywords.
- Trim curves for metal forming springback.
- Sparse equation solver for springback.
- Improved mesh generation for IGES and VDA provides a mesh that can directly be used to model tooling in metal stamping analyses.

Capabilities added in 1996-1997 in Version 940:

- Part/Material ID's may be specified with 8 digits.
- Rigid body motion can be prescribed in a local system fixed to the rigid body.
- Nonlinear least squares fit available for the Ogden rubber model.
- Least squares fit to the relaxation curves for the viscoelasticity in rubber.
- Fu-Chang rate sensitive foam.
- 6 term Prony series expansion for rate effects in model 57-now 73
- Viscoelastic material model 76 implemented for shell elements.
- Mechanical threshold stress (MTS) plasticity model for rate effects.
- Thermoelastic-plastic material model for Hughes-Liu beam element.
- Ramberg-Osgood soil model
- Invariant local coordinate systems for shell elements are optional.
- Second order accurate stress updates.
- Four noded, linear, tetrahedron element.
- Co-rotational solid element for foam that can invert without stability problems.
- Improved speed in rigid body to rigid body contacts.
- Improved searching for the a_3, a_5 and a10 contact types.
- Invariant results on shared memory parallel machines with the a_n contact types.
- Thickness offsets in type 8 and 9 tie break contact algorithms.
- Bucket sort frequency can be controlled by a load curve for airbag applications.
- In automatic contact each part ID in the definition may have unique:
 - Static coefficient of friction
 - Dynamic coefficient of friction
 - Exponential decay coefficient
 - Viscous friction coefficient
 - Optional contact thickness
 - Optional thickness scale factor
 - Local penalty scale factor
- Automatic beam-to-beam, shell edge-to-beam, shell edge-to-shell edge and single surface contact algorithm.
- Release criteria may be a multiple of the shell thickness in types a_3, a_5, a10, 13, and 26 contact.
- Force transducers to obtain reaction forces in automatic contact definitions. Defined manually via segments, or automatically via part ID's.
- Searching depth can be defined as a function of time.
- Bucket sort frequency can be defined as a function of time.
- Interior contact for solid (foam) elements to prevent "negative volumes."
- Locking joint
- Temperature dependent heat capacity added to Wang-Nefske inflator models.
- Wang Hybrid inflator model [Wang, 1996] with jetting options and bag-to-bag venting.
- Aspiration included in Wang's hybrid model [Nusholtz, Wang, Wylie, 1996].

INTRODUCTION

- Extended Wang's hybrid inflator with a quadratic temperature variation for heat capacities [Nusholtz, 1996].
- Fabric porosity added as part of the airbag constitutive model .
- Blockage of vent holes and fabric in contact with structure or itself considered in venting with leakage of gas.
- Option to delay airbag liner with using the reference geometry until the reference area is reached.
- Birth time for the reference geometry.
- Multi-material Euler/ALE fluids,
 - 2nd order accurate formulations.
 - Automatic coupling to shell, brick, or beam elements
 - Coupling using LS-DYNA contact options.
 - Element with fluid + void and void material
 - Element with multi-materials and pressure equilibrium
- Nodal inertia tensors.
- 2D plane stress, plane strain, rigid, and axisymmetric elements
- 2D plane strain shell element
- 2D axisymmetric shell element.
- Full contact support in 2D, tied, sliding only, penalty and constraint techniques.
- Most material types supported for 2D elements.
- Interactive remeshing and graphics options available for 2D.
- Subsystem definitions for energy and momentum output.
- Boundary element method for incompressible fluid dynamics and fluid-structure interaction problems.

Capabilities added during 1997-1998 in Version 950:

- Adaptive refinement can be based on tooling curvature with FORMING contact.
- The display of drawbeads is now possible since the drawbead data is output into the D3PLOT database.
- An adaptive box option, *DEFINE_BOX_ADAPTIVE, allows control over the refinement level and location of elements to be adapted.
- A root identification file, ADAPT.RID, gives the parent element ID for adapted elements.
- Draw bead box option,*DEFINE_BOX_DRAWBEAD, simplifies drawbead input.
- The new control option, CONTROL_IMPLICIT, activates an implicit solution scheme.
- 2D Arbitrary-Lagrangian-Eulerian elements are available.
- 2D automatic contact is defined by listing part ID's.
- 2D r-adaptivity for plane strain and axisymmetric forging simulations is available.
- 2D automatic non-interactive rezoning as in LS-DYNA2D.
- 2D plane strain and axisymmetric element with 2x2 selective-reduced integration are implemented.
- Implicit 2D solid and plane strain elements are available.
- Implicit 2D contact is available.
- The new keyword, *DELETE_CONTACT_2DAUTO, allows the deletion of 2D automatic contact definitions.
- The keyword, *LOAD_BEAM is added for pressure boundary conditions on 2D elements.
- A viscoplastic strain rate option is available for materials:
 - *MAT_PLASTIC_KINEMATIC
 - *MAT_JOHNSON_COOK
 - *MAT_POWER_LAW_PLASTICITY
 - *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY
 - *MAT_PIECEWISE_LINEAR_PLASTICITY
 - *MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY

INTRODUCTION

*MAT_ZERILLI-ARMSTRONG

*MAT_PLASTICITY_WITH_DAMAGE

*MAT_PLASTICITY_COMPRESSION_TENSION

- Material model, *MAT_PLASTICITY_WITH_DAMAGE, has a piecewise linear damage curve given by a load curve ID.
- The Arruda-Boyce hyper-viscoelastic rubber model is available, see *MAT_ARRUDA_BOYCE.
- Transverse-anisotropic-viscoelastic material for heart tissue, see *MAT_HEART_TISSUE.
- Lung hyper-viscoelastic material, see *MAT_LUNG_TISSUE.
- Compression/tension plasticity model, see *MAT_PLASTICITY_COMPRESSION_TENSION.
- The Lund strain rate model, *MAT_STEINBERG_LUND, is added to Steinberg-Guinan plasticity model.
- Rate sensitive foam model, *MAT_FU_CHANG_FOAM, has been extended to include engineering strain rates, etc.
- Model, *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY, is added for modeling the failure of aluminum.
- Material model, *MAT_SPECIAL_ORTHOTROPIC, added for television shadow mask problems.
- Erosion strain is implemented for material type, *MAT_BAMMAN_DAMAGE.
- The equation of state, *EOS_JWLB, is available for modeling the expansion of explosive gases.
- The reference geometry option is extended for foam and rubber materials and can be used for stress initialization, see *INITIAL_FOAM_REFERENCE_GEOMETRY.
- A vehicle positioning option is available for setting the initial orientation and velocities, see *INITIAL_VEHICLE_KINEMATICS.
- A boundary element method is available for incompressible fluid dynamics problems.
- The thermal materials work with instantaneous coefficients of thermal expansion:
 - *MAT_ELASTIC_PLASTIC_THERMAL
 - *MAT_ORTHOTROPIC_THERMAL
 - *MAT_TEMPERATURE_DEPENDENT_ORTHOTROPIC
 - *MAT_ELASTIC_WITH_VISCOSITY.
- Airbag interaction flow rate versus pressure differences.
- Contact segment search option, [bricks first optional]
- A through thickness Gauss integration rule with 1-10 points is available for shell elements. Previously, 5 were available.
- Shell element formulations can be changed in a full deck restart.
- The tied interface which is based on constraint equations, TIED_SURFACE_TO_SURFACE, can now fail if _FAILURE, is appended.
- A general failure criteria for solid elements is independent of the material type, see *MAT_ADD_EROSION
- Load curve control can be based on thinning and a flow limit diagram, see *DEFINE_CURVE_FEEDBACK.
- An option to filter the spotweld resultant forces prior to checking for failure has been added the the option, *CONSTRAINED_SPOTWELD, by appending, _FILTERED_FORCE, to the keyword.
- Bulk viscosity is available for shell types 1, 2, 10, and 16.
- When defining the local coordinate system for the rigid body inertia tensor a local coordinate system ID can be used. This simplifies dummy positioning.
- Prescribing displacements, velocities, and accelerations is now possible for rigid body nodes.
- One way flow is optional for segmented airbag interactions.
- Pressure time history input for airbag type, LINEAR_FLUID, can be used.

INTRODUCTION

- An option is available to independently scale system damping by part ID in each of the global directions.
- An option is available to independently scale global system damping in each of the global directions.
- Added option to constrain global DOF along lines parallel with the global axes. The keyword is *CONSTRAINED_GLOBAL. This option is useful for adaptive remeshing.
- Beam end code releases are available, see *ELEMENT_BEAM.
- An initial force can be directly defined for the cable material, *MAT_CABLE_DISCRETE_BEAM. The specification of slack is not required if this option is used.
- Airbag pop pressure can be activated by accelerometers.
- Termination may now be controlled by contact, via *TERMINATION_CONTACT.
- Modified shell elements types 8, 10 and the warping stiffness option in the Belytschko-Tsay shell to ensure orthogonality with rigid body motions in the event that the shell is badly warped. This is optional in the Belytschko-Tsay shell and the type 10 shell.
- A one point quadrature brick element with an exact hourglass stiffness matrix has been implemented for implicit and explicit calculations.
- Automatic file length determination for D3PLOT binary database is now implemented. This insures that at least a single state is contained in each D3PLOT file and eliminates the problem with the states being split between files.
- The dump files, which can be very large, can be placed in another directory by specifying *d=/home/user /test/d3dump* on the execution line.
- A print flag controls the output of data into the MATSUM and RBDOUT files by part ID's. The option, PRINT, has been added as an option to the *PART keyword.
- Flag has been added to delete material data from the D3THDT file. See *DATABASE_EXTENT_BINARY and column 25 of the 19th control card in the structured input.
- After dynamic relaxation completes, a file is written giving the displaced state which can be used for stress initialization in later runs.

Capabilities added during 1998-2000 in Version 960. Most new capabilities work on both the MPP and SMP versions; however, the capabilities that are implemented for the SMP version only, which were not considered critical for this release, are flagged below. These SMP unique capabilities are being extended for MPP calculations and will be available in the near future. The implicit capabilities for MPP require the development of a scalable eigenvalue solver, which is under development for a later release of LS-DYNA.

- Incompressible flow solver is available. Structural coupling is not yet implemented.
- Adaptive mesh coarsening can be done before the implicit springback calculation in metal forming applications.
- Two-dimensional adaptivity can be activated in both implicit and explicit calculations. (SMP version only)
- An internally generated smooth load curve for metal forming tool motion can be activated with the keyword: *DEFINE_CURVE_SMOOTH.
- Torsional forces can be carried through the deformable spot welds by using the contact type: *CONTACT_SPOTWELD_WITH_TORSION (SMP version only with a high priority for the MPP version if this option proves to be stable.)
- Tie break automatic contact is now available via the *CONTACT_AUTOMATIC_..._TIEBREAK options. This option can be used for glued panels. (SMP only)
- *CONTACT_RIGID_SURFACE option is now available for modeling road surfaces (SMP version only).
- Fixed rigid walls PLANAR and PLANAR_FINITE are represented in the binary output file by a single shell element.
- Interference fits can be modeled with the INTERFERENCE option in contact.

INTRODUCTION

- A layered shell theory is implemented for several constitutive models including the composite models to more accurately represent the shear stiffness of laminated shells.
- Damage mechanics is available to smooth the post-failure reduction of the resultant forces in the constitutive model *MAT_SPOTWELD_DAMAGE.
- Finite elastic strain isotropic plasticity model is available for solid elements. *MAT_FINITE_ELASTIC_STRAIN_PLASTICITY.
- A shape memory alloy material is available: *MAT_SHAPE_MEMORY.
- Reference geometry for material, *MAT_MODIFIED_HONEYCOMB, can be set at arbitrary relative volumes or when the time step size reaches a limiting value. This option is now available for all element types including the fully integrated solid element.
- Non orthogonal material axes are available in the airbag fabric model. See *MAT_FABRIC.
- Other new constitutive models include for the beam elements:
 - *MAT_MODIFIED_FORCE_LIMITED
 - *MAT_SEISMIC_BEAM
 - *MAT_CONCRETE_BEAMfor shell and solid elements:
 - *MAT_ELASTIC_VISCOPLASTIC_THERMALfor the shell elements:
 - *MAT_GURSON
 - *MAT_GEPLASTIC_SRATE2000
 - *MAT_ELASTIC_VISCOPLASTIC_THERMAL
 - *MAT_COMPOSITE_LAYUP
 - *MAT_COMPOSITE_LAYUP
 - *MAT_COMPOSITE_DIRECTfor the solid elements:
 - *MAT_JOHNSON_HOLMQUIST_CERAMICS
 - *MAT_JOHNSON_HOLMQUIST_CONCRETE
 - *MAT_INV_HYPERBOLIC_SIN
 - *MAT_UNIFIED_CREEP
 - *MAT_SOIL_BRICK
 - *MAT_DRUCKER_PRAGER
 - *MAT_RC_SHEAR_WALLand for all element options a very fast and efficient version of the Johnson-Cook plasticity model is available:
 - *MAT_SIMPLIFIED_JOHNSON_COOK
- A fully integrated version of the type 16 shell element is available for the resultant constitutive models.
- A nonlocal failure theory is implemented for predicting failure in metallic materials. The keyword *MAT_NONLOCAL activates this option for a subset of elastoplastic constitutive models.
- A discrete Kirchhoff triangular shell element (DKT) for explicit analysis with three in plane integration points is flagged as a type 17 shell element. This element has much better bending behavior than the C0 triangular element.
- A discrete Kirchhoff linear triangular and quadrilateral shell element is available as a type 18 shell. This shell is for extracting normal modes and static analysis.
- A C0 linear 4-node quadrilateral shell element is implemented as element type 20 with drilling stiffness for normal modes and static analysis.
- An assumed strain linear brick element is available for normal modes and statics.
- The fully integrated thick shell element has been extended for use in implicit calculations.
- A fully integrated thick shell element based on an assumed strain formulation is now available. This element uses a full 3D constitutive model which includes the normal stress component and, therefore, does not use the plane stress assumption.

INTRODUCTION

- The 4-node constant strain tetrahedron element has been extended for use in implicit calculations.
- Relative damping between parts is available, see *DAMPING_RELATIVE (SMP only).
- Preload forces can be input for the discrete beam elements.
- Objective stress updates are implemented for the fully integrated brick shell element.
- Acceleration time histories can be prescribed for rigid bodies.
- Prescribed motion for nodal rigid bodies is now possible.
- Generalized set definitions, i.e., SET_SHELL_GENERAL etc. provide much flexibility in the set definitions.
- The command "sw4." will write a state into the dynamic relaxation file, D3DRLF, during the dynamic relaxation phase if the D3DRLF file is requested in the input.
- Added mass by PART ID is written into the MATSUM file when mass scaling is used to maintain the time step size, (SMP version only).
- Upon termination due to a large mass increase during a mass scaled calculation a print summary of 20 nodes with the maximum added mass is printed.
- Eigenvalue analysis of models containing rigid bodies is now available using BCSLIB-EXT solvers from Boeing. (SMP version only).
- Second order stress updates can be activated by part ID instead of globally on the *CONTROL_ACCURACY input.
- Interface frictional energy is optionally computed for heat generation and is output into the interface force file (SMP version only).
- The interface force binary database now includes the distance from the contact surface for the FORMING contact options. This distance is given after the nodes are detected as possible contact candidates. (SMP version only).
- Type 14 acoustic brick element is implemented. This element is a fully integrated version of type 8, the acoustic element (SMP version only).
- A flooded surface option for acoustic applications is available (SMP version only).
- Attachment nodes can be defined for rigid bodies. This option is useful for NVH applications.
- CONSTRAINED_POINTS tie any two points together. These points must lie on a shell elements.
- Soft constraint is available for edge to edge contact in type 26 contact.
- CONSTRAINED_INTERPOLATION option for beam to solid interfaces and for spreading the mass and loads. (SMP version only).
- A database option has been added that allows the output of added mass for shell elements instead of the time step size.
- A new contact option allows the inclusion of all internal shell edges in contact type *CONTACT_GENERAL, type 26. This option is activated by adding _INTERIOR after the GENERAL keyword.
- A new option allows the use deviatoric strain rates rather than total rates in material model 24 for the Cowper-Symonds rate model.
- The CADFEM option for ASCII databases is now the default. Their option includes more significant figures in the output files.
- When using deformable spot welds, the added mass for spot welds is now printed for the case where global mass scaling is activated. This output is in the log file, D3HSP file, and the MESSAG file.
- Initial penetration warnings for edge-to-edge contact are now written into the MESSAG file and the D3HSP file.
- Each compilation of LS-DYNA is given a unique version number.
- Finite length discrete beams with various local axes options are now available for material types 66, 67, 68, 93, and 95. In this implementation the absolute value of SCOOR must be set to 2 or 3 in the *SECTION_BEAM input.
- New discrete element constitutive models are available:
 - *MAT_ELASTIC_SPRING_DISCRETE_BEAM

INTRODUCTION

*MAT_INELASTIC_SPRING_DISCRETE_BEAM
*MAT_ELASTIC_6DOF_SPRING_DISCRETE_BEAM
*MAT_INELASTIC_6DOF_SPRING_DISCRETE_BEAM

The latter two can be used as finite length beams with local coordinate systems.

- Moving SPC's are optional in that the constraints are applied in a local system that rotates with the 3 defining nodes.
- A moving local coordinate system, CID, can be used to determine orientation of discrete beam elements.
- Modal superposition analysis can be performed after an eigenvalue analysis. Stress recovery is based on type 18 shell and brick (SMP only).
- Rayleigh damping input factor is now input as a fraction of critical damping, i.e. 0.10. The old method required the frequency of interest and could be highly unstable for large input values.
- Airbag option "SIMPLE_PRESSURE_VOLUME" allows for the constant CN to be replaced by a load curve for initialization. Also, another load curve can be defined which allows CN to vary as a function of time during dynamic relaxation. After dynamic relaxation CN can be used as a fixed constant or load curve.
- Hybrid inflator model utilizing CHEMKIN and NIST databases is now available. Up to ten gases can be mixed.
- Option to track initial penetrations has been added in the automatic SMP contact types rather than moving the nodes back to the surface. This option has been available in the MPP contact for some time. This input can be defined on the fourth card of the *CONTROL_CONTACT input and on each contact definition on the third optional card in the *CONTACT definitions.
- If the average acceleration flag is active, the average acceleration for rigid body nodes is now written into the D3THDT and NODOUT files. In previous versions of LS-DYNA, the accelerations on rigid nodes were not averaged.
- A capability to initialize the thickness and plastic strain in the crash model is available through the option *INCLUDE_STAMPED_PART, which takes the results from the LS-DYNA stamping simulation and maps the thickness and strain distribution onto the same part with a different mesh pattern.
- A capability to include finite element data from other models is available through the option, *INCLUDE_TRANSFORM. This option will take the model defined in an INCLUDE file: offset all ID's; translate, rotate, and scale the coordinates; and transform the constitutive constants to another set of units.

Many new capabilities were added during 2001-2002 to create version 970 of LS-DYNA. Some of the new features, which are also listed below, were also added to later releases of version 960. Most new explicit capabilities work for both the MPP and SMP versions; however, the implicit capabilities for MPP require the development of a scalable eigenvalue solver and a parallel implementation of the constraint equations into the global matrices. This work is underway. A later release of version 970 is planned in 2003 that will be scalable for implicit solutions.

Below is list of new capabilities and features:

- MPP decomposition can be controlled using *CONTROL_MPP_DECOMPOSITION commands in the input deck.
- The MPP arbitrary Lagrangian-Eulerian fluid capability now works for airbag deployment in both SMP and MPP calculations.
- Euler-to-Euler coupling is now available through the keyword *CONSTRAINED_EULER_TO_EULER.
- Up to ten ALE multi-material groups may now be defined. The previous limit was three groups.
- Volume fractions can be automatically assigned during initialization of multi-material cells. See the GEOMETRY option of *INITIAL_VOLUME_FRACTION.
- A new ALE smoothing option is available to accurately predict shock fronts.

INTRODUCTION

- DATABASE_FSI activates output of fluid-structure interaction data to ASCII file DBFSI.
- Point sources for airbag inflators are available. The origin and mass flow vector of these inflators are permitted to vary with time.
- A majority of the material models for solid materials are available for calculations using the SPH (Smooth Particle Hydrodynamics) option.
- The Element Free Galerkin method (EFG or meshfree) is available for two-dimensional and three-dimensional solids. This new capability is not yet implemented for MPP applications.
- A binary option for the ASCII files is now available. This option applies to all ASCII files and results in one binary file that contains all the information normally spread between a large number of separate ASCII files.
- Material models can now be defined by numbers rather than long names in the keyword input. For example the keyword *MAT_PIECEWISE_LINEAR_PLASTICITY can be replaced by the keyword: *MAT_024.
- An embedded NASTRAN reader for direct reading of NASTRAN input files is available. This option allows a typical input file for NASTRAN to be read directly and used without additional input. See the *INCLUDE_NASTRAN keyword.
- Names in the keyword input can represent numbers if the *PARAMETER option is used to relate the names and the corresponding numbers.
- Model documentation for the major ASCII output files is now optional. This option allows descriptors to be included within the ASCII files that document the contents of the file.
- ID's have been added to the following keywords:
 - *BOUNDARY_PRESCRIBED_MOTION
 - *BOUNDARY_PRESCRIBED_SPC
 - *CONSTRAINED_GENERALIZED_WELD
 - *CONSTRAINED_JOINT
 - *CONSTRAINED_NODE_SET
 - *CONSTRAINED_RIVET
 - *CONSTRAINED_SPOTWELD
 - *DATABASE_CROSS_SECTION
 - *ELEMENT_MASS
- The *DATABASE_ADAMS keyword is available to output a modal neutral file d3mnf. This will be available upon customer request since it requires linking to an ADAMS library file.
- Penetration warnings for the contact option, ignore initial penetration, \hat{i} are added as an option. Previously, no penetration warnings were written when this contact option was activated.
- Penetration warnings for nodes in-plane with shell mid-surface are printed for the AUTOMATIC contact options. Previously, these nodes were ignored since it was assumed that they belonged to a tied interface where an offset was not used; consequently, they should not be treated in contact.
- For the arbitrary spot weld option, the spot welded nodes and their contact segments are optionally written into the D3HSP file. See *CONTROL_CONTACT.
- For the arbitrary spot weld option, if a segment cannot be found for the spot welded node, an option now exists to error terminate. See *CONTROL_CONTACT.
- Spot weld resultant forces are written into the SWFORC file for solid elements used as spot welds.
- Solid materials have now been added to the failed element report.
- A new option for terminating a calculation is available, *TERMINATION_CURVE.
- A 10-noded tetrahedron solid element is available with either a 4 or 5 point integration rule. This element can also be used for implicit solutions.
- A new 4 node linear shell element is available that is based on Wilson's plate element combined with a Pian-Sumihara membrane element. This is shell type 21.

INTRODUCTION

- A shear panel element has been added for linear applications. This is shell type 22. This element can also be used for implicit solutions.
- A null beam element for visualization is available. The keyword to define this null beam is *ELEMENT_PLOTEL. This element is necessary for compatibility with NASTRAN.
- A scalar node can be defined for spring-mass systems. The keyword to define this node is *NODE_SCALAR. This node can have from 1 to 6 scalar degrees-of-freedom.
- A thermal shell has been added for through-thickness heat conduction. Internally, 8 additional nodes are created, four above and four below the mid-surface of the shell element. A quadratic temperature field is modeled through the shell thickness. Internally, the thermal shell is a 12 node solid element.
- A beam OFFSET option is available for the *ELEMENT_BEAM definition to permit the beam to be offset from its defining nodal points. This has the advantage that all beam formulations can now be used as shell stiffeners.
- A beam ORIENTATION option for orienting the beams by a vector instead of the third node is available in the *ELEMENT_BEAM definition for NASTRAN compatibility.
- Non-structural mass has been added to beam elements for modeling trim mass and for NASTRAN compatibility.
- An optional checking of shell elements to avoid abnormal terminations is available. See *CONTROL_SHELL. If this option is active, every shell is checked each time step to see if the distortion is so large that the element will invert, which will result in an abnormal termination. If a bad shell is detected, either the shell will be deleted or the calculation will terminate. The latter is controlled by the input.
- An offset option is added to the inertia definition. See *ELEMENT_INERTIA_OFFSET keyword. This allows the inertia tensor to be offset from the nodal point.
- Plastic strain and thickness initialization is added to the draw bead contact option. See *CONTACT_DRAWBEAD_INITIALIZE.
- Tied contact with offsets based on both constraint equations and beam elements for solid elements and shell elements that have 3 and 6 degrees-of-freedom per node, respectively. See BEAM_OFFSET and CONSTRAINED_OFFSET contact options. These options will not cause problems for rigid body motions.
- The segment-based (SOFT=2) contact is implemented for MPP calculations. This enables airbags to be easily deployed on the MPP version.
- Improvements are made to segment-based contact for edge-to-edge and sliding conditions, and for contact conditions involving warped segments.
- An improved interior contact has been implemented to handle large shear deformations in the solid elements. A special interior contact algorithm is available for tetrahedron elements.
- Coupling with MADYMO 6.0 uses an extended coupling that allows users to link most MADYMO geometric entities with LS-DYNA FEM simulations. In this coupling MADYMO contact algorithms are used to calculate interface forces between the two models.
- Release flags for degrees-of-freedom for nodal points within nodal rigid bodies are available. This makes the nodal rigid body option nearly compatible with the RBE2 option in NASTRAN.
- Fast updates of rigid bodies for metalforming applications can now be accomplished by ignoring the rotational degrees-of-freedom in the rigid bodies that are typically inactive during sheet metal stamping simulations. See the keyword: *CONTROL_RIGID.
- Center of mass constraints can be imposed on nodal rigid bodies with the SPC option in either a local or a global coordinate system.

INTRODUCTION

- Joint failure based on resultant forces and moments can now be used to simulate the failure of joints.
- `CONSTRAINED_JOINT_STIFFNESS` now has a `TRANSLATIONAL` option for the translational and cylindrical joints.
- Joint friction has been added using table look-up so that the frictional moment can now be a function of the resultant translational force.
- The nodal constraint options `*CONSTRAINED_INTERPOLATION` and `*CONSTRAINED_LINEAR` now have a local option to allow these constraints to be applied in a local coordinate system.
- Mesh coarsening can now be applied to automotive crash models at the beginning of an analysis to reduce computation times. See the new keyword: `*CONTROL_COARSEN`.
- Force versus time seatbelt pretensioner option has been added.
- Both static and dynamic coefficients of friction are available for seat belt slip rings. Previously, only one friction constant could be defined.
- `*MAT_SPOTWELD` now includes a new failure model with rate effects as well as additional failure options.
- Constitutive models added for the discrete beam elements:
 - *`MAT_1DOF_GENERALIZED_SPRING`
 - *`MAT_GENERAL_NONLINEAR_6dof_DISCRETE_BEAM`
 - *`MAT_GENERAL_NONLINEAR_1dof_DISCRETE_BEAM`
 - *`MAT_GENERAL_SPRING_DISCRETE_BEAM`
 - *`MAT_GENERAL_JOINT_DISCRETE_BEAM`
 - *`MAT_SEISMIC_ISOLATOR`
- for shell and solid elements:
 - *`MAT_plasticity_with_damage_ortho`
 - *`MAT_simplified_johnson_cook_orthotropic_damage`
 - *`MAT_HILL_3R`
 - *`MAT_GURSON_RCDC`
- for the solid elements:
 - *`MAT_SPOTWELD`
 - *`MAT_HILL_FOAM`
 - *`MAT_WOOD`
 - *`MAT_VISCOELASTIC_HILL_FOAM`
 - *`MAT_LOW_DENSITY_SYNTHETIC_FOAM`
 - *`MAT_RATE_SENSITIVE_POLYMER`
 - *`MAT_QUASILINEAR_VISCOELASTIC`
 - *`MAT_TRANSVERSELY_ANISOTROPIC_CRUSHABLE_FOAM`
 - *`MAT_VACUUM`
 - *`MAT_MODIFIED_CRUSHABLE_FOAM`
 - *`MAT_PITZER_CRUSHABLE_FOAM`
 - *`MAT_JOINTED_ROCK`
 - *`MAT_SIMPLIFIED_RUBBER`
 - *`MAT_FHWA_SOIL`
 - *`MAT_SCHWER_MURRAY_CAP_MODEL`
- Failure time added to `MAT_EROSION` for solid elements.
- Damping in the material models `*MAT_LOW_DENSITY_FOAM` and `*MAT_LOW_DENSITY_VISCOUS_FOAM` can now be a tabulated function of the smallest stretch ratio.
- The material model `*MAT_PLASTICITY_WITH_DAMAGE` allows the table definitions for strain rate.
- Improvements in the option `*INCLUDE_STAMPED_PART` now allow all history data to be mapped to the crash part from the stamped part. Also, symmetry planes can be used to allow the use of a single stamping to initialize symmetric parts.

INTRODUCTION

- Extensive improvements in trimming result in much better elements after the trimming is completed. Also, trimming can be defined in either a local or global coordinate system. This is a new option in *DEFINE_CURVE_TRIM.
- An option to move parts close before solving the contact problem is available, see *CONTACT_AUTO_MOVE.
- An option to add or remove discrete beams during a calculation is available with the new keyword: *PART_SENSOR.
- Multiple jetting is now available for the Hybrid and Chemkin airbag inflator models.
- Nearly all constraint types are now handled for implicit solutions.
- Calculation of constraint and attachment modes can be easily done by using the option: *CONTROL_IMPLICIT_MODES.
- Penalty option, see *CONTROL_CONTACT, now applies to all *RIGIDWALL options and is always used when solving implicit problems.
- Solid elements types 3 and 4, the 4 and 8 node elements with 6 degrees-of-freedom per node are available for implicit solutions.
- The warping stiffness option for the Belytschko-Tsay shell is implemented for implicit solutions. The Belytschko-Wong-Chang shell element is now available for implicit applications. The full projection method is implemented due to its accuracy over the drill projection.
- Rigid to deformable switching is implemented for implicit solutions.
- Automatic switching can be used to switch between implicit and explicit calculations. See the keyword: *CONTROL_IMPLICIT_GENERAL.
- Implicit dynamics rigid bodies are now implemented. See the keyword *CONTROL_IMPLICIT_DYNAMIC.
- Eigenvalue solutions can be intermittently calculated during a transient analysis.
- A linear buckling option is implemented. See the new control input: *CONTROL_IMPLICIT_BUCKLE
- Implicit initialization can be used instead of dynamic relaxation. See the keyword *CONTROL_DYNAMIC_RELAXATION where the parameter, IDFLG, is set to 5.
- Superelements, i.e., *ELEMENT_DIRECT_MATRIX_INPUT, are now available for implicit applications.
- There is an extension of the option, *BOUNDARY_CYCLIC, to symmetry planes in the global Cartesian system. Also, automatic sorting of nodes on symmetry planes is now done by LS-DYNA.
- Modeling of wheel-rail contact for railway applications is now available, see *RAIL_TRACK and *RAIL_TRAIN.
- A new, reduced CPU, element formulation is available for vibration studies when elements are aligned with the global coordinate system. See *SECTION_SOLID and *SECTION_SHELL formulation 98.
- An option to provide approximately constant damping over a range of frequencies is implemented, see *DAMPING_FREQUENCY_RANGE.

Many new capabilities were added during 2003-2005 to create version 971 of LS-DYNA. Initially, the intent was to quickly release version 971 after 970 with the implicit capabilities fully functional for distributed memory processing using MPI. Unfortunately, the effort required for parallel implicit was grossly underestimated, and, as a result, the release has been delayed. Because of the delay, version 971 has turned into a major release. Some of the new features, listed below, were also added to later releases of version 970. The new explicit capabilities are implemented in the MPP version and except for one case, in the SMP version as well.

Below is list of new capabilities and features:

- A simplified method for using the ALE capability with airbags is now available with the keyword *AIRBAG_ALE.
- Case control using the *CASE keyword, which provides a way of running multiple load cases sequentially within a single run

INTRODUCTION

- New option to forming contact: *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH, which use fitted surface in contact calculation.
- Butt weld definition by using the *CONSTRAINED_BUTT_WELD option which makes the definition of butt welds simple relative to the option: *CONSTRAINED_GENERALIZED_WELD_BUTT.
- H-adaptive fusion is now possible as an option with the control input, *CONTROL_ADAPTIVE.
- Added a parameter on, *CONTROL_ADAPTIVE, to specify the number of elements generated around a 90 degree radius. A new option to better calculate the curvature was also implemented.
- Added a new keyword: *CONTROL_ADAPTIVE_CURVE, to refine the element along trimming curves
- Birth and death times for implicit dynamics on the keyword *CONTROL_IMPLICIT_DYNAMICS.
- Added an option to scale the spot weld failure resultants to account for the location of the weld on the segment surface, see *CONTROL_SPOTWELD_BEAM.
- Added an option which automatically replaces a single beam spot weld by an assembly of solid elements using the same ID as the beam that was replaced, see *CONTROL_SPOTWELD_BEAM.
- Boundary constraint in a local coordinate system using *CONSTRAINED_LOCAL keyword.
- A cubic spline interpolation element is now available, *CONSTRAINED_SPLINE.
- Static implicit analyses in of a structure with rigid body modes is possible using the option, *CONTROL_IMPLICIT_INERTIA_RELIEF.
- Shell element thickness updates can now be limited to part ID's within a specified set ID, see the *CONTROL_SHELL keyword. The thickness update for shells can now be optionally limited to the plastic part of the strain tensor for better stability in crash analysis.
- Solid element stresses in spot welds are optionally output in the local system using the SWLOCL parameter on the *CONTROL_SOLID keyword.
- SPOTHIN option on the *CONTROL_CONTACT keyword cards locally thins the spot welded parts to prevent premature breakage of the weld by the contact treatments.
- New function: *CONTROL_FORMING_PROJECT, which can initial move the penetrating slave nodes to the master surface
- New function *CONTROL_FORMING_TEMPLATE, which allows user to easily set up input deck. Its function includes auto-position, define travel curve, termination time, and most of the forming parameters for most of the typical forming process.
- New function *CONTROL_FORMING_USER, *CONTROL_FORMING_POSITION, and *CONTROL_FORMING_TRAVEL, when used together, can allow the user to define atypical forming process.
- Added new contact type *CONTACT_GUIDED_CABLE.
- Circular cut planes are available for *DATABASE_CROSS_SECTION definitions.
- New binary database FSIFOR for fluid structure coupling.
- Added *DATABASE_BINARY_D3PROP for writing the material and property data to the first D3PLOT file or to a new database D3PROP.
- DATABASE_EXTENT_BINARY has new flags to output peak pressure, surface energy density, nodal mass increase from mass scaling, thermal fluxes, and temperatures at the outer surfaces of the thermal shell.
- Eight-character alphanumeric labels can now be used for the parameters SECID, MID, EOSID, HGID, and TMID on the *PART keyword.
- Two NODOUT files are now written: one for high frequency output and a second for low frequency output.
- Nodal mass scaling information can now be optionally written to the D3PLOT file.

INTRODUCTION

- Added option, MASS_PROPERTIES, to include the mass and inertial properties in the GLSTAT and SSSTAT files.
- Added option in *CONTROL_CPU to output the cpu and elapsed time into the GLSTAT file.
- Added an option, IERODE, on the *CONTROL_OUTPUT keyword to include eroded energies by part ID into the MATSUM file. Lumped mass kinetic energy is also in the MATSUM file as part ID 0.
- Added an option, TET10, on the *CONTROL_OUTPUT keyword to output ten connectivity nodes into D3PLOT database rather than 4.
- New keyword, *ELEMENT_SOLID_T4TOT10 to convert 4 node tetrahedron elements to 10 node tetrahedron elements.
- New keyword, *ELEMENT_MASS_PART defines the total additional non-structural mass to be distributed by an area weighted distribution to all nodes of a given part ID.
- New keyword option, SET, for *INITIAL_STRESS_SHELL_SET allows a set of shells to be initialized with the state of stress.
- New option allows the number of cpu's to be specified on the *KEYWORD input.
- Tubular drawbead box option for defining the elements that are included in the drawbead contact, see *DEFINE_BOX_DRAWBEAD.
- New function: *DEFINE_CURVE_DRAWBEAD, allow user to conveniently define drawbead by using curves (in x, y format or iges format)
- New function: *DEFINE_DRAWBEAD_BEAM, which allows user to conveniently define drawbead by using beam part ID, and specify the drawbead force.
- Analytic function can be used in place of load curves with the option *DEFINE_CURVE_FUNCTION.
- Friction can now be defined between part pair using the *DEFINE_FRICTION input.
- New keyword: *DEFINE_CURVE_TRIM_3D, to allow trimming happens based on blank element normal, rather than use pre-defined direction
- A new trimming algorithm was added: *DEFINE_CURVE_TRIM_NEW, which allow seed node to be input and is much faster then the original algorithm.
- A new keyword, *DEFINE_HEX_SPOTWELD_ASSEMBLY, is available to define a cluster of solid elements that comprise a single spot weld.
- The definition of a vector, see *DEFINE_VECTOR, can be done by defining coordinates in a local coordinate system.
- The definition of a failure criteria between part pairs is possible with a table defined using the keyword, *DEFINE_SPOTWELD_FAILURE_RESULTANTS.
- A new keyword, *DEFINE_CONNECTION_PROPERTIES is available for defining failure properties of spot welds.
- Added *DEFINE_SET_ADAPTIVE to allow the adaptive level and element size to be specified by part ID or element set ID.
- Static rupture stresses for beam type spot welds can be defined in the keyword input, *DEFINE_SPOTWELD RUPTURE STRESS.
- Section properties can be define in the *ELEMENT_BEAM definitions for resultant beam elements using the SECTION option.
- Physical offsets of the shell reference surface can be specified on the shell element cards, see the OFFSET option on *ELEMENT_SHELL.
- File names can be located in remote directories and accessed through the *INCLUDE_PART keyword.
- New features to *INCLUDE_STAMPED_PART: two different mirror options, user-defined searching radius.
- *INITIAL_STRESS_SECTION allows for stress initialization across a cross-section, which consists of solid elements.
- An option, IVATN, is available for setting the velocities of slaved nodes and parts for keyword, *INITIAL_VELOCITY_GENERATION.
- Twenty-two built-in cross-section are now available in the definition of beam integration rules, see *INTEGRATION_BEAM.

INTRODUCTION

- The possibility of changing material types is now available for shells using the user defined integration rule, see *INTEGRATION_SHELL.
- The interface springback file created by using the keyword, *INTERFACE_SPRINGBACK is now optionally written as a binary file.
- An optional input line for *KEYWORD allows the definition of a prefix for all file names created during a simulation. This allows multiple jobs to be executed in the same directory.
- Body force loads can now be applied in a local coordinate system for *LOAD_BODY.
- A pressure loading feature allows moving pressures to be applied to a surface to simulate spraying a surface with stream of fluid through a nozzle. See keyword *LOAD_MOVING_PRESSURE.
- Thermal expansion can be added to any material by the keyword, *MAT_ADD_THERMAL_EXPANSION.
- Curves can now be used instead of eight digitized data points in the material model *MAT_ELASTIC_WITH_VISCOSITY_CURVE
- New options for spot weld failure in *MAT_SPOTWELD, which apply to beam and solid elements.
- Failure criteria based on plastic strain to failure is added to material *MAT_ANISOTROPIC_VISCOPLASTIC.
- Strain rate failure criterion is added to material *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY.
- Strain rate scaling of the yield stress can now be done differently in tension and compression in material with separate pressure cut-offs in tension and compression in material model *MAT_PLASTICITY_TENSION_COMPRESSION.
- The RCDC model is now available to predict failure in material *MAT_PLASTICITY_WITH_DAMAGE.
- Two additional yield surfaces have been added to material *MAT_MODIFIED_HONEYCOMB to provide more accurate predictions of the behavior of honeycomb barrier models.
- Unique coordinate systems can be assigned to the two nodal points of material *MAT_1DOF_GENERALIZED_SPRING.
- Poisson's ratio effects are available in foam defined by load curves in the material *MAT_SIMPLIFIED_RUBBER/FOAM
- Failure effects are available in the rubber/foam material defined by load curves in the *MAT_SIMPLIFIED_RUBBER/FOAM_WITH_FAILURE.
- The material option *MAT_ADD_EROSION now allows the maximum pressure at failure and the minimum principal strain at failure to be specified.
- Strains rather than displacements can now be used with the material model for discrete beams, *MAT_GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM.
- New option for MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC (ECHANGE), which allow two ways to change the Young's modulus during forming simulation.
- New Material model: *MAT_HILL_3R: includes the shear term in the yield surface calculation by using Hill's 1948 an-isotropic material model.
- New Material model: *MAT_KINEMATIC_HARDENING_TRANSVERSELY_ANISOTROPIC: which integrates Mat #37 with Yoshida's two-surface kinematic hardening model.
- Improved formulation for the fabric material, *MAT_FABRIC for formulations 2, 3, and 4. The improved formulations are types 12, 13, and 14.
- Constitutive models added for truss elements:
 - *MAT_MUSCLE
- For beam elements
 - *MAT_MOMENT-CURVATURE

For shell elements

- *MAT_RESULTANT_ANISOTROPIC
- *MAT_RATE_SENSITIVE_COMPOSITE_FABRIC.
- *MAT_SAMP-1
- *MAT_SHAPE_MEMORY is now implemented for shells.

for shell and solid elements:

- *MAT_BARLAT_YLD2000 for anisotropic aluminum alloys.
- *MAT_SIMPLIFIED_RUBBER_WITH_DAMAGE
- *MAT_VISCOELASTIC_THERMAL
- *MAT_THERMO_ELASTO_VISCOPLASTIC_CREEP

for the solid elements:

- *MAT_ARUP_ADHESIVE
- *MAT_BRAIN_LINEAR_VISCOELASTIC.
- *MAT_CSCM for modeling concrete.
- *MAT_PLASTICITY_COMPRESSION_TENSION_EOS for modeling ice.
- *MAT_COHESIVE_ELASTIC
- *MAT_COHESIVE_TH
- *MAT_COHESIVE_GENERAL
- *MAT_EOS_GASKET
- *MAT_SIMPLIFIED_JOHNSON_COOK is now implemented for solids.
- *MAT_PLASTICITY_WITH_DAMAGE is now implemented for solids.
- *MAT_SPOTWELD_DAIMLERCHRYSLER

- User defined equations-of-state are now available.
- There is now an interface with the MOLDFLOW code.
- Damping defined in *DAMPING_PART_STIFFNESS now works for the Belytschko-Schwer beam element.
- The option *NODE_TRANSFORMATION allows a node set to be transformed based on a transformation defined in *DEFINE_TRANSFORMATION.
- Parameters can be defined in FORTRAN like expressions using *PARAMETER_EXPRESSION.
- A part can be moved in a local coordinate system in *PART_MOVE.
- A simplified method for defining composite layups is available with *PART_COMPOSITE
- The rigid body inertia can be changed in restart via *CHANGE_RIGID_BODY_INERTIA.
- A part set can now be defined by combining other part sets in *SET_PART_ADD.
- Termination of the calculation is now possible if a specified number of shell elements are deleted in a give part ID. See *TERMINATION_DELETED_SHELLS.
- Added hourglass control type 7 for solid elements for use when modeling hyperelastic materials.
- Shell formulations 4, 11, 16, and 17 can now model rubber materials.
- Added a new seatbelt pretensioner type 7 in which the pretensioner and retractor forces are calculated independently and added.
- A new composite tetrahedron element made up from 12 tetrahedron is now available as solid element type 17.
- Shell thickness offsets for *SECTION_SHELL now works for most shell elements, not just the Hughes-Liu shell.
- The Hughes-Liu beam has been extended to include warpage for open cross-sections.
- A resultant beam formulation with warpage is available as beam type 12.
- Two nonlinear shell elements are available with 8 degrees-of-freedom per node to include thickness stretch.
- Tetrahedron type 13, which uses nodal pressures, is now implemented for implicit applications.
- Cohesive solid elements are now available for treating failure.
- Seatbelt shell elements are available for use with the all seatbelt capabilities.

INTRODUCTION

- Superelements can now share degrees-of-freedom and are implemented for implicit applications under MPI.
- A user defined element interface is available for solid and shell elements.
- Thermal shells are available for treating heat flow through shell elements.
- EFG shell formulations 41 and 42 are implemented for explicit analysis.
- EFGPACK is implemented in addition to BCSLIB-EXT solver on the keyword *CONTROL_EFG.
- EFG MPP version is available for explicit analysis.
- EFG fast transformation method is implemented in the EFG solid formulation.
- EFG Semi-Lagrangian kernel and Eulerian kernel options are added for the foam materials.
- EFG 3D adaptivity is implemented for the metal materials.
- EFG E.O.S. and *MAT_ELASTIC_FLUID materials are included in the 4-noded background element formulation.
- Airbag simulations by using ALE method can be switched to control volume method by *ALE_CV_SWITCH.
- *MAT_ALE_VISCOUS now supports Non-Newtonian viscosity by power law or load curve.
- *DATABASE_BINARY_FSIFOR outputs fluid-structure interaction data to binary file.
- *DATABASE_FSI_SENSOR outputs ALE element pressure to ASCII file dbSor.
- *MAT_GAS_MIXTURE supports nonlinear heat capacities.
- *INITIAL_VOLUME_FRACTION_GEOMETRY uses an enhanced algorithm to handle both concave and convex geometries and substantially reduce run time.
- A new keyword *DELETE_FSI allows the deletion of coupling definitions.
- Convection heat transfer activates by *LOAD_ALE_CONVECTION in ALE FSI analysis.
- *ALE_FSI_SWITCH_MMG is implemented to switch between ALE multi-material groups to treat immersed FSI problems.
- Type 9 option is added in *ALE_REFERENCE_SYSTEM_GROUP to deal complex ALE mesh motions including translation, rotation, expansion and contraction, etc.
- New options in *CONSTRAINED_LAGRANGE_IN_SOLID
 - Shell thickness option for coupling type 4.
 - Bulk modulus based coupling stiffness.
 - Shell erosion treatment.
 - Enable/disable interface force file.
- New coupling method for fluid flowing through porous media are implemented as type 11 (shell) and type 12 (solid) in *CONSTRAINED_LAGRANGE_IN_SOLID.
- *ALE_MODIFIED_STRAIN allows multiple strain fields in certain ALE elements to solve sticking behavior in FSI. (MPP underdevelopment)
- *ALE_FSI_PROJECTION is added as a new constraint coupling method to solve small pressure variation problem. (MPP underdevelopment)
- *BOUNDARY_PRESCRIBED_ORIENTATION_RIGID is added as a means to prescribe as a function of time the general orientation of a rigid body using a variety of methods. This feature is available in release R3 and higher of Version 971.
- *BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID is added as a means to prescribe the motion of a rigid body based un experimental data gathered from accelerometers affixed to the rigid body. . This feature is available in release R3 and higher of Version 971.

MATERIAL MODELS

Some of the material models presently implemented are:

- elastic,
- orthotropic elastic,
- kinematic/isotropic plasticity [Krieg and Key 1976],
- thermoelastoplastic [Hallquist 1979],
- soil and crushable/non-crushable foam [Key 1974],
- linear viscoelastic [Key 1974],
- Blatz-Ko rubber [Key 1974],
- high explosive burn,
- hydrodynamic without deviatoric stresses,
- elastoplastic hydrodynamic,
- temperature dependent elastoplastic [Steinberg and Guinan 1978],
- isotropic elastoplastic,
- isotropic elastoplastic with failure,
- soil and crushable foam with failure,
- Johnson/Cook plasticity model [Johnson and Cook 1983],
- pseudo TENSOR geological model [Sackett 1987],
- elastoplastic with fracture,
- power law isotropic plasticity,
- strain rate dependent plasticity,
- rigid,
- thermal orthotropic,
- composite damage model [Chang and Chang 1987a 1987b],
- thermal orthotropic with 12 curves,
- piecewise linear isotropic plasticity,
- inviscid, two invariant geologic cap [Sandler and Rubin 1979, Simo et al, 1988a 1988b],
- orthotropic crushable model,
- Mooney-Rivlin rubber,
- resultant plasticity,
- force limited resultant formulation,
- closed form update shell plasticity,
- Frazer-Nash rubber model,
- laminated glass model,
- fabric,
- unified creep plasticity,
- temperature and rate dependent plasticity,
- elastic with viscosity,
- anisotropic plasticity,
- user defined,
- crushable cellular foams [Neilsen, Morgan, and Krieg 1987],
- urethane foam model with hysteresis,

INTRODUCTION

and some more foam and rubber models, as well as many materials models for springs and dampers. The hydrodynamic material models determine only the deviatoric stresses. Pressure is determined by one of ten equations of state including:

- linear polynomial [Woodruff 1973],
- JWL high explosive [Dobratz 1981],
- Sack “Tuesday” high explosive [Woodruff 1973],
- Gruneisen [Woodruff 1973],
- ratio of polynomials [Woodruff 1973],
- linear polynomial with energy deposition,
- ignition and growth of reaction in HE [Lee and Tarver 1980, Cochran and Chan 1979],
- tabulated compaction,
- tabulated,
- TENSOR pore collapse [Burton et al. 1982].

The ignition and growth EOS was adapted from KOVEC [Woodruff 1973]; the other subroutines, programmed by the authors, are based in part on the cited references and are nearly 100 percent vectorized. The forms of the first five equations of state are also given in the KOVEC user’s manual and are retained in this manual. The high explosive programmed burn model is described by Giroux [Simo et al. 1988].

The orthotropic elastic and the rubber material subroutines use Green-St. Venant strains to compute second Piola-Kirchhoff stresses, which transform to Cauchy stresses. The Jaumann stress rate formulation is used with all other materials with the exception of one plasticity model which uses the Green-Naghdi rate.

SPATIAL DISCRETIZATION

The elements shown in Figure I.1 are presently available. Currently springs, dampers, beams, membranes, shells, bricks, thick shells and seatbelt elements are included.

The first shell element in DYNA3D was that of Hughes and Liu [Hughes and Liu 1981a, 1981b, 1981c], implemented as described in [Hallquist et al. 1985, Hallquist and Benson 1986]. This element [designated as HL] was selected from among a substantial body of shell element literature because the element formulation has several desirable qualities:

- It is incrementally objective (rigid body rotations do not generate strains), allowing for the treatment of finite strains that occur in many practical applications;
- It is compatible with brick elements, because the element is based on a degenerated brick element formulation. This compatibility allows many of the efficient and effective techniques developed for the DYNA3D brick elements to be used with this shell element;
- It includes finite transverse shear strains;
- A through-the-thickness thinning option (see [Hughes and Carnoy 1981]) is also available.

INTRODUCTION

All shells in our current LS-DYNA code must satisfy these desirable traits to at least some extent to be useful in metalforming and crash simulations.

The major disadvantage of the HL element turned out to be cost related and, for this reason, within a year of its implementation we looked at the Belytschko-Tsay [BT] shell [Belytschko and Tsay 1981, 1983, 1984] as a more cost effective, but possibly less accurate alternative. In the BT shell the geometry of the shell is assumed to be perfectly flat, the local coordinate system originates at the first node of the connectivity, and the co-rotational stress update does not use the costly Jaumann stress rotation. With these and other simplifications, a very cost effective shell was derived which today has become perhaps the most widely used shell elements in both metalforming and crash applications. Results generated by the BT shell usually compare favorably with those of the more costly HL shell. Triangular shell elements are implemented, based on work by Belytschko and co-workers [Belytschko and Marchertas 1974, Bazeley et al. 1965, Belytschko et al. 1984], and are frequently used since collapsed quadrilateral shell elements tend to lock and give very bad results. LS-DYNA automatically treats collapsed quadrilateral shell elements as C^0 triangular elements

Since the Belytschko-Tsay element is based on a perfectly flat geometry, warpage is not considered. Although this generally poses no major difficulties and provides for an efficient element, incorrect results in the twisted beam problem and similar situations are obtained where the nodal points of the elements used in the discretization are not coplanar. The Hughes-Liu shell element considers non-planar geometries and gives good results on the twisted beam. The effect of neglecting warpage in a typical application cannot be predicted beforehand and may lead to less than accurate results, but the latter is only speculation and is difficult to verify in practice. Obviously, it would be better to use shells that consider warpage if the added costs are reasonable and if this unknown effect is eliminated. Another shell published by Belytschko, Wong, and Chiang [Belytschko, Wong, and Chiang 1989, 1992] proposes inexpensive modifications to include the warping stiffness in the Belytschko-Tsay shell. An improved transverse shear treatment also allows the element to pass the Kirchhoff patch test. This element is now available in LS-DYNA. Also, two fully integrated shell elements, based on the Hughes and Liu formulation, are available in LS-DYNA, but are rather expensive. A much faster fully integrated element which is essentially a fully integrated version of the Belytschko, Wong, and Chiang element, type 16, is a more recent addition and is recommended if fully integrated elements are needed due to its cost effectiveness.

Three-dimensional plane stress constitutive subroutines are implemented for the shell elements which iteratively update the stress tensor such that the stress component normal to the shell midsurface is zero. An iterative update is necessary to accurately determine the normal strain component which is necessary to predict thinning. One constitutive evaluation is made for each integration point through the shell thickness.

Zero energy modes in the shell and solid elements are controlled by either an hourglass viscosity or stiffness. Eight node thick shell elements are implemented and have been found to perform well in many applications. All elements are nearly 100% vectorized. All element classes can be included as parts of a rigid body. The rigid body formulation is documented in [Benson and Hallquist 1986]. Rigid body point nodes, as well as concentrated masses, springs and dashpots can be added to this rigid body.

Membrane elements can be either defined directly as shell elements with a membrane formulation option or as shell elements with only one point for through thickness integration. The latter choice includes transverse shear stiffness and may be inappropriate. For airbag material a special fully integrated three and four node membrane element is available.

INTRODUCTION

Two different beam types are available: a stress resultant beam and a beam with cross section integration at one point along the axis. The cross section integration allows for a more general definition of arbitrarily shaped cross sections taking into account material nonlinearities.

Spring and damper elements can be translational or rotational. Many behavior options can be defined, e.g., arbitrary nonlinear behavior including locking and separation.

Solid elements in LS-DYNA may be defined using from 4 to 8 nodes. The standard elements are based on linear shape functions and use one point integration and hourglass control. A selective-reduced integrated (called fully integrated) 8 node solid element is available for situations when the hourglass control fails. Also, two additional solid elements, a 4 noded tetrahedron and an 8 noded hexahedron, with nodal rotational degrees of freedom, are implemented based on the idea of Allman [1984] to replace the nodal midside translational degrees of freedom of the elements with quadratic shape functions by corresponding nodal rotations at the corner nodes. The latter elements, which do not need hourglass control, require many numerical operations compared to the hourglass controlled elements and should be used at places where the hourglass elements fail. However, it is well known that the elements using more than one point integration are more sensitive to large distortions than one point integrated elements.

The thick shell element is a shell element with only nodal translations for the eight nodes. The assumptions of shell theory are included in a non-standard fashion. It also uses hourglass control or selective-reduced integration. This element can be used in place of any four node shell element. It is favorably used for shell-brick transitions, as no additional constraint conditions are necessary. However, care has to be taken to know in which direction the shell assumptions are made; therefore, the numbering of the element is important.

Seatbelt elements can be separately defined to model seatbelt actions combined with dummy models. Separate definitions of seatbelts, which are one-dimensional elements, with accelerometers, sensors, pretensioners, retractors, and slings are possible. The actions of the various seatbelt definitions can also be arbitrarily combined.

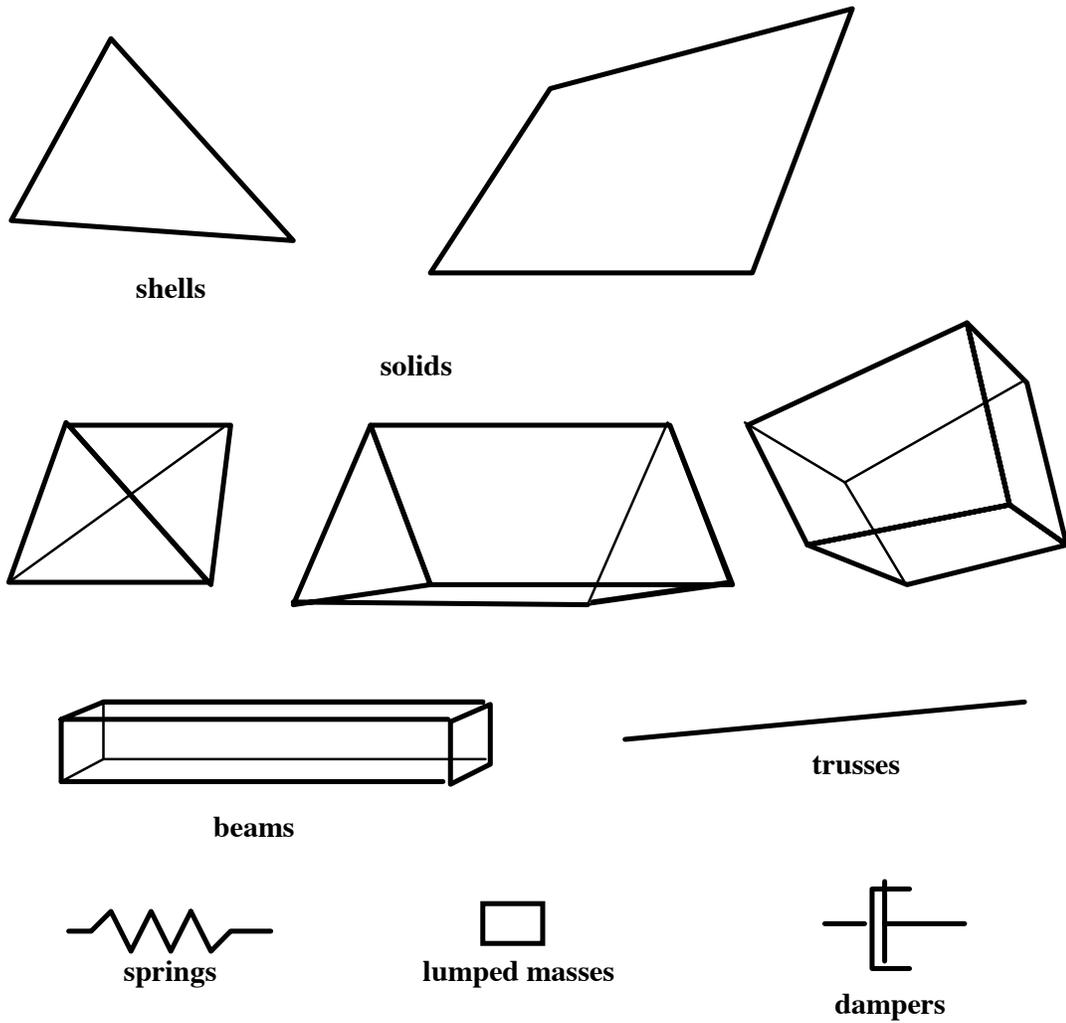


Figure I.1. Elements in LS-DYNA.

INTRODUCTION

CONTACT-IMPACT INTERFACES

The three-dimensional contact-impact algorithm was originally an extension of the NIKE2D [Hallquist 1979] two-dimensional algorithm. As currently implemented, one surface of the interface is identified as a master surface and the other as a slave. Each surface is defined by a set of three or four node quadrilateral segments, called master and slave segments, on which the nodes of the slave and master surfaces, respectively, must slide. In general, an input for the contact-impact algorithm requires that a list of master and slave segments be defined. For the single surface algorithm only the slave surface is defined and each node in the surface is checked each time step to ensure that it does not penetrate through the surface. Internal logic [Hallquist 1977, Hallquist et al. 1985] identifies a master segment for each slave node and a slave segment for each master node and updates this information every time step as the slave and master nodes slide along their respective surfaces. It must be noted that for general automatic definitions only parts/materials or three-dimensional boxes have to be given. Then the possible contacting outer surfaces are identified by the internal logic in LS-DYNA. More than 20 types of interfaces can presently be defined including:

- sliding only for fluid/structure or gas/structure interfaces,
- tied,
- sliding, impact, friction,
- single surface contact,
- discrete nodes impacting surface,
- discrete nodes tied to surface,
- shell edge tied to shell surface,
- nodes spot welded to surface,
- tiebreak interface,
- one way treatment of sliding, impact, friction,
- box/material limited automatic contact for shells,
- automatic contact for shells (no additional input required),
- automatic single surface with beams and arbitrary orientations,
- surface to surface eroding contact,
- node to surface eroding contact,
- single surface eroding contact,
- surface to surface symmetric constraint method [Taylor and Flanagan 1989],
- node to surface constraint method [Taylor and Flanagan 1989],
- rigid body to rigid body contact with arbitrary force/deflection curve,
- rigid nodes to rigid body contact with arbitrary force/deflection curve,
- edge-to-edge,
- draw beads.

Interface friction can be used with most interface types. The tied and sliding only interface options are similar to the two-dimensional algorithm used in LS-DYNA2D [Hallquist 1976, 1978, 1980]. Unlike the general option, the tied treatments are not symmetric; therefore, the surface which is more coarsely zoned should be chosen as the master surface. When using the one-way slide surface with rigid materials, the rigid material should be chosen as the master surface.

For geometric contact entities, contact has to be separately defined. It must be noted that for the contact of a rigid body with a flexible body, either the sliding interface definitions as explained above or the geometric contact entity contact can be used. Currently, the geometric

contact entity definition is recommended for metalforming problems due to high accuracy and computational efficiency.

INTERFACE DEFINITIONS FOR COMPONENT ANALYSIS

Interface definitions for component analyses are used to define surfaces, nodal lines, or nodal points (*INTERFACE_COMPONENTS) for which the displacement and velocity time histories are saved at some user specified frequency (*CONTROL_OUTPUT). This data may then be used to drive interfaces (*INTERFACE_LINKING) in subsequent analyses. This capability is especially useful for studying the detailed response of a small member in a large structure. For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized and interfaces defined to correspond with the first analysis. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest.

When starting the analysis, specify a name for the interface segment file using the Z = parameter on the LS-DYNA command line. When starting the second analysis, the name of the interface segment file (created in the first run) should be specified using the L = parameter on the LS-DYNA command line.

Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA's analysis capability.

CAPACITY

Storage allocation is dynamic. The only limit that exists on the number of boundary condition cards, number of material cards, number of pressure cards, etc., is the capacity of the computer. Typical LS-DYNA calculations may have 10,000 to 500,000 elements. Memory allocation is dynamic and can be controlled during execution.

PRECISION

The explicit time integration algorithms used in LS-DYNA are in general much less sensitive to machine precision than other finite element solution methods. Consequently, double precision is not used. The benefits of this are greatly improved utilization of memory and disk. When problems have been found we have usually been able to overcome them by reorganizing the algorithm or by converting to double precision locally in the subroutine where the problem occurs. A few of the known problems include: **(32-bit computers only!)**:

- Round-off errors can cause difficulties with extremely small deflection problems. (Maximum vibration amplitudes are $<10^{-6}$ times nodal coordinates).
Workaround: Increase the load.
- Buckling problems, which are very sensitive to small imperfections.

However, the users of LS-DYNA have to be aware of potential problems.

A major reorganization of LS-DYNA has led to a version using double precision throughout the full program. As memory and disk space of the computer is less of a problem, we prefer to provide this version for all machines. It also allows LS-DYNA to take advantage of the 64-bit technology offered by many computer manufacturers.

INTRODUCTION

GETTING STARTED

DESCRIPTION OF KEYWORD INPUT

The keyword input provides a flexible and logically organized database that is simple to understand. Similar functions are grouped together under the same keyword. For example, under the keyword **ELEMENT* are included solid, beam, shell elements, spring elements, discrete dampers, seat belts, and lumped masses. Many keywords have options that are identified as follows: “*OPTIONS*” and “*{OPTIONS}*”. The difference is that “*OPTIONS*” requires that one of the options must be selected to complete the keyword command. The option *<BLANK>* is included when *{}* are used to further indicate that these particular options are not necessary to complete the keyword.

LS-DYNA User’s Manual is alphabetically organized in logical sections of input data. Each logical section relates to a particular input. There is a control section for resetting LS-DYNA defaults, a material section for defining constitutive constants, an equation-of-state section, an element section where element part identifiers and nodal connectivities are defined, a section for defining parts, and so on. Nearly all model data can be input in block form. For example, consider the following where two nodal points with their respective coordinates and shell elements with their part identity and nodal connectivities are defined:

```

$      DEFINE TWO NODES
$$$
*NODE
  10101      x      y      z
  10201      x      y      z
$      DEFINE TWO SHELL ELEMENTS
$$$
*ELEMENT_SHELL
  10201      pid    n1    n2    n3    n4
  10301      pid    n1    n2    n3    n4

```

Alternatively, acceptable input could also be of the form:

```

$      DEFINE ONE NODE
$$$
*NODE
  10101      x      y      z
$      DEFINE ONE SHELL ELEMENT
$$$
*ELEMENT_SHELL
  10201      pid    n1    n2    n3    n4
$
$$$
  DEFINE ONE MORE NODE
$$$
*NODE
  10201      x      y      z
$      DEFINE ONE MORE SHELL ELEMENT
$$$
*ELEMENT_SHELL
  10301      pid    n1    n2    n3    n4

```

GETTING STARTED

A data block begins with a keyword followed by the data pertaining to the keyword. The next keyword encountered during the reading of the block data defines the end of the block and the beginning of a new block. A keyword must be left justified with the “*” contained in column one. A dollar sign “\$” in column one precedes a comment and causes the input line to be ignored. Data blocks are not a requirement for LS-DYNA but they can be used to group nodes and elements for user convenience. Multiple blocks can be defined with each keyword if desired as shown above. It would be possible to put all nodal points definitions under one keyword *NODE, or to define one *NODE keyword prior to each node definition. The entire LS-DYNA input is order independent with the exception of the optional keyword, *END, which defines the end of input stream. Without the *END termination is assumed to occur when an end-of-file is encountered during the reading.

Figure GS.1 attempts to show the general philosophy of the input organization and how various entities relate to each other. In this figure the data included for the keyword, *ELEMENT, is the element identifier, EID, the part identifier, PID, and the nodal points identifiers, the NID’s, defining the element connectivity: N1, N2, N3, and N4. The nodal point identifiers are defined in the *NODE section where each NID should be defined just once. A part defined with the *PART keyword has a unique part identifier, PID, a section identifier, SID, a material or constitutive model identifier, MID, an equation of state identifier, EOSID, and the hourglass control identifier, HGID. The *SECTION keyword defines the section identifier, SID, where a section has an element formulation specified, a shear factor, SHRF, a numerical integration rule, NIP, and so on. The constitutive constants are defined in the *MAT section where constitutive data is defined for all element types including solids, beams, shells, thick shells, seat belts, springs, and dampers. Equations of state, which are used only with certain *MAT materials for solid elements, are defined in the *EOS section. Since many elements in LS-DYNA use uniformly reduced numerical integration, zero energy deformation modes may develop. These modes are controlled numerically by either an artificial stiffness or viscosity which resists the formation of these undesirable modes. The hourglass control can optionally be user specified using the input in the *HOURGLASS section.

During the keyword input phase where data is read, only limited checking is performed on the data since the data must first be counted for the array allocations and then reordered. Considerably more checking is done during the second phase where the input data is printed out. Since LS-DYNA has retained the option of reading older non-keyword input files, we print out the data into the output file D3HSP (default name) as in previous versions of LS-DYNA. An attempt is made to complete the input phase before error terminating if errors are encountered in the input. Unfortunately, this is not always possible and the code may terminate with an error message. The user should always check either output file, D3HSP or MESSAG, for the word “Error”.

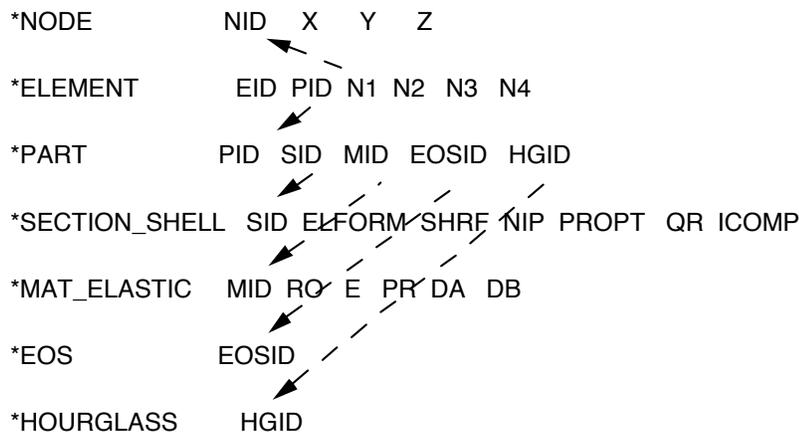


Figure GS.1 Organization of the keyword input.

GETTING STARTED

The input data following each keyword can be input in free format. In the case of free format input the data is separated by commas, i.e.,

*NODE

10101,x ,y ,z

10201,x ,y ,z

*ELEMENT_SHELL

10201,pid,n1,n2,n3,n4

10301,pid,n1,n2,n3,n4

When using commas, the formats **must not** be violated. An I8 integer is limited to a maximum positive value of 99999999, and larger numbers having more than eight characters are unacceptable. The format of the input can change from free to fixed anywhere in the input file. The input is case insensitive and keywords can be given in either upper or lower case. THE ASTERISKS “*” PRECEDING EACH KEYWORD MUST BE IN COLUMN ONE.

To provide a better understanding behind the keyword philosophy and how the options work, a brief review the keywords is given below.

*AIRBAG

The geometric definition of airbags and the thermodynamic properties for the airbag inflator models can be made in this section. This capability is not necessarily limited to the modeling of automotive airbags, but it can also be used for many other applications such as tires and pneumatic dampers.

*ALE

This keyword provides a way of defining input data pertaining to the Arbitrary-Lagrangian-Eulerian capability.

*BOUNDARY

This section applies to various methods of specifying either fixed or prescribed boundary conditions. For compatibility with older versions of LS-DYNA it is still possible to specify some nodal boundary conditions in the *NODE card section.

*CASE

This keyword option provides a way of running multiple load cases sequentially. Within each case, the input parameters, which include loads, boundary conditions, control cards, contact definitions, initial conditions, etc., can change. If desired, the results from a previous case can be used during initialization. Each case creates unique file names for all output results files by appending “**CID n** .” to the default file name.

*COMPONENT

This section contains analytical rigid body dummies that can be placed within vehicle and integrated implicitly.

GETTING STARTED

***CONSTRAINED**

This section applies constraints within the structure between structural parts. For example, nodal rigid bodies, rivets, spot welds, linear constraints, tying a shell edge to a shell edge with failure, merging rigid bodies, adding extra nodes to rigid bodies and defining rigid body joints are all options in this section.

***CONTACT**

This section is divided in to three main sections. The *CONTACT section allows the user to define many different contact types. These contact options are primarily for treating contact of deformable to deformable bodies, single surface contact in deformable bodies, deformable body to rigid body contact, and tying deformable structures with an option to release the tie based on plastic strain. The surface definition for contact is made up of segments on the shell or solid element surfaces. The keyword options and the corresponding numbers in previous code versions are:

STRUCTURED INPUT TYPE ID	KEYWORD NAME
1	SLIDING_ONLY
p 1	SLIDING_ONLY_PENALTY
2	TIED_SURFACE_TO_SURFACE
3	SURFACE_TO_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE
4	SINGLE_SURFACE
5	NODES_TO_SURFACE
a 5	AUTOMATIC_NODES_TO_SURFACE
6	TIED_NODES_TO_SURFACE
7	TIED_SHELL_EDGE_TO_SURFACE
8	TIEBREAK_NODES_TO_SURFACE
9	TIEBREAK_SURFACE_TO_SURFACE
10	ONE_WAY_SURFACE_TO_SURFACE
a 10	AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
13	AUTOMATIC_SINGLE_SURFACE
a 13	AIRBAG_SINGLE_SURFACE
14	ERODING_SURFACE_TO_SURFACE
15	ERODING_SINGLE_SURFACE
16	ERODING_NODES_TO_SURFACE
17	CONSTRAINT_SURFACE_TO_SURFACE
18	CONSTRAINT_NODES_TO_SURFACE
19	RIGID_BODY_TWO_WAY_TO_RIGID_BODY
20	RIGID_NODES_TO_RIGID_BODY
21	RIGID_BODY_ONE_WAY_TO_RIGID_BODY
22	SINGLE_EDGE
23	DRAWBEAD

The *CONTACT_ENTITY section treats contact between a rigid surface, usually defined as an analytical surface, and a deformable structure. Applications of this type of contact exist in the metal forming area where the punch and die surface geometries can be input as VDA surfaces which are treated as rigid. Another application is treating contact between rigid body occupant dummy hyper-ellipsoids and deformable structures such as airbags and instrument panels. This

GETTING STARTED

option is particularly valuable in coupling with the rigid body occupant modeling codes MADYMO and CAL3D. The *CONTACT_1D is for modeling rebars in concrete structure.

***CONTROL**

Options available in the *CONTROL section allow the resetting of default global parameters such as the hourglass type, the contact penalty scale factor, shell element formulation, numerical damping, and termination time.

***DAMPING**

Defines damping either globally or by part identifier.

***DATABASE**

This keyword with a combination of options can be used for controlling the output of ASCII databases and binary files output by LS-DYNA. With this keyword the frequency of writing the various databases can be determined.

***DEFINE**

This section allows the user to define curves for loading, constitutive behaviors, etc.; boxes to limit the geometric extent of certain inputs; local coordinate systems; vectors; and orientation vectors specific to spring and damper elements. Items defined in this section are referenced by their identifiers throughout the input. For example, a coordinate system identifier is sometimes used on the *BOUNDARY cards, and load curves are used on the *AIRBAG cards.

***DEFORMABLE_TO_RIGID**

This section allows the user to switch parts that are defined as deformable to rigid at the start of the analysis. This capability provides a cost efficient method for simulating events such as rollover events. While the vehicle is rotating the computation cost can be reduced significantly by switching deformable parts that are not expected to deform to rigid parts. Just before the vehicle comes in contact with ground, the analysis can be stopped and restarted with the part switched back to deformable.

***EF**

Exchange factors characterize radiative heat transfer between collections of flat surfaces, the union of which is a closed surface (an enclosure). LS-DYNA can calculate exchange factors and then use them as boundary conditions for thermal runs. The $(i,j)^{\text{th}}$ element of an exchange factor matrix, E_{ij} , is the fraction of the Stefan-Boltzman surface energy radiated from surface i that is absorbed by surface j . LS-DYNA employs a Monte Carlo algorithm to calculate these exchange factors. For each surface, LS-DYNA simulates photon emission one photon at a time. For each photon, LS-DYNA generates a random initial position on the emitting surfaces as well as a random initial direction that points into the enclosure. LS-DYNA ray traces each photon until it is absorbed. The path of a simulated photon can be complex involving multiple diffuse and specular reflections as well as multiple diffuse and specular transmissions. The results of this Monte Carlo algorithm are used to assemble a matrix that is related to the exchange factor matrix, for which, the $(i,j)^{\text{th}}$ entry contains the number of photons emitted from surface i that are absorbed by surface j . From this matrix LS-DYNA then assembles the exchange factor matrix.

***ELEMENT**

Define identifiers and connectivities for all elements which include shells, beams, solids, thick shells, springs, dampers, seat belts, and concentrated masses in LS-DYNA.

GETTING STARTED

***EOS**

This section reads the equations of state parameters. The equation of state identifier, EOSID, points to the equation of state identifier on the *PART card.

***HOURLASS**

Defines hourglass and bulk viscosity properties. The identifier, HGID, on the *HOURLASS card refers to HGID on *PART card.

***INCLUDE**

To make the input file easy to maintain, this keyword allows the input file to be split into subfiles. Each subfile can again be split into sub-subfiles and so on. This option is beneficial when the input data deck is very large.

***INITIAL**

Initial velocity and initial momentum for the structure can be specified in this section. The initial velocity specification can be made by *INITIAL_VELOCITY_NODE card or *INITIAL_VELOCITY cards. In the case of *INITIAL_VELOCITY_NODE nodal identifiers are used to specify the velocity components for the node. Since all the nodes in the system are initialized to zero, only the nodes with non-zero velocities need to be specified. The *INITIAL_VELOCITY card provides the capability of being able to specify velocities using the set concept or boxes.

***INTEGRATION**

In this section the user defined integration rules for beam and shell elements are specified. IRID refers to integration rule number IRID on *SECTION_BEAM and *SECTION_SHELL cards respectively. Quadrature rules in the *SECTION_SHELL and *SECTION_BEAM cards need to be specified as a negative number. The absolute value of the negative number refers to user defined integration rule number. Positive rule numbers refer to the built in quadrature rules within LS-DYNA.

***INTERFACE**

Interface definitions are used to define surfaces, nodal lines, and nodal points for which the displacement and velocity time histories are saved at some user specified frequency. This data may then be used in subsequent analyses as an interface ID in the *INTERFACE_LINKING_DISCRETE_NODE as master nodes, in *INTERFACE_LINKING_SEGMENT as master segments and in *INTERFACE_LINKING_EDGE as the master edge for a series of nodes. This capability is especially useful for studying the detailed response of a small member in a large structure. For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized in the region bounded by the interfaces. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest. When beginning the first analysis, specify a name for the interface segment file using the Z=parameter on the LS-DYNA execution line. When starting the second analysis, the name of the interface segment file created in the first run should be specified using the L=parameter on the LS-DYNA command line. Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA's analysis capabilities.

GETTING STARTED

***KEYWORD**

Flags LS-DYNA that the input deck is a keyword deck. To have an effect this must be the very first card in the input deck. Alternatively, by typing “keyword” on the execute line, keyword input formats are assumed and the “*KEYWORD” is not required. If a number is specified on this card after the word KEYWORD it defines the memory size to used in words. The memory size can also be set on the command line. NOTE THAT THE MEMORY SPECIFIED ON THE EXECUTION LINE OVERRIDES MEMORY SPECIFIED ON THE *KEYWORD CARD.

***LOAD**

This section provides various methods of loading the structure with concentrated point loads, distributed pressures, body force loads, and a variety of thermal loadings.

***MAT**

This section allows the definition of constitutive constants for all material models available in LS-DYNA including springs, dampers, and seat belts. The material identifier, MID, points to the MID on the *PART card.

***NODE**

Define nodal point identifiers and their coordinates.

***PARAMETER**

This option provides a way of specifying numerical values of parameter names that are referenced throughout the input file. The parameter definitions, if used, should be placed at the beginning of the input file following *KEYWORD. *PARAMETER_EXPRESSION permits general algebraic expressions to be used to set the values.

***PART**

This keyword serves two purposes.

1. Relates part ID to *SECTION, *MATERIAL, *EOS and *HOURLASS sections.
2. Optionally, in the case of a rigid material, rigid body inertia properties and initial conditions can be specified. Deformable material repositioning data can also be specified in this section if the reposition option is invoked on the *PART card, i.e., *PART_REPOSITION.

***PERTURBATION**

This keyword provides a way of defining deviations from the designed structure such as, buckling imperfections.

***RAIL**

This keyword provides a way of defining a wheel-rail contact algorithm intended for railway applications but can also be used for other purposes. The wheel nodes (defined on *RAIL_TRAIN) represent the contact patch between wheel and rail.

***RIGIDWALL**

Rigid wall definitions have been divided into two separate sections, _PLANAR and _GEOMETRIC. Planar walls can be either stationary or moving in translational motion with mass and initial velocity. The planar wall can be either finite or infinite. Geometric walls can be planar as well as have the geometric shapes such as rectangular prism, cylindrical prism and sphere. By default, these walls are stationary unless the option MOTION is invoked for either prescribed translational velocity or displacement. Unlike the planar walls, the motion of the

GETTING STARTED

geometric wall is governed by a load curve. Multiple geometric walls can be defined to model combinations of geometric shapes available. For example, a wall defined with the `_CYLINDER` option can be combined with two walls defined with the `_SPHERICAL` option to model hemispherical surface caps on the two ends of a cylinder. Contact entities are also analytical surfaces but have the significant advantage that the motion can be influenced by the contact to other bodies, or prescribed with six full degrees-of-freedom.

***SECTION**

In this section, the element formulation, integration rule, nodal thicknesses, and cross sectional properties are defined. All section identifiers (SECID's) defined in this section must be unique, i.e., if a number is used as a section ID for a beam element then this number cannot be used again as a section ID for a solid element.

***SENSOR**

This keyword provides a convenient way of activating and deactivating boundary conditions, airbags, discrete elements, joints, contact, rigid walls, single point constraints, and constrained nodes. The sensor capability is new in the second release of version 971 and will evolve in later releases to encompass many more LS-DYNA capabilities and replace some of the existing capabilities such as the airbag sensor logic.

***SET**

A concept of grouping nodes, elements, materials, etc., in sets is employed throughout the LS-DYNA input deck. Sets of data entities can be used for output. So-called slave nodes used in contact definitions, slaves segment sets, master segment sets, pressure segment sets and so on can also be defined. The keyword, `*SET`, can be defined in two ways:

1. Option `_LIST` requires a list of entities, eight entities per card, and define as many cards as needed to define all the entities.
2. Option `_COLUMN`, where applicable, requires an input of one entity per line along with up to four attribute values which are needed to specify, for example, failure criterion input that is needed for `*CONTACT_CONSTRAINT_NODES_TO_SURFACE`.

***TERMINATION**

This keyword provides an alternative way of stopping the calculation before the termination time is reached. The termination time is specified on the `*CONTROL_TERMINATION` input and will terminate the calculation whether or not the options available in this section are active.

***TITLE**

In this section a title for the analysis is defined.

***USER_INTERFACE**

This section provides a method to provide user control of some aspects of the contact algorithms including friction coefficients via user defined subroutines.

RESTART

This section of the input is intended to allow the user to restart the simulation by providing a restart file and optionally a restart input defining changes to the model such as deleting contacts, materials, elements, switching materials from rigid to deformable, deformable to rigid, etc.

GETTING STARTED

***RIGID_TO_DEFORMABLE**

This section switches rigid parts back to deformable in a restart to continue the event of a vehicle impacting the ground which may have been modeled with a rigid wall.

***STRESS_INITIALIZATION**

This is an option available for restart runs. In some cases there may be a need for the user to add contacts, elements, etc., which are not available options for standard restart runs. A full input containing the additions is needed if this option is invoked upon restart.

SUMMARY OF COMMONLY USED OPTIONS

The following table gives a list of the commonly used keywords related by topic.

Table GS.1. Keywords for the most commonly used options.

Topic	Component	Keyword
Geometry	Nodes Elements Discrete Elements	*NODE *ELEMENT_BEAM *ELEMENT_SHELL *ELEMENT_SOLID *ELEMENT_TSHELL *ELEMENT_DISCRETE *ELEMENT_MASS *ELEMENT_SEATBELT_ <i>Option</i>
Materials	Part (which is composed of Material and Section, equation of state and hourglass data) Material Sections Discrete sections Equation of state Hourglass	*PART *MAT_ <i>Option</i> *SECTION_BEAM *SECTION_SHELL *SECTION_SOLID *SECTION_TSHELL *SECTION_DISCRETE *SECTION_SEATBELT *EOS_ <i>Option</i> *CONTROL_HOURLASS *HOURLASS
Contacts and Rigid walls	Defaults for contacts Definition of contacts Definition of rigid walls	*CONTROL_CONTACT *CONTACT_ <i>Option</i> *RIGIDWALL_ <i>Option</i>

GETTING STARTED

Table GS.1. (continued) Keywords for the most commonly used options.

Topic	Component	Keyword
Boundary Conditions & Loadings	Restraints	*NODE
		*BOUNDARY_SPC_ <i>Option</i>
	Gravity (body) load	*LOAD_BODY_ <i>Option</i>
	Point load	*LOAD_NODE_ <i>Option</i>
	Pressure load	*LOAD_SEGMENT_ <i>Option</i>
		*LOAD_SHELL_ <i>Option</i>
	Thermal load	*LOAD_THERMAL_ <i>Option</i>
	Load curves	*DEFINE_CURVE
Constraints and spot welds	Constrained nodes	*CONSTRAINED_NODE_SET
	Welds	*CONSTRAINED_GENERALIZED_WELD_ <i>Option</i>
		*CONSTRAINED_SPOT_WELD
	Rivet	*CONSTRAINED_RIVET
Output Control	Defaults	*CONTROL_OUTPUT
	ASCII time history files	*DATABASE_ <i>Option</i>
	Binary plot, time history and restart files	*DATABASE_BINARY_ <i>Option</i>
	Items in time history blocks	*DATABASE_HISTORY_ <i>Option</i>
	Nodes for nodal reaction output	*DATABASE_NODAL_FORCE_GROUP
Termination	Termination time	*CONTROL_TERMINATION
	Termination cycle	*CONTROL_TERMINATION
	CPU termination	*CONTROL_CPU
	Degree of freedom	*TERMINATION_NODE

EXECUTION SYNTAX

The interactive execution line for LS-DYNA is as follows:

```
LS-DYNA I=inf O=otf G=ptf D=dpf F=thf U=xtf T=tpf A=rrd M=sif J=jif S=iff Z=isf1
L=isf2 B=rlf W=root E=efl X=scl C=cpu K=kill V=vda Y=c3d BEM=bof {KEYWORD}
{THERMAL} {COUPLE} {INIT} MEMORY=nwds NCPU= ncpu PARA=para
ENDTIME=time NCYCLE=ncycle JOBID=jobid D3PROP=d3prop
```

where

- inf** = input file (user specified)
- otf** = high speed printer file (default=D3HSP)
- ptf** = binary plot file for graphics (default=D3PLOT)
- dpf** = dump file for restarting (default=D3DUMP). This file is written at the end of every run and during the run as requested in the input. To stop the generation of this file set the file name to NODUMP.
- thf** = binary plot file for time histories of selected data (default=D3THDT)

GETTING STARTED

- xtf** = binary plot file for time extra data (default=XTFILE)
- tpf** = optional temperature file
- rrd** = running restart dump file (default=RUNRSF)
- sif** = stress initialization file (user specified)
- jif** = optional JOY interface file
- iff** = interface force file (user specified)
- isf1** = interface segment save file to be created (user specified)
- isf2** = existing interface segment save file to be used (user specified)
- rlf** = binary plot file for dynamic relaxation (default=D3DRFL)
- efl** = echo file containing optional input echo with or without node/element data
- root** = root file name for general print option
- scl** = scale factor for binary file sizes (default=7)
- cpu** = cpu limit in seconds, applies to total calculation not just cpu from a restart
- kill** = if LS-DYNA encounters this file name it will terminate with a restart file (default=D3KIL)
- vda** = VDA/IGES database for geometrical surfaces
- c3d** = CAL3D input file
- bof** = *BOUNDARY_ELEMENT_METHOD_ACOUSTIC output file
- nwds** = Number of words to be allocated. On engineering workstations a word is usually 32bits. **This number overwrites the memory size specified on the *KEYWORD card at the beginning of the input deck.**
- ncpu** = Overrides **NCPU** and **CONST** defined in *CONTROL_PARALLEL. A positive value sets CONST=2 and a negative values sets CONST=1. See the *CONTROL_PARALLEL command for an explanation of these parameters. The *KEYWORD command provides an alternative way to set the number of CPUs.
- npara** = Overrides **PARA** defined in *CONTROL_PARALLEL.
- time** = Overrides **ENDTIM** defined in *CONTROL_TERMINATION.
- ncycle** = Overrides **ENDCYC** defined in *CONTROL_TERMINATION.
- jobid** = Character string which acts as a prefix for all output files. Maximum length is 72 characters. **Do not** include the following characters:)(* / ? \
- d3prop** = See *DATABASE_BINARY_D3PROP input parameter IFILE for options.

In order to avoid undesirable or confusing results, each LS-DYNA run should be performed in a separate directory, unless using the command line parameter “jobid” described above. If rerunning a job in the same directory, old files should first be removed or renamed to avoid confusion since the possibility exists that the binary database may contain results from both the old and new run.

By including **KEYWORD** anywhere on the execute line or instead if ***KEYWORD** is the first card in the input file, the keyword formats are expected; otherwise, the older structured input file will be expected.

To run a coupled thermal analysis the command **COUPLE** must be in the execute line. A thermal only analysis may be run by including the word **THERMAL** in the execution line.

The **INIT** (or **sw1**. can be used instead) command on the execution line causes the calculation to run just one cycle followed by termination with a full restart file. No editing of the input deck is required. The calculation can then be restarted with or without any additional input. Sometimes this option can be used to reduce the memory on restart if the required memory is given on the execution line and is specified too large in the beginning when the amount of

GETTING STARTED

required memory is unknown. Generally, this option would be used at the beginning of a new calculation.

If the word **MEMORY** is found anywhere on the execution line and if it is not set via (**=nwds**) LS-DYNA will give the default size of memory, request, and then read in the desired memory size. This option is necessary if the default value is insufficient memory and termination occurs as a result. Occasionally, the default value is too large for execution and this option can be used to lower the default size. Memory can also be specified on the ***KEYWORD** card.

SENSE SWITCH CONTROLS

The status of an in-progress LS-DYNA simulation can be determined by using the sense switch. On UNIX versions, this is accomplished by first typing a “^C” (Control-C). This sends an interrupt to LS-DYNA which is trapped and the user is prompted to input the sense switch code. LS-DYNA has nine terminal sense switch controls that are tabulated below:

<u>Type</u>	<u>Response</u>
SW1.	A restart file is written and LS-DYNA terminates.
SW2.	LS-DYNA responds with time and cycle numbers.
SW3.	A restart file is written and LS-DYNA continues.
SW4.	A plot state is written and LS-DYNA continues.
SW5.	Enter interactive graphics phase and real time visualization.
SW7.	Turn off real time visualization.
SW8.	Interactive 2D rezoner for solid elements and real time visualization.
SW9.	Turn off real time visualization (for option SW8).
SWA.	Flush ASCII file buffers.

<u>Type</u>	<u>Response (Implicit Mode Only)</u>
lprint	Enable/Disable printing of equation solver memory, cpu requirements.
nlprint	Enable/Disable printing of nonlinear equilibrium iteration information.
iter	Enable/Disable output of binary plot database "d3iter" showing mesh after each equilibrium iteration. Useful for debugging convergence problems.
conv	Temporarily override nonlinear convergence tolerances.
stop	Halt execution immediately, closing open files.

On UNIX/LINUX systems the sense switches can still be used if the job is running in the background or in batch mode. To interrupt LS-DYNA simply create a file called D3KIL containing the desired sense switch, e.g., "sw1." LS-DYNA periodically looks for this file and if found, the sense switch contained therein is invoked and the D3KIL file is deleted. A null D3KIL file is equivalent to a "sw1."

When LS-DYNA terminates, all scratch files are destroyed: the restart file, plot files, and high-speed printer files remain on disk. Of these, only the restart file is needed to continue the interrupted analysis.

GETTING STARTED

Procedure for LS-DYNA/MPP

As described above the serial/SMP code supports the use of the SIGINT signal (usually Ctrl-C) to interrupt the execution and prompt the user for a "sense switch." The MPP code also supports this capability. However, on many systems a shell script or front end program (generally "mpirun") is required to start MPI applications. Pressing Ctrl-C on some systems will kill this process, and thus kill the running MPP-DYNA executable. As a workaround, when the MPP code begins execution it creates a file named "bg_switch" in the current working directory. This file contains the following single line:

```
rsh <machine name> kill -INT <PID>
```

where <machine name> is the hostname of the machine on which the root MPP-DYNA process is running, and <PID> is its process id. (on HP systems, "rsh" is replaced by "remsh"). Thus, simply executing this file will send the appropriate signal.

For more information about running the LS-DYNA/MPP Version see Appendix O.

Files: Input and Output

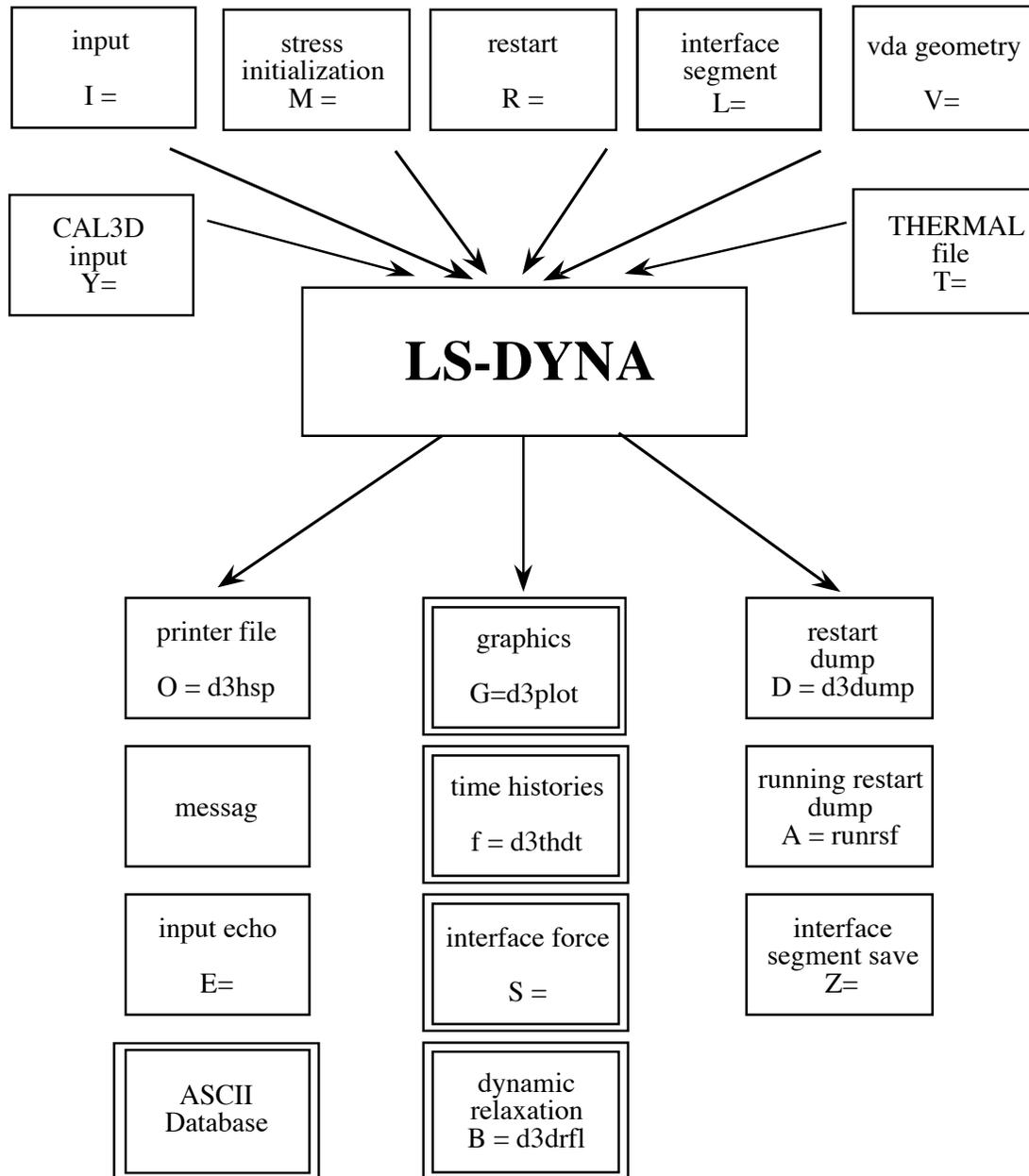


Figure GS.2

GETTING STARTED

File names must be unique. The interface force file is created only if it is specified on the execution line (S=iff). On large problems the default file sizes may not be large enough for a single file to hold either a restart dump or a plot state. Then the file size may be increased by specifying the file size on the execute line using X=scl. The default file size holds seven times one-million octal word (262144) or 1835008 words. If the core required by LS-DYNA requires more space, it is recommended that the scl be increased appropriately. Using C=cpu defines the maximum cpu usage allowed that if exceeded will cause LS-DYNA to terminate with a restart file. During a restart, cpu should be set to the total cpu used up to the current restart plus whatever amount of additional time is wanted.

When **restarting from a dump file**, the execution line becomes

```
LS-DYNA I=inf O=otf G=ptf D=dpf R=rtf F=thf U=xtf T=tpf A=rrd J=jif S=iff Z=isf1 L=isf2  
B=rlf W=root E=efl X=scl C=cpu K=kill Q=option KEYWORD MEMORY=nwds
```

where

rtf = restart filename.

The adaptive dump files contain all information required to successfully restart so that no other files are needed except when CAD surface data is used. When restarting a problem that uses VDA/IGES surface data, the vda input file must be specified, e.g.:

```
LS-DYNA R=d3dump01 V=vda .....
```

If the data from the last run is to be remapped onto a new mesh, then specify: Q=remap. The remap file is the dump file from which the remapping data is taken. The remap option is available for brick elements only. File name dropouts are permitted; for example, the following execution lines are acceptable.

```
LS-DYNA I=inf  
LS-DYNA R=rtf
```

Default names for the output file, binary plot files, and the dump file are D3HSP, D3PLOT, D3THDT, and D3DUMP, respectively.

For an analysis using interface segments the execution line in the first analysis is given by:

```
LS-DYNA I=inf Z=isf1
```

and in the second by:

```
LS-DYNA I=inf L=isf1
```

Batch execution in some installations (e.g., GM) is controlled by file NAMES on unit 88. NAMES is a 2 line file in which the second line is blank. The first line of NAMES contains the execution line:

```
I=inf
```

if this is the initial run. For a restart the execution line becomes:

```
I=inf R=rtf
```

Remark: No stress initialization is possible at restart. Also the VDA files and the CAL3D files cannot be changed.

GETTING STARTED

RESTART ANALYSIS

The LS-DYNA restart capability allows analyses to be broken down into stages. After the completion of each stage in the calculation a “restart dump” is written that contains all information necessary to continue the analysis. The size of this “dump” file is roughly the same size as the memory required for the calculation. Results can be checked at each stage by post-processing the output databases in the normal way, so the chance of wasting computer time on incorrect analyses is reduced. The restart capability is frequently used to modify models by deleting excessively distorted elements, materials that are no longer important, and contact surfaces that are no longer needed. Output frequencies of the various databases can also be altered. Often, these simple modifications permit the calculation to continue on to a successful completion. Restarting can also help to diagnose why a model is giving problems. By restarting from a dump that is written before the occurrence of a numerical problem and obtaining output at more frequent intervals, it is often possible to identify where the first symptoms appear and what aspect of the model is causing them.

The format of the restart input file is described in this manual. If, for example, the user wishes to restart the analysis from dump state *nm*, contained in file *D3DUMPnm*, then the following procedure is followed:

1. Create the restart input deck, if required, as described in the Restart Section of this manual. Call this file *restartinput*.
2. By invoking the execution line:

```
LS-DYNA I=restartinput R=D3DUMPnm
```

execution begins. If no alterations to the model are made, then the execution line:

```
LS-DYNA R=D3DUMPnm
```

will suffice. Of course, the other output files should be assigned names if the defaults have been changed in the original run.

The R=D3DUMPnm on the status line informs the program that this is a restart analysis.

The full deck restart option allows the user to begin a new analysis, with deformed shapes and stresses carried forward from a previous analysis for selected materials. The new analysis can be different from the original, e.g., more contact surfaces, different geometry (of parts which are not carried forward), etc. Examples of applications include:

- Crash analysis continued with extra contact surfaces;
- Sheet metalforming continued with different tools for modeling a multi-stage forming process.

Assume an analysis is run using the input file, *job1.inf*, and a restart dump named *d3dump01* is created. A new input file *job2.inf* is generated and submitted as a restart with R=d3dump01 as the dump file. The input file *job2.inf* contains the entire model in its original undeformed state but with more contact surfaces, new output databases, and so on. Since this is a restart job, information must be given to tell LS-DYNA which parts of the model should be initialized in the full deck restart. When the calculation begins the restart database contained in

GETTING STARTED

the file d3dump01 is read, and a new database is created to initialize the model in the input file, job2.inf. The data in file job2.inf is read and the LS-DYNA proceeds through the entire input deck and initialization. At the end of the initialization process, all the parts selected are initialized from the data saved from d3dump01. This means that the deformed position and velocities of the nodes on the elements of each part, and the stresses and strains in the elements (and, if the material of the part is rigid, the rigid body properties) will be assigned.

It is assumed during this process that any initialized part has the same elements, in the same order, with the same topology, in job1 and job2. If this is not the case, the parts cannot be initialized. However, the parts may have different identifying numbers.

For discrete elements and seat belts, the choice is all or nothing. All discrete and belt elements, retractors, slings, pretensioners and sensors must exist in both files and will be initialized.

Materials which are not initialized will have no initial deformations or stresses. However, if initialized and non-initialized materials have nodes in common, the nodes will be moved by the initialized material causing a sudden strain in the non-initialized material. This effect could give rise to sudden spikes in loading.

Points to note are:

- Time and output intervals are continuous with job1, i.e., the time is not reset to zero.
- Don't try to use the restart part of the input to change anything since this will be overwritten by the new input file.
- Usually, the complete input file part of job2.in1 will be copied from job1.inf, with the required alterations. We again mention that there is no need to update the nodal coordinates since the deformed shapes of the initialized materials will be carried forward from job1.
- Completely new databases will be generated with the time offset.

VDA/IGES DATABASES

VDA surfaces are surfaces of geometric entities which are given in the form of polynomials. The format of these surfaces is as defined by the German automobile and supplier industry in the VDA guidelines, [VDA 1987].

The advantage of using VDA surfaces is twofold. First, the problem of meshing the surface of the geometric entities is avoided and, second, smooth surfaces can be achieved which are very important in metalforming. With smooth surfaces, artificial friction introduced by standard faceted meshes with corners and edges can be avoided. This is a big advantage in springback calculations.

A very simple and general handling of VDA surfaces is possible allowing arbitrary motion and generation of surfaces. For a detailed description, see Appendix L.

GETTING STARTED

LS-PrePost®

LS-DYNA is designed to operate with a variety of commercial pre- and post-processing packages. Currently, direct support is available from TRUEGRID, PATRAN, eta/VPG, HYPERMESH, EASi-CRASH DYNA and FEMAP. Several third-party translation programs are available for PATRAN and IDEAS.

Alternately, the pre- and post-processor LS-PrePost is available from LSTC and is specialized for LS-DYNA. LS-PrePost is an advanced pre- and post-processor that is delivered free with LS-DYNA. The user interface is designed to be both efficient and intuitive. LS-PrePost runs on Windows, Linux, and Unix, utilizing OpenGL graphics to achieve fast model rendering and XY plotting.

Some of the capabilities available in LS-PrePost are:

- Complete support for all LS-DYNA keyword data.
- Importing and combining multiple models from many sources (LS-DYNA keyword, IDEAS neutral file, NASTRAN bulk data, STL ascii, and STL binary formats).
- Improved renumbering of model entities.
- Model Manipulation: Translate, Rotate, Scale, Project, Offset, Reflect
- LS-DYNA Entity Creation: Coordinate Systems, Sets, Parts, Masses, CNRBs, Boxes, Spot welds, SPCs, Rigidwalls, Rivets, Initial Velocity, Accelerometers, Cross Sections, etc.
- Mesh Generation: 2Dmesh Sketchboard, nLine Meshing, Line sweep into shell, Shell sweep into solid, Tet-Meshing, Automatic surface meshing of IGES and VDA data, Meshing of simple geometric objects (Plate, Sphere, Cylinder)
- Special Applications: Airbag folding, Dummy positioning, Seatbelt fitting, Initial penetration check, Spot weld generation using MAT_100
- Complete support of LS-DYNA results data file: d3plot file, d3thdt file, All ascii time history data file, Interface force file

LS-PrePost processes output from LS-DYNA. LS-PrePost reads the binary plot-files generated by LS-DYNA and plots contours, fringes, time histories, and deformed shapes. Color contours and fringes of a large number of quantities may be interactively plotted on meshes consisting of plate, shell, and solid type elements. LS-PrePost can compute a variety of strain measures, reaction forces along constrained boundaries.

LS-DYNA generates three binary databases. One contains information for complete states at infrequent intervals; 50 to 100 states of this sort is typical in a LS-DYNA calculation. The second contains information for a subset of nodes and elements at frequent intervals; 1000 to 10,000 states is typical. The third contains interface data for contact surfaces.

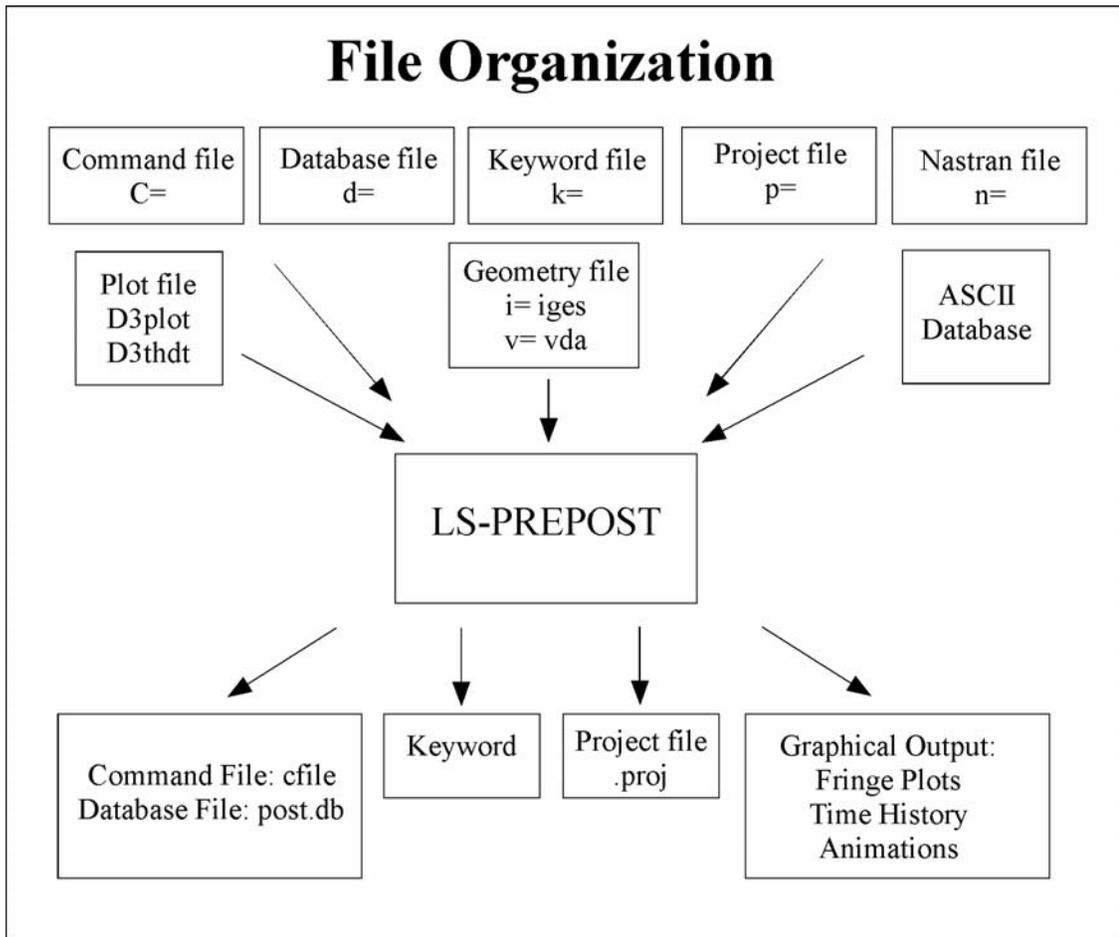


Figure GS.3

GETTING STARTED

EXECUTION SPEEDS

The relative execution speeds for various elements in LS-DYNA are tabulated below:

Element Type	Relative Cost
8 node solid with 1 point integration and default hourglass control	4
as above but with Flanagan-Belytschko hourglass control	5
constant stress and Flanagan-Belytschko hourglass control, i.e., the Flanagan-Belytschko element	7
4 node Belytschko-Tsay shell with four thickness integration points	4
4 node Belytschko-Tsay shell with resultant plasticity	3
BCIZ triangular shell with four thickness integration points	7
C ^o triangular shell with four thickness integration points	4
2 node Hughes-Liu beam with four integration points	9
2 node Belytschko-Schwer beam	2
2 node simple truss elements	1
8 node solid-shell with four thickness integration points	11

These relative timings are very approximate. Each interface node of the sliding interfaces is roughly equivalent to one-half zone cycle in cost. Figure GS.3 illustrates the relative cost of the various shell formulations in LS-DYNA.

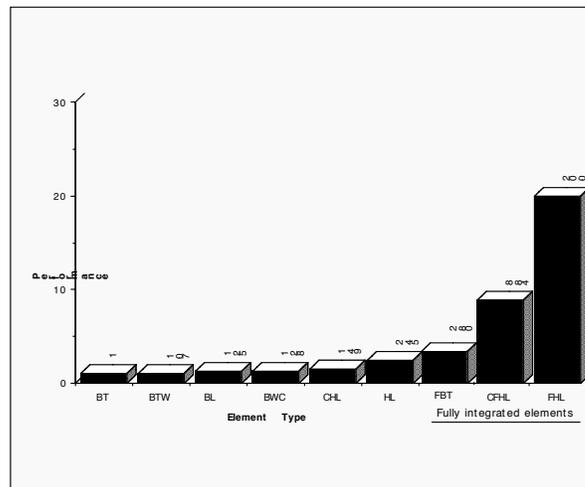


Figure GS.4 Relative cost of the four noded shells available in LS-DYNA where BT is the Belytschko-Tsay shell, BTW is the Belytschko-Tsay shell with the warping stiffness taken from the Belytschko-Wong-Chiang, BWC, shell. The BL shell is the Belytschko-Leviathan shell. CHL denotes the Hughes-Liu shell, HL, with one point quadrature and a co-rotational formulation. FBT is a Belytschko-Tsay like shell with full integration, FHL is the fully integrated Hughes-Liu shell, and the CFHL shell is its co-rotational version.

GETTING STARTED

UNITS

The units in LS-DYNA must be consistent. One way of testing whether a set of units is consistent is to check that:

$$1 \text{ (force unit)} = 1 \text{ (mass unit)} \times 1 \text{ (acceleration unit)}$$

$$\text{and that } 1 \text{ (acceleration unit)} = \frac{1(\text{lengthunit})}{[1(\text{timeunit})]^2}$$

Examples of sets of consistent units are:

	(a)	(b)	(c)
Length unit	meter	millimeter	millimeter
Time unit	second	second	millisecond
Mass unit	kilogram	tonne	kilogram
Force unit	Newton	Newton	kiloNewton
Young's Modulus of Steel	210.0E+09	210.0E+03	210.0
Density of Steel	7.85E+03	7.85E-09	7.85E-06
Yield stress of Mild Steel	200.0E+06	200.0	0.200
Acceleration due to gravity	9.81	9.81E+03	9.81E-03
Velocity equivalent to 30 mph	13.4	13.4E+03	13.4

GENERAL CARD FORMAT

The following sections specify for each keyword the cards that have to be defined. Each card is defined in its rigid format form and is shown as a number of fields in an 80 character string. **Most cards are 8 fields with a length of 10 and a sample card is shown below.**

Card Format

	1	2	3	4	5	6	7	8
--	---	---	---	---	---	---	---	---

Variable	NSID	PSID	A1	A2	A3	HAZ		
Type	I	I	F	F	F	I		
Default	none	none	1.0	1.0	0	1		
Remarks	1			2		3		

The type is the variable type and is either F, for floating point or I, for an integer. The default gives the value set if zero is specified, the field is left blank or the card is not defined. The remarks refer to comments at the end of the section. The card format is given above the card if it is other than eight fields of 10. Free formats may be used with the data separated by commas. When using comma format, the number of characters used to specify a number must not exceed

GETTING STARTED

the number which would fit into the equivalent rigid format field. An I8 number is limited to a number of 99999999 and larger numbers with more than eight characters are unacceptable. Rigid and free formats can be mixed throughout the deck but not within a card.

***AIRBAG**

Purpose: Define an airbag or control volume.

The keyword ***AIRBAG** provides a way of defining thermodynamic behavior of the gas flow into the airbag as well as a reference configuration for the fully inflated bag. The keyword cards in this section are defined in alphabetical order:

- *AIRBAG_OPTION1_{OPTION2}_{OPTION3}_{OPTION4}**
- *AIRBAG_ADVANCED_ALE**
- *AIRBAG_ALE**
- *AIRBAG_INTERACTION**
- *AIRBAG_REFERENCE_GEOMETRY_OPTION_OPTION**
- *AIRBAG_SHELL_REFERENCE_GEOMETRY**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ABID	Contact interface ID. This must be a unique number.
HEADING	Airbag descriptor. It is suggested that unique descriptions be used.

Card	1	2	3	4	5	6	7	8
Variable	SID	SIDTYP	RBID	VSCA	PSCA	VINI	MWD	SPSF
Type	I	I	I	F	F	F	F	F
Default	none	0	0	1.	1.	0.	0.	0.
Remarks			optional					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID
SIDTYP	Set type: EQ.0: segment, NE.0: part IDs.
RBID	Rigid body part ID for user defined activation subroutine: EQ.- <i>RBID</i> : Sensor subroutine flags initiates the inflator. Load curves are offset by initiation time, EQ.0: the control volume is active from time zero, EQ. <i>RBID</i> : User sensor subroutine flags the start of the inflation. Load curves are offset by initiation time. See Appendix D.
VSCA	Volume scale factor, V_{sca} (default=1.0)
PSCA	Pressure scale factor, P_{sca} (default=1.0)
VINI	Initial filled volume, V_{ini}
MWD	Mass weighted damping factor, D
SPSF	Stagnation pressure scale factor, $0 \leq \gamma \leq 1$

Remarks:

The first card is necessary for all airbag options. The sequence for the following cards which is different for each option is explained on the next pages.

Lumped parameter control volumes are a mechanism for determining volumes of closed surfaces and applying a pressure based on some thermodynamic relationships. The volume is specified by a list of polygons similar to the pressure boundary condition cards or by specifying a material subset which represents shell elements which form the closed boundary. All polygon normals must be oriented to face outwards from the control volume. If holes are detected, they are assumed to be covered by planar surfaces.

V_{sca} and P_{sca} allow for unit system changes from the inflator to the finite element model. There are two sets of volume and pressure used for each control volume. First, the finite element model computes a volume ($V_{femod\ell}$) and applies a pressure ($P_{femod\ell}$). The thermodynamics of a control volume may be computed in a different unit system; thus, there is a separate volume ($V_{cvolume}$) and pressure ($P_{cvolume}$) which are used for integrating the differential equations for the control volume. The conversion is as follows:

$$V_{cvolume} = (V_{sca} V_{femod\ell}) - V$$

$$P_{femod\ell} = P_{sca} P_{cvolume}$$

Damping can be applied to the structure enclosing a control volume by using a mass weighted damping formula:

$$F_i^d = m_i D (v_i - v_{cg})$$

where F_i^d is the damping force, m_i is the nodal mass, v_i is the velocity for a node, v_{cg} is the mass weighted average velocity of the structure enclosing the control volume, and D is the damping factor.

An alternative, separate damping is based on the stagnation pressure concept. The stagnation pressure is roughly the maximum pressure on a flat plate oriented normal to a steady state flow field. The stagnation pressure is defined as $p = \gamma \rho V^2$ where V is the normal velocity of the control volume relative to the ambient velocity, ρ is the ambient air density, and γ is a factor which varies from 0 to 1 and has to be chosen by the user. Small values are recommended to avoid excessive damping.

Sensor Input to Activate Inflator
Define if and only if $RBID$ nonzero.

Skip this input if $RBID=0$. If the rigid body ID is non-zero then define either the input for the user defined sensor subroutine (A) or define the data for the default sensor (B).

The sensor is mounted on a rigid body which is attached to the structure. The motion of the sensor is provided in the local coordinate system defined for the rigid body in the definition of the rigid material, see *MAT RIGID. This is important since the default local system is taken as the principal axes of the inertia tensor. The local system rotates and translates with the rigid material. When the user defined criterion is met for the deployment of the airbag, a flag is set and the deployment begins. All load curves relating to the mass flow rate versus time are then shifted by the initiation time.

A. Sensor Input for User Subroutine (*RBID*>0)

See Appendix D. A user supplied subroutine must be provided.

Define the following card sets which provide the input parameters for the user defined subroutine. Up to 25 parameters may be used with each control volume.

Card	1	2	3	4	5	6	7	8
Variable	N							
Type	I							
Default	none							

Card Format (Define up to 25 constants for the user subroutine. Input only the number of cards necessary, i.e. for nine constants use 2 cards)

Card	1	2	3	4	5	6	7	8
Variable	C1	C2	C3	C4	C5			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE**DESCRIPTION**

N	Number of input parameters (not to exceed 25).
C1,...CN	Up to 25 constants for the user subroutine.

B. LS-DYNA Sensor Input (RBID<0)

Define three cards which provide the input parameters for the built in sensor subroutine.

Acceleration/Velocity/Displacement Activation

Card	1	2	3	4	5	6	7	8
Variable	AX	AY	AZ	AMAG	TDUR			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Card

Variable	DVX	DVY	DVZ	DVMAG				
Type	F	F	F	F				
Default	0.	0.	0.	0.				

Card

Variable	UX	UY	UZ	UMAG				
Type	F	F	F	F				
Default	0.	0.	0.	0.				

VARIABLE	DESCRIPTION
AX	Acceleration level in local x-direction to activate inflator. The absolute value of the x-acceleration is used. EQ.0: inactive.
AY	Acceleration level in local y-direction to activate inflator. The absolute value of the y-acceleration is used. EQ.0: inactive.
AZ	Acceleration level in local z-direction to activate inflator. The absolute value of the z-acceleration is used. EQ.0: inactive.
AMAG	Acceleration magnitude required to activate inflator. EQ.0: inactive.
TDUR	Time duration acceleration must be exceeded before the inflator activates. This is the cumulative time from the beginning of the calculation, i.e., it is not continuous.
DVX	Velocity change in local x-direction to activate the inflator. (The absolute value of the velocity change is used.) EQ.0: inactive.
DVY	Velocity change in local y-direction to activate the inflator. (The absolute value of the velocity change is used.) EQ.0: inactive.
DVZ	Velocity change in local z-direction to activate the inflator. (The absolute value of the velocity change is used.) EQ.0: inactive.
DVMAG	Velocity change magnitude required to activate the inflator. EQ.0: inactive.
UX	Displacement increment in local x-direction to activate the inflator. (The absolute value of the x-displacement is used.) EQ.0: inactive.
UY	Displacement increment in local y-direction to activate the inflator. (The absolute value of the y-displacement is used.) EQ.0: inactive.
UZ	Displacement increment in local z-direction to activate the inflator. (The absolute value of the z-displacement is used.) EQ.0: inactive.
UMAG	Displacement magnitude required to activate the inflator. EQ.0: inactive.

Additional card required for SIMPLE_PRESSURE_VOLUME option

Card	1	2	3	4	5	6	7	8
Variable	CN	BETA	LCID	LCIDDR				
Type	F	F	I	I				
Default	none	none	none	0				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CN	Coefficient. Define if the load curve ID, LCID, is unspecified. LT.0.0: CN is the load curve ID, which defines the coefficient as a function of time.
BETA	Scale factor, β . Define if a load curve ID is not specified.
LCID	Optional load curve ID defining pressure versus relative volume.
LCIDDR	Optional load curve ID defining the coefficient, CN, as a function of time during the dynamic relaxation phase.

Remarks:

The relationship is the following:

$$Pressure = \beta \frac{CN}{RelativeVolume}$$

$$RelativeVolume = \frac{CurrentVolume}{InitialVolume}$$

The pressure is then a function of the ratio of current volume to the initial volume. The constant, CN, is used to establish a relationship known from the literature. The scale factor β is simply used to scale the given values. This simple model can be used when an initial pressure is given and no leakage, no temperature, and no input mass flow is assumed. A typical application is the modeling of air in automobile tires.

The load curve, LCIDDR, can be used to ramp up the pressure during the dynamic relaxation phase in order to avoid oscillations after the desired gas pressure is reached. In the DEFINE_CURVE section this load curve must be flagged for dynamic relaxation. After initialization either the constant or load curve ID, |CN| is used to determine the pressure.

Additional cards required for SIMPLE_AIRBAG_MODEL option

Card 1 1 2 3 4 5 6 7 8

Variable	CV	CP	T	LCID	MU	A	PE	RO
Type	F	F	F	I	F	F	F	F
Default	none							

Card 2

Variable	LOU	TEXT	A	B	MW	GASC		
Type	I	F	F	F	F	F		
Default	0	0.	0.	0.	0.	0.		
Remarks	0	optional	optional	optional	optional	optional		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CV	Heat capacity at constant volume
CP	Heat capacity at constant pressure
T	Temperature of input gas
LCID	Load curve ID specifying input mass flow rate. See *DEFINE_CURVE.
MU	Shape factor for exit hole, μ : LT.0.0: $ \mu $ is the load curve number defining the shape factor as a function of absolute pressure.
A	Exit area, A: GE.0.0: A is the exit area and is constant in time, LT.0.0: $ A $ is the load curve number defining the exit area as a function of absolute pressure.
PE	Ambient pressure, p_e

VARIABLE	DESCRIPTION
RO	Ambient density, ρ
LOU	Optional load curve ID giving mass flow out versus gauge pressure in bag. See *DEFINE_CURVE.
TEXT	Ambient temperature. (Define if and only if CV=0.)
A	First heat capacity coefficient of inflator gas (e.g., Joules/mole/°K). (Define if and only if CV=0.)
B	Second heat capacity coefficient of inflator gas, (e.g., Joules/mole/°K ²). (Define if and only if CV=0.)
MW	Molecular weight of inflator gas (e.g., Kg/mole). (Define if and only if CV=0.)
GASC	Universal gas constant of inflator gas (e.g., 8.314 Joules/mole/°K). (Define if and only if CV=0.)

Remarks:

The gamma law equation of state used to determine the pressure in the airbag:

$$p = (\gamma - 1)\rho e$$

where p is the pressure, ρ is the density, e is the specific internal energy of the gas, and γ is the ratio of the specific heats:

$$\gamma = \frac{c_p}{c_v}$$

From conservation of mass, the time rate of change of mass flowing into the bag is given as:

$$\frac{dM}{dt} = \frac{dM_{in}}{dt} - \frac{dM_{out}}{dt}$$

The inflow mass flow rate is given by the load curve ID, LCID. Leakage, the mass flow rate out of the bag, can be modeled in two alternative ways. One is to give an exit area with the corresponding shape factor, then the load curve ID, LOU, must be set to zero. The other is to define a mass flow out by a load curve, then μ and A have to both be set to zero.

If CV=0. then the constant-pressure specific heat is given by:

$$c_p = \frac{(a + bT)}{MW}$$

and the constant-volume specific heat is then found from:

$$c_v = c_p - \frac{R}{MW}$$

Additional card required for ADIABATIC_GAS_MODEL option

Card	1	2	3	4	5	6	7	8
Variable	PSF	LCID	GAMMA	P0	PE	RO		
Type	F	I	F	F	F	F		
Default	1.0	none	none	none	none	none		

VARIABLE**DESCRIPTION**

PSF	Pressure scale factor
LCID	Optional load curve for preload flag. See *DEFINE_CURVE.
GAMMA	Ratio of specific heats
P0	Initial pressure (gauge)
PE	Ambient pressure
RO	Initial density of gas

Remarks:

The optional load curve ID, LCID, defines a preload flag. During the preload phase the function value of the load curve versus time is zero, and the pressure in the control volume is given as:

$$p = PSF p_0$$

When the **first nonzero** function value is encountered, the preload phase stops and the ideal gas law applies for the rest of the analysis. If LCID is zero, no preload is performed.

The gamma law equation of state for the adiabatic expansion of an ideal gas is used to determine the pressure after preload:

$$p = (\gamma - 1)\rho e$$

where p is the pressure, ρ is the density, e is the specific internal energy of the gas, and γ is the ratio of the specific heats:

$$\gamma = \frac{c_p}{c_v}$$

The pressure above is the absolute pressure, the resultant pressure acting on the control volume is:

$$p_s = PSF(p - p_e)$$

where PSF is the pressure scale factor. Starting from the initial pressure p_0 an initial internal energy is calculated:

$$e_0 = \frac{p_0 + p_e}{\rho(\gamma - 1)}$$

Additional 4 cards are required for all WANG_NEFSKE models

Card 1 1 2 3 4 5 6 7 8

Variable	CV	CP	T	LCT	LCMT	TVOL	LCDT	IABT
Type	F	F	F	I	I	F	I	F
Default	none	none	0.	0	none	0.	0.	not used

Card 2

Variable	C23	LCC23	A23	LCA23	CP23	LCCP23	AP23	LCAP23
Type	F	I	F	I	F	I	F	I
Default	none	0	none	0	none	0	0.0	0

Card 3

Variable	PE	RO	GC	LCEFR	POVER	PPOP	OPT	KNKDN
Type	F	F	F	I	F	F	F	I
Default	none	none	none	0	0.0	0.0	0.0	0

***AIRBAG**

***AIRBAG_WANG_NEFSKE**

If the inflator is modeled, LCMT=0, define, the following card. If not, define but leave blank.

Card 4 1 2 3 4 5 6 7 8

Variable	IOC	IOA	IVOL	IRO	IT	LCBF		
Type	F	F	F	F	F	I		
Default	none	none	none	none	none	none		

Define the following card if and only if CV=0. This option allows temperature dependent heat capacities to be defined. See below.

Card 5 1 2 3 4 5 6 7 8

Variable	TEXT	A	B	MW	GASC			
Type	F	F	F	F	F			
Default	none	none	none	none	none			

Define the following card if and only if the POP option is specified. Use this option to specify additional criteria for initiating exit flow from the airbag.

Card 5 1 2 3 4 5 6 7 8

Variable	TDP	AXP	AYP	AZP	AMAGP	TDURP	TDA	RBIDP
Type	F	F	F	F	F	F	F	I
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	none

VARIABLE

DESCRIPTION

CV Heat capacity at constant volume

CP Heat capacity at constant pressure

<u>VARIABLE</u>	<u>DESCRIPTION</u>
T	Temperature of input gas. For temperature variations a load curve, LCT, may be defined.
LCT	Optional load curve number defining temperature of input gas versus time. This overrides columns T.
LCMT	Load curve specifying input mass flow rate or tank pressure versus time. If the tank volume, TVOL, is nonzero the curve ID is assumed to be tank pressure versus time. If LCMT=0, then the inflator has to be modeled, see Card 4. During the dynamic relaxation phase the airbag is ignored unless the curve is flagged to act during dynamic relaxation.
TVOL	Tank volume which is required only for the tank pressure versus time curve, LCMT.
LCDT	Load curve for time rate of change of temperature (dT/dt) versus time.
IABT	Initial airbag temperature. (Optional, generally not defined.)
C23	Vent orifice coefficient which applies to exit hole. Set to zero if LCC23 is defined below.
LCC23	The absolute value, LCC23 , is a load curve ID. If the ID is positive, the load curve defines the vent orifice coefficient which applies to exit hole as a function of time. If the ID is negative, the vent orifice coefficient is defined as a function of relative pressure, P_{air} / P_{bag} , see [Anagonye and Wang 1999]. A nonzero value for C23 overrides LCC23.
A23	If defined as a positive number, A23 is the vent orifice area which applies to exit hole. If defined as a negative number, the absolute value A23 is a part ID, see [Anagonye and Wang, 1999]. The area of this part becomes the vent orifice area. Set A23 to zero if LCA23 is defined below.
LCA23	Load curve number defining the vent orifice area which applies to exit hole as a function of <u>absolute</u> pressure. A nonzero value for A23 overrides LCA23.
CP23	Orifice coefficient for leakage (fabric porosity). Set to zero if LCCP23 is defined below.
LCCP23	Load curve number defining the orifice coefficient for leakage (fabric porosity) as a function of time. A nonzero value for CP23 overrides LCCP23.
AP23	Area for leakage (fabric porosity)

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCAP23	Load curve number defining the area for leakage (fabric porosity) as a function of (absolute) pressure. A nonzero value for AP23 overrides LCAP23.
PE	Ambient pressure
RO	Ambient density
GC	Gravitational conversion constant (mandatory - no default). If consistent units are being used for all parameters in the airbag definition then unity should be input.
LCEFR	Optional curve for exit flow rate versus (gauge) pressure
POVER	Initial relative overpressure (gauge), P_{over} in control volume
PPOP	Pop Pressure: relative pressure (gauge) for initiating exit flow, P_{pop}
OPT	Fabric venting option, if nonzero CP23, LCCP23, AP23, and LCAP23 are set to zero. EQ. 1: Wang-Nefske formulas for venting through an orifice are used. Blockage is not considered. EQ. 2: Wang-Nefske formulas for venting through an orifice are used. Blockage of venting area due to contact is considered. EQ. 3: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage is not considered. EQ. 4: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage of venting area due to contact is considered. EQ. 5: Leakage formulas based on flow through a porous media are used. Blockage is not considered. EQ. 6: Leakage formulas based on flow through a porous media are used. Blockage of venting area due to contact is considered. EQ. 7: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is <u>not</u> considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card. EQ. 8: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card.
KNKDN	<u>Optional</u> load curve ID defining the knock down pressure scale factor versus time. This option only applies to jetting. The scale factor defined by this load curve scales the pressure applied to airbag segments which

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	do not have a clear line-of-sight to the jet. Typically, at very early times this scale factor will be less than unity and equal to unity at later times. The full pressure is always applied to segments which can see the jets.
IOC	Inflator orifice coefficient
IOA	Inflator orifice area
IVOL	Inflator volume
IRO	Inflator density
IT	Inflator temperature
LCBF	Load curve defining burn fraction versus time
TEXT	Ambient temperature.
A	First heat capacity coefficient of inflator gas (e.g., Joules/mole/°K)
B	Second heat capacity coefficient of inflator gas, (e.g., Joules/mole/°K ²)
MW	Molecular weight of inflator gas (e.g., Kg/mole).
GASC	Universal gas constant of inflator gas (e.g., 8.314 Joules/mole/°K)
TDP	Time delay before initiating exit flow after pop pressure is reached.
AXP	Pop acceleration magnitude in local x-direction. EQ.0.0: Inactive.
AYP	Pop acceleration magnitude in local y-direction. EQ.0.0: Inactive.
AZP	Pop acceleration magnitude in local z-direction. EQ.0.0: Inactive.
AMAGP	Pop acceleration magnitude. EQ.0.0: Inactive.
TDURP	Time duration pop acceleration must be exceeded to initiate exit flow. This is a cumulative time from the beginning of the calculation, i.e., it is not continuous.
TDA	Time delay before initiating exit flow after pop acceleration is exceeded for the prescribed time duration.
RBIDP	Part ID of the rigid body for checking accelerations against pop accelerations.

Remarks:

The gamma law equation of state for the adiabatic expansion of an ideal gas is used to determine the pressure after preload:

$$p = (\gamma - 1) \rho e$$

where p is the pressure, ρ is the density, e is the specific internal energy of the gas, and γ is the ratio of the specific heats:

$$\gamma = \frac{c_p}{c_v}$$

where c_v is the specific heat at constant volume, and c_p is the specific heat at constant pressure. A pressure relation is defined:

$$Q = \frac{P_e}{p}$$

where p_e is the external pressure and p is the internal pressure in the bag. A critical pressure relationship is defined as:

$$Q_{crit} = \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma}{\gamma - 1}}$$

where γ is the ratio of specific heats:

$$\gamma = \frac{c_p}{c_v}$$

If

$$Q \leq Q_{crit} \quad \text{then} \quad Q = Q_{crit}$$

Wang and Nefske define the mass flow through the vents and leakage by

$$\dot{m}_{23} = C_{23} A_{23} \frac{p}{R \sqrt{T_2}} Q^{\frac{\gamma}{\gamma - 1}} \sqrt{2 g_c \left(\frac{\gamma R}{\gamma - 1} \right) \left(1 - Q^{\frac{\gamma - 1}{\gamma}} \right)}$$

and

$$\dot{m}'_{23} = C'_{23} A'_{23} \frac{p}{R \sqrt{T_2}} Q^{\frac{\gamma}{\gamma - 1}} \sqrt{2 g_c \left(\frac{\gamma R}{\gamma - 1} \right) \left(1 - Q^{\frac{\gamma - 1}{\gamma}} \right)}$$

It must be noted that the gravitational conversion constant has to be given in consistent units. As an alternative to computing the mass flow out of the bag by the Wang-Nefske model, a curve for the exit flow rate depending on the internal pressure can be taken. Then, no definitions for C23, LCC23, A23, LCA23, CP23, LCCP23, AP23, and LCAP23 are necessary.

The airbag inflator assumes that the control volume of the inflator is constant and that the amount of propellant reacted can be defined by the user as a tabulated curve of fraction reacted versus time. A pressure relation is defined:

$$Q_{crit} = \frac{p_c}{p_i} = \left(\frac{2}{\gamma+1} \right)^{\frac{\gamma}{\gamma-1}}$$

where p_c is a critical pressure at which sonic flow occurs, p_i , is the inflator pressure. The exhaust pressure is given by

$$\begin{aligned} p_e &= p_a \quad \text{if } p_a \geq p_c \\ p_e &= p_c \quad \text{if } p_a < p_c \end{aligned}$$

where p_a is the pressure in the control volume. The mass flow into the control volume is governed by the equation:

$$\dot{m}_in = C_o A_o \sqrt{2 p_i \rho_i} \sqrt{\frac{g_c \gamma \left(Q^{\frac{2}{\gamma}} - Q^{\frac{\gamma+1}{\gamma}} \right)}{\gamma-1}}$$

where C_o , A_o , and ρ_i are the inflator orifice coefficient, area, and gas density, respectively.

If OPT is defined, then for OPT set to 1 or 2 the mass flow rate out of the bag, \dot{m}_{out} is given by:

$$\dot{m}_{out} = \sqrt{g_c} \cdot \left[\sum_{n=1}^{nairmats} (FLC(t)_n \cdot FAC(p)_n \cdot Area_n) \right] \cdot \sqrt{2 p \rho} \sqrt{\frac{\gamma \left(Q^{\frac{2}{\gamma}} - Q^{\frac{\gamma+1}{\gamma}} \right)}{\gamma-1}}$$

where, ρ is the density of airbag gas, *nairmats* is the number of fabrics used in the airbag, and $Area_n$ is the current unblocked area of fabric number n.

If OPT set to 3 or 4 then:

$$\dot{m}_{out} = \left[\sum_{n=1}^{nairmats} (FLC(t)_n \cdot FAC(p)_n \cdot Area_n) \right] \cdot \sqrt{2(p - p_{ext}) \rho}$$

and for OPT set to 5 or 6:

$$\dot{m}_{out} = \left[\sum_{n=1}^{nairmats} (FLC(t)_n \cdot FAC(p)_n \cdot Area_n) \right] \cdot (p - p_{ext})$$

and for OPT set to 7 or 8 (may be comparable to an equivalent model ALE model):

$$\dot{m}_{out} = \sum_{n=1}^{nairmats} FLC(t)_n \cdot FAC(p)_n \cdot Area_n \cdot \rho_n$$

Note that for different OPT settings, $FAC(p)_n$ has different meanings (all units shown just as demonstrations):

- For OPT of 1, 2, 3 and 4, FAC(P) is unit-less.
- For OPT of 5 and 6, FAC(P) has a unit of (s/m).
- For OPT of 7 or 8, FAC(P) is the gas volume outflow through a unit area per unit time thus has the unit of speed, i.e. $vol \approx m^3 / (m^2 \cdot s) \approx m/s \approx vel(P)$.

Multiple airbags may share the same part ID since the area summation is over the airbag segments whose corresponding part ID's are known. Currently, we assume that no more than ten materials are used per bag for purposes of the output. This constraint can be eliminated if necessary.

The total mass flow out will include the portion due to venting, i.e., constants C23 and A23 or their load curves above.

If CV=0, then the constant-pressure specific heat is given by:

$$c_p = \frac{(a + bT)}{MW}$$

and the constant-volume specific heat is then found from:

$$c_v = c_p - \frac{R}{MW}$$

Further additional 2 cards are required for JETTING models

The following additional cards are defined for the WANG_NEFSKE_JETTING and WANG_NEFSKE_MULTIPLE_JETTING options, two further cards are defined for each option. The jet may be defined by specifying either the coordinates of the jet focal point, jet vector head and secondary jet focal point, or by specifying three nodes located at these positions. The nodal point option is recommended when the location of the airbag changes as a function of time.

Define either card below but not both:

1st additional card of 2 required for WANG_NEFSKE_JETTING option

Card 1 1 2 3 4 5 6 7 8

Variable	XJFP	YJFP	ZJFP	XJVH	YJVH	ZJVH	CA	BETA
Type	F	F	F	F	F	F	F	F
Default	none	1.0						
Remark	1	1	1	1	1	1		

1st additional card of 2 required for WANG_NEFSKE_MULTIPLE_JETTING option

Card 1 1 2 3 4 5 6 7 8

Variable	XJFP	YJFP	ZJFP	XJVH	YJVH	ZJVH	LCJRV	BETA
Type	F	F	F	F	F	F	F	F
Default	none	1.0						
Remark	1	1	1	1	1	1		

2nd additional card of 2 required for WANG_NEFSKE_JETTING and WANG_NEFSKE_MULTIPLE_JETTING option

Card 2	1	2	3	4	5	6	7	8
Variable	XSJFP	YSJFP	ZSJFP	PSID	ANGLE	NODE1	NODE2	NODE3
Type	F	F	F	I	F	I	I	I
Default	none	none	none	none	none	0	0	0
Remark						1	1	1

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XJFP	x-coordinate of jet focal point, i.e., the virtual origin in Figure 1.1. See Remark 1 below.
YJFP	y-coordinate of jet focal point, i.e., the virtual origin in Figure 1.1.
ZJFP	z-coordinate of jet focal point, i.e., the virtual origin in Figure 1.1.
XJVH	x-coordinate of jet vector head to defined code centerline
YJVH	y-coordinate of jet vector head to defined code centerline
ZJVH	z-coordinate of jet vector head to defined code centerline
CA	Cone angle, α , defined in radians. LT.0.0: $ \alpha $ is the load curve ID defining cone angle as a function of <i>time</i>
LCJRV	Load curve ID giving the spatial jet relative velocity distribution, see Figures 1.2 and 1.3. The jet velocity is determined from the inflow mass rate and scaled by the load curve function value corresponding to the value of the angle ψ . Typically, the values on the load curve vary between 0 and unity. See *DEFINE_CURVE.
BETA	Efficiency factor, β , which scales the final value of pressure obtained from Bernoulli's equation. LT.0.0: $ \beta $ is the load curve ID defining the efficiency factor as a function of <i>time</i>

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XSJFP	x-coordinate of secondary jet focal point, passenger side bag. If the coordinates of the secondary point are (0,0,0) then a conical jet (driver's side airbag) is assumed.
YSJFP	y-coordinate of secondary jet focal point
ZSJFP	z-coordinate of secondary jet focal point
PSID	Optional part set ID, see *SET_PART. If zero all elements are included in the airbag.
ANGLE	Cutoff angle in degrees. The relative jet velocity is set to zero for angles greater than the cutoff. See Figure 1.3. This option applies to the MULTIPLE jet only.
NODE1	Node ID located at the jet focal point, i.e., the virtual origin in Figure 1.1. See Remark 1 below.
NODE2	Node ID for node along the axis of the jet.
NODE3	Optional node ID located at secondary jet focal point.

Remarks:

1. It is assumed that the jet direction is defined by the coordinate method (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH) unless both NODE1 and NODE2 are defined. In which case the coordinates of the nodes give by NODE1, NODE2 and NODE3 will override (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH). The use of nodes is recommended if the airbag system is undergoing rigid body motion. The nodes should be attached to the vehicle to allow for the coordinates of the jet to be continuously updated with the motion of the vehicle.

The jetting option provides a simple model to simulate the real pressure distribution in the airbag during the breakout and early unfolding phase. Only the surfaces that are in the line of sight to the virtual origin have an increased pressure applied. With the optional load curve LCRJV, the pressure distribution with the code can be scaled according to the so-called relative jet velocity distribution.

For passenger side airbags the cone is replaced by a wedge type shape. The first and secondary jet focal points define the corners of the wedge and the angle α then defines the wedge angle.

Instead of applying pressure to all surfaces in the line of sight of the virtual origin(s), a part set can be defined to which the pressure is applied.

2. Care must be used to place the jet focal point within the bag. If the focal point is outside the bag, inside surfaces will not be visible so jetting pressure will not be applied correctly.

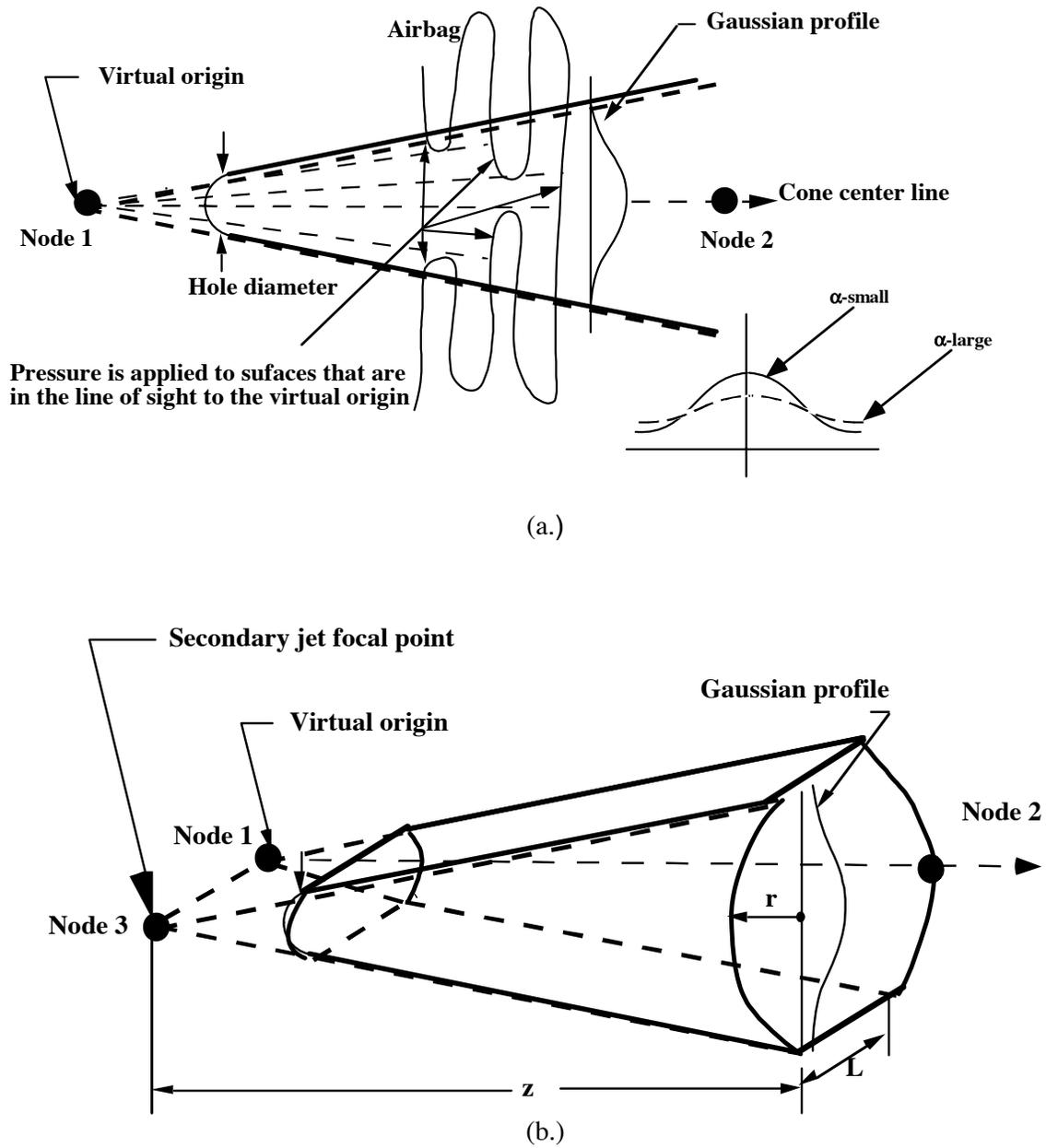


Figure 1.1 Jetting configuration for (a.) driver's side airbag (pressure applied only if centroid of surface is in line-of-sight) and (b.) the passenger's side bag.

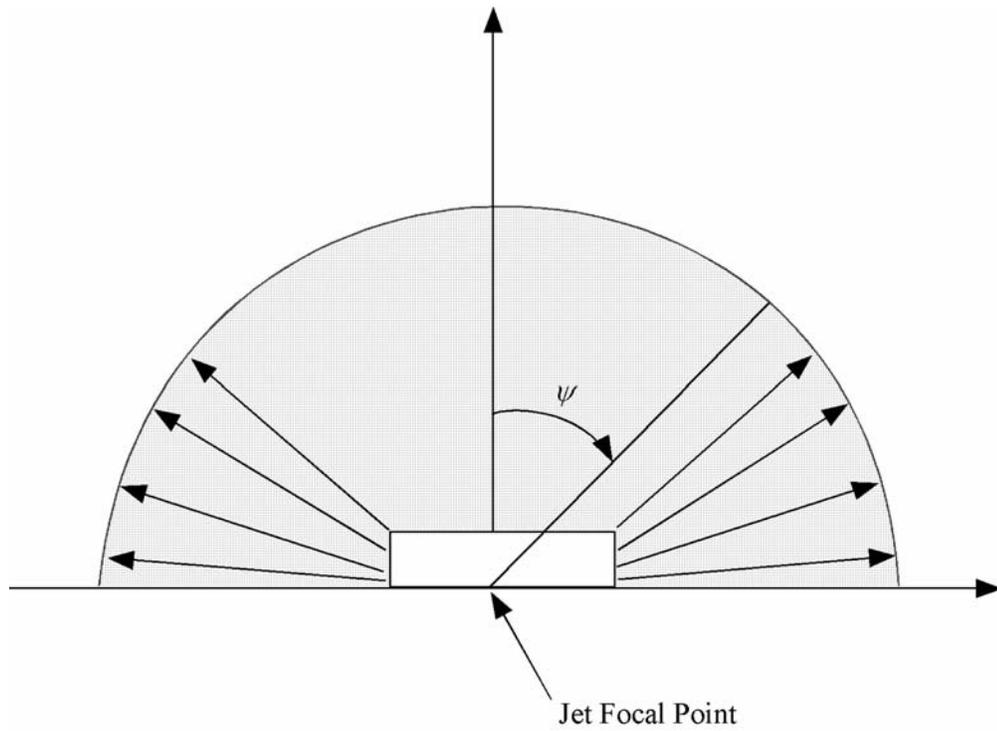


Figure 1.2 Multiple jet model for driver's side airbag.

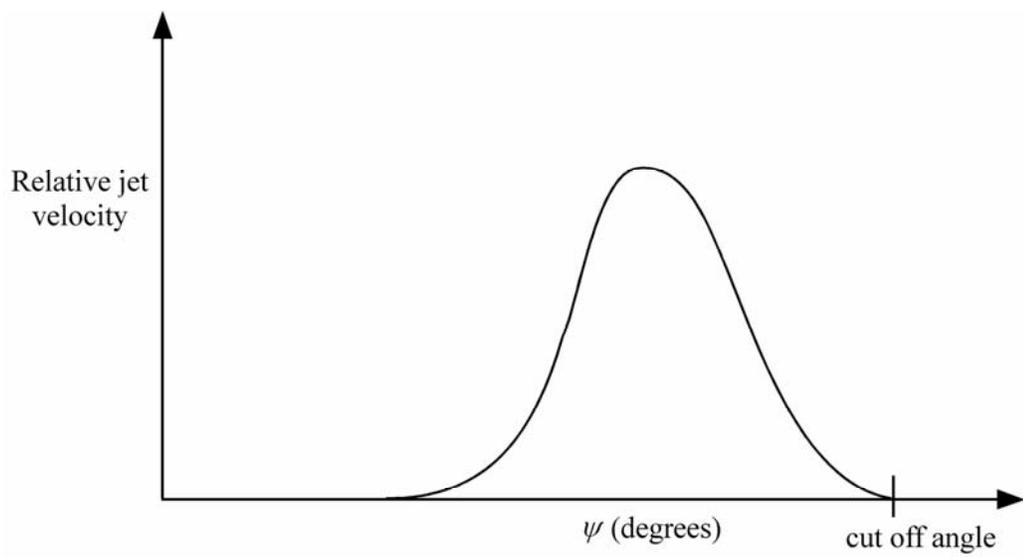


Figure 1.3 Normalized jet velocity versus angle for multiple jet driver's side airbag.

Further additional required for CM option.

The following additional card is defined for the WANG_NEFSKE_JETTING_CM and WANG_NEFSKE_MULTIPLE_JETTING_CM options.

Additional card required for _CM option

Card 1 1 2 3 4 5 6 7 8

Variable	NREACT								
Type	I								
Default	None								
Remark									

VARIABLE

DESCRIPTION

NREACT

Node for reacting jet force. If zero the jet force will not be applied.

Remarks:

Compared with the standard LS-DYNA jetting formulation, the Constant Momentum option has several differences. Overall, the jetting usually has a more significant effect on airbag deployment than the standard LS-DYNA jetting: the total force is often greater, and does not reduce with distance from the jet.

The velocity at the jet outlet is assumed to be a choked (sonic) adiabatic flow of a perfect gas. Therefore the velocity at the outlet is given by:

$$v_{outlet} = \sqrt{(\gamma RT)} = \sqrt{\left(\frac{(c_p - c_v) T c_p}{c_v} \right)}$$

The density in the nozzle is then calculated from conservation of mass flow.

$$\rho_0 v_{outlet} A_{outlet} = \dot{m}$$

This is different from the standard LS-DYNA jetting formulation, which assumes that the density of the gas in the jet is the same as atmospheric air, and then calculates the jet velocity from conservation of mass flow.

The velocity distribution at any radius, r , from the jet centerline and distance, z , from the focus, $v_{r,z}$, relates to the velocity of the jet centerline, $v_r = 0, z$, in the same way as the standard LS-DYNA jetting options:

$$v_{r,z} = v_{r=0,z} e^{-\left(\frac{r}{\alpha z}\right)^2}$$

The velocity at the jet centerline, $v_r = 0$, at the distance, z , from the focus of the jet is calculated such that the momentum in the jet is conserved.

Momentum at nozzle = Momentum at z

$$\begin{aligned} \rho_0 v_{outlet}^2 A_{outlet} &= \rho_0 \int v_{jet}^2 dA_{jet} \\ &= \rho_0 v_{r=0,z}^2 \{b + F\sqrt{b}\} \end{aligned}$$

where

$$b = \frac{\pi(\alpha z)^2}{2}$$

$F = \text{distance between jet focii for a passenger jet}$

Finally, the pressure exerted on an airbag element in view of the jet is given as:

$$p_{r,z} = \beta \rho_0 v_{r,z}^2$$

By combining the equations above

$$p_{r,z} = \frac{\beta \dot{m} v_{outlet} \left[e^{-(r/\alpha z)^2} \right]^2}{\left\{ \frac{\pi(\alpha z)^2}{2} + F \sqrt{\frac{\pi(\alpha z)^2}{2}} \right\}}$$

The total force exerted by the jet is given by:

$$F_{jet} = \dot{m} v_{outlet} \quad (\text{independent of distance from the nozzle})$$

Mass flow in the jet is not necessarily conserved, because gas is entrained into the jet from the surrounding volume. By contrast, the standard LS-DYNA jetting formulation conserves mass flow but not momentum. This has the effect of making the jet force reduce with distance from the nozzle.

The jetting forces can be reacted onto a node (NREACT), to allow the reaction force through the steering column or the support brackets to be modeled. The jetting force is written to the ASCII ABSTAT file and the binary XTF file.

Additional card required for LOAD_CURVE option

Card	1	2	3	4	5	6	7	8
Variable	STIME	LCID	RO	PE	P0	T	T0	
Type	F	I	F	F	F	F	F	
Default	0.0	none	none	none	none	none	none	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
STIME	Time at which pressure is applied. The load curve is offset by this amount.
LCID	Load curve ID defining pressure versus time, see *DEFINE_CURVE.
RO	Initial density of gas (ignored if LCID > 0)
PE	Ambient pressure (ignored if LCID > 0)
P0	Initial gauge pressure (ignored if LCID > 0)
T	Gas Temperature (ignored if LCID > 0)
T0	Absolute zero on temperature scale (ignored if LCID > 0)

Remarks:

Within this simple model the control volume is inflated with a pressure defined as a function of time or calculated using the following equation if LCID = 0.

$$P_{total} = C\rho(T - T_0)$$

$$P_{gauge} = P_{total} - P_{ambient}$$

The pressure is uniform throughout the control volume.

Additional card required for LINEAR_FLUID option

Card	1	2	3	4	5	6	7	8
Variable	BULK	RO	LCINT	LCOUTT	LCOUTP	LCFIT	LCBULK	LCID
Type	F	F	I	I	I	I	I	I
Default	none	none	none	optional	optional	optional	optional	none

<u>VARIABLE</u>	<u>DESCRIPTION</u>
BULK	K, bulk modulus of the fluid in the control volume. Constant as a function of time. Define if LCBULK=0.
RO	ρ , density of the fluid
LCINT	$F(t)$ input flow curve defining mass per unit time as a function of time, see *DEFINE_CURVE.
LCOUTT	$G(t)$, output flow curve defining mass per unit time as a function of time. This load curve is optional.
LCOUTP	$H(p)$, output flow curve defining mass per unit time as a function of pressure. This load curve is optional.
LCFIT	$L(t)$, added pressure as a function of time. This load curve is optional.
LCBULK	Curve defining the bulk modulus as a function of time. This load curve is optional, but if defined, the constant, BULK, is not used.
LCID	Load curve ID defining pressure versus time, see *DEFINE_CURVE.

Remarks:

If LCID = 0 then the pressure is determined from:

$$P(t) = K(t) \ln \left(\frac{V_0(t)}{V(t)} \right) + L(t)$$

where

$P(t)$ Pressure,

$V(t)$ Volume of fluid in compressed state,

$$V_0(t) = V_0(t) = \frac{M(t)}{\rho} \quad \text{Volume of fluid in uncompressed state,}$$

$$M(t) = M(0) + \int F(t)dt - \int G(t)dt - \int H(p)dt \quad \text{Current fluid mass,}$$

$$M(0) = V(0)\rho \quad \text{Mass of fluid at time zero } P(0) = 0.$$

By setting LCID $\neq 0$ a pressure time history may be specified for the control volume and the mass of fluid within the volume is then calculated from the volume and density.

This model is for the simulation of hydroforming processes or similar problems. The pressure is controlled by the mass flowing into the volume and by the current volume. The pressure is uniformly applied to the control volume.

Note the signs used in the equation for $M(t)$. The mass flow should always be defined as positive since the output flow is subtracted.

Additional cards required for HYBRID and HYBRID_JETTING options

Card 1 2 3 4 5 6 7 8

Variable	ATMOST	ATMOSP	ATMOSD	GC	CC			
Type	F	F	F	F	F			
Default	none	none	none	none	1.0			

Card

Variable	C23	LCC23	A23	LCA23	CP23	LCP23	AP23	LCAP23
Type	F	I	F	I	F	I	F	I
Default	none	0	none	0	none	0	none	0

Card

Variable	OPT	PVENT	NGAS					
Type	I	F	I					
Default	none	none	none					

***AIRBAG**

***AIRBAG_HYBRID**

Define 2*NGAS cards below, two for each gas type.

Card	1	2	3	4	5	6	7	8
Variable	LCIDM	LCIDT		MW	INITM	A	B	C
Type	I	I	F	F	F	F	F	F
Default	none	none	not used	none	none	none	none	none

Card

Variable	FMASS							
Type	F							
Default	none							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ATMOST	Atmospheric temperature
ATMOSP	Atmospheric pressure
ATMOSD	Atmospheric density
GC	Universal molar gas constant
CC	Conversion constant EQ: .0 Set to 1.0.
C23	Vent orifice coefficient which applies to exit hole. Set to zero if LCC23 is defined below.
LCC23	The absolute value, LCC23 , is a load curve ID. If the ID is positive, the load curve defines the vent orifice coefficient which applies to exit hole as a function of time. If the ID is negative, the vent orifice coefficient is defined as a function of relative pressure, P_{air} / P_{bag} , see [Anagonye and Wang 1999]. A nonzero value for C23 overrides LCC23.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
A23	If defined as a positive number, A23 is the vent orifice area which applies to exit hole. If defined as a negative number, the absolute value A23 is a part ID, see [Anagonye and Wang 1999]. The area of this part becomes the vent orifice area. Set A23 to zero if LCA23 is defined below.
LCA23	Load curve number defining the vent orifice area which applies to exit hole as a function of <u>absolute</u> pressure. A nonzero value for A23 overrides LCA23.
CP23	Orifice coefficient for leakage (fabric porosity). Set to zero if LCCP23 is defined below.
LCCP23	Load curve number defining the orifice coefficient for leakage (fabric porosity) as a function of time. A nonzero value for CP23 overrides LCCP23.
AP23	Area for leakage (fabric porosity)
LCAP23	Load curve number defining the area for leakage (fabric porosity) as a function of (absolute) pressure. A nonzero value for AP23 overrides LCAP23.
OPT	Fabric venting option, if nonzero CP23, LCCP23, AP23, and LCAP23 are set to zero. EQ. 1: Wang-Nefske formulas for venting through an orifice are used. Blockage is not considered. EQ. 2: Wang-Nefske formulas for venting through an orifice are used. Blockage of venting area due to contact is considered. EQ. 3: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage is not considered. EQ. 4: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage of venting area due to contact is considered. EQ. 5: Leakage formulas based on flow through a porous media are used. Blockage due to contact is not considered. EQ. 6: Leakage formulas based on flow through a porous media are used. Blockage due to contact is considered. EQ. 7: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is not considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card. EQ. 8: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card.

VARIABLE	DESCRIPTION
PVENT	Gauge pressure when venting begins
NGAS	Number of gas inputs to be defined below. (Including initial air)
LCIDM	Load curve ID for inflator mass flow rate (eq. 0 for gas in the bag at time 0) GT.0: piece wise linear interpolation LT.0: cubic spline interpolation
LCIDT	Load curve ID for inflator gas temperature (eq.0 for gas in the bag at time 0) GT.0: piece wise linear interpolation LT.0: cubic spline interpolation
BLANK	(not used)
MW	Molecular weight
INITM	Initial mass fraction of gas component
A	Coefficient for molar heat capacity of inflator gas at constant pressure, (e.g., Joules/mole/°K)
B	Coefficient for molar heat capacity of inflator gas at constant pressure, (e.g., Joules/mole/°K ²)
C	Coefficient for molar heat capacity of inflator gas at constant pressure, (e.g., Joules/mole/°K ³)
FMASS	Fraction of additional aspirated mass.

Further additional cards are required for HYBRID_JETTING and ..._CM models

The following two additional cards are defined for the HYBRID_JETTING options. The jet may be defined by specifying either the coordinates of the jet focal point, jet vector head and secondary jet focal point, or by specifying three nodes located at these positions. The nodal point option is recommended when the location of the airbag changes as a function of time.

Card 1 1 2 3 4 5 6 7 8

Variable	XJFP	YJFP	ZJFP	XJVH	YJVH	ZJVH	CA	BETA
Type	F	F	F	F	F	F	F	F
Default	none							
Remark	1	1	1	1	1	1		

Card 2

Variable	XSJFP	YSJFP	ZSJFP	PSID	IDUM	NODE1	NODE2	NODE3
Type	F	F	F	I	F	I	I	I
Default	none	none	none	none	none	0	0	0
Remark					2	1	1	1

Additional card required for HYBRID_JETTING_CM option

Card 1 1 2 3 4 5 6 7 8

Variable	NREACT							
Type	I							
Default	None							
Remark	4							

VARIABLE	DESCRIPTION
XJFP	x-coordinate of jet focal point, i.e., the virtual origin in Figure 1.1. See Remark 1 below.
YJFP	y-coordinate of jet focal point, i.e., the virtual origin in Figure 1.1.
ZJFP	z-coordinate of jet focal point, i.e., the virtual origin in Figure 1.1.
XJVH	x-coordinate of jet vector head to defined code centerline
YJVH	y-coordinate of jet vector head to defined code centerline
ZJVH	z-coordinate of jet vector head to defined code centerline
CA	Cone angle, α , defined in radians. LT.0.0: $ \alpha $ is the load curve ID defining cone angle as a function of <i>time</i>
BETA	Efficiency factor, β , which scales the final value of pressure obtained from Bernoulli's equation. LT.0.0: $ \beta $ is the load curve ID defining the efficiency factor as a function of <i>time</i>
XSJFP	x-coordinate of secondary jet focal point, passenger side bag. If the coordinate of the secondary point is (0,0,0) then a conical jet (driver's side airbag) is assumed.
YSJFP	y-coordinate of secondary jet focal point
ZSJFP	z-coordinate of secondary jet focal point
PSID	Optional part set ID, see *SET_PART. If zero all elements are included in the airbag.
IDUM	Dummy field (Variable not used)
NODE1	Node ID located at the jet focal point, i.e., the virtual origin in Figure 1.1. See Remark 1 below.
NODE2	Node ID for node along the axis of the jet.
NODE3	Optional node ID located at secondary jet focal point.
NREACT	Node for reacting jet force. If zero the jet force will not be applied.

Remarks:

1. It is assumed that the jet direction is defined by the coordinate method (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH) unless both NODE1 and NODE2 are defined. In which case the coordinates of the nodes given by NODE1, NODE2 and NODE3 will override (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH). The use of nodes is recommended if the airbag system is undergoing rigid body motion. The nodes should be attached to the vehicle to allow for the coordinates of the jet to be continuously updated with the motion of the vehicle.

The jetting option provides a simple model to simulate the real pressure distribution in the airbag during the breakout and early unfolding phase. Only the surfaces that are in the line of sight to the virtual origin have an increased pressure applied. With the optional load curve LCRJV, the pressure distribution with the code can be scaled according to the so-called relative jet velocity distribution.

For passenger side airbags the cone is replaced by a wedge type shape. The first and secondary jet focal points define the corners of the wedge and the angle α then defines the wedge angle.

Instead of applying pressure to all surfaces in the line of sight of the virtual origin(s), a part set can be defined to which the pressure is applied.

2. This variable is not used and has been included to maintain the same format as the WANG_NEFSKE_JETTING options.
3. Care must be used to place the jet focal point within the bag. If the focal point is outside the bag, inside surfaces will not be visible so jetting pressure will not be applied correctly.
4. See the description related to the WANG_NEFSKE_JETTING_CM option. For the hybrid inflator model the heat capacities are compute from the combination of gases which inflate the bag.

Additional cards required for HYBRID_CHEMKIN model

The HYBRID_CHEMKIN model includes 3 control cards. For each gas species an additional set of cards must follow consisting of a control card and several thermodynamic property data cards.

Card 1 1 2 3 4 5 6 7 8

Variable	LCIDM	LCIDT	NGAS	DATA	ATMT	ATMP	RG	
Type	I	I	I	I	F	F	F	
Default	none	none	none	none	none	none	none	

Card 2

Variable	HCONV							
Type	F							
Default	0.							

Card 3

Variable	C23	A23						
Type	F	F						
Default	0.	0.						

VARIABLE

DESCRIPTION

LCIDM Load curve specifying input mass flow rate versus time.
 GT.0: piece wise linear interpolation
 LT.0: cubic spline interpolation

LCIDT Load curve specifying input gas temperature versus time.
 GT.0: piece wise linear interpolation
 LT.0: cubic spline interpolation

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NGAS	Number of gas inputs to be defined below. (Including initial air)
DATA	Thermodynamic database EQ.1. NIST database (3 additional property cards are required below) EQ.2. CHEMKIN database (no additional property cards are required) EQ.3. Polynomial data (1 additional property card is required below)
ATMT	Atmospheric temperature.
ATMP	Atmospheric pressure
RG	Universal gas constant
HCONV	Convection heat transfer coefficient
C23	Vent orifice coefficient
A23	Vent orifice area

For each gas species include a set of cards consisting of a control card followed by several thermo-dynamic property data cards. The next "*" card terminates the reading of this data.

Control Card

Card 1 1 2 3 4 5 6 7 8

Variable	CHNAME	MW	LCIDN	FMOLE	FMOLET			
Type	A	F	I	F	F			
Default	none	none	0	none	0.			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CHNAME	Chemical symbol for this gas species (e.g., N2 for nitrogen, AR for argon). Required for DATA=2 (CHEMKIN), optional for DATA=1 or DATA=3.
MW	Molecular weight of this gas species.
LCIDN	Load curve specifying the input mole fraction versus time for this gas species. If >0, FMOLE is not used.

*AIRBAG

*AIRBAG_HYBRID_CHEMKN

<u>VARIABLE</u>	<u>DESCRIPTION</u>
FMOLE	Mole fraction of this gas species in the inlet stream.
FMOLET	Initial mole fraction of this gas species in the tank.

Additional thermodynamic data cards for each gas species. No additional cards are needed if using the CHEMKN database (DATA=2).

If DATA=1, include the following 3 cards for the NIST database. The required data can be found on the NIST web site at <http://webbook.nist.gov/chemistry/>

Card 1 1 2 3 4 5 6 7 8

Variable	TLOW	TMID	THIGH					
Type	F	A8	F					
Default	none	none	none					

Card 2

Variable	a low	b low	c low	d low	e low	f low	h low	
Type	F	F	F	F	F	F	F	
Default	none							

Card 3

Variable	a high	b high	c high	d high	e high	f high	h high	
Type	F	F	F	F	F	F	F	
Default	none							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TLOW	Curve fit low temperature limit.
TMID	Curve fit low-to-high transition temperature.
THIGH	Curve fit high temperature limit.
a_{low}, \dots, h_{low}	Low temperature range NIST polynomial curve fit coefficients (see below).
$a_{high}, \dots, h_{high}$	High temperature range NIST polynomial curve fit coefficients (see below).

If DATA=3, include the following card for the polynomial curve fit.

Card 1	1	2	3	4	5	6	7	8
Variable	a	b	c	d	e			
Type	F	F	F	F	F			
Default	none	0.	0.	0.	0.			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
a	Coefficient, see below.
b	Coefficient, see below.
c	Coefficient, see below.
d	Coefficient, see below.
e	Coefficient, see below.

Heat capacity curve fits:

NIST
$$c_p = \frac{1}{M} \left(a + bT + cT^2 + dT^3 + \frac{e}{T^2} \right)$$

CHEMKIN
$$c_p = \frac{\bar{R}}{M} (a + bT + cT^2 + dT^3 + eT^4)$$

\bar{R} = universal gas constant (8.314 Nm / mole K)
 M = gas molecular weight

Polynomial
$$c_p = \frac{1}{M} (a + bT + cT^2 + dT^3 + eT^4)$$

***AIRBAG_ADVANCED_ALE**

Purpose: This *AIRBAG_ADVANCED_ALE (AAA) card provides an alternate, simplified approach to simulating airbag deployment with ALE capabilities. Its input is translated into ALE keywords internally inside LS-DYNA (please review an output file called “advalebak.kw” which contains all the translated ALE commands). It has an option to start the simulation with the ALE method then switch to the control volume (CV) method at a chosen switch time. The CV input data may be used directly. A basic Lagrangian airbag shell structure may consist of an inflator compartment, an airbag, and possibly vent holes. The Lagrangian airbag interacts with the ALE inflator gas. The AAA card requires, at the minimum, one or more *DEFINE_ALEBAG_INFLATOR cards (one for each inflator), and one or more *DEFINE_ALEBAG_BAG cards (one for each bag structure). In addition, one optional card, *DEFINE_ALEBAG_HOLE, may be used to define each physical vent hole on the airbag. Multiple fluid-to-structure interactions may be simulated using multiple AAA definitions. However, more complex couplings most likely must be modeled via the traditional ALE approach (remark 1)..

Card 1 Format

Card 1 1 2 3 4 5 6 7 8

Variable	BAGID1	BAGID2	BAGID3	BAGID4	BAGID5	BAGID6	BAGID7	BAGID8
Type	I	I	I	I	I	I	I	I
Default	none							
Remarks								

Card 2

Variable	HOLEID1	HOLEID2	HOLEID3	HOLEID4	HOLEID5	HOLEID6	HOLEID7	HOLEID8
Type	I	I	I	I	I	I	I	I
Default	none							
Remarks								

***AIRBAG**

***AIRBAG_ADVANCED_ALE**

Card 3 1 2 3 4 5 6 7 8

Variable	INFLID1	INFLID2	INFLID3	INFLID4	INFLID5	INFLID6	INFLID7	INFLID8
Type	I	I	I	I	I	I	I	I
Default	none							
Remarks	8							

Card 4 Format (Define information related to the automatic ALE mesh generation)

Card 4 1 2 3 4 5 6 7 8

Variable	NX/IDA	NY	NZ	unused	ARSNID	IDCENT	EXSID	
Type	I	I	I		I	I	I	
Default	none	none	none		none	none	none	
Remarks	4	4	4					

Card 4b (optional, only define if NX & NY & NZ are nonzero, for automatic ALE mesh generation)

Card 4 1 2 3 4 5 6 7 8

Variable	LX	LY	LZ	ITRANS	UIDAIR			
Type	F	F	F	I	I			
Default	none	none	none					
Remarks								

Card 5: Parameters for defining ambient environment CV *AIRBAG_PARAMETERS

Card 5 1 2 3 4 5 6 7 8

Variable	ATMOST	ATMOSP	unused	GC	CC	unused	MWD	SPSF
Type	F	F		F	F		F	F
Default	0.	0.		none	1.0		0.0	0.0
Remarks	5	5						

Card 6: Parameters for defining ambient environment CV *AIRBAG_PARAMETERS

Card 6 1 2 3 4 5 6 7 8

Variable	SWTIME	unused	HG	NAIR				
Type	F		F	I				
Default	0.		0.	0				
Remarks	6							

Card 7+ (Define air: repeat this card “NAIR” times. NAIR is defined in Card 6)

Card 1 2 3 4 5 6 7 8

Variable	unused	unused	unused	MWAIR	INITM	AIRA	AIRB	AIRC
Type				F	F	F	F	F
Default				0	0	0	0.	0.
Remarks						2		

VARIABLE	DESCRIPTION
BAGID#	Each BAGID refers to an airbag definition defined via a corresponding *DEFINE_ALEBAG_BAG (DAB) card. Each DAB card defines (a) one Lagrangian shell structure representing an airbag (or part of one), (b) its venting characteristics, and (c) its coupling behavior..
HOLEID#	Each HOLEID refers to a vent hole definition defined via a corresponding *DEFINE_ALEBAG_HOLE (DAH) card. Each DAH defines the information required for modeling a physical venting hole including the switching of the ALE multi-material group (AMMG) ID when a gas passes through this vent hole. The information in DAH is translated into a simplified form of the *ALE_FSI_SWITCH_MMG_ID (AFSM) card.
INFLID#	Each INFLID refers to one inflator definition defined via a corresponding *DEFINE_ALEBAG_INFLATOR (DAI) card. Each DAI defines the information required for the modeling of a physical inflator (which pumps a gas mixture in to inflate the airbag). INFLID# > 0: INFLID is associated with an inflator definition defined via a corresponding DAI card
NX/IDA	Option 1: NX is defined as the number of ALE elements to be generated by LS-DYNA automatically in the x direction. If option 1 is chosen, option 1 must also be used for NY and NZ, and card 4B must also be defined. Option 2: IDA is the Part ID of the initial background air mesh (remarks 4 and 9). If IDA is a negative integer, the PID of the air mesh is the same as the one used in a previous AAA card. This approach is only used in the cases of (a) bag-in-bag or (b) bag-to-bag model where there may be more than one AAA card defined.
NY	Option 1: NY is defined as the number of ALE elements to be generated by LS-DYNA automatically in the y direction. If option 1 is chosen, option 1 must also be used for NX and NZ, and card 4B must also be defined. Option 2: Leave blank or 0 (remark 4).
NZ	Option 1: NZ is defined as the number of ALE elements to be generated by LS-DYNA automatically in the z direction. If option 1 is chosen, option 1 must also be used for NX and NY, and card 4B must also be defined Option 2: Leave blank or 0 (remark 4).

VARIABLE	DESCRIPTION
ARSNID	An ID associated with an *ALE_REFERENCE_SYSTEM_NODE card defining 3 nodes making up a local coordinate system. The ALE mesh system will automatically follow this local coordinate system. If there is mesh expansion, and if IDCENT > 0, the expansion will occur along these local axes. Else if IDCENT is not defined (0), the expansion will occur along the global coordinate axes.
IDCENT	A Lagrangian node ID defines the center of expansion for the ALE mesh system. For example, this node may be located on a rigid section of a steering wheel, near the inflator region. The ALE mesh can then move with the steering wheel as it expands. This provides a similar expansion mechanism the *ALE_REFERENCE_SYSTEM_GROUP command (PRTYPE=9).
EXSID	A part ID which makes up a portion of the overall the ALE mesh system to be excluded from expansion. This part can still move with the rest of the ALE system, but will not expand with the rest. This may be used to exclude the inner ALE mesh region surrounding the inflator (which is typically finer resolution to better resolve the inflator gas inflow at the orifices).
LX	Absolute length in the global X-coordinate of the ALE mesh
LY	Absolute length in the global Y-coordinate of the ALE mesh
LZ	Absolute length in the global Z-coordinate of the ALE mesh
ITRANS	An integer ID pointing an ID of a *DEFINE_TRANSFORMATION card which may be used to position the automatically generated ALE mesh at its chosen location and orientation. The mesh is first generated with respect to the origin of the global coordinate system. Then it is moved to the t=0 position via this *DEFINE_TRANSFORMATION card.
UIDAIR	User-defined PID number for the background ALE air mesh. If this is left blank, LS-DYNA will assign a PID for this automatically generated ALE mesh. This may be used to actively define the PID number for the ALE mesh
ATMOST	ATMOST Atmospheric ambient temperature (See Remark 5).
ATMOSP	ATMOSP Atmospheric ambient pressure (See Remark 5).
GC	Universal molar gas constant.
CC	Conversion constant. EQ: 0.0 Set to 1.0.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
MWD	Mass weighted damping factor, D
SPSF	Stagnation pressure scale factor.
SWTIME	Time to switch from ALE method to control volume (CV) method. Once switched, the airbag inflation is simulated by a control volume method similar to that used by the *AIRBAG_HYBRID card (See Remark 6). EQ.0.0: switch to CV method at time = 0.0. EQ.blank: switch time is set at 1.0E16 (ALE method is used). EQ.t: switch from ALE to CV method at time = t.
HG	Hourglass coefficient for ALE fluid mesh(es). A typical value on the order of 1.0E-5 or 1.0E-6 may be used for gases and liquids.
NAIR	Number of species for air. If air is defined as 1 single gas then NAIR=1. Card 7 is repeated "NAIR" number of times, one for each species. For example, NAIR=2 for a 2-component air model containing 80% of N2 and 20% of O2. Then card 7 is defined twice.
MWAIR	Molecular weight of this air component
INITMFA	Initial Mass Fraction of this air component
AIRA	First Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K, See Remark 2).
AIRB	Second Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K ²).
AIRC	Third Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K ³).

Remarks:

1. This card defines information for simulating an airbag deployment. It defines:
 - The IDs of the bags, vent holes, and inflators;
 - Information about the ALE background (air) mesh;
 - Basic ambient condition (similar to that from *AIRBAG_HYBRID);
 - Switch time (time for switching from ALE to CV method);
 - Material properties of air and some other miscellaneous information.

As the input to this command, *AIRBAG_ADVANCED_ALE, is simply translated into the traditional ALE keywords, this command will not have all the flexibilities that the general ALE method allows. It does not provide a grammatical framework for understanding the modeling of fluid structure interaction. It is recommended that the users familiarize themselves with the traditional method of modeling airbag deployment using the ALE method. This is critical in understanding the interactions between the ALE inflator gas and the Lagrangian airbag structure.

2. The per-mass-unit, temperature-dependent, constant-pressure heat capacity is

$$C_p(T) = \frac{[A + B * T + C * T^2]}{MW} \sim \frac{J}{kg * K}$$

$$A = \tilde{C}_{p0} \sim J / (mole * K) \qquad B \sim J / (mole * K^2)$$

$$\qquad \qquad \qquad C \sim J / (mole * K^3)$$

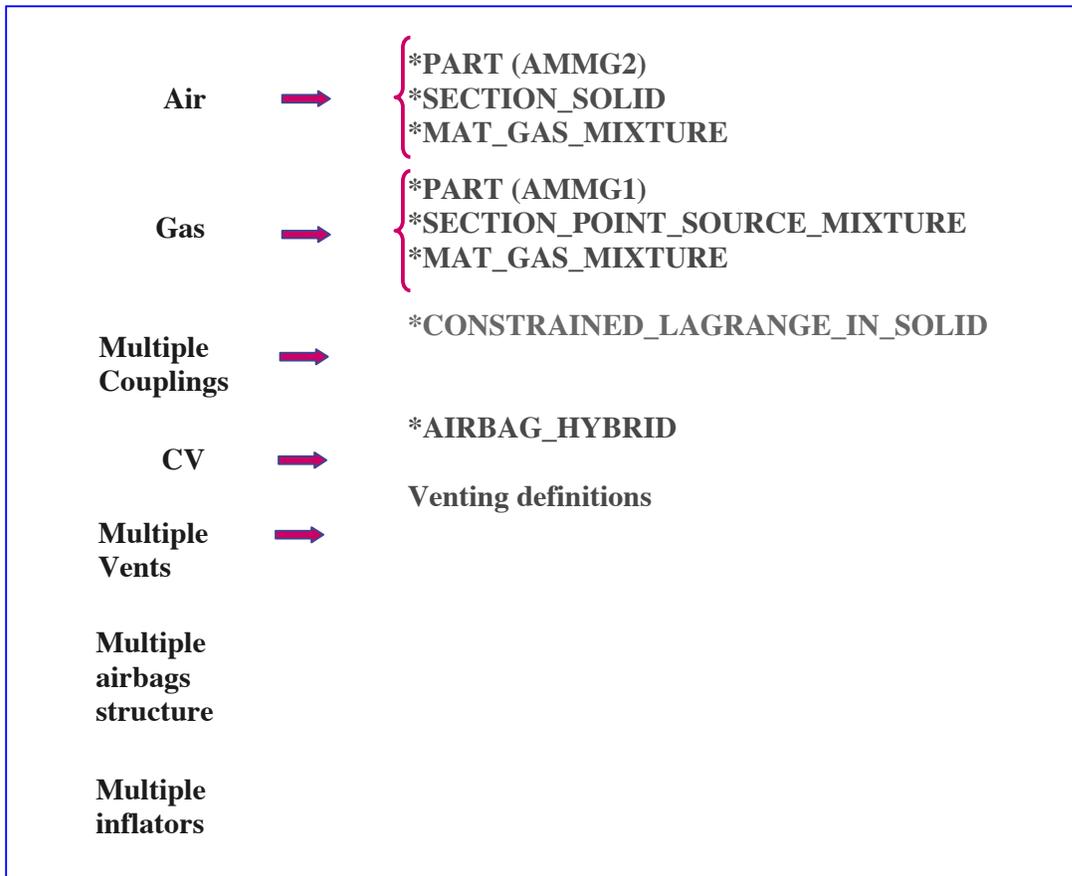
The units shown are only for demonstration of the equation.

3. In general, it is best to locate a point source near the center of an ALE element. Associated with each point source is an area and a vector indicating flow direction. Each point source should occupy 1 ALE element by itself, and there should be at least 2 empty ALE elements between any 2 point sources. A point source should be located at least 3 elements away from the free surface of an ALE mesh for stability (see *SECTION_POINT_SOURCE_MIXTURE card)
4. There are 2 options for defining the ALE mesh for an airbag deployment simulation. Option 1 lets LS-DYNA generate the background ALE mesh. NX, NY and NZ must be defined. Card 4b must also be defined. Option 2 uses an existing ALE mesh for the background air. The PID of this ALE mesh is defined via IDA (same as the NX field). In this case, the background ALE air mesh should envelope the airbag deployment space. There is no need to define card 4b. If IDA is a negative integer, the PID for the air mesh is the same as the one used in another AAA card (multiple AAA cards implied).
5. Atmospheric density for the ambient gas (air) can be computed from $\rho_{amb} = P_{amb} / (R * T_{amb})$. This card is similar to the 1st additional card for the *AIRBAG_HYBRID command with 2 additional parameters (MWD & SPSF) from the 1st card of the *AIRBAG_ command.
6. Since ALL ALE related activities will be turned off after the switch from ALE method to control-volume method, no other ALE coupling will exist beyond t=SWTIME. Using the traditional ALE modeling method, this switching can also be modeled via the command

*ALE_UP_SWITCH. The user should review its usage as it can provide extensively more coupling controls for general airbag modeling.

- 7. The *AIRBAG_ADVANCED_ALE (AAA) card may be considered an alternate approach for inputting the information required by the traditional ALE card:

***AIRBAG_ADVANCED_ALE**



***AIRBAG_ALE**

Purpose: The input in this section provides a simplified approach to defining the deployment of the airbag using the ALE capabilities with an option to switch from the initial ALE method to control volume (CV) method (*AIRBAG_HYBRID) at a chosen time. An enclosed airbag (and possibly the airbag canister/compartments and/or a simple representation of the inflator) shell structure interacts with the inflator gas(es). This definition provides a single fluid to structure coupling for the airbag-gas interaction during deployment in which the CV input data may be used directly.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	SIDTYP					MWD	SPSF
Type	I	I					F	F
Default	none	none					0	0
Remarks	1							

VARIABLE

DESCRIPTION

SID Set ID as defined on *AIRBAG card. This set ID contains the Lagrangian elements (segments) which make up the airbag and possibly the airbag canister/compartments and/or a simple representation of the inflator. See Remark 1.

SIDTYP Set type:
 EQ.0: Segment set.
 EQ.1: Part set.

MWD Mass weighted damping factor, D. This is used during the CV phase for *AIRBAG_HYBRID.

SPSF Stagnation pressure scale factor, $0 \leq \gamma \leq 1$. This is used during the CV phase for *AIRBAG_HYBRID.

Parameters for defining ambient environment.

Card 2 1 2 3 4 5 6 7 8

Variable	ATMOST	ATMOSP		GC	CC	TNKVOL	TNKFINP	
Type	F	F		F	F	F	F	
Default	0.	0.		none	1.0	0.0	0.0	
Remarks	2	2				10	10	

VARIABLE

DESCRIPTION

- ATMOST Atmospheric ambient temperature. See Remark 2.
- ATMOSP Atmospheric ambient pressure. See Remark 2.
- GC Universal molar gas constant.
- CC Conversion constant. If EQ: .0 Set to 1.0.
- TNKVOL Tank volume from the inflator tank test – or – Inflator canister volume. See remark 10.
 Option 1: (LCVEL = 0) This is defined as Tank volume (must also define TNKFINP). Inlet gas velocity is estimated by LS-DYNA method (testing).
 Option 2: (LCVEL = 0) This is defined as estimated inflator canister volume (must NOT define TNKFINP). Inlet gas velocity is estimated automatically by the Lian-Bhalsod-Olovsson method.
 Option 3: (LCVEL .NE. 0) This must be left blank.
- TNKFINP Tank final pressure from the inflator tank test data. Only define this parameter for option 1 of TNKVOL definition above. See Remark 10.

Parameters for coupling, see keyword *CONSTRAINED_LAGRANGE_IN_SOLID.

Card 3 1 2 3 4 5 6 7 8

Variable	NQUAD	CTYPE	PFAC	FRIC	FRCMIN	NORMTYP	ILEAK	PLEAK
Type	I	I	F	F	F	I	I	F
Default	4	4	0.1	0.0	0.3	0	2	0.1
Remarks	13	13	14					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NQUAD	Number of (quadrature) coupling points for coupling Lagrangian slave parts to ALE master solid parts. If NQUAD=n, then nXn coupling points will be parametrically distributed over the surface of each Lagrangian slave segment (default=4). See Remark 13.
CTYPE	Coupling type (default=4, see Remark 13): EQ.4: (default) penalty coupling with DIREC=2 implied. EQ.6: penalty coupling in which DIREC is automatically set to DIREC=1 for the unfolded region and DIREC=2 for folded region.
PFAC	Penalty factor. PFAC is a scale factor for scaling the estimated stiffness of the interacting (coupling) system. It is used to compute the coupling forces to be distributed on the slave and master parts. If positive real: Fraction of estimated critical stiffness (default=0.1). If negative integer, -n: Refers to load curve ID n. The curve defines the relative coupling pressure (y-axis) as a function of the tolerable fluid penetration distance (x-axis).
FRIC	Coupling coefficient of friction.
FRCMIN	Minimum fluid volume fraction in an ALE element to activate coupling (default is 0.3).
NORMTYP	Penalty coupling spring direction (DIREC 1 and 2): EQ.0: normal vectors are interpolated from nodal normals (default) EQ.1: normal vectors are interpolated from segment normals.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
I LEAK	Leakage control flag. Default=2 (with energy compensation).
P LEAK	Leakage control penalty factor (default=0.1)

Parameters for airbag venting hole

Card 4 1 2 3 4 5 6 7 8

Variable	IVSETID	IVTYPE	IBLOCK	VNTCOF				
Type	I	I	I	F				
Default	0	0	0	0.0				
Remarks	4		5	6				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IVSETID	Set ID defining the venting hole surface(s). See Remark 4.
IVTYPE	Set type of IVSETID: EQ.0: Part Set (default). EQ.1: Part ID. EQ.2: Segment Set.
IBLOCK	Flag for considering blockage effects for porosity and vents (see Remark 5): EQ.0: no (blockage is NOT considered, default). EQ.1: yes (blockage is considered).
VNTCOF	Vent Coefficient for scaling the flow. See Remark 6.

Parameters for ALE mesh automatic definition and its transformation.

Card 5 1 2 3 4 5 6 7 8

Variable	NX/IDA	NY/IDG	NZ	MOVERN	ZOOM			
Type	I	I	I	I	I			
Default	None	None	None	0	0			
Remarks	7	7	7	8	9			

VARIABLE**DESCRIPTION**

NX/IDAIR	Option 1: NX is defined (as the number of ALE elements to be generated in the x direction). This must go together with option 1 for NY and NZ. Option 2: IDAIR is defined as Part ID of the initial air mesh. See remark 7.
NY/IDGAS	Option 1: NY is defined (as the number of ALE elements to be generated in the y direction). This must go together with option 1 for NX and NZ. Option 2: IDGAS is defined as Part ID of the initial gas mesh. See remark 7.
NZ	Option 1: NZ is defined (as the number of ALE elements to be generated in the z direction). This must go together with option 1 for NX and NY. Option 2: Leave blank. See remark 7.
MOVERN	ALE mesh automatic motion option (see Remark 8): EQ.0: ALE mesh is fixed in space. GT.0: Node group id. See *ALE_REFERENCE_SYSTEM_NODE ALE mesh can be moved with PRTYP=5, mesh motion follows a coordinate system defined by 3 reference nodes.
ZOOM	ALE mesh automatic expansion option (see Remark 9): EQ.0: do not expand ALE mesh EQ.1: Expand/contract ALE mesh by keeping all airbag parts contained within the ALE mesh (equivalent to PRTYP=9).

Define card 5a and 5b if NZ > 0

Card 5a 1 2 3 4 5 6 7 8

Variable	X0	Y0	Z0	X1	Y1	Z1	IPAIR	IPGAS
Type	F	F	F	F	F	F	I	I
Default	None	None						

Card 5b

Variable	X2	Y2	Z2	Z3	Y3	Z3		
Type	F	F	F	F	F	F		
Default	None	None	None	None	None	None		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
X0, Y0, Z0	Coordinates of origin for ALE mesh generation (node0).
X1, Y1, Z1	Coordinates of point 1 for ALE mesh generation (node1). node0=>node1 = x
X2, Y2, Z2	Coordinates of point 2 for ALE mesh generation (node2). node0=>node2 = y
X3, Y3, Z3	Coordinates of point 3 for ALE mesh generation(node3). node0=>node3 = z

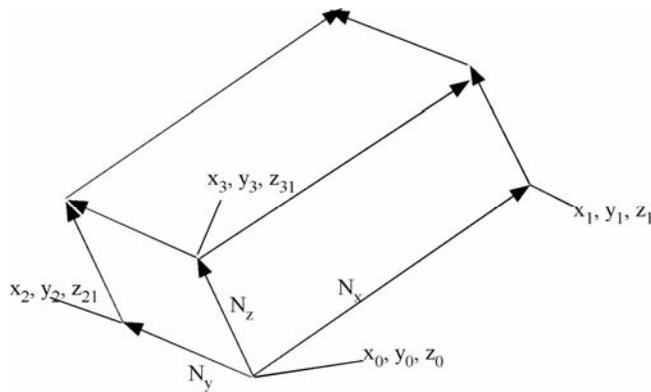


Figure 1.4.

Card 6	1	2	3	4	5	6	7	8
Variable	SWTIME		HG	NAIR	NGAS	NORIF	LCVEL	LCT
Type	F		F	I	I	I	I	I
Default	0.		0.	0	0	0	0	0
Remarks	3						10	11

VARIABLE**DESCRIPTION**

SWTIME	Time to switch from ALE method to control volume (CV) method. Once switched, a method similar to that used by the *AIRBAG_HYBRID card is used. EQ.0.0: switch to CV method at time=0.0. EQ.blank: switch time is set at 1.0E16 (ALE method is used). EQ.t: switch from ALE to CV method at time=t.
HG	Hourglass control for ALE fluid mesh(es).
NAIR	Number of Air components. For example, this equals 2 in case air contains 80% of N2 and 20% of O2. If air is defined as 1 single gas then NAIR=1.
NGAS	Number of inflator Gas components.
NORIF	Number of point sources or orifices (defined below) EQ.n: will require n lines of card 9 definitions below, one for each point source.
LCVEL	Load curve ID for inlet velocity (see also TNKVOL & TNKFINP of card 2 above). This is the same estimated velocity curve used in *SECTION_POINT_SOURCE_MIXTURE card.
LCT	Load curve ID for inlet gas temperature (see *AIRBAG_HYBRID).

*AIRBAG

*AIRBAG_ALE

Define NAIR cards below for air component

Card 7 1 2 3 4 5 6 7 8

Variable				MWAIR	INITM	AIRA	AIRB	AIRC
Type				F	F	F	F	F
Default				0	0	0	0.	0.
Remarks						12	12	12

VARIABLE

DESCRIPTION

MWAIR	Molecular weight of air component
INITA	Initial Mass Fraction of Air component(s)
AIRA	First Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K, remark 12).
AIRB	Second Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K ² , remark 12).
AIRC	Third Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K ³ , remark 12).

Define NGAS cards below for the GAS components

Cards 8... 1 2 3 4 5 6 7 8

Variable	LCMF			MWGAS		GASA	GASB	GASC
Type	I			F		F	F	F
Default	none			0		0	0.	0.
Remarks	11					12	12	12

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCMF	Load curve ID for mass flow rate (see *AIRBAG_HYBRID, e.g., kg/s).
MWGAS	Molecular weight of inflator gas components.
GASA	First Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K, remark 12).
GASB	Second Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K ² , remark 12).
GASC	Third Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K ³ , remark 12).

Define NORIF cards below for each point source

Cards 9... 1 2 3 4 5 6 7 8

Variable	NODEID	VECID	ORIFARE					
Type	I	I	I					
Default	0	0	0					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NODEID	The node ID defining the point source.
VECID	The vector ID defining the direction of flow at the point source.
ORIFARE	The orifice area at the point source.

Remarks:

1. This set ID typically contains the Lagrangian segments of the 3 parts that are coupled to the inflator gas: airbag, airbag canister (compartment), inflator. As in all control-volume, orientation of elements representing bag and canister should point outward. During the ALE phase the segment normal will be reversed automatically for fluid-structure coupling. *However, the orientation of inflator element normal vectors should point to its center.* See Figure 1.5.
2. Atmospheric density for the ambient gas (air) can be computed from $\rho_{amb} = P_{amb} / (R * T_{amb})$
3. Since ALL ALE related activities will be turned off after the switch from ALE method to control-volume method, no other ALE coupling will exist beyond t=SWTIME.
4. Vent definition will be used for ALE venting. Upon switching area of the segments will be used for venting as a23 in *AIRBAG_HYBRID.
5. Fabric porosity for ALE and *AIRBAG_HYBRID can be defined on MAT_FABRIC. Define FLC and FAC on *MAT_FABRIC. FVOPT 7 and 8 will be used for both ALE and *AIRBAG_HYBRID. IBLOCK=0 will use FVOPT=7 and IBLOCK=1 will use FVOPT=8.
6. VCOF will be used to scale the vent area for ALE venting and this coefficient will be used as vent coefficient c23 for *AIRBAG_HYBRID upon switching.

- 7. If Nz, Ny and Nz are defined (option 1), card 5a and card 5b should be defined to let LS-DYNA generate the mesh for ALE. Alternatively if Nz is 0 (option 2), then Nx=IDAIR and Ny=IDGAS. In the later case the user need to supply the ALE mesh whose PID=IDAIR.
- 8. If the airbag moves with the vehicle, set MOVERN=GROUPID, this GROUPID is defined using *ALE_REFERENCE_SYSTEM_NODE. The 3 nodes defined in ALE_REFERENCE_SYSTEM_NODE will be used to transform the ALE mesh. The point sources will also follow this motion. This simulates PRTYP=5 in the *ALE_REFERENCE_SYSTEM_GROUP card.
- 9. Automatic expansion/contraction of the ALE mesh to follow the airbag expansion can be turned on by setting zoom=1. This feature is particularly useful for fully folded airbags requiring very fine ale mesh initially. As the airbag inflates the ale mesh will be automatically scaled such that the airbag will be contained within the ALE mesh. This simulates PRTYP=9 in the *ALE_REFERENCE_SYSTEM_GROUP card.
- 10. There are 3 methods for defining the inlet gas velocity:
Option 1: define LCVEL = 0 → TNKVOL = Tank volume, and TNKFINP = Tank final pressure from tank test data. Inlet gas velocity is estimated by LSDYNA method (testing).
Option 2: define LCVEL = 0 → TNKVOL = inflator can volume, and TNKFINP = blank. Inlet gas velocity is estimated automatically by Lian-Bhalsod-Olovsson method.
Option 3: define LCVEL = n → TNKVOL =0, and TNKFINP = 0. Inlet gas velocity is defined by user via a load curve ID = n.
- 11. LCT and LCIDM should have the same number of sampling points.
- 12. The per-mass-unit, temperature-dependent, constant-pressure heat capacity is

$$C_p(T) = \frac{[A + B * T + C * T^2]}{MW} \sim \frac{J}{kg * K} \quad \begin{matrix} B \sim J/(mole * K^2) \\ C \sim J/(mole * K^3) \end{matrix}$$

$$A = \tilde{C}_{p0} \sim J/(mole * K)$$

The units shown are only for demonstration of the equation.

- 13. Sometimes CTYPE=6 may be used for complex folded airbag. NQUAD=2 may be used as a starting value and increase as necessary depending on the relative mesh resolutions of the Lagrangian and ALE meshes.
- 14. Use a load curve for PFAC whenever possible. It tends to be more robust.

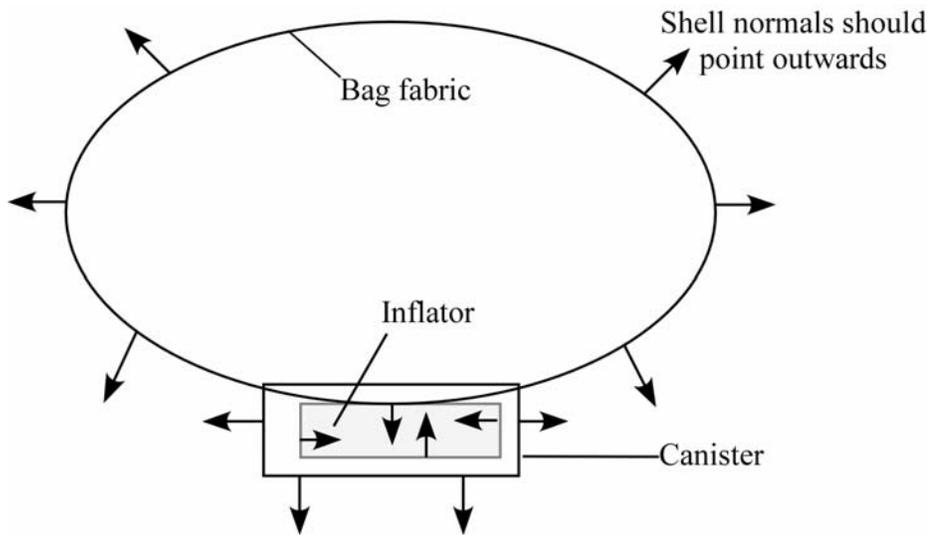
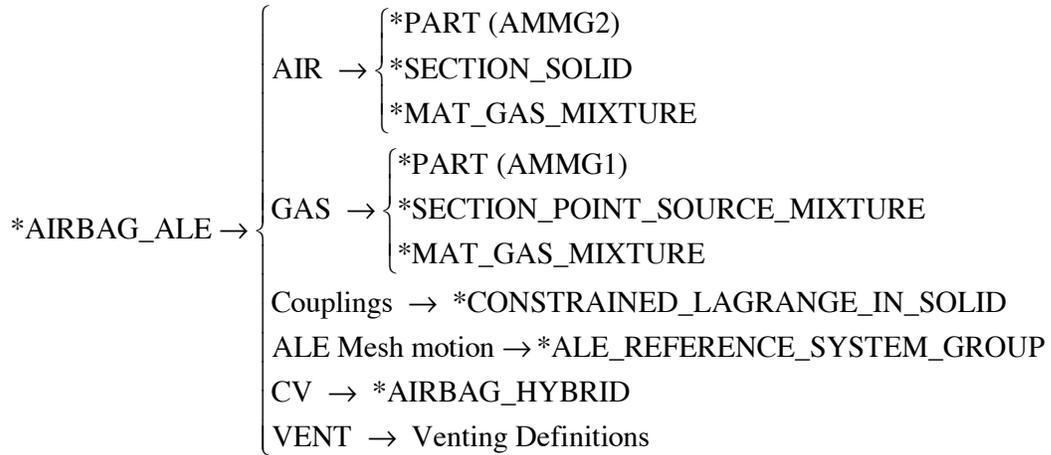


Figure 1.5.

Example 1:

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*AIRBAG_ALE
$#1  SID      SIDTYPE      NONE      NONE      NONE      NONE      MWD      SPSF
      123      1          0          0          0          0          0.0      0.0
$#2  ATMOST  ATMOSP      NONE      GC         CC         TNKVOL    TNKFP
      298.15  1.0132E-4  0          8.314     0.0        0.0        0.0
$#3  NQUAD   CTYPE      PFAC      FRIC      FRCMIN    NORMTYPE  ILEAK     PLEAK
      4        4          -1000     0.0       0.3        0          2          0.1
$#4  VSETID  IVSETTYP  IBLOCK    VENTCOEF
      1        2          0          1.00
$#5  NXIDAIR NYIDGAS    NZ         MOVERN     ZOOM
      50000    50003     0          0          0
$#6  SWTIME  NONE      HG         NAIR       NGAS      NORIF     LCVEL     LCT
      1000.00  0.000    1.e-4      1          1          8         2002     2001
$#7  AIR     NONE      NONE      MWAIR     INITM     AIRA     AIRB     AIRC
      0        0          0          0.02897   1.00     29.100   0.00000  0.00000
$#8  GASLCM  NONE      NONE      MWGAS     NONE      GASA     GASB     GASC
      2003     0          0          0.0235    0         28.000   0.00000  0.00000
$#9  NODEID  VECTID    ORIFAREA
      100019   1          13.500000
      100020   2          13.500000
      100021   3          13.500000
      100022   4          13.500000
      100023   5          13.500000
      100024   6          13.500000
      100017   7          13.500000
      100018   8          13.500000
$ PFAC CURVE = penalty factor curve.
*DEFINE_CURVE
$  lcid      sidr      sfa      sfo      offa      offo      dattyp
$  1000      0         0.0      2.0      0.0       0.0
$           a1         o1
           0.0         0.00000000
           1.00000000  4.013000e-04
*SET_SEGMENT_TITLE
vent segments (defined in IVSETID)
      1        0.0      0.0      0.0      0.0
      1735     1736     661     1697     0.0      0.0      0.0      0.0
      1735     2337     1993     1736     0.0      0.0      0.0      0.0
      1735     1969     1988     2337     0.0      0.0      0.0      0.0
      1735     1697     656     1969     0.0      0.0      0.0      0.0
*DEFINE_VECTOR
$#  vid      xt      yt      zt      xh      yh      zh
      1        0.0     0.0-16.250000  21.213200  21.213200-16.250000
      2        0.0     0.0-16.250000  30.000000-1.000e-06-16.250000
      3        0.0     0.0-16.250000  21.213200-21.213200-16.250000
      4        0.0     0.0-16.250000-1.000e-06-30.000000-16.250000
      5        0.0     0.0-16.250000-21.213200-21.213200-16.250000
      6        0.0     0.0-16.250000-30.0000001.000e-06-16.250000
      7        0.0     0.0-16.250000-21.213200  21.213200-16.250000
      8        0.0     0.0-16.2500001.000e-06  30.000000-16.250000
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

In this example, pre-existing background air mesh with part ID 50000 and gas mesh with part ID 50003 are used. Thus NZ = 0. There is no mesh motion nor expansion allowed. An inlet gas velocity curve is provided.

Example 2:

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ SIDTYP: 0=SGSID; 1=PSID
*AIRBAG_ALE
$#1  SID  SIDTYPE  NONE  NONE  NONE  NONE  MWD  SPSF
      1      1      0      0.      0.      0.      0.      0.
$#2  ATMOST  ATMOSP  NONE  GC      CC      TNKVOL  TNKFP
      298.  101325.  0.0  8.314  1.      6.0E-5  0
$#3  NQUAD  CTYPE  PFAC  FRIC  FRCMIN  NORMTYPE  ILEAK  PLEAK
      2      6      -321  0.0  0.3      1      2      0.1
$#4  VSETID  IVSETTYP  IBLOCK  VENTCOEF
      0      0      0      0
$#5NXIDAIR  NYIDGAS  NZ  MOVERN  ZOOM
      11      11      9
$5b  x0      y0      z0      x1      y1      z1  NOT-USED  NOT-USED
      -0.3  -0.3  -0.135  0.3  -0.3  -0.135
$5c  x2      y2      z2      x3      y3      z3  NOT-USED  NOT-USED
      -0.3  0.3  -0.135  -0.3  -0.3  0.39
$#6  SWTIME  NONE  HG  NAIR  NGAS  NORIF  LCVEL  LCT
      0.04000  0.005  1.e-4  2  1  1  0  2
$#7  AIR      NONE  NONE  MWAIR  INITM  AIRA  AIRB  AIRC
      0.028  0.80  27.296  0.00523
      0.032  0.20  25.723  0.01298
$#8  GASLCM  NONE  NONE  MWGAS  NONE  GASA  GASB  GASC
      1      0.0249  29.680  0.00880
$#9  NODEID  VECTID  ORIFAREA
      9272  1  1.00e-4
$ Lagrangian shell structure to be coupled to the inflator gas
*SET_PART_LIST
      1      0.0      0.0      0.0      0.0
      1      2      3
*DEFINE VECTOR
$0.100000E+01, 10.000000000
$ vid  xt  yt  zt  xh  yh  zh
      1  0.0  0.0  0.0  0.0  0.0  0.100000
$ bag penetration ~ 1 mm <====> P_coup ~ 500000 pascal ==> ~ 5 atm
*DEFINE_CURVE
$ lcid  sidr  sfa  sfo  offa  offo  dattyp
      321  0  0.0  0.0  0.0  0.0
$ al  ol
      0.0  0.0
      0.00100000  5.0000000e+05
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

In this example, LS-DYNA automatically creates the background ALE mesh with:
 NX = 11 elements in the x direction.
 NY = 11 elements in the y direction.
 NZ = 9 elements in the z direction.

***AIRBAG_INTERACTION**

Purpose: To define two connected airbags which vent into each other.

Define one card for each airbag interaction definition

Card 1 2 3 4 5 6 7 8

Variable	AB1	AB2	AREA	SF	PID	LCID	IFLOW	
Type	I	I	F	F	I	I	I	
Default	none	none	none	none	0	0	0	

VARIABLE	DESCRIPTION
AB1	First airbag ID, as defined on *AIRBAG card.
AB2	Second airbag ID, as defined on *AIRBAG card.
AREA	Orifice area between connected bags. LT.0.0: AREA is the load curve ID defining the orifice area as a function of absolute pressure. EQ.0.0: AREA is taken as the surface area of the part ID defined below.
SF	Shape factor. LT.0.0: SFI is the load curve ID defining vent orifice coefficient as a function of relative time.
PID	Optional part ID of the partition between the interacting control volumes. AREA is based on this part ID.
LCID	Load curve ID defining mass flow rate versus pressure difference, see *DEFINE_CURVE. If LCID is defined AREA, SF and PID are ignored.
IFLOW	Flow direction LT.0: One way flow from AB1 to AB2 only. EQ.0: Two way flow between AB1 and AB2. GT.0: One way flow from AB2 to AB1 only.

Remarks:

Mass flow rate and temperature load curves for the secondary chambers must be defined as null curves, for example, in the DEFINE_CURVE definitions give two points (0.0,0.0) and (10000.,0.0).

All input options are valid for the following airbag types:

- *AIRBAG_SIMPLE_AIRBAG_MODEL
- *AIRBAG_WANG_NEFSKE
- *AIRBAG_WANG_NEFSKE_JETTING
- *AIRBAG_WANG_NEFSKE_MULTIPLE_JETTING
- *AIRBAG_HYBRID
- *AIRBAG_HYBRID_JETTING

The LCID defining mass flow rate vs. pressure difference may additionally be used with:

- *AIRBAG_LOAD_CURVE
- *AIRBAG_LINEAR_FLUID

If the AREA, SF, and PID defined method is used to define the interaction then the airbags must contain the same gas, i.e. C_p , C_v and g must be the same. The flow between bags is governed by formulas which are similar to those of Wang-Nefske, except that choked flow is currently ignored. This will be added later.

***AIRBAG_REFERENCE_GEOMETRY_{OPTION}_{OPTION}**

Available options include:

<BLANK>

BIRTH

RDT

The reference geometry becomes active at time BIRTH. Until this time the input geometry is used to inflate the airbag. Until the birth time is reached the actual geometry is used to determine the time step size even if RDT is active.

If RDT is active the time step size will be based on the reference geometry once the solution time exceeds the birth time. This option is useful for shrunken bags where the bag does not carry compressive loads and the elements can freely expand before stresses develop. If this option is not specified, the time step size will be based on the current configuration and will increase as the area of the elements increase. The default may be much more expensive but possibly more stable.

Purpose: If the reference configuration of the airbag is taken as the folded configuration, the geometrical accuracy of the deployed bag will be affected by both the stretching and the compression of elements during the folding process. Such element distortions are very difficult to avoid in a folded bag. By reading in a reference configuration such as the final unstretched configuration of a deployed bag, any distortions in the initial geometry of the folded bag will have no effect on the final geometry of the inflated bag. This is because the stresses depend only on the deformation gradient matrix:

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

where the choice of X_j may coincide with the folded or unfold configurations. It is this unfolded configuration which may be specified here.

Note that a reference geometry which is smaller than the initial airbag geometry will not induce initial tensile stresses.

If a liner is included and the parameter LNRC set to 1 in *MAT_FABRIC, compression is disabled in the liner until the reference geometry is reached, i.e., the fabric element becomes tensile.

*AIRBAG

*AIRBAG_REFERENCE_GEOMETRY

Define the follow card if and only if the option BIRTH is specified in the keyword.

Card	1	2	3	4	5	6	7	8
Variable	BIRTH							
Type	F							
Default	0.0							

Card Format (I8,3E16.0) The next “*” keyword card terminates this input.

Card 2,...	1	2	3	4	5	6	7	8	9	10
Variable	NID	X	Y	Z						
Type	I	F	F	F						
Default	none	0.	0.	0.						
Remarks										

VARIABLE

DESCRIPTION

BIRTH	Time at which the reference geometry activates (default=0.0)
NID	Node ID for which a reference configuration is defined. Nodes defined in this section must also appear under the *NODE input. It is only necessary to define the reference coordinates of nodal points, if their coordinates are different than those defined in the *NODE section.
X	x coordinate
Y	y coordinate
Z	z coordinate

***AIRBAG_SHELL_REFERENCE_GEOMETRY**

Purpose: Usually, the input in this section is not needed; however, sometimes it is convenient to use disjoint pre-cut airbag parts to define the reference geometries. If the reference geometry is based only on nodal input, this is not possible since in the assembled airbag the boundary nodes are merged between parts. By including the shell connectivity with the reference geometry, the reference geometry can be based on the pre-cut airbag parts instead of the assembled airbag. The elements, which are defined in this section, must have identical element ID's as those defined in the *ELEMENT_SHELL input, but the nodal ID's, which may be unique, are only used for the reference geometry. These nodes are defined in the *NODE section, but can also be additionally defined above under *AIRBAG_REFERENCE_GEOMETRY. The element orientation and n1-n4 ordering must be identical to the *ELEMENT_SHELL input.

Card Format (6I8)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N1	N2	N3	N4				
Type	I	I	I	I	I	I				
Default	none	none	none	none	none	none				
Remarks			3	3	3	3				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Element ID
PID	Optional part ID, see *PART, the part ID is not used in this section.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
N4	Nodal point 4

***AIRBAG**

***AIRBAG_SHELL_REFERENCE_GEOMETRY**

***ALE**

The keyword ***ALE** provides a way of defining input data pertaining to the Arbitrary-Lagrangian-Eulerian capability. The keyword cards in this section are defined in alphabetical order:

- *ALE_FSI_PROJECTION**
- *ALE_FSI_SWITCH_MMG_{OPTION}**
- *ALE_MULTI-MATERIAL_GROUP**
- *ALE_REFERENCE_SYSTEM_CURVE**
- *ALE_REFERENCE_SYSTEM_GROUP**
- *ALE_REFERENCE_SYSTEM_NODE**
- *ALE_REFERENCE_SYSTEM_SWITCH**
- *ALE_SMOOTHING**
- *ALE_TANK_TEST**
- *ALE_UP_SWITCH**

For other input information related to the ALE capability, see keywords:

- *ALE_TANK_TEST**
- *BOUNDARY_AMBIENT_EOS**
- *CONSTRAINED_EULER_IN_EULER**
- *CONSTRAINED_LAGRANGE_IN_SOLID**
- *CONTROL_ALE**
- *DATABASE_FSI**
- *INITIAL_VOID**
- *INITIAL_VOLUME_FRACTION**
- *INITIAL_VOLUME_FRACTION_GEOMETRY**
- *SECTION_SOLID**
- *SECTION_POINT_SOURCE** (for gas only)
- *SECTION_POINT_SOURCE_MIXTURE**
- *SET_MULTI-MATERIAL_GROUP_LIST**
- *CONSTRAINED_EULER_IN_EULER**

SINGLE GASEOUS MATERIAL	MULTIPLE GASEOUS MATERIAL
*EOS_LINEAR_POLYNOMIAL *EOS_IDEAL_GAS *MAT_NULL	*MAT_GAS_MIXTURE *INITIAL_GAS_MIXTURE

*ALE

*ALE_FSI_PROJECTION

*ALE_FSI_PROJECTION

Purpose: This card provides a coupling method for simulating the interaction between a Lagrangian material set (structure) and ALE material set (fluid). The nearest ALE nodes are projected onto the Lagrangian structure surface at each time step. This method does not conserve energy, as mass and momentum are transferred via constrained based approach.

Card 1 1 2 3 4 5 6 7 8

Variable	LAGSID	ALESID	LSIDTYP	ASIDTYP	SMMGID	ICORREC	INORM	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Card 2

Variable	BIRTH	DEATH						
Type	F	F						
Default	0.0	1.E+10						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LAGSID	A set ID defining the Lagrangian part(s) for this coupling (structures).
ALESID	A set ID defining the ALE part(s) for this coupling (fluids).
LSIDTYP	Lagrangian set ID type EQ.0: Part set ID (PSID), EQ.1: Part ID (PID).
ASIDTYP	ALE set ID type EQ.0: Part set ID (PSID), EQ.1: Part ID (PID).
SMMGID	A set ID referring to a group of one or more ALE-Multi-Material-Group (AMMG) IDs which represents the ALE materials interacting with the Lagrangian structure. This SMMGID is a set ID defined by *SET_MULTI-MATERIAL_GROUP_LIST.

VARIABLE	DESCRIPTION
ICORREC	Advection error correction method (See Remark 1). EQ.1: ALE mass is conserved. Leaked mass is moved, EQ.2: ALE mass is almost conserved, EQ.3: No correction performed (default). ALE mass is conserved. Some leakage may occur. This may be the best solution.
INORM	Type of coupling. EQ.0: Couple in all directions, EQ.1: Couple in compression and tension (free sliding), EQ.2: Couple in compression only (free sliding). This choice requires ICORREC=3.
BIRTH	Start time for coupling.
DEATH	End time for coupling.

Remarks:

1. As the ALE nodes are projected onto the closest Lagrangian surface, there may be some advection errors introduced. These errors may result in a small element mass fraction being present on the “wrong” side of the coupled Lagrangian surface. There are 3 possible scenarios:
 - a. Mass on the wrong side of the Lagrangian structure may be moved to the right side. This may cause P oscillations. No leakage will occur.
 - b. Mass on the wrong side is deleted. Mass on the right side is scaled up to compensate for the lost mass. No leakage will occur.
 - c. Mass on the wrong side is allowed (no correction performed). Some leakage may occur. This may be the most robust and simplest approach.

Example:

Model Summary:

H1 = AMMG1 = background air mesh.

H2 = AMMG2 = fluid inside container S3.

S3 = cylinder containing AMMG2.

S4 = dummy target cylinder for impact.

The gas inside S3 is AMMG2. S3 is given an initial velocity and it will impact S4.

```
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_MULTI-MATERIAL_GROUP
  1      1
  2      1
*SET_MULTI-MATERIAL_GROUP_LIST
  22
  2
*ALE_FSI_PROJECTION
$  LAGSID  ALESID  LSIDTYP  ASIDTYP  SMMGID  ICORREC  INORM
   3      1      1      1      22      3      2
$  BIRTH  DEATH
   0.0    20.0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
```

***ALE_FSI_SWITCH_MMG_{OPTION}**

Purpose: This card is used to allow the switching of an ALE multi-material-group ID (AMMGID) of a fluid as that fluid passes across a monitoring surface. This monitoring surface may be a Lagrangian shell structure, or a segment set. It does not have to be included in the slave set of the coupling card (CLIS).

Available options include:

<BLANK>

ID

An ID number (up to 8 digits) may be defined for this switch command in the first 10-character space.

or

TITLE

A title for the card may be input between the 11th and 80th character on the title-ID line. The optional title line precedes all other cards for this command.

The user can explicitly define a title for this coupling.

Title Card Format

The following card is read if and only if the ID or TITLE option is specified.

Optional

Variable	ID	TITLE
Type	I10	A70

Card 1 1 2 3 4 5 6 7 8

Variable	SID	STYPE	NQUAD	XOFF	BTIME	DTIME	NFREQ	NFOLD
Type	I	I	I	F	F	F	I	I
Default	none	0	1	0.0	0.0	1.0E20	1	0

Card 2 1 2 3 4 5 6 7 8

Variable	FR_MMG	TO_MMG	XLEN					
Type	I	I	F					
Default	none	none	0.0					

VARIABLE	DESCRIPTION
SID	A set ID defining a monitoring surface over which an ALE fluid flows across, and its ALE multi-material-group-ID (AMMGID) is switched. The monitoring surface may be a Lagrangian shell structure, or a segment set. This surface, if Lagrangian, does not have to be included in the coupling definition (see remark 4).
STYPE	Set ID type of the above SID. EQ.0: Part set ID (PSID) (default). EQ.1: Part ID (PID). EQ.2: Segment set ID (SGSID).
NQUAD	The number of flow-sensor points to be distributed over each monitoring surface/segment. There should be enough sensor points distributed to monitor the flow in each ALE element intersected by this monitoring surface (default=1, see remark 3).
XOFF	An offset distance away from the monitoring surface, beyond which the AMMGID switching occurs. The direction of XOFF is defined by the normal vector of the monitoring segment. This offset distance, in general, should be at least 2 ALE element widths away from, and beyond the monitoring interface (default=0.0).
BTIME	Start time for the AMMGID switch to be activated (default=0.0).
DTIME	Ending time for the AMMGID switch (default=1.0E20).
NFREQ	Number of computational cycles between ALE switch check (default=1).
NFOLD	Flag for checking folding logic (default=0=off). If NFOLD=1=on, then LS-DYNA will check if the monitoring segment is in the fold, applicable to airbag. If the monitoring segment is still located within a folded (shell) region, then no switching is allowed yet until it has unfolded.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
FR_MMG	This is the AMMG-SID before the switch. The AMMG-SID corresponds to the SID defined under the *SET_MULTI-MATERIAL_GROUP_LIST (SMMGL) card. This SID points to one or more AMMGs. See Remark 1.
TO_MMG	This is the AMMG-SID after the switch. The AMMG-SID corresponds to the SID defined under the *SET_MULTI-MATERIAL_GROUP_LIST card. This SID points to one or more AMMGs. See Remark 1.
XLEN	This is an absolute distance for distributing the flow sensor points over the ALE elements. To make sure that at least 1 sensor point, defined on each Lagrangian segment, is present in each ALE element to track the flow of an AMMG, XLEN may be estimated as roughly half the length of the smallest ALE element in the mesh. See Remark 3.

Remarks:

1. There is a correspondence between the FR_MMG and TO_MMG. Consider an example where:
 - a. The FR_MMG SID points to a SID=12 (the SID of its SMMGL card is 12, and this SID contains AMMG 1 and AMMG 2)
 - b. The TO_MMG points to a SID=34 (the SID of the SMMGL card is 34, and this SID contains AMMG 3 and AMMG 4)

Then, AMMG 1, if switched, will become AMMG 3, and AMMG 2, if switched, will become AMMG 4.
2. The ID option must be activated if the parameter SWID is used in the *DATABAS_FSI card. Then the accumulated mass of an AMMG that goes through a tracking surface, and being switched, will be reported via the parameter "PLEAK" in the "dbfsi" ASCII output file (or equivalently the "POROSITY" parameter inside LS-Prepost ASCII plotting option).
3. When both NQUAD and XLEN are defined, whichever gives smaller sensor-point interval distance will be used. XLEN may give better control as in the case of a null shell acting as the monitoring surface. As this null shell is stretched, NQUAD distribution of sensor-points may not be adequate, but XLEN would be.
4. The monitoring surface does not have to be included in the slave set of the coupling card. However, at least one coupling card must be present in the model. The monitoring segment set can be made up of Lagrangian or ALE nodes.

Example:

Consider a simple airbag model with 3 part IDs:

H25 = AMMG1 = Inflator gas injected into the airbag.

H24 = AMMG2 = Air outside the airbag = background mesh

H26 = AMMG3 = Dummy AMMG of inflator gas after it passes through a vent hole.

S9 = A Lagrangian shell part representing a vent hole.

S1 = A Lagrangian shell part representing the top half of an airbag.

S2 = A Lagrangian shell part representing the bottom half of an airbag.

The inflator gas inside the airbag is distinguished from the inflator gas that has passed through the monitoring surface (vent hole) to the outside of the airbag by assigning different ALE multi-material group set ID to each. The dummy fluid part (H26) should have the same material and EOS model IDs as the before-switched fluid (H25).

Fr_MMG=1 ==> points to AMMGID=1 ==> points to H25 (inflator gas if inside)

To_MMG=2 ==> points to AMMGID=3 ==> points to H26 (inflator gas if outside)

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_MULTI-MATERIAL_GROUP
    25      1
    24      1
    26      1
*DATABASE_FSI
$      TOUT      [STYPE: 0=PSID ; 1=PID ; 2=SGSID]
    0.1000
$ DBFSI_ID      SID      STYPE  AMMGSWID  LDCONVID
    1           1        1
    2           2        1
    3           9        1      90000
*SET_MULTI-MATERIAL_GROUP_LIST
    125
    1
*SET_MULTI-MATERIAL_GROUP_LIST
    126
    3
*ALE_FSI_SWITCH_MMG_ID
    90000
$      SID      SIDTYPE      NQUAD      XOFF      BTIME      DTIME      NFREQ      FOLD
    9           1            3      -20.0      5.0        0.0        1          1
$  Fr_MMG  To_MMG  XCLEN
    125    126    5.
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

NOTE:

1. The *DATABASE_FSI card tracks 3 surface entities: (a) top half of an airbag, (b) bottom half of an airbag, and (c) the vent hole monitoring surface where the AMMGID of the inflator gas is switched.
2. The amount of mass passing through the vent hole during the switch is output to a parameter called "pleak" in a "dbfsi" ASCII file. See *DATABASE_FSI.
3. The *ALE_FSI_SWITCH_MMG_ID card track any flow across S9 and switch the AMMGSID from 125 (AMMG 1) to 126 (AMMG 3).

***ALE_MULTI-MATERIAL_GROUP**

Purpose: This command defines the appropriate ALE material groupings for interface reconstruction when many ALE Multi-Material Groups (AMMG) are present in a model. This card is required when ELFORM=11 in the *SECTION_SOLID card. This is the ALE Multi-Material element formulation requiring at least 2 ALE materials to be present in a model. Each data line represents 1 ALE multi-material group (AMMG), with the first line referring to group 1, second line group 2, etc. Each AMMG represents one unique “fluid” which may undergo interaction with any Lagrangian structure in the model.

Card 1 2 3 4 5 6 7 8

Variable	SID	IDTYPE						
Type	I	I						
Default	none	0						
Remarks	1							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID.
IDTYPE	Set type: EQ.0: Part set, EQ.1: Part.

Remarks:

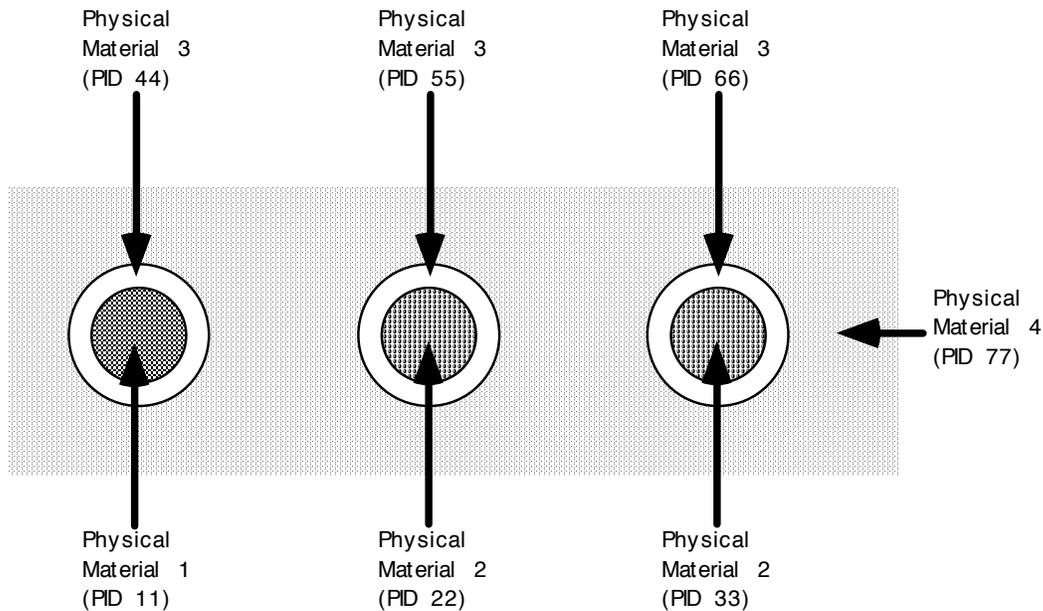
1. When ELFORM=12 in the *SECTION_SOLID card (single material and void), this card should not be used. In one model, ELFORM=12 cannot be used together with ELFORM=11. If possible, it is recommended that ELFORM=11 be used as it is the most robust and versatile formulation for treating multi-material ALE parts.
2. Each AMMG is given an ID (AMMGID), and consists of one or more PART ID’s. The interface of each AMMGID is reconstructed as it evolves dynamically. Each AMMGID is represented by one material contour color in LS-PREPOST.
3. The maximum number of AMMGIDs allowed has been increased to 20. However, there may be 2, at most 3, AMMGs inside an ALE element at anytime. If there are more than 3 AMMGs inside any 1 ALE element, the ALE mesh needs refinement. Better accuracy is obtained with 2 AMMGs in mixed elements.

- 4. To plot these AMMGIDs in LS-PREPOST:
[FCOMP] ⇒ [MISC] ⇒ [VOLUME FRACTION OF AMMGID #] ⇒ [APPLY]
(Note: Contour definitions maybe different for gas mixture application)
- 5. It is very important to distinguish among the
 - (a) Physical materials,
 - (b) PART IDs, and
 - (c) AMMGIDs.

A *PART may be any mesh component. In ALE formulation, it is simply a geometric entity and a time=0 concept. This means a *PART may be a mesh region that can be filled with one or more AMMGIDs at time zero, via a volume filling command (*INITIAL_VOLUME_FRACTION_GEOMETRY). An AMMGID represents a physical material group which is treated as one material entity (represented by 1 material color contour in LS-PREPOST plotting). AMMGID is used in dealing with multiple ALE or Eulerian materials. For example, it can be used to specify a master ALE group in a coupling card.

Example 1:

Consider a purely Eulerian model containing 3 containers containing 2 different physical materials (fluids 1 and 2). All surrounded by the background material (maybe air). The containers are made of the same material, say, metal. Assume that these containers explode and spill the fluids. We want to track the flow and possibly mixing of the various materials. Note that all 7 parts have ELFORM=11 in their *SECTION_SOLID cards. So we have total of 7 PIDs, but only 4 different physical materials.



Approach 1: If we want to track only the interfaces of the **physical** materials.

```

$... | ...1... | ...2... | ...3... | ...4... | ...5... | ...6... | ...7... | ...8
*SET_PART
  1
  11
*SET_PART
  2
  22      33
*SET_PART
  3
  44      55      66
*SET_PART
  4
  77
*ALE_MULTI-MATERIAL_GROUP
  1      0  <= 1st line = 1st AMMG => AMMGID=1
  2      0  <= 2nd line = 2nd AMMG => AMMGID=2
  3      0  <= 3rd line = 3rd AMMG => AMMGID=3
  4      0  <= 4th line = 4th AMMG => AMMGID=4
$... | ...1... | ...2... | ...3... | ...4... | ...5... | ...6... | ...7... | ...8

```

With this approach, we define only 4 AMMGs (NALEGP=4). So in LS-PREPOST, when plotting the material-group (history variable) contours, we will see 4 colors, one for each material group. One implication is that when the fluids from part 22 and part 33 flow into the same element, they will coalesce and no boundary distinction between them is maintained subsequently. While this may be acceptable for fluids at similar thermodynamic states, this may not be intuitive for solids. For example, if the solid container materials from parts 44, 55 and 66 flow into one element, they will coalesce “like a single fluid”, and no interfaces among them are tracked. If this is undesirable, an alternate approach may be taken. It is presented next.

Approach 2: If we want to reconstruct as many interfaces as necessary, in this case, we follow the interface of each part.

```

$... | ...1... | ...2... | ...3... | ...4... | ...5... | ...6... | ...7... | ...8
*ALE_MULTI-MATERIAL_GROUP
  1      1  <= 1st line = 1st AMMG => AMMGID=1
  2      1  <= 2nd line = 2nd AMMG => AMMGID=2
  3      1  <= 3rd line = 3rd AMMG => AMMGID=3
  4      1  <= 4th line = 4th AMMG => AMMGID=4
  5      1  <= 5th line = 5th AMMG => AMMGID=5
  6      1  <= 6th line = 6th AMMG => AMMGID=6
  7      1  <= 7th line = 7th AMMG => AMMGID=7
$... | ...1... | ...2... | ...3... | ...4... | ...5... | ...6... | ...7... | ...8

```

There are 7 AMMGs in this case (NALEGP=7). This will involve more computational cost for the additional tracking. Realistically, accuracy will be significantly reduced if there are more than 3 or 4 materials in any one element. In that case, higher mesh resolution may be required.

Example 2:

OIL	WATER	AIR
GROUP 1	GROUP 2	GROUP 3
PART ID'S 1 AND 2	PART ID 3	PART ID'S 5, 6, AND 7

The above example defines a mixture of three groups of materials (or “fluids”), oil, water and air, that is, the number of ALE multi-material groups (AMMGs) NALEGP=3.

The first group contains two parts (materials), part ID's 1 and 2.

The second group contains one part (material), part ID 3.

The third group contains three parts (materials), part ID's 5, 6 and 7.

***ALE_REFERENCE_SYSTEM_CURVE**

Purpose: This command defines a motion and/or a deformation prescribed for a geometric entity (where a geometric entity may be any part, part set, node set, or segment set). The motion or deformation may be completely defined by 12 parameters (shown in the equation below). These 12 parameters are defined in terms of 12 load curves. This command is required only when PRTYPE=3 in the *ALE_REFERENCE_SYSTEM_GROUP (ARSG) command.

Card 1 1 2 3 4 5 6 7 8

Variable	ID							
Type	I							
Default	none							

Card 2

Variable	LCID1	LCID2	LCID3	LCID4	LCID5	LCID6	LCID7	LCID8
Type	I	I	I	I	I	I	I	I
Default	none							

Card 3

Variable	LCID9	LCID10	LCID11	LCID12				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ID	Curve group ID.
LCID1...LCID12	Load curve ID's.

Remarks:

- The velocity of a node at coordinate (x, y, z) is defined as:

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_5 \\ f_9 \end{Bmatrix} + \begin{bmatrix} f_2 & f_3 & f_4 \\ f_6 & f_7 & f_8 \\ f_{10} & f_{11} & f_{12} \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

$f_1(t)$ is the value of load curve LCID1 at time t etc. Note that $f_1(t)$, $f_5(t)$, $f_9(t)$ correspond to the translation components in global x, y, and z direction, respectively. $f_2(t)$, $f_7(t)$, and $f_{12}(t)$ correspond to the expansion or contraction component. The remaining functions give rotation contribution.

Example 1:

Consider a motion that consists of translation in the x and y direction only. Thus only $f_1(t)$ and $f_5(t)$ are required. Hence only 2 load curve ID's need be defined:

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_REFERENCE_SYSTEM_GROUP
$   SID   STYPE  PRTYP   PRID   BCTRAN   BCEXP   BCROT   ICOORD
$     1     0     3     11     0         7         0
$   XC     YC     ZC   EXPLIM
$     0     0     0         0
*ALE_REFERENCE_SYSTEM_CURVE
$ CURVESID
$    11
$  LCID1  LCID2  LCID3  LCID4  LCID5  LCID6  LCID7  LCID8
$    111    0    0    0    222    0    0    0
$  LCID9  LCID10 LCID11 LCID12
$     0     0     0     0
*DEFINE_CURVE
$   lcid   sidr   sfa   sfo   offa   offo   dattyp
$     111
$           a1           o1
$           0.00         5.0
$           0.15         4.0
*DEFINE_CURVE
$   lcid   sidr   sfa   sfo   offa   offo   dattyp
$     222
$           a1           o1
$           0.00        -1.0
$           0.15        -5.0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

***ALE_REFERENCE_SYSTEM_GROUP**

Purpose: This card is used to associate a geometric entity to a reference system type. A geometric entity may be any part, part set, node set, or segment set of a model (or a collection of meshes). A reference system type refers to the possible transformation allowed for a geometric entity (or mesh). This command defines the type of reference system or transformation that a geometric entity undergoes. In other words, it prescribes how certain mesh can translate, rotate, expand, contract, or be fixed in space, etc.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	STYPE	PRTYPE	PRID	BCTRAN	BCEXP	BCROT	ICR/NID
Type	I	I	I	I	I	I	I	I
Default	none	0	0	0	0	0	0	0

Card 2

Variable	XC	YC	ZC	EXPLIM	EFAC		FRCPAD	IEXPND
Type	F	F	F	F	F		F	I
Default	0.0	0.0	0.0	inf.	0.0		0.1	0

Card 3 is optional

Card 3 1 2 3 4 5 6 7 8

Variable	IPIDXCL	IPIDTYP						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
SID	Set ID.
STYPE	Set type: EQ.0: part set, EQ.1: part, EQ.2: node set, EQ.3: segment set.
PRTYPE	Reference system type (See Remark 1 below) EQ.0: Eulerian, EQ.1: Lagrangian, EQ.2: Normal ALE mesh smoothing, EQ.3: Prescribed motion following load curves, see *ALE_REFERENCE_SYSTEM_CURVE, EQ.4: Automatic mesh motion following mass weighted average velocity in ALE mesh, EQ.5: Automatic mesh motion following a local coordinate system defined by three user defined nodes, see *ALE_REFERENCE_SYSTEM_NODE, EQ.6: Switching in time between different reference system types, see *ALE_REFERENCE_SYSTEM_SWITCH, EQ.7: Automatic mesh expansion in order to enclose up to twelve user defined nodes, see *ALE_REFERENCE_SYSTEM_NODE. EQ.8: Mesh smoothing option for shock waves, where the element grid contracts in the vicinity of the shock front. This may be referred to as the Delayed-ALE option. It controls how much the mesh is to be moved during the remap step. This option requires the definition of the 5th parameter in the 2nd card, EFAC; see below for definition. EQ.9: Allowing the ALE mesh(es) to: -Translate and/or rotate to follow a local Lagrangian reference coordinate system (whose *ALE_REFERENCE_SYSTEM_NODE card ID is defined by the <u>BCTRAN</u> parameter) -Expand or contract to enclose a Lagrangian part-set ID defined by the <u>PRID</u> parameter. -Has a Lagrangian node ID be defined by the <u>ICR/NID</u> parameter to be the center of the ALE mesh expansion.
PRID	A parameter giving additional information depending on the reference system (PRTYPE) choice: PRTYPE= 3: PRID defines a load curve group ID specifying an *ALE_REFERENCE_SYSTEM_CURVE card for mesh translation. This defines up to 12 curves which prescribe the motion of the system.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>PRTYPE= 5: PRID defines a node group ID specifying an *ALE_REFERENCE_SYSTEM_NODE card, via which, three nodes forming a local coordinate system are defined.</p> <p>PRTYPE= 6: PRID defines a switch list ID specifying an *ALE_REFERENCE_SYSTEM_SWITCH card. This defines the switch times and the reference system choices for each time interval between the switches.</p> <p>PRTYPE= 7: PRID defines a node group ID specifying an *ALE_REFERENCE_SYSTEM_NODE card. Up to 12 nodes in space forming a region to be enveloped by the ALE mesh are defined.</p> <p>PRTYPE= 9: PRID defines a Lagrangian part set ID (PSID) defining the Lagrangian part(s) whose range of motion is to be enveloped by the ALE mesh(es). This is useful for airbag modeling.</p>
BCTRAN	<p>For PRTYPE 4 & 5: BCTRAN is a translational constraint (remark 3). EQ.0: no constraints, EQ.1: constrained x translation, EQ.2: constrained y translation, EQ.3: constrained z translation, EQ.4: constrained x and y translation, EQ.5: constrained y and z translation, EQ.6: constrained z and x translation, EQ.7: constrained x, y, and z translation.</p> <p>For PRTYPE= 9: BCTRAN defines a node group ID defined by *ALE_REFERENCE_SYSTEM_NODE card prescribing a local coordinate system (3 node IDs) whose motion is to be followed by the ALE mesh(es).</p>
BCEXP	<p>For PRTYPE= 4 & 7: BCTRAN is an expansion constraint (remark 3). EQ.0: no constraints, EQ.1: constrained x expansion, EQ.2: constrained y expansion, EQ.3: constrained z expansion, EQ.4: constrained x and y expansion, EQ.5: constrained y and z expansion, EQ.6: constrained z and x expansion, EQ.7: constrained x, y, and z expansion.</p>

VARIABLE	DESCRIPTION
BCROT	For PRTYPE= 4: BCROT is a rotational constraint (remark 3). EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotation, EQ.5: constrained y and z rotation, EQ.6: constrained z and x rotation, EQ.7: constrained x, y, and z rotation.
ICR/NID	A flag defining the center of mesh expansion and/or rotation (remark 3). PRTYPE=4: <u>ICR</u> is a center of mesh expansion and rotation flag, EQ.0: The center is at center of gravity of the ALE mesh. EQ.1: The center is at (XC, YC, ZC), just a point in space (it does not have to be a defined node) PRTYPE=9: <u>NID</u> (node ID) is a Lagrangian NID. LSDYNA uses this node as an anchored center of ALE mesh expansion (remark 2).
XC,YC,ZC	Center of mesh expansion if PRTYPE= 4. This supplements the ICR parameter above.
EXPLIM	Limit ratio for mesh expansion and contraction. Each Cartesian direction is treated separately. The distance between the nodes is not allowed to increase by more than a factor EXPLIM, or decrease to less than a factor 1/EXPLIM. This flag applies only for PRTYPE=4.
EFAC	Initial mesh remapping factor for PRTYPE=8 only, ranging between 0.0 and 1.0. When EFAC approaches 1.0, the remapping approaches pure Eulerian behavior. The smaller the value of EFAC, the closer the mesh will initially follow the material flow in the vicinity of a shock front, i.e. approaching Lagrangian behavior. Thus, a very small value might lead to severe mesh distortions because the mesh must deform severely to follow the material flow initially. Eventually over time, the mesh smoothing behavior will approach an Eulerian system.
FRCPAD	For PRTYPE=9: This is an ALE mesh padding fraction ranging from 0.01 to 0.2. If the characteristic Lagrange mesh dimension (dL_L) exceeds $(1-2 * FRCPAD)$ times the characteristic ALE mesh dimension (dL_A), then the ALE mesh is expanded so that $dL_A = dL_L / (1-2 * FRCPAD)$. This provides extra few layers of ALE elements beyond the maximum Lagrangian range of motion. EQ.0.01: $dL_A = dL_L / 0.98 = dL_L * 1.020408$ EQ.0.20: $dL_A = dL_L / 0.60 = dL_L * 1.666667$

VARIABLE	DESCRIPTION
IEXPND	If PRTYPE=9: This is an ALE mesh expansion control flag. EQ.0: Both mesh expansion and contraction are allowed. EQ.1: Only mesh expansion is allowed.
IPIDXCL	An ALE set ID to be excluded from the expansion and/or contraction only. Translation and rotation are allowed. For example, this may be used to prevent the ALE mesh (or part) at the inflator gas inlet region from expanding too much. High ALE mesh resolution is usually required to resolve the high speed flow of the gas into the airbag via point sources (remark 2).
IPIDTYPE	Set ID type of IPIDXCL: 0 = PSID; 1 = PID

Remarks:

1. Some PRTYP may require a supplemental definition defined via corresponding PRID. For example, PRTYP=3 requires a *ALE_REFERENCE_SYSTEM_CURVE card. If PRID=n, then in the corresponding *ALE_REFERENCE_SYSTEM_CURVE card, ID=n. Similar association applies for any PRTYP (i.e. 3, 5, 6, or 7) which requires a definition for its corresponding PRID parameter.
2. For PRTYPE=9: ICR/NID can be useful to keep a high density ALE mesh centered on the region of greatest interest, (such as the inflator orifices region in an airbag model). For example, in the case of nonsymmetrical airbag deployment, assuming that the ALE mesh is initially finer near the inlet orifices, and gradually coarsened away from it. Defining an “anchor node” at the center of the orifice location will keep the fine ALE mesh region centered on the orifice region. So that this fine ALE mesh region will not be shifted away (from the point sources) during expansion and translation. The ALE mesh can move and expand outward to envelop the Lagrangian airbag in such a way that the inlet is well resolved throughout the deployment.
3. The table below shows the applicability of the various choices of PRTYPE. Simple deductions from the functional definitions of the PRTYPE choices will clarify the applications of the various constraints. For example, when PRTYP=3, nodal motion of the ALE mesh is completely controlled by the 12 curves. Therefore, no constraints are needed.

PRTYPE	ICR/NID	BCTHAN	BCROT	BCEXP
3	NO	NO	NO	NO
4	YES (ICR)	YES	YES	YES
5	NO	YES	NO	NO
6	NO	NO	NO	NO
7	NO	NO	NO	YES
8	NO	NO	NO	NO
9	YES (NID)	NO	NO	NO

Example 1:

Consider a bird-strike model containing 2 ALE parts: a bird is surrounded by air (or void). A part-set ID 1 is defined containing both parts. To allow for the meshes of these 2 parts to move with their combined mass-weighted-average velocity, PRTYPE=4 is used. Note that BCEXP=7 indicating mesh expansion is constrained in all global directions.

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_REFERENCE_SYSTEM_GROUP
$   SID      STYPE    PRTYP    PRID    BCTHAN    BCEXP    BCROT    ICOORD
$   1         0        4         0         0         7         0
$   XC        YC        ZC      EXPLIM
$   0         0         0         0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

Example 2:

Consider a bouncing ball model containing 2 ALE parts: a solid ball (PID 1) is surrounded by air or void (PID 2). A part-set ID 1 is defined containing both parts. To allow for the meshes of these 2 parts to move with 2 reference system types: (a) first, they move with their combined mass-weighted-average velocity between 0.0 and 0.01 second; and subsequently (between 0.01 and 10.0 seconds) their reference system is switched to (b) an Eulerian system (thus the mesh is fixed in space), a reference system “SWITCH” is required. This is done by setting PRTYPE=6. This PRTYPE requires a corresponding *ALE_REFERENCE_SYSTEM_SWITCH card. Note that PRID=11 in the *ALE_REFERENCE_SYSTEM_GROUP card corresponds to the SWITCHID=11 in *ALE_REFERENCE_SYSTEM_SWITCH card.

```
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_REFERENCE_SYSTEM_GROUP
$   SID      STYPE      PRTYP      PRID      BCTRAN      BCEXP      BCROT      ICOORD
$   1         0         6         11         0         7         7
$   XC        YC        ZC        EXPLIM    EULFACT    SMOOTHVMX
$   0         0         0         0         0.0
*ALE_REFERENCE_SYSTEM_SWITCH
$ SWITCHID
$   11
$   t1        t2        t3        t4        t5        t6        t7
$   0.01      10.0
$   TYPE1     TYPE2     TYPE3     TYPE4     TYPE5     TYPE6     TYPE7     TYPE8
$   4         0
$   ID1      ID2      ID3      ID4      ID5      ID6      ID7      ID8
$   0         0         0         0         0         0         0         0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
```

*ALE

*ALE_REFERENCE_SYSTEM_NODE

*ALE_REFERENCE_SYSTEM_NODE

Purpose: This command defines a group of nodes that control the motion of an ALE mesh. It is used only when PRTYPE=5 or 7 in a corresponding *ALE_REFERENCE_SYSTEM_GROUP card.

Card 1 1 2 3 4 5 6 7 8

Variable	ID							
Type	I							
Default	none							

Card 2

Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I
Default	none							

Card 3

Variable	NID9	NID10	NID11	NID12				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ID	Node group ID for PRTYPE 5 or 7, see *ALE_REFERENCE_SYSTEM_GROUP.
NID1...NID12	User specified nodes.

Remarks:

- For PRTYPE=5 the ALE mesh is forced to follow the motion of a coordinate system, which is defined by three nodes (NID1, NID2, NID3). These nodes are located at x_1 , x_2 and x_3 , respectively. The axes of the coordinate system, x' , y' and z' , are defined as:

$$\begin{aligned}
 x' &= (x_2 - x_1) / |x_2 - x_1| \\
 z' &= x' \times (x_3 - x_1) / |x' \times (x_3 - x_1)| \\
 y' &= z' \times x'
 \end{aligned}$$

Note that $x_1 \rightarrow x_2$ is the local x' axis, $x_1 \rightarrow x_3$ is the local y' axis and x' crosses y' gives the local z' axis. These 3 nodes are used to locate the reference system at any time. Therefore, their positions relative to each other should be as close to an orthogonal system as possible for better transformation accuracy of the ALE mesh.

- For PRTYPE=7, the ALE mesh is forced to move and expand, so as to enclose up to twelve user defined nodes (NID1...NID12). This is a rarely used option.

Example 1:

Consider modeling sloshing of water inside a rigid tank. Assuming there are 2 ALE parts, the water (PID 1) and air or void (PID 2) contained inside a rigid (Lagrangian) tank (PID 3). The outer boundary nodes of both ALE parts are merged with the inner tank nodes. A part-set ID 1 is defined containing both ALE parts (PIDs 1 and 2). To allow for the meshes of the 2 ALE parts to move with the rigid Lagrangian tank, PRTYPE=5 is used. The motion of the ALE parts then follows 3 reference nodes on the rigid tank. These 3 reference nodes must be defined by a corresponding *ALE_REFERENCE_SYSTEM_NODE card. In this case the reference nodes have the nodal IDs of 5, 6 and 7. Note that PRID=12 in the

*ALE_REFERENCE_SYSTEM_GROUP card corresponds to the SID=12 in the *ALE_REFERENCE_SYSTEM_NODE card.

```

$... | ...1... | ...2... | ...3... | ...4... | ...5... | ...6... | ...7... | ...8
*ALE_REFERENCE_SYSTEM_GROUP
$   SID      STYPE      PRTYP      PRID      BCTRAN      BCEXP      BCROT      ICOORD
$   1         0         5         12
$   XC        YC        ZC        EXPLIM
$   0         0         0         0
*ALE_REFERENCE_SYSTEM_NODE
$   NSID
$   12
$   N1        N2        N3        N4        N5        N6        N7        N8
$   5         6         7
$... | ...1... | ...2... | ...3... | ...4... | ...5... | ...6... | ...7... | ...8
    
```

*ALE

*ALE_REFERENCE_SYSTEM_SWITCH

*ALE_REFERENCE_SYSTEM_SWITCH

Purpose: The PRTYPE parameter in the *ALE_REFERENCE_SYSTEM_GROUP (ARSG) card allows many choices of the reference system types for any ALE geometric entity. This command allows for the time-dependent switches between these different types of reference systems, i.e., switching to multiple PRTYPEs at different times during the simulation. This command is required only when PRTYPE=6 in ARSG card. Please see example 2 in the ARSG section.

Card 1 1 2 3 4 5 6 7 8

Variable	ID							
Type	I							
Default	none							

Card 2

Variable	T1	T2	T3	T4	T5	T6	T7	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Card 3

Variable	TYPE1	TYPE2	TYPE3	TYPE4	TYPE5	TYPE6	TYPE7	TYPE8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Card 4	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I
Default	none							

VARIABLE	DESCRIPTION
ID	Switch list ID, see *ALE_REFERENCE_SYSTEM_GROUP,
T1...T7	Times for switching reference system type. By default, the reference system TYPE1 occurs between time=0 and time=T1, and TYPE2 occurs between time=T1 and time=T2, etc.
TYPE1...TYPE8	Reference system types (also see PRTYPE under ARSG): EQ.0: Eulerian, EQ.1: Lagrangian, EQ.2: Normal ALE mesh smoothing, EQ.3: Prescribed motion following load curves, see *ALE_REFERENCE_SYSTEM_CURVE, EQ.4: Automatic mesh motion following mass weighted average velocity in ALE mesh, EQ.5: Automatic mesh motion following a local coordinate system defined by three user defined nodes, see *ALE_REFERENCE_SYSEM_NODE,
ID1...ID8	The corresponding PRID parameters supporting each PRTYPE used during the simulation.

Remarks:

1. The beginning time is assumed to be t=0, and the starting PRTYPE is TYPE1. So at T1, the 1st switching time, PRTYPE is switched from TYPE1 to TYPE2, and so forth. This option can be complex in nature so it is seldom applied.

***ALE_SMOOTHING**

Purpose: This smoothing constraint keeps a node at its initial parametric location along a line between two other nodes. This constraint is active during each mesh smoothing operation.

Card 1	1	2	3	4	5	6	7	8
Variable	SNID	MNID1	MNID2	IPRE	XCO	YCO	ZCO	
Type	I	I	I	I	F	F	F	
Default	none	none	none	0	0.0	0.0	0.0	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SNID	Slave node ID, see Figure 2.1.
MNID1	First master node ID.
MNID2	Second master node ID.
IPRE	EQ.0: smoothing constraints are performed after mesh relaxation, EQ.1: smoothing constraints are performed before mesh relaxation.
XCO	x-coordinate of constraint vector
YCO	y-coordinate of constraint vector
ZCO	z-coordinate of constraint vector

Remarks:

1. Arbitrary Lagrangian Eulerian meshes are defined via the choice of the element type and the *CONTROL_ALE card. This can only be used with solid elements.

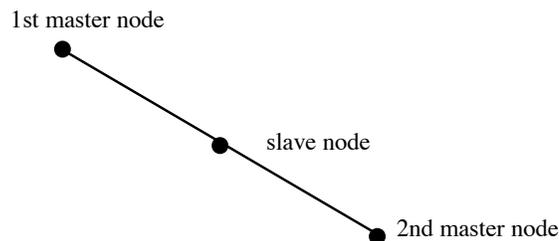


Figure 2.1 This simple constraint, which ensures that a slave node remains on a straight line between two master nodes, is sometimes necessary during ALE smoothing.

*ALE_TANK_TEST

Purpose: This command allows for the airbag information input $(\dot{m}(t), \bar{T}_{gas}(t))$ of the control volume (*AIRBAG_) approach to be used as input for the ALE/Eulerian fluid-structure interaction model of the airbag. It complements and must be used together with the *SECTION_POINT_SOURCE command. Please see *SECTION_POINT_SOURCE for additional information.

Card 1 1 2 3 4 5 6 7 8

Variable	MDOTLC	TANKV	PAMB	PFINAL	MACHL	VELMAX	AORIF	
Type	I	I	I	I	F	F	F	
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	

Card 2

Variable	AMGIDG	AMGIDA	NUMPNT					
Type	I	I	I					
Default	0	0	50					

VARIABLE**DESCRIPTION**

MDOTLC	LCID for mass flow rate as a function of time. This may be obtained directly from the control-volume type input data.
TANKV	Volume of the tank used in a tank test from which the tank pressure is measured, and $\dot{m}(t)$ and $\bar{T}_{gas}(t)$ are computed from this tank pressure data.
PAMB	The pressure inside the tank before jetting (usually 1bar).
PFINAL	The final equilibrated pressure inside the tank from the tank test.
MACHL	A limiting MACH number for the gas at the throat (MACH=1 preferred).

<u>VARIABLE</u>	<u>DESCRIPTION</u>
VELMAX	Maximum allowable gas velocity across the inflator orifice (not preferred).
AORIF	Total inflator orifice area (optional, only needed if the *SECTION_POINT_SOURCE card is not used).
AMGIDG	The ALE multi-material group ID (AMMGID) of the gas.
AMGIDA	The ALE multi-material group ID (AMMGID) of the air.
NUMPNT	The number of points in $\dot{m}(t)$ and $\bar{T}_{gas}(t)$ curves. If NUMPNT=0, defaults to 50 points.

Remarks:

- In an airbag inflator tank test, the tank pressure data is measured. This pressure is used to derive $\dot{m}(t)$ and the estimated $\bar{T}_{gas}(t)$, usually via a lumped-parameter method, a system of conservation equations and EOS. These 2 curves are used as the direct input for the control volume method in LS-DYNA via the *AIRBAG_ cards. Typically, $\bar{T}_{gas}(t)$ is the stagnation temperature of the incoming inflator gas. In an ALE or Eulerian fluid-structure interaction analysis, the gas velocity, $vel(t)$, and density, $\rho(t)$, at the inlet must be computed. Since only $\dot{m}(t)$ is known, additional assumptions about the inlet condition must be made to compute both $vel(t)$ and $\rho(t)$ curves from the information available. If this computation is done outside of LS-DYNA, then $\dot{m}(t)$ and $\bar{T}_{gas}(t)$ are used to compute 3 curves which are then used as the input for the ALE model: $\bar{T}_{gas_corrected}(t)$, $vel(t)$ and $\rho(t)$. This *ALE_TANK_TEST card allows for this inlet condition conversion to be done inside LS-DYNA. Thus, with this card together with the *SECTION_POINT_SOURCE card, LS-DYNA can take in directly the control volume input ($\dot{m}(t)$ and $\bar{T}_{gas}(t)$) and performs an ALE or Eulerian fluid-structure interaction analysis. The users do not have to do the conversion themselves.

If the *ALE_TANK_TEST card is present:

- The definitions of the relative volume, $v_r(t)$ and $vel(t)$ curves in the *SECTION_POINT_SOURCE card will be ignored. They are computed internally inside LS-DYNA.
- The $\dot{m}(t)$ curve will be read in on *ALE_TANK_TEST card.
- The $\bar{T}_{gas}(t)$ curve (stagnation temperature) will be read in on *SECTION_POINT_SOURCE card (not $\bar{T}_{gas_corrected}(t)$). A fine distinction between the two temperatures may be made. $\bar{T}_{gas}(t)$ is derived directly from the tank pressure data

based on a lump-parameter approach. $\bar{T}_{gas_corrected}(t)$ is computed from $\dot{m}(t)$ and $\bar{T}_{gas}(t)$ with additional isentropic and sonic flow assumption for the maximum velocity at an orifice ($\bar{T}_{gas_corrected}(t)$ is the static temperature). These assumptions are necessary since in $\dot{m}(t) = \rho(t) * vel(t) * A$, we only know $\dot{m}(t)$ (1 known) but we need $\rho(t)$ and $vel(t)$ (2 unknowns).

5. The inflator area is computed from the *SECTION_POINT_SOURCE card that has the AMMGID of the inflator gas in the *ALE_TANK_TEST card. If the *BOUNDARY_AMBIENT_EOS card is used instead of the *SECTION_POINT_SOURCE card, then the area may be input in this *ALE_TANK_TEST card.
6. The reference density of the propellant “gas”, ρ_0 , is computed internally and automatically used for the calculation. The ρ_0 value from the *MAT_NULL card is ignored.

Example:

Consider a tank test model consists of the inflator gas (PID 1) and the air inside the tank (PID 2). The following information from the control volume model is available:

- $\dot{m}(t)$ (LCID 1 is from control volume model input).
- $\bar{T}_{gas}(t)$ (LCID 2 is from control volume model input).
- Volume of the tank used in the inflator tank test.
- Final equilibrated pressure inside the tank.
- Ambient pressure in the air.

Also available are:

- The nodal IDs of the nodes defining the orifice holes through which the gas flows into the tank.
- The area associated with each hole (the node is assumed to be at the center of this area).
- The vector associated with each hole defining the direction of flow.

In the input below LCID 1 and 2 are $\dot{m}(t)$ and $\bar{T}_{gas}(t)$, respectively. LCID 4 and 5 will be ignored when the *ALE_TANK_TEST card is present. If it is not present, all 3 curves in the *SECTION_POINT_SOURCE card will be used. When the *SECTION_POINT_SOURCE card is present, the element formulation is equivalent to an ELFORM=11.

***ALE**

***ALE_TANK_TEST**

```
$...|...1....|...2....|...3....|...4....|...5....|...6....|...7....|...8
*PART
inflator gas
$   PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   1         1         1         0         0         0         0         0
*PART
air inside the tank
$   PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   2         2         2         0         0         0         0         0
*SECTION SOLID
$   SECID    ELFORM      AET
   2         11         0
*ALE_MULTI-MATERIAL_GROUP
$   SID      SIDTYPE
   1         1
   2         1
*SECTION POINT_SOURCE
$   SECID    LCIDT    LCIDVOLR    LCIDVEL      <= 3 curves in tempvolrvel.k file
   1         2         4         5
$   NODEID    VECTID    AREA
   24485     3         15.066
   ...
   24557     3         15.066
*ALE TANK TEST
$   MDOTLC    TANKV      PAMB      PFINAL      MACHL      VELMAX      AORIF
   1         6.0E7     1.0E-4    5.288E-4    1.0        0.0
$   AMGIDG    AMGIDA     NUPNT
   1         2         80
$...|...1....|...2....|...3....|...4....|...5....|...6....|...7....|...8
```

***ALE_UP_SWITCH**

Purpose: For the simulation of airbag inflation process, this card allows the switching from an ALE computation to a control volume (CV) or uniform pressure (UP) method at a user-defined switch time.

Card 1 1 2 3 4 5 6 7 8

Variable	UPID	SWTIME						
Type	I	F						
Default	0	1.0e+16						
Remarks	1							

Card 2

Variable	FSI_ID1	FSI_ID2	FSI_ID3	FSI_ID4	FSI_ID5	FSI_ID6	FSI_ID7	FSI_ID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0
Remarks								

Optional Card 3 Format – input only if UPID=0 or not defined.

Card 3	1	2	3	4	5	6	7	8
Variable	SID	SIDTYPE	MMGAIR	MMGGAS				
Type	I	I	I	I				
Default	0	0	0	0				
Remarks								

<u>VARIABLE</u>	<u>DESCRIPTION</u>
UPID	<p>An ID defines a corresponding *AIRBAG_HYBRID_ID card for use in an ALE-method-switching-to-CV-method simulation. The simulation starts with ALE computational method, then switches to a CV (or UP) method at some given time.</p> <p>EQ.0: (or blank) The code will construct an equivalent *AIRBAG_HYBRID_ID card automatically internally, (default). The 3rd optional line is then a required input.</p> <p>NE.0: An ID points to a corresponding *AIRBAG_HYBRID_ID card which must be defined for use after the switch. If UPID is defined, do not define the 3rd optional card.</p>
SWTIME	<p>The time at which the computation does a switch from an ALE-method-to-CV-method.</p>
FSI_ID1→FSI_ID8	<p>Coupling IDs for one or more ALE fluid-structure-interaction (FSI) *CONSTRAINED_LAGRANGE_IN_SOLID_ID cards. These couplings are deleted during the 2nd, CV computational phase.</p>
SID	<p>A set ID defines the Lagrangian parts which make up the airbag.</p>
SIDTYPE	<p>Set ID type for the above SETID (following the conventions in *AIRBAG_HYBRID card).</p> <p>EQ.0: SID is a segment set ID (SGSID).</p> <p>NE.0: SID is a part set ID (PSID).</p>
MMGAIR	<p>The AMMG (ALE multi-material group) ID of surrounding air.</p>
MMGGAS	<p>The AMMG ID of inflator gas injected into the airbag.</p>

Remarks:

1. If UPID is zero or blank, optional card 3 must be defined. LSDYNA will construct an equivalent *AIRBAG_HYBRID_ID card automatically.

Example 1:

Consider an airbag model with a 2-phase simulation: an ALE calculation being switched to a CV method. During the CV phase, the simulation is defined by an *AIRBAG_HYBRID_ID card.

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_UP_SWITCH
$  UP_ID  SW_time
  100000  2.0000
$ FSI_ID_1 FSI_ID_2 FSI_ID_3 FSI_ID_4 FSI_ID_5 FSI_ID_6 FSI_ID_7 FSI_ID_8
   1      2
$-----
*AIRBAG_HYBRID_ID
$  ID
  100000
$  SID  SIDTYP  RBID  VSCA  PSCA  VINI  MWD  SPSF
   2      1      0    1.0    1.0    0.0    0.0    0.0
$ 2  ATMT  ATMP  ATMD  GC  CC
   293.  1.0130e-4  1.200E-9  8.3143  1.
$  C23  LCC23  A23  LCA23  CP23  LCP23  AP23  LCAP23

$  OPT  PVENT  NGAS
                   4
$bac LCIDM  LCIDT  NOTUSED  MW  INITM  A  B  C
   1001  1002                0.0288691  1.0  28.98
$  FMASS

$air LCIDM  LCIDT  NOTUSED  MW  INITM  A  B  C
   1600  1603                28.97E-3  0.0  26.38  8.178e-3 -1.612e-6
$  FMASS

$pyroLCIDM  LCIDT  NOTUSED  MW  INITM  A  B  C
   1601  1603                43.45E-3  0.0  32.87  2.127e-2 -5.193E-6
$  FMASS

$sto_LCIDM  LCIDT  NOTUSED  MW  INITM  A  B  C
   1602  1603                39.49E-3  0.0  22.41  2.865e-3 -6.995e-7
$  FMASS
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

Example 2:

Consider the same airbag model with the same 2-phase simulation. However, all the *AIRBAG_HYBRID_ID card definitions are extracted automatically from the ALE model. There is no need to define the *AIRBAG_HYBRID_ID card. The 3rd optional card is required.

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*ALE_UP_SWITCH
$  UP_ID  SW_time
  100000  2.0000
   0      2.0000
$ FSI_ID_1 FSI_ID_2 FSI_ID_3 FSI_ID_4 FSI_ID_5 FSI_ID_6 FSI_ID_7 FSI_ID_8
   1      2
$  SETID  SETYPE  MMG_AIR  MMG_GAS
   2      1      2      1
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```


***BOUNDARY**

The keyword ***BOUNDARY** provides a way of defining imposed motions on boundary nodes. The keyword control cards in this section are defined in alphabetical order:

***BOUNDARY_ACOUSTIC_COUPLING**

***BOUNDARY_AMBIENT_EOS**

***BOUNDARY_CONVECTION_OPTION**

***BOUNDARY_CYCLIC**

***BOUNDARY_ELEMENT_METHOD_OPTION**

***BOUNDARY_FLUX_OPTION**

***BOUNDARY_MCOL**

***BOUNDARY_NON_REFLECTING**

***BOUNDARY_NON_REFLECTING_2D**

***BOUNDARY_PRESCRIBED_MOTION_{OPTION1}_{OPTION2}**

***BOUNDARY_PRESCRIBED_ORIENTATION_RIGID_OPTION**

***BOUNDARY_PRESSURE_OUTFLOW_OPTION**

***BOUNDARY_RADIATION_OPTION**

***BOUNDARY_SLIDING_PLANE**

***BOUNDARY_SPC_{OPTION1}_{OPTION2}**

***BOUNDARY_SPH_FLOW**

***BOUNDARY_SPH_SYMMETRY_PLANE**

***BOUNDARY_SYMMETRY_FAILURE**

***BOUNDARY_TEMPERATURE_OPTION**

***BOUNDARY_THERMAL_WELD**

***BOUNDARY_USA_SURFACE**

*BOUNDARY

*BOUNDARY_ACOUSTIC_COUPLING

*BOUNDARY_ACOUSTIC_COUPLING

Purpose: Define a segment set for acoustic coupling. The segments should define the surface of a shell or solid (structural) element. This option allows for acoustic elements (type 8 solid elements) to couple on either one side of a shell or solid element structure or both sides of a shell structure. The nodal points of the shell segments and those on either side of the segments must be coincident. If the fluid exists on just one side of the segment and if the nodes are merged, no input is necessary and input data in this section is not needed. Two-sided coupling will not work if the interface nodes are merged out.

Card 1 2 3 4 5 6 7 8

Variable	SSID							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

SSID

Segment set ID, see *SET_SEGMENT

Remarks:

For the stability of the acoustic-structure coupling, the following condition must be satisfied:

$$\frac{2\rho_a D}{\rho_s t_s} < 5$$

where ρ_a is the density of the acoustic medium, D is the total thickness of the acoustic elements adjacent to the structural element, ρ_s is the density, and t_s is the thickness of the structural shell element.

***BOUNDARY_ AMBIENT_EOS**

Purpose: This command defines the IDs of 2 load curves: (1) internal energy per unit reference specific volume (or temperature if using *EOS_IDEAL_GAS) and (2) relative volume. These 2 curves completely prescribe the thermodynamic state as a function of time for any ALE or Eulerian part with an “ambient” type element formulation (please see Remark 4).

Card 1 2 3 4 5 6 7 8

Variable	PID	LCID1	LCID2						
Type	I	I	I						
Default	none	none	none						

VARIABLE

DESCRIPTION

PID The ambient Part ID for which the thermodynamic state is being defined.

LCID1 A load curve ID for internal energy per unit reference specific volume (please read the beginning of the EOS section for details). If *EOS_IDEAL_GAS is being used, this ID then refers to a temperature load curve ID.

LCID2 Load curve ID for relative volume, $v_r = \left(\frac{v}{v_0} = \frac{\rho_0}{\rho} \right)$. (Please read the beginning of the EOS section for details).

Remarks:

1. The term “ambient” refers to a medium that has predetermined thermodynamic state throughout the simulation. All “ambient” parts/elements will have its thermodynamic state reset back to this predetermined state every cycle. If this state is defined via the *EOS_ card, then this predetermined thermodynamic state is constant throughout the simulation. If it is defined via this card, *BOUNDARY_ AMBIENT_EOS, then its thermodynamic state will vary according to these defined load curves. “Ambient” part is sometimes also referred to as “reservoir” part as it may be used to simulate semi-infinite region.
2. In general, a thermodynamic state of a non-reacting and no-phase-change material may be defined by 2 thermodynamic variables. By defining (a) an internal energy per unit reference specific volume load curve (or a temperature load curve if using

*EOS_IDEAL_GAS) and (b) a relative volume load curve, the pressure as a function of time for this ambient part ID can be computed directly via the equation of state (*EOS_).

3. A reference specific volume, $v_0 = \frac{1}{\rho_0}$, is the inverse of a reference density, ρ_0 . The reference density is defined as the density at which the material is under a reference or nominal state. Please refer to the *EOS section for additional explanation on this.
4. The internal energy per unit reference specific volume may be defined as $e_{ipv0} = \frac{C_v T}{v_0}$. The specific internal energy (or internal energy per unit mass) is defined as $C_v T$.
5. This card is only to be used with “ambient” element type as defined by the parameters under the *SECTION_SOLID card:
 - ELFORM = 7, or
 - ELFORM = 11 and AET=4, or
 - ELFORM = 12 and AET=4.

Example:

Consider an ambient ALE part ID 1 which has its internal energy per unit reference specific volume in a load curve ID 2 and relative volume load curve ID 3:

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*BOUNDARY_AMBIENT_EOS
$      PID  e/T_LCID  rvol_LCID
      1      2      3
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

***BOUNDARY_CONVECTION_OPTION**

Available options include:

SEGMENT

SET

Purpose: Define convection boundary conditions for a thermal or coupled thermal/structural analysis. Two cards are defined for each option.

For the **SET** option define the following card:

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	SSID							
Type	I							
Default	none							

For the **SEGMENT** option define the following card:

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

*BOUNDARY

*BOUNDARY_CONVECTION

Define the following card for both options:

(Card 2 of 2)

Card 2	1	2	3	4	5	6	7	8
--------	---	---	---	---	---	---	---	---

Variable	HLCID	HMULT	TLCID	TMULT	LOC			
Type	I	F	I	F	I			
Default	none	0.	none	0.	0			

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT.
N1,N2...	Node ID's defining segment.
HLCID	Load curve ID for heat transfer coefficient, h : GT.0: function versus time, EQ.0: use constant multiplier value, HMULT, LT.0: function versus temperature.
HMULT	Curve multiplier for h .
TLCID	Load curve ID for T_∞ versus time, see *DEFINE_CURVE: EQ.0: use constant multiplier value, TMULT.
TMULT	Curve multiplier for T_∞
LOC	Application of surface for thermal shell elements, see parameter, TSHELL, in the *CONTROL_SHELL input: EQ.-1: lower surface of thermal shell element EQ. 1: upper surface of thermal shell element

Remarks:

A convection boundary condition is calculated using $\dot{q}'' = (T - T_\infty)$ where

h heat transfer coefficient

$(T - T_\infty)$ temperature potential

Three alternatives are possible for the heat transfer coefficient which can be a function of time, a function of temperature, or constant. Also, the temperature of the boundary T_∞ can be either constant or a function of time. For both curves, multipliers can be used to scale the values.

***BOUNDARY_CYCLIC**

Purpose: Define nodes in boundary planes for cyclic symmetry.

These boundary conditions can be used to model a segment of an object that has rotational symmetry such as an impeller, i.e., Figure 3.1. The segment boundary, denoted as a side 1 and side 2, may be curved or planar. In this section, a paired list of points are defined on the sides that are to be joined.

Card	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	NSID1	NSID2	IGLOBAL	ISORT	
Type	F	F	F	I	I	I	I	
Default	none	none	none	none	none	0	2	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XC	x-component axis vector of axis of rotation
YC	y-component axis vector of axis of rotation
ZC	z-component axis vector of axis of rotation
NSID1	Node set ID for first boundary plane (side 1, see Figure 3.1).
NSID2	Node set ID for second boundary plane (side 2, see Figure 3.1). Each boundary node in this boundary plane is constrained to its corresponding node in the first node set. Node sets NSID1 and NSID2 must contain the same number of nodal points. Care has to be taken that the nodes in both node sets have a location which, if given in cylindrical coordinates, all differ by the same angle.
IGLOBAL	Flag for repeating symmetry: EQ.0: Cyclic symmetry (default) EQ.1: Repeating symmetry in planes normal to global X EQ.2: Repeating symmetry in planes normal to global Y EQ.3: Repeating symmetry in planes normal to global Z
ISORT	Flag for automatic sorting of boundary nodes: EQ.0: No automatic sorting (default) EQ.1: Automatic sorting of nodes.

Remarks:

1. Each node set should lie on a plane, and these should generally be boundaries of the model.
2. Previous versions of LS-DYNA, prior to version 970, it was assumed that the nodes are correctly ordered within each set, i.e. the n^{th} node in NSID1 is equivalent to the n^{th} node in NSID2. If the ISORT flag is active, the nodes in NSID2 are automatically sorted to achieve equivalence, so the nodes can be picked by the quickest available method. However, for cyclic symmetry (IGLOBAL=0), it is assumed that the axis passes through the origin, i.e., only globally defined axes of rotation are possible.

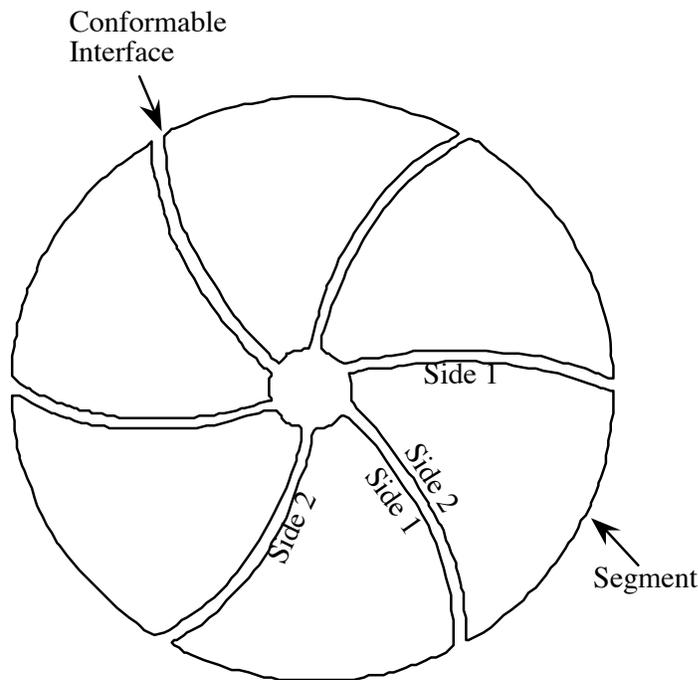


Figure 3.1 With cyclic symmetry only one segment is modeled.

***BOUNDARY_ELEMENT_METHOD_OPTION**

Available options include:

ACOUSTIC

CONTROL

FLOW

NEIGHBOR

SYMMETRY

WAKE

Purpose: Define input parameters for boundary element method analysis of incompressible fluid dynamics or fluid-structure interaction problems.

The boundary element method (BEM) can be used to compute the steady state or transient fluid flow about a rigid or deformable body. The theory which underlies the method (see the LS-DYNA Theory Manual) is restricted to inviscid, incompressible, attached fluid flow. The method should not be used to analyze flows where shocks or cavitation are present.

In practice the method can be successfully applied to a wider class of fluid flow problems than the assumption of inviscid, incompressible, attached flow would imply. Many flows of practical engineering significance have large Reynolds numbers (above 1 million). For these flows the effects of fluid viscosity are small if the flow remains attached, and the assumption of zero viscosity may not be a significant limitation. Flow separation does not necessarily invalidate the analysis. If well-defined separation lines exist on the body, then wakes can be attached to these separation lines and reasonable results can be obtained. The Prandtl-Glauert rule can be used to correct for non-zero Mach numbers in a gas, so the effects of aerodynamic compressibility can be correctly modeled (as long as no shocks are present).

The **BOUNDARY_ELEMENT_METHOD_FLOW** card turns on the analysis, and is mandatory.

*BOUNDARY

*BOUNDARY_ELEMENT_METHOD

*BOUNDARY_ELEMENT_METHOD_ACOUSTIC

Purpose: Use boundary element method in frequency domain for acoustic problems.

Card 1 1 2 3 4 5 6 7 8

Variable	Dens	Sp	Min_Freq	Max_freq	NFreq	NORM	DT	Pref
Type	F	F	F	F	F	I	F	F
Default						0		
Remark								

Card 2

Variable	SSID	SSTYPE	LcID	OutID	OutTYPE	Method	Iwindow	
Type	I	I	I	I	I	I	I	I
Default								
Remark								

Optional Card (Required if iterative solver is used in BEM)

Card 3 1 2 3 4 5 6 7 8

Variable	IDim	Max_Iter	Res					
Type	I	I	F					
Default	50	100	10.e-6					

VARIABLE	DESCRIPTION
Dens	Fluid Density.
Sp	Speed of Sound of the Fluid
Min_Freq	Minimum value of output frequencies
Max_Freq	Maximum value of output frequencies
NFreq	Number of output frequencies
NORM	NORM should be set such that the normals are facing the Fluid. EQ.0: Normals are not inverted (Default) EQ.1: Normals are inverted.
DT	Time interval used for writing data to the binary file defined by the command line parameter “bem”. For method = 0, 1, 3 the boundary element velocity is written and for method = 2 both boundary element velocity and pressure are written. See Getting Started / Execution Syntax section of this manual for information about “bem”.
Pref	Reference pressure for calculation of pressure. This value will be used to output pressure in dB. If Pref = 0 the pressure will be in the problem units.
SSID	Part, Part set ID, or Segment set ID of boundary elements
SSTYPE	Boundary element type EQ.0: Part Set ID EQ.1: Part ID EQ.2: Segment set ID
LcID	Load curve for velocity boundary condition (and pressure boundary condition when Kirchoff method is used) in frequency domain LT.0: Velocity is given in the frequency domain EQ.0: Velocity (pressure) is provided by the LS-DYNA analysis, the velocity is written in a binary file. The boundary element method is processed at the end of the LS-DYNA analysis. An FFT algorithm is used to transform time domain data into frequency domain in order to use the boundary element method acoustic. For LcId=0, a unique name must be specified during execution with the command line parameter bem=bof, where “bof” is a filename. See Getting Started / Execution Syntax section of this manual for information about “bem”.
OutID	Node set ID, or Segment set ID of output field points.

*BOUNDARY

*BOUNDARY_ELEMENT_METHOD

<u>VARIABLE</u>	<u>DESCRIPTION</u>
OutTYPE	Output field points type. EQ.1: Node set ID. EQ.2: Segment set ID.
Method	Acoustic formulation. EQ.0: BEM (Direct solver). EQ.1: Iterative solver is used. The next card is mandatory. EQ.2: Kirchhoff method coupled to FEM for acoustics (*MAT_ACOUSTIC) with Non Reflecting Boundary condition, see *BOUNDARY_NON_REFLECTING . In this case, at least one fluid layer with non reflecting boundary condition is to be merged to the vibrating structure. This additional fluid is given in *MAT_ACOUSTIC by the same density and sound speed as used in this card. EQ.3: Rayleigh method.
Iwindow	FFT windows (Default=1). EQ.0: Hanning window. EQ.1: Rectangular window. EQ.2: Raised cosine window. EQ.3: Hamming window . EQ.4: Blackman window.
IDim	Subspace size (Default= 50)
Max_iter	Maximum of iterations (Default =100)
Res	Residual (Default=1.E-6)

***BOUNDARY_ELEMENT_METHOD_CONTROL**

Purpose: Control the execution time of the boundary element method calculation. The **CONTROL** option is used to control the execution time of the boundary element method calculation, and the use of this option is strongly recommended. The BEM calculations can easily dominate the total execution time of a LS-DYNA run unless the parameters on this card (especially DTBEM and/or IUPBEM) are used appropriately.

DTBEM is used to increase the time increment between calls to the BEM routines. This can usually be done with little loss in accuracy since the characteristic times of the structural dynamics and the fluid flow can differ by several orders of magnitude. The characteristic time of the structural dynamics in LS-DYNA is given by the size of the smallest structural element divided by the speed of sound of its material. For a typical problem this characteristic time might be equal to 1 microsecond. Since the fluid in the boundary element method is assumed to be incompressible (infinite speed of sound), the characteristic time of the fluid flow is given by the streamwise length of the smallest surface in the flow divided by the fluid velocity. For a typical problem this characteristic time might be equal to 10 milliseconds. For this example DTBEM might be set to 1 millisecond with little loss of accuracy. Thus, for this example, the boundary element method would be called only once for every 1000 LS-DYNA iterations, saving an enormous amount of computer time.

IUPBEM is used to increase the number of times the BEM routines are called before the matrix of influence coefficients is recomputed and factored (these are time-consuming procedures). If the motion of the body is entirely rigid body motion there is no need to ever recompute and factor the matrix of influence coefficients after initialization, and the execution time of the BEM can be significantly reduced by setting IUPBEM to a very large number. For situations where the structural deformations are modest an intermediate value (e.g., 10) for IUPBEM can be used.

Define one card.

Card 1 2 3 4 5 6 7 8

Variable	LWAKE	DTBEM	IUPBEM	FARBEM				
Type	I	F	I	F				
Default	50	0.	100	2.0				
Remark	1			2				

VARIABLE	DESCRIPTION
LWAKE	Number of elements in the wake of lifting surfaces. Wakes must be defined for all lifting surfaces.
DTBEM	Time increment between calls to the boundary element method. The fluid pressures computed during the previous call to the BEM will continue to be used for subsequent LS-DYNA iterations until a time increment of DTBEM has elapsed.
IUPBEM	The number of times the BEM routines are called before the matrix of influence coefficients is recomputed and refactored.
FARBEM	Nondimensional boundary between near-field and far-field calculation of influence coefficients.

Remarks:

1. Wakes convect with the free-stream velocity. The number of elements in the wake should be set to provide a total wake length equal to 5-10 times the characteristic streamwise length of the lifting surface to which the wake is attached. Note that each wake element has a streamwise length equal to the magnitude of the free stream velocity multiplied by the time increment between calls to the boundary element method routines. This time increment is controlled by DTBEM.
2. The most accurate results will be obtained with FARBEM set to 5 or more, while values as low as 2 will provide slightly reduced accuracy with a 50% reduction in the time required to compute the matrix of influence coefficients.

***BOUNDARY_ELEMENT_METHOD_FLOW**

Purpose: Turn on the boundary element method calculation, specify the set of shells which define the surface of the bodies of interest, and specify the onset flow.

The *BOUNDARY_ELEMENT_METHOD_FLOW card turns on the BEM calculation. This card also identifies the shell elements which define the surfaces of the bodies of interest, and the properties of the onset fluid flow. The onset flow can be zero for bodies which move through a fluid which is initially at rest.

Define one card.

Card 1 2 3 4 5 6 7 8

Variable	SSID	VX	VY	VZ	RO	PSTATIC	MACH	
Type	I	F	F	F	F	F	F	
Default	none	none	none	none	none	0.	0.	
Remark	1					2	3	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SSID	Shell set ID for the set of shell elements which define the surface of the bodies of interest (see *SET_SHELL). The nodes of these shells should be ordered so that the shell normals point into the fluid.
VX, VY, VZ	x, y, and z components of the free-stream fluid velocity.
RO	Fluid density.
PSTATIC	Fluid static pressure.
MACH	Free-stream Mach number.

Remarks:

1. It is recommended that the shell segments in the SSID set use the NULL material (see *MAT_NULL). This will provide for the display of fluid pressures in the post-processor. For triangular shells the 4th node number should be the same as the 3rd node number. For fluid-structure interaction problems it is recommended that the boundary element shells use the same nodes and be coincident with the structural shell elements (or the outer face of solid elements) which define the surface of the body. This approach guarantees that the boundary element segments will move with the surface of the body as it deforms.
2. A pressure of PSTATIC is applied uniformly to all segments in the segment set. If the body of interest is hollow, then PSTATIC should be set to the free-stream static pressure minus the pressure on the inside of the body.
3. The effects of subsonic compressibility on gas flows can be included using a non-zero value for MACH. The pressures which arise from the fluid flow are increased using the Prandtl-Glauert compressibility correction. MACH should be set to zero for water or other liquid flows.

***BOUNDARY_ELEMENT_METHOD_NEIGHBOR**

Purpose: Define the neighboring elements for a given boundary element segment.

The pressure at the surface of a body is determined by the gradient of the doublet distribution on the surface (see the LS-DYNA Theory Manual). The “Neighbor Array” is used to specify how the gradient is computed for each boundary element segment. Ordinarily, the Neighbor Array is set up automatically by LS-DYNA, and no user input is required. The NEIGHBOR option is provided for those circumstances when the user desires to define this array manually.

For the **NEIGHBOR** option define the following cards:

Cards 1, 2, 3, ... (The next “*” card terminates the input.)

Card 1 2 3 4 5 6 7 8

Variable	NELEM	NABOR1	NABOR2	NABOR3	NABOR4			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NELEM	Element number.
NABOR1	Neighbor for side 1 of NELEM.
NABOR2	Neighbor for side 2 of NELEM.
NABOR3	Neighbor for side 3 of NELEM.
NABOR4	Neighbor for side 4 of NELEM.

Remarks:

Each boundary element has 4 sides (Figure 3.2). Side 1 connects the 1st and 2nd nodes, side 2 connects the 2nd and 3rd nodes, etc. The 4th side is null for triangular elements.

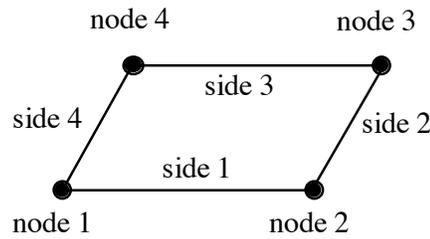


Figure 3.2 Each segment has 4 sides.

For most elements the specification of neighbors is straightforward. For the typical case a quadrilateral element is surrounded by 4 other elements, and the neighbor array is as shown in Figure 3.3.

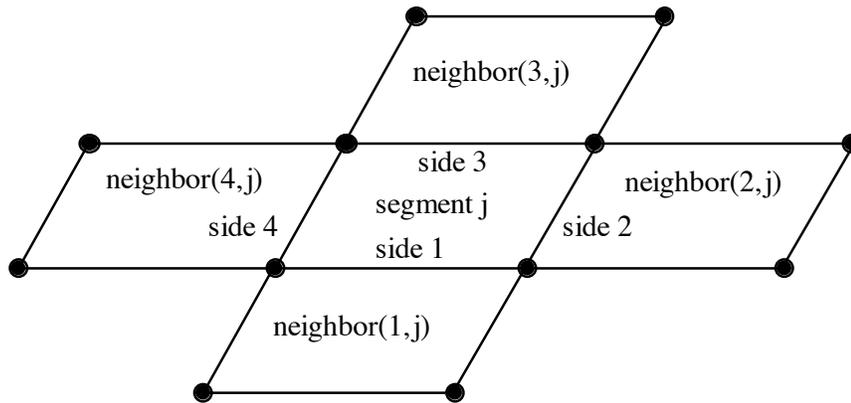


Figure 3.3 Typical neighbor specification.

There are several situations for which the user may desire to directly specify the neighbor array for certain elements. For example, boundary element wakes result in discontinuous doublet distributions, and neighbors which cross a wake should not be used. Figure 3.4 illustrates a situation where a wake is attached to side 2 of segment j. For this situation two options exist. If neighbor(2,j) is set to zero, then a linear computation of the gradient in the side 2 to side 4 direction will be made using the difference between the doublet strengths on segment j and segment neighbor(4,j). This is the default setup used by LS-DYNA when no user input is provided. By specifying neighbor(2,j) as a negative number a more accurate quadratic curve fit will be used to compute the gradient. The curve fit will use segment j, segment neighbor(4,j), and segment -neighbor(2,j); which is located on the opposite side of segment neighbor(4,j) as segment j.

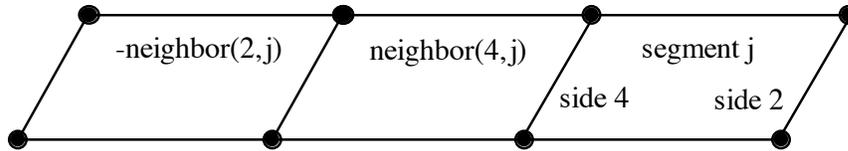


Figure 3.4 If neighbor(2,j) is a negative number it is assumed to lie on the opposite side of neighbor(4,j) as segment j.

Another possibility is that no neighbors at all are available in the side 2 to side 4 direction. In this case both neighbor(2,j) and neighbor(4,j) can be set to zero, and the gradient in that direction will be assumed to be zero. This option should be used with caution, as the resulting fluid pressures will not be accurate for three-dimensional flows. However, this option is occasionally useful where quasi-two dimensional results are desired. All of the above options apply to the side 1 to side 3 direction in the obvious ways.

For triangular boundary elements side 4 is null. Gradients in the side 2 to side 4 direction can be computed as described above by setting neighbor(4,j) to zero for a linear derivative computation (this is the default setup used by LS-DYNA when no user input is provided) or to a negative number to use the segment on the other side of neighbor(2,j) and a quadratic curve fit. There may also be another triangular segment which can be used as neighbor(4,j) (see Figure 3.5).

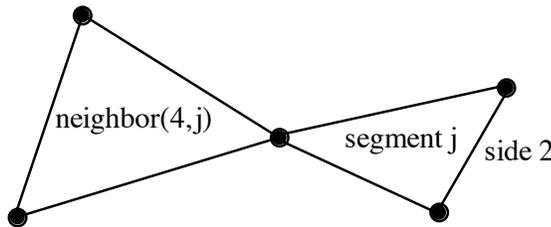


Figure 3.5 Sometimes another triangular boundary element segment can be used as neighbor (4,j).

The rules for computing the doublet gradient in the side 2 to side 4 direction can be summarized as follows (the side 1 to side 3 case is similar):

Table 3.1 Surface pressure computation for element j.

NABOR2	NABOR4	Doublet Gradient Computation
GT.0	GT.0	quadratic fit using elements j, NABOR2, and NABOR4
LT.0	GT.0	quadratic fit using elements j, -NABOR2, and NABOR4. -NABOR2 is assumed to lie on the opposite side of NABOR4 as segment j (see Fig. 3.4)
GT.0	LT.0	quadratic fit using elements j, NABOR2, and -NABOR4. -NABOR4 is assumed to lie on the opposite side of NABOR2 as segment j
EQ.0	GT.0	linear fit using elements j and NABOR4
GT.0	EQ.0	linear fit using elements j and NABOR2
EQ.0	EQ.0	zero gradient

***BOUNDARY_ELEMENT_METHOD_SYMMETRY**

Purpose: To define a plane of symmetry for the boundary element method. The SYMMETRY option can be used to reduce the time and memory required for symmetric configurations. For these configurations the reduction in the number of boundary elements by a factor of 2 will reduce the memory used by the boundary element method by a factor of 4, and will reduce the computer time required to factor the matrix of influence coefficients by a factor of 8. Only 1 plane of symmetry can be defined.

For the SYMMETRY option define the following card:

Define one card.

Card 1 2 3 4 5 6 7 8

Variable	BESYMS							
Type	I							
Default	0							
Remark								

VARIABLE

DESCRIPTION

BESYMS

Defines symmetry plane for boundary element method.

EQ.0: no symmetry plane is defined

EQ.1: $x = 0$ is a symmetry plane

EQ.2: $y = 0$ is a symmetry plane

EQ.3: $z = 0$ is a symmetry plane

*BOUNDARY

*BOUNDARY_ELEMENT_METHOD

*BOUNDARY_ELEMENT_METHOD_WAKE

Purpose: To attach wakes to the trailing edges of lifting surfaces. Wakes should be attached to boundary elements at the trailing edge of a lifting surface (such as a wing, propeller blade, rudder, or diving plane). Wakes should also be attached to known separation lines when detached flow is known to exist (such as the sharp leading edge of a delta wing at high angles of attack). Wakes are required for the correct computation of surface pressures for these situations. As described above, two segments on opposite sides of a wake should never be used as neighbors.

For the **WAKE** option define the following cards:

Cards 1, 2, 3, ... (The next "*" card terminates the input.)

Card	1	2	3	4	5	6	7	8
Variable	NELEM	NSIDE						
Type	I	I						
Default	none	none						
Remark	1							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NELEM	Element number to which a wake is attached.
NSIDE	The side of NELEM to which the wake is attached (see Fig. 3.2). This should be the "downstream" side of NELEM.

Remarks:

1. Normally two elements meet at a trailing edge (one on the "upper" surface and one on the "lower" surface). The wake can be attached to either element, but not to both.

***BOUNDARY_FLUX_OPTION**

Available options include:

SEGMENT

SET

Purpose: Define flux boundary conditions for a thermal or coupled thermal/structural analysis. Two or more cards are defined for each option. History variables can be associated with the boundary condition which will invoke a call to a user defined boundary flux subroutine for computing the flux.

For the **SET** option define the following card:

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	SSID								
Type	I								
Default	none								

For the **SEGMENT** option define the following card:

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	N3	N4					
Type	I	I	I	I					
Default	none	none	none	none					

*BOUNDARY

*BOUNDARY_FLUX

Define the following card for both options:

(Card 2)

Card 2 1 2 3 4 5 6 7 8

Variable	LCID	MLC1	MLC2	MLC3	MLC4	LOC	NHISV	
Type	I	F	F	F	F	I	I	
Default	none	0.	0.	0.	0.	0	0	

Define as many cards as necessary to initialize NHISV history variables.

(Card 3 ...)

Card 3 1 2 3 4 5 6 7 8

Variable	HISV1	HISV2	HISV3	HISV4	HISV5	HISV6	HISV7	HISV8
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE

DESCRIPTION

SSID	Segment set ID, see *SET_SEGMENT
N1,N2...	Node ID's defining segment
LCID	Load curve ID for heat flux, see *DEFINE_CURVE: GT.0: function versus time, EQ.0: use constant multiplier values at nodes, LT.0: function versus temperature.
MLC1	Curve multiplier at node N_1 , see Figure 3.2.
MLC2	Curve multiplier at node N_2 , see Figure 3.2.
MLC3	Curve multiplier at node N_3 , see Figure 3.2.
MLC4	Curve multiplier at node N_4 , see Figure 3.2.

VARIABLE	DESCRIPTION
LOC	Application of surface for thermal shell elements, see parameter, TSHELL, in the *CONTROL_SHELL input: EQ.-1: lower surface of thermal shell element EQ.1: upper surface of thermal shell element
NHISV	Number of history variables associated with the flux definition: GT.0: A user defined subroutine will be called to compute the flux. See Remark 1.
HISV1	Initial value of history variable 1
HISV2	Initial value of history variable 2
.	.
HISVn	Initial value of history variable n, where n = NHISV

Remarks:

1. If NHISV is a number greater than 0, the user subroutine
`subroutine usrflux(fl, flp, ...)`
will be called to compute the flux (fl) defined as heat (energy) per time per surface area. For more details see Appendix S.
2. Three definitions for heat flux are possible. Heat flux can be a function of time, a function of temperature, or constant values that are maintained throughout the calculation. With the definition of multipliers at each node of the segment, a bilinear spatial variation can be assumed.

By convention, heat flow is negative in the direction of the surface outward normal vector. Surface definition is in accordance with the left hand rule. The outward normal vector points to the left as one progresses from node $N_1-N_2-N_3-N_4$. See Figure 3.6.

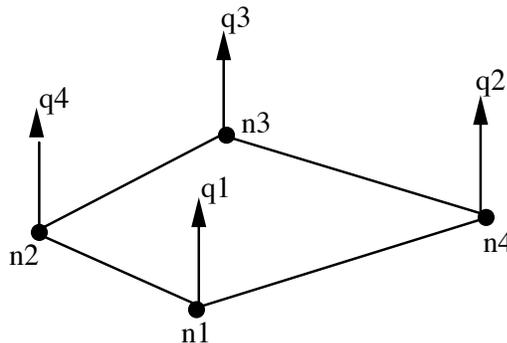


Figure 3.6. Nodal number determines outward normal.

*BOUNDARY

*BOUNDARY_MCOL

*BOUNDARY_MCOL

Purpose: Define parameters for MCOL coupling. The MCOL Program is a rigid body mechanics program for modeling the dynamics of ships. See Remark 1 for more information.

Card 1 2 3 4 5 6 7 8

Variable	NMCOL	MXSTEP	ENDTMCOL	TSUBC	PRTMCOL			
Type	I	I	F	F	F			
Default	2	none	0.0	0.0	none			
Remarks			2					

Card 2 must be defined for each ship

Card 2 1 2 3 4 5

Variable	RBMCOL	MCOLFILE			
Type	I	A60			
Default		None			
Remarks					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NMCOL	Number of ships in MCOL coupling.
MXSTEP	Maximum of time step in MCOL calculation. If the number of MCOL time steps exceeds MXSTEP, then LS-DYNA will terminate.
ENDTMCOL	Uncoupling termination time, see Remark 2 below. EQ.0.0: set to LS-DYNA termination time
TSUBC	Time interval for MCOL subcycling. EQ.0.0: no subcycling

VARIABLE	DESCRIPTION
PRTMCOL	Time interval for output of MCOL rigid body data.
RBMCOL	LS-DYNA rigid body material assignment for the ship.
MCOLFILE	Filename containing MCOL input parameters for the ship.

Remarks:

1. The basis for MCOL is a convolution integral approach for simulating the equations of motion. A mass and inertia tensor are required as input for each ship. The masses are then augmented to include the effects of the mass of the surrounding water. A separate program determines the various terms of the damping/buoyancy force formulas which are also input to MCOL. The coupling is accomplished in a simple manner: at each time step LS-DYNA computes the resultant forces and moments on the MCOL rigid bodies and passes them to MCOL. MCOL then updates the positions of the ships and returns the new rigid body locations to LS-DYNA. A more detailed theoretical and practical description of MCOL can be found in a separate report (to appear).
2. After the end of the LS-DYNA / MCOL calculation, the analysis can be pursued using MCOL alone. ENDTMCOL is the termination time for this analysis. If ENDTMCOL is lower than the LS-DYNA termination time, the uncoupled analysis will not be activated.
3. The MCOL output is set to the files MCOLOUT (ship position) and MCOLENERGY (energy breakdown). In LS-PREPOST, MCOLOUT can be plotted through the rigid body time history option and MCOLENERGY.

*BOUNDARY

*BOUNDARY_NON_REFLECTING

*BOUNDARY_NON_REFLECTING

Purpose: Define a non-reflecting boundary. This option applies to continuum domains modeled with solid elements, as indefinite domains are usually not modeled. For geomechanical problems this option is important for limiting the size of the models.

Card 1 1 2 3 4 5 6 7 8

Variable	SSID	AD	AS					
Type	I	F	F					
Default	none	0.0	0.0					
Remarks	1, 2	3	3					

VARIABLE

DESCRIPTION

SSID	Segment set ID, see *SET_SEGMENT.
AD	Default activation flag for dilatational waves. (on.EQ.0.0, off.NE.0.0)
AS	Default activation flag for shear waves. (on.EQ.0.0, off.NE.0.0)

Remarks:

1. Non-reflecting boundaries defined with this keyword are only used with three-dimensional solid elements. Boundaries are defined as a collection of segments, and segments are equivalent to element faces on the boundary. Segments are defined by listing the corner nodes in either a clockwise or counterclockwise order.
2. Non-reflecting boundaries are used on the exterior boundaries of an analysis model of an infinite domain, such as a half-space to prevent artificial stress wave reflections generated at the model boundaries from reentering the model and contaminating the results. Internally, LS-DYNA computes an impedance matching function for all non-reflecting boundary segments based on an assumption of linear material behavior. Thus, the finite element mesh should be constructed so that all significant nonlinear behavior is contained within the discrete analysis model.
3. With the two optional switches, the influence of reflecting waves can be studied.

4. During the dynamic relaxation phase (optional), nodes on non-reflecting segments are constrained in the normal direction. Nodal forces associated with these constraints are then applied as external loads and held constraint in the transient phase while the constraints are replaced with the impedance matching functions. In this manner, soil can be quasi-statically prestressed during the dynamic relaxation phase and dynamic loads (with non-reflecting boundaries) subsequently applied in the transient phase.

*BOUNDARY

*BOUNDARY_NON_REFLECTING_2D

*BOUNDARY_NON_REFLECTING_2D

Purpose: Define a non-reflecting boundary. This option applies to continuum domains modeled with two-dimensional solid elements in the xy plane, as indefinite domains are usually not modeled. For geomechanical problems, this option is important for limiting the size of the models.

Card 1 2 3 4 5 6 7 8

Variable	NSID							
Type	I							
Default	none							
Remarks	1, 2							

VARIABLE

DESCRIPTION

NSID Node set ID, see *SET_NODE. See Figure 3.7.

Remarks:

1. Non-reflecting boundaries defined with this keyword are only used with two-dimensional solid elements in either plane strain or axisymmetric geometries. Boundaries are defined as a sequential string of nodes moving counterclockwise around the boundary.
2. Non-reflecting boundaries are used on the exterior boundaries of an analysis model of an infinite domain, such as a half-space to prevent artificial stress wave reflections generated at the model boundaries from reentering the model and contaminating the results. Internally, LS-DYNA computes an impedance matching function for all non-reflecting boundary segments based on an assumption of linear material behavior. Thus, the finite element mesh should be constructed so that all significant nonlinear behavior is contained within the discrete analysis model.

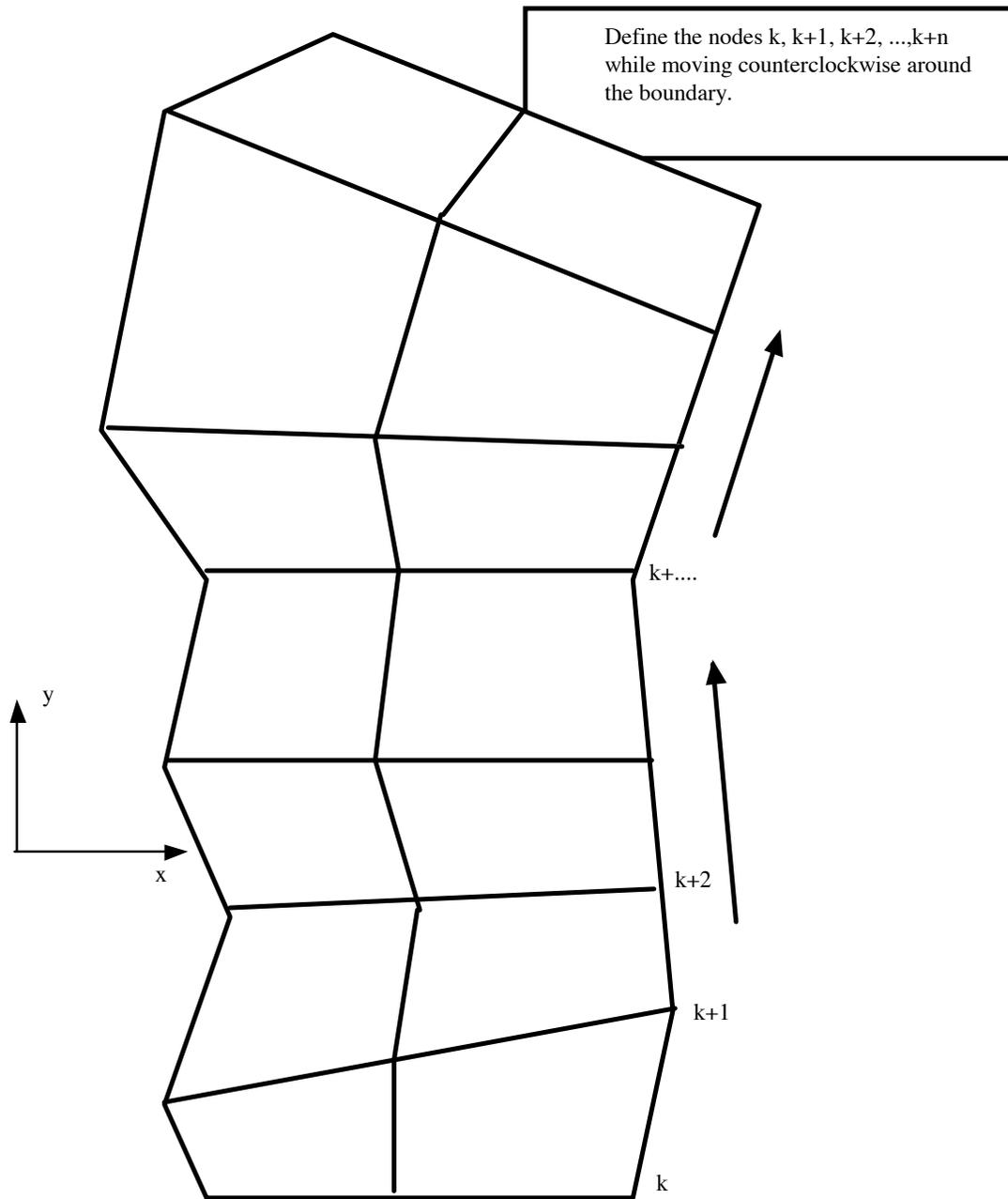


Figure 3.7. When defining a transmitting boundary in 2D define the node numbers in the node set in consecutive order while moving counterclockwise around the boundary.

***BOUNDARY** ***BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID**

***BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID**

Purpose: Prescribe the motion of a rigid body based on experimental data obtained from accelerometers affixed to the rigid body.

Note: This feature is available only in release R3 and higher of Version 971.

Card 1 1 2 3 4 5 6 7 8

Variable	PID								
Type	I								
Default	none								

Card 2, 3, 4, etc. Define one card for each accelerometer affixed to the rigid body. Input is terminated when a “*” card is found. A minimum of three accelerometers are required (see Remarks below).

Card 1 2 3 4 5 6 7 8

Variable	NID	CID	LCIDX	LCIDY	LCIDZ				
Type	I	I	I	I	I				
Default	none	none	none	none	none				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID for rigid body whose motion is prescribed.
NID	Node ID corresponding to the location of the accelerometer.
CID	Coordinate system ID describing the orientation of the accelerometer’s local axes (see *DEFINE_COORDINATE).
LCIDX	Load curve ID containing the local x-acceleration time history from the accelerometer.
LCIDY	Load curve ID containing the local y-acceleration time history from the accelerometer.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCIDZ	Load curve ID containing the local z-acceleration time history from the accelerometer.

Remarks:

1. Acceleration time histories from a minimum of three accelerometers each providing output from three channels are required.
2. Local axes of the accelerometers must be orthogonal.

*BOUNDARY

*BOUNDARY_PRESCRIBED_MOTION

VARIABLE	DESCRIPTION
	EQ.10: z/x degrees-of-freedom for node rotating about the y-axis at location (OFFSET1,OFFSET2) in the zx-plane, <i>point</i> (z,x). Radial motion is NOT permitted. Not applicable to rigid bodies. EQ.-10: z/x degrees-of-freedom for node rotating about the y-axis at location (OFFSET1,OFFSET2) in the zx-plane, <i>point</i> (z,x). Radial motion is permitted. Not applicable to rigid bodies. EQ.11: x/y degrees-of-freedom for node rotating about the z-axis at location (OFFSET1,OFFSET2) in the xy-plane, <i>point</i> (x,y). Radial motion is NOT permitted. Not applicable to rigid bodies. EQ.-11: x/y degrees-of-freedom for node rotating about the z-axis at location (OFFSET1,OFFSET2) in the xy-plane, <i>point</i> (x,y). Radial motion is permitted. Not applicable to rigid bodies.
VAD	Velocity/Acceleration/Displacement flag: EQ.0: velocity (rigid bodies and nodes), EQ.1: acceleration (rigid bodies and nodes), EQ.2: displacement (rigid bodies and nodes). EQ.3: velocity versus displacement (rigid bodies and nodes) EQ.4: relative displacement (rigid bodies only)
LCID	Load curve ID to describe motion value versus time, see *DEFINE_CURVE.
SF	Load curve scale factor. (default=1.0)
VID	Vector ID for DOF values of 4 or 8, see *DEFINE_VECTOR.
DEATH	Time imposed motion/constraint is removed: EQ.0.0: default set to 10 ²⁸
BIRTH	Time imposed motion/constraint is activated starting from the curve's initial abscissa value.
OFFSET1	Offset for DOF types 9-11 (y, z, x direction)
OFFSET2	Offset for DOF types 9-11 (z, x, y direction)
MRB	Master rigid body for measuring the relative displacement.
NODE1	Optional orientation node, n1, for relative displacement
NODE2	Optional orientation node, n2, for relative displacement

Remarks:

When DOF=5, 6, 7, or 8, nodal rotational degrees-of-freedom are prescribed in the case of deformable nodes (*OPTION1*=NODE or SET) whereas body rotations are prescribed in the case of a rigid body (*OPTION1*=RIGID). In the case of a rigid body, the axis of prescribed

rotation always passes through the body's center of mass. For $|\text{IDOF}|=8$, the axis of the prescribed rotation is parallel to vector **VID**. To prescribe a body rotation of a set of deformable nodes, with the axis of rotation parallel to global axes *x*, *y*, or *z*, use *OPTION1=SET* with $|\text{IDOF}| = 9, 10, \text{ or } 11$, respectively. The load curve scale factor can be used for simple modifications or unit adjustments.

The relative displacement can be measured in either of two ways:

1. Along a straight line between the mass centers of the rigid bodies,
2. Along a vector beginning at node *n1* and terminating at node *n2*.

With option 1, a positive displacement will move the rigid bodies further apart, and, likewise a negative motion will move the rigid bodies closer together. The mass centers of the rigid bodies must not be coincident when this option is used. With option 2 the relative displacement is measured along the vector, and the rigid bodies may be coincident. Note that the motion of the master rigid body is not directly affected by this option, i.e., no forces are generated on the master rigid body.

The activation time, **BIRTH**, is the time during the solution that the constraint begins to act. Until this time, the prescribed motion card is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and **BIRTH**, i.e., (solution time-**BIRTH**). Relative displacements that occur prior to reaching **BIRTH** are ignored. Only relative displacements that occur after **BIRTH** are prescribed.

When the constrained node is on a rigid body, the translational motion is imposed without altering the angular velocity of the rigid body by calculating the appropriate translational velocity for the center of mass of the rigid body using the equation:

$$v_{cm} = v_{node} - \omega \times (x_{cm} - x_{node})$$

where v_{cm} is the velocity of the center of mass, v_{node} is the specified nodal velocity, ω is the angular velocity of the rigid body, x_{cm} is the current coordinate of the mass center, and x_{node} is the current coordinate of the nodal point. Extreme care must be used when prescribing motion of a rigid body node. Typically, for nodes on a given rigid body, the motion of no more than one node should be prescribed or unexpected results may be obtained.

When the **RIGID** option is used to prescribe rotation of a rigid body, the axis of rotation will always be shifted such that it passes through the center-of-mass of the rigid body. By using ***PART_INERTIA** or ***CONSTRAINED_NODAL_RIGID_BODY_INERTIA**, one can override the internally-calculated location of the center-of-mass.

When the **RIGID_LOCAL** option is invoked, the orientation of the local coordinate system rotates with time in accordance with rotation of the rigid body.

Angular displacements are applied in an incremental fashion hence it is not possible to correctly prescribe a successive set of rotations about multiple axes. In light of this the command ***BOUNDARY_PRESCRIBED_ORIENTATION_RIGID** should be used for the purpose of prescribing the general orientation of a rigid body.

***BOUNDARY_PRESCRIBED_ORIENTATION_RIGID_OPTION**

Available options include:

DIRCOS

ANGLES

EULERP

Purpose: Prescribe the orientation of rigid body as a function of time.

Note: This feature is available only in release R3 and higher of Version 971.

Card Formats:

Card 1 is common to all orientation methods.
Cards 2 to 3 are unique for each orientation method.

Card 1 - Required for all orientation methods.

Card 1 1 2 3 4 5 6 7 8

Variable	PIDB	PIDA	INTRP						
Type	I	I	I						
Default	none	0	1						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PIDB	Part ID for rigid body B whose orientation is prescribed. See Remark 1.
PIDA	Part ID for rigid body A. If zero then orientation of PIDB is performed with respect to the global reference frame. See Remark 1.
INTRP	Interpolation method used on time history curves: EQ.1: linear interpolation (default) EQ.2: cubic spline interpolation

*BOUNDARY

*BOUNDARY_PRESCRIBED_ORIENTATION_RIGID

Card 2 of 3 - Required for DIRCOS option.

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDC11	LCIDC12	LCIDC13	LCIDC21	LCIDC22	LCIDC23	LCIDC31	LCIDC32
Type	I	I	I	I	I	I	I	I
Default	none							

Card 3 1 2 3 4 5 6 7 8

Variable	LCIDC33							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

LCIDCij

Load curve ID specifying direction cosine C_{ij} as a function of time. C_{ij} is defined as:

$$C_{ij} \triangleq \mathbf{a}_i \cdot \mathbf{b}_j$$

where \mathbf{a}_i ($i=1,2,3$) are mutually perpendicular unit vectors fixed in PIDA and \mathbf{b}_j ($j=1,2,3$) are mutually perpendicular unit vectors fixed in PIDB. If PIDA=0 then \mathbf{a}_j ($j=1,2,3$) are unit vectors aligned, respectively, with the global axes X, Y, and Z. See Remark 2.

Card 2 - Required for ANGLES option.

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDQ1	LCIDQ2	LCIDQ3	ISEQ	ISHFT			
Type	I	I	I	I	I			
Default	none	none	none	none	1			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
-----------------	--------------------

LCIDQi	Load curve ID specifying the orientation angle q_i as a function of time.
--------	---

ISEQ	<p>Specifies the sequence in which the rotations are effected. In this first set of sequences three unique axes are involved.</p> <p>EQ.123: the first rotation is performed about the x axis an amount q_1, the second about the y axis an amount q_2 and the third about the z axis an amount q_3.</p> <p>EQ.231: the first rotation is performed about the y axis an amount q_1, the second about the z axis an amount q_2 and the third about the x axis an amount q_3.</p> <p>EQ.312: the first rotation is performed about the z axis an amount q_1, the second about the x axis an amount q_2 and the third about the y axis an amount q_3.</p> <p>EQ.132: the first rotation is performed about the x axis an amount q_1, the second about the z axis an amount q_2 and the third about the y axis an amount q_3.</p> <p>EQ.213: the first rotation is performed about the y axis an amount q_1, the second about the x axis an amount q_2 and the third about the z axis an amount q_3.</p> <p>EQ.321: the first rotation is performed about the z axis an amount q_1, the second about the y axis an amount q_2 and the third about the x axis an amount q_3.</p>
------	---

The second set of sequences involve only two unique axes where the first and third are repeated.

EQ.121: the first rotation is performed about the x axis an amount q_1 , the second about the y axis an amount q_2 and the third about the x axis an amount q_3 .

EQ.131: the first rotation is performed about the x axis an amount q_1 , the second about the z axis an amount q_2 and the third about the x axis an amount q_3 .

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>EQ.212: the first rotation is performed about the y axis an amount q_1, the second about the x axis an amount q_2 and the third about the y axis an amount q_3.</p> <p>EQ.232: the first rotation is performed about the y axis an amount q_1, the second about the z axis an amount q_2 and the third about the y axis an amount q_3.</p> <p>EQ.313: the first rotation is performed about the z axis an amount q_1, the second about the x axis an amount q_2 and the third about the z axis an amount q_3.</p> <p>EQ.323: the first rotation is performed about the z axis an amount q_1, the second about the x axis an amount q_2 and the third about the z axis an amount q_3.</p>
ISHFT	<p>Angle shift.</p> <p>EQ.1: Angle curves are unaltered.</p> <p>EQ.2: Shifts angle data in the LCIDQi curves as necessary to eliminate discontinuities. If angles are confined to the range $[-\pi, \pi]$ and the data contains excursions exceeding π then set ISHFT=2.</p>

Remarks:

1. For the ANGLES Option note the following: If PIDA=0 then the successive rotations are performed about the global axes. If PIDA=PIDB then the rotations are performed about local axes fixed in PIDB. Angles are specified in radians.
2. For the DIRCOS Option the load curves LCIDCij must contain the same number of points with corresponding time values on the abscissa.

Card 2 - Required for EULERP option.

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDE1	LCIDE2	LCIDE3	LCIDE4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE**DESCRIPTION**

LCIDEi

Load curve ID specifying Euler parameter e_i as a function of time. The Euler parameters are defined as follows.

$$\varepsilon_i \triangleq \boldsymbol{\varepsilon} \cdot \mathbf{a}_i = \boldsymbol{\varepsilon} \cdot \mathbf{b}_i \quad (i = 1, 2, 3)$$

$$\varepsilon_4 \triangleq \cos\left(\frac{\theta}{2}\right)$$

where $\boldsymbol{\varepsilon}$ is the Euler vector, \mathbf{a}_i and \mathbf{b}_i ($i=1,2,3$) are dextral sets of unit vectors fixed in PIDA and PIDB, respectively, and θ is the angle associated with the rotation of PIDB in PIDA about Euler vector. If PIDA=0 then \mathbf{a}_j ($j=1,2,3$) are unit vectors aligned, respectively, with the global axes X, Y, and Z.

***BOUNDARY**

***BOUNDARY_PRESSURE_OUTFLOW**

***BOUNDARY_PRESSURE_OUTFLOW_OPTION**

Available options include:

SEGMENT

SET

Purpose: Define pressure outflow boundary conditions. These boundary conditions are attached to solid elements using the Eulerian ambient formulation (refer to ELFORM in *SECTION_SOLID_ALE) and defined to be pressure outflow ambient elements (refer to AET in *SECTION_SOLID_ALE).

For the SET option define the following card

Card 1 1 2 3 4 5 6 7 8

Variable	SSID								
Type	I								
Default	none								

For the SEGMENT option define the following card

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	N3	N4					
Type	I	I	I	I					
Default	none	none	none	none					

VARIABLE

DESCRIPTION

SSID

Segment set ID

N1,N2...

Node ID's defining segment

***BOUNDARY_RADIATION_OPTION1_{OPTION2}**

OPTION1 specifies radiation boundary surface definition by a surface set (**SET**) or by a segment list (**SEGMENT**).

OPTION2 indicates the radiation boundary surface is part of an enclosure. *OPTION2* specifies the use of view factors (**VF**) or exchange factors (**EF**) for the radiation calculations. The suffix (**READ**) indicates that the view factors should be read from the file “**viewfl**” or exchange factors from the file “**exchfl**”. The suffix (**CALCULATE**) indicates that the view factors or exchange factors should be calculated. The Stefan Boltzmann constant must be defined for radiation in an enclosure. See ***CONTROL_THERMAL_SOLVER**.

A list of acceptable keywords are:

***BOUNDARY_RADIATION_SEGMENT**

***BOUNDARY_RADIATION_SEGMENT_VF_READ**

***BOUNDARY_RADIATION_SEGMENT_VF_CALCULATE**

***BOUNDARY_RADIATION_SET**

***BOUNDARY_RADIATION_SET_VF_READ**

***BOUNDARY_RADIATION_SET_VF_CALCULATE**

***BOUNDARY_RADIATION_SET_EF_READ**

***BOUNDARY_RADIATION_SET_EF_CALCULATE**

*BOUNDARY

*BOUNDARY_RADIATION

*BOUNDARY_RADIATION_SEGMENT

Include the following 2 cards for each segment. This keyword defines surface segments that transfer energy by radiation to the environment. Setting TYPE=1 on Card 1 below indicates that the segment transfers energy to the environment.

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	N3	N4	TYPE			
Type	I	I	I	I	I			
Default	none	none	none	none	1			

(Card 2 of 2)

Card 2 1 2 3 4 5 6 7 8

Variable	FLCID	FMULT	TILCID	TIMULT	LOC			
Type	I	F	I	F	I			
Default	none	0.	none	0.	0			

BOUNDARY_RADIATION**BOUNDARY*****BOUNDARY_RADIATION_SEGMENT_VF_READ*****BOUNDARY_RADIATION_SEGMENT_VF_CALCULATE**

Include the following 2 cards for each segment. This keyword defines surface segments that transfer energy by radiation within an enclosure using view factors. Setting TYPE=2 on Card 1 below specifies that the segment belongs to an enclosure. The file “viewfl” must be present for the **READ** option. The file “viewfl” will be created for the **CALCULATE** option. If the file “viewfl” exists when using the **CALCULATE** option, LS-DYNA will terminate with an error message to prevent overwriting the file. The format of “viewfl” is defined at the end of this section.

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	N3	N4	TYPE	BLOCK	NINT	
Type	I	I	I	I	I	I	I	
Default	none	none	none	none	2	0	0	

(Card 2 of 2)

Card 2 1 2 3 4 5 6 7 8

Variable	SELCID	SEMULT						
Type	I	F						
Default	none	0.						

*BOUNDARY

*BOUNDARY_RADIATION

*BOUNDARY_RADIATION_SET

Include the following 2 cards for each set. This keyword defines surface segment sets that transfer energy by radiation to the environment. Setting TYPE=1 on Card 1 below indicates that the segment transfers energy to the environment.

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	SSID	TYPE						
Type	I	I						
Default	none	1						

(Card 2 of 2)

Card 2 1 2 3 4 5 6 7 8

Variable	RFLCID	RFMULT	TILCID	TIMULT	LOC			
Type	I	F	I	F	I			
Default	none	0.	none	0.	0			

BOUNDARY_RADIATION**BOUNDARY*****BOUNDARY_RADIATION_SET_VF_READ*****BOUNDARY_RADIATION_SET_VF_CALCULATE**

Include the following 2 cards for each set. This keyword defines surface segment sets that transfer energy by radiation within an enclosure using view factors. Setting TYPE=2 on Card 1 below specifies that the segment set belongs to an enclosure. The file “viewfl” must be present for the **READ** option. The file “viewfl” will be created for the **CALCULATE** option. If the file “viewfl” exists when using the **CALCULATE** option, LS-DYNA will terminate with an error message to prevent overwriting the file. The format of “viewfl” is defined at the end of this section.

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	SSID	TYPE	RAD_GRP	FILE_NO	BLOCK	NINT		
Type	I	I	I	I	I	I		
Default	none	2	0	0	0	0		

(Card 2 of 2)

Card 2 1 2 3 4 5 6 7 8

Variable	SELCID	SEMULT						
Type	I	F						
Default	none	0.						

*BOUNDARY

*BOUNDARY_RADIATION

*BOUNDARY_RADIATION_SET_EF_READ

*BOUNDARY_RADIATION_SET_EF_CALCULATE

Include the following card for each set. This keyword defines surface segment sets that transfer energy by radiation within an enclosure using exchange factors. The file “exchfl” must be present for the **READ** option. The file “exchfl” will be created for the **CALCULATE** option. If the file “exchfl” exists when using the **CALCULATE** option, LS-DYNA will terminate with an error message to prevent overwriting the file. The format of “exchfl” is defined at the end of this section.

Card 1 1 2 3 4 5 6 7 8

Variable	SSID	NMAT	NPHT	ERRMAX				
Type	I	I	I	I				
Default	none	none	1	1.0e-02				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
BLOCK	Flag indicating if this surface blocks the view between any other 2 surfaces. EQ.0: no blocking (default) EQ.1: blocking
ERRMAX	ERRMAX is the convergence error tolerance for the surface.
FILE_NO	File number for view factor file. FILE_NO is added to viewfl_ to form the name of the file containing the view factors. For example if FILE_NO is specified as 22, then the view factors are read from viewfl_22 . For radiation enclosure group zero FILE_NO is ignored and view factors are read from viewfl . The same file may be used for different radiation enclosure group definitions.
LOC	Shell surface flag for thermal shell elements. See the parameter TSHELL on the *CONTROL_SHELL keyword. EQ.-1: lower surface of thermal shell element EQ. 1: upper surface of thermal shell element
N1, N2, N3, N4	Node ID's defining segment

VARIABLE	DESCRIPTION
NINT	Number of integration points for viewfactor calculation EQ.0: LS-DYNA determines the number of integration points based on the segment size and separation distance $1 \leq \text{NINT} \leq 10$: User specified number
NMAT	NMAT specifies the material type for the portion of the boundary specified by SSID. NMAT must be an exchange factor material ID. See the *EF_MATERIAL keyword.
NPHT	The segments specified by SSID will emit NPHT*NPHOTON photons. See the *EF_CONTROL keyword.
RAD_GRP	Radiation enclosure group ID. The segment sets from all radiation enclosure definitions with the same group ID are augmented to form a single enclosure definition. If RAD_GRP is not specified or set to zero, then the segments are placed in group zero. All segments defined by the SEGMENT option are placed in set zero.
FLCID	Load curve ID for radiation factor f , see *DEFINE_CURVE GT.0: function versus time EQ.0: use constant multiplier value, FMULT LT.0: function versus temperature
FMULT	Curve multiplier for f for use in the equation $\dot{q}'' = \sigma \epsilon F (T_2^4 - T_1^4) = f (T_{surface}^4 - T_{\infty}^4)$
SELCID	Load curve ID for surface emissivity, see *DEFINE_CURVE GT.0: function versus time EQ.0: use constant multiplier value, SEMULT LT.0: function versus temperature
SEMULT	Curve multiplier for surface emissivity, see *DEFINE_CURVE
SSID	SSID specifies the ID for a set of segments that comprise a portion of, or possibly, the entire enclosure. See *SET_SEGMENT.
TILCID	Load curve ID for T_{∞} versus time, see *DEFINE_CURVE: EQ.0: use constant multiplier, TIMULT
TIMULT	Curve multiplier for T_{∞}
TYPE	Radiation type: EQ.1: Radiation to environment EQ.2: Radiation within an enclosure

Remarks:

The exchange factor, F , is a characterization of the effect of the system geometry, emissivity and reflectivity on the capability of radiative transport between surfaces. The radiation boundary condition data cards require specification of the product, $f = F\sigma$, and T_∞ for the boundary surface.

1. When using view factors, the file “viewfl” must exist when using the READ option. This file contains the surface-to-surface area*view factor products (i.e., $A_i F_{ij}$). The $A_i F_{ij}$ products must be stored in this file by row and formatted as 5E16.0.
2. When using exchange factors, the file “exchfl” must exist when using the READ option. This file contains the surface-to-surface emittance*exchange fraction products (i.e., $\epsilon_i F_{ij}$). The $\epsilon_i F_{ij}$ products must be stored in this file by row and formatted as 5E16.0.
3. Multiple enclosures can be modeled when using view factors. Consider the following example input. The order of segments in the view factor file follows the order the sets are assigned to the boundary radiation definition.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *BOUNDARY_RADIATION_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Make boundary enclosure radiation groups 8 and 9.
$
*BOUNDARY_RADIATION_SET_VF_READ
*   SSID      TYPE     RAD_GRP   FILE_NO
      15        2         9         10
      1.0      1.0
*BOUNDARY_RADIATION_SET_VF_READ
*   SSID      TYPE     RAD_GRP   FILE_NO
      12        2         9         10
      1.0      1.0
*BOUNDARY_RADIATION_SET_VF_READ
*   SSID      TYPE     RAD_GRP   FILE_NO
      13        2         8         21
      1.0      1.0
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

Enclosure radiation group 9 is composed of all the segments in segment set 15 followed by those in segment set 12. The view factors are stored in the file viewfl_10. Enclosure radiation group 8 is composed of the segments in segment set 13. The view factors are stored in the file viewfl_21.

***BOUNDARY_SLIDING_PLANE**

Purpose: Define a sliding symmetry plane. This option applies to continuum domains modeled with solid elements.

Card 1 2 3 4 5 6 7 8

Variable	NSID	VX	VY	VZ	COPT			
Type	I	F	F	F	I			
Default	none	0	0	0	0			

VARIABLE**DESCRIPTION**

NSID	Nodal set ID, see *SET_NODE
VX	x-component of vector defining normal or vector
VY	y-component of vector defining normal or vector
VZ	z-component of vector defining normal or vector
COPT	Option: EQ.0: node moves on normal plane, EQ.1: node moves only in vector direction.

Remarks:

Any node may be constrained to move on an arbitrarily oriented plane or line depending on the choice of COPT. Each boundary condition card defines a vector originating at (0,0,0) and terminating at the coordinates defined above. Since an arbitrary magnitude is assumed for this vector, the specified coordinates are non-unique and define only a direction. Use of *BOUNDARY_SPC is preferred over *BOUNDARY_SLIDING_PLANE as the boundary conditions imposed via the latter have been seen to break down somewhat in lengthy simulations owing to numerical roundoff.

VARIABLE	DESCRIPTION
ID	Optional SPC set ID to which this node or node set belongs. This ID does not need to be unique
HEADING	An optional SPC descriptor that will be written into the D3HSP file and the SPCFORC file.
NID/NSID	Node ID or nodal set ID, see *SET_NODE.
CID	Coordinate system ID, see *DEFINE_COORDINATE_SYSTEM.
DOFX	Insert 1 for translational constraint in local x-direction.
DOFY	Insert 1 for translational constraint in local y-direction.
DOFZ	Insert 1 for translational constraint in local z-direction.
DOFRX	Insert 1 for rotational constraint about local x-axis.
DOFRY	Insert 1 for rotational constraint about local y-axis.
DOFRZ	Insert 1 for rotational constraint about local z-axis.

Remarks:

Constraints are applied if a value of 1 is given for DOFxx. A value of zero means no constraint. No attempt should be made to apply SPCs to nodes belonging to rigid bodies (see *MAT_RIGID for application of rigid body constraints).

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *BOUNDARY_SPC_NODE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Make boundary constraints for nodes 6 and 542.
$
*BOUNDARY_SPC_NODE
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$ nid cid dofx dofy dofz dofrx dofry dofrz
$ 6 0 1 1 1 1 1 1
$ 542 0 0 1 0 1 0 1
$
$ Node 6 is fixed in all six degrees of freedom (no motion allowed).
$
$ Node 542 has a symmetry condition constraint in the x-z plane,
$ no motion allowed for y translation, and x & z rotation.
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

*BOUNDARY

*BOUNDARY_SPH_FLOW

*BOUNDARY_SPH_FLOW

Purpose: Define a flow of particles. This option applies to continuum domains modeled with SPH elements.

Card 1 Format

Card 1 1 2 3 4 5 6 7 8

Variable	ID	STYP	DOF	VAD	LCID□	SF	DEATH	BIRTH
Type	I	I	I	I	I	F	F	F
Default	none	none	None	0	none	1.	1.E+20	0.0

Card 2 Format

Card 2 1 2 3 4 5 6 7 8

Variable	NODE	VID						
Type	I	I						
Default	none	0						

VARIABLE

DESCRIPTION

NSID, PID	Nodal set ID (NSID), SEE *SET_NODE, or part ID (PID), see *PART.
STYP	Set type: EQ.1: part set ID, see *SET_PART, EQ.2: part ID, see *PART, EQ.3: node set ID, see *NODE_SET,
DOF	Applicable degrees-of-freedom: EQ.1: x-translational degree-of-freedom, EQ.2: y-translational degree-of-freedom, EQ.3: z-translational degree-of-freedom, EQ.4: translational motion in direction given by the VID. Movement on plane normal to the vector is permitted.

VARIABLE	DESCRIPTION
VAD	Velocity/Acceleration/Displacement flag applied to SPH elements before activation: EQ.0: velocity, EQ.1: acceleration, EQ.2: displacement.
LCID	Load curve ID to describe motion value versus time, see *DEFINE_CURVE.
SF	Load curve scale factor. (default=1.0)
DEATH	Time imposed motion/constraint is removed: EQ.0.0: default set to 10^{20} .
BIRTH	Time imposed motion/constraint is activated.
NODE	Node fixed in space which determines the boundary between activated particles and deactivated particles.
VID	Vector ID for DOF value of 4, see *DEFINE_VECTOR

Remarks:

Initially, the user defines the set of particles that are representing the flow of particles during the simulation. At time $t=0$, all the particles are deactivated which means that no particle approximation is calculated. The boundary of activation is a plane determined by the NODE, and normal to the vector VID. The particles are activated when they reached the boundary. Since they are activated, particle approximation is started.

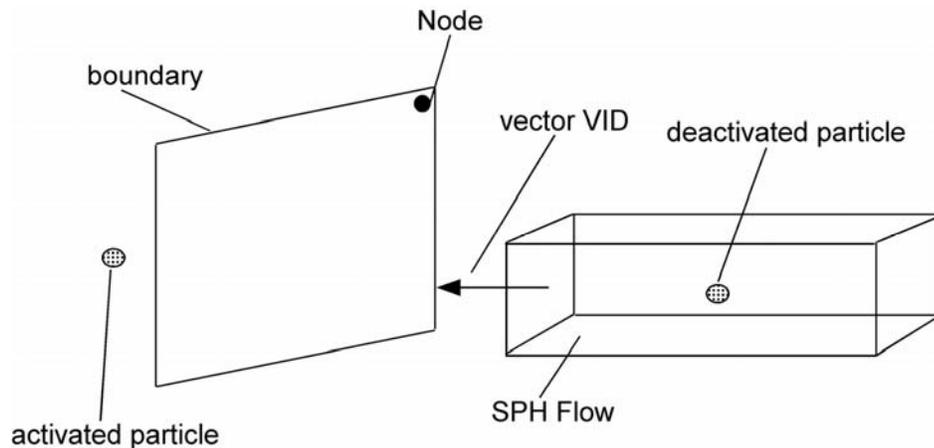


Figure 3.9. Vector **VID** determines the orientation of the SPH flow.

*BOUNDARY

*BOUNDARY_SPH_SYMMETRY_PLANE

*BOUNDARY_SPH_SYMMETRY_PLANE

Purpose: Define a symmetry plane for SPH. This option applies to continuum domains modeled with SPH elements.

Card 1 2 3 4 5 6 7 8

Variable	VTX	VTY	VTZ	VHX	VHY	VHZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE

DESCRIPTION

VTX	x-coordinate of tail of a normal vector originating on the wall (tail) and terminating in the body (head) (i.e., vector points from the symmetry plane into the body).
VTY	y-coordinate of tail
VTZ	z-coordinate of tail
VHX	x-coordinate of head
VHY	y-coordinate of head
VHZ	z-coordinate of head

Remarks:

1. A plane of symmetry is assumed for all SPH elements defined in the model.
2. The plane of symmetry has to be normal to either the x, y or z direction.

***BOUNDARY_SYMMETRY_FAILURE**

Purpose: Define a symmetry plane with a failure criterion. This option applies to continuum domains modeled with solid elements.

Card 1 2 3 4 5 6 7 8

Variable	SSID	FS	VTX	VTY	VTZ	VHX	VHY	VHZ
Type	I	F	F	F	F	F	F	F
Default	none	0.	0.	0.	0.	0.	0.	0.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SSID	Segment set ID, see *SET_SEGMENT
FS	Tensile failure stress > 0.0. The average stress in the elements surrounding the boundary nodes in a direction perpendicular to the boundary is used.
VTX	x-coordinate of tail of a normal vector originating on the wall (tail) and terminating in the body (head) (i.e., vector points from the symmetry plane into the body).
VTY	y-coordinate of tail
VTZ	z-coordinate of tail
VHX	x-coordinate of head
VHY	y-coordinate of head
VHZ	z-coordinate of head

Remarks:

A plane of symmetry is assumed for the nodes on the boundary at the tail of the vector given above. Only the motion perpendicular to the symmetry plane is constrained. After failure the nodes are set free.

*BOUNDARY

*BOUNDARY_TEMPERATURE

*BOUNDARY_TEMPERATURE_OPTION

Available options include:

NODE

SET

Purpose: Define temperature boundary conditions for a thermal or coupled thermal/structural analysis.

Card	1	2	3	4	5	6	7	8
Variable	NID/SID	LCID	CMULT	LOC				
Type	I	I	F	I				
Default	none	0	0.	0				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NID/SID	Node ID/Node Set ID, see *SET_NODE_OPTION
LCID	Load curve ID for temperature versus time: EQ.0: use the constant multiplier value given below by CMULT.
CMULT	Curve multiplier for temperature
LOC	Application of surface for thermal shell elements, see parameter, TSHELL, in the *CONTROL_SHELL input: EQ.-1: lower surface of thermal shell element EQ.0: middle surface of thermal shell element EQ.1: upper surface of thermal shell element

Remarks:

If no load curve ID is given, then a constant boundary temperature is assumed. CMULT is also used to scale the load curve values.

***BOUNDARY_THERMAL_WELD**

Purpose: Define a moving heat source to model welding. Only applicable for a coupled thermal-structural simulations in which the weld source or work piece is moving.

Card 1 Format

Card 1 1 2 3 4 5 6 7 8

Variable	PID	PTYP	NID	NFLAG	X0	Y0	Z0	N2ID
Type	I	I	I	I	F	F	F	I
Default	none	1	none	1	none	none	none	none

Card 2 Format

Card 2 1 2 3 4 5 6 7 8

Variable	a	b	c_f	c_r	LCID	Q	F_f	F_r
Type	F	F	F	F	I	F	F	F
Default	none	none	none	none	none	none	none	none

Optional Card 3 Format (define this card only if N2ID = -1 on card 1 above)

Card 3 1 2 3 4 5 6 7 8

Variable	tx	ty	tz					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part ID or Part Set ID to which weld source is applied
PTYP	PID type: EQ.1: PID defines a single part ID EQ.2: PID defines a part set ID
NID	Node ID giving location of weld source EQ.0: location defined by (X0,Y0,Z0) below
NFLAG	Flag controlling motion of weld source EQ.1: source moves with node NID EQ.2: source is fixed in space at original position of node NID
X0,Y0,Z0	Coordinates of weld source, which remains fixed in space (optional, ignored if NID nonzero above)
N2ID	Second node ID for weld beam aiming direction GT.0: beam is aimed from N2ID to NID, moves with these nodes EQ.-1: beam aiming direction is (tx,ty,tz) input on optional card 3
a	Weld pool width
b	Weld pool depth (in beam aiming direction)
c _f	Weld pool forward direction
c _r	Weld pool rearward direction
LCID	Load curve ID for weld energy input rate vs. time EQ.0: use constant multiplier value Q.
Q	Curve multiplier for weld energy input rate [energy/time, e.g., Watt]
F _f	Forward distribution function
F _r	Rear distribution function (Note: F _f + F _r = 2.0)
tx,ty,tz	Weld beam direction vector in global coordinates (N2ID = -1 only)

Remarks:

This boundary condition allows simulation of a moving weld heat source, following the work of Goldak, Chakravarti, and Bibby [1984]. Heat is generated in an ellipsoidal region centered at the weld source, and decaying exponentially with distance according to:

$$q = \frac{6\sqrt{3}FQ}{\pi\sqrt{\pi abc}} e^{\left(\frac{-3x^2}{a^2}\right)} e^{\left(\frac{-3y^2}{b^2}\right)} e^{\left(\frac{-3z^2}{c^2}\right)}$$

where:

q = weld source power density

(x, y, z) = coordinates of point p in weld material

$$F = \begin{cases} F_f & \text{if point } p \text{ is in front of beam} \\ F_r & \text{if point } p \text{ is behind beam} \end{cases}$$

$$c = \begin{cases} c_f & \text{if point } p \text{ is in front of beam} \\ c_r & \text{if point } p \text{ is behind beam} \end{cases}$$

A local coordinate system is constructed which is centered at the heat source. The relative velocity vector of the heat source defines the "forward" direction, so material points that are approaching the heat source are in "front" of the beam. The beam aiming direction is used to compute the weld pool depth. The weld pool width is measured normal to the relative velocity - aiming direction plane.

*BOUNDARY

*BOUNDARY_USA_SURFACE

*BOUNDARY_USA_SURFACE

Purpose: Define a surface for coupling with the USA boundary element code [DeRuntz 1993]. The outward normal vectors should point into the fluid media.

Card 1 2 3 4 5 6 7 8

Variable	SSID	WETDRY	NBEAM					
Type	I	I	I					
Default	none	0	0					

VARIABLE

DESCRIPTION

SSID Segment set ID, see *SET_SEGMENT

WETDRY Wet surface flag:
EQ.0: dry, no coupling,
EQ.1: wet, coupled with USA.

NBEAM The number of nodes touched by USA Surface-of-Revolution (SOR) elements. It is not necessary that the LS-DYNA model has beams where USA has beams (i.e., SOR elements), merely that the LS-DYNA model has nodes to receive the forces that USA will return.

Remarks:

The wet surface of 3 and 4-noded USA General boundary elements is defined in LS-DYNA with a segment set of 4-noded surface segments, where the fourth node can duplicate the third node to form a triangle. The segment normals should be directed into the USA fluid. If USA overlays are going to be used to reduce the size of the DAA matrices, the user should nonetheless define the wet surface here as if no overlay were being used. If Surface-of - Revolution elements (SORs) are being used in USA, then NBEAM should be non-zero on one and only one card in this section.

When running a coupled problem with USA, the procedure involves several steps. First, LS-DYNA is executed to create a LS-DYNA dump file "d3dump" and a linking file "strnam" which contains the nodal grid point data and wet segment connectivity data for the FLUMAS processor, and the dof-equation table and structural mass vector for the AUGMAT processor. "Dyna.pre" is denoted "grdnam" in the FLUMAS manual and "strnam" in the AUGMAT manual. The execution line in the first step is:

LS-DYNA memory=nwds i=inputfilename > outputfilename

where "inputfilename" is the LS-DYNA input file.

In the second step, the DAA fluid mass matrix is created through execution of the USA FLUMAS processor:

FLUMAS -m nwds < flumasinputfilename > flumasoutputfilename

In the third step, the modified augmented DAA equations for the coupled problem are calculated and saved through execution of the USA AUGMAT processor:

AUGMAT -m nwds < augmatinputfilename > augmatoutputfilename

This step is repeated whenever one wishes to change DAA formulations.

In the fourth step the actual coupled time-integration is conducted using the execution line:

LS-DYNA memory=nwds r=d3dump usa=usainputfilename > outputfilename

The input files, flumasinputfilename, augmatinputfilename, and usainputfilename, are prepared in accordance with the USA code documentation.

It is advisable when running coupled problems to check the ASCII output files to ensure that each run completed normally.

***BOUNDARY**

***BOUNDARY_USA_SURFACE**

*CASE

This keyword option provides a way of running multiple load cases sequentially. Within each case, the input parameters, which include loads, boundary conditions, control cards, contact definitions, initial conditions, etc., can change. If desired, the results from a previous case can be used during initialization. Each case creates unique file names for all output results files by appending “**CID n** .” to the default file name.

Note: To use the *CASE keyword requires a stand alone program that is started by the LS-DYNA executable. To obtain that program contact LSTC or your local distributor.

*CASE_{OPTION}

Available options include:

<BLANK>

BEGIN_CID n

END_CID n

Purpose: Begin a case definition. In options *CASE_BEGIN_ and *CASE_END_, CID n is a numeric identification number that may or may not be active for a particular case. All keywords between *CASE_BEGIN_CID n and *CASE_END_CID n will be included for the case if CID n is activated for the case. See the definition of the *CASE keyword below. *CASE_BEGIN/*CASE_END can be nested, overlapped, and disjointed. An example below demonstrates the use of these options. However convenient, the optional keywords, BEGIN_ and END_ are not necessary. All keywords cards that contain an ID that are active for a case are included. Any *non-case* keyword card can be tagged with an ID by adding the string “CID=CID n .” This keyword will then be active only for those cases having this CID in their active list. There can be more than one space before the CID= identifier. Any keyword without CID= is active for all cases.

Card 1	1	2	3	4	5	6	7	8
Variable	CASEID							
Type	I							
Default	None							

*CASE

Optional: Define if additional command line arguments are required for this case ID.

Cards 2, ... 1 2 3 4 5 6 7 8

Variable	COMMANDS
Type	A
Default	Not Required

Define as many Active IDs for this case. Use as many cards as necessary.

Cards ... 1 2 3 4 5 6 7 8

Variable	CID1	CID2	CID7	...
Type	I	I	I	I	I	I	I	I
Default	none							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CASEID	Identification number for CASE.
COMMANDS	Command line arguments.
CID n	Active numeric ID for case, CASEID. Any non-case keyword card can be tagged with a case ID by adding the string "CID=CID n . This keyword will then be active only for those cases having this CID in their active list. There can be more than one space before the CID= identifier. Any keyword without CID= is active for all cases.

Example:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ $ *CASE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*CASE
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7
$  CASEID
    101
MEMORY=20M
    1
$
*CASE
$  CASEID
    102
MEMORY=20M  NCYCLE=1845
$
*TITLE  CID=1
THIS IS THE TITLE FOR CASE 101
$
*TITLE  CID=5
THIS IS THE TITLE FOR CASE 102
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Example:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ $ *CASE_BEGIN_5
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*CASE_BEGIN_5
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>
*DATABASE_BINARY_D3THDT
1.e-5
*CASE_BEGIN_3
*DATABASE_NODOUT
1.e-5
*CASE_END_5
*DATABASE_ELOUT
1.e-5
*CASE_END_3
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

The last example will generate D3THDT and NODOUT for CID=5, and NODOUT and ELOUT for CID=3. If there are no CASE statements, but there are *CASE_BEGIN/*CASE_END statements, then one case is created for each unique ID on the *CASE_BEGIN lines. Since optional command line arguments may be required, *CASE_BEGIN can optionally be followed by extra command line arguments.

If *CASE statements are used, cases are not generated for each *CASE_BEGIN, but only for each *CASE. If no *CASE keywords appear, it is an error to have CID= on any keyword. If multiple *CASE or *CASE_BEGIN statements appear that have the same ID, their command line arguments and active id lists are merged.

***CASE**

***COMPONENT**

The keyword ***COMPONENT** provides a way of incorporating specialized components and features. The keyword control cards in this section are defined in alphabetical order:

***COMPONENT_GEBOD_OPTION**

***COMPONENT_GEBOD_JOINT_OPTION**

***COMPONENT_HYBRIDIII**

***COMPONENT_HYBRIDIII_JOINT_OPTION**

*COMPONENT

*COMPONENT_GEBOD

*COMPONENT_GEBOD_OPTION

Purpose: Generate a rigid body dummy based on dimensions and mass properties from the GEBOD database. The motion of the dummy is governed by equations integrated within LS-DYNA separately from the finite element model. Default joint characteristics (stiffnesses, stop angles, etc.) are set internally and should give reasonable results, however, they may be altered using the *COMPONENT_GEBOD_JOINT command. Contact between the segments of the dummy and the finite element model is defined using the *CONTACT_GEBOD command. The use of a positioning file is essential with this feature, see Appendix N for further details.

OPTION specifies the human subject type. The male and female types represent adults while the child is genderless.

MALE

FEMALE

CHILD

(Card 1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	DID	UNITS	SIZE					
Type	I	I	F					
Default	none	none	none					

VARIABLE

DESCRIPTION

DID Dummy ID. A unique number must be specified.

UNITS System of units used in the finite element model.
 EQ.1: lbf*sec²/in - inch - sec
 EQ.2: kg - meter - sec
 EQ.3: kgf*sec²/mm - mm - sec
 EQ.4: metric ton - mm - sec
 EQ.5: kg - mm - msec

SIZE Size of the dummy. This represents a combined height and weight percentile ranging from 0 to 100 for the male and female types. For the child the number of months of age is input with an admissible range from 24 to 240.

***COMPONENT**

***COMPONENT_GEBOD_JOINT**

***COMPONENT_GEBOD_JOINT_OPTION**

Purpose: Alter the joint characteristics of a GEBOD rigid body dummy. Setting a joint parameter value to zero retains the default value set internally. See Appendix N for further details.

The following options are available.

PELVIS

WAIST

LOWER_NECK

UPPER_NECK

LEFT_SHOULDER

RIGHT_SHOULDER

LEFT_ELBOW

RIGHT_ELBOW

LEFT_HIP

RIGHT_HIP

LEFT_KNEE

RIGHT_KNEE

LEFT_ANKLE

RIGHT_ANKLE

Card 1 - Required.

Card 1 1 2 3 4 5 6 7 8

Variable	DID	LC1	LC2	LC3	SCF1	SCF2	SCF3	
Type	F	I	I	I	F	F	F	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DID	Dummy ID, see *COMPONENT_GEBOD_OPTION.
LCi	Load curve ID specifying the loading torque versus rotation (in radians) for the i-th degree of freedom of the joint.
SCFi	Scale factor applied to the load curve of the i-th joint degree of freedom.

Card 2 - Required.

Card 2 1 2 3 4 5 6 7 8

Variable	C1	C2	C3	NEUT1	NEUT2	NEUT3		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
Ci	Linear viscous damping coefficient applied to the i-th DOF of the joint. Units are torque*time/radian, where the units of torque and time depend on the choice of UNITS in card 1 of *COMPONENT_GEBOD_OPTION.
NEUTi	Neutral angle (degrees) of joint's i-th DOF.

Card 3 - Required.

Card 3 1 2 3 4 5 6 7 8

Variable	LOSA1	HISA1	LOSA2	HISA2	LOSA3	HISA3		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LOSAi	Value of the low stop angle (degrees) for the i-th DOF of this joint.
HISAi	Value of the high stop angle (degrees) for the i-th DOF of this joint.

*COMPONENT

*COMPONENT_GEBOD_JOINT

Card 4 - Required.

Card 4 1 2 3 4 5 6 7 8

Variable	UNK1	UNK2	UNK3					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE

DESCRIPTION

UNKi

Unloading stiffness (torque/radian) for the i-th degree of freedom of the joint. This must be a positive number. Units of torque depend on the choice of UNITS in card 1 of *COMPONENT_GEBOD_OPTION.

*COMPONENT

*COMPONENT_HYBRIDIII

*COMPONENT_HYBRIDIII

Purpose: Define a HYBRID III dummy. The motion of the dummy is governed by equations integrated within LS-DYNA separately from the finite element model. The dummy interacts with the finite element structure through contact interfaces. Joint characteristics (stiffnesses, damping, friction, etc.) are set internally and should give reasonable results, however, they may be altered using the *COMPONENT_HYBRIDIII_JOINT command. Joint force and moments can be written to an ASCII file (see *DATABASE_H3OUT).

Card 1	1	2	3	4	5	6	7	8
Variable	DID	SIZE	UNITS	DEFRM	VX	VY	VZ	
Type	I	I	I	I	F	F	F	
Default	none	none	none	1	0.	0.	0.	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DID	Dummy ID. A unique number must be specified.
SIZE	Size of dummy. EQ.1: 5th percentile adult EQ.2: 50th percentile adult EQ.3: 95th percentile adult Note: If negative then the best of currently available joint properties are applied.
UNITS	System of units used in the finite element model. EQ.1: lbf*sec ² /in - inch - sec EQ.2: kg - meter - sec EQ.3: kgf*sec ² /mm - mm - sec EQ.4: metric ton - mm - sec EQ.5: kg - mm - msec
DEFRM	Deformability type. EQ.1: all dummy segments entirely rigid EQ.2: deformable abdomen (low density foam, mat #57) EQ.3: deformable jacket (low density foam, mat #57) EQ.4: deformable headskin (viscoelastic, mat #6) EQ.5: deformable abdomen/jacket EQ.6: deformable jacket/headskin EQ.7: deformable abdomen/headskin EQ.8: deformable abdomen/jacket/headskin
VX,VY,VZ	Initial velocity of the dummy in the global x, y and z directions.

*COMPONENT

*COMPONENT_HYBRIDIII_JOINT

*COMPONENT_HYBRIDIII_JOINT_OPTION

Purpose: Alter the joint characteristics of a HYBRID III dummy. Setting a joint parameter value to zero retains the default value set internally. Joint force and moments can be written to an ASCII file (see *DATABASE_H3OUT). Further details pertaining to the joints are found in the Hybrid III Dummies section of Appendix N.

The following options are available:

LUMBAR	RIGHT_ELBOW	RIGHT_KNEE
LOWER_NECK	LEFT_WRIST	LEFT_ANKLE
UPPER_NECK	RIGHT_WRIST	RIGHT_ANKLE
LEFT_SHOULDER	LEFT_HIP	STERNUM
RIGHT_SHOULDER	RIGHT_HIP	LEFT_KNEE_SLIDER
LEFT_ELBOW	LEFT_KNEE	RIGHT_KNEE_SLIDER

Card 1 - Required.

Card 1 1 2 3 4 5 6 7 8

Variable	DID	Q1	Q2	Q3	FRIC			
Type	F	F	F	F	F			

Card 2 - Required.

Card 2 1 2 3 4 5 6 7 8

Variable	C1	ALO1	BLO1	AHI1	BHI1	QLO1	QHI1	SCLK1
Type	F	F	F	F	F	F	F	F

Card 3 - Required. Left blank if joint has only one degree of freedom.

Card 3	1	2	3	4	5	6	7	8
Variable	C2	ALO2	BLO2	AHI2	BHI2	QLO2	QHI2	SCLK2
Type	F	F	F	F	F	F	F	F

Card 4 - Required. Left blank if the joint has only two degrees of freedom.

Card 4	1	2	3	4	5	6	7	8
Variable	C3	ALO3	BLO3	AHI3	BHI3	QLO3	QHI3	SCLK3
Type	F	F	F	F	F	F	F	F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DID	Dummy ID, see *COMPONENT_HYBRIDIII
Qi	Initial value of the joint's i-th degree of freedom. Units of degrees are defined for rotational DOF. See Appendix N for a listing of the applicable DOF.
FRIC	Friction load on the joint.
Ci	Linear viscous damping coefficient applied to the i-th DOF of the joint.
ALOi	Linear coefficient for the low regime spring of the joint's i-th DOF.
BLOi	Cubic coefficient for the low regime spring of the joint's i-th DOF.
AHli	Linear coefficient for the high regime spring of the joint's i-th DOF.
BHli	Cubic coefficient for the high regime spring of the joint's i-th DOF.
QLOi	Value at which the low regime spring definition becomes active.
QHli	Value at which the high regime spring definition becomes active.
SCLKi	Scale value applied to the stiffness of the joint's i-th DOF (default=1.0).

***CONSTRAINED**

The keyword ***CONSTRAINED** provides a way of constraining degrees of freedom to move together in some way. The keyword cards in this section are defined in alphabetical order:

- *CONSTRAINED_ADAPTIVITY**
- *CONSTRAINED_BUTT_WELD**
- *CONSTRAINED_EULER_IN_EULER**
- *CONSTRAINED_EXTRA_NODES_OPTION**
- *CONSTRAINED_GENERALIZED_WELD_OPTION_{OPTION}**
- *CONSTRAINED_GLOBAL**
- *CONSTRAINED_INTERPOLATION_{OPTION}**
- *CONSTRAINED_JOINT_OPTION_{OPTION}_{OPTION}_{OPTION}**
- *CONSTRAINED_JOINT_STIFFNESS_OPTION**
- *CONSTRAINED_LAGRANGE_IN_SOLID**
- *CONSTRAINED_LINEAR_GLOBAL**
- *CONSTRAINED_LINEAR_LOCAL**
- *CONSTRAINED_LOCAL**
- *CONSTRAINED_NODAL_RIGID_BODY_{OPTION}_{OPTION}**
- *CONSTRAINED_NODE_SET_{OPTION}**
- *CONSTRAINED_POINTS**
- *CONSTRAINED_RIGID_BODIES**
- *CONSTRAINED_RIGID_BODY_STOPPERS**
- *CONSTRAINED_RIVET_{OPTION}**
- *CONSTRAINED_SHELL_TO_SOLID**
- *CONSTRAINED_SPLINE**
- *CONSTRAINED_SPOTWELD_{OPTION}_{OPTION}**
- *CONSTRAINED_TIE-BREAK**
- *CONSTRAINED_TIED_NODES_FAILURE**

***CONSTRAINED**

***CONSTRAINED_ADAPTIVITY**

***CONSTRAINED_ADAPTIVITY**

Purpose: Define an adaptive constraint which constrains a node to the midpoint along an edge of a shell element. This keyword is also created by LS-DYNA during an adaptive calculation. This option applies to shell elements.

Card 1 2 3 4 5 6 7 8

Variable	SN	MN1	MN2					
Type	I	I	I					
Default	none	none	none					

VARIABLE

DESCRIPTION

- SN Slave node. This is the node constrained at the midpoint of an edge of a shell element.
- MN1 One node along the edge of the shell element.
- MN2 The second node along the edge.

***CONSTRAINED_BUTT_WELD**

Purpose: Define a line of coincident nodes that represent a structural butt weld between two parts defined by shell elements. Failure is based on nodal plastic strain for ductile failure and stress resultants for brittle failure. This input is much simpler than the alternative approach for defining butt welds, see *CONSTRAINED_GENERALIZED_WELD_BUTT. The local coordinate system, the effective length, and thickness for each pair of butt welded nodes are determined automatically in the definition below. In the GENERALIZED option these quantities must be defined in the input.

Card 1 2 3 4 5 6 7 8

Variable	SNSID	MNSID	EPPF	SIGF	BETA			
Type	I	I	F	F	F			
Default	none	none	0.	1.e+16	1.0			
Remarks		1, 2	3, 4	3	3			

VARIABLE

DESCRIPTION

- SNSID Slave node set ID, see *SET_NODE_OPTION.
- MNSID Master node set ID, see *SET_NODE_OPTION.
- EPPF Plastic strain at failure
- SIGF σ_f , stress at failure for brittle failure.
- BETA β , failure parameter for brittle failure.

Remarks:

1. Nodes in the master and slave sets must be given in the order they appear as one moves along the edge of the surface. An equal number of coincident nodes must be defined in each set. In a line weld the first and last node in a string of nodes can be repeated in the two sets. If the first and last pair of nodal points are identical, a circular or closed loop butt weld is assumed. See Figure 6.1, where the line butt weld and closed loop weld are illustrated.

2. Butt welds may not cross. For complicated welds, this option can be combined with the input in `*CONSTRAINED_GENERALIZED_WELD_BUTT` to handle the case where crossing occurs. Nodes in a butt weld must not be members of rigid bodies.
3. If the average volume-weighted effective plastic strain in the shell elements adjacent to a node exceeds the specified plastic strain at failure, the node is released. Brittle failure of the butt welds occurs when:

$$\beta \sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f$$

where

- σ_n = normal stress (local x)
- τ_n = shear stress in direction of weld (local y)
- τ_t = shear stress normal to weld (local z)
- σ_f = failure stress
- β = failure parameter

Component σ_n is nonzero for tensile values only. The nodes defining the slave and master sides of the butt weld must coincide. The local z-axis at a master node is normal to the *master side* plane of the butt weld at the node, and the local y-axis is taken as the vector in the direction of a line connecting the mid-points of the line segments lying on either side of the master node. The normal vector is found by summing the unit normal vectors of all shell elements on the *master side* sharing the butt welded node. The direction of the normal vector at the node is chosen so that the x-local vector points towards the elements on the slave side in order to identify tensile versus compressive stresses. The thickness of the butt weld and length of the butt weld are needed to compute the stress values. The thickness is based on the average thickness of the shell elements that share the butt welded nodal pair, and the chosen length of the butt weld is shown in Figure 6.1.

4. Butt welds may be used to simulate the effect of failure along a predetermined line, such as a seam or structural joint. When the failure criterion is reached at a nodal pair, the nodes begin to separate. As this effect propagates, the weld will appear to “unzip,” thus simulating failure of the connection.

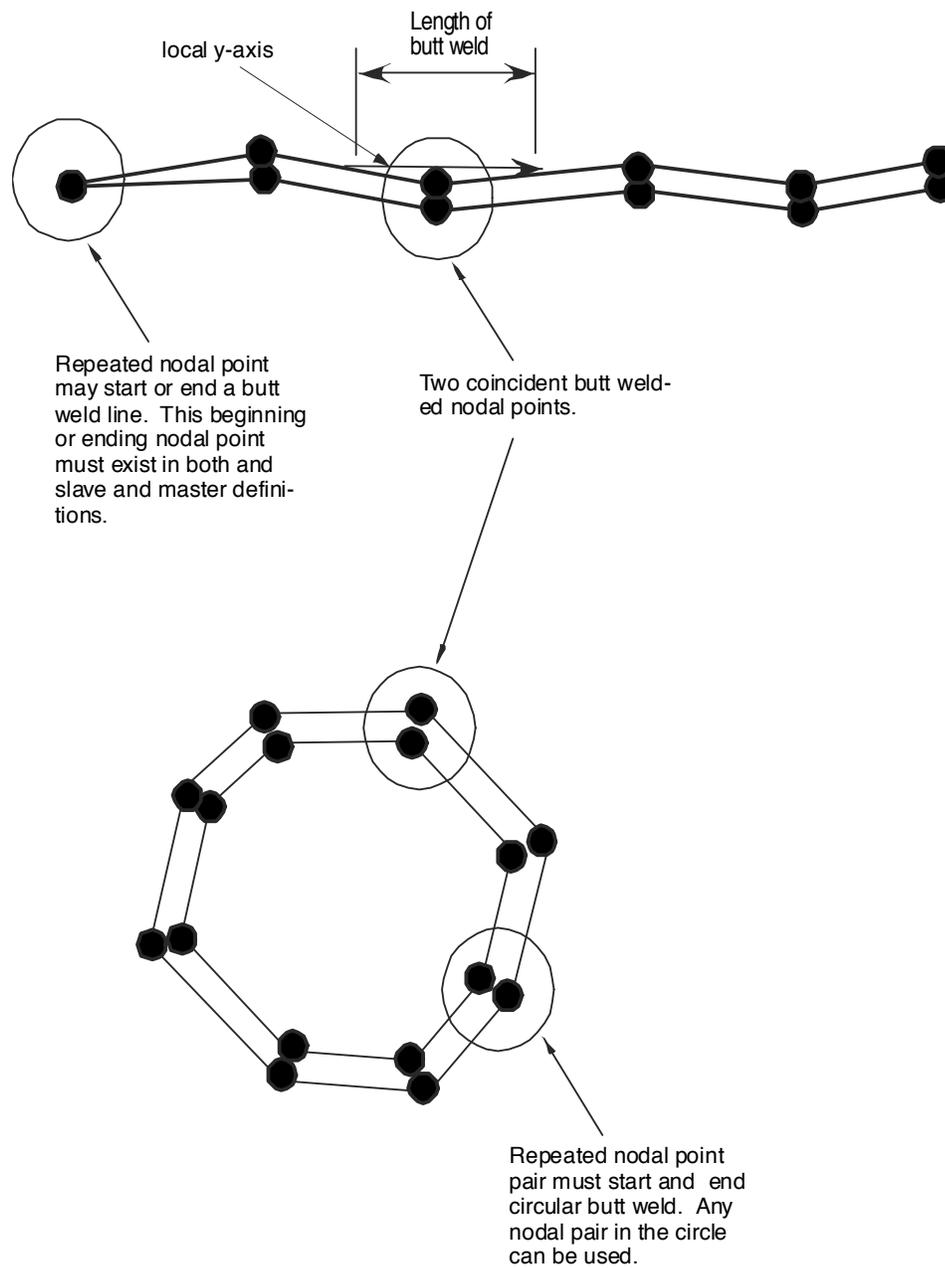


Figure 6.1. Definition of butt welds are shown above. The butt weld can be represented by a line of nodal points or by a closed loop.

*CONSTRAINED

*CONSTRAINED_EULER_IN_EULER

*CONSTRAINED_EULER_IN_EULER

Purpose: This command defines the coupling interaction between EULERIAN materials in two overlapping, geometrically similar, multi-material Eulerian mesh sets. The command allows a frictionless “contact” between two or more different Eulerian materials.

Card 1 2 3 4 5 6 7 8

Variable	PSIDSLV	PSIDMST	PFAC					
Type	I	I	F					
Default	0	0	0.1					

VARIABLE

DESCRIPTION

PSIDSLV	Part set ID of the 1 st ALE or Eulerian set of mesh(es) (slave).
PSIDMST	Part set ID of the 2 nd ALE or Eulerian set of mesh(es) (master).
PFAC	A penalty factor for the coupling interaction between the two PSIDs.

Remarks:

1. The 2 meshes must be of Eulerian formulation (the meshes are fixed in space, not moving). Consider 2 overlapping Eulerian meshes. Each Eulerian mesh contains 2 physical materials, say a vacuum and a metal. This card provides a frictionless “contact” or interaction between the 2 metals, each resides in a different Eulerian mesh system. Due to its restrictive nature, this option is currently only an experimental feature.
2. Contact pressure is built up in two overlapping Eulerian elements if their combined material fill fraction exceeds 1.0 (penalty formulation).
3. This feature needs to be combined with *MAT_VACUUM (element formulation 11).

Example:

Consider an ALE/Eulerian multi-material model (ELFORM=11) consisting of:

- PID 1 = *MAT_NULL (material 1)
- PID 2 = *MAT_VACUUM ⇒ PID 1 is merged at its boundary to PID 2.
- PID 3 = *MAT_NULL (material 3)
- PID 4 = *MAT_VACUUM ⇒ PID 3 is merged at its boundary to PID 4.

The mesh set containing PID 1 & 2 intersects or overlaps with the mesh set containing PID 3 & 4. PID 1 is given an initial velocity in the positive x direction. This will cause material 1 to contact material 3 (note that materials 2 & 4 are void). The interaction between materials 1 & 3 is possible by defining this coupling command. In this case material 1 can flow within the mesh region of PID 1 & 2 only, and material 3 can flow within the mesh region of PID 3 & 4 only.

```

$... | .....1..... | .....2..... | .....3..... | .....4..... | .....5..... | .....6..... | .....7..... | .....8
*ALE_MULTI-MATERIAL_GROUP
$   SID   SIDYTPĒ
    1       1
    2       1
    3       1
    4       1
*CONSTRAINED_EULER_IN_EULER
$   PSID1   PSID2   PENAL
    11      12      0.1
*SET_PART_LIST
  11
  1         2
*SET_PART_LIST
  12
  3         4
$... | .....1..... | .....2..... | .....3..... | .....4..... | .....5..... | .....6..... | .....7..... | .....8

```

*CONSTRAINED

*CONSTRAINED_EXTRA_NODES

*CONSTRAINED_EXTRA_NODES_OPTION

Available options include:

NODE

SET

Purpose: Define extra nodes for rigid body.

Card	1	2	3	4	5	6	7	8
Variable	PID	NID/NSID	IFLAG					
Type	I	I	I					
Default	none	none	0					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID of rigid body to which the nodes will be added, see *PART.
NID/NSID	Node (option: <u>_NODE</u>) or node set ID (option: <u>_SET</u>), see *SET_NODE, of added nodes.
IFLAG	This flag is meaningful if and only if the inertia properties of the Part ID are defined in PART_INERTIA. If set to unity, the center-of-gravity, the translational mass, and the inertia matrix of the PID will be updated to reflect the merged nodal masses of the node or node set. If IFLAG is defaulted to zero, the merged nodes will not affect the properties defined in PART_INERTIA since it is assumed the properties already account for merged nodes.

Remarks:

Extra nodes for rigid bodies may be placed anywhere, even outside the body, and they are assumed to be part of the rigid body. They have many uses including:

1. The definition of draw beads in metal forming applications by listing nodes along the draw bead.
2. Placing nodes where joints will be attached between rigid bodies.

- 3. Defining a node where point loads are to be applied or where springs may be attached.
- 4. Defining a lumped mass at a particular location.

and so on. The coordinates of the extra nodes are updated according to the rigid body motion.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONSTRAINED_EXTRA_NODES_NODE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Rigidly attach nodes 285 and 4576 to part 14. (Part 14 MUST be a rigid body.)
$
*CONSTRAINED_EXTRA_NODES_NODE
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$ pid nid
$ 14 285
$ 14 4576
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONSTRAINED_EXTRA_NODES_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Rigidly attach all nodes in set 4 to part 17. (Part 17 MUST be a rigid body.)
$
$ In this example, four nodes from a deformable body are attached
$ to rigid body 17 as a means of joining the two parts.
$
*CONSTRAINED_EXTRA_NODES_SET
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$ pid nsid
$ 17 4
$
$
*SET_NODE_LIST
$ sid
$ 4
$ nid1 nid2 nid3 nid4 nid5 nid6 nid7 nid8
$ 665 778 896 827
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

***CONSTRAINED**

***CONSTRAINED_GENERALIZED_WELD**

***CONSTRAINED_GENERALIZED_WELD_OPTION_{OPTION}**

Available options include:

SPOT

FILLET

BUTT

CROSS_FILLET

COMBINED

To define an ID for the weld use the option:

ID

Purpose: Define spot, fillet, butt, and other types of welds. Coincident nodes are permitted if the local coordinate ID is defined. For the spot weld a local coordinate ID is not required if the nodes are offset. Failures can include both the plastic and brittle failures. These can be used either independently or together. Failure occurs when either criteria is met. The welds may undergo large rotations since the equations of rigid body mechanics are used to update their motion.

ID Card - Required if the option `_ID` is active on the keyword card.

Card 1 1 2 3 4 5 6 7 8

Variable	WID								
Type	I								
Default	0								

This card is required for all weld options.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	CID	FILTER	WINDOW	NPR	NPRT		
Type	I	I	I	E	I	I		
Default	none	none						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
WID	Optional weld ID.
NSID	Nodal set ID, see <i>*SET_NODE_OPTION</i> .
CID	Coordinate system ID for output of spot weld data to SWFORC in local system, see <i>*DEFINE_COORDINATE_OPTION</i> . CID is not required for spot welds if the nodes are not coincident.
FILTER	Number of force vectors saved for filtering. This option can eliminate spurious failures due to numerical force spikes; however, memory requirements are significant since 6 force components are stored with each vector. LE.1: no filtering EQ.n: simple average of force components divided by n or the maximum number of force vectors that are stored for the time window option below.
WINDOW	Time window for filtering. This option requires the specification of the maximum number of steps which can occur within the filtering time window. If the time step decreases too far, then the filtering time window will be ignored and the simple average is used. EQ.0: time window is not used
NPR	NFW, number of individual nodal pairs in the cross fillet or combined general weld.
NPRT	Print option in file RBDOUT. EQ.0: default from the control card, <i>*CONTROL_OUTPUT</i> , is used, see variable name IPRTF. EQ.1: data is printed EQ.2: data is not printed

*CONSTRAINED

*CONSTRAINED_GENERALIZED_WELD

Additional Card required for the CONSTRAINED_GENERALIZED_WELD_SPOT option:

Card 2 1 2 3 4 5 6 7 8

Variable	TFAIL	EPSF	SN	SS	N	M		
Type	F	F	F	F	F	F		

VARIABLE

DESCRIPTION

TFAIL	Failure time for constraint set, t_f . (default=1.E+20)
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SN	S_n , normal force at failure, only for the brittle failure of spot welds.
SS	S_s , shear force at failure, only for the brittle failure of spot welds.
N	n, exponent for normal force, only for the brittle failure of spot welds.
M	m, exponent for shear force, only for the brittle failure of spot welds.

Remarks:

Spot weld failure due to plastic straining occurs when the effective nodal plastic strain exceeds the input value, ϵ_{fail}^p . This option can model the tearing out of a spot weld from the sheet metal since the plasticity is in the material that surrounds the spot weld, not the spot weld itself. A least squares algorithm is used to generate the nodal values of plastic strains at the nodes from the element integration point values. The plastic strain is integrated through the element and the average value is projected to the nodes via a least square fit. This option should only be used for the material models related to metallic plasticity and can result in slightly increased run times.

Brittle failure of the spot welds occurs when:

$$\left(\frac{\max(f_n, 0)}{S_n} \right)^n + \left(\frac{|f_s|}{S_s} \right)^m \geq 1$$

where f_n and f_s are the normal and shear interface force. Component f_n contributes for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 6.1 the ordering of the nodes is shown for the 2 node and 3 node spot welds. This order is with respect to the local coordinate system where the local z-axis determines the tensile direction. The nodes in the spot weld may coincide. The

failure of the 3 node spot weld may occur gradually with first one node failing and later the second node may fail. For n noded spot welds the failure is progressive starting with the outer nodes (1 and n) and then moving inward to nodes 2 and n-1. Progressive failure is necessary to preclude failures that would create new rigid bodies.

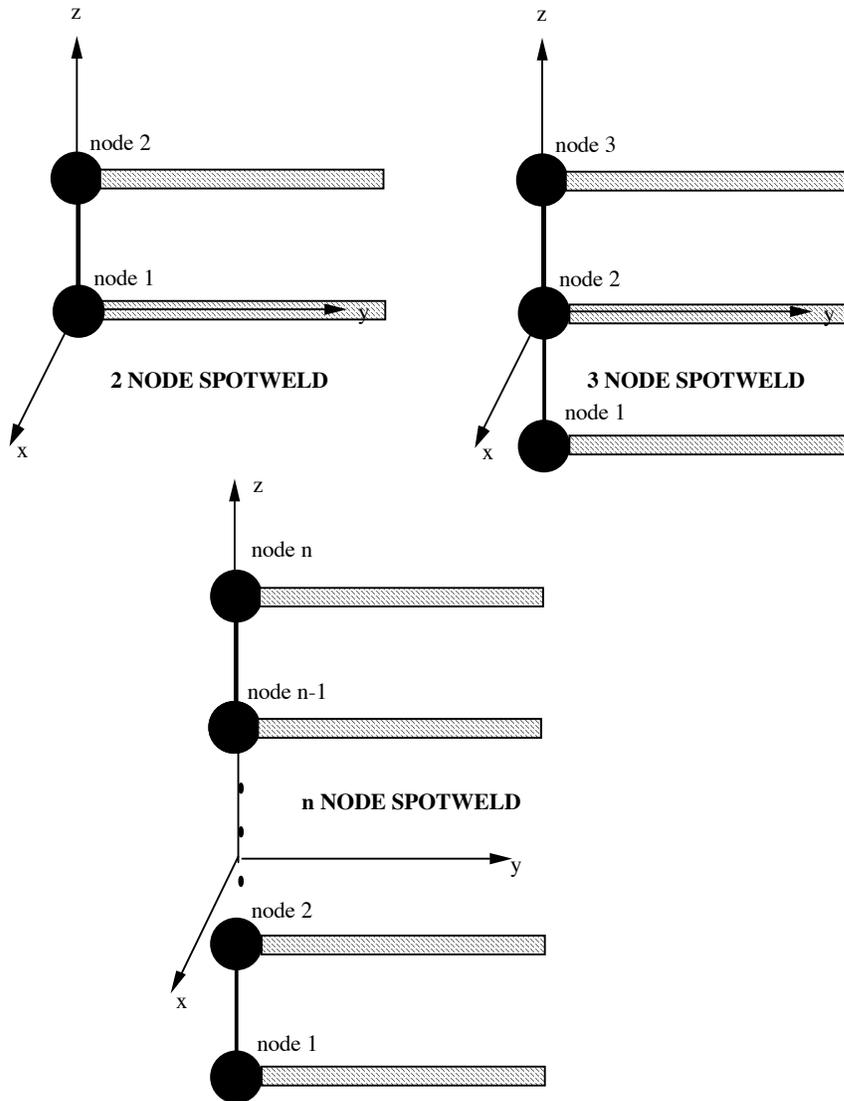


Figure 6.2. Nodal ordering and orientation of the local coordinate system is important for determining spotweld failure.

*CONSTRAINED

*CONSTRAINED_GENERALIZED_WELD

Additional Card required for the FILLET option:

Card 2 1 2 3 4 5 6 7 8

Variable	TFAIL	EPSF	SIGF	BETA	L	W	A	ALPHA
Type	F	F	F	F	F	F	F	F

VARIABLE

DESCRIPTION

TFAIL	Failure time for constraint set, t_f . (default=1.E+20)
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SIGF	σ_f , stress at failure for brittle failure.
BETA	β , failure parameter for brittle failure.
L	L, length of fillet/butt weld (see Figure 6.2 and 6.3).
W	w, width of flange (see Figure 6.2).
A	a, width of fillet weld (see Figure 6.2).
ALPHA	α , weld angle (see Figure 6.2) in degrees.

Remarks:

Ductile fillet weld failure, due to plastic straining, is treated identically to spot weld failure. Brittle failure of the fillet welds occurs when:

$$\beta \sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f$$

where

- σ_n = normal stress
- τ_n = shear stress in direction of weld (local y)
- τ_t = shear stress normal to weld (local x)
- σ_f = failure stress
- β = failure parameter

Component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 6.2 the ordering of the nodes is shown for the 2 node and 3 node fillet welds. This order is with respect

to the local coordinate system where the local z axis determines the tensile direction. The nodes in the fillet weld may coincide. The failure of the 3 node fillet weld may occur gradually with first one node failing and later the second node may fail.

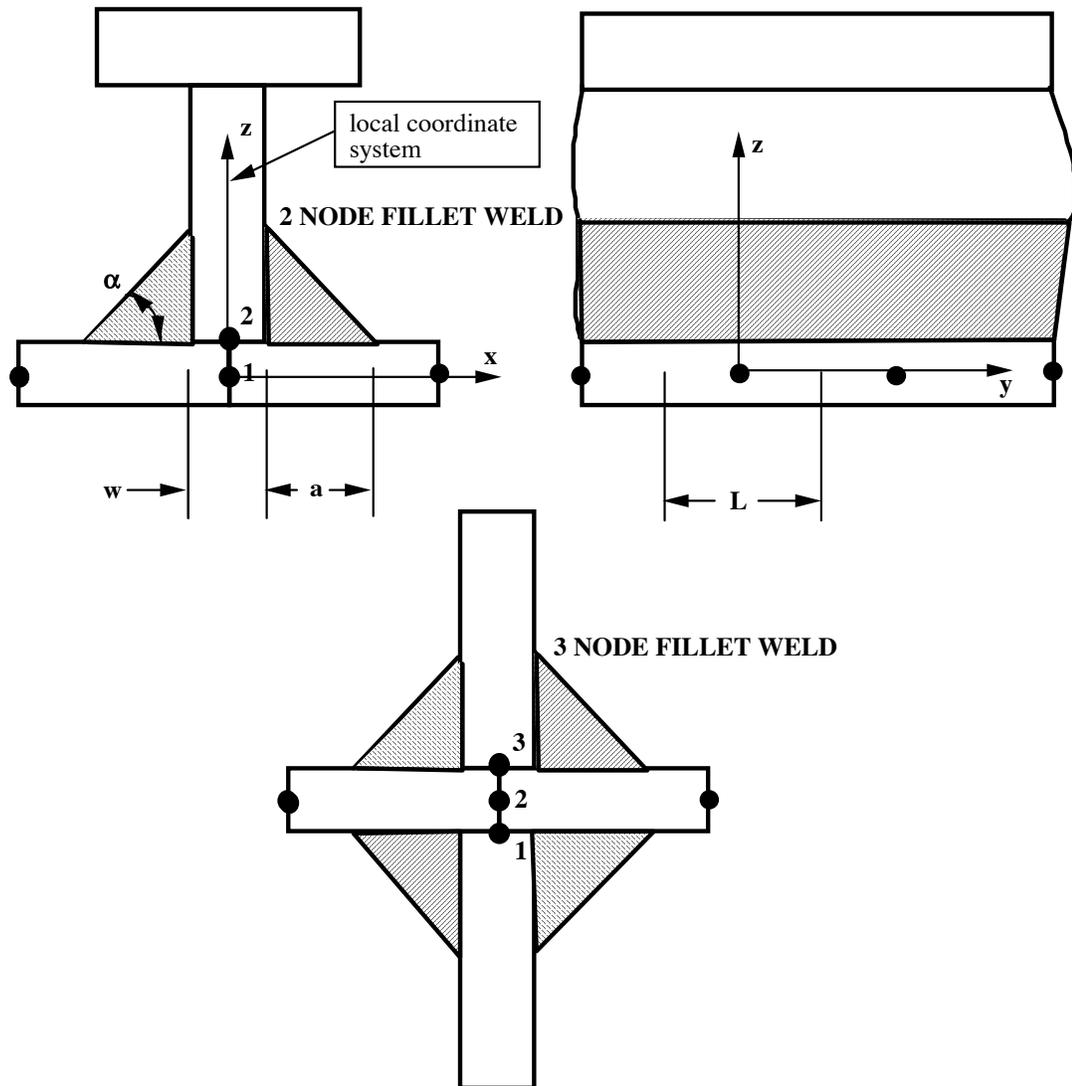


Figure 6.3. Nodal ordering and orientation of the local coordinate system is shown for fillet weld failure. The angle is defined in degrees.

*CONSTRAINED

*CONSTRAINED_GENERALIZED_WELD

Additional Card required for the BUTT option:

Card 2 1 2 3 4 5 6 7 8

Variable	TFAIL	EPSF	SIGY	BETA	L	D	LT	
Type	F	F	F	F	F	F	F	

VARIABLE

DESCRIPTION

TFAIL	Failure time for constraint set, t_f . (default=1.E+20)
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SIGY	σ_f , stress at failure for brittle failure.
BETA	β , failure parameter for brittle failure.
L	L, length of fillet/butt weld (see Figure 6.2 and 6.3).
D	d, thickness of butt weld (see Figure 6.3).
LT	L_t , transverse length of butt weld (see Figure 6.3).

Remarks:

Ductile butt weld failure, due to plastic straining, is treated identically to spot weld failure. Brittle failure of the butt welds occurs when:

$$\beta \sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f$$

where

- σ_n = normal stress
- τ_n = shear stress in direction of weld (local y)
- τ_t = shear stress normal to weld (local z)
- σ_f = failure stress
- β = failure parameter

Component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. The nodes in the butt weld may coincide.

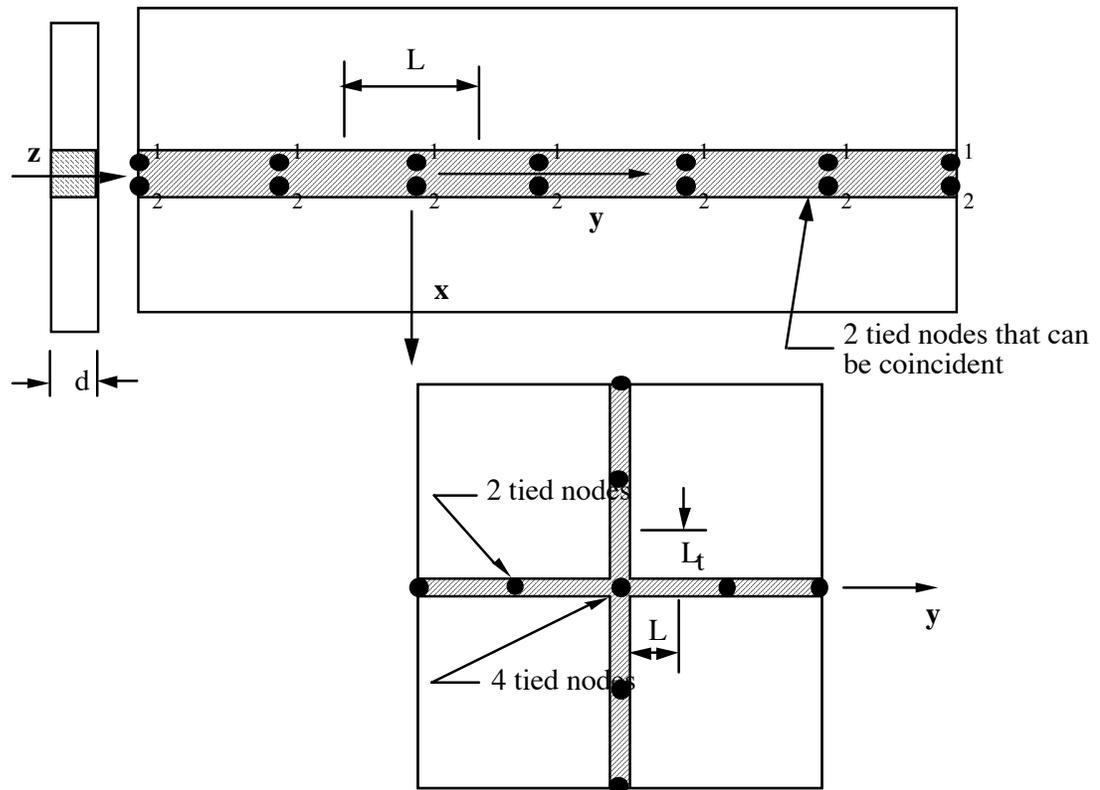


Figure 6.4. Orientation of the local coordinate system and nodal ordering is shown for butt weld failure.

*CONSTRAINED

*CONSTRAINED_GENERALIZED_WELD

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$ *CONSTRAINED_GENERALIZED_WELD_BUTT
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Weld two plates that butt up against each other at three nodal pair
$ locations. The nodal pairs are 32-33, 34-35 and 36-37.
$
$ This requires 3 separate *CONSTRAINED_GENERALIZED_WELD_BUTT definitions,
$ one for each nodal pair. Each weld is to have a length (L) = 10,
$ thickness (D) = 2, and a transverse length (Lt) = 1.
$
$ Failure is defined two ways:
$ Ductile failure if effective plastic strain exceeds 0.3
$ Brittle failure if the stress failure criteria exceeds 0.25
$ - scale the brittle failure criteria by beta = 0.9.
$ Note: beta > 1 weakens weld beta < 1 strengthens weld
$
*CONSTRAINED_GENERALIZED_WELD_BUTT
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$ nsid cid
$ 21
$ tfail epsf sigy beta L D Lt
$ 0.3 0.250 0.9 10.0 2.0 1.0
$
$
*CONSTRAINED_GENERALIZED_WELD_BUTT
$ nsid cid
$ 23
$ tfail epsf sigy beta L D Lt
$ 0.3 0.250 0.9 10.0 2.0 1.0
$
$
*CONSTRAINED_GENERALIZED_WELD_BUTT
$ nsid cid
$ 25
$ tfail epsf sigy beta L D Lt
$ 0.3 0.250 0.9 10.0 2.0 1.0
$
$
*SET_NODE_LIST
$ sid
$ 21
$ nid1 nid2
$ 32 33
*SET_NODE_LIST
$ 23
$ 34 35
*SET_NODE_LIST
$ 25
$ 36 37
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

Additional Cards (1+NPR) required for the CROSS_FILLET option:

Card 2 1 2 3 4 5 6 7 8

Variable	TFAIL	EPSF	SIGY	BETA	L	W	A	ALPHA
Type	F	F	F	F	F	F	F	F

Cards 3,4,
...,2+NPR

Variable	NODEA	NODEB	NCID					
Type	I	I	I					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TFAIL	Failure time for constraint set, t_f . (default=1.E+20)
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SIGY	σ_f , stress at failure for brittle failure.
BETA	β , failure parameter for brittle failure.
L	L, length of fillet/butt weld (see Figure 6.2 and 6.3).
W	w, width of flange (see Figure 6.2).
A	a, width of fillet weld (see Figure 6.2).
ALPHA	α , weld angle (see Figure 6.2) in degrees.
NODEA	Node ID, A, in weld pair (CROSS or COMBINED option only). See Figure 6.4.
NODEB	Node ID, B, in weld pair (CROSS or COMBINED option only).
NCID	Local coordinate system ID (CROSS or COMBINED option only).

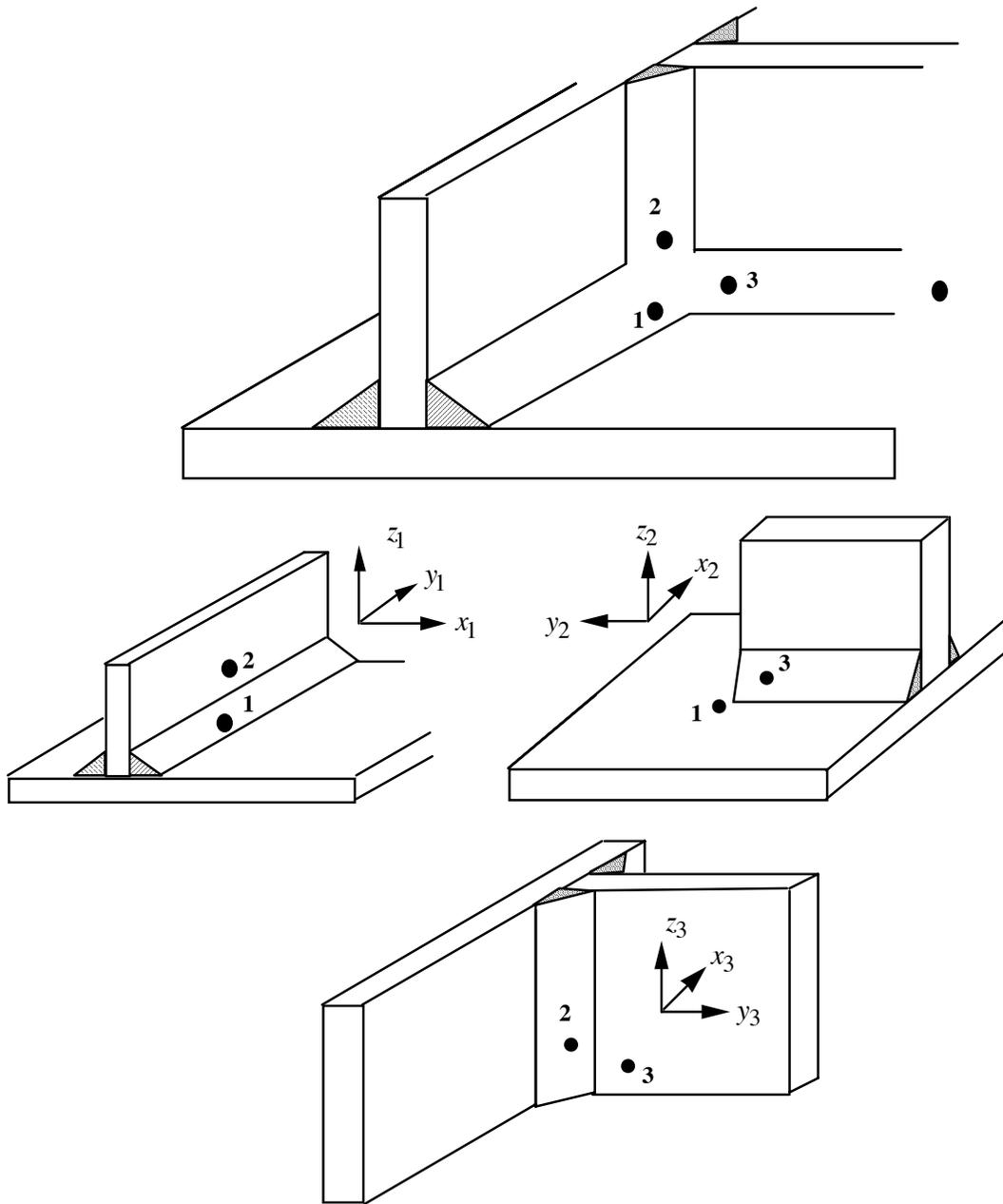


Figure 6.5. A simple cross fillet weld illustrates the required input. Here NFW=3 with nodal pairs (A=2, B=1), (A=3, B=1), and (A=3, B=2). The local coordinate axes are shown. These axes are fixed in the rigid body and are referenced to the local rigid body coordinate system which tracks the rigid body rotation.

Additional NPR Card Sets required for the COMBINED option. Repeat cards 2 and 3 below NPR times:

Card 2 1 2 3 4 5 6 7 8

Variable	TFAIL	EPSF	SIGY	BETA	L	W	A	ALPHA
Type	F	F	F	F	F	F	F	F

Card 3

Variable	NODEA	NODEB	NCID	WTYP				
Type	I	I	I	I				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TFAIL	Failure time for constraint set, t_f . (default=1.E+20)
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SIGY	σ_f , stress at failure for brittle failure.
BETA	β , failure parameter for brittle failure.
L	L, length of fillet/butt weld (see Figure 6.2 and 6.3).
W	w, width of flange (see Figure 6.2).
A	a, width of fillet weld (see Figure 6.2).
ALPHA	α , weld angle (see Figure 6.2) in degrees.
NODEA	Node ID, A, in weld pair (CROSS or COMBINED option only).
NODEB	Node ID, B, in weld pair (CROSS or COMBINED option only).
NCID	Local coordinate system ID (CROSS or COMBINED option only).
WTYPE	Weld pair type (GENERAL option only). See Figure 6.5. EQ.0: fillet weld EQ.1: butt weld

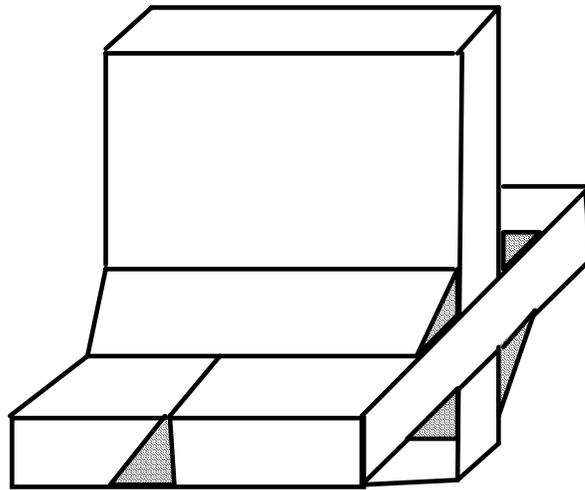


Figure 6.5. A combined weld is a mixture of fillet and butt welds.

***CONSTRAINED_GLOBAL**

Purpose: Define a global boundary constraint plane.

Card 1 2 3 4 5 6 7 8

Variable	TC	RC	DIR	X	Y	Z		
Type	I	I	I	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
TC	Translational Constraint: EQ.1: constrained x translation, EQ.2: constrained y translation, EQ.3: constrained z translation, EQ.4: constrained x and y translations, EQ.5: constrained y and z translations, EQ.6: constrained x and z translations, EQ.7: constrained x, y, and z translations,
RC	Rotational Constraint: EQ.1: constrained x-rotation, EQ.2: constrained y-rotation, EQ.3: constrained z-rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.
DIR	Direction of normal EQ.1: global x, EQ.2: global y, EQ.3: global z.
X	x-offset coordinate
Y	y-offset coordinate
Z	z-offset coordinate

Remarks:

Nodes within a mesh-size-dependent tolerance are constrained on a global plane. This option is recommended for use with r-method adaptive remeshing where nodal constraints are lost during the remeshing phase.

***CONSTRAINED_INTERPOLATION_{OPTION}**

Available options include:

<BLANK>

LOCAL

Purpose: Define an interpolation constraint. With this constraint type, the motion of a single dependent node is interpolated from the motion of a set of independent nodes. This option is useful for the load redistribution of a load, which can be either a translational force or moment, applied to the dependent node to the surrounding independent nodes, and it can also be used to model shell-brick and beam-brick interfaces. The mass and rotary inertia of the dependent nodal point is also redistributed. This constraint is applied in the global coordinate system unless the option LOCAL is active. One *CONSTRAINED_INTERPOLATION card is required for each constraint definition. The input list of independent nodes is terminated when the next "*" card is found. In explicit calculations the independent nodes cannot be dependent nodes in other constraints such as nodal rigid bodies. This latter restriction does not apply to implicit calculations.

Card 1 2 3 4 5 6 7 8

Variable	ICID	DNID	DDOF	CIDD				
Type	I	I	I	I				
Default	0	0	123456	optional				

Cards 2, 3, 4, etc. Define one card per independent node. If the option LOCAL is active, define two cards per independent node. Input is terminated when a "*" card is found.

 1 2 3 4 5 6 7 8

Variable	INID	IDOF	TWGHTX	TWGHTY	TWGHTZ	RWGHTX	RWGHTY	RWGHTZ
Type	I	I	F	F	F	F	F	F
Default	0	123456	1.0	TWGHTX	TWGHTX	TWGHTX	TWGHTX	TWGHTX

*CONSTRAINED

*CONSTRAINED_INTERPOLATION

Define the second card if and only if the option LOCAL is active

Card	1	2	3	4	5	6	7	8
Variable	CIDI							
Type	I							
Default	0							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ICID	Interpolation constraint ID.
DNID	Dependent node ID. This node should not be a member of a rigid body, or elsewhere constrained in the input.
DDOF	Dependent degrees-of-freedom. The list of dependent degrees-of-freedom consists of a number with up to six digits, with each digit representing a degree of freedom. For example, the value 1356 indicates that degrees of freedom 1, 3, 5, and 6 are controlled by the constraint. The default is 123456. Digit: degree of freedom ID's: EQ.1: x EQ.2: y EQ.3: z EQ.4: rotation about x axis EQ.5: rotation about y axis EQ.6: rotation about z axis
CIDD	Local coordinate system ID if LOCAL option is active. If blank the global coordinate system is assumed.
INID	Independent node ID.
IDOF	Independent degrees-of-freedom using the same form as for the dependent degrees-of-freedom, DDOF, above.
TWGHTX	Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the x-translational component. It is normally sufficient to define only TWGHTX even if its degree-of-freedom is inactive since the other factors are set equal to this input value as the default. There is no requirement on the values that are chosen as the weighting factors, i.e., that they sum to unity. The default value for the weighting factor is unity.

VARIABLE	DESCRIPTION
TWGHTY	Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the y-translational component.
TWGHTZ	Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the z-translational component.
RWGHTX	Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the x-rotational component.
RWGHTY	Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the y-rotational component.
RWGHTZ	Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the z-rotational component.
CIDI	Local coordinate system ID if LOCAL option is active. If blank the global coordinate system is assumed.

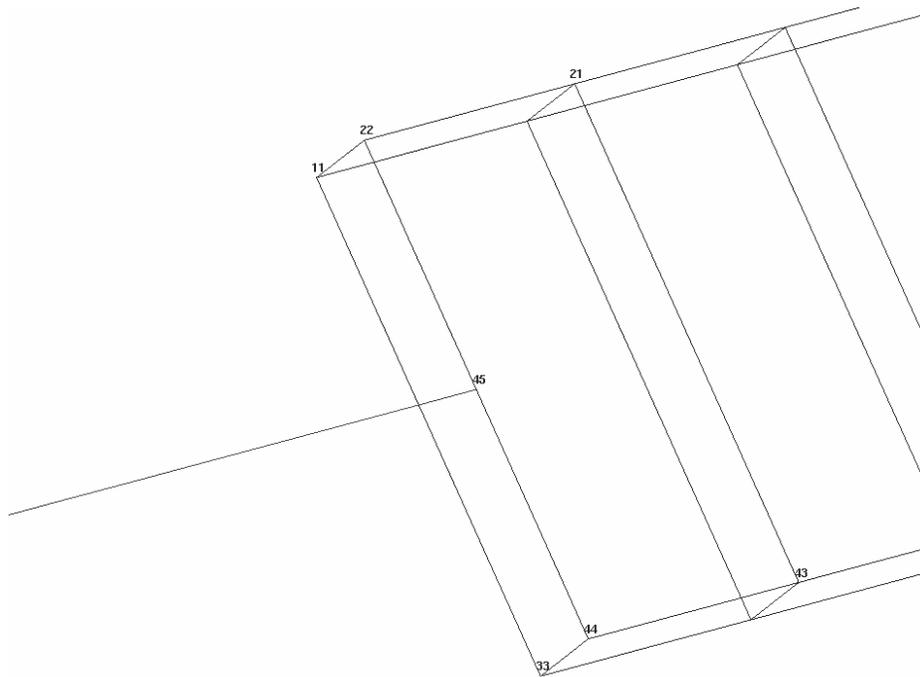
```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONSTRAINED_INTERPOLATION (Beam to solid coupling)
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Tie a beam element to a solid element.
$
$ The node of the beam to be tied does not share a common node with the solids.
$ If the beam node is shared, for example, then set ddof=456.
$
*CONSTRAINED_INTERPOLATION
$
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$ icid dnid ddof
$ 1 45 123456
$ inid idof twghtx twghty twghtz rwghtx rwghty rwghtz
$ 22 123
$ 44 123
$ 43 123
$
*.....
$

```

*CONSTRAINED

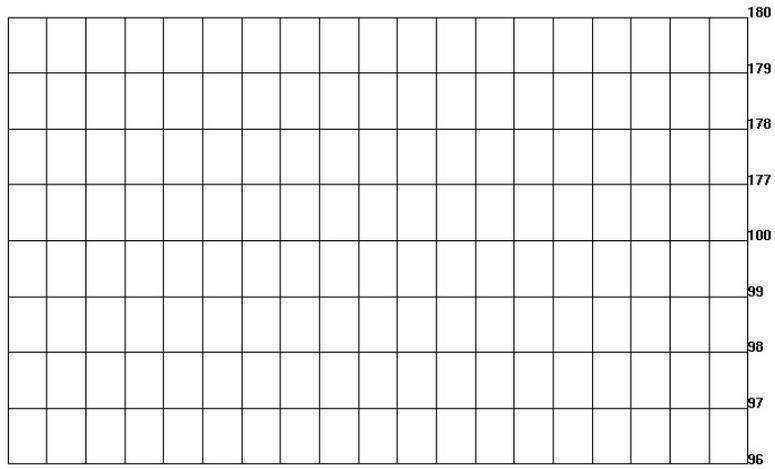
*CONSTRAINED_INTERPOLATION



```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONSTRAINED_INTERPOLATION (Load redistribution)
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Moment about normal axis of node 100 is converted to an equivalent load by
$ applying x-force resultants to the nodes lying along the right boundary
$
*DEFINE_CURVE
1,0,0.,0.,0.,0.,0
0.,0.
.1,10000.
*LOAD_NODE_POINT
100,6,1,1.0
$
*CONSTRAINED_INTERPOLATION
$
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$    icid    dnid    ddof
$      1      100     5
$    inid    idof    twghtx   twghty   twghtz   rwghtx   rwghty   rwghtz
$      96      1
$      97      1
$      98      1
$      99      1
$     177      1
$     178      1
$     179      1
$     180      1
$
*.....
$

```



***CONSTRAINED_JOINT_OPTION_{OPTION}_{OPTION}_{OPTION}**

Available forms include (one is mandatory):

CONSTRAINED_JOINT_SPHERICAL
CONSTRAINED_JOINT_REVOLUTE
CONSTRAINED_JOINT_CYLINDRICAL
CONSTRAINED_JOINT_PLANAR
CONSTRAINED_JOINT_UNIVERSAL
CONSTRAINED_JOINT_TRANSLATIONAL
CONSTRAINED_JOINT_LOCKING
CONSTRAINED_JOINT_TRANSLATIONAL_MOTOR
CONSTRAINED_JOINT_ROTATIONAL_MOTOR
CONSTRAINED_JOINT_GEAR
CONSTRAINED_JOINT_RACK_AND_PINION
CONSTRAINED_JOINT_CONSTANT_VELOCITY
CONSTRAINED_JOINT_PULLEY
CONSTRAINED_JOINT_SCREW

If the force output data is to be transformed into a local coordinate use the option:

LOCAL

to define a joint ID and heading the following option is available:

ID

and to define failure for penalty-based joints (LMF=0 in *CONTROL_RIGID) use:

FAILURE

The ordering of the bracketed options is arbitrary.

Purpose: Define a joint between two rigid bodies, see Figure 6.6.

Card Format:

Card 1 is required for all joint types.

Card 2 is required for joint types: MOTOR, GEARS, RACK_AND_PINION, PULLEY, and SCREW

Optional Card is required only if LOCAL is specified in the keyword.

In the first seven joint types above excepting the Universal joint, the nodal points within the nodal pairs (1,2), (3,4), and (5,6) (see Figure 6.6) should coincide in the initial configuration, and the nodal pairs should be as far apart as possible to obtain the best behavior. For the Universal Joint the nodes within the nodal pair (3,4) do not coincide, but the lines drawn between nodes (1,3) and (2,4) must be perpendicular.

***CONSTRAINED**

***CONSTRAINED_JOINT**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
N3	Node 3, in rigid body A. Define for all joint types except SPHERICAL.
N4	Node 4, in rigid body B. Define for all joint types except SPHERICAL.
N5	Node 5, in rigid body A. Define for joint types TRANSLATIONAL, LOCKING, ROTATIONAL_MOTOR, CONSTANT_VELOCITY, GEARS, RACK_AND_PINION, PULLEY, and SCREW
N6	Node 6, in rigid body B. Define for joint types TRANSLATIONAL, LOCKING, ROTATIONAL_MOTOR, CONSTANT_VELOCITY, GEARS, RACK_AND_PINION, PULLEY, and SCREW
RPS	Relative penalty stiffness (default = 1.0).
DAMP	Damping scale factor on default damping value. (Revolute and Spherical Joints): EQ.0.0: default is set to 1.0, LE.0.01 and GT.0.0: no damping is used.

Card 2. Required for joint types MOTOR, GEARS, RACK_AND_PINION, PULLEY, and SCREW only.

Card 1	1	2	3	4	5	6	7	8
Variable	PARAM	LCID	TYPE	R1				
Type	F	I	I	F				
Default	None							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PARAM	Parameter which a function of joint type. Leave blank for MOTORS. Gears: define R_2 / R_1 Rack and Pinion: define h Pulley: define R_2 / R_1 Screw: define \dot{x} / ω
LCID	Define load curve ID for MOTOR joints.
TYPE	Define integer flag for MOTOR joints as follows: EQ.0: translational/rotational velocity EQ.1: translational/rotational acceleration EQ.2: translational/rotational displacement
R1	Radius, R_1 , for the gear and pulley joint type. If left undefined, nodal points 5 and 6 are assumed to be on the outer radius.

***CONSTRAINED_JOINT**

***CONSTRAINED**

Optional: Required only if LOCAL is specified after the keyword.

Card 1 1 2 3 4 5 6 7 8

Variable	RAID	LST						
Type	I	I						
Default	0	0						

VARIABLE

DESCRIPTION

RAID Rigid body or accelerometer ID. The force resultants are output in the local system of the rigid body or accelerometer.

LST Flag for local system type:
EQ.0: rigid body
EQ.1: accelerometer

Optional: Required only if FAILURE is specified after the keyword.

Card 1 1 2 3 4 5 6 7 8

Variable	CID	TFAIL	COUPL					
Type	I	F	F					
Default	0	0	0.					

*CONSTRAINED

*CONSTRAINED_JOINT

Card 2 1 2 3 4 5 6 7 8

Variable	NXX	NYY	NZZ	MXX	MYY	MZZ		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE

DESCRIPTION

CID	Coordinate ID for resultants in the failure criteria. If zero, the global coordinate system is used.
TFAIL	Time for joint failure. If zero, joint never fails.
COUPL	Coupling between the force and moment failure criteria. If COUPL is less than or equal to zero, the failure criteria is identical to the spotwelds. When COUPL is greater than zero, the force and moment results are considered independently. See the remark below.
NXX	Axial force resultant N_{xxF} at failure. If zero, failure due to this component is not considered.
NYY	Force resultant N_{yyF} at failure. If zero, failure due to this component is not considered.
NZZ	Force resultant N_{zzF} at failure. If zero, failure due to this component is not considered.
MXX	Torsional moment resultant M_{xxF} at failure. If zero, failure due to this component is not considered.
MYY	Moment resultant M_{yyF} at failure. If zero, failure due to this component is not considered.
MZZ	Moment resultant M_{zzF} at failure. If zero, failure due to this component is not considered.

Remarks:

The moments for the revolute, cylindrical, planar, translational, and locking joints are calculated at the midpoint of nodes N1 and N3. The moments for the spherical, universal, constant velocity, gear, pulley, and rack and pinion joints are calculated at node N1.

When COUPL is less than or equal to zero, the failure criteria is

$$\left(\frac{\max(N_{xx}, 0)}{N_{xx_F}}\right)^2 + \left(\frac{N_{yy}}{N_{yy_F}}\right)^2 + \left(\frac{N_{zz}}{N_{zz_F}}\right)^2 + \left(\frac{M_{xx}}{M_{xx_F}}\right)^2 + \left(\frac{M_{yy}}{M_{yy_F}}\right)^2 + \left(\frac{M_{zz}}{M_{zz_F}}\right)^2 - 1 = 0$$

otherwise, it is

$$\left(\frac{\max(N_{xx}, 0)}{N_{xx_F}}\right)^2 + \left(\frac{N_{yy}}{N_{yy_F}}\right)^2 + \left(\frac{N_{zz}}{N_{zz_F}}\right)^2 - 1 = 0 \text{ and } \left(\frac{M_{xx}}{M_{xx_F}}\right)^2 + \left(\frac{M_{yy}}{M_{yy_F}}\right)^2 + \left(\frac{M_{zz}}{M_{zz_F}}\right)^2 - 1 = 0.$$

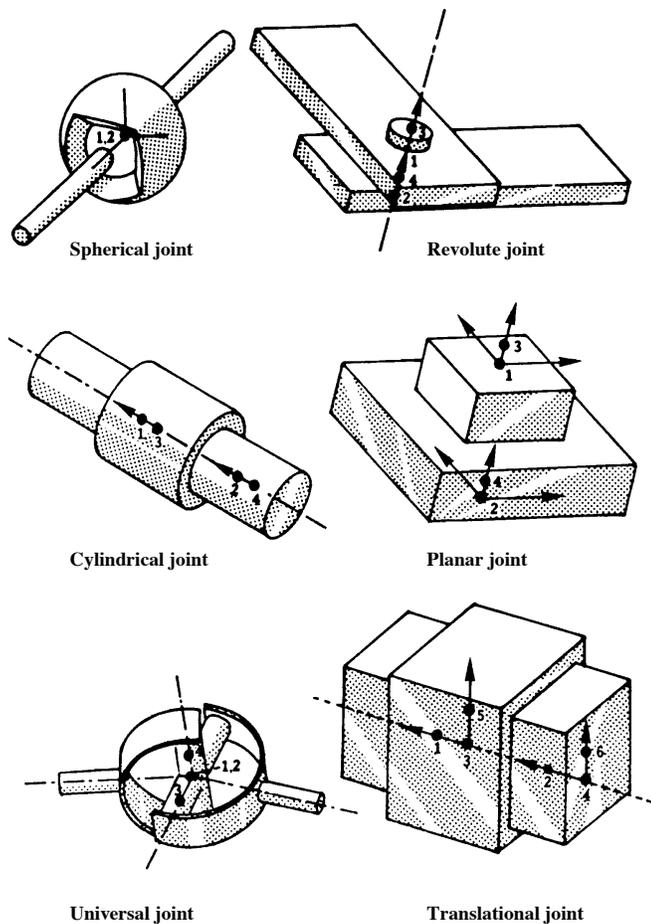


Figure 6.6 Joint definitions 1-6.

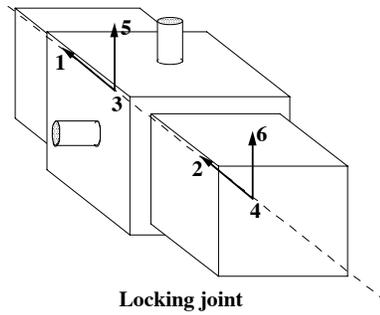


Figure 6.7. Locking joint.

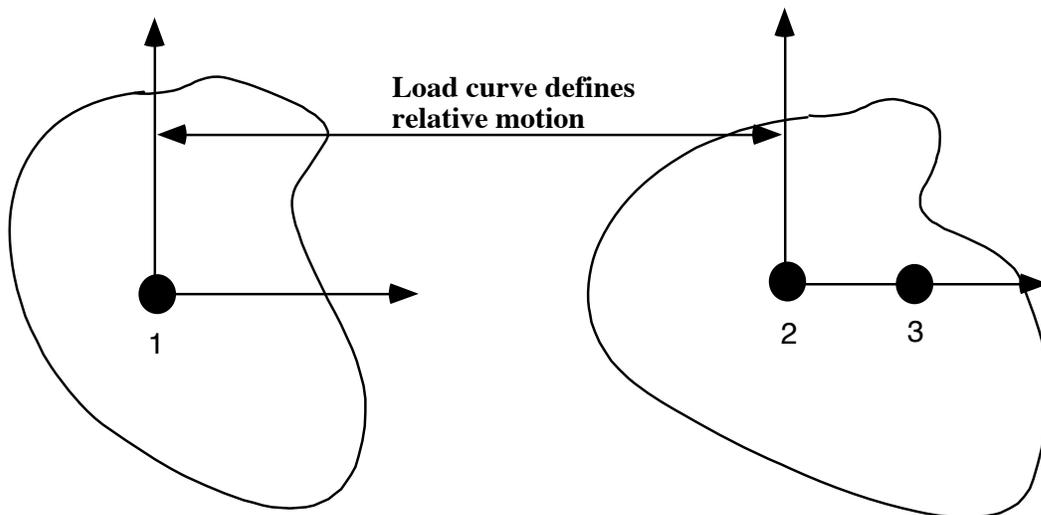


Figure 6.8. Translational motor joint. This joint can be used in combination with the translational or the cylindrical joint.

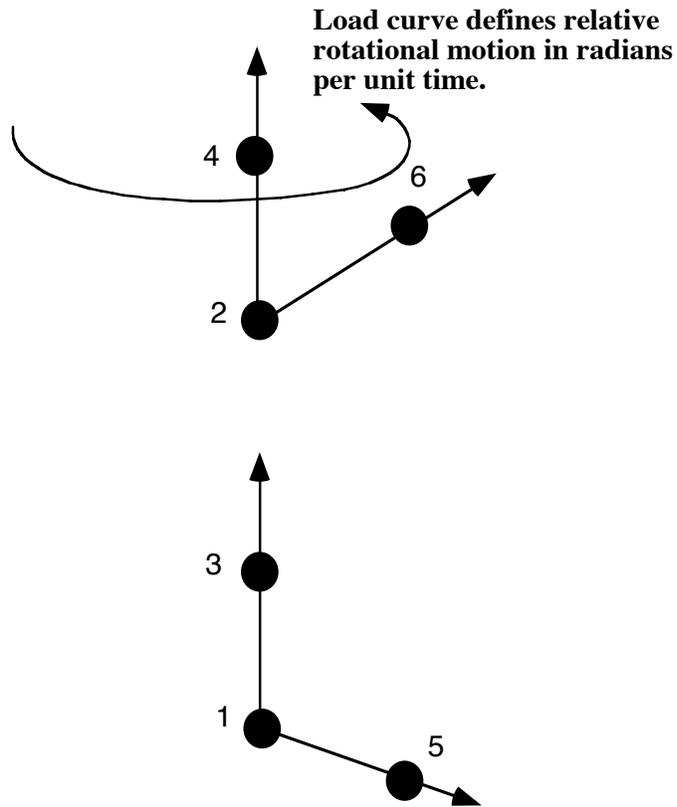


Figure 6.9. Rotational motor joint. This joint can be used in combination with other joints such as the revolute or cylindrical joints.

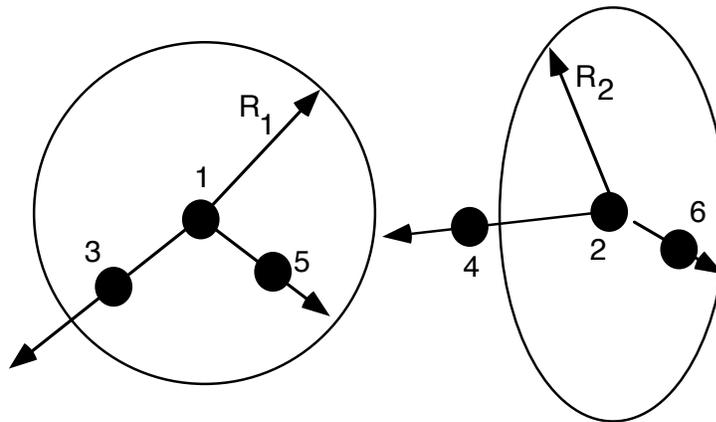


Figure 6.10. Gear joint. Nodal pairs (1,3) and (2,4) define axes that are orthogonal to the gears. Nodal pairs (1,5) and (2,6) define vectors in the plane of the gears. The ratio $\frac{R_2}{R_1}$ is specified.

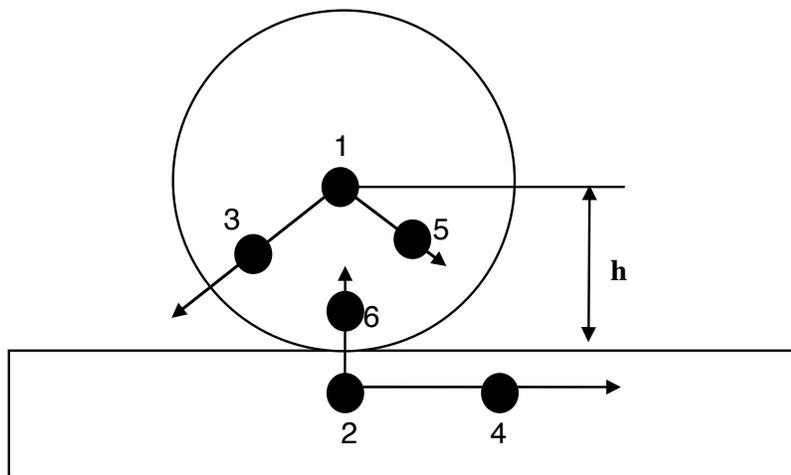


Figure 6.11. Rack and pinion joint. Nodal pair (1,3) defines a vector that is orthogonal to the plane of the gear. Nodal pair (1,5) is a vector in the plane of the gear. Nodal pair (2,4) defines the direction of travel for the second body. The value h is specified. The velocity of the rack is $\omega_{\text{pinion}} \times h$.

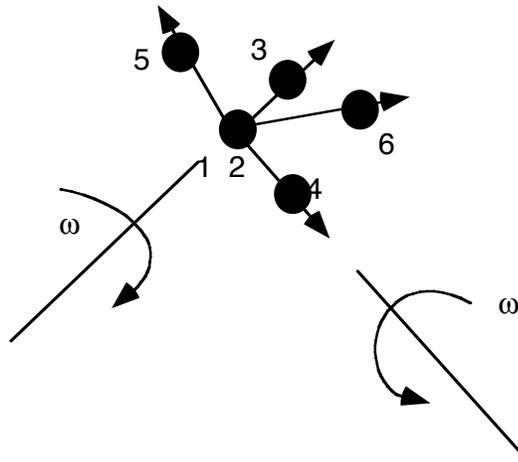


Figure 6.12. Constant velocity joint. Nodal pairs (1,3) and (2,4) define an axes for the constant angular velocity, and nodal pairs (1,5) are orthogonal vectors. Here nodal points 1 and 2 must be coincident.

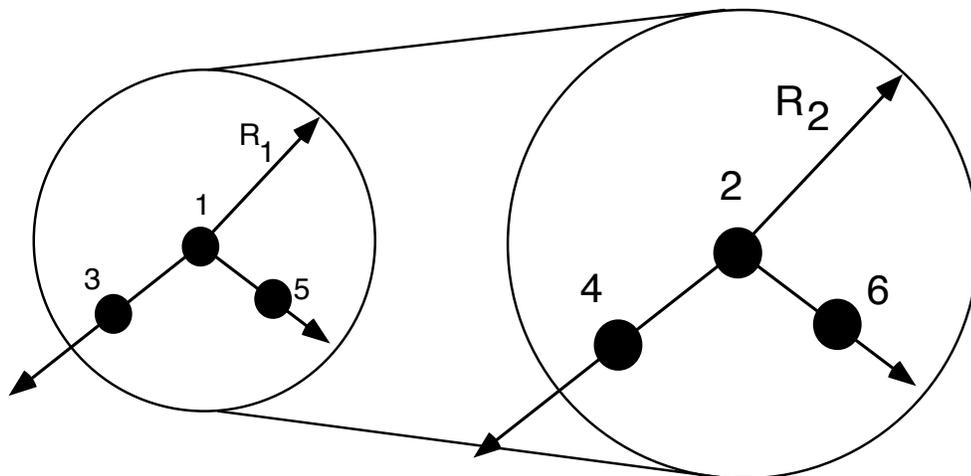


Figure 6.13. Pulley joint. Nodal pairs (1,3) and (2,4) define axes that are orthogonal to the pulleys. Nodal pairs (1,5) and (2,6) define vectors in the plane of the pulleys. The ratio $\frac{R_2}{R_1}$ is specified.

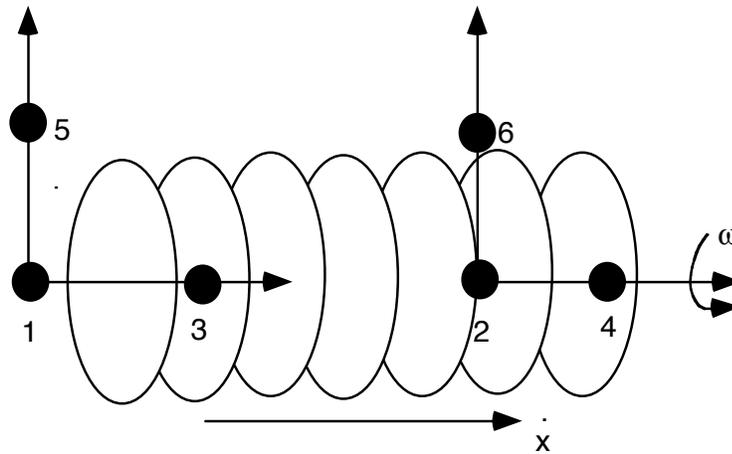


Figure 6.14. Screw joint. The second body translates in response to the spin of the first body. Nodal pairs (1,3) and (2,4) lie along the same axis and nodal pairs (1,5) and (2,6) are orthogonal vectors. The helix ratio, $\frac{\dot{x}}{\omega}$, is specified.

```

$$$
$$$$ *CONSTRAINED_JOINT_PLANAR
$$$$
$ Define a planar joint between two rigid bodies.
$ - Nodes 91 and 94 are on rigid body 1.
$ - Nodes 21 and 150 are on rigid body 2.
$ - Nodes 91 and 21 must be coincident.
$ * These nodes define the origin of the joint plane.
$ - Nodes 94 and 150 must be coincident.
$ * To accomplish this, massless node 150 is artificially created at
$ the same coordinates as node 94 and then added to rigid body 2.
$ * These nodes define the normal of the joint plane (e.g., the
$ vector from node 91 to 94 defines the planes' normal).
$
*CONSTRAINED_JOINT_PLANAR
$
$>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$      n1      n2      n3      n4      n5      n6      rps
$      91      21      94      150          0.000E+00
$
$
*NODE
$      nid      x      y      z      tc      rc
$      150      0.00      3.00      0.00      0      0
$
*CONSTRAINED_EXTRA_NODES_SET
$      pid      nsid
$      2      6
$

```


*CONSTRAINED

*CONSTRAINED_JOINT_STIFFNESS

*CONSTRAINED_JOINT_STIFFNESS_OPTION

Available options include:

FLEXION-TORSION

GENERALIZED

TRANSLATIONAL

Purpose: Define optional rotational and translational joint stiffness for joints defined by *CONSTRAINED_JOINT_OPTION. These definitions apply to all joints even though degrees of freedom that are considered in the joint stiffness capability may be constrained out in some joint types. The energy that is dissipated with the joint stiffness option is written for each joint in joint force file with the default name, JNTFORC. In the global energy balance this energy is included with the energy of the discrete elements, i.e., the springs and dampers.

Card Formats:

Card 1 is common to all joint stiffness types.
Cards 2 to 4 are unique for each stiffness type.

Card 1 - Required for all joint stiffness types.

Card 1 1 2 3 4 5 6 7 8

Variable	JSID	PIDA	PIDB	CIDA	CIDB	JID		
Type	I	I	I	I	I	I		
Default	none	none	none	none	CIDA	none		

VARIABLE

DESCRIPTION

JSID	Joint stiffness ID
PIDA	Part ID for rigid body A, see *PART.
PIDB	Part ID for rigid body B, see *PART.

VARIABLE	DESCRIPTION
CIDA	Coordinate ID for rigid body A, see *DEFINE_COORDINATE_OPTION. For the translational stiffness the local coordinate system must be defined by nodal points, *DEFINE_COORDINATE_NODES, since the first nodal point in each coordinate system is used to track the motion.
CIDB	Coordinate ID for rigid body B. If zero, the coordinate ID for rigid body A is used, see *DEFINE_COORDINATE_OPTION. For the translational stiffness the local coordinate system must be defined by nodal points, *DEFINE_COORDINATE_NODES, since the first nodal point in each coordinate system is used to track the motion.
JID	Joint ID for the joint reaction forces. If zero, tables can't be used in place of load curves for defining the frictional moments.

Card 2 of 4 - Required for FLEXION-TORSION stiffness.

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDAL	LCIDG	LCIDBT	DLCIDAL	DLCIDG	DLCIDBT		
Type	I	I	I	I	I	I		
Default	none	1.0	none	none	1.0	none		

Card 3

Variable	ESAL	FMAL	ESBT	FMBT				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

CONSTRAINED**CONSTRAINED_JOINT_STIFFNESS**

Card 4 1 2 3 4 5 6 7 8

Variable	SAAL	NSABT	PSABT					
Type	F	F	F					
Default	not used	not used	not used					

VARIABLE**DESCRIPTION**

LCIDAL	Load curve ID for α -moment versus rotation in radians. See Figure 6.15 where it should be noted that $0 \leq \alpha \leq \pi$. If zero, the applied moment is set to zero. See *DEFINE_CURVE.
LCIDG	Load curve ID for γ versus a scale factor which scales the bending moment due to the α rotation. This load curve should be defined in the interval $-\pi \leq \gamma \leq \pi$. If zero the scale factor defaults to 1.0. See *DEFINE_CURVE.
LCIDBT	Load curve ID for β -torsion moment versus twist in radians. If zero the applied twist is set to zero. See *DEFINE_CURVE.
DLCIDAL	Load curve ID for α -damping moment versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDG	Load curve ID for γ -damping scale factor versus rate of rotation in radians per unit time. This scale factor scales the α -damping moment. If zero, the scale factor defaults to one. See *DEFINE_CURVE.
DLCIDBT	Load curve ID for β -damping torque versus rate of twist. If zero damping is not considered. See *DEFINE_CURVE.
ESAL	Elastic stiffness per unit radian for friction and stop angles for α rotation, see Figure 6.15. If zero, friction and stop angles are inactive for α rotation.
FMAL	Frictional moment limiting value for α rotation. If zero, friction is inactive for α rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus α rotation, see Figure 6.15. A table permits the moment to also be a

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	function of the joint reaction force and requires the specification of JID on Card 1.
ESBT	Elastic stiffness per unit radian for friction and stop angles for β twist, see Figure 6.15. If zero, friction and stop angles are inactive for β twist.
FMBT	Frictional moment limiting value for β twist. If zero, friction is inactive for β twist. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus β rotation, see Figure 6.15 A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on Card 1.
SAAL	Stop angle in degrees for α rotation where $0 \leq \alpha \leq \pi$. Ignored if zero.
NSABT	Stop angle in degrees for negative β rotation. Ignored if zero.
PSABT	Stop angle in degrees for positive β rotation. Ignored if zero.

Remarks:

This option simulates a flexion-torsion behavior of a joint in a slightly different fashion than with the generalized joint option.

After the stop angles are reached the torques increase linearly to resist further angular motion using the stiffness values on Card 3. If the stiffness value is too low or zero, the stop will be violated.

The moment resultants generated from the moment versus rotation curve, damping moment versus rate-of-rotation curve, and friction are evaluated independently and are added together.

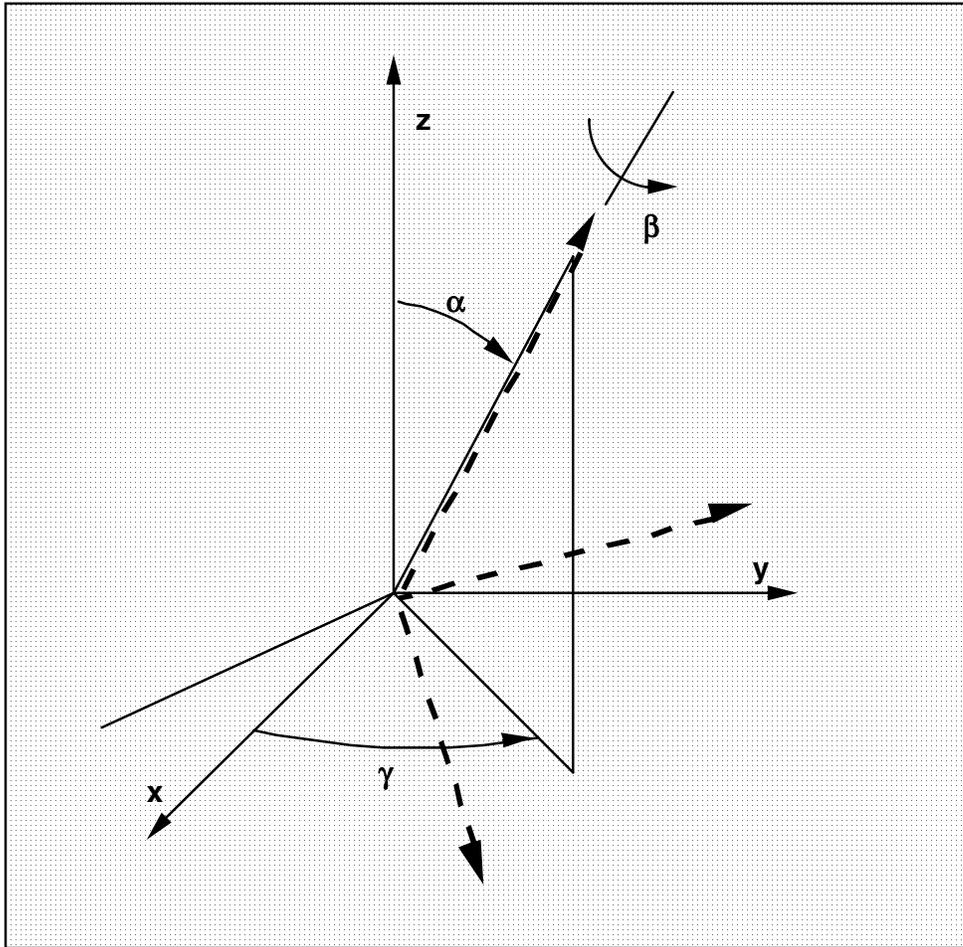


Figure 6.15. Flexion-torsion joint angles. If the initial positions of the local coordinate axes of the two rigid bodies connected by the joint do not coincide, the angles, α and γ , are initialized and torques will develop instantaneously based on the defined load curves. The angle β is also initialized but no torque will develop about the local axis on which β is measured. Rather, β will be measured relative to the computed offset.

Card 2-4 - Required for GENERALIZED stiffness.

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDPH	LCIDT	LCIDPS	DLCIDPH	DLCIDT	DLCIDPS		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	none		

Card 3

Variable	ESPH	FMPH	EST	FMT	ESPS	FMPS		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 4

Variable	NSAPH	PSAPH	NSAT	PSAT	NSAPS	PSAPS		
Type	F	F	F	F	F	F		
Default	not used							

VARIABLE**DESCRIPTION**

LCIDPH	Load curve ID for ϕ -moment versus rotation in radians. See Figure 6.16. If zero, the applied moment is set to 0.0. See *DEFINE_CURVE.
LCIDT	Load curve ID for θ -moment versus rotation in radians. If zero, the applied moment is set to 0.0. See *DEFINE_CURVE.
LCIDPS	Load curve ID for ψ -moment versus rotation in radians. If zero, the applied moment is set to 0.0. See *DEFINE_CURVE.

VARIABLE	DESCRIPTION
DLCIDPH	Load curve ID for ϕ -damping moment versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDT	Load curve ID for θ -damping moment versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDPS	Load curve ID for ψ -damping torque versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.
ESPH	Elastic stiffness per unit radian for friction and stop angles for ϕ rotation. See Figure 6.17. If zero, friction and stop angles are inactive for ϕ rotation.
FMPH	Frictional moment limiting value for ϕ rotation. If zero, friction is inactive for ϕ rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus ϕ rotation. See Figure 6.17. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on card 1.
EST	Elastic stiffness per unit radian for friction and stop angles for θ rotation. See Figure 6.17. If zero, friction and stop angles are inactive for θ rotation.
FMT	Frictional moment limiting value for θ rotation. If zero, friction is inactive for θ rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus θ rotation. See Figure 6.17. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on card 1.
ESPS	Elastic stiffness per unit radian for friction and stop angles for ψ rotation. See Figure 6.17. If zero, friction and stop angles are inactive for ψ rotation.
FMPS	Frictional moment limiting value for ψ rotation. If zero, friction is inactive for ψ rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	ψ rotation. See Figure 6.17. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on card 1.
NSAPH	Stop angle in degrees for negative ϕ rotation. Ignored if zero.
PSAPH	Stop angle in degrees for positive ϕ rotation. Ignored if zero.
NSAT	Stop angle in degrees for negative θ rotation. Ignored if zero.
PSAT	Stop angle in degrees for positive θ rotation. Ignored if zero.
NSAPS	Stop angle in degrees for negative ψ rotation. Ignored if zero.
PSAPS	Stop angle in degrees for positive ψ rotation. Ignored if zero.

Remarks:

After the stop angles are reached the torques increase linearly to resist further angular motion using the stiffness values on Card 3. Reasonable stiffness values have to be chosen. If the stiffness values are too low or zero, the stop will be violated.

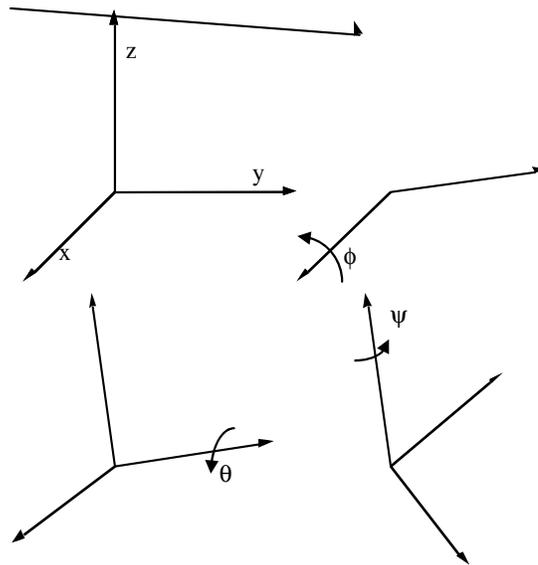


Figure 6.16. Definition of angles for the generalized joint stiffness. The magnitude of the angular rotations are limited by the stop angles defined on Card 4. If the initial local coordinate axes do not coincide, the angles, ϕ , θ , and ψ , will be initialized and torques will develop instantaneously based on the defined load curves.

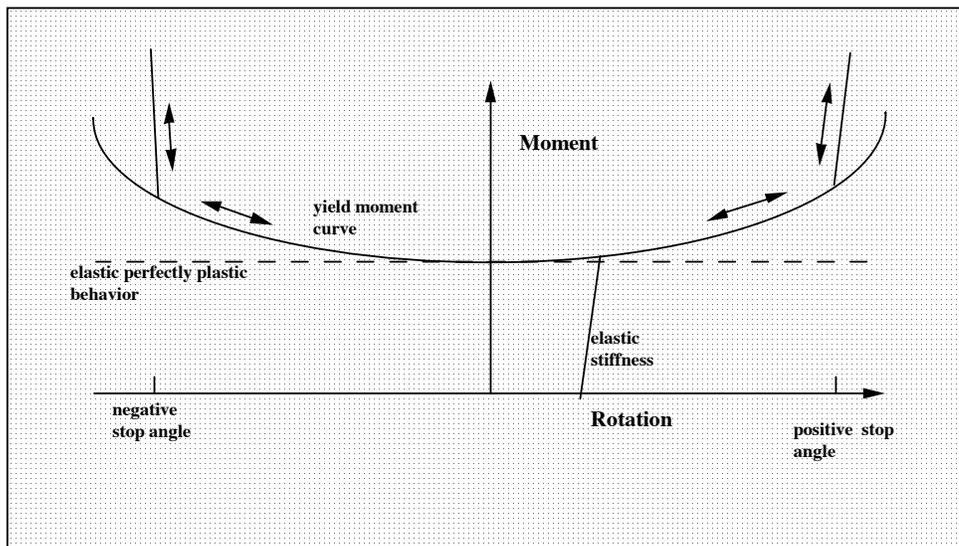


Figure 6.17. Frictional behavior is modeled by a plasticity model. Elastic behavior is obtained once the stop angles are reached. The same elastic stiffness is used to simulate sticking situations.

Card 2- 4 - Required for TRANSLATIONAL stiffness.

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDX	LCIDY	LCIDZ	DLCIDX	DLCIDY	DLCIDZ		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	none		

Card 3

Variable	ESX	FFX	ESY	FFY	ESZ	FFZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 4

Variable	NSDX	PSDX	NSDY	PSDY	NSDZ	PSDZ		
Type	F	F	F	F	F	F		
Default	not used							

VARIABLE

DESCRIPTION

LCIDX Load curve ID for x–force versus x-translational relative displacement between the origins of CIDA and CIBD based on the x-direction of CIBD. If zero, the applied force is set to 0.0. See *DEFINE_CURVE. See Figure 6.18.

LCIDY Load curve ID for y–force versus y-translational relative displacement between the origins of CIDA and CIBD based on the y-direction of CIBD. If zero, the applied force is set to 0.0. See *DEFINE_CURVE.

VARIABLE	DESCRIPTION
LCIDZ	Load curve ID for z–force versus z-translational relative displacement between the origins of CIDA and CIDB based on the z-direction of CIDB. If zero, the applied force is set to 0.0. See *DEFINE_CURVE.
DLCIDX	Load curve ID for x–damping force versus rate of x-translational displacement per unit time between the origins of CIDA and CIDB based on the x-direction of CIDB. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDY	Load curve ID for y–damping force versus rate of y-translational displacement per unit time between the origins of CIDA and CIDB based on the y-direction of CIDB. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDZ	Load curve ID for z–damping force versus rate of z-translational displacement per unit time between the origins of CIDA and CIDB based on the z-direction of CIDB. If zero, damping is not considered. See *DEFINE_CURVE.
ESX	Elastic stiffness for friction and stop displacement for x-translation. If zero, friction and stop angles are inactive for x-translation.
FFX	Frictional force limiting value for x-translation. If zero, friction is inactive for x-translation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve ID defining the yield force versus x-translation.
ESY	Elastic stiffness for friction and stop displacement for y-translation. If zero, friction and stop angles are inactive for y-translation.
FFY	Frictional force limiting value for y-translation. If zero, friction is inactive for y-translation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve ID defining the yield force versus y-translation.
ESZ	Elastic stiffness for friction and stop displacement for z-translation. If zero, friction and stop angles are inactive for z-translation..
FMZ	Frictional force limiting value for z-translation. If zero, friction is inactive for z-translation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve ID defining the yield force versus z-translation.

VARIABLE	DESCRIPTION
NSDX	Stop displacement for negative x-translation. Ignored if zero.
PSDX	Stop displacement for positive x-translation. Ignored if zero.
NSDY	Stop displacement for negative y-translation. Ignored if zero.
PSDY	Stop displacement for positive y-translation. Ignored if zero.
NSDZ	Stop displacement for negative z-translation. Ignored if zero.
PSDZ	Stop displacement for positive z-translation. Ignored if zero.

Remarks:

After the stop displacements are reached the force increases linearly to resist further translational motion using the stiffness values on Card 3. Reasonable stiffness values must be chosen. If the stiffness values are too low or zero, the stop will be violated.

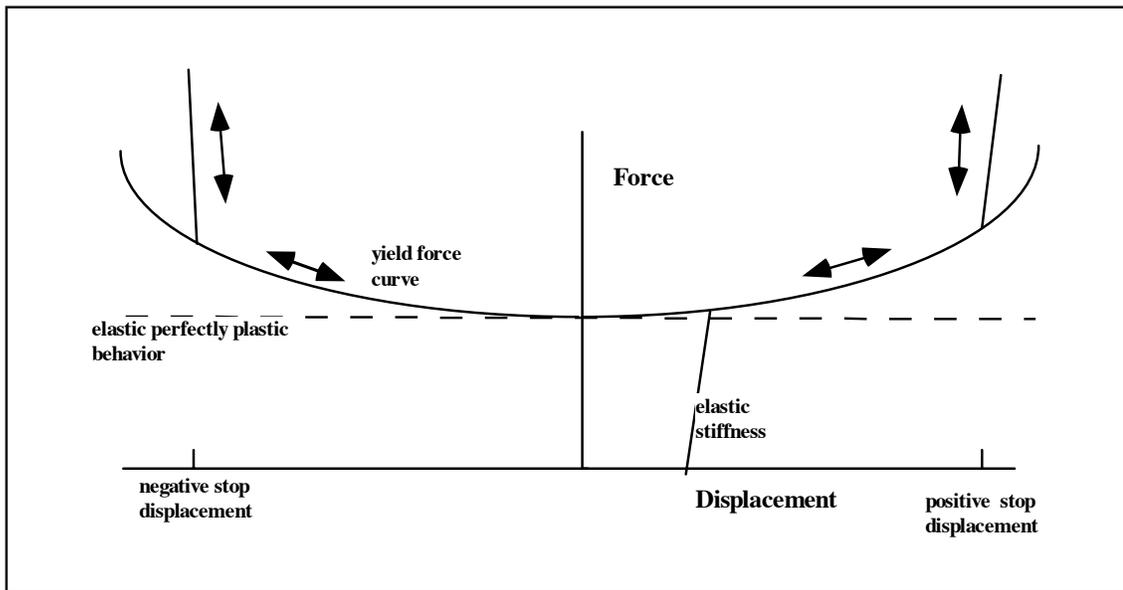


Figure 6.18. Frictional behavior is modeled by a plasticity model. Elastic behavior is obtained once the stop displacements are reached. The same elastic stiffness is used to simulate sticking situations.

*CONSTRAINED

*CONSTRAINED_JOINT_STIFFNESS

```
$$$  
$$$ *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED  
$$$  
$ Define a joint stiffness for the revolute joint described in  
$ *CONSTRAINED_JOINT_REVOLUTE  
$  
$ Attributes of the joint stiffness:  
$ - Used for defining a stop angle of 30 degrees rotation  
$ (i.e., the joint allows a positive rotation of 30 degrees and  
$ then imparts an elastic stiffness to prevent further rotation)  
$ - Define between rigid body A (part 1) and rigid body B (part 2)  
$ - Define a local coordinate system along the revolute axis  
$ on rigid body A - nodes 1, 2 and 3 (cid = 5). This is used to  
$ define the revolute angles phi (PH), theta (T), and psi (PS).  
$ - The elastic stiffness per unit radian for the stop angles  
$ are 100, 10, 10 for PH, T, and PS, respectively.  
$ - Values not specified are not used during the simulation.  
$  
*CONSTRAINED_JOINT_STIFFNESS_GENERALIZED  
$  
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8  
$      jsid      pida      pidb      cida      cidb  
$           1         1         2         5         5  
$  
$      lcidph     lcidt     lcidps     dlcidph     dlcidt     dlcidps  
$  
$      esph       fmps       est        fmt        esps       fmps  
$      100.0      10.0      10.0      10.0      10.0      10.0  
$  
$      nsaph      psaph      nsat       psat       nsaps      psaps  
$           30.0  
$  
$  
*DEFINE_COORDINATE_NODES  
$      cid       n1        n2        n3  
$           5         1         2         3  
$  
$$$  
$
```

***CONSTRAINED_LAGRANGE_IN_SOLID_{OPTION1}_{OPTION2}**

Purpose: This command provides the coupling mechanism for modeling Fluid-Structure Interaction (FSI). The structure can be constructed from Lagrangian shell and/or solid entities. The multi-material fluids are modeled by ALE formulation (see Remark 15).

Note: For RIGID slave PARTS a penalty coupling method (CTYPE=4) must be used, see parameter CTYPE below.

Available options for *OPTION1* include:

<BLANK>

EDGE

This option may be used to allow the coupling between the edge of a shell part or part set and one or more ALE multi-material groups (AMMG). It accounts for the shell thickness in the coupling calculation. The edge thickness is the same as the shell thickness. This option only works when the Lagrangian slave set is defined as a part or a part set ID. It will not work for a slave segment set. One application of this option is a simulation of a Lagrangian blade (a shell part) cutting through some ALE material.

Available options for *OPTION2* include:

<BLANK>

TITLE

To define a coupling (card) ID number and title for each coupling card. If a title is not defined LS-DYNA will automatically create an internal title for this coupling definition. The ID number can be used to delete coupling action in a restart input deck via the *DELETE_FSI card.

Card Format (I10, A70)

The following card is read if and only if the TITLE option is specified.

Optional	1	2
Variable	COUPID	TITLE
Type	I	A70

*CONSTRAINED

*CONSTRAINED_LAGRANGE_IN_SOLID

Card 1 is mandatory for all coupling definitions.

Card 1 1 2 3 4 5 6 7 8

Variable	SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
Type	I	I	I	I	I	I	I	I
Default	none	none	0	0	0	2	1	0

Card 2 is mandatory for all coupling definitions.

Card 2 1 2 3 4 5 6 7 8

Variable	START	END	PFAC	FRIC	FRCMIN	NORM	NORMTYP	DAMP
Type	F	F	F	F	F	I	I	F
Default	0	1.0E10	0.1	0.0	0.5	0	0	0.0

Card 3 is mandatory for all coupling definitions.

Card 3 1 2 3 4 5 6 7 8

Variable	CQ	HMIN	HMAX	ILEAK	PLEAK	LCIDPOR	NVENT	IBLOCK
Type	F	F	F	I	F	I	I	I
Default	0.0	none	none	0	0.1	none	0	0

Optional Card 4a may be defined with or without 4b. It is required for CTYPE 11 & 12. (define THKF if CTYPE=11).

Card 4a 1 2 3 4 5 6 7 8

Variable	IBOXID	IPENCHK	INTFORC	IALESOF	LAGMUL	PFACMM	THKF	
Type	I	I	I	I	F	I	F	
Default	0	0	0	0	0.0	0	0.0	

Optional Card 4b is required for CTYPE 11 & 12. If 4b is defined, 4a must be defined before 4b.

Card 4b 1 2 3 4 5 6 7 8

Variable	A1	B1	A2	B2	A3	B3		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Optional 4c card(s) defining venting geometry. It is repeated NVENT times (one line for defining each vent hole). It is defined only if NVENT > 0 in card 3. If either or both 4a and 4b are defined, they are defined before card(s) 4c.

Card 4c 1 2 3 4 5 6 7 8

Variable	VENTSID	VENTYP	VTCOEF	POPPRES	COEFLC			
Type	I	I	I	F	I			
Default	0	0	0	0.0	0			

*CONSTRAINED

*CONSTRAINED_LAGRANGE_IN_SOLID

VARIABLE	DESCRIPTION
COUPID	Coupling (card) ID number (I10). This ID can be used in a restart input deck to delete or reactivate this coupling action via the *DELETE_FSI card. If not defined, LSDYNA will assign an internal coupling ID based on the order of appearance in the input deck.
TITLE	A description of this coupling definition (A70).
SLAVE	Slave set ID defining a part, part set or segment set ID of the Lagrangian or slave structure (see *PART, *SET_PART or *SET_SEGMENT). See Remark 1.
MASTER	Master set ID defining a part or part set ID of the ALE or master solid elements (see *PART or *SET_PART, and see Remark 1).
SSTYP	Slave set type of "SLAVE" (see Remark 1): EQ.0: part set ID (PSID). EQ.1: part ID (PID). EQ.2: segment set ID (SGSID).
MSTYP	Master set type of "MASTER" (see Remark 1): EQ.0: part set ID (PSID). EQ.1: part ID (PID).
NQUAD	Number of coupling points distributed over each coupled Lagrangian surface segment. EQ.0: NQUAD will be set by default to 2, EQ.n: An nXn coupling points distribution over each Lagrangian segment is defined, EQ.-n: NQUAD is reset to a positive value. Coupling at nodes is obsolete.
CTYPE	Fluid-Structure coupling method: EQ.1: constrained acceleration. EQ.2: constrained acceleration and velocity (default, see Remark 3). EQ.3: constrained acceleration and velocity, normal direction only. EQ.4: penalty coupling for shell (with or without erosion) and solid elements (without erosion). EQ.5: penalty coupling allowing erosion in the Lagrangian entities (solid elements, see Remark 3). EQ.6: penalty coupling designed for airbag modeling which automatically controls the DIREC parameter internally. It is equivalent to setting {CTYPE=4; DIREC=1} for unfolded region; and {CTYPE=4; DIREC=2}; in folded region. For both cases: {ILEAK=2; FRCMIN=0.3}.

VARIABLE	DESCRIPTION
	<p>EQ.11: coupling designed to couple Lagrangian porous shell to ALE material. When this option is used, THKF, the 7th column parameter of optional card 4a and the first 2 parameters of optional card 4b must be defined. See *LOAD_BODY_POROUS and remark 14 below.</p> <p>EQ.12: coupling designed to couple Lagrangian porous solid to ALE material. When this option is used, A_i & B_i parameters of optional card 4b must be defined (card 4a must be defined but can be blank). See *LOAD_BODY_POROUS and Remark 14 below.</p>
DIREC	<p>Coupling direction (CTYPE 4 and 5, see Remark 4).</p> <p>EQ.1: normal direction, compression and tension (default)</p> <p>EQ.2: normal direction, compression only,</p> <p>EQ.3: all directions.</p>
MCOUP	<p>Multi-material option (CTYPE 4, 5, 6, 11 and 12, see Remark 5).</p> <p>EQ.0: couple with all multi-material groups,</p> <p>EQ.1: couple with material with highest density.</p> <p>EQ.-n: refers to a set ID of an ALE multi-material groups defined in *SET_MULTI-MATERIAL_GROUP card in which its set ID=n.</p>
START	<p>Start time for coupling.</p>
END	<p>End time for coupling.</p>
PFAC	<p>Penalty factor (CTYPE 4, 5 and 6). PFAC is a scale factor for scaling the estimated stiffness of the interacting (coupling) system. It is used to compute the coupling forces to be distributed on the slave and master parts.</p> <p>GT.0.0: Fraction of estimated critical stiffness.</p> <p>LT.0: -n: where n refers to a load curve ID. The curve defines the coupling pressure (y-axis) as a function of the penetration (x-axis) (See Remark 6).</p>
FRIC	<p>Coefficient of friction (used with DIREC 2 only).</p>
FRCMIN	<p>Minimum volume fraction of a coupled ALE multi-material group (AMMG) or fluid in a multi-material ALE element to activate coupling. Default value is 0.5. Reducing FRCMIN (typically, between 0.1 and 0.3) would turn on coupling earlier to prevent leakage in hypervelocity impact cases.</p>
NORM	<p>A flag indicating the rule for defining which side of the Lagrangian segment the fluid is supposed to be coupled to. By default (NORM=0) the fluid on the side pointed to by the Lagrangian segment normal (head-side) is coupled to. To couple to the fluid on the side not pointed to by the segment normals (tail-side), set NORM=1 (see Remark 7).</p> <p>EQ.0: Couple fluid to head-side of Lagrangian segment.</p> <p>EQ.1: Couple fluid to tail-side of Lagrangian segment.</p>

VARIABLE	DESCRIPTION
NORMTYP	Penalty coupling spring (or force) direction (DIREC 1 and 2): EQ.0: normal vectors are interpolated from nodal normals (default). EQ.1: normal vectors are interpolated from segment normals. This is sometimes a little more robust for sharp Lagrangian corners, and folds.
DAMP	Damping factor for penalty coupling. This is a coupling-damping scaling factor. Typically it may be between 0 and 1 (see Remark 8).
CQ	Heat transfer coefficient, C_q (see Remark 9).
HMIN	Minimum air gap in heat transfer, h_{\min} (see Remark 9).
HMAX	Maximum air gap in heat transfer, h_{\max} . There is no heat transfer above this value (see Remark 9).
ILEAK	Coupling leakage control flag (Remark 10): EQ.0: none (default), EQ.1: weak, leakage control is turned off if the penetrating volfrac > FRCMIN+0.1. EQ.2: strong, with improved energy consideration. Leakage control is turned off if the penetrating volfrac > FRCMIN+0.3.
PLEAK	Leakage control penalty factor, $0 < \text{PLEAK} < 0.2$ is recommended. This factor influences the additional coupling force magnitude to prevent leakage. It is conceptually similar to PFAC. Almost always, the default value (0.1) is adequate.
LCIDPOR	If this is a positive integer: A load curve ID (LCID) defining porous flow through coupling segment: Abscissa = $x = (P_{\text{up}} - P_{\text{down}})$ Ordinate = $y =$ relative porous fluid velocity Where P_{up} and P_{down} are, respectively, the upstream and downstream pressures across of the porous coupling segment. The relative porous velocity is the ALE fluid velocity relative to the moving Lagrangian segment. This experimental data curve must be provided by the user. If LCIDPOR is a negative integer: The porous flow is controlled by the parameters FLC, FAC, ELA under *MAT_FABRIC card. CAUTION: The pressure under the FAC load curve is “absolute upstream pressure” (see Remark 11). Abscissa = $x =$ absolute upstream pressure Ordinate = $y =$ relative porous fluid velocity

VARIABLE	DESCRIPTION
NVENT	The number of vent surface areas to be defined. Each venting flow surface is represented by one or more Lagrangian segments (or surfaces). For airbag application, this may be referred to as “isentropic” venting where the isentropic flow equation is used to compute the mass flow rate across an area given P_{up}/P_{down} ratio. If NVENT > 0, there will be NVENT number of lines required (optional cards 4c). Each line defines the geometrical and flow information for each vent surface: VENTSID, VNTYPE, VTCOEF, POPPRES, COEFLC. The vented mass will simply be deleted from the system and cannot be visualized as in the case of physical venting (see Remark 12).
IBLOCK	Flag (1=ON or 0=OFF) to control the venting (or porous) flow blockage due to Lagrangian contact during ALE computation. The venting definition is defined in this command. However, the venting flow may be defined via either the LCIDPOR parameter in this command or via the *MAT_FABRIC parameters (FLC, FAC, ELA). However, note that FVOPT (blocking) parameter under *MAT_FABRIC applies only to CV computation.
IBOXID	A box ID defining a box region in space in which ALE coupling is activated. At time=0.0, the number of Lagrangian segments inside this box is remembered. In subsequent coupling computation steps, there is no need to search for the Lagrangian segments again.
IPENCHK	Initial penetration check flag (only for CTYPE=4, Remark 13): EQ.0: Do not check for initial penetration. EQ.1: Check and save initial ALE material penetration across a Lagrangian surface (d_0), but do not activate coupling at $t=0$. In subsequent steps ($t>0$) the actual penetration is computed as follows actual penetration = total penetration – initial penetration $d_a = d_T - d_0$
INTFORC	A flag to turn on or off (0=OFF or 1=ON) the output of ALE coupling pressure and forces on the slave Lagrangian segments (or surfaces). Note that the coupling pressures and forces are computed based on the ALE fluid penetrations and coupling stiffness of the system. When (1) INTFORC=1 and (2) a *DATABASE_BINARY_FSIFOR (DBF) card is defined, LS-DYNA writes out the segment coupling pressure and forces to the binary interface force file for contour plotting. This interface force file is activated by executing ls971 as follows (3): ls971 i=inputfilename.k ... h=interfaceforcefilename The time interval between output is defined by “dt” in the DBF card. To plot the binary data in this file: lsprepost interfaceforcefilename.

VARIABLE	DESCRIPTION
IALESOF	<p>An integer flag to turn ON/OFF a supplemental Lagrange multiplier FSI constraint which provides a coupling force in addition to the basic penalty coupling contribution. This is a hybrid coupling method.</p> <p>EQ.0: OFF (default). EQ.1: Turn ON the hybrid Lagrange-multiplier method. LAGMUL multiplier factor is read.</p>
LAGMUL	<p>A Lagrange multiplier factor with a range between 0.0 and 0.05 may be defined. A typical value may be 0.01. This should never be greater than 0.1.</p> <p>EQ.0: OFF (default). GT.0: Turn ON the Lagrange-multiplier method and use LAGMUL as a coefficient for scaling the penalty factor.</p>
PFACMM	<p>Mass-based penalty stiffness factor computational options. This works in conjunction with PFAC=constant (not a load curve). The coupling penalty stiffness (CPS) is computed based on an estimated effective coupling mass.</p> <p>EQ.0: $CPS \propto PFAC \cdot \min(m_{slave}, m_{master})$, default. EQ.1: $CPS \propto PFAC \cdot \max(m_{slave}, m_{master})$. EQ.2: $CPS \propto PFAC \cdot \sqrt{m_{slave} m_{master}}$, geometric-mean of the masses. EQ.3: $CPS \propto PFAC \cdot K_{Lagrangian}$ where K is the bulk modulus of the slave or Lagrangian part</p>
THKF	<p>(For all CTYPE choices except 11) A flag to account for the coupling thickness of the Lagrangian shell (slave) part.</p> <p>LT.0: Use positive value of THKF for coupling segment thickness. EQ.0: Do not consider coupling segment thickness. GT.0: Coupling segment thickness scale factor.</p> <p>For CTYPE=11 case (see Remark 14): This thickness is required for volume calculation.</p> <p>GT.0: (Fabric) Thickness scale factor. The base shell thickness is taken from the *PART definition. LT.0: User-defined (Fabric) thickness. The fabric thickness is set to THKF .</p>
A1	<p>Viscous coefficient for the porous flow Ergun equation (see Remark 14). For CTYPE=11, A1 = A_n = coefficient for normal-to-segment direction. For CTYPE=12: A1 = A_x = coefficient for global X-direction.</p>
B1	<p>Inertial coefficient for the porous flow Ergun equation (see Remark 14). For CTYPE=11, B1 = B_n = coefficient for normal-to-segment direction. For CTYPE=12: B1 = B_x = coefficient for global X-direction.</p>

VARIABLE	DESCRIPTION
A2	Viscous coefficient for the porous flow Ergun equation (see Remark 14). For CTYPE=12: A2 = A_y = coefficient for global Y-direction.
B2	Inertial coefficient for the porous flow Ergun equation (see Remark 14). For CTYPE=12: B2 = B_y = coefficient for global Y-direction.
A3	Viscous coefficient for the porous flow Ergun equation (see Remark 14). For CTYPE=12: A3 = A_z = coefficient for global Z-direction.
B3	Inertial coefficient for the porous flow Ergun equation (see Remark 14). For CTYPE=12: B3 = B_z = coefficient for global Z-direction.
VENTSID	Set ID of the vent hole shape.
VENTYYP	Vent surface area set ID type: EQ.0: Part set ID (PSID). EQ.1: Part ID (PID). EQ.2: Segment set ID (SGSID).
VTCOEF	Flow coefficient for each vent surface area.
POPPRES	Venting pop pressure limit. If the pressure inside the airbag is lower than this pressure, then nothing is vented. Only when the pressure inside the airbag is greater than POPPRES that venting can begin.
COEFLC	A time-dependent multiplier load curve for correcting the vent flow coefficient, with values ranging from 0.0 to 1.0.

Remarks:

1. In order for a fluid-structure interaction (FSI) to occur, a Lagrangian (structure or slave) mesh must spatially overlap with an ALE (fluid or master) mesh. Each mesh should be defined with independent node IDs. LS-DYNA searches for the spatial intersection of between the Lagrangian and ALE meshes. Where the meshes overlap, there is a possibility that interaction may occur. In general, SLAVE, MASTER, SSTYP and MSTYPE are required definitions for specifying overlapping-domains coupling search.
2. The number of coupling points, NQUAD, is distributed over the surface of each Lagrangian segment. Generally, 2 or 3 coupling points per each Eulerian/ALE element width is adequate. Consequently, the appropriate NQUAD values must be estimated based on the relative resolutions between the Lagrangian and ALE meshes. Consider Case 1 where 1 Lagrangian shell element spans, say, 2 ALE elements. Then NQUAD for each Lagrangian segment should be 4 or 6. Consider Case 2 where 2 or 3 Lagrangian segments span 1 ALE element, then maybe NQUAD=1 would be adequate. If either mesh compresses or expands during the interaction, the number of coupling points per ALE element will also change. The user must account for this and try to maintain at least 2 coupling points per each ALE element side length during the whole process to

prevent leakage. Too many coupling points can result in instability, and not enough can result in leakage.

3. CTYPE=2 is sometimes used to couple, via constraint method (energy not conserved), Lagrangian beam nodes to ALE solid, for example, in the modeling of rebar in concrete, or tire cords in rubber. The slave set is coupled to the master set. Constraint based method does not try to conserve energy and is seldom used. Penalty approach is the current method of choice. For better accuracy, when using the penalty method, it is better to couple to a specific set of AMMGs (set MCOUP to a negative integer, see MCOUP).
4. DIREC=2 may be generally a more stable and robust choice for coupling direction. However a choice of when to activate coupling should be made based on the physics of the problem. DIREC=1 couples under both tension and compression. This is sometimes useful as in the case of suddenly accelerating liquid container. DIREC=3 is rarely applicable to real physics (because it simulates an extremely sticky fluid).
5. When MCOUP is a negative integer, say for example MCOUP= -123, then an ALE multi-material set-ID (AMMSID) of 123 must exist. This is an ID defined by a *SET_MULTI-MATERIAL_GROUP_LIST card. This generally seems to be a better approach to couple to a specific set of AMMGs, and have a clearly defined fluid interface interacting with a Lagrangian surface. That way, any leakage may be visualized and the penalty force can be computed more precisely.
6. The user can usually start with PFAC=default (0.1). If leakage occurs, leakage control may be turned on, ILEAK=2.

The next thing to try may be using a load curve for PFAC. When PFAC is a negative integer, for example PFAC= -321, then a load curve with LCID=321 must exist via a *DEFINE_CURVE card. This choice allows the application of an estimated coupling pressure (y-axis) given a penetration distance (x-axis). The curve consists of {0,0} as the first point and {maximum allowable penetration (MAP), estimated maximum coupling pressure (EMCP)} as a second point. MAP may be a small penetration with respect to the minimum ALE element width (maybe 10% or less). EMCP can be estimated from a maximum fluid pressure observed from a previous run when leakage first occurs. This curve may be scaled to vary the stiffness of the coupling spring. The approach is to gradually increase the coupling stiffness until leakage stops. The best coupling stiffness is one which provides just enough force to prevent leakage and not more.

A 3rd approach is to try PFACMM=3 on optional card 4a (with PFAC=constant).

Consider a coupling between a Lagrangian airbag with the ALE inflator gas, a penetration of 1.0E-3 m of the gas across the bag surface will trigger a coupling pressure (say, about 4 atm or 405300 Pascals). The coupling pressure for arbitrary penetration is scaled from this curve.

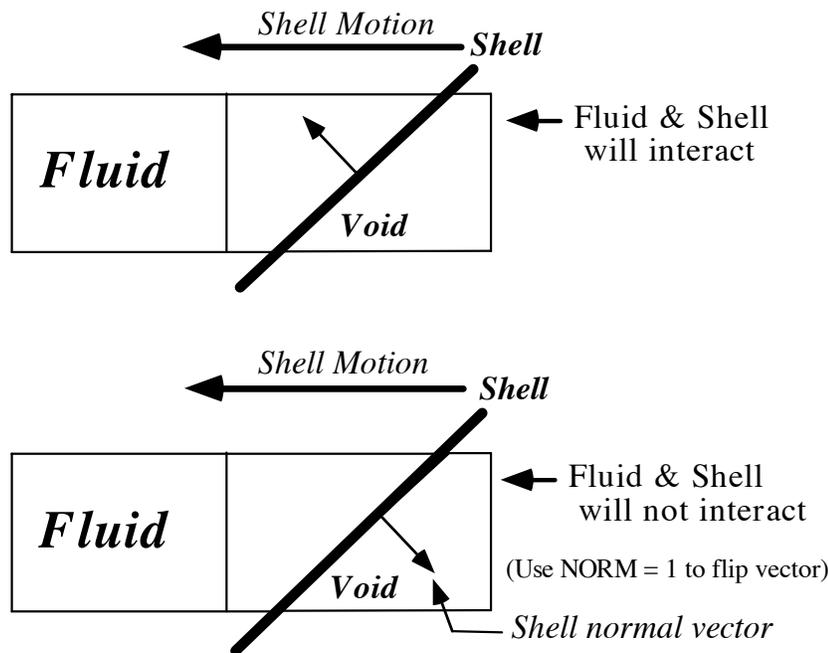
This example demonstrates the usage of both MCOUP and PFAC when they are negative integers.

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ PID 21 = ALE inflator gas; PID 22 = air mesh surrounding the airbag
*ALE_MULTI-MATERIAL_GROUP
$   SID   IDTYPE
    21     1
    22     1
$ ALEMMGID = 1 <= PID 21 <= see 1st line of the *ALE_MULTI-MATERIAL_GROUP card.
*SET_MULTI-MATERIAL_GROUP_LIST
$   AMMSID
    123
$ ALEMMGID
    1
$ Lagrangian = slave = PSID 1 contains all airbag parts
*SET_PART
$   SID   DA1   DA2   DA3   DA4
    1     0.    0.    0.    0.
$   PID1  PID2  PID3  PID4  PID5  PID6  PID7  PID8
    3     4     5     6     7     8     9     0
$ Eulerian = master = PSID 11 contains all fluid mesh (geometrical space to search).
*SET_PART LIST
    11
    2
*CONSTRAINED_LAGRANGE_IN_SOLID
$   SLAVE  MASTER  SSTYP  MSTYP  NQUAD  CTYPE  DIREC  MCOUP
    1      11      0       0       4       4       2      -123
$   START  END     PFAC   FRIC   FRCMIN  NORM
    0.0    0.0    -321   0.00  0.3     1
$   CQ     HMIN   HMAX   ILEAK  PLEAK  VLK_PLCID
    0      0       0       2     0.10
$ We should couple to only the inflator gas → MCOUP = -123
$ fluid penetration ~ 1 mm <====> Pmax = 405300 pascal ~ 4 atm
*DEFINE_CURVE
    321
                0.0000                0.0
                1.0e-3                405300.0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

- The normal vectors (NV) of a Lagrangian shell part are defined by the order of the nodes in *ELEMENT definitions, via the right hand rule, and for a segment set, the order of nodes defined in *SET_SEGMENT. Let the side pointed to by NV be “positive”. The penalty method measure penetration as the distance the ALE fluid penetrates from the positive side to the negative side of the Lagrangian segment. Only fluid on the positive side will be “seen” and coupled to. Hence, all normal vectors of the Lagrangian segments should point uniformly toward the ALE fluid(s), AMMGs, to be coupled to. If NV point uniformly away from the fluid, coupling is not activated. In this case, coupling can be activated by setting NORM=1. Sometimes a shell part or mesh is generated such that its normal vectors do not point uniformly in a consistent direction (all toward the inside or outside of a container, etc.) The user should always check for the normal vectors of any Lagrangian shell part interacting with any fluid. The NORM parameter may be used to flip the normal direction of all the segments included in the Lagrangian slave set.



8. The user-input coupling-damping factor (DAMP) is used to scale down the critical-damping force (\sim damper constant*velocity). For a mass-to-rigid-wall system connected by a parallel-spring-damper connector, we can obtain solution for a critically-damped case. DAMP is a factor for scaling down the amount of damping, with DAMP=1 being a critically-damped case.
9. The method used is similar to that done by *CONTACT_..._THERMAL_... card, except radiation heat transfer is not considered. A gap (l) is assumed to exist between the 2 materials undergoing heat transfer (one is Lagrangian and the other ALE). The convection heat transfer in the gap is assumed to approach simple conduction across the medium in the gap.

$$q = \kappa \frac{dT}{dx} \sim h\Delta T \Rightarrow$$

$$h \sim \frac{\kappa}{l}$$

The heat flux is typically defined as an energy transfer rate per unit area, $q \sim \frac{[J/s]}{m^2}$. κ is the thermal conductivity of the material in the gap, h , the equivalent convection heat transfer coefficient, and ΔT , the temperature difference between the master and slave sides. There are 3 possible scenarios:

- (a) $l > l_{max} \rightarrow$ No heat transfer.

$$(b) \ l_{\min} \leq l \leq l_{\max} \rightarrow h \sim \frac{\kappa}{\max(l_{\min}, l)} .$$

$$(c) \ l < l_{\min} \rightarrow h \sim \frac{\kappa}{l_{\min}} .$$

CQ (κ), **HMIN** (l_{\min}), **HMAX** (l_{\max}) are defined for this heat transfer estimation.

10. In general, a coupling force stopping “fluid” leakage across a Lagrangian surface should come predominantly from the (penalty) coupling force. Leakage control force should be of secondary effect and should not be bigger than that from the main penalty coupling action. *DATABASE_FSI command output (dbfsi) allows for the monitoring of both the coupling forces and the leakage control force contribution. It may be used as a debugging or fine-tuning tool for coupling design. ILEAK=2 has a slightly more accurate energy accounting algorithm, thus is better for airbag applications. Leakage control should only be turned on when coupling to a specific AMMG (MCOUP as a negative integer) with its fluid interface clearly defined and tracked via the *ALE_MULTI-MATERIAL_GROUP card.
11. There are currently 2 methods to model porous flow across a Lagrangian shell structure. Both methods involve defining an empirical data curve of relative porous gas velocity as a function of system pressure. However the pressure definitions are slightly different depending on the choice of parameter defined:
 - a) Via LCIDPOR parameter under *CONSTRAINED_LAGRANGE_IN_SOLID (CLIS). If this option is used the data curve contains ($P_{\text{upstream}} - P_{\text{downstream}}$) in the x-axis of the curve.
 - b) Via *MAT_FABRIC’s FLC, FAC, ELA parameters. If LCIDPOR is negative, and FAC defines a load curve, then this data curve contains absolute upstream pressure (not pressure difference) in the x-axis.

When *AIRBAG_ALE is used, it assumes that absolute upstream P is given in the curve defined by FAC under *MAT_FABRIC. During CV phase it uses that. In ALE phase, LS-DYNA automatically subtracts 1 atmospheric pressure from the given pressure in the FAC curve. Thus giving it a gauge pressure for the porous coupling calculation. The amount of accumulated mass flowing across a porous Lagrangian surface may be tracked via the *DATABASE_FSI card (“pleak” parameter in the “dbfsi” ASCII output file).

12. There are 2 methods to model (airbag) venting. The accumulated mass output of both may be tracked via the *DATABASE_FSI card (“pleak” parameter in the “dbfsi” ASCII output file).
 - a) In isentropic venting, (define NVENT on card 3) the flow crossing the vent hole surface is estimated from isentropic equation. All airbag shell normal vectors should point uniformly in the same direction (typically, inward). The shell elements for the

vent holes, included in the Lagrangian coupling set, should also point in the same direction as the airbag (i.e., inward).

- b) In physical venting, there can be either physical holes in the Lagrangian structure (airbag), or shell parts covering the holes would have their normal vectors pointing outward. Either way, there is no coupling force to stop fluid leakage. It is recommended that *ALE_FSI_SWITCH_MMG_ID (AFSM) be used to switch the AMMG ID of the vented gas so that the vented gas outside the bag does not cause leakage (when the same AMMG is present on both sides of the same Lagrangian shell surface, penalty coupling can break down). The shell parts representing the vent holes may be either (i) excluded from the Lagrangian coupling set, or (ii) if included, they should have their normals pointing in opposite direction with respect to the rest of the airbag (typically outward since the rest of the airbag have their normals pointing inward).
13. Typically, penetration check (IPENCHK) should only be used if there is high coupling force applied at $t=0$. For example, consider a Lagrangian container, filled with non-gaseous fluid (i.e. ALE liquid or solid) via the *INITIAL_VOLUME_FRACTON_GEOMETRY command. Sometimes due to mesh resolution or complex container geometry, there is initial penetration of the fluid across the container surface. This can give rise to a sharp and immediate coupling force on the fluid at $t=0$. Turning on IPENCHK may help eliminate this spike in coupling force.
14. For shell, CTYPE=11, the Ergun-type empirical porous flow equation is applied to the normal flow direction across the porous surface. The pressure gradient along the segment normal direction is

$$\frac{dP}{dx_n} = A_n(\varepsilon, \mu)V_n + B_n(\varepsilon, \rho)|V_n|V_n$$

where the subscript “n” refers to the direction normal to the porous Lagrangian shell surface.

V_n is the relative normal-to-porous-shell-surface fluid velocity component.

$A_n(\varepsilon, \mu) = A_1(\varepsilon, \mu)$ is a viscous coefficient of the Ergun-type porous flow equation.

As applied here it should contain the fluid dynamic viscosity (μ) and shell porosity (ε) information.

$B_n(\varepsilon, \rho) = B_1(\varepsilon, \rho)$ is an inertial coefficient of the Ergun-type porous flow equation.

As applied here it should contain the fluid density (ρ) and shell porosity (ε) information.

The force increment applied per segment is

$$F_n = \frac{dp}{dx_n} \cdot t \times S \text{ where}$$

S is the segment surface area.

t is the shell thickness (THKF).

$A1 (A_i(\varepsilon, \mu))$, $B1 (B_i(\varepsilon, \mu))$ and $THKF (t)$ are required input for porous shell coupling.

For porous solid, $CTYPE=12$, the pressure gradient along each global direction (i) maybe computed similarly.

$$\frac{dP}{dx_i} = A_i(\varepsilon, \mu)V_i + B_i(\varepsilon, \rho)|V_i|V_i$$

$$i = 1, 2, 3$$

V_i is the relative fluid velocity component through the porous solid in the 3 global directions.

$A_i(\varepsilon, \mu)$ is a viscous coefficient of the Ergun-type porous flow equation in the i^{th} direction. As applied here it should contain the fluid dynamic viscosity (μ) and shell porosity (ε) information.

$B_i(\varepsilon, \rho)$ is an inertial coefficient of the Ergun-type porous flow equation in the i^{th} direction. As applied here it should contain the fluid density (ρ) and solid porosity (ε) information.

$A_i (A_i(\varepsilon, \mu))$, $B_i (B_i(\varepsilon, \mu))$ are required input for porous solid coupling.

Currently, only constant porosity structures are considered. See also *LOAD_BODY_POROUS.

15. Due to the complexity of this card, some comments on simple, efficient and robust coupling approach are given here. These are not rigid guidelines, but simply some experience-based observations.

The term “fluid”, in the Fluid-Structure Interaction (FSI), refers to materials with ALE element formulation, not indicating the phase (solid, liquid or gas) of those materials. In fact, solid, liquid and gas can all be modeled by the ALE formulation. The term “structure” refers to materials with Lagrangian element formulation.

In general, penalty coupling ($CTYPE$ 4 & 5) is recommended, and $MCOUP$ =negative integer is a better choice to define a specific ALE multi-material group (AMMG) to be coupled to the Lagrangian surface. At the minimum, all parameters on card 1 are to be specified, and the default values for most are good starting choices (except $MCOUP$).

If there is leakage, $PFAC$, $FRCMIN$, $NORMTYPE$ and $ILEAK$ are the 4 parameters that can be adjusted.

*CONSTRAINED

*CONSTRAINED_LAGRANGE_IN_SOLID

For hard structure (steel) and very compressible fluid (air), PFAC may be set to 0.1 (or higher). PFAC=constant value.

Next, keeping PFAC=constant and set PFACMM=3 (optional card 4a). This option scales the penalty factor by the bulk modulus of the Lagrangian structure. This new approach has also shown to be effective for some airbag application.

The next approach may be switching from constant PFAC to a load curve approach (i.e. PFAC=load curve, and PFACMM=0). By looking at the pressure in the system near leakage original location, we can get a feel for the pressure required to stop it.

If leakage persists after some iterations on the coupling force controls, one can subsequently try to set ILEAK=2 in combination with the other controls to prevent leakage.

If the modifications fail to stop the leakage, maybe the meshes have to be redesigned to allow better interactions between the Lagrangian and Ale materials.

In the example below, the underlined parameters are usually defined parameters. A full card definition is shown for reference.

```
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*CONSTRAINED_LAGRANGE_IN_SOLID
$  SLAVE  MASTER  SSTYP  MSTYP  NQUAD  CTYPE  DIREC  MCOUP
$    1      11      0      0      4      4      2      -123
$  START  END      PFAC  FRIC  FRCMIN  NORM  NORMTYPE  DAMP
$    0.0   0.0    0.1   0.00  0.3     0     0      0.0
$    CQ    HMIN    HMAX  ILEAK  PLEAK  LCIDPOR  NVENT  IBLOCK
$    0     0     0     0     0.0    0     0     0
$4A IBOXID IPENCHK INTFORC IALESOF LAGMUL  PFACMM  THKF
$    0     0     0     0     0     0     0
$4B  A1     B1     A2     B2     A3     B3
$    0.0   0.0   0.0   0.0   0.0   0.0
$4C VNTSID  VENTYPE  VENTCOEF  POPPRES  COEFLCID  (STYPE:0=PSID;1=PID;2=SGSID)
$    0     0     0     0.0   0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
```

***CONSTRAINED_LINEAR_GLOBAL**

Purpose: Define linear constraint equations between displacements and rotations, which can be defined in global coordinate systems. For a newer and for a more general constraint see *CONSTRAINED_INTERPOLATION

Card 1 - Required

Card 1 1 2 3 4 5 6 7 8

Variable	LCID							
Type	I							
Default	none							

Card 2 - Define one card for each constrained degree-of-freedom. Input is terminated when a "*" card is found.

Card 2 1 2 3 4 5 6 7 8

Variable	NID	DOF	COEF					
Type	I	I	I					
Default	none	0	0					
Remark	1							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCID	Linear constraint definition ID. This ID can be used to identify a set to which this constraint is a member.
NID	Node ID
DOF	Degree of freedom in the global coordinate system; EQ.1: displacement along global x-direction EQ.2: displacement along global y-direction EQ.3: displacement along global z-direction EQ.4: global rotation about global x-axis

VARIABLE	DESCRIPTION
	EQ.5: global rotation about global y-axis EQ.6: global rotation about global z-axis
COEF	Nonzero coefficient, C_k

Remarks:

Nodes of a nodal constraint equation cannot be members of another constraint equation or constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body; i.e. nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also care must be taken to ensure that single point constraints applied to nodes in a constraint equation do not conflict with the constraint sets constrained degrees-of-freedom.

In this section linear constraint equations of the form:

$$\sum_{k=1}^n C_k u_k = C_0$$

can be defined, where u_k are the displacements and C_k are user defined coefficients. Unless LS-DYNA is initialized by linking to an implicit code to satisfy this equation at the beginning of the calculation, the constant C_0 is assumed to be zero. The first constrained degree-of-freedom is eliminated from the equations-of-motion:

$$u_1 = C_0 - \sum_{k=2}^n \frac{C_k}{C_1} u_k$$

Its velocities and accelerations are given by

$$\dot{u}_1 = - \sum_{k=2}^n \frac{C_k}{C_1} \dot{u}_k$$

$$\ddot{u}_1 = - \sum_{k=2}^n \frac{C_k}{C_1} \ddot{u}_k,$$

respectively. In the implementation a transformation matrix, \underline{L} , is constructed relating the unconstrained, \underline{u} , and constrained, \underline{u}_c , degrees-of-freedom. The constrained accelerations used in the above equation are given by:

$$\ddot{\underline{u}}_c = [\underline{L}^T \underline{M} \underline{L}]^{-1} \underline{L}^T \underline{F}$$

where \underline{M} is the Diagonal lumped mass matrix and \underline{F} is the right hand side force vector. This requires the inversion of the condensed mass matrix which is equal in size to the number of constrained degrees-of-freedom minus one.

```
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
$$$ *CONSTRAINED_LINEAR_GLOBAL  
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
$ Constrain nodes 40 and 42 to move identically in the z-direction.  
$  
$ When the linear constraint equation is applied, it goes like this:  
$  
$   0 = C40uz40 + C42uz42  
$  
$   = uz40 - uz42  
$  
$   uz40 = uz42  
$  
$ where,  
$   C40 = 1.00 coefficient for node 40  
$   C42 = -1.00 coefficient for node 42  
$   uz40 = displacement of node 40 in z-direction  
$   uz42 = displacement of node 42 in z-direction  
$  
$  
$ *CONSTRAINED_LINEAR  
$  
$ . . . > . . . 1 . . . > . . . 2 . . . > . . . 3 . . . > . . . 4 . . . > . . . 5 . . . > . . . 6 . . . > . . . 7 . . . > . . .  
$ .8  
$     i  
$     id  
$     2  
$  
$      nid      dof      coef  
$      40         3      1.00  
$      42         3     -1.00  
$  
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$
```

*CONSTRAINED

*CONSTRAINED_LINEAR_LOCAL

*CONSTRAINED_LINEAR_LOCAL

Purpose: Define linear constraint equations between displacements and rotations, which can be defined in a local coordinate system. Each node may have a unique coordinate ID.

Card 1 - Required

Card 1 1 2 3 4 5 6 7 8

Variable	LCID							
Type	I							
Default	none							

Card 2 - Define one card for each constrained degree-of-freedom. Input is terminated when a "*" card is found.

Card 2 1 2 3 4 5 6 7 8

Variable	NID	DOF	CID	COEF				
Type	I	I	I	I				
Default	none	0	0	0				
Remark	1							

VARIABLE

DESCRIPTION

LCID	LCID for linear constraint definition. This ID can be used to identify a set to which this constraint is a member.
NID	Node ID
DOF	Degree of freedom in the local coordinate system; EQ.1: displacement along local x-direction EQ.2: displacement along local y-direction EQ.3: displacement along local z-direction EQ.4: local rotation about local x-axis EQ.5: local rotation about local y-axis EQ.6: local rotation about local z-axis

VARIABLE	DESCRIPTION
CID	Local coordinate system ID number. If the number is zero, the global coordinate system is used.
COEF	Nonzero coefficient, C_k

Remarks:

In this section linear constraint equations of the form:

$$\sum_{k=1}^n C_k u_k^L = C_0$$

can be defined, where u_k^L are the displacements in the local coordinate systems and C_k are user defined coefficients. Unless LS-DYNA is initialized by linking to an implicit code to satisfy this equation at the beginning of the calculation, the constant C_0 is assumed to be zero. The first constrained degree-of-freedom is eliminated from the equations-of-motion:

$$u_1^L = C_0 - \sum_{k=2}^n \frac{C_k}{C_1} u_k^L$$

Its velocities and accelerations are given by

$$\dot{u}_1^L = -\sum_{k=2}^n \frac{C_k}{C_1} \dot{u}_k^L$$

$$\ddot{u}_1^L = -\sum_{k=2}^n \frac{C_k}{C_1} \ddot{u}_k^L$$

respectively. The local displacements are calculated every time step using the local coordinate systems defined by the user. More than one degree of freedom for a node can be constrained by specifying a card for each degree of freedom.

Nodes of a nodal constraint equation cannot be members of another constraint equation or constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body; i.e. nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also care must be taken to ensure that single point constraints applied to nodes in a constraint equation do not conflict with the constraint sets constrained degrees-of-freedom.

*CONSTRAINED

*CONSTRAINED_LOCAL

*CONSTRAINED_LOCAL

Purpose: Define a boundary constraint in a local coordinate system.

Card	1	2	3	4	5	6	7	8
Variable	TC	RC	CIR	X	Y	Z	CID	
Type	1	1	1	F	F	F	1	
Default	0	0	0	0	0	0	none	

VARIABLE

DESCRIPTION

TC Translational Constraint:
EQ.1: constrained x translation,
EQ.2: constrained y translation,
EQ.3: constrained z translation,
EQ.4: constrained x and y translations,
EQ.5: constrained y and z translations,
EQ.6: constrained x and z translations,
EQ.7: constrained x, y, and translations.

RC Rotational Constraint:
EQ.1: constrained x-rotation,
EQ.2: constrained y-rotation,
EQ.3: constrained z-rotation,
EQ.4: constrained x and y rotations,
EQ.5: constrained y and z rotations,
EQ.6: constrained z and x rotations,
EQ.7: constrained x, y, and z rotations.

DIR Direction of normal
EQ.1: local x,
EQ.2: local y,
EQ.3: local z.

X x-offset coordinate

Y y-offset coordinate

Z z-offset coordinate

CID Coordinate ID defining the orientation of the wall.

Remarks:

Nodes within a mesh-size-dependent tolerance are constrained on a local plane. This option is recommended for use with r-method adaptive remeshing where nodal constraints are lost during the remeshing phase.

*CONSTRAINED

*CONSTRAINED_NODAL_RIGID_BODY

*CONSTRAINED_NODAL_RIGID_BODY_{OPTION}_{OPTION}_{OPTION}

Available options include:

<BLANK>

SPC

INERTIA

TITLE

If the center of mass is constrained use the **SPC** option. If the inertial properties are defined rather than computed use the **INERTIA** option. A description for the nodal rigid body can be defined with the **TITLE** option.

Purpose: Define a nodal rigid body. This is a rigid body which consists of the defined nodes. If the **INERTIA** option is not used, then the inertia tensor is computed from the nodal masses. Arbitrary motion of this rigid body is allowed. If the **INERTIA** option is used, constant translational and rotational velocities can be defined in a global or local coordinate system.

The first node in the nodal rigid body definition is treated as the master for the case where **DRFLAG** and **RRFLAG** are nonzero. The first node always has six degrees-of-freedom. The release conditions applied in the global system are sometimes convenient in small displacement linear analysis, but, otherwise, are not recommended. It is strongly recommended, especially for implicit calculations, that release conditions are only used for a two noded nodal rigid body.

Card Format:

Card 1 is required.

Cards 2 - 4 are required for the **INERTIA** option.

Card 5 is required if a local coordinate system is used to specify the inertia tensor when the **INERTIA** option is used.

Remarks:

1. Unlike the ***CONSTRAINED_NODE_SET** which permits only constraints on translational motion, here the equations of rigid body dynamics are used to update the motion of the nodes and therefore rotations of the nodal sets are admissible. Mass properties are determined from the nodal masses and coordinates. Inertial properties are defined if and only if the **INERTIA** option is specified.

The following card is read if and only if the **TITLE option is specified.**

Optional

Variable	TITLE
Type	A80

CONSTRAINED_NODAL_RIGID_BODY**CONSTRAINED**

Card 1 1 2 3 4 5 6 7 8

Variable	PID	CID	NSID	PNODE	IPRT	DRFLAG	RRFLAG	
Type	I	I	I	I	I	I	I	
Default	none	none	none	0	0	0	0	

Define if and only if SPC is specified in the keyword.

Card 2 1 2 3 4 5 6 7 8

Variable	CMO	CON1	CON2					
Type	F	F	F					
Default	0	0	0					

VARIABLE**DESCRIPTION**

PID	Part ID of the nodal rigid body.
CID	Optional coordinate system ID for the rigid body local system, see *DEFINE_COORDINATE_OPTION. Output of the rigid body data and the degree-of- freedom releases are done in this local system. This local system rotates with the rigid body.
NSID	Nodal set ID, see *SET_NODE_OPTION. This nodal set defines the rigid body. If NSID=0, then NSID=PID, i.e., the node set ID and the part ID are assumed to be identical.
PNODE	An optional, possibly massless, nodal point located at the mass center of the nodal rigid body. The initial nodal coordinates will be reset if necessary to ensure that they lie at the mass center. In the output files, the coordinates, accelerations, velocities, and displacements of this node will correspond to the mass center of the nodal rigid body. If CID is defined, the velocities and accelerations of PNODE will be output in the local system in the D3PLOT and D3THDT files unless PNODE is specified as a negative number in which case the global system is used.

VARIABLE	DESCRIPTION
IPRT	Print flag. For nodal rigid bodies the following values apply: EQ.1: write data into RBDOUT EQ.2: do not write data into RBDOUT Printing is suppressed for two noded rigid bodies unless IPRT is set to unity. This is to avoid excessively large RBDOUT files when many, two-noded welds are used.
DRFLAG	Displacement release flag for all nodes except the first node in the definition. EQ.-7: release x, y, and z displacement in global system EQ.-6: release z and x displacement in global system EQ.-5: release y and z displacement in global system EQ.-4: release x and y displacement in global system EQ.-3: release z displacement in global system EQ.-2: release y displacement in global system EQ.-1: release x displacement in global system EQ. 0: off for rigid body behavior EQ. 1: release x displacement in rigid body local system EQ. 2: release y displacement in rigid body local system EQ. 3: release z displacement in rigid body local system EQ. 4: release x and y displacement in rigid body local system EQ. 5: release y and z displacement in rigid body local system EQ. 6: release z and x displacement in rigid body local system EQ. 7: release x, y, and z displacement in rigid body local system
RRFLAG	Rotation release flag for all nodes except the first node in the definition. EQ.-7: release x, y, and z rotations in global system EQ.-6: release z and x rotations in global system EQ.-5: release y and z rotations in global system EQ.-4: release x and y rotations in global system EQ.-3: release z rotation in global system EQ.-2: release y rotation in global system EQ.-1: release x rotation in global system EQ. 0: off for rigid body behavior EQ. 1: release x rotation in rigid body local system EQ. 2: release y rotation in rigid body local system EQ. 3: release z rotation in rigid body local system EQ. 4: release x and y rotations in rigid body local system EQ. 5: release y and z rotations in rigid body local system EQ. 6: release z and x rotations in rigid body local system EQ. 7: release x, y, and z rotations in rigid body local system
CMO	Center of mass constraint option, CMO: EQ.+1.0: constraints applied in global directions, EQ.0.0: no constraints, EQ.-1.0: constraints applied in local directions (SPC constraint).

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CON1	<p>First constraint parameter:</p> <p><u>If CMO=+1.0, then specify global translational constraint:</u> EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.</p> <p><u>If CMO=-1.0, then specify local coordinate system ID. See *DEFINE_</u> <u>COORDINATE_OPTION:</u> This coordinate system is fixed in time.</p>
CON2	<p>Second constraint parameter:</p> <p><u>If CMO=+1.0, then specify global rotational constraint:</u> EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.</p> <p><u>If CMO=-1.0, then specify local (SPC) constraint:</u> EQ.000000 no constraint, EQ.100000 constrained x translation, EQ.010000 constrained y translation, EQ.001000 constrained z translation, EQ.000100 constrained x rotation, EQ.000010 constrained y rotation, EQ.000001 constrained z rotation.</p> <p>Any combination of local constraints can be achieved by adding the number 1 into the corresponding column.</p>

*CONSTRAINED

*CONSTRAINED_NODAL_RIGID_BODY

Required for the INERTIA option.

Card 2 1 2 3 4 5 6 7 8

Variable	XC	YC	ZC	TM	IRCS	NODEID		
Type	F	F	F	F	I	I		
Default	0	0	0	0	0	0		

VARIABLE

DESCRIPTION

XC	x-coordinate of center of mass. If nodal point, NODEID, is defined, XC, YC, and ZC are ignored and the coordinates of the nodal point, NODEID, are taken as the center of mass.
YC	y-coordinate of center of mass
ZC	z-coordinate of center of mass
TM	Translational mass
IRCS	Flag for inertia tensor reference coordinate system: EQ.0: global inertia tensor, EQ.1: principal moments of inertias with orientation vectors as given below.
NODEID	Optional nodal point defining the CG of the rigid body. If this node is not a member of the set NSID above, its motion will not be updated to correspond with the nodal rigid body after the calculation begins. PNODE and NODEID can be identical if and only if PNODE physically lies at the mass center at time zero.

Required for the INERTIA option.

Card 3 1 2 3 4 5 6 7 8

Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		
Default	none	0	0	none	0	0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>							
IXX	I _{xx} , xx component of inertia tensor							
IXY	I _{xy} (set to zero if IRCS=1)							
IXZ	I _{xz} (set to zero if IRCS=1)							
IYY	I _{yy} , yy component of inertia tensor							
IYZ	I _{yz} (set to zero if IRCS=1)							
IZZ	I _{zz} , zz component of inertia tensor							

Required for the INERTIA option.

Card 4 1 2 3 4 5 6 7 8

Variable	VTX	VTY	VTZ	VRX	VRX	VRZ		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>							
VTX	x-rigid body initial translational velocity in global coordinate system.							
VTY	y-rigid body initial translational velocity in global coordinate system.							
VTZ	z-rigid body initial translational velocity in global coordinate system.							
VRX	x-rigid body initial rotational velocity in global coordinate system.							
VRX	y-rigid body initial rotational velocity in global coordinate system.							
VRZ	z-rigid body initial rotational velocity in global coordinate system.							

*CONSTRAINED

*CONSTRAINED_NODAL_RIGID_BODY

Remarks:

The velocities defined above can be overwritten by the *INITIAL_VELOCITY card.

Optional card required for IRCS=1. Define two local vectors or a local coordinate system ID.

Card 5 1 2 3 4 5 6 7 8

Variable	XL	YL	ZL	XLIP	YLIP	ZLIP	CID2	
Type	F	F	F	F	F	F	I	
Default	none							

VARIABLE

DESCRIPTION

XL	x-coordinate of local x-axis. Origin lies at (0,0,0).
YL	y-coordinate of local x-axis
ZL	z-coordinate of local x-axis
XLIP	x-coordinate of local in-plane vector
YLIP	y-coordinate of local in-plane vector
ZLIP	z-coordinate of local in-plane vector
CID2	Local coordinate system ID, see *DEFINE_COORDINATE_.... With this option leave fields 1-6 blank.

Remarks:

The local coordinate system is set up in the following way. After the local x-axis is defined, the local z-axis is computed from the cross-product of the local x-axis vector with the given in-plane vector. Finally, the local y-axis is determined from the cross-product of the local z-axis with the local x-axis. The local coordinate system defined by CID has the advantage that the local system can be defined by nodes in the rigid body which makes repositioning of the rigid body in a preprocessor much easier since the local system moves with the nodal points.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *CONSTRAINED_NODAL_RIGID_BODY
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define a rigid body consisting of the nodes in nodal set 61.
$
$ This particular example was used to connect three separate deformable
$ parts. Physically, these parts were welded together. Modeling wise,
$ however, this joint is quit messy and is most conveniently modeled
$ by making a rigid body using several of the nodes in the area. Physically,
$ this joint was so strong that weld failure was never of concern.
$
*CONSTRAINED_NODAL_RIGID_BODY
$
$.>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$      pid      cid      nsid
$      45          61
$
$      nsid = 61   nodal set ID number, requires a *SET NODE option
$      cid       not used in this example, output will be in global coordinates
$
$
*SET_NODE_LIST
$      sid
$      61
$      nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
$      823       1057      1174      1931      2124      1961      2101
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

*CONSTRAINED

*CONSTRAINED_NODE_SET

*CONSTRAINED_NODE_SET_{OPTION}

To define an ID for the constrained node set the following option is available:

<BLANK>

ID

If the ID is defined an additional card is required.

Purpose: Define nodal constraint sets for translational motion in global coordinates. No rotational coupling. See Figure 6.19. Nodal points included in the sets should not be subjected to any other constraints including prescribed motion, e.g., with the *BOUNDARY_PRESCRIBED_MOTION options.

ID Card - Required if the option ID is active on the keyword card.

Card 1 1 2 3 4 5 6 7 8

Variable	CNSID								
Type	I								
Default	0								

Card 1 2 3 4 5 6 7 8

Variable	NSID	DOF	TF						
Type	I	I	F						
Default	none	none	1.E+20						
Remarks	1		2						

VARIABLE

DESCRIPTION

CNSID Optional constrained node set ID.

NSID Nodal set ID, see *SET_NODE_OPTION.

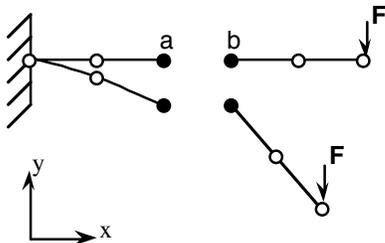
VARIABLE	DESCRIPTION
DOF	Applicable degrees-of-freedom: EQ.1: x-translational degree-of-freedom, EQ.2: y-translational degree-of-freedom, EQ.3: z-translational degree-of-freedom, EQ.4: x and y-translational degrees-of-freedom, EQ.5: y and z-translational degrees-of-freedom, EQ.6: z and x-translational degrees-of-freedom, EQ.7: x, y, and z-translational degrees-of-freedom.
TF	Failure time for nodal constraint set.

Remarks:

- The masses of the nodes are summed up to determine the total mass of the constrained set. It must be noted that the definition of a nodal rigid body is not possible with this input. For nodal rigid bodies the keyword input: *CONSTRAINED_NODAL_RIGID_BODY_OPTION, must be used.
- When the failure time, *TF*, is reached the nodal constraint becomes inactive and the constrained nodes may move freely.

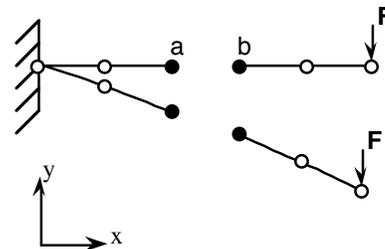
*** CONSTRAINED_NODE_SET**

Since no rotation is permitted, this option should not be used to model rigid body behavior that involves rotations.



*** CONSTRAINED_NODAL_RIGID_BODY
* CONSTRAINED_SPOTWELD**

Behavior is like a rigid beam. These options may be used to model spotwelds.



Offset nodes a and b are constrained to move together.

Figure 6.19. *CONSTRAINED_NODE_SET can lead to nonphysical responses.

*CONSTRAINED

*CONSTRAINED_NODE_SET

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONSTRAINED_NODE_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Constrain all the nodes in a nodal set to move equivalently
$ in the z-direction.
$
*CONSTRAINED_NODE_SET
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
   nsid    dof      tf
      7      3     10.0
$
$ nsid = 7 nodal set ID number, requires a *SET_NODE_option
$ dof = 3 nodal motions are equivalent in z-translation
$ tf = 3 at time=10. the nodal constraint is removed
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

***CONSTRAINED_POINTS**

Purpose: Constrain two points with the specified coordinates connecting two shell elements at locations other than nodal points. In this option, the penalty method is used to constrain the translational and rotational degrees-of-freedom of the points. Force resultants are written into the SWFORC ASCII file for post-processing.

Card Format (I10)

Card 1 1 2 3 4 5 6 7 8

Variable	CID								
Type	I								
Default	none								

Card Format (I8,3E16.0)

Card 2 1 2 3 4 5 6 7 8 9 10

Variable	EID1	X1	Y1	Z1						
Type	I	F	F	F						
Default	none	0.	0.	0.						

Card 3

Variable	EID2	X2	Y2	Z2						
Type	I	F	F	F						
Default	none	0.	0.	0.						

*CONSTRAINED

*CONSTRAINED_POINTS

Card Format (4E10.0)

Card 4 1 2 3 4 5 6 7 8

Variable	PSF	FAILA	FAILS	FAILM				
Type	F	F	F	F				
Default	1.0	0.0	0.0	0.0				

VARIABLE

DESCRIPTION

CID	Constrained points ID.
Xi, Yi, Zi	Coordinates of the constrained points, i=1,2.
EIDi	Shell element ID, i=1,2.
PSF	Penalty scale factor (Default=1.0).
FAILA	Axial force resultant failure value (Skip if zero).
FAILS	Shear force resultant failure value (Skip if zero).
FAILM	Moment resultant failure value (Skip if zero).

***CONSTRAINED_RIGID_BODIES**

Purpose: Merge two rigid bodies. One rigid body, called slave rigid body, is merged to the other one called a master rigid body.

Card 1 2 3 4 5 6 7 8

Variable	PIDM	PIDS						
Type	I	I						
Default	none	none						

VARIABLE

DESCRIPTION

- PIDM Master rigid body part ID, see *PART.
- PIDS Slave rigid body part ID, see *PART.

Remarks:

The slave rigid body is merged to the master rigid body. The inertial properties computed by LS-DYNA are based on the combination of the master rigid body plus all the rigid bodies which are slaved to it unless the inertial properties of the master rigid body are defined via the *PART_INERTIA keyword in which case those properties are used for the combination of the master and slave rigid bodies. Note that a master rigid body may have many slaves.

Independent rigid bodies must not share common nodes since each rigid body updates the motion of its nodes independently of the other rigid bodies. If common nodes exist between rigid bodies the rigid bodies sharing the nodes must be merged.

It is also possible to merge rigid bodies that are completely separated and share no common nodal points or boundaries. All actions valid for the master rigid body, e.g., constraints, given velocity, are now also valid for the newly-created rigid body.

***CONSTRAINED_RIGID_BODY_STOPPERS**

***CONSTRAINED**

***CONSTRAINED_RIGID_BODY_STOPPERS**

Purpose: Rigid body stoppers provide a convenient way of controlling the motion of rigid tooling in metalforming applications. The motion of a “master” rigid body is limited by load curves. This option will stop the motion based on a time dependent constraint. The stopper overrides prescribed motion boundary conditions (except relative displacement) operating in the same direction for both the master and slaved rigid bodies. See Figure 6.20.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	LCMAX	LCMIN	PSIDMX	PSIDMN	LCVMNX	DIR	VID
Type	I	I	I	I	I	I	I	I
Default	required	0	0	0	0	0	required	0

Card 2

Variable	TB	TD						
Type	F	F						
Default	0	10 ²¹						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID of master rigid body, see *PART.
LCMAX	Load curve ID defining the maximum coordinate or displacement as a function of time. See *DEFINE_CURVE: LT.0: Load Curve ID LCMAX provides an upper bound for the displacement of the rigid body EQ.0: no limitation of the maximum displacement. GT.0: Load Curve ID LCMAX provides an upper bound for the position of the rigid body center of mass

VARIABLE	DESCRIPTION
LCMIN	Load curve ID defining the minimum coordinate or displacement as a function of time. See *DEFINE_CURVE: LT.0: Load Curve ID LCMIN defines a lower bound for the displacement of the rigid body EQ.0: no limitation of the minimum displacement. GT.0: Load Curve ID LCMIN defines a lower bound for the position of the rigid body center of mass
PSIDMX	Optional part set ID of rigid bodies that are slaved in the maximum coordinate direction to the master rigid body. The part set definition, (see *SET_PART_COLUMN) may be used to define the closure distance (D_1 and D_2 in Figure 6.20) which activates the constraint. The constraint does not begin to act until the master rigid body stops. If the distance between the master rigid body is greater than or equal to the closure distance, the slave rigid body motion away from the master rigid body also stops. However, the slaved rigid body is free to move towards the master. If the closure distance is input as zero (0.0) then the slaved rigid body stops when the master stops.
PSIDMN	Optional part set ID of rigid bodies that are slaved in the minimum coordinate direction to the master rigid body. The part set definition, (see *SET_PART_COLUMN) may be used to define the closure distance (D_1 and D_2 in Figure 6.20) which activates the constraint. The constraint does not begin to act until the master rigid body stops. If the distance between the master rigid body is less than or equal to the closure distance, the slave rigid body motion towards the master rigid body also stops. However, the slaved rigid body is free to move away from the master. If the closure distance is input as zero (0.0) then the slaved rigid body stops when the master stops.
LCVMX	Load curve ID which defines the maximum absolute value of the velocity as a function of time that is allowed for the master rigid body. See *DEFINE_CURVE: EQ.0: no limitation on the velocity.
DIR	Direction stopper acts in: EQ.1: x-translation, EQ.2: y-translation, EQ.3: z-translation, EQ.4: arbitrary, defined by vector VID (see below), EQ.5: x-axis rotation, EQ.6: y-axis rotation, EQ.7: z-axis rotation, EQ.8: arbitrary, defined by vector VID (see below).

VARIABLE	DESCRIPTION
VID	Vector for arbitrary orientation of stopper, see *DEFINE_VECTOR.
TB	Time at which stopper is activated.
TD	Time at which stopper is deactivated.

Remarks:

The optional definition of part sets in minimum or maximum coordinate direction allows the motion to be controlled in arbitrary direction.

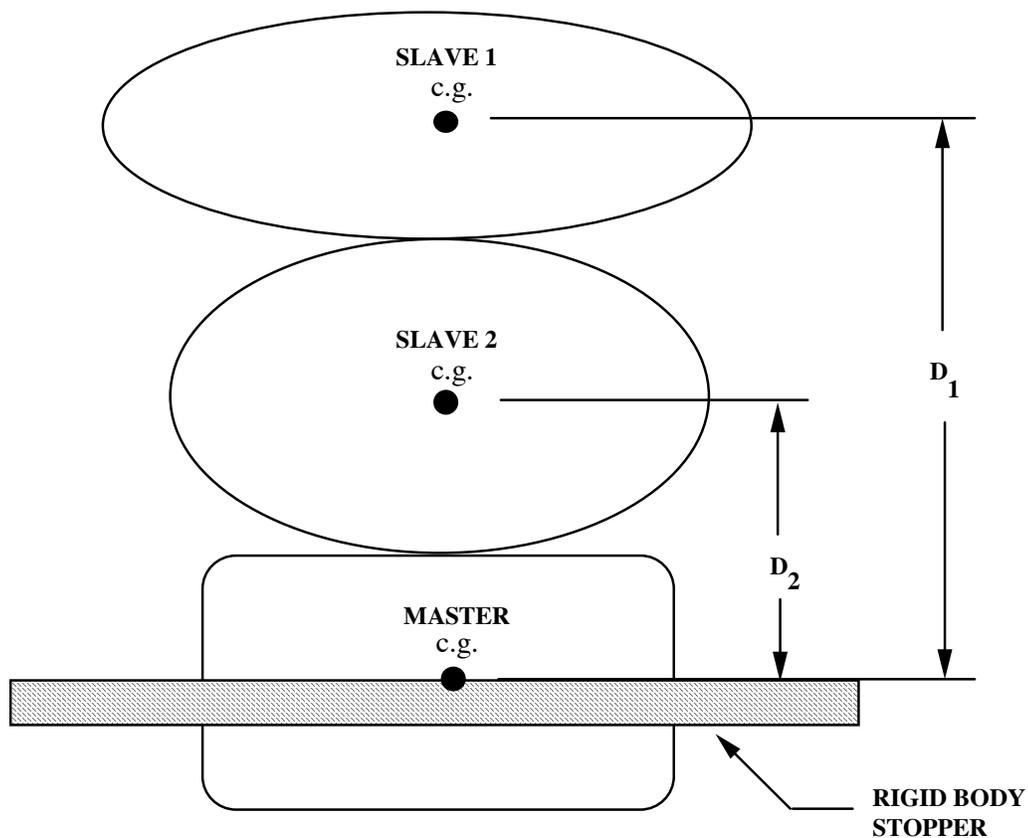


Figure 6.20 When the master rigid body reaches the rigid body stopper, the velocity component into the stopper is set to zero. Slave rigid bodies 1 and 2 also stop if the distance between their mass centers and the master rigid body is less than or equal to the input values D_1 and D_2 , respectively. (c.g. + center of gravity).

*CONSTRAINED

*CONSTRAINED_RIVET

*CONSTRAINED_RIVET_{OPTION}

To define an ID for the rivet, the following option is available:

ID

If the ID is defined an additional card is required.

Purpose: Define massless rivets between non-contiguous nodal pairs. The nodes must not have the same coordinates. The action is such that the distance between the two nodes is kept constant throughout any motion. No failure can be specified.

ID Card - Required if the option ID is active on the keyword card.

Card 1 1 2 3 4 5 6 7 8

Variable	RID							
Type	I							
Default	0							

Card 1

Variable	N1	N2	TF					
Type	I	I	F					
Default	none	none	1.E+20					
Remarks	1		2					

VARIABLE

DESCRIPTION

RID	Optional rivet ID.
N1	Node ID
N2	Node ID
TF	Failure time for nodal constraint set.

Remarks:

1. Nodes connected by a rivet cannot be members of another constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body, i.e., nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also care must be taken to ensure that single point constraints applied to nodes in a constraint set do not conflict with the constraint sets constrained degrees-of-freedom.
2. When the failure time, *TF*, is reached the rivet becomes inactive and the constrained nodes may move freely.

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONSTRAINED_RIVET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Connect node 382 to node 88471 with a massless rivet.
$
*CONSTRAINED_RIVET
$
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$      n1      n2      tf
$      382      88471    0.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

*CONSTRAINED

*CONSTRAINED_SHELL_TO_SOLID

*CONSTRAINED_SHELL_TO_SOLID

Purpose: Define a tie between a shell edge and solid elements. Nodal rigid bodies can perform the same function and may also be used.

Card 1 2 3 4 5 6 7 8

Variable	NID	NSID						
Type	I	I						
Default	none	none						
Remarks								

VARIABLE

DESCRIPTION

NID	Shell node ID
NSID	Solid nodal set ID, see *SET_NODE_OPTION.

Remarks:

The shell-brick interface, an extension of the tied surface capability, ties regions of hexahedron elements to regions of shell elements. A shell node may be tied to up to nine brick nodes lying along the tangent vector to the nodal fiber. See Figure 6.21. During the calculation, the brick nodes thus constrained, must lie along the fiber but can move relative to each other in the fiber direction. The shell node stays on the fiber at the same relative spacing between the first and last brick node. The brick nodes must be input in the order in which they occur, in either the plus or minus direction, as one moves along the shell node fiber.

This feature is intended to tie four node shells to eight node shells or solids; it is not intended for tying eight node shells to eight node solids.

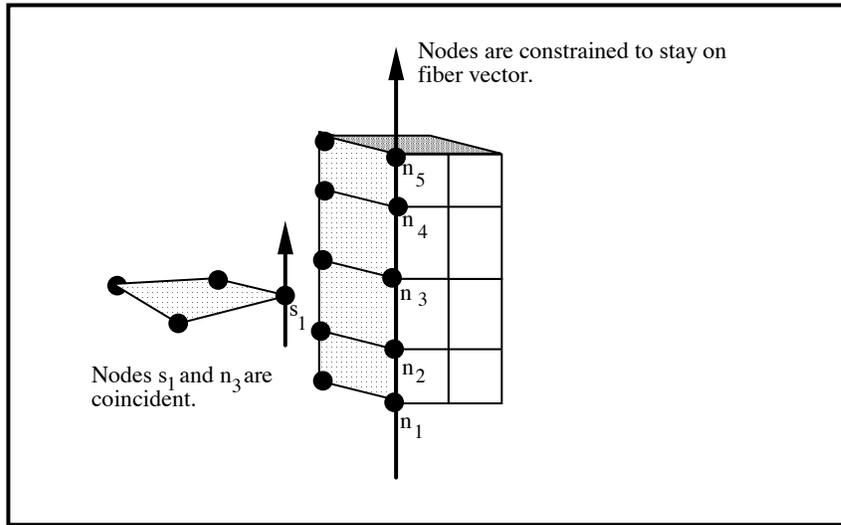


Figure 6.21 The interface between shell elements and solids ties shell node s_1 to a line of nodes on the solid elements n_1 - n_5 . It is very important for the nodes to be aligned.

```

$$$
$$$$ *CONSTRAINED_SHELL_TO_SOLID
$$$
$ Tie shell element, at node 329, to a solid element at node 203.
$ - nodes 329 and 203 are coincident
$
$ Additionally, define a line of nodes on the solids elements, containing
$ node 203, that must remain in the same direction as the fiber of the shell
$ containing node 329. In other words:
$
$ - Nodes 119, 161, 203, 245 and 287 are nodes on a solid part that
$ define a line on that solid part.
$ - This line of nodes will be constrained to remain linear throughout
$ the simulation.
$ - The direction of this line will be kept the same as the fiber of the
$ of the shell containing node 329.
$
*CONSTRAINED_SHELL_TO_SOLID
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$ nid nsid
$ 329 4
$
*SET_NODE_LIST
$ sid
$ 4
$ nid1 nid2 nid3 nid4 nid5 nid6 nid7 nid8
$ 119 161 203 245 287
$
$$$$

```

*CONSTRAINED

*CONSTRAINED_SPLINE

*CONSTRAINED_SPLINE

Purpose: Define an elastic cubic spline interpolation constraint. The displacements and slopes at the end points are continuous. The first and last nodes, which define the constraint, must be independent. The degrees-of-freedom of interior nodes may be either dependent or independent.

Card 1 1 2 3 4 5 6 7 8

Variable	SPLID	DLRATIO						
Type	I	I						
Default	0	0.10						

Cards 2, 3, 4, etc. Define one card per independent/dependent node. The first and last nodes must be independent. The next “*” card terminates this input.

Card 2... 1 2 3 4 5 6 7 8

Variable	NID	DOF						
Type	I	I						
Default	0	0						

VARIABLE

DESCRIPTION

SPLID

Spline constraint ID.

DLRATIO

Ratio of bending to torsional stiffness for an elastic tubular beam which connects the independent degrees-of-freedom. The default value is set to 0.10.

NID

Independent/dependent node ID. For explicit problems this node should not be a member of a rigid body, or elsewhere constrained in the input.

VARIABLE	DESCRIPTION
DOF	Degrees-of-freedom. The list of dependent degrees-of-freedom consists of a number with up to six digits, with each digit representing a degree of freedom. For example, the value 1356 indicates that degrees of freedom 1, 3, 5, and 6 are controlled by the constraint. The default is 123456. Digit: degree of freedom ID's: EQ.1: x EQ.2: y EQ.3: z EQ.4: rotation about x axis EQ.5: rotation about y axis EQ.6: rotation about z axis

*CONSTRAINED

*CONSTRAINED_SPOTWELD

*CONSTRAINED_SPOTWELD_{OPTION}_{OPTION}

If it is desired to use a time filtered force calculation for the forced based failure criterion then the following option is available:

FILTERED_FORCE

and one additional card must be defined below. To define an ID for the spotweld the following option is available:

ID

If the ID is defined an additional card is required. The ordering of the options is arbitrary.

Purpose: Define massless spot welds between non-contiguous nodal pairs. The spot weld is a rigid beam that connects the nodal points of the nodal pairs; thus, nodal rotations and displacements are coupled. The spot welds must be connected to nodes having rotary inertias, i.e., beams or shells. If this is not the case, for example, if the nodes belong to solid elements, use the option: *CONSTRAINED_RIVET. For Implicit, this case is treated like a rivet, constraining only the displacements. Note that shell elements do not have rotary stiffness in the normal direction and, therefore, this component cannot be transmitted. Spot welded nodes must not have the same coordinates. Coincident nodes in a spot weld can be handled by the *CONSTRAINED_NODAL_RIGID_BODY option. Brittle and ductile failures can be specified. Brittle failure is based on the resultant forces acting on the weld, and ductile failure is based on the average plastic strain value of the shell elements which include the spot welded node. Spot welds, which are connected to massless nodes, are automatically deleted in the initialization phase and a warning message is printed in the MESSAG file and the D3HSP file.

Warning: The accelerations of spot welded nodes are output as zero into the various databases, but if the acceleration of spotwelded nodes are required, use either the *CONSTRAINED_GENERALIZED_WELD or the *CONSTRAINED_NODAL_RIGID_BODY input. However, if the output interval is frequent enough accurate acceleration time histories can be obtained from the velocity time history by differentiation in the post-processing phase.

ID Card - Required if the option ID is active on the keyword card.

Card 1 1 2 3 4 5 6 7 8

Variable	WID								
Type	I								
Default	0								

CONSTRAINED_SPOTWELD**CONSTRAINED**

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	SN	SS	N	M	TF	EP
Type	I	I	F	F	F	F	F	F
Default	none	none	optional	optional	none	none	1.E+20	1.E+20
Remarks	1.		2.				3	4

Define if and only if the option FILTERED_FORCE is specified.

Card 2 1 2 3 4 5 6 7 8

Variable	NF	TW						
Type	I	F						
Default	none	none						
Remarks								

VARIABLE**DESCRIPTION**

WID	Optional weld ID.
N1	Node ID
N2	Node ID
SN	Normal force at spotweld failure (see Remark 2 below).
SS	Shear force at spotweld failure (see Remark 2 below).
N	Exponent for normal spotweld force (see Remark 2 below).
M	Exponent for shear spotweld force (see Remark 2 below).

VARIABLE	DESCRIPTION
TF	Failure time for nodal constraint set.
EP	Effective plastic strain at failure.
NF	Number of force vectors stored for filtering.
TW	Time window for filtering.

Remarks:

- Nodes connected by a spot weld cannot be members of another constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body, i.e., nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also, care must be taken to ensure that single point constraints applied to nodes in a constraint set do not conflict with the constraint sets constrained degrees-of-freedom.

- Failure of the spot welds occurs when:

$$\left(\frac{|f_n|}{S_n}\right)^n + \left(\frac{|f_s|}{S_s}\right)^m \geq 1$$

where f_n and f_s are the normal and shear interface force. Component f_n is nonzero for tensile values only.

- When the failure time, TF , is reached the spot weld becomes inactive and the constrained nodes may move freely.
- Spot weld failure due to plastic straining occurs when the effective nodal plastic strain exceeds the input value, ϵ_{fail}^p . This option can model the tearing out of a spotweld from the sheet metal since the plasticity is in the material that surrounds the spotweld, not the spotweld itself. A least squares algorithm is used to generate the nodal values of plastic strains at the nodes from the element integration point values. The plastic strain is integrated through the element and the average value is projected to the nodes via a least square fit. This option should only be used for the material models related to metallic plasticity and can result in slightly increased run times. Failures can include both the plastic and brittle failures.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$   *CONSTRAINED_SPOTWELD
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$   Spotweld two nodes (34574 and 34383) with the approximate strength
$   of a 3/8" SAE Grade No 3 bolt.
$
*CONSTRAINED_SPOTWELD
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$      n1          n2          sn          sf          n          m          tf          ps
$      34574      34383      36.0       18.0       2.0       2.0       10.       1.0
$
$
$   sn = 36.0   normal failure force is 36 kN
$   sf = 18.0   shear failure force is 18 kN
$   n  = 2.0    normal failure criteria is raised to the power of 2
$   m  = 2.0    shear failure criteria is raised to the power of 2
$   tf = 10.0   failure occurs at time 10 unless strain failure occurs
$   ps = 2.0    plastic strain at failure
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

*CONSTRAINED

*CONSTRAINED_TIE-BREAK

*CONSTRAINED_TIE-BREAK

Purpose: Define a tied shell edge to shell edge interface that can release locally as a function of plastic strain of the shells surrounding the interface nodes. A rather ductile failure is achieved.

Card	1	2	3	4	5	6	7	8
Variable	SNSID	MNSID	EPPF					
Type	I	I	F					
Default	none	none	0.					
Remarks		1, 2	3, 4					

VARIABLE

DESCRIPTION

SNSID	Slave node set ID, see *SET_NODE_OPTION.
MNSID	Master node set ID, see *SET_NODE_OPTION.
EPPF	Plastic strain at failure

Remarks:

1. Nodes in the master node set must be given in the order they appear as one moves along the edge of the surface.
2. Tie-breaks may not cross.
3. Tie-breaks may be used to tie shell edges together with a failure criterion on the joint. If the average volume-weighted effective plastic strain in the shell elements adjacent to a node exceeds the specified plastic strain at failure, the node is released. The default plastic strain at failure is defined for the entire tie-break but can be overridden in the slave node set to define a unique failure plastic strain for each node.
4. Tie-breaks may be used to simulate the effect of failure along a predetermined line, such as a seam or structural joint. When the failure criterion is reached in the adjoining elements, nodes along the slideline will begin to separate. As this effect propagates, the tie-breaks will appear to “unzip,” thus simulating failure of the connection.

***CONSTRAINED_TIED_NODES_FAILURE**

Purpose: Define a tied node set with failure based on plastic strain. The nodes must be coincident.

Card 1 2 3 4 5 6 7 8

Variable	NSID	EPPF	ETYPE					
Type	I	F						
Default	none	0.						
Remarks	1, 2, 3, 4							

VARIABLE

DESCRIPTION

NSID	Nodal set ID, see *SET_NODE_OPTION.
EPPF	Plastic strain at failure
ETYPE	Element type for nodal group: EQ.0: shell, EQ.1: solid element

Remarks:

1. This feature applies to solid and shell elements using plasticity material models, and to solid elements using the honeycomb material *MAT_HONEYCOMB (EPPF=plastic volume strain). The specified nodes are tied together until the average volume weighted plastic strain exceeds the specified value. Entire regions of individual shell elements may be tied together unlike the tie-breaking shell slidelines. The tied nodes are coincident until failure. When the volume weighted average of the failure value is reached for a group of constrained nodes, the nodes of the elements that exceed the failure value are released to simulate the formation of a crack.
2. To use this feature to simulate failure, each shell element in the failure region should be generated with unique node numbers that are coincident in space with those of adjacent elements. Rather than merging these coincident nodes, the *CONSTRAINED_TIED_NODES_FAILURE option ties the nodal points together. As plastic strain develops and exceeds the failure strain, cracks will form and propagate through the mesh.

***CONSTRAINED**

***CONSTRAINED_TIED_NODES_FAILURE**

- Entire regions of individual shell elements may be tied together, unlike the *CONSTRAINED_TIE-BREAK option. This latter option is recommended when the location of failure is known, e.g., as in the plastic covers which hide airbags in automotive structures.
- When using surfaces of shell elements defined using the *CONSTRAINED_TIED_NODES_FAILURE option in contact, it is best to defined each node in the surface as a slave node with the NODE_TO_SURFACE contact options. If this is not possible, the automatic contact algorithms beginning with *CONTACT_AUTOMATIC_... all of which include thickness offsets are recommended.

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$
  *CONSTRAINED_TIED_NODES_FAILURE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Tie shell elements together at the nodes specified in nodal set 101. The
$ constraint will be broken when the plastic strain at the nodes exceeds 0.085.
$
$ In this example, four shell elements come together at a common point.
$ The four corners of the shells are tied together with failure as opposed
$ to the more common method of merging the nodes in the pre-processing stage.
$
*CONSTRAINED_TIED_NODES_FAILURE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$   nsid     eppf
$     101     0.085
$
$
*SET_NODE_LIST
$   sid
$   101
$   nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
$     775       778       896       897
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

***CONTACT**

The keyword ***CONTACT** provides a way of treating interaction between disjoint parts. Different types of contact may be defined:

***CONTACT_OPTION1_{OPTION2}_{OPTION3}_{OPTION4}_{OPTION5}**

***CONTACT_AUTO_MOVE**

***CONTACT_COUPLING**

***CONTACT_ENTITY**

***CONTACT_GEBOD_OPTION**

***CONTACT_GUIDED_CABLE**

***CONTACT_INTERIOR**

***CONTACT_RIGID_SURFACE**

***CONTACT_1D**

***CONTACT_2D_OPTION1_{OPTION2}_{OPTION3}**

The first, ***CONTACT_...**, is the general 3D contact algorithms. The second, ***CONTACT_COUPLING**, provides a means of coupling to deformable surfaces to MADYMO. The third, ***CONTACT_ENTITY**, treats contact using mathematical functions to describe the surface geometry for the master surface. The fourth, ***CONTACT_GEBOD** is a specialized form of the contact entity for use with the rigid body dummies (see ***COMPONENT_GEBOD**). The fifth, ***CONTACT_INTERIOR**, is under development and is used with soft foams where element inversion is sometimes a problem. Contact between layers of brick elements is treated to eliminate negative volumes. The sixth, ***CONTACT_RIGID_SURFACE** is for modeling road surfaces for durability and NVH calculations. The seventh, ***CONTACT_1D**, remains in LS-DYNA for historical reasons, and is sometimes still used to model rebars which run along edges of brick elements. The last, ***CONTACT_2D**, is the general 2D contact algorithm based on those used previously in LS-DYNA2D.

***CONTACT**

***CONTACT_OPTION1_{OPTION2}...**

***CONTACT_OPTION1_{OPTION2}_{OPTION3}_{OPTION4}_{OPTION5}**

Purpose: Define a contact interface.

OPTION1 specifies the contact type. Not all options are implemented for implicit solutions. A list of available contact options is given in Remark 4:

AIRBAG_SINGLE_SURFACE
AUTOMATIC_GENERAL
AUTOMATIC_GENERAL_INTERIOR
AUTOMATIC_NODES_TO_SURFACE
AUTOMATIC_NODES_TO_SURFACE_SMOOTH
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH
AUTOMATIC_SINGLE_SURFACE
AUTOMATIC_SINGLE_SURFACE_SMOOTH
AUTOMATIC_SURFACE_TO_SURFACE
AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK
AUTOMATIC_SURFACE_TO_SURFACE_SMOOTH
CONSTRAINT_NODES_TO_SURFACE
CONSTRAINT_SURFACE_TO_SURFACE
DRAWBEAD
ERODING_NODES_TO_SURFACE
ERODING_SINGLE_SURFACE
ERODING_SURFACE_TO_SURFACE
FORCE_TRANSDUCER_CONSTRAINT
FORCE_TRANSDUCER_PENALTY
FORMING_NODES_TO_SURFACE
FORMING_NODES_TO_SURFACE_SMOOTH
FORMING_ONE_WAY_SURFACE_TO_SURFACE
FORMING_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH
FORMING_SURFACE_TO_SURFACE
FORMING_SURFACE_TO_SURFACE_SMOOTH
NODES_TO_SURFACE
NODES_TO_SURFACE_INTERFERENCE
NODES_TO_SURFACE_SMOOTH
ONE_WAY_SURFACE_TO_SURFACE

ONE_WAY_SURFACE_TO_SURFACE_INTERFERENCE
ONE_WAY_SURFACE_TO_SURFACE_SMOOTH
RIGID_NODES_TO_RIGID_BODY
RIGID_BODY_ONE_WAY_TO_RIGID_BODY
RIGID_BODY_TWO_WAY_TO_RIGID_BODY
SINGLE_EDGE
SINGLE_SURFACE
SLIDING_ONLY
SLIDING_ONLY_PENALTY
SPOTWELD
SPOTWELD_WITH_TORSION
SURFACE_TO_SURFACE
SURFACE_TO_SURFACE_INTERFERENCE
SURFACE_TO_SURFACE_SMOOTH
SURFACE_TO_SURFACE_CONTRACTION_JOINT
TIEBREAK_NODES_TO_SURFACE
TIEBREAK_NODES_ONLY
TIEBREAK_SURFACE_TO_SURFACE
TIED_NODES_TO_SURFACE
TIED_SHELL_EDGE_TO_SURFACE
TIED_SURFACE_TO_SURFACE
TIED_SURFACE_TO_SURFACE_FAILURE

OPTION2 specifies a thermal contact and is defined by either:

THERMAL
THERMAL_FRICTION

Only the **SURFACE_TO_SURFACE** contact type may be used with this option.

OPTION3 specifies that the first card to read defines the heading and ID number of contact interface and takes the single option:

ID

OPTION4 specifies that offsets may be used with the tied contacts types. If one of these three offset options is set, then offsets are permitted for these contact types, and, if not, the nodes are projected back to the contact surface during the initialization phase and a constraint formulation is used. Note that in a constraint formulation the nodes of rigid bodies are not permitted in the definition.

OFFSET

Contact types TIED_NODES_TO_SURFACE, TIED_SHELL_EDGE_TO_SURFACE, and TIED_SURFACE_TO_SURFACE may be used with this option. The OFFSET option switches the formulation from a constraint type formulation to one that is penalty based where the force and moment (if applicable) resultants are transferred discrete spring elements between the slave nodes and master segments. For the TIED_SHELL_EDGE_TO_SURFACE contact the BEAM_OFFSET option may be preferred. Rigid bodies can be used with this option. *The nodal points in the TIED_NODES_TO_SURFACE option and the TIED_SURFACE_TO_SURFACE may not be connected to structural nodes, i.e., nodes with rotational degrees-of-freedom, since the rotational degrees-of-freedom are not affected, which will lead to an instability since the translational motions due to rotation are imposed on the slave nodes.*

BEAM_OFFSET

This option applies only to contact type TIED_SHELL_EDGE_TO_SURFACE. If this option is set, then offsets are permitted for this contact type. The BEAM_OFFSET option switches the formulation from a constraint type formulation to one that is penalty based. Beam like springs are used to transfer force and moment resultants between the slave nodes and the master segments. Rigid bodies can be used with this option.

CONSTRAINED_OFFSET

Contact types TIED_NODES_TO_SURFACE, TIED_SHELL_EDGE_TO_SURFACE, and TIED_SURFACE_TO_SURFACE may be used with this option. If this option is set, then offsets are permitted for these contact types. The CONSTRAINED_OFFSET option is a constraint type formulation. *The nodal points in the TIED_NODES_TO_SURFACE option and the TIED_SURFACE_TO_SURFACE may not be connected to structural nodes, i.e., nodes with rotational degrees-of-freedom, since the rotational degrees-of-freedom are not affected, which will lead to an instability since the translational motions due to rotation are imposed on the slave nodes.*

OPTION5 gives extra options specifically for MPP implementation.

MPP**Remarks:**

1. OPTION1, OPTION2, OPTION3 and OPTION4 may appear in any order in the keyword command line. The data must be in the order specified below.
2. OPTION1 is mandatory.
3. OPTION2, OPTION3 and OPTION4 are optional.
4. The following contact types are available for implicit calculations:

SURFACE_TO_SURFACE

SURFACE_TO_SURFACE_SMOOTH
SURFACE_TO_SURFACE_CONTRACTION_JOINT
NODES_TO_SURFACE
NODES_TO_SURFACE_SMOOTH
ONE_WAY_SURFACE_TO_SURFACE
ONE_WAY_SURFACE_TO_SURFACE_SMOOTH
FORMING_SURFACE_TO_SURFACE
FORMING_SURFACE_TO_SURFACE_SMOOTH
FORMING_NODES_TO_SURFACE
FORMING_NODES_TO_SURFACE_SMOOTH
FORMING_ONE_WAY_SURFACE_TO_SURFACE
FORMING_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH
AUTOMATIC_SURFACE_TO_SURFACE
AUTOMATIC_SURFACE_TO_SURFACE_SMOOTH
AUTOMATIC_NODES_TO_SURFACE
AUTOMATIC_NODES_TO_SURFACE_SMOOTH
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH
AUTOMATIC_SINGLE_SURFACE
AUTOMATIC_SINGLE_SURFACE_SMOOTH
TIED_SURFACE_TO_SURFACE_OFFSET
TIED_NODES_TO_SURFACE_OFFSET
2D_AUTOMATIC_SURFACE_TO_SURFACE

5. For smooth contact, a smooth curve-fitted surface is used to represent the master segment, so that it can provide a more accurate representation of the actual surface, reduce the contact noise, and produce smoother results with coarse mesh. All the smooth contacts are available for MPP, only the forming smooth contacts are available for SMP.

DISCUSSION AND EXAMPLES:

A brief discussion on the contact types and a few examples are provided at the end of this section. A theoretical discussion is provided in the LS-DYNA Theory Manual.

Card ordering is important in this section:

- **Card for the ID option is inserted here; otherwise, do not define this card.**

Define the ID and heading card first.

- **Card for the _MPP option is inserted here; otherwise, do not define this card.**
- **Cards 1 to 3 are mandatory for all contact types.**
- **Card 4 is mandatory for the following contact types:**

***CONTACT_CONSTRAINT_type**

***CONTACT_DRAWBEAD**

***CONTACT_ERODING_type**

***CONTACT_..._INTERFERENCE**

***CONTACT_RIGID_type**

***CONTACT_TIEBREAK_type**

***CONTACT_..._CONTRACTION_JOINT_type**

Each of these types have different Card 4 formats. These card formats are presented in this manual after the optional cards specified above but, if used, Card 4 needs to be specified in your dyna deck before the optional cards.

- **Card for the THERMAL option is inserted here; otherwise, do not define this card.**

Additional parameters are required for thermal contact and are defined on this card.

- **Optional Card A**

Additional contact parameters that may be user specified. Default values have evolved over time to become pretty good values for most circumstances.

- **Optional Card B**

Additional contact parameters that may be user specified. Default values have evolved over time to become pretty good values for most circumstances. If Optional Card B is used, then Optional Card A is mandatory (use a blank line if no changes are desired for Card A parameters).

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

*CONTACT..._...MPP

Card 1	1	2	3	4	5	6	7	8
Variable	IGNORE	BUCKET	LCBUCKET	NS2TRACK	INITITER	PARMAX		CPARM8
Type	I	I	I	I	I	F		I
Default	0	200	none	3	2	1.0005		0

VARIABLE	DESCRIPTION
IGNORE	This is the same as the “ignore initial penetrations” option on the *CONTROL_CONTACT Optional Card C entry 2 and can also be specified in the normal contact control cards. It predates both of those, and is not really needed anymore since both are honored by the MPP code. That is, if any of the three are on, initial penetrations are tracked.
BUCKET	Bucketsort frequency. This field is the only way to specify the bucketsort frequency for the MPP code. The BSORT option on Optional Card A is ignored.
LCBUCKET	Loadcurve for bucketsort frequency. The normal input for this is ignored by MPP.
NS2TRACK	Number of potential contacts to track for each slave node. The normal input for this (DEPTH on Optional Card A) is ignored.
INITITER	Number of iterations to perform when trying to eliminate initial penetrations.
PARMAX	The parametric extension distance for contact segments. The MAXPAR parameter on Optional Card A is not used.
CPARM8	Exclude beam to beam contact from the same PID for AUTOMATIC_GENERAL. EQ.0: disable (default) EQ.1: enable

The following card is read if and only if “&” is defined in column 1 of the first field.

Optional	1	2	3	4	5	6	7	8
Variable		CHKSEGS	PENSF	GRPABLE				
Type		I	F	I				
Default		0	1.0	0				

VARIABLE**DESCRIPTION**

CHKSEGS	If this value is non-zero, then the node to surface and surface to surface contacts will perform a special check at time 0 for elements that are inverted (or nearly so), and remove them from contact. These poorly formed elements have been known to occur on the tooling in metalforming problems, which allows these problems to run. It should not normally be needed for reasonable meshes.
PENSF	This option is used together with IGNORE for 3D forging problems. If non-zero, the IGNORED penetration distance is multiplied by this value each cycle, effectively pushing the slave node back out to the surface. This is useful for nodes that might get generated below the master surface during 3D remeshing. Care should be exercised, as energy may be generated and stability may be effected for values lower than 0.95. A value in the range of 0.98 to 0.99 or higher (but < 1.0) is recommended.
GRPABLE	This option is experimental at this time. It activates a different contact algorithm available for SINGLE_SURFACE, NODE_TO_SURFACE, and SURFACE_TO_SURFACE contacts. The new algorithm does not support all options as of yet, and is still under development. It can be significantly faster and scale better than the normal algorithm.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

Card 1 is mandatory for all contact types.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	MSID	SSTYP	MSTYP	SBOXID	MBOXID	SPR	MPR
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none			0	0
Remarks	1	2			optional	optional	0=off	0=off

VARIABLE

DESCRIPTION

SSID	Slave segment, node set ID, part set ID, part ID, or shell element set ID, see *SET_SEGMENT, *SET_NODE_OPTION, *PART, *SET_PART or *SET_SHELL_OPTION. For eroding contact use either a part ID or a part set ID. EQ.0: all part IDs are included for single surface contact, automatic single surface, and eroding single surface.
MSID	Master segment set ID, part set ID, part ID, or shell element set ID, see *SET_SEGMENT, *SET_NODE_OPTION, *PART, *SET_PART, or *SET_SHELL_OPTION: EQ.0: for single surface contact, automatic single surface, and eroding single surface.
SSTYP	Slave segment or node set type. The type must correlate with the number specified for SSID: EQ.0: segment set ID for surface-to-surface contact, EQ.1: shell element set ID for surface-to-surface contact, EQ.2: part set ID, EQ.3: part ID, EQ.4: node set ID for node to surface contact, EQ.5: include all for single surface definition. EQ.6: part set ID for exempted parts. All non-exempted parts are included in the contact.
MSTYP	Master segment set type. The type must correlate with the number specified for MSID: EQ.0: segment set ID, EQ.1: shell element set ID, EQ.2: part set ID, EQ.3: part ID.

VARIABLE	DESCRIPTION
SBOXID	Include in contact definition only those slave nodes/segments within box SBOXID (corresponding to BOXID in *DEFINE_BOX), or if SBOXID is negative, only those slave nodes/segments within contact volume SBOXID (corresponding to CVID in *DEFINE_CONTACT_VOLUME). A positive value of SBOXID can be used only if SSTYP is set to 2 or 3, i.e., SSID is a part ID or part set ID.
MBOXID	Include in contact definition only those master segments within box MBOXID (corresponding to BOXID in *DEFINE_BOX), or if MBOXID is negative, only those master segments within contact volume MBOXID (corresponding to CVID in *DEFINE_CONTACT_VOLUME). A positive value of MBOXID can be used only if MSTYP is set to 2 or 3, i.e., MSID is a part ID or part set ID.
SPR	Include the slave side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: slave side forces included.
MPR	Include the master side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: master side forces included.

Remarks:

1. Giving a slave set ID equal to zero is valid only for the single surface contact algorithms, i.e., the options SINGLE_SURFACE, and the AUTOMATIC_, AIRBAG_, and ERODING_SINGLE_SURFACE options.
2. A master set ID is not defined for the single surface contact algorithms (including AUTOMATIC_GENERAL). A master set ID is optional for FORCE_TRANSDUCERS. If a master set is defined for the FORCE_TRANSDUCER option, only those force that develop between and master and slave surfaces are considered. *The master surface option is only implemented for the _PENALTY option and works only with the AUTOMATIC_SINGLE_SURFACE contact types.*

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

Card 2 is mandatory for all contact types.

Card 2	1	2	3	4	5	6	7	8
Variable	FS	FD	DC	VC	VDC	PENCHK	BT	DT
Type	F	F	F	F	F	I	F	F
Default	0.	0.	0.	0.	0.	0	0.	1.0E20
Remarks								

VARIABLE

DESCRIPTION

FS

Static coefficient of friction if FS is >0 and not equal to 2. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC|v_{rel}|}$. The two other possibilities are:

EQ.-2: If the frictional coefficients defined in the *DEFINE_FRICTION section are to be used, set FS to the negative number, -2.0.

EQ.-1: If the frictional coefficients defined in the *PART section are to be used, set FS to the negative number, -1.0.

WARNING: Please note that the FS=-1.0 and FS=-2.0 options apply only to contact types:

SINGLE_SURFACE,
 AUTOMATIC_GENERAL,
 AUTOMATIC_SINGLE_SURFACE,
 AUTOMATIC_NODES_TO_SURFACE,
 AUTOMATIC_SURFACE_TO_SURFACE,
 AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,
 ERODING_SINGLE_SURFACE.

EQ.2: For contact types SURFACE_TO_SURFACE and ONE_WAY_SURFACE_TO_SURFACE, the dynamic coefficient of friction points to the table, see DEFINE_TABLE (The table ID is given by FD below.), giving the coefficient of friction as a function of the relative velocity and pressure. This option must be used in combination with the thickness offset option. See Figure 7.1.

FD

Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC|v_{rel}|}$. Give table ID if FS=2.

VARIABLE	DESCRIPTION
	<p>Note: For the special contact option "TIED_SURFACE_TO_SURFACE_FAILURE" only, the variables FS and FD act as failure stresses, i.e.,</p> <p>failure occurs if $\left[\frac{\max(0.0, \sigma_{normal})}{FS} \right]^2 + \left[\frac{\sigma_{shear}}{FD} \right]^2 - 1 > 0$ where σ_{normal} and σ_{shear} are the interface normal and shear stresses.</p>
FS	Normal tensile stress at failure
FD	Shear stress at failure
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$
VC	Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{lim} = VC \cdot A_{cont}$. A_{cont} being the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_o is the yield stress of the contacted material.
VDC	Viscous damping coefficient in percent of critical. In order to avoid undesirable oscillation in contact, e.g., for sheet forming simulation, a contact damping perpendicular to the contacting surfaces is applied. Damping coefficient $\xi = \frac{VDC}{100} \xi_{crit}$, eg VDC = 20. ξ_{crit} is determined in the following fashion by LS-DYNA. $\xi_{crit} = 2mw; \quad m = \min(m_{slave}, m_{master}) \quad \begin{array}{l} \text{mass of master} \\ \text{resp. slave node} \end{array}$ $w = \sqrt{k \cdot \frac{m_{slave} + m_{master}}{m_{slave} \cdot m_{master}}} \quad k \text{ interface stiffness}$
PENCHK	Small penetration in contact search option. If the slave node penetrates more than the segment thickness times the factor XPENE, see *CONTROL_CONTACT, the penetration is ignored and the slave node is set free. The thickness is taken as the shell thickness if the segment belongs to a shell element or it is taken as 1/20 of its shortest diagonal if the segment belongs to a solid element. This option applies to the surface-to-surface contact algorithms: See Table 7.1 for contact types and more details.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.0: check is turned off, EQ.1: check is turned on, EQ.2: check is on but shortest diagonal is used.
BT	Birth time (contact surface becomes active at this time). EQ.0: Birth time is inactive, i.e., contact is always active LT.0: Birth time, BT , is active during dynamic relaxation, and after dynamic relaxation contact is always active GT.0: If DT>0, birth time applies both during and after dynamic relaxation.
DT	Death time (contact surface is deactivated at this time). LT.0: Birth time, BT , is active after dynamic relaxation is completed and DT= DT . During dynamic relaxation, the contact is inactive. EQ.0: DT defaults to 1.E+20.

Card 3 is mandatory for all contact types.

Card 3 1 2 3 4 5 6 7 8

Variable	SFS	SFM	SST	MST	SFST	SFMT	FSF	VSF
Type	F	F	F	F	F	F	F	F
Default	1.	1.	element thickness	element thickness	1.	1.	1.	1.

VARIABLE**DESCRIPTION**

SFS	Scale factor on default slave penalty stiffness, see also *CONTROL_CONTACT.
SFM	Scale factor on default master penalty stiffness, see also *CONTROL_CONTACT.
SST	Optional thickness for slave surface (overrides true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements. For the *CONTACT_TIED... options, SST and MST below can be defined as negative values, which will cause the determination of whether or not a node is tied to depend only on the separation distance relative to the absolute value of these thicknesses. More information is given under <u>General Remarks on *CONTACT</u> following Optional Card C.
MST	Optional thickness for master surface (overrides true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements. For the TIED options see SST above.
SFST	Scale factor for slave surface thickness (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
SFMT	Scale factor for master surface thickness (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
FSF	Coulomb friction scale factor. The Coulomb friction value is scaled as $\mu_{sc} = FSF \cdot \mu_c$, see above.
VSF	Viscous friction scale factor. If this factor is defined then the limiting force becomes: $F_{lim} = VSF \cdot VC \cdot A_{cont}$, see above.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

Remarks:

The variables FSF and VSF above can be overridden segment by segment on the *SET_SEGMENT or *SET_SHELL_OPTION cards for the **slave surface only** as A3 and A4, and for the **master surface only** as A1 and A2. See *SET_SEGMENT and *SET_SHELL_OPTION.

This Card 4 is mandatory for:

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK

*CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK

Card 4 1 2 3 4 5 6 7 8

Variable	OPTION	NFLS	SFLS	PARAM	ERATEN	ERATES	CT2CN	
Type	I	F	F	F	F	F	F	
Default	required	required	required	opt=2,6,...,9	opt=7,9	opt=7,9	opt=9	

VARIABLE

DESCRIPTION

OPTION

Response:

EQ.-3: see 3, moments are transferred. SMP only.

EQ.-2: see 2, moments are transferred. SMP only.

EQ.-1: see 1, moments are transferred. SMP only.

EQ.1: slave nodes in contact and which come into contact will permanently stick. Tangential motion is inhibited.

EQ.2: tiebreak is active for nodes which are initially in contact. Until failure, tangential motion is inhibited. If PARAM is set to unity, (1.0) shell thickness offsets are ignored, and the orientation of the shell surfaces is required such that the outward normals point to the opposing contact surface.

EQ.3: as 1 above but with failure after sticking.

EQ.4: tiebreak is active for nodes which are initially in contact but tangential motion with frictional sliding is permitted.

EQ.5: tiebreak is active for nodes which are initially in contact. Stress is limited by the yield condition described in Remark 5 below. Damage is a function of the crack width opening. The damage function is defined by a load curve which starts at unity for a crack width of zero and decays in some way to zero at a given value of the crack opening. This interface can be used to represent deformable glue bonds.

VARIABLE	DESCRIPTION
	<p>EQ.6: This option is for use with solids and thick shells only. Tiebreak is active for nodes which are initially in contact. Failure stress must be defined for tiebreak to occur. After the failure stress tiebreak criterion is met, damage is a linear function of the distance C between points initially in contact. When the distance is equal to PARAM damage is fully developed and interface failure occurs. After failure, this contact option behaves as a surface-to-surface contact.</p> <p>EQ.7: Dycoss Discrete Crack Model. “_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK” definition is recommended for this option. See Remark 7.</p> <p>EQ.8: This is similar to OPTION=6 but works with offset shell elements. “_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK” definition is recommended for this option.</p> <p>EQ.9: Extension of OPTION=7. Discrete Crack Model with power law and B-K damage models. “_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK” definition is recommended for this option. See Remark 8.</p>
NFLS	Normal failure stress for OPTION=2, 3, 4, 6, 7 or 8. For OPTION=5 NFLS becomes the plastic yield stress as defined in Remark 5.
SFLS	Shear failure stress for OPTION=2, 3, 6, 7 or 8. For OPTION=4, SFLS is a frictional stress limit if PARAM=1. This frictional stress limit is independent of the normal force at the tie. For OPTION=5 SFLS becomes the load curve ID of the damage model.
PARAM	For OPTION=2, setting PARAM=1 causes the shell thickness offsets to be ignored. For OPTION=4, setting PARAM=1 causes SFLS to be a frictional stress limit. For OPTION=6 or 8, PARAM is the critical distance, CCRIT, at which the interface failure is complete. For OPTION=7 PARAM is the friction angle in degrees. For OPTION=9, it is the exponent in the damage model. A positive value invokes the power law, while a negative one, the B-K model. See MAT_138 for additional details.
ERATEN	For OPTION=7,9 only. Normal energy release rate used in damage calculation, see Lemmen and Meijer [2001].
ERATES	For OPTION=7,9 only. Shear energy release rate used in damage calculation, see Lemmen and Meijer [2001].
CT2CN	The ratio of the tangential stiffness to the normal stiffness for OPTION=9. The default is 1.0.

Remarks:

1. After failure, this contact option behaves as a surface-to-surface contact with thickness offsets. After failure, no interface tension is possible.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

- The soft constraint option with SOFT=2 is not implemented for the tiebreak option.
- For OPTION = 2, 3, and 6 the tiebreak failure criterion has normal and shear components:

$$\left(\frac{|\sigma_n|}{NFLS}\right)^2 + \left(\frac{|\sigma_s|}{SFLS}\right)^2 \geq 1$$

- For OPTION = 4, the tiebreak failure criterion has only a normal stress component:

$$\frac{|\sigma_n|}{NFLS} \geq 1$$

- For OPTION = 5, the stress is limited by a perfectly plastic yield condition. For ties in tension, the yield condition is

$$\frac{\sqrt{\sigma_n^2 + 3|\sigma_s|^2}}{NLFS} \leq 1$$

For ties in compression, the yield condition is

$$\frac{\sqrt{3|\sigma_s|^2}}{NLFS} \leq 1$$

The stress is also scaled by the damage function which is obtained from the load curve. For ties in tension, both normal and shear stress are scaled. For ties in compression, only shear stress is scaled.

- For OPTION = 6, damage initiates when the stress meets the failure criterion. The stress is then scaled by the damage function. Assuming no load reversals, the energy released due to the failure of the interface is approximately $0.5*S*CCRIT$, where

$$S = \sqrt{\max(\sigma_n, 0)^2 + |\sigma_s|^2}$$

at initiation of damage. This interface may be used for simulating crack propagation. For the energy release to be correct, the contact penalty stiffness must be much larger than

$$\frac{MIN(NFLF, SFLS)}{CCRIT}$$

- OPTION = 7 is the Dycoss Discrete Crack Model as described in Lemmen and Meijer [2001]. The relation for the crack initiation is given as

$$\left(\frac{\max(\sigma_n, 0)}{NFLS}\right)^2 + \left(\frac{\sigma_s}{SFLS(1 - \sin(PARAM) \min(0, \sigma_n))}\right)^2 = 1$$

- OPTION = 9 is an extension of the Dycoss Discrete Crack Model based on the fracture model in the cohesive material model *MAT_COHESIVE_MIXED_MODE, where the model is described in detail.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

This Card 4 is mandatory for:

*CONTACT_DRAWBEAD

*CONTACT_DRAWBEAD_INITIALIZE

Card 4 1 2 3 4 5 6 7 8

Variable	LCIDRF	LCIDNF	DBDTH	DFSCL	NUMINT	DBPID	ELOFF	
Type	I	I	F	F	I	I	I	
Default	required	none	0.0	1.0	0	0	0	

If the option INITIALIZE is active, then define the following card to initialize the plastic strain and thickness of elements that pass under the drawbead.

Optional 1 2 3 4 5 6 7 8

Variable	LCEPS	TSCALE	LCEPS2	OFFSET				
Type	I	F	I	F				
Default	required	1.0	optional	optional				

VARIABLE

DESCRIPTION

LCIDRF

If LCIDRF is positive then it defines the load curve ID giving the bending component of the restraining force, $F_{bending}$, per unit draw bead length as a function of displacement, δ , see Figure 7.2. This force is due to the bending and unbending of the blank as it moves through the draw bead. The total restraining force is the sum of the bending and friction components.

If LCIDRF is negative then the absolute value gives the load curve ID defining max bead force versus normalized draw bead length. The abscissa values are between zero and 1 and is the normalized draw bead length. The ordinate gives the maximum allowed draw bead, retaining force when the bead is in the fully closed position. If the draw bead is not fully closed linear interpolation is used to compute the draw bead force.

VARIABLE	DESCRIPTION
LCIDNF	Load curve ID giving the normal force per unit draw bead length as a function of displacement, δ , see Figure 7.2. This force is due to the bending of the blank into the draw bead as the binder closes on the die and represents a limiting value. The normal force begins to develop when the distance between the die and binder is less than the draw bead depth. <i>As the binder and die close on the blank this force should diminish or reach a plateau, see the Remarks section.</i>
DBDTH	Draw bead depth, see Figure 7.2. Necessary to determine correct δ displacement from contact displacements.
DFSCL	Scale factor for load curve. Default=1.0. This factor scales load curve ID, LCIDRF above.
NUMINT	Number of equally spaced integration points along the draw bead: EQ.0: Internally calculated based on element size of elements that interact with draw bead. This is necessary for the correct calculation of the restraining forces. More integration points may increase the accuracy since the force is applied more evenly along the bead.
DBPID	Optional part ID for the automatically generated truss elements for the draw bead display in the post-processor. If undefined LS-DYNA assigns a unique part ID.
ELOFF	Option to specify and element ID offset for the truss elements that are automatically generated for the draw bead display. If undefined LS-DYNA chooses a unique offset.
LCEPS	Load curve ID defining the plastic strain versus the parametric coordinate through the shell thickness. The parametric coordinate should be defined in the interval between -1 and 1 inclusive. The value of plastic strain at the integration point is interpolated from this load curve. If the plastic strain at an integration point exceeds the value of the load curve at the time initialization occurs, the plastic strain at the point will remain unchanged.
TSCALE	Scale factor that multiplies the shell thickness as the shell element moves under the draw bead.
LCEPS2	Optional load curve ID defining the plastic strain versus the parametric coordinate through the shell thickness, which is used after an element has traveled a distance equal to OFFSET. The parametric coordinate should be defined in the interval between -1 and 1 inclusive. The value of plastic strain at the integration point is interpolated from this load curve. If the plastic strain at an integration point exceeds the value of

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	the load curve at the time initialization occurs, the plastic strain at the point will remain unchanged. Input parameters LCEPS2 and OFFSET provides a way to model the case where a material moves under two draw beads. In this latter case the curve would be the sum of the plastic strains generate by moving under two consecutive beads.
OFFSET	If the center of an element has moved a distance equal to OFFSET, the load curve ID, LCEPS2 is used to reinitialize the plastic strain. The TSCALE scale factor is also applied.

Remarks:

The draw bead is defined three ways:

1. A *consecutive* list of *slave* nodes that lie along the bead.
2. A part ID of a beam that lies along the draw bead.
3. A part set ID of beams that lie along the draw bead.

For straight draw beads only two nodes or a single beam need to be defined, i.e., one at each end, but for curved beads sufficient nodes or beams are required to define the curvature of the bead geometry. When beams are used to define the bead, with the exception of the first and last node, each node must connect with two beam elements. This requirement means that the number of slave nodes equals the number of beam elements plus one. The integration points along the bead are equally spaced and are independent of the nodal spacing used in the definition of the draw bead. By using the capability of tying extra nodes to rigid bodies (see *CONSTRAINED_EXTRA_NODES or *CONSTRAINED_RIGID_BODIES) the draw bead nodal points do not need to belong to the element connectivities of the die and binder. The blank makes up the master surface. IT IS HIGHLY RECOMMENDED TO DEFINE A BOXID AROUND THE DRAWBEAD TO LIMIT THE SIZE OF THE MASTER SURFACE CONSIDERED FOR THE DRAW BEAD. THIS WILL SUBSTANTIALLY REDUCE COST AND MEMORY REQUIREMENTS.

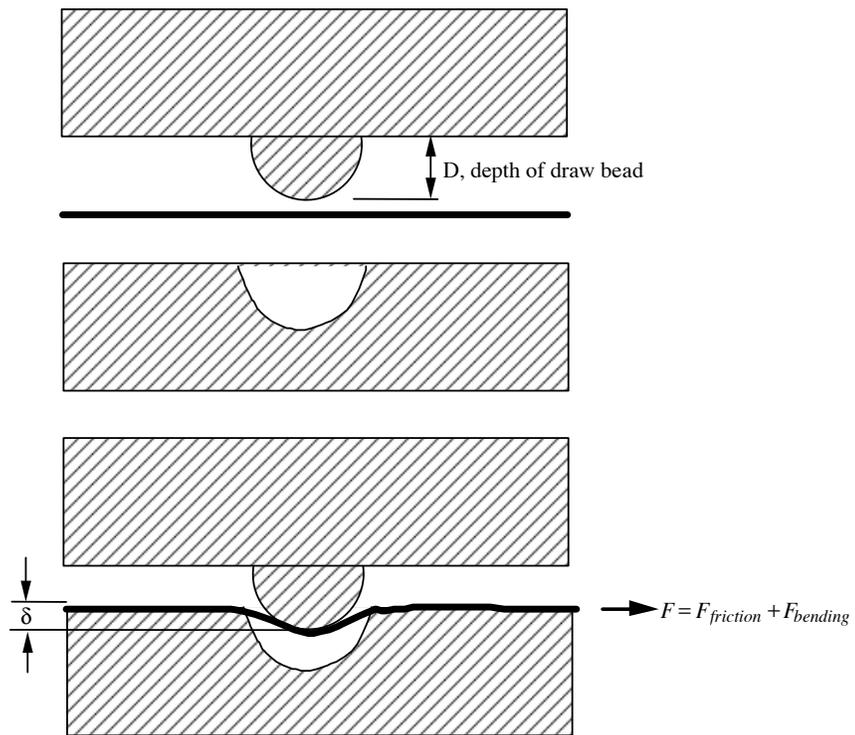


Figure 7.2.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

This Card 4 is mandatory for:

*CONTACT_ERODING_NODES_TO_SURFACE

*CONTACT_ERODING_SINGLE_SURFACE

*CONTACT_ERODING_SURFACE_TO_SURFACE

Card 4 1 2 3 4 5 6 7 8

Variable	ISYM	EROSOP	IADJ					
Type	I	I	I					
Default	0	0	0					

VARIABLE

DESCRIPTION

ISYM

Symmetry plane option:
 EQ.0: off,
 EQ.1: do not include faces with normal boundary constraints (e.g., segments of brick elements on a symmetry plane).
 This option is important to retain the correct boundary conditions in the model with symmetry.

EROSOP

Erosion/Interior node option:
 EQ.0: only exterior boundary information is saved,
 EQ.1: storage is allocated so that eroding contact can occur.
 Otherwise, no contact is assumed after erosion of the corresponding element.

IADJ

Adjacent material treatment for solid elements:
 EQ.0: solid element faces are included only for free boundaries,
 EQ.1: solid element faces are included if they are on the boundary of the material subset. This option also allows the erosion within a body and the subsequent treatment of contact.

Remarks:

Eroding contact may control the timestep (see ECDT in *CONTROL_CONTACT). For ERODING_NODES_TO_SURFACE, define the slave side using a node set, not a part ID or part set ID.

This Card 4 is mandatory for:

*CONTACT_NODES_TO_SURFACE_INTERFERENCE

*CONTACT_ONE_WAY_SURFACE_TO_SURFACE_INTERFERENCE

*CONTACT_SURFACE_TO_SURFACE_INTERFERENCE

Purpose: This contact option provides a means of modeling parts which are shrink fitted together and are, therefore, prestressed in the initial configuration. This option turns off the nodal interpenetration checks (which changes the geometry by moving the nodes to eliminate the interpenetration) at the start of the simulation and allows the contact forces to develop to remove the interpenetrations. The load curves defined in this section scale the interface stiffness constants such that the stiffness can increase slowly from zero to a final value with effect that the interface forces also increase gradually to remove the overlaps.

Card 4 1 2 3 4 5 6 7 8

Variable	LCID1	LCID2						
Type	I	I						
Default	0	0						

VARIABLE

DESCRIPTION

LCID1	Load curve ID which scales the interface stiffness during dynamic relaxation. This curve must originate at (0,0) at time=0 and gradually increase.
LCID2	Load curve ID which scales the interface stiffness during the transient calculation. This curve generally has a constant value of unity for the duration of the calculation if LCID1 is defined. If LCID1=0, this curve must originate at (0,0) at time=0 and gradually increase to a constant value.

Remarks:

Extreme caution must be used with this option. First, shell thickness offsets are taken into account for deformable shell elements. Furthermore, SEGMENT ORIENTATION FOR SHELL ELEMENTS AND INTERPENETRATION CHECKS ARE SKIPPED. Therefore, it is necessary in the problem setup to ensure that all contact segments which belong to shell elements are properly oriented, i.e., the outward normal vector of the segment based on the right hand rule relative to the segment numbering, must point to the opposing contact surface; consequently, automatic contact generation should be avoided for parts composed of shell elements unless automatic generation is used on the slave side of a nodes to surface interface.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

This Card 4 is mandatory for:

***CONTACT_RIGID_NODES_TO_RIGID_BODY**

***CONTACT_RIGID_BODY_ONE_WAY_TO_RIGID_BODY**

***CONTACT_RIGID_BODY_TWO_WAY_TO_RIGID_BODY**

Card 4 1 2 3 4 5 6 7 8

Variable	LCID	FCM	US					
Type	I	I	F					
Default	required	required	from LCID					

VARIABLE

DESCRIPTION

LCID	Load curve ID giving force versus penetration behavior for RIGID_contact. See also the definition of FCM below.
FCM	Force calculation method for RIGID_contact: EQ.1: Load curve gives total normal force on surface versus maximum penetration of any node (RIGID_BODY_ONE_WAY only). EQ.2: Load curve gives normal force on each node versus penetration of node through the surface (all RIGID_contact types). EQ.3: Load curve gives normal pressure versus penetration of node through the surface (RIGID_BODY_TWO_WAY and RIGID_BODY_ONE_WAY only). EQ.4: Load curve gives total normal force versus maximum soft penetration. In this case the force will be followed based on the original penetration point. (RIGID_BODY_ONE_WAY only).
US	Unloading stiffness for RIGID_contact. The default is to unload along the loading curve. This should be equal to or greater than the maximum slope used in the loading curve.

This Card 4 is mandatory for:

***CONTACT_TIEBREAK_NODES_TO_SURFACE and**

***CONTACT_TIEBREAK_NODES_ONLY**

Card 4 1 2 3 4 5 6 7 8

Variable	NFLF	SFLF	NEN	MES				
Type	F	F	F	F				
Default	required	required	2.	2.				

VARIABLE

DESCRIPTION

NFLF	Normal failure force. Only tensile failure, i.e., tensile normal forces, will be considered in the failure criterion.
SFLF	Shear failure force
NEN	Exponent for normal force
MES	Exponent for shear force. Failure criterion:

$$\left(\frac{|f_n|}{NFLF} \right)^{NEN} + \left(\frac{|f_s|}{SFLF} \right)^{MES} \geq 1.$$

Failure is assumed if the left side is larger than 1. f_n and f_s are the normal and shear interface force.

Remarks:

These attributes can be overridden node by node on the *SET_NODE_option cards.

Both NFLF and SFLF must be defined. If failure in only tension or shear is required then set the other failure force to a large value (1E+10).

After failure, the contact_tiebreak_nodes_to_surface behaves as a nodes-to-surface contact with no thickness offsets (no interface tension possible) whereas the contact_tiebreak_nodes_only stops acting altogether. Prior to failure, the two contact types behave identically.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

This Card 4 is mandatory for:

*CONTACT_TIEBREAK_SURFACE_TO_SURFACE

Card 4 1 2 3 4 5 6 7 8

Variable	NFLS	SFLS	TBLCID	THKOFF				
Type	F	F	I	I				
Default	required	required	0	0				

VARIABLE

DESCRIPTION

NFLS Tensile failure stress. See remark below.

SFLS Shear failure stress. Failure criterion:

$$\left(\frac{|\sigma_n|}{NFLS} \right)^2 + \left(\frac{|\sigma_s|}{SFLS} \right)^2 \geq 1.$$

TBLCID Optional load curve number defining the resisting stress versus gap opening for the post failure response. This can be used to model the failure of adhesives.

THKOFF Thickness offsets are considered if THKOFF=1. If shell offsets are included in the meshed geometry, this option is highly recommended since segment orientation can be arbitrary and the contact surfaces can be disjoint.

Remarks:

The failure attributes can be overridden segment by segment on the *SET_SEGMENT or *SET_SHELL_option cards for the **slave surface only** as A1 and A2. These variables do not apply to the master surface.

Both NFLS and SFLS must be defined. If failure in only tension or shear is required then set the other failure stress to a large value (1E+10). When used with shells, contact segment normals are used to establish the tension direction (as opposed to compression). Compressive stress does not contribute to the failure equation.

After failure, this contact option behaves as a surface-to-surface contact with no thickness offsets. After failure, no interface tension is possible.

This Card 4 is mandatory for:

***CONTACT_SURFACE_TO_SURFACE_CONTRACTION_JOINT**

Purpose: This contact option turns on the contraction joint model designed to simulate the effects of sinusoidal joint surfaces (shear keys) in the contraction joints of arch dams and other concrete structures. The sinusoidal functions for the shear keys are defined according to the following three methods [Solberg and Noble 2002]:

Method 1: $\hat{g} = g - A(1 - \cos(B(s_2 - s_1)))$

Method 2: $\hat{g} = g - 2A|\sin(B(s_2 - s_1)/2)|$

Method 3: (default) $\hat{g} = g - A\cos(Bs_2) + A\cos(Bs_1)$

Where g is gap function for contact surface, \hat{g} is gap function for the joint surface. A is key amplitude parameter, and B is key frequency parameter. s_1 and s_2 are referential surfaces:

$$s_1 = \mathbf{X}_{surface1} * \mathbf{T}_{key}$$

$$s_2 = \mathbf{X}_{surface2} * \mathbf{T}_{key}$$

$$\mathbf{T}_{key} = \mathbf{T}_{slide} \times \mathbf{n}$$

Where \mathbf{T}_{slide} is the free sliding direction of the keys, \mathbf{n} is the surface normal in reference.

Card 4 1 2 3 4 5 6 7 8

Variable	MTCJ	ALPHA	BETA	TSVX	TSVY	TSVZ		
Type	I	F	F	F	F	F		
Default	0	0.0	0.0	0.0	0.0	0.0		

VARIABLE

DESCRIPTION

MTCJ	The method option for the gap function, \hat{g}
ALPHA	Key amplitude parameter A
BETA	Key frequency parameter B
TSVX	X component of the free sliding direction \mathbf{T}_{slide}
TSVY	Y component of the free sliding direction \mathbf{T}_{slide}
TSVZ	Z component of the free sliding direction \mathbf{T}_{slide}

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

This Card is mandatory for the THERMAL option, i.e.,:

Reminder: If Card 4 is required, then it must go before this optional card. (Card 4 is required for certain contact types - see earlier in this section for the list, later in this section for details of Card 4.)

*CONTACT_ ..._THERMAL_.....

Optional	1	2	3	4	5	6	7	8
Variable	K	FRAD	H0	LMIN	LMAX	CHLM	BC_FLG	ALGO
Type	F	F	F	F	F	F	I	I
Default	none	none	none	none	none	1.0	0	0

VARIABLE	DESCRIPTION
----------	-------------

K	Thermal conductivity of fluid between the contact surfaces. If a gap with a thickness l_{gap} exists between the contact surfaces, then the conductance due to thermal conductivity between the contact surfaces is
---	---

$$h_{cond} = \frac{K}{l_{gap}}$$

Note that LS- DYNA calculates l_{gap} based on deformation.

FRAD	Radiation factor between the contact surfaces.
------	--

$$f_{rad} = \frac{\sigma}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1}$$

Where: σ = Stefan Boltzman constant
 ϵ_1 = emissivity of master surface
 ϵ_2 = emissivity of slave surface

LS-DYNA calculates a radiant heat transfer conductance

$$h_{rad} = f_{rad} (T_m + T_s)(T_m^2 + T_s^2)$$

VARIABLE	DESCRIPTION
H0	Heat transfer conductance for closed gaps. Use this heat transfer conductance for gaps in the range $0 \leq l_{gap} \leq l_{min}$
LMIN	Minimum gap (l_{min}), use the heat transfer conductance defined (H0) for gap thicknesses less than this value. If $l_{min} < 0$, then $abs(l_{min})$ is a load curve number defining l_{min} vs. time.
LMAX	No thermal contact if gap is greater than this value (l_{max}).
CHLM	Is a multiplier used on the element characteristic distance for the search routine. The characteristic length is the largest interface surface element diagonal. EQ.0: Default set to 1.0
BC_FLAG	Thermal boundary condition flag EQ.0: thermal boundary conditions are on when parts are in contact EQ.1: thermal boundary conditions are off when parts are in contact
ALGO	Contact algorithm type. EQ.0: two way contact, both surfaces change temperature due to contact EQ.1: one way contact, master surface does not change temperature due to contact. Slave surface does change temperature.

Remarks:

In summary:

$$h = h_0, \text{ if the gap thickness is } 0 \leq l_{gap} \leq l_{min}$$

$$h = h_{cond} + h_{rad}, \text{ if the gap thickness is } l_{min} \leq l_{gap} \leq l_{max}$$

$$h = 0, \text{ if the gap thickness is } l_{gap} > l_{max}$$

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

*CONTACT_ ..._THERMAL_FRICTION_ ...

WARNING: This is only implemented for the keyword

*CONTACT_SURFACE_TO_SURFACE_THERMAL_FRICTION. Check with LSTC for information on other contact types.

This card is required if the FRICTION suffix is added to THERMAL. The keyword *LOAD_SURFACE_STRESS must be used so that LS-DYNA will calculate contact pressure which is needed in the h(P) formulas. The blank (or work piece) must be defined as the slave surface in a metal forming model.

Purpose:

1. Used to define the mechanical static and dynamic friction coefficients as a function of temperature.
2. Used to define the thermal contact conductance as a function of temperature and pressure.

Card 1 1 2 3 4 5 6 7 8

Variable	LCFST	LCFDT	FORMULA	a	b	c	d	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0.	0	0	

VARIABLE

DESCRIPTION

LCFST	Load curve number for static coefficient of friction as a function of temperature. The load curve value multiplies the coefficient value FS.
LCFDT	Load curve number for dynamic coefficient of friction as a function of temperature. The load curve value multiplies the coefficient value FD.
FORMULA	Formula that defines the contact heat conductance as a function of temperature and pressure.

Note that the following coefficients take on different definitions based on their use.

- a Load curve number for the “a” coefficient used in the formula.
- b Load curve number for the “b” coefficient used in the formula.
- c Load curve number for the “c” coefficient used in the formula.
- d Load curve number for the “d” coefficient used in the formula.

Remarks:

FORMULA = 1

 $h(P)$ is defined by load curve “a”

“a” defines a load curve for the contact conductance as a function of interface pressure.

FORMULA = 2

$$h(P) = a + bP + cP^2 + dP^3$$

Although defined by load curves, the coefficients a, b, c, and d are typically constants for use in this formula. The load curves are functions of temperature.

FORMULA = 3

$$h(P) = \frac{\pi k_{gas}}{4\lambda} \left[1 + 85 \left(\frac{P}{\sigma} \right)^{0.8} \right] = \frac{a}{b} \left[1 + 85 \left(\frac{P}{c} \right)^{0.8} \right]$$

The above formula is from [Shvets and Dyban 1964].

“a” defines a load curve for the thermal conductivity (k_{gas}) of the gas in the gap as a function of temperature.

“b” defines a load curve for the parameter grouping $\pi/4\lambda$. Therefore, this load curve should be set to a constant value. λ is the surface roughness.

“c” defines a stress metric for deformation (e.g., yield) as a function of temperature.

FORMULA = 4

$$h(P) = a \left[1 - \exp\left(-b \frac{P}{c}\right) \right]^d$$

The above formula is from [Li and Sellars 1996].

“a” defines a load curve as a function of temperature.

“b” defines a load curve as a function of temperature.

“c” defines a stress metric for deformation (e.g., yield) as a function of temperature.

“d” defines a load curve as a function of temperature.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

Optional Card A

Reminder: If Card 4 is required, then it must go before this optional card. (Card 4 is required for certain contact types - see earlier in this section for the list, later in this section for details of Card 4.)

Optional Card A	1	2	3	4	5	6	7	8
Variable	SOFT	SOFSCS	LCIDAB	MAXPAR	SBOPT	DEPTH	BSORT	FRCFRQ
Type	I	F	I	F	F	I	I	I
Default	0	.1	0	1.025.	0.	2	10-100	1
Remarks			type a13					

VARIABLE

DESCRIPTION

SOFT

Soft constraint option:

- EQ.0: penalty formulation,
- EQ.1: soft constraint formulation,
- EQ.2: segment-based contact.
- EQ.4: constraint approach for FORMING contact option.

The soft constraint may be necessary if the material constants of the elements which make up the surfaces in contact have a wide variation in the elastic bulk moduli. In the soft constraint option, the interface stiffness is based on the nodal mass and the global time step size. This method of computing the interface stiffness will typically give much higher stiffness value than would be obtained by using the bulk modulus; therefore, this method the preferred approach when soft foam materials interact with metals. See the remark below for the segment-based penalty formulation.

SOFSCS

Scale factor for constraint forces of soft constraint option (default=.10). Values greater than .5 for single surface contact and 1.0 for a one-way treatment are inadmissible.

LCIDAB

Load curve ID defining airbag thickness as a function of time for type a13 contact (*CONTACT_AIRBAG_SINGLE_SURFACE).

VARIABLE	DESCRIPTION
MAXPAR	<p>Maximum parametric coordinate in segment search (values 1.025 and 1.20 recommended). Larger values can increase cost. If zero, the default is set to 1.025 for most contact options. Other defaults are:</p> <p>EQ.1.006:SPOTWELD, EQ.1.006:TIED_SHELL_..._CONSTRAINED_OFFSET, EQ.1.006:TIED_SHELL_..._OFFSET, EQ.1.006:TIED_SHELL_...:BEAM_OFFSET, EQ.1.100:AUTOMATIC_GENERAL</p> <p>This factor allows an increase in the size of the segments which may be useful at sharp corners. For the SPOTWELD and ..._OFFSET options larger values can sometimes lead to numerical instabilities; however, a larger value is sometimes necessary to ensure that all nodes of interest are tied.</p>
SBOPT	<p>Segment-based contact options (SOFT=2).</p> <p>EQ.0: defaults to 2. EQ.1: pinball edge-edge contact (not recommended) EQ.2: assume planer segments (default) EQ.3: warped segment checking EQ.4: sliding option EQ.5: do options 3 and 4</p>
DEPTH	<p>Search depth in automatic contact. Value of 1 is sufficiently accurate for most crash applications and is much less expensive. LS-DYNA for improved accuracy sets this value to 2. If zero, the default is set to 2.</p> <p>LT.0: DEPTH is the load curve ID defining searching depth versus time.</p> <p>See remarks below for segment-based contact options controlled by DEPTH.</p>
BSORT	<p>Number of cycles between bucket sorts. Values of 25 and 100 are recommended for contact types 4 and 13 (SINGLE_SURFACE), respectively. Values of 10-15 are okay for the surface to surface and node to surface contact. If zero, LS-DYNA determines the interval.</p> <p>LT.0: BSORT load curve ID defining bucket sorting frequency versus time.</p>
FRFCRQ	<p>Number of cycles between contact force updates for penalty contact formulations. This option can provide a significant speed-up of the contact treatment. If used, values exceeding 3 or 4 are dangerous. Considerable care must be exercised when using this option, as this option assumes that contact does not change FRFCRQ cycles.</p> <p>EQ.0: FRFCRQ is set to 1 and force calculations are performed each cycle-strongly recommended.</p>

Remarks:

Setting SOFT=1 or 2 on optional contact card A will cause the contact stiffness to be determined based on stability considerations, taking into account the time step and nodal masses. This approach is generally more effective for contact between materials of dissimilar stiffness or dissimilar mesh densities.

SOFT=2 is for general shell and solid element contact. This option is available for all SURFACE_TO_SURFACE, ONE_WAY_SURFACE_TO_SURFACE, and SINGLE_SURFACE options including eroding and airbag contact. When the AUTOMATIC option is used, orientation of shell segment normals is automatic. When the AUTOMATIC option is not used, the segment or element orientations are used as input. The segment-based penalty formulation contact algorithm checks for segments vs. segment penetration rather than node vs. segment. After penetrating segments are found, an automatic judgment is made as to which is the master segment, and penalty forces are applied normal to that segment. The user may override this automatic judgment by using the ONE_WAY options in which case the master segment normals are used as input by the user. All parameters on the first three cards are active except for VC, and VSF. On optional card A, some parameters have different meanings than they do for the default contact.

For SOFT=2, the SBOPT parameter on optional card A controls several options. The pinball edge-to-edge checking is not recommended and is included only for back compatibility. For edge-to-edge checking setting DEPTH=5 is recommended instead (see below). The warped segment option more accurately checks for penetration of warped surfaces. The sliding option uses neighbor segment information to improve sliding behavior. It is primarily useful for preventing segments from incorrectly catching nodes on a sliding surface.

For SOFT=2, the DEPTH parameter controls several additional options for segment based contact. When DEPTH=2 (default), surface penetrations measured at nodes are checked. When DEPTH=3, surface penetration may also be measured at the edge. This option is useful mainly for airbags. When DEPTH=5, both surface penetrations and edge-to-edge penetration is checked. The airbag contact has two additional options, DEPTH=1 and 4. DEPTH=4 activates additional airbag logic that uses neighbor segment information when judging if contact is between interior or exterior airbag surfaces. This option is not recommended and is maintained only for backward compatibility. Setting DEPTH=1 suppresses all airbag logic.

For SOFT=2 contact, the MAXPAR has a totally different use. Positive values of MAXPAR are ignored. If a negative value is input for MAXPAR, the absolute value of MAXPAR will be used as an assumed time step for scaling the contact stiffness. This option is useful for maintaining consistent contact behavior of an airbag deployment when a validated airbag is inserted into an automobile model. For the new run, setting MAXPAR=the negative of the solution time step of the validated run will cause the airbag contact stiffness to be unchanged in the new run even if the solution time step of the new run is smaller.

For SOFT=2 contact, only the ISYM, I2D3D, SLDTHK, and SLDSTF parameters are active on optional card B.

Optional Card B

Reminder: If Optional Card B is used, then Optional Card A must be defined. (Optional Card A may be a blank line).

Optional Card B	1	2	3	4	5	6	7	8
Variable	PENMAX	THKOPT	SHLTHK	SNLOG	ISYM	I2D3D	SLDTHK	SLDSTF
Type	F	I	I	I	I	I	F	F
Default	0	0	0	0	0	0	0	0
Remarks		Old types 3, 5, 10	Old types 3, 5, 10					

VARIABLE**DESCRIPTION**

PENMAX

Maximum penetration distance for old type 3, 5, 8, 9, and 10 contact or the segment thickness multiplied by PENMAX defines the maximum penetration allowed (as a multiple of the segment thickness) for contact types a 3, a 5, a10, 13, 15, and 26. (see discussion at end of section, including Table 7.1):

EQ.0.0 for old type contacts 3, 5, and 10: Use small penetration search and value calculated from thickness and XPENE, see *CONTROL_CONTACT.

EQ.0.0 for contact types a 3, a 5, a10, 13, and 15: Default is 0.4, or 40 percent of the segment thickness

EQ.0.0 for contact type26: Default is 200.0 times the segment thickness

THKOPT

Thickness option for contact types 3, 5, and 10:

EQ.0: default is taken from control card, *CONTROL_CONTACT,

EQ.1: thickness offsets are included,

EQ.2: thickness offsets are not included (old way).

SHLTHK

Define if and only if THKOPT above equals 1. Shell thickness considered in type surface to surface and node to surface type contact options, where options 1 and 2 below activate the new contact algorithms. The thickness offsets are always included in single surface and constraint method contact types:

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.0: thickness is not considered, EQ.1: thickness is considered but rigid bodies are excluded, EQ.2: thickness is considered including rigid bodies.
SNLOG	Disable shooting node logic in thickness offset contact. With the shooting node logic enabled, the first cycle that a slave node penetrates a master segment, that node is moved back to the master surface without applying any contact force. EQ.0: logic is enabled (default), EQ.1: logic is skipped (sometimes recommended for metalforming calculations or for contact involving foam materials).
ISYM	Symmetry plane option: EQ.0: off, EQ.1: do not include faces with normal boundary constraints (e.g., segments of brick elements on a symmetry plane). This option is important to retain the correct boundary conditions in the model with symmetry. For the <code>_ERODING_</code> contacts this option may also be defined on card 4.
I2D3D	Segment searching option: EQ.0: search 2D elements (shells) before 3D elements (solids, thick shells) when locating segments. EQ.1: search 3D (solids, thick shells) elements before 2D elements (shells) when locating segments.
SLDTHK	Optional solid element thickness. A nonzero positive value will activate the contact thickness offsets in the contact algorithms where offsets apply. The contact treatment will then be equivalent to the case where null shell elements are used to cover the brick elements. The contact stiffness parameter below, <code>SLDSTF</code> , may also be used to override the default value.
SLDSTF	Optional solid element stiffness. A nonzero positive value overrides the bulk modulus taken from the material model referenced by the solid element.

Optional Card C

Reminder: If Optional Card C is used, then Optional Cards A and B must be defined. (Optional Cards A and B may be blank lines).

Optional Card C	1	2	3	4	5	6	7	8
Variable	IGAP	IGNORE	DPRFAC	DTSTIF			FLANGL	
Type	I	I	F	F			F	
Default	1	0	0	0			0	
Remarks								

VARIABLE**DESCRIPTION**

IGAP	Flag to improve implicit convergence behavior at the expense of creating some sticking if parts attempt to separate. (IMPLICIT ONLY) EQ.1: apply method to improve convergence (DEFAULT) EQ.2: do not apply method
IGNORE	Ignore initial penetrations in the *CONTACT_AUTOMATIC options. EQ.0: Take the default value from the fourth card of the CONTROL_CONTACT input. EQ.1: Allow initial penetrations to exist by tracking the initial penetrations. EQ.2: Allow initial penetrations to exist by tracking the initial penetrations. However, penetration warning messages are printed with the original coordinates and the recommended coordinates of each slave node given.
DPRFAC	Depth of penetration reduction factor for SOFT=2 contact. EQ.0.0: Initial penetrations are always ignored. GT.0.0: Initial penetrations are penalized over time. LT.0.0: DPRFAC is the load curve ID defining DPRFAC versus time.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

VARIABLE	DESCRIPTION
DTSTIF	Time step used in stiffness calculation for SOFT=1 and SOFT=2 contact. EQ.0.0: Use the initial value that is used for time integration. GT.0.0: Use the value specified. LT.0.0: DTSTIF is the load curve ID defining DTSTIF versus time.
FLANGL	Angle tolerance in radians for feature lines option in smooth contact. EQ.0.0: No feature line is considered for surface fitting in smooth contact. GT.0.0: Any edge with angle between two contact segments bigger than this angle will be treated as feature line during surface fitting in smooth contact.

Remarks:

1. DPRFAC is used only by segment based contact (SOFT=2). By default, SOFT=2 contact measures the initial penetration between segment pairs that are found to be in contact and subtracts the measured value from the total penetration for as long as a pair of segments remains in contact. The penalty force is proportional to this modified value. This approach prevents shooting nodes, but may allow unacceptable penetration. DPRFAC can be used to decrease the measured value over time until the full penetration is penalized. Setting DPRFAC=0.01 will cause ~1% reduction in the measured value each cycle. The maximum allowable value for DPRFAC is 0.1. A small value such as 0.001 is recommended. DPRFAC does not apply to initial penetrations at the start of the calculation, only those that are measured at later times. This prevents nonphysical movement and energy growth at the start of the calculation.

The anticipated use for the load curve option is to allow the initial penetrations to be reduced at the end of a calculation if the final geometry is to be used for a subsequent analysis. To achieve this, load curve should have a y-value of zero until a time near the end of the analysis and then ramp up to a positive value such as 0.01 near the end of the analysis.

2. DTSTIF is used only by the SOFT=1 and SOFT=2 contact options. By default when the SOFT option is active, the contact uses the initial solution time step to scale the contact stiffness. If the user sets DTSIFF larger than the initial solution time step, the larger value will be used. A smaller value will be ignored. Because the square of the time step appears in the denominator of the stiffness calculation, a larger value reduces the contact stiffness. This option could be used when one component of a larger model has been analyzed independently and validated. When the component is inserted into the larger model, the larger model may run at a smaller time step due to higher mesh frequencies. In the full model analysis, setting DTSTIF equal to the component analysis time step for the contact interface that treats the component will cause consistent contact stiffness between the analyses.

The load curve option allows a variable contact stiffness. This should be done with care as energy will not be conserved. A special case of the load curve option is when |DTSTIF| = LCTM on *CONTROL_CONTACT. LCTM sets an upper bound on the

solution time step. For $|DTSTIF| = LCTM$, the contact stiffness time step value will track LCTM whenever the LCTM value is less than the initial solution time step. If the LCTM value is greater, the initial solution time step is used. This option could be used to stiffen the contact at the end of an analysis. To achieve this, the LCTM curve should be defined such that it is larger than the solution time step until near the end of the analysis. Then the LCTM curve should ramp down below the solution time step causing it to decrease and the contact to stiffen. A load curve value of 0.1 of the calculated solution time step will cause penetrations to reduce by about 99%. To prevent shooting nodes, the rate at which the contact stiffness increases is automatically limited. Therefore, to achieve 99% reduction, the solution should be run for perhaps 1000 cycles with a small time step.

General Remarks on *CONTACT:

1. Modeling airbag interactions with structures and occupants using the actual fabric thickness, which is approximate 0.30 mm, may result in a contact breakdown that leads to inconsistent occupant behavior between different machines. Based on our experience, using a two-way automatic type contact definition, i.e., AUTOMATIC_SURFACE_TO_SURFACE, between any airbag to structure/occupant interaction and setting the airbag fabric contact thickness to at least 10 times the actual fabric thickness has helped improved contact behavior and eliminates the machine inconsistencies. Due to a large stiffness difference between the airbag and the interacting materials, the soft constraint option (SOFT=1) or the segment based pinball option (SOFT=2) is recommended. It must be noted that with the above contact definition, only the airbag materials should be included in any *AIRBAG_SINGLE_SURFACE definitions to avoid duplicate contact treatment that can lead to numerical instabilities.
2. TIED_NODES_TO_SURFACE
TIED_SHELL_EDGE_TO_SURFACE
TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET
SPOTWELD
SPOTWELD_WITH_TORSION
TIED_SURFACE_TO_SURFACE

These contact definitions are based on constraint equations and will not work with rigid bodies. However, tied interfaces with the offset option can be used with rigid bodies, i.e.,

TIED_NODES_TO_SURFACE_OFFSET
TIED_SHELL_EDGE_TO_SURFACE_OFFSET
TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET
TIED_SURFACE_TO_SURFACE_OFFSET

Also, it may sometimes be advantageous to use the CONSTRAINED_EXTRA_NODE_OPTION instead for tying deformable nodes to rigid bodies since in this latter case the tied nodes may be an arbitrary distance away from the rigid body.

Tying will only work if the surfaces are near each other. The criteria used to determine whether a slave node is tied down is that it must be “close”. For shell elements “close” is defined as distance, δ , less than:

$$\delta_1 = 0.60 * (\textit{thickness_slave_node} + \textit{thickness_master_segment})$$

$$\delta_2 = 0.05 * \min(\textit{master_segment_diagonals})$$

$$\delta = \max(\delta_1, \delta_2)$$

If a node is further away it will not be tied and a warning message will be printed. For solid elements the slave node thickness is zero and the segment thickness is the element volume divided by the segment area; otherwise, the same procedure is used.

If there is a large difference in element areas between the master and slave side, the distance, δ_2 , may be too large and may cause the unexpected projection of nodes that

should not be tied. This can occur during calculation when adaptive remeshing is used. To avoid this difficulty the slave and master thickness can be specified as negative values on Card 3 in which case

$$\delta = abs(\delta_1)$$

3. The contact algorithm for tying spot welds with torsion, SPOTWELD_WITH_TORSION, must be used with care. Parts that are tied by this option should be subjected to stiffness proportional damping of approximately ten percent, i.e., input a coefficient of 0.10. This can be defined for each part on the *DAMPING_PART_STIFFNESS input. Stability problems may arise with this option if damping is not used.

4. CONSTRAINT_NODES_TO_SURFACE
CONSTRAINT_SURFACE_TO_SURFACE

These contact definitions must be used with care. The surface and the nodes which are constrained to a surface are not allowed to be used in any other CONSTRAINT_... contact definition. If, however, contact has to be defined from both sides as in sheet metalforming, one of these contact definitions can be a CONSTRAINT_ type; the other one could be a standard penalty type such as SURFACE_TO_SURFACE or NODES_TO_SURFACE.

5. AIRBAG_SINGLE_SURFACE
AUTOMATIC_GENERAL
AUTOMATIC_GENERAL_INTERIOR
AUTOMATIC_NODES_TO_SURFACE
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
AUTOMATIC_SINGLE_SURFACE
AUTOMATIC_SURFACE_TO_SURFACE
SINGLE_SURFACE

These contact definitions require thickness to be taken into account for rigid bodies modeled with shell elements. Therefore, care should be taken to ensure that realistic thicknesses are specified for the rigid body shells. A thickness that is too small may result in loss of contact and an unrealistically large thickness may result in a degradation in speed during the bucket sorts as well as nonphysical behavior. The SHLTHK option on the *CONTROL_CONTACT card is ignored for these contact types.

6. Two methods are used in LS-DYNA for projecting the contact surface to account for shell thicknesses. The choice of methods can influence the accuracy and cost of the calculation. Segment based projection is used in contact types:

AIRBAG_SINGLE_SURFACE
AUTOMATIC_GENERAL
AUTOMATIC_NODES_TO_SURFACE
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
AUTOMATIC_SINGLE_SURFACE
AUTOMATIC_SURFACE_TO_SURFACE

FORMING_NODES_TO_SURFACE
FORMING_ONE_WAY_SURFACE_TO_SURFACE
FORMING_SURFACE_TO_SURFACE

The remaining contact types use nodal normal projections if projections are used. The main advantage of nodal projections is that a continuous contact surface is obtained which is much more accurate in applications such as metal forming. The disadvantages of nodal projections are the higher costs due to the nodal normal calculations, difficulties in treating T-intersections and other geometric complications, and the need for consistent orientation of contact surface segments. The contact type:

SINGLE_SURFACE

uses nodal normal projections and consequently is slower than the alternatives.

7. FORCE_TRANSDUCER_PENALTY
FORCE_TRANSDUCER_CONSTRAINT

This contact allows the total contact forces applied by all contacts to be picked up. This contact does not apply any force to the model and will have no effect on the solution. Only the slave set and slave set type need be defined for this contact type. Generally, only the first three cards are defined. The force transducer option, `_PENALTY`, works with penalty type contact algorithms only, i.e., it does not work with the `CONSTRAINT` or `TIED` options. For these latter options, use the `_CONSTRAINT` option. *If the interactions between two surfaces are needed, a master surface should be defined. In this case, only the contact forces applied between the slave and master surfaces are kept. The master surface option is only implemented for the `_PENALTY` option and works only with the `AUTOMATIC` contact types.*

8. FORMING_... These contacts are mainly used for metal forming applications. A connected mesh is not required for the master (tooling) side but the orientation of the mesh **must** be in the same direction. These contact types are based on the `AUTOMATIC` type contacts and consequently the performance is better than the original two surface contacts.

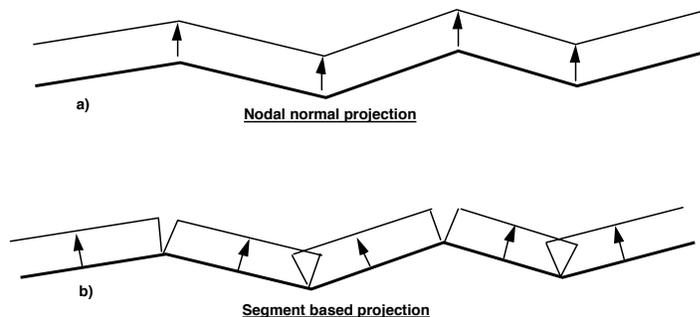


Figure 7.3. Nodal normal and segment based projection is used in the contact options.

INTERFACE TYPE ID	PENCHK	ELEMENT TYPE	FORMULA FOR RELEASE OF PENETRATING NODAL POINT
1, 2, 6, 7 3, 5, 8, 9, 10 (without thickness)	0	solid	d=PENMAX if and only if PENMAX>0 d=1.e+10 if PENMAX=0
		shell	d=PENMAX if and only if PENMAX>0 d=1.e+10 if PENMAX=0
	1	solid	d=XPENE*thickness of solid element
		shell	d=XPENE*thickness of shell element
	2	solid	d=0.05*minimum diagonal length
		shell	d=0.05*minimum diagonal length
3, 5, 10 (thickness) 17, and 18 a3, a5, a10, 13, 15	————	solid	d=XPENE*thickness of solid element
		shell	d=XPENE*thickness of shell element
	————	solid	d=PENMAX*thickness of solid element [default: PENMAX=0.5]
		shell	d=PENMAX*(slave thickness+master thickness) [default: PENMAX=0.4]
4	————	solid	d=0.5*thickness of solid element
		shell	d=0.4*(slave thickness+master thickness)
26	————	solid	d=PENMAX*thickness of solid element [default: PENMAX=200.0]
		shell	d=PENMAX*(slave thickness+master thickness) [default: PENMAX=200.]

Table 7.1. Criterion for node release for nodal points which have penetrated too far. Larger penalty stiffnesses are recommended for the contact interface which allows nodes to be released. For node-to-surface type contacts (5, 5a) the element thicknesses which contain the node determines the nodal thickness. The parameter is defined on the *CONTROL_CONTACT input.

*CONTACT

*CONTACT_OPTION1_{OPTION2}...

The keyword options for the contact type and the corresponding Version 92X, 93X, 94X, 95X type numbers are:

STRUCTURED INPUT TYPE ID	KEYWORD NAME
a13	AIRBAG_SINGLE_SURFACE
26	AUTOMATIC_GENERAL
i26	AUTOMATIC_GENERAL_INTERIOR
a 5	AUTOMATIC_NODES_TO_SURFACE
a 5	AUTOMATIC_NODES_TO_SURFACE_TIEBREAK
a10	AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
13	AUTOMATIC_SINGLE_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK
18	CONSTRAINT_NODES_TO_SURFACE
17	CONSTRAINT_SURFACE_TO_SURFACE
23	DRAWBEAD
16	ERODING_NODES_TO_SURFACE
14	ERODING_SURFACE_TO_SURFACE
15	ERODING_SINGLE_SURFACE
27	FORCE_TRANSDUCER_CONSTRAINT
25	FORCE_TRANSDUCER_PENALTY
m 5	FORMING_NODES_TO_SURFACE
m10	FORMING_ONE_WAY_SURFACE_TO_SURFACE
m 3	FORMING_SURFACE_TO_SURFACE
5	NODES_TO_SURFACE
5	NODES_TO_SURFACE_INTERFERENCE
10	ONE_WAY_SURFACE_TO_SURFACE
20	RIGID_NODES_TO_RIGID_BODY
21	RIGID_BODY_ONE_WAY_TO_RIGID_BODY
19	RIGID_BODY_TWO_WAY_TO_RIGID_BODY
22	SINGLE_EDGE
4	SINGLE_SURFACE
1	SLIDING_ONLY
p 1	SLIDING_ONLY_PENALTY
3	SURFACE_TO_SURFACE
3	SURFACE_TO_SURFACE_INTERFERENCE

STRUCTURED INPUT TYPE ID	KEYWORD NAME
8	TIEBREAK_NODES_TO_SURFACE
9	TIEBREAK_SURFACE_TO_SURFACE
6	TIED_NODES_TO_SURFACE
o 6	TIED_NODES_TO_SURFACE_OFFSET
7	TIED_SHELL_EDGE_TO_SURFACE
7	SPOTWELD
s 7	SPOTWELD_WITH_TORSION
2	TIED_SURFACE_TO_SURFACE
o 2	TIED_SURFACE_TO_SURFACE_OFFSET

***CONTACT**

***CONTACT_OPTION1_{OPTION2}...**

CONTACT EXAMPLES

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *CONTACT_NODES_TO_SURFACE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Make a simple contact that prevents the nodes in part 2 from
$ penetrating the segments in part 3.
$
*CONTACT_NODES_TO_SURFACE
$
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$      ssid      msid      sstyp      mstyp      sboxid      mboxid      spr      mpr
$              2          3          3          3
$
$      fs      fd      dc      vc      vdc      penchk      bt      dt
$
$      sfs      sfm      sst      mst      sfst      sfmt      fsf      vsf
$
$
$ sstype, mstype = 3 id's specified in ssid and msid are parts
$      ssid = 2 use slave nodes in part 2
$      msid = 3 use master segments in part 3
$
$ Use defaults for all parameters.
$
$$$$ Optional Cards A and B not specified (default values will be used).
$
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```


*CONTACT

*CONTACT_OPTION1_{OPTION2}...

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *CONTACT_DRAWBEAD
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define a draw bead contact:
$ - the draw bead is to be made from the nodes specified in node set 2
$ - the master segments are to be those found in the box defined by box 2
$ that are in part 18
$ - include slave and master forces in interface file (spr, mpr = 1)
$
*CONTACT_DRAWBEAD
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$ ssid msid sstyp mstyp sboxid mboxid spr mpr
$ 2 18 4 3 2 1 1
$
$ fs fd dc vc vdc penchk bt dt
$ 0.10
$
$ sfs sfm sst mst sfst sfmt fsf vsf
$
$
$$$$ Card 4 required because it's a drawbead contact
$
$ lcdidrf lcdidf dbdth dfsc1 numint
$ 3 0.17436 2.0
$
$ lcdidrf = 3 load curve 3 specifies the bending component of the
$ restraining force per unit draw bead length
$ dbdth = 0.17436 draw bead depth
$ dfsc1 = 2.0 scale load curve 3 (lcdidrf) by 2
$
$$$$ Optional Cards A and B not specified (default values will be used).
$
$
*DEFINE_BOX
$ boxid xmin xmax ymn ymx zmn zmx
$ 2 0.000E+00 6.000E+00 6.000E+00 1.000E+02 -1.000E+03 1.000E+03
$
*SET_NODE_LIST
$ sid da1 da2 da3 da4
$ 2
$ nid1 nid2 nid3 nid4 nid5 nid6 nid7 nid8
$ 2580 2581 2582 2583 2584 2585 2586 2587
$ 2588 2589 2590
$
*DEFINE_CURVE
$ lcid sidr scla sclo offa offo
$ 3
$
$ a o
$ DEPTH FORC/LGTH
$ 0.000E+00 0.000E+00
$ 1.200E-01 1.300E+02
$ 1.500E-01 2.000E+02
$ 1.800E-01 5.000E+02
```

***CONTACT_AUTO_MOVE**

Purpose: Move the master surface in a contact definition to close an initial gap between the slave and master surfaces. The contact surfaces will then start in contact thereby saving calculational cost. The master surface in metalforming applications will typically be the punch and the blank will be the slave surface.

Define one card. Card Format (I0)

Cards 1 1 2 3 4 5 6 7 8

Variable	ID	CID	VID	LCID	ATIME			
Type	I	I	I	I	F			
Default	required	required	required	0	0.0			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID	ID for this auto positioning input.
CID	Contact ID.
VID	Vector ID for a vector oriented in the direction of the movement of the master surface. See *DEFINE_VECTOR. The origin of this vector is unimportant since the direction cosines of the vector are computed and used.
LCID	Optional load curve ID defining velocity versus time. The load curve will be adjusted to account for the movement of the master surface. The load curve should be defined by four points, and its shape should resemble a trapezoid with the longest parallel side along the abscissa. The abscissa is adjusted (shortened) in the flat part of the curve where the velocity is constant to account for the movement.
ATIME	Activation time. At this time the master surface is moved.

*CONTACT

*CONTACT_COUPLING

*CONTACT_COUPLING

Purpose: Define a coupling surface for MADYMO to couple LS-DYNA with deformable and rigid parts within MADYMO. In this interface, MADYMO computes the contact forces acting on the coupling surface, and LS-DYNA uses these forces in the update of the motion of the coupling surface for the next time step. Contact coupling can be used with other coupling options in LS-DYNA.

Card 1 1 2 3 4 5 6 7 8

Variable	ID							
Type	I							
Default	required							

Cards 2, 3, 4, ... Define as cards as necessary. The next "*" card terminates this input.

Cards 2,3,.. 1 2 3 4 5 6 7 8

Variable	SID	STYPE						
Type	I	I						
Default	required	0						

VARIABLE

DESCRIPTION

SID	Set ID for coupling. See Remark 1 below.
STYPE	Set type: EQ.0: part set EQ.1: shell element set EQ.2: solid element set EQ.3: thick shell element set

Remarks:

1. Only one coupling surface can be defined. If additional surfaces are defined, the coupling information will be added to the first definition.
2. The units and orientation can be converted by using the CONTROL_COUPLING keyword. It is not necessary to use the same system of units in MADYMO and in LS-DYNA if unit conversion factors are defined.

*CONTACT

*CONTACT_ENTITY

*CONTACT_ENTITY

Purpose: Define a contact entity. Geometric contact entities treat the impact between a deformable body defined as a set of slave nodes or nodes in a shell part set and a rigid body. The shape of the rigid body is determined by attaching geometric entities. Contact is treated between these geometric entities and the slave nodes using a penalty formulation. The penalty stiffness is optionally maximized within the constraint of the Courant criterion. As an alternative, a finite element mesh made with shells can be used as geometric entity. Also, axisymmetric entities with arbitrary shape made with multi-linear polygons are possible. The latter is particularly useful for metalforming simulations.

WARNING: If the problem being simulated involves dynamic motion of the entity, care should be taken to insure that the inertial properties of the entity are correct. It may be necessary to use the *PART_INERTIA option to specify these properties.

Define 5 cards for the contact entity definition below.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	GEOTYP	SSID	SSTYP	SF	DF	CF	INTORD
Type	I	I	I	I	F	F	F	I
Default	required	required	required	0	1.	0.	0.	0

VARIABLE

DESCRIPTION

PID

Part ID of the rigid body to which the geometric entity is attached, see *PART.

GEOTYP

Type of geometric entity:

EQ.1: plane,

EQ.2: sphere,

EQ.3: cylinder,

EQ.4: ellipsoid,

EQ.5: torus,

EQ.6: CAL3D/MADYMO Plane, see Appendix I,

EQ.7: CAL3D/MADYMO Ellipsoid, see Appendix I,

EQ.8: VDA surface, see Appendix L,

EQ.9: rigid body finite element mesh (shells only),

EQ.10: finite plane,

EQ.11: load curve defining line as surface profile of axisymmetric rigid bodies.

VARIABLE	DESCRIPTION
SSID	Slave set ID, see *SET_NODE_OPTION, *PART, or *SET_PART.
SSTYP	Slave set type: EQ.0: node set, EQ.1: part ID, EQ.2: part set ID.
SF	Penalty scale factor. Useful to scale maximized penalty.
DF	Damping option, see description for *CONTACT_OPTION: EQ.0: no damping, GT.0: viscous damping in percent of critical, e.g., 20 for 20% damping, EQ.-n: n! is the load curve ID giving the damping force versus relative normal velocity (see remark 1 below).
CF	Coulomb friction coefficient. Assumed to be constant.
INTORD	Integration order (slaved materials only). This option is not available with entity types 8 and 9 where only nodes are checked: EQ.0: check nodes only, EQ.1: 1 point integration over segments, EQ.2: 2x2 integration, EQ.3: 3x3 integration, EQ.4: 4x4 integration, EQ.5: 5x5 integration. This option allows a check of the penetration of the rigid body into the deformable (slaved) material. Then virtual nodes at the location of the integration points are checked.

Remarks:

1. The optional load curves that are defined for damping versus relative normal velocity and for force versus normal penetration should be defined in the positive quadrant. The sign for the damping force depends on the direction of the relative velocity and the treatment is symmetric if the damping curve is in the positive quadrant. If the damping force is defined in the negative and positive quadrants, the sign of the relative velocity is used in the table look-up.

*CONTACT

*CONTACT_ENTITY

Card 2 1 2 3 4 5 6 7 8

Variable	BT	DT	SO	GO				
Type	F	F	I	I				
Default	0.	1.E+20	0	0				

VARIABLE

DESCRIPTION

BT	Birth time
DT	Death time
SO	Flag to use penalty stiffness as in surface-to-surface contact: EQ.0: contact entity stiffness formulation, EQ.1: surface to surface contact method, EQ.-n: n is the load curve ID giving the force versus the normal penetration.
GO	Flag for mesh generation of the contact entity for entity types 1-5 and 10-11. This is used for visualization in post-processing only. EQ.0: mesh is not generated, EQ.1: mesh is generated.

Card 3 1 2 3 4 5 6 7 8

Variable	XC	YC	ZC	AX	AY	AZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0		

Card 4

Variable	BX	BY	BZ					
Type	F	F	F					
Default	0.	0.	0.					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XC	x-center, x_c , see remarks below.
YC	y-center, y_c , see remarks below.
ZC	z-center, z_c . See remarks below.
AX	x-direction for local axis A, A_x , see remarks below.
AY	y-direction for local axis A, A_y , see remarks below.
AZ	z-direction for local axis A, A_z , see remarks below.
BX	x-direction for local axis B, B_x , see remarks below.
BY	y-direction for local axis B, B_y , see remarks below.
BZ	z-direction for local axis B, B_z , see remarks below.

Remarks:

1. The coordinates, (x_c, y_c, z_c) , are the positions of the local origin of the geometric entity in global coordinates. The entity's local A-axis is determined by the vector (A_x, A_y, A_z) and the local B-axis by the vector (B_x, B_y, B_z) .
2. Cards 3 and 4 define a local to global transformation. The geometric contact entities are defined in a local system and transformed into the global system. For the ellipsoid, this is necessary because it has a restricted definition for the local position. For the plane, sphere, and cylinder, the entities can be defined in the global system and the transformation becomes $(x_c, y_c, z_c)=(0,0,0)$, $(A_x, A_y, A_z)=(1,0,0)$, and $(B_x, B_y, B_z)=(0,1,0)$.

Card 5 1 2 3 4 5 6 7 8

Variable	INOUT	G1	G2	G3	G4	G5	G6	G7
Type	I	F	F	F	F	F	F	F
Default	0	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
INOUT	In-out flag. Allows contact from the inside or the outside (default) of the entity: EQ.0: slave nodes exist outside of the entity, EQ.1: slave nodes exist inside the entity.
G1	Entity coefficient g_1 (CAL3D/MADYMO plane or ellipse number) for coupled analysis (see Appendix I).
G2	Entity coefficient g_2 , see remarks below.
G3	Entity coefficient g_3 , see remarks below.
G4	Entity coefficient g_4 , see remarks below.
G5	Entity coefficient g_5 , see remarks below.
G6	Entity coefficient g_6 , see remarks below.
G7	Entity coefficient g_7 , see remarks below.

Remarks:

Figures 7.4a and 7.4b show the definitions of the geometric contact entities. The relationships between the entity coefficients and the Figure 7.4a and 7.4b variables are as follows (please note that (P_x, P_y, P_z) is a position vector and that (Q_x, Q_y, Q_z) is a direction vector):

$$\begin{array}{ll}
 \text{GEOTYP} = 1: & g_1 = P_x & g_4 = Q_x \\
 & g_2 = P_y & g_5 = Q_y \\
 & g_3 = P_z & g_6 = Q_z \\
 & & g_7 = L
 \end{array}$$

If automatic generation is used, a square plane of length L on each edge is generated which represents the infinite plane. If generation is inactive, then g_7 may be ignored.

*CONTACT

*CONTACT_ENTITY

GEOTYP = 2: g1 = Px g4 = r
g2 = Py
g3 = Pz

GEOTYP = 3: g1 = Px g4 = Qx
g2 = Py g5 = Qy
g3 = Pz g6 = Qz
g7 = r

If automatic generation is used, a cylinder of length $\sqrt{Qx^2 + Qy^2 + Qz^2}$ and radius r is generated which represents the infinite cylinder.

GEOTYP = 4: g1 = Px g4 = a
g2 = Py g5 = b
g3 = Pz g6 = c
g7 = n (order of the ellipsoid)

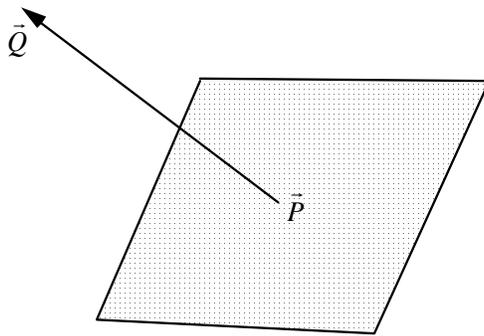
GEOTYP = 5: g1 = Radius of torus
g2 = r
g3 = number of elements along minor circumference
g4 = number of elements along major circumference

GEOTYP = 8: g1 = Blank thickness (option to override true thickness)
g2 = Scale factor for true thickness (optional)
g3 = Load curve ID defining thickness versus time. (optional)

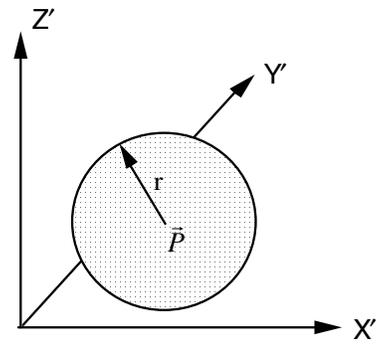
GEOTYP = 9: g1 = Shell thickness (option to override true thickness).
NOTE: The shell thickness specification is necessary if the slave surface is generated from solid elements.
g2 = Scale factor for true thickness (optional)
g3 = Load curve ID defining thickness versus time. (optional)

GEOTYP =10: g1 = Length of edge along X' axis
g2 = Length of edge along Y' axis

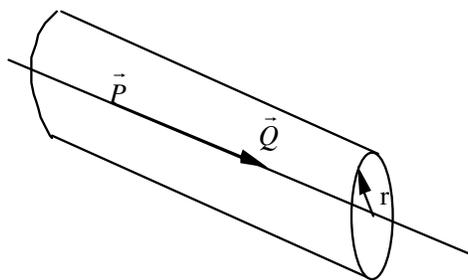
GEOTYP=11: g1 =Load curve ID defining axisymmetric surface profile about Z'-axis. Load curves defined by the keywords *DEFINE_CURVE or *DEFINE_CURVE_ENTITY can be used.
g2 = Number of elements along circumference
EQ.0: default set to 10
g3 = Number of elements along axis
EQ.0: default set to 20
EQ.-1: the elements are generated from the points on the load curve
g4 = Number of sub divisions on load curve used to calculate contact
EQ:0 default set to 1000



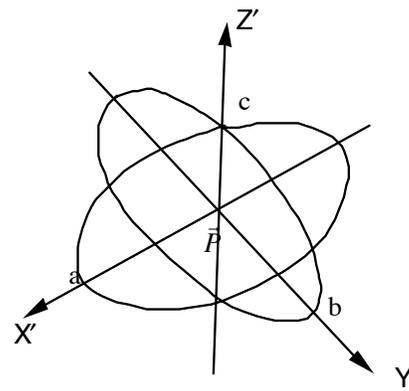
IGTYPE= 1: Infinite Plane



IGTYPE= 2: Sphere



IGTYPE= 3: Infinite Cylinder



$$\left(\frac{X'}{a}\right)^n + \left(\frac{Y'}{b}\right)^n + \left(\frac{Z'}{c}\right)^n = 1$$

IGTYPE= 4: Hyperellipsoid

Figure 7.4a. Contact Entities.

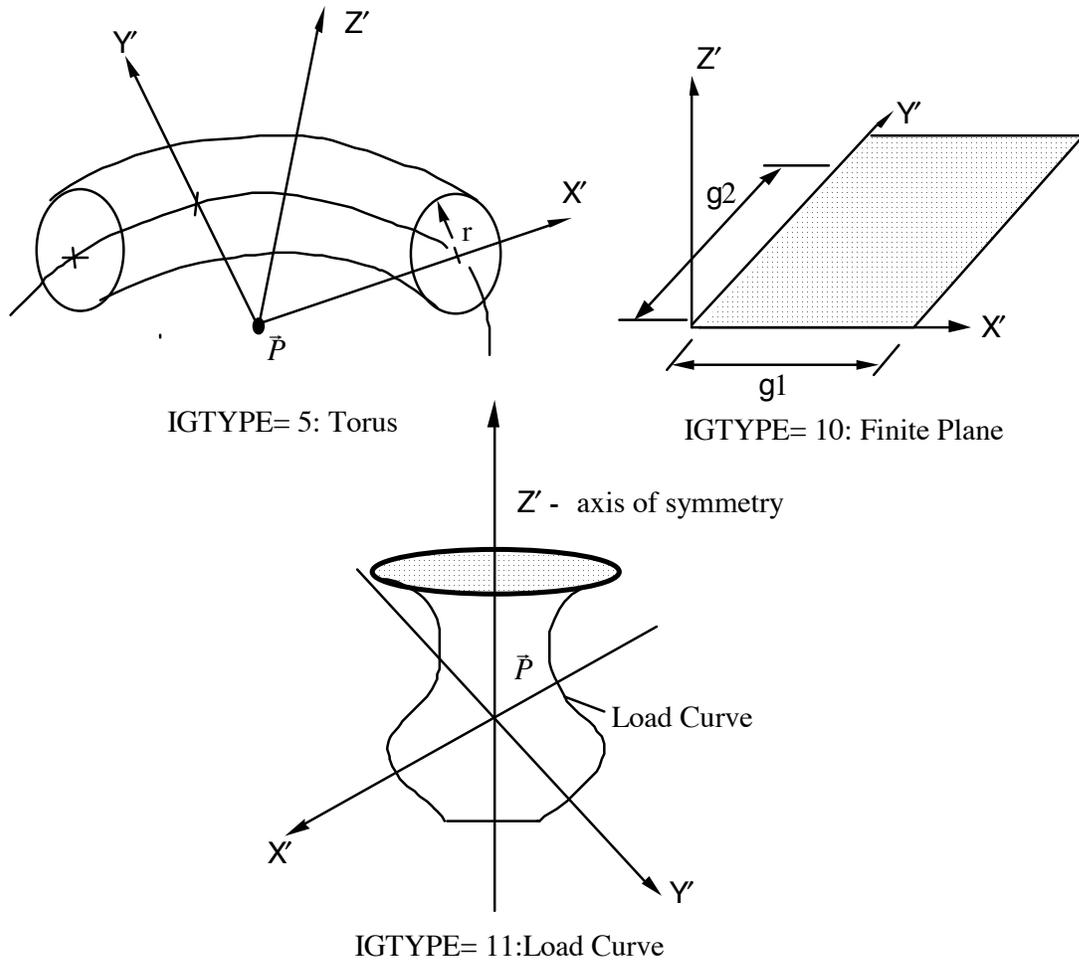


Figure 7.4b. Contact Entities.

***CONTACT_GEBOD_OPTION**

Purpose: Define contact interaction between the segment of a GEBOD dummy and parts or nodes of the finite element model. This implementation follows that of the contact entity, however, it is specialized for the dummies. Forces may be output using the *DATABASE_GCEOUT command. See *COMPONENT_GEBOD and Appendix N for further details.

Conventional *CONTACT_OPTION treatment (surface-to-surface, nodes-to-surface, etc.) can also be applied to the segments of a dummy. To use this approach it is first necessary to determine part ID assignments by running the model through LS-DYNA's initialization phase.

The following options are available and refer to the ellipsoids which comprise the dummy. Options involving **HAND** are not applicable for the child dummy since its lower arm and hand share a common ellipsoid.

LOWER_TORSO
MIDDLE_TORSO
UPPER_TORSO
NECK
HEAD
LEFT_SHOULDER
RIGHT_SHOULDER
LEFT_UPPER_ARM
RIGHT_UPPER_ARM
LEFT_LOWER_ARM
RIGHT_LOWER_ARM
LEFT_HAND
RIGHT_HAND
LEFT_UPPER_LEG
RIGHT_UPPER_LEG
LEFT_LOWER_LEG
RIGHT_LOWER_LEG
LEFT_FOOT
RIGHT_FOOT

*CONTACT

*CONTACT_GEBOD

Card 1 1 2 3 4 5 6 7 8

Variable	DID	SSID	SSTYP	SF	DF	CF	INTORD	
Type	I	I	I	F	F	F	I	
Default	required	required	required	1.	20.	0.5	0	

VARIABLE

DESCRIPTION

DID	Dummy ID, see *COMPONENT_GEBOD_OPTION.
SSID	Slave set ID, see *SET_NODE_OPTION, *PART, or *SET_PART.
SSTYP	Slave set type: EQ.0: node set, EQ.1: part ID, EQ.2: part set ID.
SF	Penalty scale factor. Useful to scale maximized penalty.
DF	Damping option, see description for *CONTACT_OPTION: EQ.0: no damping, GT.0: viscous damping in percent of critical, e.g., 20 for 20% damping, EQ.-n: $\ln l$ is the load curve ID giving the damping force versus relative normal velocity (see Remark 1 below).
CF	Coulomb friction coefficient (see Remark 2 below). Assumed to be constant.
INTORD	Integration order (slaved materials only). EQ.0: check nodes only, EQ.1: 1 point integration over segments, EQ.2: 2×2 integration, EQ.3: 3×3 integration, EQ.4: 4×4 integration, EQ.5: 5×5 integration. This option allows a check of the penetration of the dummy segment into the deformable (slaved) material. Then virtual nodes at the location of the integration points are checked.

Card 2 1 2 3 4 5 6 7 8

Variable	BT	DT	SO					
Type	F	F	I					
Default	0.	1.E+20	0					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
BT	Birth time
DT	Death time
SO	Flag to use penalty stiffness as in surface-to-surface contact: EQ.0: contact entity stiffness formulation, EQ.1: surface to surface contact method, EQ.-n: n is the load curve ID giving the force versus the normal penetration.

Remarks:

1. The optional load curves that are defined for damping versus relative normal velocity and for force versus normal penetration should be defined in the positive quadrant. The sign for the damping force depends on the direction of the relative velocity and the treatment is symmetric if the damping curve is in the positive quadrant. If the damping force is defined in the negative and positive quadrants, the sign of the relative velocity is used in the table look-up.
2. Insofar as these ellipsoidal contact surfaces are continuous and smooth it may be necessary to specify Coulomb friction values larger than those typically used with faceted contact surfaces.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	PID/PSID	SOFT	SSFAC	FRIC			
Type	I	I	I	F	F			
Default	none	none	0	1.0	none			

VARIABLE

DESCRIPTION

- NSID Node set ID that guides the 1D elements.
- PID/PSID Part ID or part set ID if SET is included in the keyword line.
- SOFT Flag for soft constraint option. Set to 1 for soft constraint.
- SSFAC Stiffness scale factor for penalty stiffness value. The default value is unity. This applies to SOFT set to 0 and 1.
- FRIC Contact friction.

*CONTACT

*CONTACT_INTERIOR

*CONTACT_INTERIOR

Purpose: Define interior contact for foam hexahedral and tetrahedral elements. Frequently, when foam materials are compressed under high pressure, the solid elements used to discretize these materials may invert leading to negative volumes and error terminations. In order to keep these elements from inverting, it is possible to consider interior contacts within the foam between layers of interior surfaces made up of the faces of the solid elements. Since these interior surfaces are generated automatically, the part (material) ID's for the materials of interest are defined here, prior to the interface definitions. ONLY ONE PART SET ID CAN BE DEFINED.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

PSID Part set ID including all parts for which interior contact is desired.

Four attributes should be defined for the part set:

- Attribute 1: PSF, penalty scale factor (Default=1.00).
- Attribute 2: Activation factor, F_a (Default=0.10). When the crushing of the element reaches F_a times the initial thickness the contact algorithm begins to act.
- Attribute 3: ED, Optional modulus for interior contact stiffness.
- Attribute 4: TYPE, Formulation for interior contact.
EQ.1.0: Default, recommended for uniform compression
EQ.2.0: Designed to control the combined modes of shear and compression. Works for type 1 brick formulation and type 10 tetrahedron formulation.

Remarks:

The interior penalty is determined by the formula:

$$K = \frac{SLSFAC \cdot PSF \cdot Volume^{\frac{2}{3}} \cdot E}{Min.Thickness}$$

where SLSFAC is the value specified on the *CONTROL_CONTACT card , volume is the volume of the brick element, E is a constitutive modulus, and min. thickness is approximately the thickness of the solid element through its thinnest dimension. If ED, is defined above the interior penalty is then given instead by:

$$K = \frac{Volume^{\frac{2}{3}} \cdot ED}{Min.Thickness}$$

where the scaling factors are ignored. Generally, ED should be taken as the locking modulus specified for the foam constitutive model.

Caution should be observed when using this option since if the time step size is too large an instability may result. The time step size is not affected by the use of interior contact.

*CONTACT

*CONTACT_RIGID_SURFACE

*CONTACT_RIGID_SURFACE

Purpose: Define rigid surface contact. The purpose of rigid surface contact is to model large rigid surfaces, e.g., road surfaces, with nodal points and segments that require little storage and are written out at the beginning of the binary databases. The rigid surface motion, which can be optionally prescribed, is defined by a displacement vector which is written with each output state. The nodal points defining the rigid surface must be defined in the *NODE_RIGID_SURFACE section of this manual. These rigid nodal points do not contribute degrees-of-freedom.

Card 1 1 2 3 4 5 6 7 8

Variable	CID	PSID	BOXID	SSID	FS	FD	DC	VC
Type	I	I	I	I	F	F	F	F
Default	none	none	0	none	0.	0.	0.	0.

Card 2

Variable	LCIDX	LCIDY	LCIDZ	FSLCID	FDLCID			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

Card 3

Variable	SFS	STTHK	SFTHK	XPENE	BSORT			
Type	F	F	F	F	I			
Default	1.0	0.0	1.0	4.0	10			

VARIABLE	DESCRIPTION
CID	Contact interface ID. This must be a unique number.
PSID	Part set ID of all parts that may contact the rigid surface. See *SET_PART.
BOXID	Include only nodes of the part set that are within the specified box, see *DEFINE_BOX, in contact. If BOXID is zero, all nodes from the part set, PSID, will be included in the contact.
SSID	Segment set ID defining the rigid surface. See *SET_SEGMENT.
FS	Static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$. If FSLCID is defined, see below, then FS is overwritten by the value from the load curve.
FD	Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$. If FDLCID is defined, see below, then FD is overwritten by the value from the load curve.
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$.
VC	Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{lim} = VC \cdot A_{cont}$. A_{cont} being the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_o is the yield stress of the contacted material.
LCIDX	Load curve ID defining x-direction motion. If zero, there is no motion in the x-coordinate system.
LCIDY	Load curve ID defining y-direction motion. If zero, there is no motion in the y-coordinate system.
LCIDZ	Load curve ID defining z-direction motion. If zero, there is no motion in the z-coordinate system.
FSLCID	Load curve ID defining the static coefficient of friction as a function of interface pressure. This option applies to shell segments only.

VARIABLE	DESCRIPTION
F DLCID	Load curve ID defining the dynamic coefficient of friction as a function of interface pressure. This option applies to shell segments only.
SFS	Scale factor on default slave penalty stiffness, see also *CONTROL_CONTACT.
STTHK	Optional thickness for slave surface (overrides true thickness). This option applies to contact with shell, solid, and beam elements. True thickness is the element thickness of the shell elements. Thickness offsets are not used for solid element unless this option is specified.
SFTHK	Scale factor for slave surface thickness (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
XPENE	Contact surface maximum penetration check multiplier. If the penetration of a node through the rigid surface exceeds the product of XPENE and the slave node thickness, the node is set free. EQ.0: default is set to 4.0.
BSORT	Number of cycles between bucket sorts. The default value is set to 10 but can be much larger, e.g., 50-100, for fully connected surfaces.

Remarks:

Thickness offsets do not apply to the rigid surface. There is no orientation requirement for the segments in the rigid surface, and the surface may be assembled from disjoint, but contiguous, arbitrarily oriented meshes. With disjoint meshes, the global searches must be done frequently, about every 10 cycles, to ensure a smooth movement of a slave node between mesh patches. For fully connected meshes this frequency interval can be safely set to 50-200 steps between searches.

The modified binary database (D3PLOT) contains the road surface information prior to the state data. This information contains:

NPDS	=	Total number of rigid surface points in problem.
NRSC	=	Total number of rigid surface contact segments summed over all definitions.
NSID	=	Number of rigid surface definitions.
NVELQ	=	Number of words at the end of each binary output state defining the rigid surface motion. This equals 6 x NSID if any rigid surface moves or zero if all rigid surfaces are stationary.
PIDS	=	An array equal in length to NPDS. This array defines the ID for each point in the road surface.
XC	=	An array equal in length to 3 x NPDS. This array defines the global x, y, and z coordinates of each point.

For each road surface define the following NSID sets of data.

ID = Rigid surface ID.
NS = Number of segments in rigid surface.
IXRS = An array equal in length to 4 x NS. This is the connectivity of the rigid surface in the internal numbering system.

At the end of each state, 6 x NVELQ words of information are written. For each road surface the x, y, and z displacements and velocities are written. If the road surface is fixed, a null vector should be output. Skip this section if NVELQ=0. LS-PREPOST currently displays rigid surfaces and animates their motion.

*CONTACT

*CONTACT_ID

*CONTACT_ID

Purpose: Define one-dimensional slide lines for rebar in concrete.

Card 1 1 2 3 4 5 6 7 8

Variable	NSIDS	NSIDM	ERR	SIGC	GB	SMAX	EXP	
Type	I	I	F	F	F	F	F	
Default	none	none	0.	0.	0.	0.	0.	

VARIABLE

DESCRIPTION

NSIDS	Nodal set ID for the slave nodes, see *SET_NODE.
NSIDM	Nodal set ID for the master nodes, see *SET_NODE.
ERR	External radius of rebar
SIGC	Compressive strength of concrete
GB	Bond shear modulus
SMAX	Maximum shear strain
EXP	Exponent in damage curve

Remarks:

With this option the concrete is defined with solid elements and the rebar with truss elements, each with their own unique set of nodal points. A string of consecutive nodes, called slave nodes, related to the truss elements may slide along a string of consecutive nodes, called master nodes, related to the solid elements. The sliding commences after the rebar debonds.

The bond between the rebar and concrete is assumed to be elastic perfectly plastic. The maximum allowable slip strain is given as:

$$u_{max} = SMAX \cdot e^{-EXP \cdot D}$$

where D is the damage parameter $D_{n+1} = D_n + \Delta u$. The shear force, acting on area A_s , at time $n+1$ is given as:

$$f_{n+1} = \min(f_n - GB \cdot A_s \cdot \Delta u, GB \cdot A_s \cdot u_{max})$$

***CONTACT_2D_OPTION1_{OPTION2}_{OPTION3}**

Purpose: Define a 2-dimensional contact or slide line. This option is to be used with 2D solid and shell elements using the plane_stress, plane_strain or axisymmetric formulations, see *SECTION_SHELL.

OPTION1 specifies the contact type. The following options should be used with deformable materials only (i.e., not rigid):

SLIDING_ONLY

TIED_SLIDING

SLIDING_VOIDS

since these methods are based on the imposition of constraints. The constraint methods may be used with rigid bodies if the rigid body is the master surface and all rigid body motions are prescribed. The following options may be used with rigid materials as well:

PENALTY_FRICTION

PENALTY

AUTOMATIC_SINGLE_SURFACE

AUTOMATIC_SURFACE_TO_SURFACE

AUTOMATIC_NODE_TO_SURFACE

AUTOMATIC_SURFACE_IN_CONTINUUM

AUTOMATIC_TIED

AUTOMATIC_TIED_ONE_WAY

OPTION2 specifies a thermal contact and takes the single option:

THERMAL

Only the AUTOMATIC types: SINGLE_SURFACE, SURFACE_TO_SURFACE, and NODE_TO_SURFACE may be used with this option.

OPTION3 specifies that the first card to read defines the title and ID number of contact interface and takes the single option:

TITLE

Note: *OPTION2* and *OPTION3* may appear in any order.

For the PENALTY_FRICTION option define the following additional card

Card 3 1 2 3 4 5 6 7 8

Variable	FRIC	FRIC_L	FRIC_H	FRIC_S				
Type	F	F	F	F				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SSID	Nodal set ID for the slave nodes, see *SET_NODE. The slave surface must be to the left of the master surface.
MSID	Nodal set ID for the master nodes, see *SET_NODE.
TBIRTH	Birth time for contact.
TDEATH	Death time for contact.
EXT_PAS	Slideline extension bypass option. EQ.0: extensions are use EQ.1: extensions are not used
THETA1	Angle in degrees of slideline extension at first master node. EQ.0: extension remains tangent to first master segment.
THETA2	Angle in degrees of slideline extension at last master node. EQ.0: extension remains tangent to last master segment.
TOL_IG	Tolerance for determining initial gaps. EQ.0.0: default set to 0.001
PEN	Scale factor or penalty. EQ.0.0: default set to 0.10
TOLOFF	Tolerance for stiffness insertion for implicit solution only. The contact stiffness is inserted when a node approaches a segment a distance equal to the segment length multiplied by TOLOFF. The stiffness is increased as the node moves closer with the full stiffness being used when the nodal point finally makes contact. EQ.0.0: default set to 0.025.
FRC_SCL	Scale factor for the interface friction. EQ.0.0: default set to 0.010

***CONTACT**

***CONTACT_2D_OPTION1_{OPTION2}_{OPTION3}**

VARIABLE	DESCRIPTION
ONEWAY	Flag for one way treatment. If set to 1.0 the nodal points on the slave surface are constrained to the master surface. This option is generally recommended if the master surface is rigid. EQ.1.0: activate one way treatment.
FRIC	Coefficient of friction
FRIC_L	Coefficient of friction at low velocity.
FRIC_H	Coefficient of friction at high velocity.
FRIC_S	Friction factor for shear.

***CONTACT_2D_OPTION1_{OPTION2}_{OPTION3}**

***CONTACT**

For the AUTOMATIC options define the following two cards:

Card 1 1 2 3 4 5 6 7 8

Variable	SIDS	SIDM	SFACT	FREQ	FS	FD	DC	MEMBS
Type	I	I	F	I	F	F	F	I
Default	none	none	1.0	50	0.	0.	0.	12
Remarks	1,2	1,2						

Card 2

Variable	TBIRTH	TDEATH	SOS	SOM	NDS	NDM	COF	INIT
Type	F	F	F	F	I	I	I	I
Default	0.	1.e20	1.0	1.0	0	0	0	0
Remarks			3	3	4	5		8

This Card is mandatory for the THERMAL option, i.e.,:

***CONTACT_AUTOMATIC_..._THERMAL_....**

Optional 1 2 3 4 5 6 7 8

Variable	K	RAD	H	LMIN	LMAX	CHLM	BC_FLAG	
Type	F	F	F	F	F	F	I	
Default	none	none	none	none	none	1.0	0	

*CONTACT

*CONTACT_2D_OPTION1_{OPTION2}_{OPTION3}

Optional Card A

Card A 1 2 3 4 5 6 7 8

Variable	VC	VDC	IPF	SLIDE	ISTIFF	TIEDGAP		
Type	F	F	I	I	I	R		
Default	0.	10.0	0	0	0			
Remarks				10	11	12		

VARIABLE

DESCRIPTION

SIDS	Set ID to define the slave surface. If SIDS>0, a part set is assumed, see *SET_PART. If SIDS<0, a node set with ID equal to the absolute value of SIDS is assumed, see *SET_NODE.
SIDM	Set ID to define the master surface. If SIDM>0, a part set is assumed, see *SET_PART. If SIDM<0, a node set with ID equal to the absolute value of SIDM is assumed, see *SET_NODE. Do not define for single surface contact.
SFACT	Scale factor for the penalty force stiffness.
FREQ	Search frequency. The number of timesteps between bucket sorts. For implicit contact this parameter is ignored and the search frequency is 1. EQ.0: default set to 50.
FS	Static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact according to the relationship given by: $\mu_c = FD + (FS - FD)e^{-DC \cdot v_{rel} }$.
FD	Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC \cdot v_{rel} }$.
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC \cdot v_{rel} }$.
MEMBS	Parameter to allocate memory for bucket sort pair information.

VARIABLE	DESCRIPTION
TBIRTH	Birth time for contact.
TDEATH	Death time for contact.
SOS	Surface offset from midline for 2D shells of slave surface EQ.0.0: default to 1. GT.0.0: scale factor applied to actual thickness LT.0.0: absolute value is used as the offset
SOM	Surface offset from midline for 2D shells of master surface EQ.0: default to 1. GT.0: scale factor applied to actual thickness LT.0: absolute value is used as the offset
NDS	Normal direction flag for 2D shells of slave surface EQ.0: Normal direction is determined automatically EQ.1: Normal direction is in the positive direction EQ.-1: Normal direction is in the negative direction
NDM	Normal direction flag for 2D shells of master surface EQ.0: Normal direction is determined automatically EQ.1: Normal direction is in the positive direction EQ.-1: Normal direction is in the negative direction
COF	Closing/Opening flag for implicit contact EQ.0: Recommended for most problem where gaps are only closing. EQ.1: Recommended when gaps are opening to avoid sticking.
INIT	Special processing during initialization EQ.0: No special processing. EQ.1: Forming option.
K	Thermal conductivity (k) of fluid between the slide surfaces. If a gap with a thickness l_{gap} exists between the slide surfaces, then the conductance due to thermal conductivity between the slide surfaces is $h_{cond} = \frac{k}{l_{gap}}$ <p>Note that LS- DYNA calculates l_{gap} based on deformation.</p>

*CONTACT

*CONTACT_2D_OPTION1_{OPTION2}_{OPTION3}

VARIABLE	DESCRIPTION
RAD	Radiation factor, f , between the slide surfaces. A radiant-heat-transfer coefficient (h_{rad}) is calculated (see *BOUNDARY_RADIATION). If a gap exists between the slide surfaces, then the contact conductance is calculated by $h = h_{cond} + h_{rad}$
H	Heat transfer conductance (h_{cont}) for closed gaps. Use this heat transfer conductance for gaps in the range $0 \leq l_{gap} \leq l_{min}$ where l_{min} is GCRIT defined below.
LMIN	Critical gap (l_{min}), use the heat transfer conductance defined (HTC) for gap thicknesses less than this value.
LMAX	No thermal contact if gap is greater than this value (l_{max}).
CHLM	Is a multiplier used on the element characteristic distance for the search routine. The characteristic length is the largest interface surface element diagonal. EQ.0: Default set to 1.0
BC_FLAG	Thermal boundary condition flag EQ.0: thermal boundary conditions are on when parts are in contact EQ.1: thermal boundary conditions are off when parts are in contact
VC	Coefficient for viscous friction. This is used to limit the friction force to a maximum. A limiting force is computed $F_{lim} = VC \cdot A_{cont}$. A_{cont} being the area of contacted between segments. The suggested value for VC is to use the yield stress in shear: $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_o is the yield stress of the contacted material.
VDC	Viscous damping coefficient in percent of critical for explicit contact.
IPF	Initial penetration flag for explicit contact. EQ.0: Allow initial penetrations to remain EQ.1: Push apart initially penetrated surfaces

VARIABLE	DESCRIPTION
SLIDE	Sliding option. EQ.0: Off EQ.1: On
ISTIFF	Stiffness scaling option. EQ.0: Use default option. EQ.1: Scale stiffness using segment masses and explicit time step (default for explicit contact) EQ.2: Scale stiffness using segment stiffness and dimensions (default for implicit contact)
TIEDGAP	Search gap for tied contacts. EQ.0: Default, use 1% of the master segment length GT.0: Use the input value LT.0: Use n% of the master segment length where $n= TIEDGAP $

Remarks:

Remarks 1 through 12 pertain to 2D_AUTOMATIC contact.

1. For AUTOMATIC_SURFACE_TO_SURFACE, AUTOMATIC_SINGLE_SURFACE contact and AUTOMATIC_NODE_TO_SURFACE contact, penetration of 2D shell elements and external faces of 2D continuum elements is prevented by penalty forces. Parts in the slave part set are checked for contact with parts in the master part set. Self contact is checked for any part in both sets. If the slave part set is omitted, all parts are checked for contact. If the master part set is omitted, it is assumed to be identical to the slave part set.
2. For AUTOMATIC_SURFACE_IN_CONTINUUM contact, penalty forces prevent the flow of slave element material (the continuum) through the master surfaces. Flow of the continuum tangent to the surface is permitted. Only 2D solid parts are permitted in the slave part set. Both 2D solid and 2D shell parts are permitted in the master part set. Neither the slave part set ID or the master part set ID may be omitted.
3. By default, the true thickness of 2D shell elements is taken into account for AUTOMATIC_SURFACE_TO_SURFACE and AUTOMATIC_NODE_TO_SURFACE contact. The user can override the true thickness by using SOS and SOM. If the surface offset is reduced to a small value, the automatic normal direction algorithm may fail, so it is best to specify the normal direction using NDS or NDM. Thickness of 2D shell elements is not considered for AUTOMATIC_SURFACE_IN_CONTINUUM contact.
4. By default, the normal direction of 2D shell elements is evaluated automatically for AUTOMATIC_SURFACE_TO_SURFACE and AUTOMATIC_NODE_TO_SURFACE contact. The user can override the automatic algorithm using NDS or NDM and contact will occur with the positive or negative face of the element.

5. For SURFACE_IN_CONTINUUM contact, flow through 2D shell elements is prevented in both directions by default. If NDM is set to ± 1 , flow in the direction of the normal is permitted.
6. When using AUTOMATIC_SURFACE_IN_CONTINUUM contact, there is no need to mesh the continuum around the structure because contact is not with continuum nodes but with material in the interior of the continuum elements. The algorithm works well for Eulerian or ALE elements since the structure does not interfere with remeshing. However, a structure will usually not penetrate the surface of an ALE continuum since the nodes are Lagrangian normal to the surface. Therefore, if using an ALE fluid, the structure should be initially immersed in the fluid and remain immersed throughout the calculation. Penetrating the surface of an Eulerian continuum is not a problem.
7. For all types of 2D_AUTOMATIC contact, eroding materials are treated by default. At present, subcycling is not possible.
8. Currently only one special initialization option is available. The forming option is intended for implicit solutions of thin solid parts when back side segments may interfere with the solution. It automatically removes back side segments during initialization. If slave or master surfaces are defined by node sets, the forming option should be turned off.
9. For the thermal option:

$$h = h_{cont}, \text{ if the gap thickness is } 0 \leq l_{gap} \leq l_{min}$$

$$h = h_{cond} + h_{rad}, \text{ if the gap thickness is } l_{min} \leq l_{gap} \leq l_{max}$$

$$h = 0, \text{ if the gap thickness is } l_{gap} > l_{max}$$

10. When turned on, the sliding option activates additional logic intended to improve sliding when surfaces in contact have kinks or corners. This option is off by default.
11. The ISTIFF option allows control of the equation used in calculating the penalty stiffness. For backward compatibility, the default values are different for implicit and explicit solutions. When ISTIFF=1 is used, the explicit time step appears in the stiffness equation regardless if the calculation is implicit or explicit.
12. AUTOMATIC_TIED_ONE_WAY contact creates two degree of freedom translational kinematic constraints to nodes on the slave surface which are initially located on or near master segments. AUTOMATIC_TIED contact creates kinematic constraints between slave nodes and master segments, and also creates penalty constraints between master nodes and slave segments. With either contact option, a kinematic constraint may be switched to penalty if there is a conflict with another constraint. The TIEDGAP parameter determines the maximum normal distance from a segment to a node for a constraint to be formed. Nodes will not be moved to eliminate an initial gap, and the initial gap will be maintained throughout the calculation.

The remaining discussion applies to the SLIDING_ONLY, TIED_SLIDING, SLIDING_VOIDS, PENALTY_FRICTION, and PENALTY options. These options were adopted from LS-DYNA2D and originated in the public domain version of DYNA2D from the Lawrence

Livermore National Laboratory. The AUTOMATIC contact options are generally recommended excepted for the TIED option.

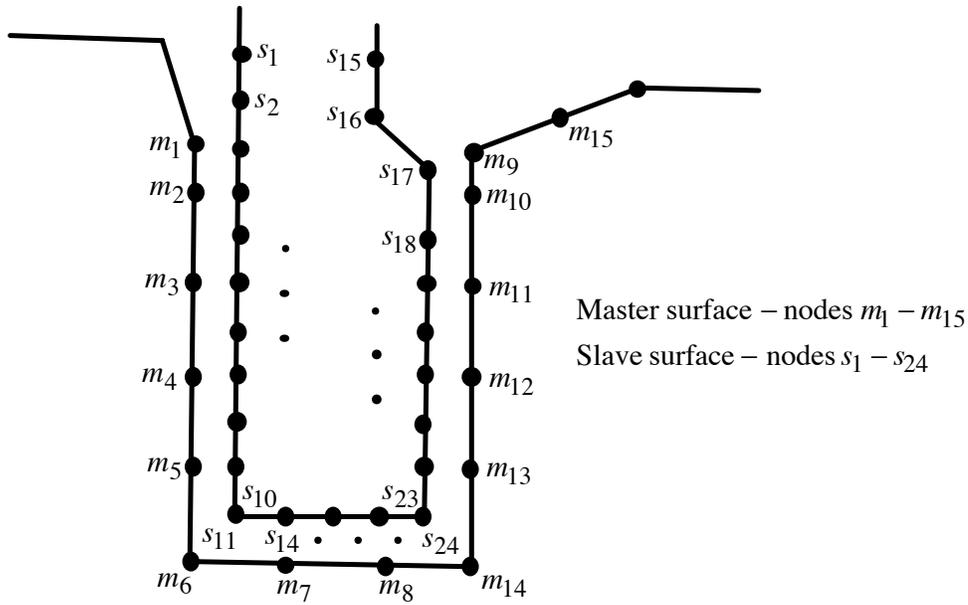
Consider two slideline surfaces in contact. It is necessary to designate one as a slave surface and the other as a master surface. Nodal points defining the slave surface are called slave nodes, and similarly, nodes defining the master surface are called master nodes. Each slave-master surface combination is referred to as a slideline.

Many potential problems with the algorithm can be avoided by observing the following precautions:

- Metallic materials should contain the master surface along high explosive-metal interfaces.
- Sliding only type slidelines are appropriate along high explosive-metal interfaces. The penalty formulation is not recommended along such interfaces.
- If one surface is more finely zoned, it should be used as the slave surface. If penalty slidelines are used, PENALTY and PENALTY_FRICTION, the slave-master distinction is irrelevant.
- A slave node may have more than one master segment, and may be included as a member of a master segment if a slideline intersection is defined.
- Angles in the master side of a slideline that approach 90° must be avoided.

Whenever such angles exist in a master surface, two or more slidelines should be defined. This procedure is illustrated in Figure 7.5. An exception for the foregoing rule arises if the surfaces are tied. In this case, only one slideline is needed.

- Whenever two surfaces are in contact, the smaller of the two surfaces should be used as the slave surface. For example, in modeling a missile impacting a wall, the contact surface on the missile should be used as the slave surface.
- Care should be used when defining a master surface to prevent the extension from interfering with the solution. In Figures 7.6 and 7.7, slideline extensions are shown.



1		2		3	
<u>Slaves</u>	<u>Masters</u>	<u>Slaves</u>	<u>Masters</u>	<u>Slaves</u>	<u>Masters</u>
s_1	m_1	s_{11}	m_6	s_{24}	m_{14}
s_2	m_2	s_{12}	m_7	s_{23}	m_{13}
.	.	.	m_8	.	.
.	.	.	m_{14}	.	.
.
		s_{14}			m_9
s_{11}	m_6	s_{24}		s_{15}	m_{15}

Figure 7.5. Proper definition of illustrated slave-master surface requires three slidelines (note that slave surface is to the left of the master surface as one moves along master nodes in order of definition).

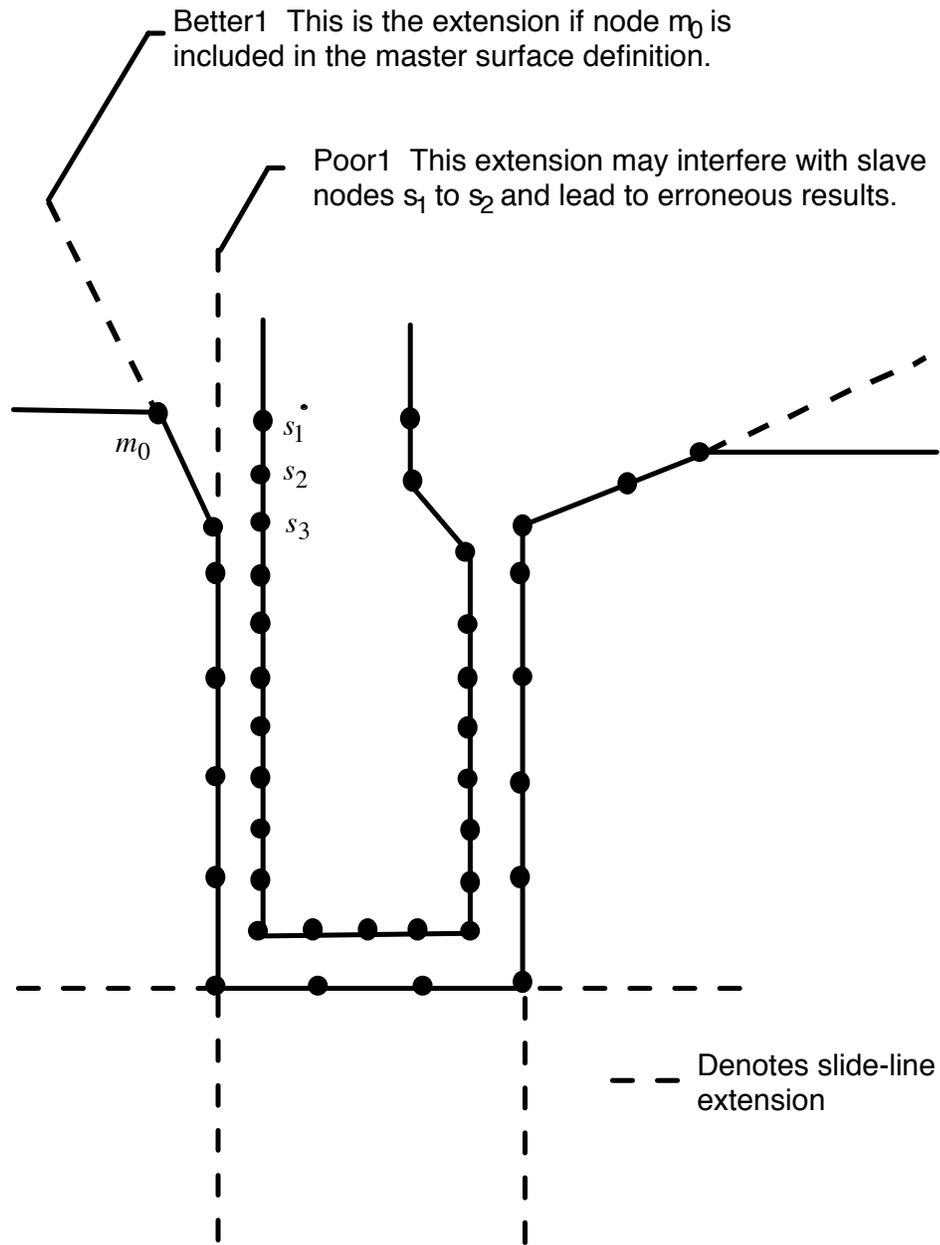


Figure 7.6. Master surface extensions defined automatically by DYNA (extensions are updated every time step to remain tangent to ends of master sides of slidelines unless angle of extension is defined in input).

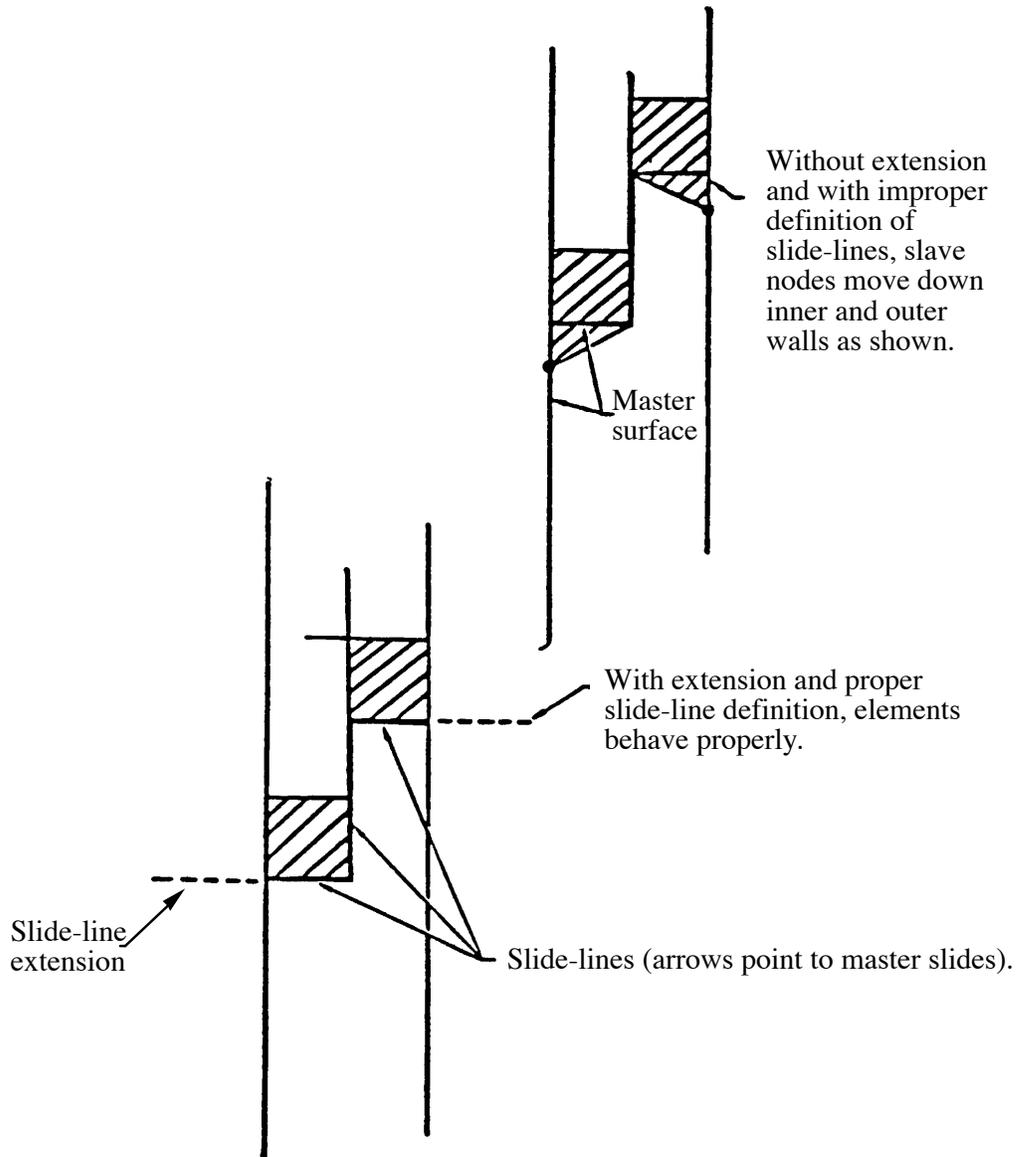


Figure 7.7. Example of slideline extensions helping to provide realistic response.

***CONTROL**

The keyword control cards are optional and can be used to change defaults, activate solution options such as mass scaling, adaptive remeshing, and an implicit solution; however, it is advisable to define the *CONTROL_TERMINATION card. **The ordering of the control cards in the input file is arbitrary. To avoid ambiguities, define no more than one control card of each type.** The following control cards are organized in alphabetical order:

- *CONTROL_ACCURACY**
- *CONTROL_ADAPSTEP**
- *CONTROL_ADAPTIVE**
- *CONTROL_ALE**
- *CONTROL_BULK_VISCOSITY**
- *CONTROL_CHECK_{OPTION}**
- *CONTROL_COARSEN**
- *CONTROL_CONTACT**
- *CONTROL_COUPLING**
- *CONTROL_CPU**
- *CONTROL_DYNAMIC_RELAXATION**
- *CONTROL_EFG**
- *CONTROL_ENERGY**
- *CONTROL_EXPLOSIVE_SHADOW**
- *CONTROL_FORMING_POSITION**
- *CONTROL_FORMING_TEMPLATE**
- *CONTROL_FORMING_TRAVEL**
- *CONTROL_FORMING_USER**
- *CONTROL_HOURLASS_{OPTION}**
- *CONTROL_IMPLICIT_AUTO**
- *CONTROL_IMPLICIT_BUCKLE**
- *CONTROL_IMPLICIT_DYNAMICS**
- *CONTROL_IMPLICIT_EIGENVALUE**
- *CONTROL_IMPLICIT_GENERAL**
- *CONTROL_IMPLICIT_INERTIA_RELIEF**
- *CONTROL_IMPLICIT_JOINTS**
- *CONTROL_IMPLICIT_MODES**
- *CONTROL_IMPLICIT_SOLUTION**
- *CONTROL_IMPLICIT_SOLVER**

***CONTROL**

***CONTROL_IMPLICIT_STABILIZATION**
***CONTROL_IMPLICIT_TERMINATION**
***CONTROL_MPP_DECOMPOSITION_AUTOMATIC**
***CONTROL_MPP_DECOMPOSITION_CHECK_SPEED**
***CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE**
***CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE**
***CONTROL_MPP_DECOMPOSITION_FILE**
***CONTROL_MPP_DECOMPOSITION_METHOD**
***CONTROL_MPP_DECOMPOSITION_NUMPROC**
***CONTROL_MPP_DECOMPOSITION_SHOW**
***CONTROL_MPP_DECOMPOSITION_TRANSFORMATION**
***CONTROL_MPP_IO_NOD3DUMP**
***CONTROL_MPP_IO_NODUMP**
***CONTROL_MPP_IO_NOFULL**
***CONTROL_MPP_IO_SWAPBYTES**
***CONTROL_NONLOCAL**
***CONTROL_OUTPUT**
***CONTROL_PARALLEL**
***CONTROL_REMESHING**
***CONTROL_RIGID**
***CONTROL_SHELL**
***CONTROL_SOLID**
***CONTROL_SOLUTION**
***CONTROL_SPH**
***CONTROL_SPOTWELD_BEAM**
***CONTROL_STAGED_CONSTRUCTION**
***CONTROL_STRUCTURED_{OPTION}**
***CONTROL_SUBCYCLE**
***CONTROL_TERMINATION**
***CONTROL_THERMAL_NONLINEAR**
***CONTROL_THERMAL_SOLVER**
***CONTROL_THERMAL_TIMESTEP**
***CONTROL_TIMESTEP**

LS-DYNA's implicit mode may be activated in two ways. Using the ***CONTROL_IMPLICIT_GENERAL** keyword, a simulation may be flagged to run entirely in implicit mode. Alternatively, an explicit simulation may be seamlessly switched into implicit

mode at the termination time using the *INTERFACE_SPRINGBACK_SEAMLESS keyword. The seamless switching feature is intended to simplify metal forming springback calculations, where the forming phase can be run in explicit mode, followed immediately by an implicit static springback simulation. In case of difficulty, restart capability is supported. Eight keywords are available to support implicit analysis. Default values are carefully selected to minimize input necessary for most simulations. These are summarized below:

- *CONTROL_IMPLICIT_GENERAL
Activates implicit mode, selects time step size.
- *CONTROL_IMPLICIT_INERTIA_RELIEF
Allows linear analysis of models with rigid body modes.
- *CONTROL_IMPLICIT_SOLVER
Selects parameters for solving system of linear equations $[K]\{x\}=\{f\}$.
- *CONTROL_IMPLICIT_SOLUTION
Selects linear or nonlinear solution method, convergence tolerances.
- *CONTROL_IMPLICIT_AUTO
Activates automatic time step control.
- *CONTROL_IMPLICIT_DYNAMICS
Activates and controls dynamic implicit solution using Newmark method.
- *CONTROL_IMPLICIT_EIGENVALUE
Activates and controls eigenvalue analysis.
- *CONTROL_IMPLICIT_MODES
Activates and controls computation of constraint and attachment modes.
- *CONTROL_IMPLICIT_STABILIZATION
Activates and controls artificial stabilization for multi-step springback.

*CONTROL

*CONTROL_ACCURACY

*CONTROL_ACCURACY

Purpose: Define control parameters that can improve the accuracy of the calculation.

Card 1	1	2	3	4	5	6	7	8
Variable	OSU	INN	PIDOSU					
Type	I	I	I					
Default	0 (off)		optional					

VARIABLE

DESCRIPTION

OSU	Global flag for 2nd order objective stress updates (See Remark 1 below). Generally, for explicit calculations only those parts undergoing large rotations, such as rolling tires, need this option. Objective stress updates can be activated for a subset of part IDs by defining the part set in columns 21-30. EQ.0: Off (default) EQ.1: On
INN	Invariant node numbering for shell and solid elements. (See Remarks 2 and 3 below). EQ.1: Off (default for explicit) EQ.2: On for shell elements only (default for implicit) EQ.3: On for solid elements only EQ.4: On for both shell and solid elements
PIDOSU	Part set ID for objective stress updates. If this part set ID is given only those part IDs listed will use the objective stress update; therefore, OSU is ignored.

Remarks:

1. Second order objective stress updates are occasionally necessary. Some examples include spinning bodies such as turbine blades in a jet engine, high velocity impacts generating large strains in a few time steps, and large time step sizes due to mass scaling in metal forming. There is a significantly added cost which is due in part to the added cost of the second order terms in the stress update when the Jaumann rate is used and the need to compute the strain-displacement matrix at the mid-point geometry. This option is available for one point brick elements, the selective-reduced integrated brick element which uses eight integration points, the fully integrated plane strain and axisymmetric volume weighted (type 15) 2D solid elements, the fully integrated thick shell element, and the following shell elements: Belytschko-Tsay, Belytschko-Tsay with warping

stiffness, Belytschko-Chiang-Wong, S/R Hughes-Liu, and the type 16 fully integrated shell element.

2. Invariant node numbering for shell elements affects the choice of the local element shell coordinate system. The orientation of the default local coordinate system is based on the shell normal vector and the direction of the 1-2 side of the element. If the element numbering is permuted, the results will change in irregularly shaped elements. With invariant node numbering, permuting the nodes shifts the local system by an exact multiple of 90 degrees. In spite of its higher costs [$<5\%$], the invariant local system is recommended for several reasons. First, element forces are nearly independent of node sequencing; secondly, the hourglass modes will not substantially affect the material directions; and, finally, stable calculations over long time periods are achievable.
3. Invariant node numbering for solid elements is available for anisotropic materials only. This option has no effect on solid elements of isotropic material. This option is recommended when solid elements of anisotropic material undergo significant deformation.

*CONTROL

*CONTROL_ADAPSTEP

*CONTROL_ADAPSTEP

Purpose: Define control parameters for contact interface force update during each adaptive cycle.

Card 1 1 2 3 4 5 6 7 8

Variable	FACTIN	DFACTR						
Type	F	F						
Default	1.0	0.01						

VARIABLE

DESCRIPTION

FACTIN	Initial relaxation factor for contact force during each adaptive remesh. To turn this option off set FACTIN=1.0. Unless stability problems occur in the contact, FACTIN=1.0 is recommended since this option can create some numerical noise in the resultant tooling forces. A typical value for this parameter is 0.10.
DFACTR	Incremental increase of FACTIN during each time step after the adaptive step. FACTIN is not allowed to exceed unity. A typical value might be 0.01.

Remarks:

1. This command applies to contact with thickness offsets including contact types:
*CONTACT_FORMING_...,
*CONTACT_NODES_TO_SURFACE,
*CONTACT_SURFACE_TO_SURFACE,
*CONTACT_ONE_WAY_SURFACE_TO_SURFACE.

***CONTROL_ADAPTIVE**

Purpose: Activate adaptive meshing. The parts which are adaptively meshed are defined by *PART. See remarks below.

Card 1 1 2 3 4 5 6 7 8

Variable	ADPFREQ	ADPTOL	ADPOPT	MAXLVL	TBIRTH	TDEATH	LCADP	IOFLAG
Type	F	F	I	I	F	F	I	I
Default	none	10 ²⁰	1	3	0.0	10 ²⁰	0	0

(This card is optional).

Card 2 1 2 3 4 5 6 7 8

Variable	ADPSIZE	ADPASS	IREFLG	ADPENE	ADPTH	MEMORY	ORIENT	MAXEL
Type	F	I	I	F	F	I	I	I
Default		0	0	0.0	inactive	inactive	0	inactive

(This card is optional).

Card 3 1 2 3 4 5 6 7 8

Variable	IADPN90	IADPGH	NCFREQ	IADPCL	ADPCTL	CBIRTH	CDEATH	LCLVL
Type	I	I	I	I	F	F	F	F
Default	0	0	none	1	none	0.0	10 ²⁰	

*CONTROL

*CONTROL_ADAPTIVE

(This card is optional)

Card 4 1 2 3 4 5 6 7 8

Variable	CNLA							
Type	F							
Default	0							

VARIABLE	DESCRIPTION
-----------------	--------------------

- | | |
|---------|--|
| ADPFREQ | Time interval between adaptive refinements, see Figure 8.1. |
| ADPTOL | Adaptive error tolerance in degrees for ADPOPT set to 1 or 2 below. If ADPOPT is set to 8, ADPTOL is the characteristic element size. |
| ADPOPT | Adaptive options:
EQ.1: angle change in degrees per adaptive refinement relative to the surrounding elements for each element to be refined.
EQ.2: total angle change in degrees relative to the surrounding element for each element to be refined. For example, if the adptol=5 degrees, the element will be refined to the second level when the total angle change reaches 5 degrees. When the angle change is 10 degrees the element will be refined to the third level.
EQ.4: adapts when the error norm base on: |

$$\sqrt{\sum_{i,j} (\sigma_{ij} - \hat{\sigma}_{ij})^2}$$

evaluated at the element integration points exceeds ADPTOL/100 times the mean stress in the mesh. The stress $\hat{\sigma}_{ij}$ is the value of the stress interpolated from the least squares fit of the stress to the nodes. This option works for shell types 2, 4, 16, 18, and 20.
 EQ.7: 3D r-adaptive remeshing for solid elements. Solid element type 13, a tetrahedron, and 3-D EFG type 41, are used in the adaptive remeshing process. A completely new mesh is generated which is initialized from the old mesh using a least squares approximation. The mesh size is currently based on the minimum and maximum edge lengths defined on the *CONTROL_REMESHING keyword input. This option remains under development, and, we are not sure of its reliability on complex geometries.

VARIABLE	DESCRIPTION
	<p>Q.181: 2D r-adaptive remeshing for axisymmetric and plane strain solid elements. A completely new mesh is generated which is initialized from the old mesh using a least squares approximation. The mesh size is currently based on the value, ADPTOL, which gives the characteristic element size. This option is based on earlier work by Dick and Harris [1992]. If ADPOPT is negative, then self-contacting material <u>will not</u> be merged together. The self-merging is often preferred since it eliminates sharp folds in the boundary; however, if the sharp fold is being simulated unexpected results are generated.</p>
MAXLVL	<p>Maximum number of refinement levels. Values of 1, 2, 3, 4, ... allow a maximum of 1, 4, 16, 64, ... elements, respectively, to be created for each original element. The refinement level can be overridden by *DEFINE_BOX_ADAPTIVE, or *DEFINE_SET_ADAPTIVE.</p>
TBIRTH	<p>Birth time at which the adaptive remeshing begins, see Figure 8.1.</p>
TDEATH	<p>Death time at which the adaptive remeshing ends, see Figure 8.1.</p>
LCADP	<p>Adaptive interval is changed as a function of time given by load curve ID, LCADP. If this option is nonzero, the ADPFREQ will be replaced by LCADP. The x-axis is time and the y-axis is the varied adaptive time interval.</p>
IOFLAG	<p>Flag to generate adaptive mesh at exit including *NODE, *ELEMENT, *SHELL, and *BOUNDARY_, *CONTACT_NODE_, and *CONSTRAINED_ADAPTIVITY, to be saved in the file, <i>adapt.msh</i>. EQ.1: generate adaptive mesh.</p>
ADPSIZE	<p>Minimum element size to be adapted based on element edge length. If undefined the edge length limit is ignored.</p>
ADPASS	<p>One or two pass adaptivity flag: EQ.0: two pass adaptivity as shown in Figure 8.1a, EQ.1: one pass adaptivity as shown in Figure 8.1b.</p>
IREFLG	<p>Uniform refinement level. A value of 1, 2, 3 ... allow 4, 16, 64 ... elements, respectively, to be created uniformly for each original element. If negative, IIREFLG is taken as a load curve ID. With the curve option, the abscissa values define the refinement time, and the ordinate values define the minimum element size. Only one refinement level is performed per time step. An advantage of the load curve option is that the mesh is adapted to honor the minimum element size, but with the uniform option, IREFLG>0, this is not possible.</p>
<p>Note: If the element size defined with *DEFINE_CURVE is positive the element size will override the element size defined with</p>	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	*CONTROL_ADAPTIVE and *DEFINE_SET_ADAPTIVE. Also, if the element size defined with *DEFINE_CURVE is negative the element size is used for refinement only.
ADPENE	Adapt the mesh when the contact surfaces approach or penetrate the tooling surface depending on whether the value of ADPENE is positive (<i>approach</i>) or negative (<i>penetrates</i>), respectively. The tooling adaptive refinement is based on the curvature of the tooling. If ADPENE is positive the refinement generally occurs before contact takes place; consequently, it is possible that the parameter ADPASS can be set to 1 in invoke the one pass adaptivity.
ADPTH	.EQ.0.0 This parameter is ignored .GT.0.0 Absolute shell thickness level below which adaptive remeshing should began. .LT.0.0 Element thickness ratio. If the ratio of the element thickness to the original element thickness is less than the absolute value of ADPTHK, the element will be refined. <i>This option works only if ADPTOL is nonzero. If thickness based adaptive remeshing is desired without angle changes, then, set ADPTOL to a large angle.</i>
MEMORY	This flag can have two meanings depending on whether the memory environmental variable is or is not set. The command " <i>setenv LSTC_MEMORY auto</i> " sets the memory environmental variable which causes LS-DYNA to expand memory automatically. Note that automatic memory expansion is not always 100% reliable depending on the machine and operating system level; consequently, it is not yet the default. To see if this is set on a particular machine type the command " <i>env</i> ". If the environmental variable <u>is not set</u> then when memory usage reaches this percentage, MEMORY, further adaptivity is prevented to avoid exceeding the memory specified at execution time. Caution is necessary since memory usage is checked after each adaptive step, and, if the memory usage increases by more than the residual percentage, 100-PERCENT, the calculation will terminate. If the memory environmental variable <u>is set</u> then when the number of words of memory allocated reaches or exceeds this value, MEMORY, further adaptivity is stopped.
ORIENT	This option applies to the FORMING contact option only. If this flag is set to one (1), the user orientation for the contact interface is used. If this flag is set to zero (0), LS-DYNA sets the global orientation of the contact surface the first time a potential contact is observed after the birth time. If slave nodes are found on both sides of the contact surface, the orientation is set based on the principle of "majority rules". Experience has shown that this principle is not always reliable.

VARIABLE	DESCRIPTION
MAXEL	Adaptivity is stopped if this number of elements is exceeded.
IADPN90	Maximum number of elements covering 90 degree of radii. See Remark 6.
IADPGH	Fission flag for neighbor splitting. EQ.0: split all neighbor elements EQ.1: do not split neighbor elements
NCFREQ	Frequency of fission to fusion steps. For example, if NCFREQ=4, then fusion will occur on the fourth, eighth, twelfth, etc., fission steps, respectively. If this option is used NCFREQ>1 is recommended.
IADPCL	Fusion will not occur until the fission level reaches IADPCL. Therefore, if IADPCL=2, MAXLVL=5, any element can be split into 256 elements. If the surface flattens out, the number of elements will be reduced if the fusion option is active, i.e., the 256 elements can be fused and reduced to 16.
ADPCTL	Adaptivity error tolerance in degrees for activating fusion. It follows the same rules as ADPOPT above.
CBIRTH	Birth time for adaptive fusion. If ADPENE>0, look-ahead adaptivity is active. In this case, fission, based on local tool curvature, will occur while the blank is still relatively flat. The time value given for CBIRTH should be set to a time later in the simulation after the forming process is well underway.
CDEATH	Death time for adaptive fusion.
LCLVL	Load curve ID of a curve that defines the maximum refinement level as a function of time
CNLA	Limit angle for corner nodes. See Remark 7.

Remarks:

1. D3DUMP and RUNRSF files contain all information necessary to restart an adaptive run. This did not work in version 936 of LS-DYNA.
2. Card 2 input is optional and is not required.
3. In order for this control card to work, the flag ADPOPT=1 must be set in the *PART definition. Otherwise, adaptivity will not function.
4. In order for adaptivity to work optimally, the parameter SNLOG=1, must be set on Optional Control Card B in the *CONTACT Section. On disjoint tooling meshes the contact option *CONTACT_FORMING_..... is recommended.

5. A file *adapt.rid* is left on disk after the adaptive run is completed. This file contains the root ID of all elements that are created during the calculation, and it does not need to be kept if it is not used in post-processing.
6. For springback analysis, IADPN90 is usually chosen between 4 and 6.
7. When using 2D r-adaptive remeshing, the generated new mesh should have a node at each corner so that corners are not smoothed. By default, the mesher will assume a corner wherever the interior angle between adjacent edges is less than 110 degrees. Setting CNLA larger than 110 enables angles larger than 110 to be corners. Care should be taken to avoid an unnecessarily large value of CNLA as this may prevent the mesher from generating smooth meshes.

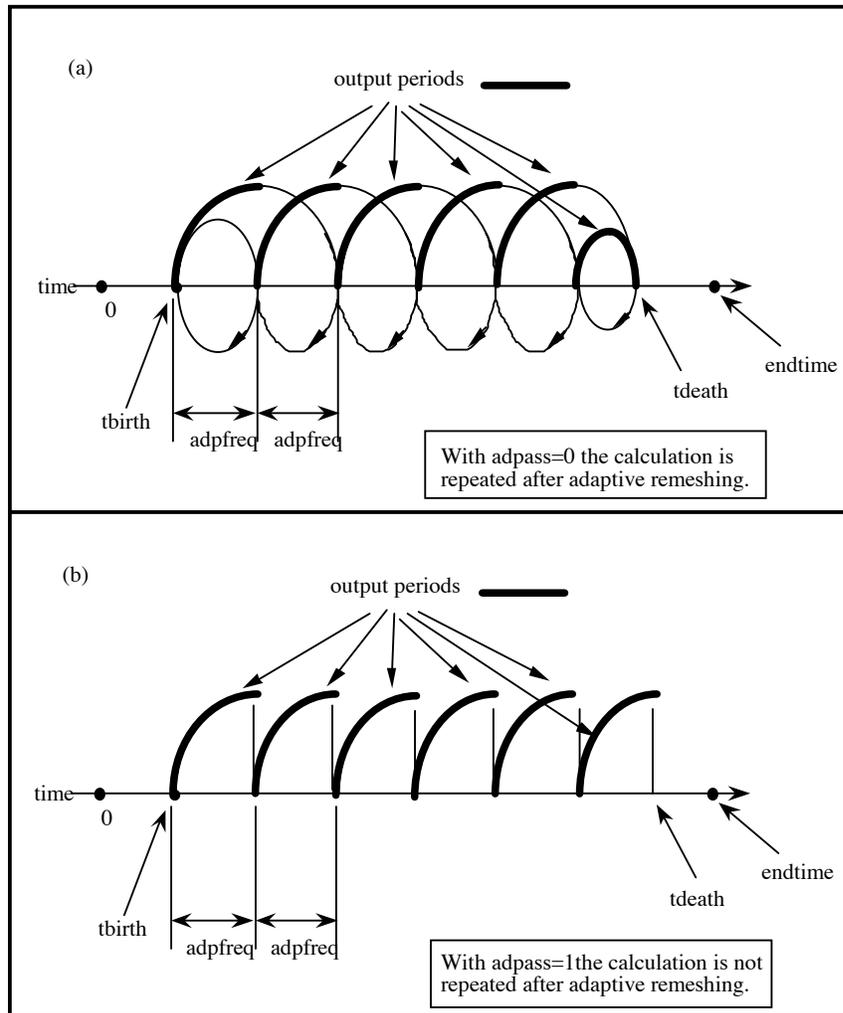


Figure 8.1. At time= t_{birth} the adaptive calculation begins. After computing for a time interval $adpfreq$ error norms are computed. If $ADPASS=0$, then the mesh that existed at time= t_{birth} is refined based on the computed error norms. With the new mesh, the calculation continues to time= $t_{birth}+2 \times adpfreq$ where the error norms are again computed. The mesh that existed at time= $t_{birth}+adpfreq$ is refined and the calculation continues to time= $t_{birth}+3 \times adpfreq$, and so on. However, if $ADPASS=1$, then the mesh that exist at time= $t_{birth}+adpfreq$ is refined and the calculation continues. Errors that develop between adaptive remeshing are preserved. Generally, $ADPASS=0$ is recommended but this option is considerably more expensive.

*CONTROL

*CONTROL_ADAPTIVE_CURVE

*CONTROL_ADAPTIVE_CURVE

Purpose: To refine the element mesh along a curve. All curves defined by the keyword *DEFINE_CURVE_TRIM are used in the refinement. This option provides additional refinement to that created by *CONTRO_ADAPTIVE.

Card 1	1	2	3	4	5	6	7	8
Variable	IDSET	ITYPE	N	SMIN	ITRIOPT			
Type	I	I	I	F	I			

VARIABLE	DESCRIPTION
IDSET	Set ID
ITYPE	Set type: EQ.1: IDSET is shell set ID. EQ.2: IDSET is part set ID.
N	Refinement option: EQ.1: Refine until there are no adaptive constraints remaining in the element mesh around the curve.. GT.1: Refine no more than N levels
SMIN	If the element dimension is smaller than this value, do not refine.
ITRIOPT	Option to refine an enclosed area of a trim curve. EQ.0: Refine the elements along the trim curve EQ.1: Refine the elements along the trim curve and enclosed by the trim curve.

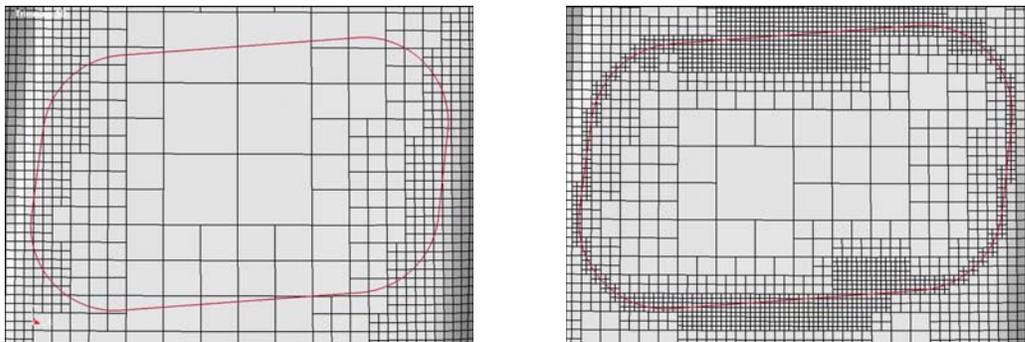


Figure 8.2. Refinement along a curve.

***CONTROL_ALE**

Purpose: Set global control parameters for the Arbitrary Lagrange-Eulerian (ALE) and Eulerian calculations. This is required when ELFORM = 5, 6, 7, 11, and 12.

Card 1 1 2 3 4 5 6 7 8

Variable	DCT	NADV	METH	AFAC	BFAC	CFAC	DFAC	EFAC
Type	I	I	I	F	F	F	F	F
Default	1	0	1	0	0	0	0	0

Card 2

Variable	START	END	AAFAC	VFACT	PRIT	EBC	PREF	NSIDEBC
Type	F	F	F	F	F	I	F	I
Default	0	1.0E+20	1	1.0E-06	0.0	0	0.0	none

Optional Card 3

Card 3 1 2 3 4 5 6 7 8

Variable	NCPL	NBKT	IMASCL	CHECKR				
Type	I	I	I	F				
Default	1	50	0	0.0				

VARIABLE	DESCRIPTION
DCT	Default continuum treatment: EQ.1: Lagrangian (default), EQ.2: Eulerian, EQ.3: Arbitrary Lagrangian Eulerian, EQ.4: Eulerian Ambient.
NADV	Number of cycles between advections (almost always set to 1).
METH	Advection method: EQ.1: donor cell + HIS (Half-Index-Shift, first order accurate), EQ.2: Van Leer + HIS (Half-Index-Shift, second order). EQ.3: donor cell + HIS, first order accurate, conserving total energy over each advection step instead of conserving internal energy (See Remark 5).
AFAC	ALE smoothing weight factor - Simple average: EQ.-1: turn smoothing off. (See Remark 6).
BFAC	ALE smoothing weight factor – Volume weighting
CFAC	ALE smoothing weight factor – Isoparametric
DFAC	ALE smoothing weight factor – Equipotential
EFAC	ALE smoothing weight factor – Equilibrium
START	Start time for ALE smoothing
END	End time for ALE smoothing
AAFAC	ALE advection factor (donor cell options, default=1.0)
VFACT	Volume fraction limit for stresses in single material and void formulation. All stresses are set to zero for elements with lower volume fraction than VFACT. EQ.0.0: set to default 1.0E-06.
PRIT	A flag to turn on or off the pressure equilibrium iteration option for multi-material elements (See Remark 1). EQ.0: Off (default) EQ.1: On
EBC	Automatic Eulerian boundary condition (See Remark 2). EQ.0: Off EQ.1: On with stick condition EQ.2: On with slip condition

VARIABLE	DESCRIPTION
PREF	Reference pressure applied to the free surfaces of the ALE mesh boundary. (See Remark 3).
NSIDEBC	A node set ID (NSID) which is to be excluded from the EBC constraint.
NCPL	Number of Lagrangian cycles between coupling calculations. This is typically done every cycle; therefore, its default is 1. This is on optional card 3.
NBKT	Number of Lagrangian cycles between global bucket-sort searches to locate the position of the Lagrangian structure (mesh) relative to the ALE fluid (mesh). Default is 50. This is on optional card 3.
IMASCL	A flag for turning ON/OFF mass scaling for ALE parts. The global mass scaling control (parameter DT2MS under *CONTROL_TIMESTEP card) must be ON. If the run dt is lower than the mass scaling dt, then IMASCL has the following effects: EQ.0: (Default) No mass scaling for ALE parts. Print out maximum 20 warnings. EQ.1: No mass scaling for ALE parts. Stop the run. EQ.2: Do mass scaling for ALE parts (the result may not be correct due to this scaling).
CHECKR	A parameter for reducing or eliminating an ALE pressure locking pattern. It may range from 0.01 to 0.1 (See Remark 4).

Remarks:

1. By default, all materials in a multi-material element are assumed to undergo the same element averaged strain rates. This assumption may not be robust when mixing materials with very different compressibility. In this case, an assumption of pressure equilibrium (PRIT=1) in the element may be more appropriate.
2. This option, used for EULER formulations, automatically defines velocity boundary condition constraints for the user. The constraints, once defined, are applied to all nodes on free surfaces of an Eulerian domain. For problems where the normal velocity of the material at the boundary is zero such as injection molding problems, the automatic boundary condition parameter is set to 2. This will play the same role as the nodal single point constraint. For EBC=1, the material velocity of all free surface nodes of an Eulerian domain is set to zero.
3. The PREF definition is equivalent to using the *LOAD_SEGMENT card to provide pressure loading on the free surfaces of all ALE or Eulerian mesh(es) in a model. This cannot be used to initialize the internal pressure of the material (that must be done via the *EOS_ or *BOUNDARY_AMBIENT_EOS cards).

4. Due to one point integration, ALE elements may experience a spatial instability in the pressure field referred to as checker boarding. CHECKR is a scale for diffusive flux calculation to alleviate this problem.
5. Generally, it is not possible to conserve both momentum and kinetic energy (KE) at the same time. Typically, internal energy (IE) is conserved and KE may not be. This may result in some KE loss (hence, total energy loss). For many analyses this is tolerable, but for airbag application, this may lead to the reduction of the inflating potential of the inflator gas. METH=3 tries to eliminate this loss in KE over the advection step by storing any loss KE under IE, thus conserving total energy of the system.
6. All the smoothing factors (AFAC, BFAC, CFAC, DFAC, EFAC) are generally most applicable to ELFORM=5 (single material ALE formulation).
7. Although this card has many parameters, only a few are required definitions. Typically, one can try, as a first run, setting NADV=1, METH=1, AFAC=-1 and the rest as "0". Sometimes when needed, PREF may be defined. This is adequate for most cases. METH may be changed to 2 or 3 later depending on the physics of the problem during fine-tuning of the model.
8. Due to the fact that we have internal forces at the nodes, while the pressure is stored at the element center, sometimes there is a "checker-board pattern" in the pressure distribution. It is a kind of locking effect that normally occurs only in problems having very small volumetric strains, i.e., at small pressures. "CHECKR" is designed for alleviating this problem.

***CONTROL_BULK_VISCOSITY**

Purpose: Reset the default values of the bulk viscosity coefficients globally. This may be advisable for shock wave propagation and some materials. Bulk viscosity is used to treat shock waves. A viscous term q is added to the pressure to smear the shock discontinuities into rapidly varying but continuous transition regions. With this method the solution is unperturbed away from a shock, the Hugoniot jump conditions remain valid across the shock transition, and shocks are treated automatically.

Card 1 2 3 4 5 6 7 8

Variable	Q1	Q2	TYPE					
Type	F	F	I					
Default	1.5	.06	1					

VARIABLE**DESCRIPTION**

Q1	Default quadratic viscosity coefficient.
Q2	Default linear viscosity coefficient.
TYPE	Default bulk viscosity type, IBQ (Default=1) EQ. -2: standard (also types 2, 4, 10, 16, and 17). With this option the internal energy dissipated by the viscosity in the shell elements is computed and included in the overall energy balance. EQ. -1: standard (also types 2, 4, 10, 16, and 17 shell elements). The internal energy is not computed in the shell elements. EQ.+1: standard . Solid elements only and internal energy is always computed and included in the overall energy balance.

Remarks:

The bulk viscosity creates an additional additive pressure term given by:

$$q = \rho l (Q_1 l \dot{\epsilon}_{kk}^2 - Q_2 a \dot{\epsilon}_{kk}) \quad \text{if } \dot{\epsilon}_{kk} < 0$$

$$q = 0 \quad \text{if } \dot{\epsilon}_{kk} \geq 0$$

where Q_1 and Q_2 are dimensionless input constants which default to 1.5 and .06, respectively, and l is a characteristic length given as the square root of the area in two dimensions and as the cube root of the volume in three, a is the local sound speed, Q_1 defaults to 1.5 and Q_2 defaults to .06. See Chapter 21 in the LS-DYNA Theory Manual for more details.

*CONTROL

*CONTROL_CHECK

*CONTROL_CHECK_{OPTION}

Available options include:

<BLANK>

SHELL

Purpose: Check for various problems in the mesh. Checking is performed during the input phase.

For the SHELL option, shell element integrity checks which have been identified as important in metal forming applications are performed. These checks can improve springback convergence and accuracy. This option will repair bad elements created, for example, during trimming operations.

(OPTION = SHELL) (include one card for each part to be checked)

Card 1	1	2	3	4	5	6	7	8
--------	---	---	---	---	---	---	---	---

Variable	PID	IFAUTO	CONVEX	ADPT	ARATIO	ANGLE	SMIN	
Type	I	I	I	I	F	F	F	
Default	0	0	1	1	0.25	150.0	0.0	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID to be checked: EQ.0: do not check
IFAUTO	Flag to automatically correct bad elements: EQ.0: write warning message only EQ.1: fix bad element, write message
CONVEX	Check element convexity (internal angles less than 180 degrees) EQ.0: do not check EQ.1: check
ADPT	Check adaptive constraints EQ.0: do not check EQ.1: check
ARATIO	Minimum allowable aspect ratio. Elements which do not meet minimum aspect ratio test will be treated according to IFAUTO above.

VARIABLE	DESCRIPTION
ANGLE	Maximum allowable internal angle. Elements which fail this test will be treated according to IFAUTO above.
SMIN	Minimum element size. Elements which fail this test will be treated according to IFAUTO above.

Remarks:

1. If the convexity test is activated, all failed elements will be fixed regardless of IFAUTO.
2. In addition to illegal constraint definitions (slave which is also a master), checks are performed for mesh connectivities which have been found to cause convergence trouble in implicit springback applications.

*CONTROL

*CONTROL_COARSEN

*CONTROL_COARSEN

Purpose: Adaptively de-refine (coarsen) a shell mesh by selectively merging four adjacent elements into one. Adaptive constraints are added and removed as necessary.

Card 1 1 2 3 4 5 6 7 8

Variable	ICOARSE	ANGLE	NSEED	PSID	SMAX			
Type	I	F	I	I	F			
Default	0	none	0	0	0			

Card 2

Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ICOARSE	Coarsening flag: EQ.0: do not coarsen (default), EQ.1: coarsen mesh at beginning of simulation for forming model EQ.2: coarsen mesh at beginning of simulation for crash model
ANGLE	Allowable angle change between neighboring elements. Adjacent elements which are flat to within ANGLE degrees are merged. (Suggested starting value = 8.0 degrees)
NSEED	Number of seed nodes (optional). EQ.0: use only automatic searching. EQ.n: also search starting with node IDs given below (maximum = 8 nodes)

VARIABLE	DESCRIPTION
PSID	Part set ID. All the parts defined in this set will be prevented from been coarsened.
SMAX	Maximum element size. For ICOARSE=2, no elements larger than this size will be created.
N1...N8	Optional list of seed node IDs for extra searching.

Remarks:

1. Coarsening is performed at the start of a simulation. The first plot state represents the coarsened mesh. By setting the termination time to zero and including the keyword *INTERFACE_SPRINGBACK_LSDYNA a keyword input deck can be generated containing the coarsened mesh.
2. By default, an automatic search is performed to identify elements for coarsening. In some meshes, isolated regions of refinement may be overlooked. Seed nodes can be identified in these regions to assist the automatic search. Seed nodes identify the central node of a four-element group which is coarsened into a single element if the angle criterion is satisfied.
3. The keyword *DEFINE_BOX_COARSEN can be used to indicate regions of the mesh which are protected from coarsening.

*CONTROL

*CONTROL_CONTACT

*CONTROL_CONTACT

Purpose: Change defaults for computation with contact surfaces.

Card 1 1 2 3 4 5 6 7 8

Variable	SLSFAC	RWPNAL	ISLCHK	SHLTHK	PENOPT	THKCHG	ORIEN	ENMASS
Type	F	F	I	I	I	I	I	I
Default	.1	none	1	0	1	0	1	0

Card 2

Variable	USRSTR	USRFRC	NSBCS	INTERM	XPENE	SSTHK	ECDT	TIEDPRJ
Type	I	I	I	I	F	I	I	I
Default	0	0	10-100	0	4.0	0	0	0

Card 3 is optional. The following parameters are the default values used by parts in automatic contacts. These frictional coefficients apply only to contact types: SINGLE_SURFACE, AUTOMATIC_GENERAL, AUTOMATIC_SINGLE_SURFACE, AUTOMATIC_NODES_TO_..., AUTOMATIC_SURFACE_..., and AUTOMATIC_ONE_WAY_..., and ERODING_SINGLE_SURFACE. Also see *CONTACT and *PART. Note that these default values will override the values specified for these contact types in the *CONTACT section.

Card 3 1 2 3 4 5 6 7 8

Variable	SFRIC	DFRIC	EDC	VFC	TH	TH_SF	PEN_SF	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

***CONTROL_CONTACT**

***CONTROL**

Card 4 is optional. If this card is defined, then Card 3 above must be included. A blank card may be inserted for Card 3.

Card 4 1 2 3 4 5 6 7 8

Variable	IGNORE	FRCENG	SKIPRWG	OUTSEG	SPOTSTP	SPOTDEL	SPOTHIN	
Type	I	I	I	I	I	I	F	
Default	0	0	0	0	0	0	inactive	

Card 5 is optional. If this card is defined, then Cards 3 and 4 above must be included. Blank cards may be inserted.

Card 5 1 2 3 4 5 6 7 8

Variable	ISYM	NSEROD	RWGAPS	RWGDTH	RWKSF	ICOV	SWRADF	ITHOFF
Type	I	I	I	F	F	I	F	I
Default	0	0	0	0.	1.0	0	0.	0

Card 6 is optional. If this card is defined, then Cards 3 to 5 above must be included. Blank cards may be inserted.

Card 5 1 2 3 4 5 6 7 8

Variable	SHLEDG							
Type	I							
Default	0							

VARIABLE

DESCRIPTION

SLSFAC

Scale factor for sliding interface penalties, SLSFAC:
EQ.0: default = .1.

VARIABLE	DESCRIPTION
RWPNAL	<p>Scale factor for rigid wall penalties, which treat nodal points interacting with rigid walls, RWPNAL. The penalties are set so that an absolute value of unity should be optimal; however, this penalty value may be very problem dependent. If rigid/deformable materials switching is used, this option should be used if the switched materials are interacting with rigid walls.</p> <p>LT.0.0: all nodes are treated by the penalty method. This is required for implicit calculations. Since seven (7) variables are stored for each slave node, only the nodes that may interact with the wall should be included in the node list.</p> <p>EQ.0.0: the constraint method is used and nodal points which belong to rigid bodies are not considered.</p> <p>GT.0.0: rigid bodies nodes are treated by the penalty method and all other nodes are treated by the constraint method.</p>
ISLCHK	<p>Initial penetration check in contact surfaces with indication of initial penetration in output files (see remarks below):</p> <p>EQ.0: the default is set to 1, EQ.1: no checking, EQ.2: full check of initial penetration is performed.</p>
SHLTHK	<p>Shell thickness considered in type surface to surface and node to surface type contact options, where options 1 and 2 below activate the new contact algorithms. The thickness offsets are always included in single surface, constraint method, and automatic surface to surface and node to surface contact types (See remarks below.):</p> <p>EQ.0: thickness is not considered, EQ.1: thickness is considered but rigid bodies are excluded, EQ.2: thickness is considered including rigid bodies.</p>
PENOPT	<p>Penalty stiffness value option. For default calculation of the penalty value please refer to the LS-DYNA Theory Manual.</p> <p>EQ.0: the default is set to 1, EQ.1: minimum of master segment and slave node (default for most contact types), EQ.2: use master segment stiffness (old way), EQ.3: use slave node value, EQ.4: use slave node value, area or mass weighted, EQ.5: same as 4 but inversely proportional to the shell thickness. This may require special scaling and is not generally recommended. Options 4 and 5 can be used for metalforming calculations.</p>
THKCHG	<p>Shell thickness changes considered in single surface contact:</p> <p>EQ.0: no consideration (default), EQ.1: shell thickness changes are included.</p>

VARIABLE	DESCRIPTION
ORIEN	Optional automatic reorientation of contact interface segments during initialization: EQ.0: default is set to 1. EQ.1: active for automated (part) input only. Contact surfaces are given by *PART definitions. EQ.2: active for manual (segment) and automated (part) input. EQ.3: inactive for non-forming contact. EQ.4: inactive for forming contact.
ENMASS	Treatment of the mass of eroded nodes in contact. This option affects all contact types where nodes are removed after surrounding elements fail. Generally, the removal of eroded nodes makes the calculation more stable; however, in problems where erosion is important the reduction of mass will lead to incorrect results. EQ.0: eroding nodes are removed from the calculation. EQ.1: eroding nodes of solid elements are retained and continue to be active in contact. EQ.2: the eroding nodes of solid and shell elements are retained and continue to be active in contact.
USRSTR	Storage per contact interface for user supplied interface control subroutine, see Appendix F. If zero, no input data is read and no interface storage is permitted in the user subroutine. This storage should be large enough to accommodate input parameters and any history data. This input data is available in the user supplied subroutine.
USRFRC	Storage per contact interface for user supplied interface friction subroutine, see Appendix G. If zero, no input data is read and no interface storage is permitted in the user subroutine. This storage should be large enough to accommodate input parameters and any history data. This input data is available in the user supplied subroutine.
NSBCS	Number of cycles between contact searching using three dimensional bucket searches. Defaults recommended.
INTERM	Flag for intermittent searching in old surface-to-surface contact using the interval specified as NSBCS above: EQ.0: off, EQ.1: on.
XPENE	Contact surface maximum penetration check multiplier. If the small penetration checking option, PENCHK, on the contact surface control card is active, then nodes whose penetration then exceeds the product of XPENE and the element thickness are set free, see *CONTACT_OPTION_...: EQ.0: default is set to 4.0.

VARIABLE	DESCRIPTION
SSTHK	Flag for using actual shell thickness in single surface contact logic-types 4, 13, 15 and 26. See remarks 1 and 2 below. EQ.0: Actual shell thickness is not used in the contacts. (default), EQ.1: Actual shell thickness is used in the contacts. (sometimes recommended for metal forming calculations).
ECDT	Time step size override for eroding contact: EQ.0: contact time size may control Dt. EQ.1: contact is not considered in Dt determination.
TIEDPRJ	Bypass projection of slave nodes to master surface in types: *CONTACT_TIED_NODES_TO_SURFACE, *CONTACT_TIED_SHELL_EDGE_TO_SURFACE, and *CONTACT_TIED_SURFACE_TO_SURFACE tied interface options: EQ.0: eliminate gaps by projection nodes, EQ.1: bypass projection. Gaps create rotational constraints which can substantially affect results.
SFRIC	Default static coefficient of friction (see *PART_CONTACT)
DFRIC	Default dynamic coefficient of friction (see *PART_CONTACT)
EDC	Default exponential decay coefficient (see *PART_CONTACT)
VFC	Default viscous friction coefficient (see *PART_CONTACT)
TH	Default contact thickness (see *PART_CONTACT)
TH_SF	Default thickness scale factor (see *PART_CONTACT)
PEN_SF	Default local penalty scale factor (see *PART_CONTACT)
IGNORE	Ignore initial penetrations in the *CONTACT_AUTOMATIC options. In the SMP contact this flag is not implement for the AUTOMATIC_GENERAL option. "Initial" in this context refers to the first timestep that a penetration is encountered. This option can also be specified for each interface on the third optional card under the keyword, *CONTACT. The value defined here will be the default. EQ.0: move nodes to eliminate initial penetrations in the model definition. EQ.1: allow initial penetrations to exist by tracking the initial penetrations. EQ.2: allow initial penetrations to exist by tracking the initial penetrations. However, penetration warning messages are printed with the original coordinates and the recommended coordinates of each slave node given.

VARIABLE	DESCRIPTION
FRCENG	Flag to activate the calculation of frictional sliding energy: EQ.0: do not calculate, EQ.1: calculate frictional energy in contact and store as “Surface Energy Density” in the binary INTFOR file. Convert mechanical frictional energy to heat when doing a coupled thermal-mechanical problem. When PKP_SEN=1 on the keyword card *DATABASE_EXTENT_BINARY, it is possible to identify the energies generated on the upper and lower shell surfaces, which is important in metal forming applications. This data is mapped after each H-adaptive remeshing.
SKIPRWG	Flag not to display stationary rigid wall by default. EQ.0: generate 4 extra nodes and 1 shell element to visualize stationary planar rigid wall. EQ.1: do not generate stationary rigid wall.
OUTSEG	Flag to output each beam spot weld slave node and its master segment for contact type: *CONTACT_SPOTWELD into the D3HSP file. EQ.0: no, do not write out this information. EQ.1: yes, write out this information.
SPOTSTP	If a spot weld node or face, which is related to a *MAT_SPOTWELD beam or solid element, respectively, cannot be found on the master surface, should an error termination occur? EQ.0: no, continue calculation, EQ.1: yes, print error message and terminate, EQ.2: no, delete unconstrained weld and continue calculation.
SPOTDEL	If the nodes of a spot weld beam or solid element are attached to a shell element that fails and are deleted, then the attached spot weld element is deleted if this flag is on. There is a small cost penalty related to this option on non-vector processors. On vector processors, however, this option can significantly slow down the calculation if many weld elements fail since the vector lengths are reduced. EQ.0: no, do not delete the spot weld beam or solid element, EQ.1: yes, delete the weld elements when the attached shells on one side of the element fail.
SPOTHIN	Optional thickness scale factor. If active, define a factor greater than zero, but less than one. Premature failure of spot welds can occur due to contact of the spot welded parts in the vicinity of the spot weld. This contact creates tensile forces in the spot weld. Although this seems physical, the compressive forces generated in the contact are large enough to fail the weld in tension before failure is observed in experimental test. With this option, the thickness of the parts in the vicinity of the weld are automatically scaled, the contact forces do not develop, and the problem is avoided. We recommend setting the IGNORE option to 1 or 2 if SPOTHIN is active. This option applies

VARIABLE	DESCRIPTION
	only to the AUTOMATIC_SINGLE_SURFACE option.
ISYM	Symmetry plane option default for automatic segment generation when contact is defined by part ID's: EQ.0: off, EQ.1: do not include faces with normal boundary constraints (e.g., segments of brick elements on a symmetry plane). This option is important to retain the correct boundary conditions in the model with symmetry.
NSEROD	Flag to use one-way node to surface erosion EQ.0: use two-way algorithm EQ.1: use one-way algorithm
RWGAPS	Flag to add rigid wall gap stiffness, see parameter RWGDTH below. EQ.1: add gap stiffness EQ.2: do not add gap stiffness
RWGDTH	Death time for gap stiffness. After this time the gap stiffness is no longer added.
RWKSF	Rigid wall penalty scale factor for contact with deformable parts during implicit calculations. This value is independent of SLSFAC and RWPNAL. If RWKSF is also specified in *RIGIDWALL_PLANAR, the stiffness is scaled by the product of the two values.
ICOV	Invokes the covariant formulation of Konyukhov and Schweizerhof in the FORMING contact option. This option is available in the third revision of version 971, but is not recommended since it is still being implemented. EQ.0: standard formulation (default) EQ.1: covariant contact formulation.
SWRADF	Spot weld radius scale factor for neighbor segment thinning EQ.0: neighbor segments not thinned (default) GT.0: The radius of beam spot welds are scaled by SWRADF when searching for close neighbor segments to thin.
ITHOFF	Flag for offsetting thermal contact surfaces for thick thermal shells EQ.0: No offset, if thickness is not included in the contact the heat will be transferred between the mid-surfaces of the corresponding contact segments (shells). EQ.1: Offsets are applied so that contact heat transfer is always between the outer surfaces of the contact segments (shells).
SHLEDG	Flag for assuming edge shape for shells when measuring penetration. This is available for segment based contact (see SOFT on *CONTACT) EQ.0: Shell edges are assumed round (default), EQ.1: Shell edges are assumed square.

Remarks:

1. The shell thickness change option must be activated in CONTROL_SHELL control input (see ISTUPD) and a nonzero flag specified for SHLTHK above before the shell thickness changes can be included in the surface-to-surface contact types. An additional flag must be set, see THKCHG above, if thickness changes are included in the single surface contact algorithms. The contact algorithms that include the shell thickness are relatively recent and are now fully optimized and parallelized. The searching in these algorithms is considerably more extensive and therefore slightly more expensive.
2. In the single surface contacts types SINGLE_SURFACE, AUTOMATIC_SINGLE_SURFACE, AUTOMATIC_GENERAL, AUTOMATIC_GENERAL_INTERIOR and ERODING_SINGLE_SURFACE, the default contact thickness is taken as the smaller of two values -- the shell thickness or 40% of the minimum edge length. This may create unexpected difficulties if it is the intent to include thickness effects when the in-plane shell element dimensions are less than the thickness. The default is based on years of experience where it has been observed that sometimes rather large nonphysical thicknesses are specified to achieve high stiffness values. Since the global searching algorithm includes the effects of shell thicknesses, it is possible to slow the searches down considerably by using such nonphysical thickness dimensions.
3. The initial penetration check option is always performed in v. 950 irregardless of the value of ISLCHK. If you do not want to remove initial penetrations then set the contact birth time (see *CONTACT_...) so that the contact is not active at time 0.
4. Automatic reorientation requires offsets between the master and slave surface segments. The reorientation is based on segment connectivity and, once all segments are oriented consistently based on connectivity, a check is made to see if the master and slave surfaces face each other based on the right hand rule. If not, all segments in a given surface are reoriented. This procedure works well for non-disjoint surfaces. If the surfaces are disjoint, the AUTOMATIC contact options, which do not require orientation, are recommended. In the FORMING contact options automatic reorientation works for disjoint surfaces.
5. If SPOTHIN is greater than zero and SWRADF is greater than zero, a neighbor segment thinning option is active. The radius of a beam spot weld is scaled by SWRADF, and then a search is made for shell segments that are neighbors of the tied shell segments that are touched by the weld but not tied by it.

*CONTROL

*CONTROL_COUPLING

*CONTROL_COUPLING

Purpose: Change defaults for MADYMO3D/CAL3D coupling, see Appendix I.

Card 1 1 2 3 4 5 6 7 8

Variable	UNLENG	UNTIME	UNFORC	TIMIDL	FLIPX	FLIPY	FLIPZ	SUBCYL
Type	F	F	F	F	I	I	I	I
Default	1.	1.	1.	0.	0	0	0	1

VARIABLE

DESCRIPTION

UNLENG	Unit conversion factor for length. MADYMO3D/GM-CAL3D lengths are multiplied by UNLENG to obtain LS-DYNA lengths.
UNTIME	Unit conversion factor for time, UNTIME. MADYMO3D/GM-CAL3D time is multiplied by UTIME to obtain LS-DYNA time.
UNFORC	Unit conversion factor for force, UNFORC. MADYMO3D/GM-CAL3D force is multiplied by UNFORC to obtain LS-DYNA force.
TIMIDL	Idle time during which CAL3D or MADYMO is computing and LS-DYNA remains inactive. Important for saving computer time.
FLIPX	Flag for flipping X-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model: EQ.0: off, EQ.1: on.
FLIPY	Flag for flipping Y-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model: EQ.0: off, EQ.1: on.
FLIPZ	Flag for flipping Z-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model: EQ.0: off, EQ.1: on.

VARIABLE	DESCRIPTION
SUBCYL	CAL3D/MADYMO3D subcycling interval (# of cycles): EQ.0: Set to 1, EQ.n: number of LS-DYNA time steps between each CAL3D/ MADYMO3D step. Then the position of the contacting rigid bodies is assumed to be constant for n LS-DYNA time steps. This may result in some increase in the spikes in contact, thus this option should be used carefully. As the CAL3D/MADYMO3D programs usually work with a very small number of degrees of freedom, not much gain in efficiency can be achieved.

*CONTROL

*CONTROL_CPU

*CONTROL_CPU

Purpose: Control cpu time.

Card 1 2 3 4 5 6 7 8

Variable	CPUTIM	IGLST						
Type	F	I						

VARIABLE

DESCRIPTION

CPUTIM

Seconds of cpu time:
EQ.0.0: no cpu time limit set

IGLST

Flag for outputting cpu and elapsed times in glstat file
EQ.0: no
EQ.1: yes

Remarks:

The CPU time limit applies to the current phase of the analysis or restart. The limit is not checked until after the initialization stage of the calculation. Upon reaching the cpu limit, the code will output a restart dump file and terminate. The CPU limit can also be specified on the input control line to LS-DYNA. If a value is specified on both the control line and in the input deck, the minimum value will be used.

***CONTROL_DYNAMIC_RELAXATION**

Purpose: Initialize stresses and deformation in a model to simulate a preload. Examples of preload include load due to gravity, load due to a constant angular velocity, and load due to torquing of a bolt. After the preloaded state is achieved by one of three methods described below, the time resets to zero and the normal phase of the solution automatically begins from the preloaded state.

IDRFLG controls the manner in which the preloaded state is computed. If IDRFLG is 1 or -1, a transient 'dynamic relaxation' analysis is invoked in which an explicit analysis, damped by means of scaling nodal velocities by the factor DRFCTR each time step, is performed. When the ratio of current distortional kinetic energy to peak distortional kinetic energy falls below the convergence tolerance (DRTOL) or when the time reaches DRTERM, the dynamic relaxation analysis stops and the current state becomes the initial state of the subsequent normal analysis. Distortional kinetic energy is total kinetic energy less the kinetic energy due to rigid body motion. A history of the distortional kinetic energy computed during the dynamic relaxation phase is automatically written to a file called "relax". This file can be read as an "ASCII" file by LS-PrePost and its data plotted. To create a binary output database having the same format as a d3plot database but which pertains to the dynamic relaxation analysis, use *DATABASE_BINARY_D3DRLF. The output interval is given by this command as an integer representing the number of convergence checks between output states. The frequency of the convergence checks is controlled by the parameter NRCYCK. Dynamic relaxation will be invoked if SIDR is set to 1 or 2 in any of the *DEFINE_CURVE commands, even if there is no *CONTROL_DYNAMIC_RELAXATION command. Curves so tagged are applicable to the preload analysis phase. Curves with SIDR set to 0 or 2 are applicable to the normal phase of the solution. At the conclusion of the dynamic relaxation phase and before the start of the normal solution phase, a binary dump file (d3dump01) and a 'prescribed geometry' file (drdisp.sif) are written by LS-DYNA. Either of these files can be used in a subsequent analysis to quickly initialize to the preloaded state without having to repeat the dynamic relaxation run. The binary dump file is utilized via a restart analysis (see the *RESTART section of the LS-DYNA Users Manual). The drdisp.sif file is utilized by setting IDRFLG=2 as described below and discussed in Remark 1.

If IDRFLG is set to 2, the preloaded state is quickly reached by linearly ramping nodal displacements, rotations, and temperatures to prescribed values over 100 time steps. See Remark 1.

If IDRFLG is set to 5, an implicit analysis is performed to obtain the preloaded state and in this case, the preload analysis completes when 'time' is equal to DRTERM. The implicit step size is specified with a *CONTROL_IMPLICIT_GENERAL command. The implicit analysis is, by default, static but can be made transient via the *CONTROL_IMPLICIT_DYNAMICS command (see Remark 3).

*CONTROL

*CONTROL_DYNAMIC_RELAXATION

Card	1	2	3	4	5	6	7	8
Variable	NRCYCK	DRTOL	DRFCTR	DRTERM	TSSFDR	IRELAL	EDTTL	IDRFLG
Type	I	F	F	F	F	I	F	I
Default	250	0.001	0.995	infinity	TSSFAC	0	0.04	0
Remarks				3				1, 2, 3

VARIABLE

DESCRIPTION

NRCYCK	Number of iterations between convergence checks, for dynamic relaxation option (default = 250).
DRTOL	Convergence tolerance for dynamic relaxation option (default = 0.001).
DRFCTR	Dynamic relaxation factor (default = .995).
DRTERM	Optional termination time for dynamic relaxation. Termination occurs at this time or when convergence is attained (default = infinity).
TSSFDR	Scale factor for computed time step during dynamic relaxation. If zero, the value is set to TSSFAC defined on *CONTROL_TIMESTEP. After converging, the scale factor is reset to TSSFAC.
IRELAL	Automatic control for dynamic relaxation option based on algorithm of Papadrakakis [1981]: EQ.0: not active, EQ.1: active.
EDTTL	Convergence tolerance on automatic control of dynamic relaxation.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IDRFLG	Dynamic relaxation flag for stress initialization: EQ.-999: dynamic relaxation not activated even if specified on a load curve, see *DEFINE_CURVE, EQ.-1: dynamic relaxation is activated and time history output is produced during dynamic relaxation, see Remark 2. EQ.0: not active, EQ.1: dynamic relaxation is activated, EQ.2: initialization to a prescribed geometry, see Remark 1, EQ.5: initialize implicitly and run explicitly, see Remark 3.

Remarks:

1. When IDRFLG=2, an ASCII file specified by "m=" on the LS-DYNA execution line is read which describes the initialized state. The ASCII file contains each node ID with prescribed values of nodal displacement (x, y, z), nodal rotation (x, y, z) and nodal temperature in (I8, 7E15.0) format.
2. If IDRFLG is set to -1 the dynamic relaxation proceeds as normal but time history data is written to the D3THDT file in addition to the normal data being written to the D3DRLF file. At the end of dynamic relaxation, the problem time is reset to zero. However, information is written to the D3THDT file with an increment to the time value. The time increment used is reported at the end of dynamic relaxation.
3. When IDRFLG=5, LS-DYNA performs an implicit dynamic relaxation by invoking the implicit solver. Parameters for implicit dynamic relaxation can be defined using appropriate CONTROL_IMPLICIT keywords to specify solver type, implicit time step, etc. When using this option, it is required to set appropriate value for DTERM to indicate the termination of the implicit dynamic relaxation. When DTERM is reached during the implicit dynamic relaxation phase, LS-DYNA switches to either implicit or explicit solver depending on IMFLAG in *CONTROL_IMPLICIT_GENERAL. For example, if it is desired to run an implicit dynamic relaxation phase and switch to explicit solver, IMFLAG should be set to 0.

*CONTROL

*CONTROL_EFG

*CONTROL_EFG

Purpose: Define controls for the mesh-free computation.

Card 1 1 2 3 4 5 6 7 8

Variable	ISPLANE	IDILA	ININT					
Type	I	I	I					
Default	0	0	12					
Remarks			1					

Card 2

Variable	IMLM	ETOL						
Type	I	F						
Default	0	1.eE-4						

VARIABLE

DESCRIPTION

- ISPLINE Optional choice for the mesh-free kernal functions:
 EQ.0: Cubic spline function (default)
 EQ.1: Quadratic spline function
 EQ.2: Cubic spline function with circular disk.
- IDILA Optional choice for the normalized dilation parameter:
 EQ.0: Maximum distance based on the background element.
 EQ.1: Maximum distance based on surrounding nodes
- ININT This is the factor needed for the estimation of maximum workspace
(MWSPAC) that can be used during the initialization phase.

VARIABLE	DESCRIPTION
IMLM	Optional choice for the matrix operation, linear solving and memory usage: EQ.1: Original BCSLIB-EXT solvers. EQ.2: EFGPACK.
ETOL	Error tolerance in the IMLM. When IMLM=2 is used, ININT in card one becomes redundant. IMLM = 2 is recommended.

Remarks:

1. The mesh-free computation requires calls to use BCSLIB-EXT solvers during the initialization phase. The maximum workspace (MWSPAC) that can be used during the call is calculated according to:

$$MWSPAC = ININT^{**3} * NUMNEFG$$

where NUMNEFG is the total number of mesh-free nodes. The ININT default value is 12. This value implicitly tells you how many nodes are within the domain of influence per each node in one direction. Increasing this number when the larger normalized dilation parameters are used.

2. When ISPLINE=2 is used, the input of the normalized dilation parameters (DX, DY, DZ) for the kernel function in *SECTION_SOILD_EFG and SECTIOL_SHELL_EFG only requires DX value.
3. EFGPACK was added to automatically compute the required maximum workspace in the initialization phase and to improve efficiency in the matrix operation, linear solving and memory usage. The original BCSLIB-EXT solver requires an explicit workspace (ININT) for the initialization.

*CONTROL

*CONTROL_ENERGY

*CONTROL_ENERGY

Purpose: Provide controls for energy dissipation options.

Card	1	2	3	4	5	6	7	8
Variable	HGEN	RWEN	SLNTEN	RYLEN				
Type	I	I	I	I				
Default	1	2	1	1				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
HGEN	Hourglass energy calculation option. This option requires significant additional storage and increases cost by ten percent: EQ.1: hourglass energy is not computed (default), EQ.2: hourglass energy is computed and included in the energy balance. The hourglass energies are reported in the ASCII files GLSTAT and MATSUM, see *DATABASE_OPTION.
RWEN	Stonewall energy dissipation option: EQ.1: energy dissipation is not computed, EQ.2: energy dissipation is computed and included in the energy balance (default). The stonewall energy dissipation is reported in the ASCII file GLSTAT, see *DATABASE_OPTION.
SLNTEN	Sliding interface energy dissipation option (This parameter is always set to 2 if contact is active. The option SLNTEN=1 is not available.): EQ.1: energy dissipation is not computed, EQ.2: energy dissipation is computed and included in the energy balance. The sliding interface energy is reported in ASCII files GLSTAT and SLEOUT, see *DATABASE_OPTION.
RYLEN	Rayleigh energy dissipation option (damping energy dissipation): EQ.1: energy dissipation is not computed (default), EQ.2: energy dissipation is computed and included in the energy balance. The damping energy is reported in ASCII file GLSTAT, see *DATABASE_OPTION.

***CONTROL_EXPLOSIVE_SHADOW**

Purpose: Compute detonation times in explosive elements for which there is no direct line of sight. If this control card is missing, the lighting time for an explosive element is computed using the distance from the center of the element to the nearest detonation point, L_d ; the detonation velocity, D ; and the lighting time for the detonator, t_d :

$$t_L = t_d + \frac{L_d}{D}$$

The detonation velocity for this option is taken from the element whose lighting time is computed and does not account for the possibilities that the detonation wave may travel through other explosives with different detonation velocities or that the line of sight may pass outside of the explosive material.

If this control card is present, the lighting time is based on the shortest distance through the explosive material. If inert obstacles exist within the explosive material, the lighting time will account for the extra time required for the detonation wave to travel around the obstacles. The lighting times also automatically accounts for variations in the detonation velocity if different explosives are used. No additional input is required for this control option. This option works for two and three-dimensional solid elements. Also, see *INITIAL_DETONATION and *MAT_HIGH_EXPLOSIVE.

*CONTROL

*CONTROL_FORMING_POSITION

*CONTROL_FORMING_POSITION

Purpose: Provides a simple interface for stamping analysis. It is used to position the blank and tooling. When this keyword is used, all the tools must be in the home position, which is the position of the tooling at maximum stroke. From this position each tool will be moved based on this keyword. This keyword is used with the keywords *CONTROL_FORMING_USER and *CONTROL_FORMING_TRAVEL. One *CONTROL_FORMING_POSITION card may be needed for each part.

Card	1	2	3	4	5	6	7	8
Variable	PID	PREMOVE	TARGET					
Type	I	F	I					
Default	none	none	I					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID
PREMOV	The distance to pre-move the tool in the reverse direction of the movement of the tool.
TARGET	Move part (PID) in the reverse direction of this tool movement, and make sure the minimum distance between PID and TARGET is defined by GAP.

***CONTROL_FORMING_PROJECTION**

Purpose: To remove initial penetrations between the blank and the tooling by projecting the penetrated blank (slave) nodes along a normal direction to the surface of the blank with the specified gap between the node and the tooling surface.

Card 1 2 3 4 5 6 7 8

Variable	IDPS	IDPM	GAP	NRSST	NRMST			
Type	I	I	F	I	I			
Default								

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IDPS	Part ID for the blank (slave) side.
IDPM	Part ID for the tool (master) side.
GAP	A distance, which defines the minimum gap required.
NRSST	Normal direction of blank: EQ.0: the normal to the surface of the blank is pointing towards the tool, EQ.1: the normal to the surface of the blank is pointing away from the tool.
NRMST	Normal direction of tool: EQ.0: the normal to the surface of the tool is pointing towards the blank, EQ.1: the normal to the surface of the tool is pointing away from blank.

Remarks:

This option requires consistent normal vectors for both the rigid tooling surface and the blank surface.

*CONTROL

*CONTROL_FORMING_TEMPLATE

*CONTROL_FORMING_TEMPLATE

Purpose: This keyword is used to simplify the required input for sheet metal stamping simulations. With this keyword, five templates are given: three-piece air draw, three-piece toggle draw, four-piece draw, trimming, and springback.

Card 1 1 2 3 4 5 6 7 8

Variable	IDTEMP	BLKID	DIEID	PNCH	BNDU	BNDL	TYPE	PREBD
Type	I	I	I	I	I	I	I	F
Default	none	none	none	none	none	none	0	0.0
Remarks	1	2						

Card 2

Variable	LCSS	AL/FE	R00	R45	R90	E	DENSITY	PR
Type	I	C	F	F	F	F	F	F
Default	none	Fe	1.0	R00	R00	none	none	none
Remarks								

Card 3 1 2 3 4 5 6 7 8

Variable	K	N	MTYP	UNIT	THICK	GAP	FS	
Type	F	F	I	I	F	F	F	
Default	none	none	37	1	none	1.1t	0.1	
Remarks								

Card 4

Variable	PATERN	VMAX	VX	VY	VZ	VID	AMAX	
Type	I	F	F	F	F	I	F	
Default	1	1000	0	0	-1	none	1.0e+6	
Remarks								

Card 5

Variable	LVLADA	SIZEADA	TIMSADA	D3PLT				
Type	I	F	I	I				
Default	1	none	20	10				
Remarks								

VARIABLE	DESCRIPTION
IDTEMP	Type of forming process (See Remarks below.) EQ.1: 3-piece air-draw EQ.2: 3-piece Toggle-draw EQ.3: 4-piece draw EQ.4: Springback EQ.5: Trimming
BLKID	Part or part set ID (see TYPE) that defines the blank.
DIEID	Part or part set ID that defines the die. See Figures 8.3a, 8.3b and 8.3c for more information
PNCHID	Part or part set ID that defines the punch.
BNDUID	Part or part set ID that defines the upper binder.
BNDLID	Part or part set ID that defines the lower binder.
TYPE	Flag for part or part set ID used in the definition of BLKID, DIEID, PNCHID, BNDUID, and BNDLID. EQ.0: Part ID, EQ.1: Part set ID.
PREBD	Distance between the lower binder and punch in the 4 piece draw. See Figure 8.3c for more information.
LCSS	If the material for the blank is not user defined, this curve ID will define the stress-strain relationship; otherwise, this curve is ignored.
AL/FE	This parameter is used to define the Young's Modulus and density of the blank. If this parameter is defined, E and DENSITY will defined in the units given by UNIT below. EQ.A: the blank is aluminum EQ.F: the blank is steel (default)
R00, R45, R90	Material anisotropic parameters. For transverse anisotropy the R value is set to the average value of R00, R45, and R90.
E	Young's Modulus. If AL/FE is user defined, E is unnecessary
DENSITY	Material density of blank. If AL/FE is user defined, this parameter is unnecessary
PR	Poisson's ratio.
K	Strength coefficient for exponential hardening. If LCSS is defined, or if a blank material is user defined, this parameter is ignored. ($\bar{\sigma} = k\bar{\epsilon}^n$)

VARIABLE	DESCRIPTION
N	Exponent for exponential hardening. If LCSS is defined, or if a blank material user defined, this parameter is ignored.
MTYP	Material model types 36 and 37 are supported: *MAT_3-PARAMETER_BARAT and *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC, respectively.
UNIT	Units adopted in this simulation. Define a number between 1 and 10. See Table 8.1 to determine the value for UNIT. This unit is used to obtain proper punch velocity, acceleration, time step, and material properties.
THICK	Blank thickness. If the blank thickness is already defined, this parameter is ignored.
GAP	The home gap between rigid tools for automatic positioning and tooling motion. If *BOUNDARY_PRESCRIBED_RIGID_BODY is user defined, this parameter is ignored. The default is 1.1 x blank thickness.
FS	Friction coefficient (default=0.10). If the contact is user defined, this parameter is ignored.
PATERN	Velocity profile of moving tool. If the velocity is user defined by *BOUNDARY_PRESCRIBED_RIGID_BODY, PATERN is ignored. EQ.1: Ramped velocity profile EQ.2: Smooth velocity curve
VX, VY, VZ	Vector components defining the direction of the movement of the punch. The default direction is defined by VID
VID	VID is the vector ID defining the direction of the movement of the punch. The vector, VID, overrides the vector (VX, VY, VZ). If VID and (VX, VY, VZ) are undefined, the punch is assumed to move in the negative z-direction.
AMAX	The maximum allowable acceleration.
LVLADA	Maximum adaptive level.
SIZEADA	Minimum element size permitted in the adaptive mesh.
TIMSADA	Total number of adaptive steps during the forming simulation.
D3PLT	The total number of output states in the D3PLOT database.

UNIT	1	2	3	4	5	6	7	8	9	10
Mass	Ton	Gm	Gm	Gm	Gm	Kg	Kg	Kg	Kg	Kg
Length	Mm	Mm	Mm	Cm	Cm	Mm	Cm	Cm	Cm	m
Time	S	Ms	S	Us	S	Ms	Us	Ms	S	S
Force	N	N	1.e-6N	1e7N	Dyne	KN	1e10N	1e4N	1e-2N	N

Table 8.1 Available units for metal stamping simulations.

IDTEMP = 1: 3-Piece Air-Draw

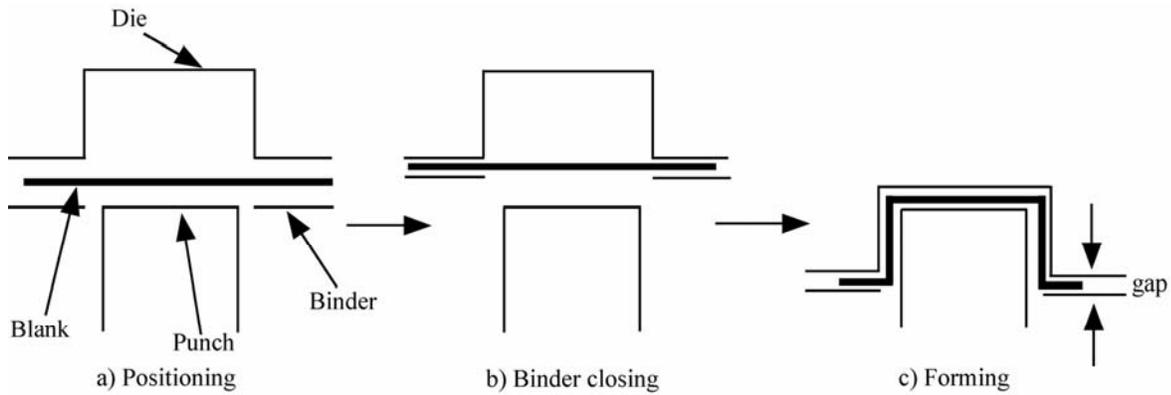


Figure 8.3a

IDTEMP = 2: 3-Piece Toggle Draw

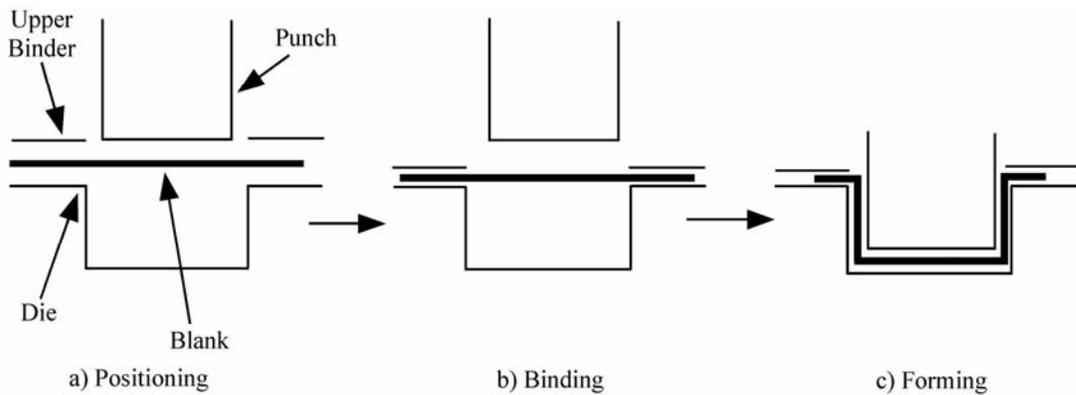
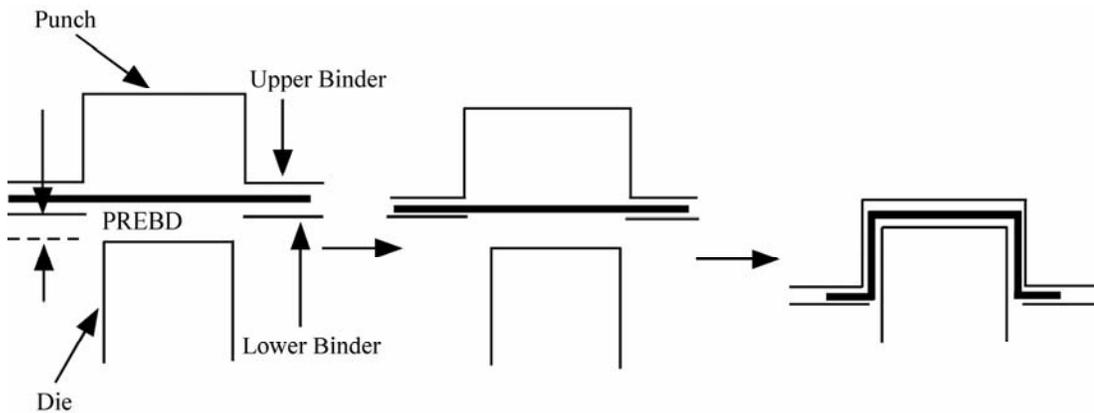


Figure 8.3b

- Notes:
1. In step a), LS-DYNA will automatically position the tools and minimize the punch travel.
 2. In steps b) and c), LS-DYNA will calculate the binder and punch travel based on the blank thickness and the home gap.
 3. Termination time will be automatically set based on b) and c).
 4. The rigid body motion of the tooling is automatically defined.
 5. All the contacts between the blank and rigid tools are defined.
 6. All necessary control parameters are defined.
 7. If user defines any of the parameter, it will override the automatic setting
 8. User does not need to use keywords, such as *PART, *CONTROL, *SECTION, *MAT_..., *CONTACT_...(Drawbead definition is an exception), *BOUNDARY_PRESCRIPTION_RIGID, etc.

IDTEMP=3: Four-Piece Draw**Figure 8.3c**

- Note:
- a) PREBD is used for the preposition of the lower
 - b) First, upper binder goes down to meet lower binder
 - c) Second, lower binder and upper binder move together and move to the home position
 - d) Third, punch move down to its home position

IDTEMP=4 Springback Simulation

Note: The necessary keywords, excluding this one, are *BOUNDARY to specify the constraints, and keywords include in dynain file. A new dynain file will be automatically output.

IDTEMP=5 Trimming operation

Note: The necessary keywords, excluding this one, are *DEFINE_CURVE_TRIM, and keywords included in dynain file.

*CONTROL

*CONTROL_FORMING_TRAVEL

*CONTROL_FORMING_TRAVEL

Purpose: Provide a simple interface for arbitrary stamping analysis. This keyword provides information on how to move the tools in each phase. This keyword should be used with *CONTROL_FORMING_POSITION and *CONTROL_FORMING_USER.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	VID	TRAVEL	TARGET	GAP	PHASE	FOLLOW	
Type	I	I	F	I	F	I	I	
Default	none	none	none	none	none	none	none	

VARIABLE

DESCRIPTION

PID	Part ID of tool.
VID	Vector ID defining the direction of travel for the tool defined by PID.
TRAVEL	Move tool this distance in the direction specified by the VID. If TRAVEL is defined, it is not necessary to define TARGET.
TARGET	Move tool (PID) to meet the TARGET, where the TARGET is the part ID of the matching tool.
GAP	The minimum distance between tool and TARGET in the home position. The GAP is by default the blank thickness.
PHASE	Phase number. Start sequentially from 1. For example, phase 1 is the binder closing, and phase 2 is the stamping operation.
FOLLOW	The tool (PID) can also move by following the part ID, FOLLOW. During this phase, the distance between the tool (PID) and part ID, FOLLOW, will be constant.

CONTROL_FORMING_USER**CONTROL*****CONTROL_FORMING_USER**

Purpose: To provide a simple interface for arbitrary stamping analysis. It provides blank material information. This keyword should be used with *CONTROL_FORMING_POSITION and *CONTROL_FORMING_TRAVEL.

Card 1 1 2 3 4 5 6 7 8

Variable	BLANK	TYPE	THICK	R00	R45	R90	AL/FE	UNIT
Type	I	I	F	F	F	F	A	I
Default	none	0	none	1.0	R00	R00	F	1
Remarks								

Card 2

Variable	LCSS	K	N	E	DENSITY	PR	FS	MTYPE
Type	I	F	F	F	F	F	F	I
Default	none	none	none	none	none	none	0.1	37
Remarks				1	1	1		

Card 3 1 2 3 4 5 6 7 8

Variable	PATERN	VMAX	AMAX	LVLADA	SIZEADA	ADATIMS	D3PLT	GAP
Type	I	F	F	I	F	I	I	F
Default	1	1000.0	500000.	0	0	0	10	1.1t
Remarks								

<u>VARIABLE</u>	<u>DESCRIPTION</u>
BLANK	Blank ID
TYPE	Flag for part or part set ID. EQ.0: Part ID, EQ.1: Part set ID.
THICK	Blank thickness. If the blank thickness is already defined, this parameter is ignored.
R00, R45, R90	Material anisotropic parameters. For transverse anisotropy the R value is set to the average value of R00, R45, and R90.
AL/FE	This parameter is used to define blank Young's Modulus and density. If this parameter is defined, E and Density will be found by using the proper unit, which is specified below. EQ.A: the blank is aluminum EQ.F: the blank is steel (default)
UNIT	Units adopted in this simulation. Define a number between 1 and 10. See Table 8.1 to determine the value for UNIT. This unit is used to obtain proper punch velocity, acceleration, time step, and material properties.
LCSS	If the material for the blank has not been defined, this curve will be used to define the stress-strain relation. Otherwise, this curve is ignored.
PREBD	Distance between the lower binder and punch in the 4 piece draw. See Figure 8.3c for more information.
K	Strength coefficient for exponential hardening. If LCSS is defined, or if a blank material is user defined, this parameter is ignored. ($\bar{\sigma} = k\bar{\epsilon}^n$)

VARIABLE	DESCRIPTION
N	Exponent for exponential hardening. If LCSS is defined, or if a blank material user defined, this parameter is ignored.
E	Young's Modulus. If AL/FE is user defined, E is unnecessary.
DENSITY	Material density of blank. If AL/FE is user defined, this parameter is unnecessary.
PR	Poisson's ratio. If AL/FE is user defined, E is unnecessary.
FS	Friction coefficient. If contact is defined, this parameter is ignored.
MTYP	Material model types 36 and 37 are supported: *MAT_3-PARAMETER_BARAT and *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC, respectively.
PATERN	Velocity profile of moving tool. If the velocity is user defined by *BOUNDARY_PRESCRIBED_RIGID_BODY, PATERN is ignored. EQ.1: Ramped velocity profile EQ.2: Smooth velocity curve
VMAX	The maximum allowable tool velocity
AMAX	The maximum allowable acceleration.
LVLADA	Maximum adaptive level.
SIZEADA	Minimum element size permitted in the adaptive mesh.
ADATIMS	Total number of adaptive steps during the forming simulation.
D3PLT	The total number of output states in the D3PLOT database.
GAP	Minimum gap between tools.

*CONTROL

*CONTROL_HOURLASS

*CONTROL_HOURLASS_{OPTION}

Available options include:

<BLANK>

936

which switches the hourglass formulation so that it is identical to that used in LS-DYNA version 936. The modification in the hourglass control from version 936 was to ensure that all components of the hourglass force vector are orthogonal to rigid body rotations. However, problems that run under version 936 sometimes lead to different results in versions 940 and later. This difference in results is primarily due to the modifications in the hourglass force vector. Versions released after 936 should be more accurate.

Purpose: Set the default values of the hourglass control to override the default values.

Card	1	2	3	4	5	6	7	8
Variable	IHQ	QH						
Type	I	F						
Default	1	0.1						
Remarks	1							

VARIABLE

DESCRIPTION

IHQ

Default hourglass viscosity type:

- EQ.1: standard LS-DYNA,
- EQ.2: Flanagan-Belytschko integration,
- EQ.3: Flanagan-Belytschko with exact volume integration,
- EQ.4: stiffness form of type 2 (Flanagan-Belytschko),
- EQ.5: stiffness form of type 3 (Flanagan-Belytschko),
- EQ.6: Belytschko-Bindeman [1993] assumed strain co-rotational stiffness form for 2D and 3D solid elements only. This form is available for explicit and IMPLICIT solution methods. In fact, type 6 is mandatory for the implicit options.
- EQ.8: Applicable to the type 16 fully integrated shell element. IHQ=8 activate warping stiffness for accurate solutions. A speed penalty of 25% is common for this option.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>EQ.9: Puso [2000] enhanced assumed strain stiffness form for 3D hexahedral elements. This form is available for explicit and implicit solution methods, hence it is an alternative to the Belytschko-Bindeman hourglass type 6 for implicit simulations.</p> <p>In the shell elements, $IHQ < 4$ is the viscous form based on Belytschko-Tsay. If $IHQ = 4, 5$ or 6, the stiffness form is obtained. The stiffness forms, however, can stiffen the response, especially if the deformations are large, and therefore should be used with care. For high velocities the viscous forms are recommended and for low velocities the stiffness forms are recommended. For large deformations and non-regular solids, option 3 or 5 is recommended.</p>
QH	<p>Default hourglass coefficient, QH. Values of QH that exceed .15 may cause instabilities. The recommended default applies to all options except for $IHQ=6$ (See remark 2). For hourglass type 9, see remark 3.</p>

Remarks:

1. Hourglass coefficients and type can be set by part ID in the *HOURLASS Section.
2. Type 6 hourglass control ($IHQ=6$) is for 2D and 3D solid elements only. Any underintegrated shell element parts that do not have hourglass type defined by *HOURLASS data will be automatically switched to type 4 hourglass control. If this behavior is not desired, it may be better to use *HOURLASS to change individual solid parts to type 6 hourglass control. For a more detailed discussion of type 6 hourglass control, please see Remark 4 in the *HOURLASS section.
3. Hourglass type 9 is available for hexahedral elements and is based on physical stabilization using an enhanced assumed strain method. In performance it is similar to the Belytschko-Bindeman hourglass formulation (type 6) but gives more accurate results for distorted meshes, e.g., for skewed elements. If $QH=1.0$, it produces accurate coarse bending results for elastic materials. The hourglass stiffness is by default based on elastic properties, hence the QH parameter should be reduced to about 0.1 for plastic materials in order not to stiffen the structure during plastic deformation. For materials 3, 18 and 24 there is the option to use a negative value of QH. With this option, the hourglass stiffness is based on the current material properties, i.e., the plastic tangent modulus, and scaled by $|QH|$.

*CONTROL

*CONTROL_IMPLICIT_AUTO

*CONTROL_IMPLICIT_AUTO

Purpose: Define parameters for automatic time step control during implicit analysis (see also *CONTROL_IMPLICIT_GENERAL).

Card	1	2	3	4	5	6	7	8
Variable	IAUTO	ITEOPT	ITEWIN	DTMIN	DTMAX	DTEXP		
Type	I	I	I	F	F	F		
Default	0	11	5	DT/1000.	DT*10.	none		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IAUTO	Automatic time step control flag EQ.0: constant time step size EQ.1: automatically adjusted time step size
ITEOPT	Optimum equilibrium iteration count per time step. See Figure 8.4.
ITEWIN	Defines range of allowable iteration window. If iteration count is within ITEWIN iterations of ITEOPT, step size will not be adjusted for the next step. In other words, the time step will be reduced if the iteration count to convergence is greater than ITEOPT+ITEWIN and the time step will be increased if the iteration count to convergence is less than ITEOPT-ITEWIN. See Figure 8.4.
DTMIN	Minimum allowable time step size. Simulation stops with error termination if time step falls below DTMIN.
DTMAX	Maximum allowable time step size. LT.0: curve ID = (-DTMAX) gives max step size as a function of time. Also, the step size is adjusted automatically so that the time value of each point in the curve is reached exactly (see Figures 8.5 and 8.6).
DTEXP	Time interval to run in explicit mode before returning to implicit mode. Applies only when automatic implicit-explicit switching is active (IMFLAG= 4 or 5 on *CONTROL_IMPLICIT_GENERAL).

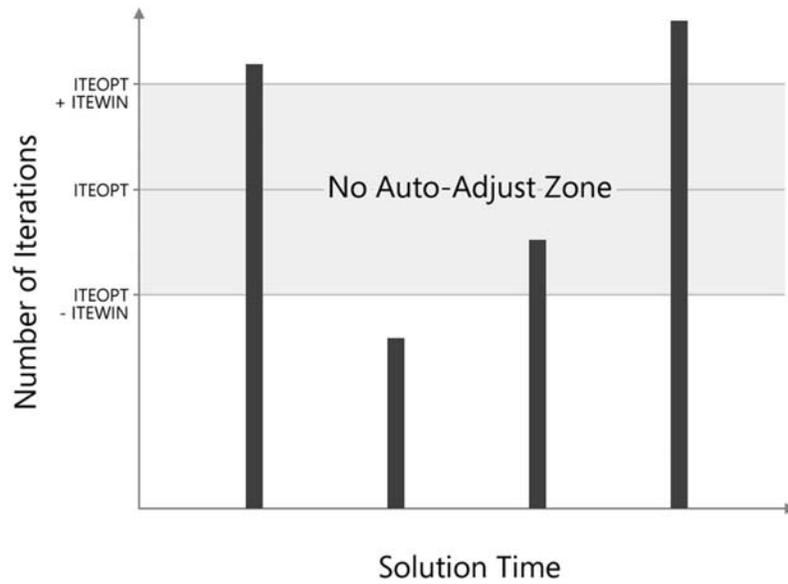


Figure 8.4. Iteration Window as defined by ITEOPT and ITEWIN.

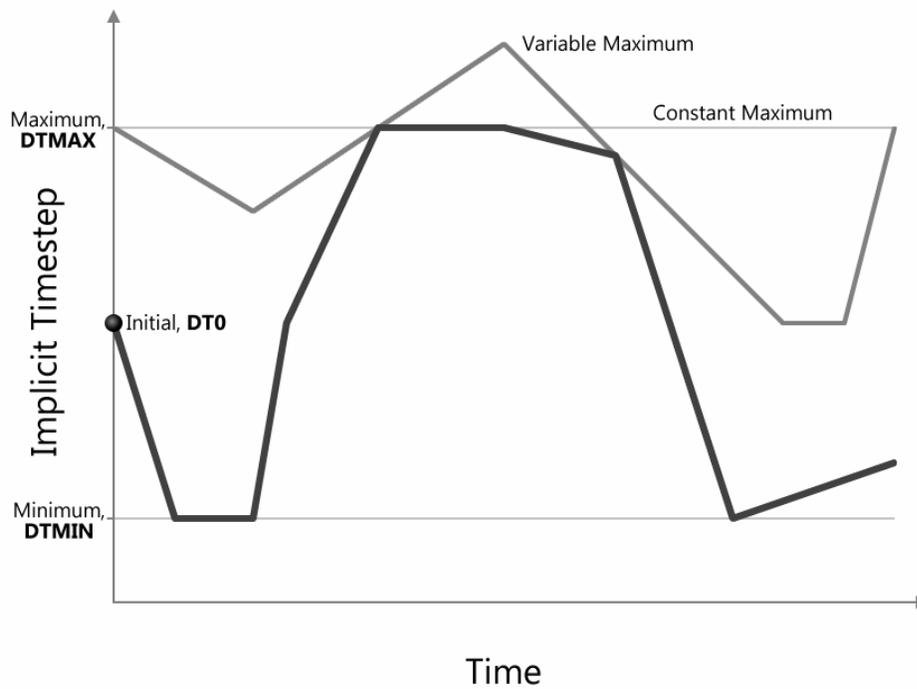


Figure 8.5. The implicit time step size changes continuously as a function of convergence within the bounds set by DTMIN and DTMAX.

***CONTROL_IMPLICIT_BUCKLE**

Purpose: Activate implicit buckling analysis when termination time is reached (see also *CONTROL_IMPLICIT_GENERAL).

Card 1 2 3 4 5 6 7 8

Variable	NMODE							
Type	I							
Default	0							

VARIABLE

DESCRIPTION

NMODE

Number of buckling modes to compute
EQ.0: none (DEFAULT)
EQ.n: compute n lowest buckling modes

Remarks:

Buckling analysis is performed at the end of a static implicit simulation. The simulation may be linear or nonlinear. After loads have been applied to the model, the buckling eigenproblem is solved:

$$[\mathbf{K}_M + \lambda\mathbf{K}_G]\{u\} = 0$$

where \mathbf{K}_M is the material tangent stiffness matrix, and the geometric or initial stress stiffness matrix \mathbf{K}_G is a function of internal stress in the model. The lowest n eigenvalues and eigenvectors are computed. The eigenvalues, written to text file "eigout", represent multipliers to the applied loads which give buckling loads. The eigenvectors, written to binary database "d3eigv", represent buckling mode shapes. View and animate these modes using LS-PrePost.

The geometric stiffness terms needed for buckling analysis will be automatically computed when the termination time is reached, regardless of the value of the geometric stiffness flag IGS on *CONTROL_IMPLICIT_GENERAL.

A double precision executable should be used for best accuracy in buckling analysis.

*CONTROL

*CONTROL_IMPLICIT_DYNAMICS

*CONTROL_IMPLICIT_DYNAMICS

Purpose: Activate implicit dynamic analysis and define time integration constants (see also *CONTROL_IMPLICIT_GENERAL).

Card 1 2 3 4 5 6 7 8

Variable	IMASS	GAMMA	BETA	TDYBIR	TDYDTH	TDYBUR	IRATE	
Type	I	F	F	F	F	F	I	
Default	0	.50	.25	0.0	1.E+28	1.E+28	0	

VARIABLE

DESCRIPTION

IMASS	Implicit analysis type EQ.0: static analysis EQ.1: dynamic analysis using Newmark time integration. EQ.2: dynamic analysis by modal superposition following the solution of the eigenvalue problem EQ.3: dynamic analysis by modal superposition using the eigenvalue solution in the d3eigv files that are in the runtime directory.
GAMMA	Newmark time integration constant (see remarks below.)
BETA	Newmark time integration constant.
TDYBIR	Birth time for application of dynamic terms. See Figure 8.7.
TDYDTH	Death time for application of dynamic terms.
TDYBUR	Burial time for application of dynamic terms.
IRATE	Rate effects switch: EQ.0: rate effects are on in constitutive models EQ.1: rate effects are off in constitutive models

Remarks:

For the dynamic problem, the linearized equilibrium equations may be written in the form

$$M\ddot{u}^{n+1} + D\dot{u}^{n+1} + K_t(x^n)\Delta u = P(x^n)^{n+1} - F(x^n)$$

where

M = lumped mass matrix

D = damping matrix

$u^{n+1} = x^{n+1} - x^0$ = nodal displacement vector

\dot{u}^{n+1} = nodal point velocities at time n+1

\ddot{u}^{n+1} = nodal point accelerations at time n+1.

Between the birth and death times 100% of the dynamic terms, that is the terms involving M and D , are applied. Between the death and burial time the dynamic terms are decreased linearly with respect to time until 0% of the dynamic terms are applied after the burial time. This feature is useful for problems that are initially singular because the parts are not in contact initially such as in metal stamping. For these problems dynamics is required for stable convergence. When contact is established the problem becomes well conditioned and the dynamic terms are no longer required for stable convergence. It is recommend that for such problems the user set the death time to be after contact is established and the burial time for 2 or 3 time steps after the death time.

The time integration is by the unconditionally stable, one-step, Newmark- β time integration scheme

$$\ddot{u}^{n+1} = \frac{\Delta u}{\beta \Delta t^2} - \frac{\dot{u}^n}{\beta \Delta t} - \frac{1}{\beta} \left(\frac{1}{2} - \beta \right) \ddot{u}^n$$

$$\dot{u}^{n+1} = \dot{u}^n + \Delta t (1 - \gamma) \ddot{u}^n + \gamma \Delta t \ddot{u}^{n+1}$$

$$x^{n+1} = x^n + \Delta u$$

Here, Δt is the time step size, and β and γ are the free parameters of integration. For $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$ the method reduces to the trapezoidal rule and is energy conserving. If

$$\gamma > \frac{1}{2}$$

$$\beta > \frac{1}{4} \left(\frac{1}{2} + \gamma \right)^2$$

numerical damping is induced into the solution leading to a loss of energy and momentum.

When modal superposition is invoked, NEIGV on *CONTROL_IMPLICIT_EIGENVALUE indicates the number of modes to be used. With modal superposition, stresses are computed only for linear shell formulation 18.

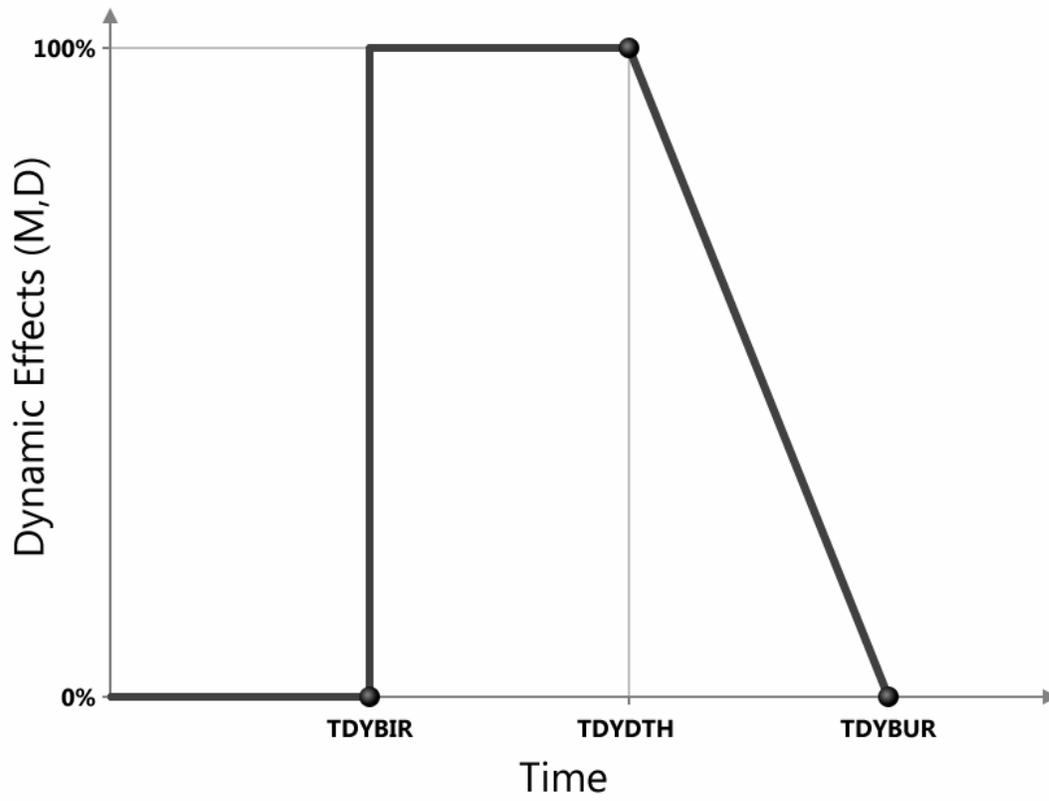


Figure 8.7. Birth, death, and burial time for implicit dynamics.

***CONTROL_IMPLICIT_EIGENVALUE**

Purpose: Activate implicit eigenvalue analysis and define associated input parameters (see also *CONTROL_IMPLICIT_GENERAL).

Card 1 2 3 4 5 6 7 8

Variable	NEIG	CENTER	LFLAG	LFTEND	RFLAG	RHTEND	EIGMTH	SHFSCL
Type	I	F	I	F	I	F	I	F
Default	0	0.0	0	-infinity	0	+infinity	2	0.0

Optional Card 2

Card 1 2 3 4 5 6 7 8

Variable	ISOLID	IBEAM	ISHELL	ITSHELL				
Type	I	I	I	I				
Default	0	0	0	0				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NEIG	Number of eigenvalues to extract. This must be specified. The other parameters below are optional. LT.0: curve ID = (-NEIG) used for intermittent eigenvalue analysis
CENTER	Center frequency. This option finds the nearest NEIG eigenvalues located about this value.
LFLAG	Left end point finite flag. EQ.0: left end point is -infinity EQ.1: left end point is LFTEND.
LFTEND	Left end point of interval. Only used when LFLAG = 1.
RFLAG	Right end point finite flag: EQ.0: right end point is +infinity EQ.1: right end point is RHTEND.

*CONTROL

*CONTROL_IMPLICIT_EIGENVALUE

<u>VARIABLE</u>	<u>DESCRIPTION</u>
RHTEND	Right end point of interval. Only used when RFLAG = 1.
EIGMTH	Eigenvalue extraction method: EQ.2: Block Shift and Invert Lanczos (default). EQ.3: Lanczos with [M] = [I] (for debug only). EQ.5: Same as 3 but include Dynamic Terms
SHFSCL	Shift scale. Generally not used, but see explanation below.
ISOLID	If nonzero, reset all solid element formulations to ISOLID for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
IBEAM	If nonzero, reset all beam element formulations to IBEAM for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
ISHELL	If nonzero, reset all shell element formulations to ISHELL for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
ITSHELL	If nonzero, reset all thick shell element formulations to ITSHELL for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.

Remarks:

To perform an eigenvalue analysis, activate the implicit method by selecting IMFLAG=1 on *CONTROL_IMPLICIT_GENERAL, and indicate a nonzero value for NEIG above. By default, the lowest NEIG eigenvalues will be found. If a nonzero center frequency is specified, the NEIG eigenvalues nearest to CENTER will be found.

When NEIG > 0, eigenvalues will be computed at time=0 and LS-DYNA will terminate.

When NEIG < 0, an intermittent eigenvalue analysis will be performed. This is a transient simulation during which loads are applied, with eigenvalues computed periodically during the simulation. Changes in geometry, stress, material, and contact conditions will affect the eigenvalues. The transient simulation can be either implicit or explicit according to IMFLAG=1 or IMFLAG=6, respectively, on *CONTROL_IMPLICIT_GENERAL. The curve ID = -NEIG indicates when to extract eigenvalues, and how many to extract. Define one curve point at each desired extraction time, with a function value equal to the number of eigenvalues desired at that time. A d3plot database will be produced for the transient solution results. Consecutively numbered d3eigv and eigout databases will be produced for each intermittent extraction. The extraction time is indicated in each database's analysis title.

The Block Shift and Invert Lanczos code is from BCSLIB-EXT, Boeing's Extreme Mathematical Library.

When using Block Shift and Invert Lanczos, the user can specify a semifinite or finite interval region in which to compute eigenvalues. Setting LFLAG = 1 changes the left end point from -infinity to the value specified by LFTEND. Setting RFLAG = 1 changes the right end point from +infinity to the values given by RHTEND. If the interval includes CENTER (default value of 0.0) then the problem is to compute the NEIG eigenvalues nearest to CENTER. If the interval does not include CENTER, the problem is to compute the smallest in magnitude NEIG eigenvalues.

If all of the eigenvalues are desired in an interval where both end points are finite just input a large number for NEIG. The software will automatically compute the number of eigenvalues in the interval and lower NEIG to that value. The most general problem specification is to compute NEIG eigenvalues nearest CENTER in the interval [LFTEND,RHTEND]. Computing the lowest NEIG eigenvalues is equivalent to computing the NEIG eigenvalues nearest 0.0.

For some problems it is useful to override the internal heuristic for picking a starting point for Lanczos shift strategy, that is the initial shift. In these rare cases, the user may specify the initial shift via the parameter SHFSCL. SHFSCL should be in the range of first few nonzero frequencies.

Eigenvectors are written to an auxiliary binary plot database named "d3eigv", which is automatically created. These can be viewed using a postprocessor in the same way as a standard "d3plot" database. The time value associated with each eigenvector plot is the corresponding circular frequency. A summary table of eigenvalue results is printed to the "eigout" file.

The print control parameter, LPRINT, and ordering method parameter, ORDER, from the *CONTROL_IMPLICIT_SOLVER keyword card also apply to the Block Shift and Invert Eigensolver.

*CONTROL

*CONTROL_IMPLICIT_GENERAL

*CONTROL_IMPLICIT_GENERAL

Purpose: Activate implicit analysis and define associated control parameters. This keyword is required for all implicit analyses.

Card	1	2	3	4	5	6	7	8
Variable	IMFLAG	DT0	IMFORM	NSBS	IGS	CNSTN	FORM	ZERO_V
Type	I	F	I	I	I	I	I	I
Default	0	none	2	1	2	0	0	0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IMFLAG	Implicit/Explicit analysis type flag EQ.0: explicit analysis EQ.1: implicit analysis EQ.2: explicit followed by implicit (activates “seamless” <i>springback</i>) EQ.4: implicit with automatic implicit-explicit switching EQ.5: implicit with automatic switching and mandatory implicit finish EQ.6: explicit with intermittent eigenvalue extraction EQ.-n: curve ID=n gives IMFLAG as a function of time.
DT0	Initial time step size for implicit analysis
IMFORM	Element formulation flag for “seamless” springback (IMFLAG=2 or *INTERFACE_SPRINGBACK_SEAMLESS) EQ.1: switch to fully integrated shell formulation for springback EQ.2: retain original element formulation (default)
NSBS	Number of implicit steps in “seamless” springback (IMFLAG=2 or *INTERFACE_SPRINGBACK_SEAMLESS)
IGS	Geometric (initial stress) stiffness flag EQ.1: include EQ.2: ignore
CNSTN	Indicator for consistent tangent stiffness (solid materials 3 & 115 only): EQ.0: do not use (default) EQ.1: use.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
FORM	Fully integrated element formulation (IMFLAG=2 and IMFORM=1 only) EQ.0: type 16 EQ.1: type 6.
ZERO_V	Zero out the velocity before switching from explicit to implicit. EQ.0: The velocities are not zeroed out. EQ.1: The velocities are set to zero.

Remarks:

IMFLAG The default value 0 indicates a standard explicit analysis will be performed. Using value 1 causes an entirely implicit analysis to be performed. Value 2 is automatically activated when the keyword *INTERFACE_SPRINGBACK_SEAMLESS is present, causing the analysis type to switch from explicit to implicit when the termination time is reached. Other nonzero values for IMFLAG can also be used with *INTERFACE_SPRINGBACK_SEAMLESS. After this switch, the termination time is extended by NSBS*DT0, or reset to twice its original value if DT0=0.0. The implicit simulation then proceeds until the new termination time is reached. Contact interfaces are automatically disabled during the implicit phase of “seamless” springback analysis.

When the automatic implicit-explicit switching option is activated (IMFLAG=4 or 5), the solution method will begin as implicit. If convergence of the equilibrium iterations fails, the solution will automatically switch to explicit for a time interval of DTEXP (see *CONTROL_IMPLICIT_AUTO). After this time interval, the solution method will switch back to implicit and attempt to proceed. The implicit simulation may be either static or dynamic. When this feature is used in a static implicit job, simulation time is no longer arbitrary, and must be chosen along with DTEXP in a realistic way to allow efficient execution of any explicit phases. Mass scaling may also be activated (see *CONTROL_TIMESTEP), and will apply only during the explicit phases of the calculation. In cases where much switching occurs, users must exercise caution to ensure that negligible dynamic effects are introduced by the explicit phases.

When IMFLAG=5, the final step of the simulation must be implicit. The termination time will be extended automatically as necessary, until a successfully converged implicit step can be obtained. This is useful for example in difficult metal forming springback simulations.

When IMFLAG=6, an explicit simulation will be performed. Eigenvalues will be extracted intermittently according to a curve indicated by NEIG=(-curve ID) on *CONTROL_IMPLICIT_EIGENVALUE. Beware that dynamic stress oscillations which may occur in the explicit simulation will influence the geometric (initial stress) stiffness terms used in the eigen solution, potentially producing misleading results and/or spurious modes. As an alternative, eigenvalues can also be extracted intermittently during an implicit analysis, using IMFLAG=1 and NEIG=(-curve ID).

When IMFLAG < 0, a curve ID is indicated which gives the solution method as a function of time. Define a curve value of zero during explicit phases, and a value of one during implicit phases. Use steeply sloping sections between phases. An arbitrary number of formulation switches may be activated with this method. See Figure 8.8.

- DT0 This parameter selects the initial time step size for the implicit phase of a simulation. The step size may change during a multiple step simulation if the automatic time step size control feature is active (see *CONTROL_IMPLICIT_AUTO.)

- IMFORM Adaptive mesh must be activated when using element formulation switching. For best springback accuracy, use of shell type 16 is recommended during the entire stamping and springback analysis, in spite of the increased cost of using this element during the explicit stamping phase.

- NSBS The NSBS option allows a “seamless” springback analysis to use multiple unloading steps (*CONTROL_IMPLICIT_STABILIZATION is also required in this case).

- IGS The geometric stiffness adds the effect of initial stress to the global stiffness matrix. This effect is seen in a piano string whose natural frequency changes with tension. Geometric stiffness does not always improve nonlinear convergence, especially when compressive stresses are present, so its inclusion is optional.

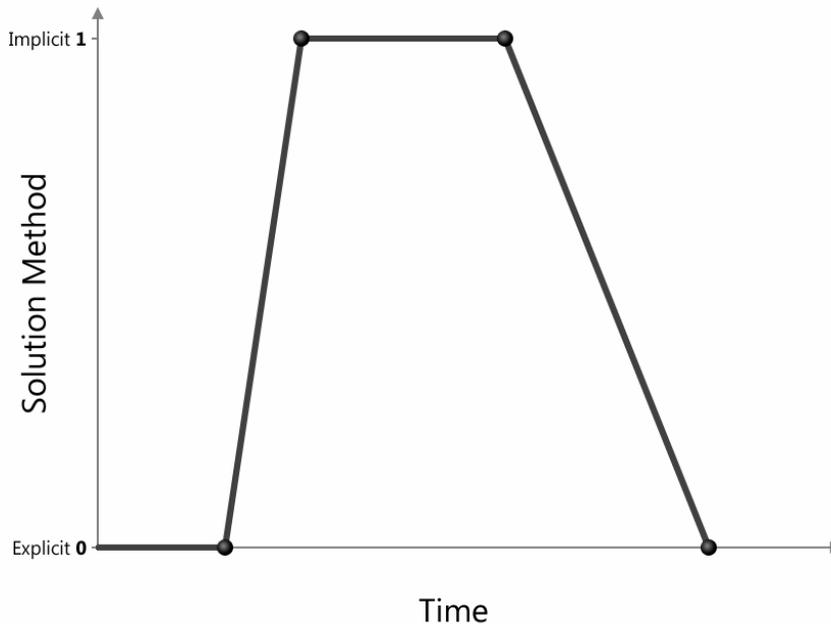


Figure 8.8. Solution method, implicit or explicit, controlled by a load curve.

***CONTROL_IMPLICIT_INERTIA_RELIEF**

***CONTROL**

***CONTROL_IMPLICIT_INERTIA_RELIEF**

Purpose: Allows analysis of linear static problems that have rigid body modes.

Card 1 2 3 4 5 6 7 8

Variable	IRFLAG	THRESH						
Type	I	F						
Default	0	0.001						

VARIABLE

DESCRIPTION

IRFLAG

Inertia relief flag

EQ.0: do not perform inertia relief

EQ.1: do perform inertia relief

THRESH

Threshold for what is a rigid body mode. The default is set to 0.001 Hertz where it is assumed that the units are in seconds.

*CONTROL

*CONTROL_IMPLICIT_JOINTS

*CONTROL_IMPLICIT_JOINTS

Purpose: Specify explicit or implicit treatment of joints for implicit analysis.

Card 1 2 3 4 5 6 7 8

Variable	ISPHER	IREVOL	ICYLIN					
Type	I	I	I					
Default	1	1	1					

VARIABLE

DESCRIPTION

ISPHER	Treatment of spherical joints EQ.1: use constraint method for all spherical joints (default) EQ.2: use penalty method for all spherical joints
IREVOL	Treatment of revolute joints EQ.1: use constraint method for all revolute joints (default) EQ.2: use penalty method for all revolute joints
ICYLIN	Treatment of cylindrical joints EQ.1: use constraint method for all cylindrical joints (default) EQ.2: use penalty method for all cylindrical joints

Remarks:

For most implicit applications one should use the constraint (default) method for the treatment of joints. When explicit-implicit switching is used the joint treatment should be consistent. This keyword allows the user to choose the appropriate treatment for their application.

***CONTROL_IMPLICIT_MODES**

Purpose: Request calculation of constraint and/or attachment modes for later use in modal analysis using *PART_MODES (see also *CONTROL_IMPLICIT_GENERAL).

Card 1 2 3 4 5 6 7 8

Variable	NSIDC	NSIDA						
Type	I	I						
Default	0	0						

VARIABLE

DESCRIPTION

NSIDC Node set ID for constraint modes
EQ.0: no constraint modes will be generated

NSIDA Node set ID for attachment modes
EQ.0: no attachment modes will be generated

Remarks:

To use this feature, an implicit analysis must be requested using IMFLAG=1 on *CONTROL_IMPLICIT_GENERAL, and a non-zero termination time must be specified on *CONTROL_TERMINATION. A double precision version of LS-DYNA should be used for best accuracy. Care must be taken to apply a sufficient number of constraints to the model to eliminate static rigid body motion. Computed modes are written to binary output file d3mode, which can be viewed using LS-PREPOST.

Constraint and attachment modes are generated by applying unit displacements and unit forces, respectively, to each specified degree of freedom. By default, modes are computed for all degrees of freedom for each node in sets NSIDC and NSIDA. The first and second node set attribute parameters can be optionally used to restrict the translational and rotational degrees of freedom for which modes are requested, respectively, according to the following syntax:

*CONTROL

*CONTROL_IMPLICIT_MODES

Node set attribute parameters DA1 and A1: translational degree of freedom codes

Node set attribute parameters DA2 and A2: rotational degree of freedom codes

<u>code</u>	<u>modes computed</u>
0	(See note below.)
1	X degree of freedom only
2	Y degree of freedom only
3	Z degree of freedom only
4	X, Y degrees of freedom only
5	Y, Z degrees of freedom only
6	X, Z degrees of freedom only
7	X, Y, Z degrees of freedom

Setting both node set attributes to zero is equivalent to setting both node set attributes to 7 (X, Y, and Z for translational and rotational degrees of freedom).

If one node set attribute is nonzero (codes 1 to 7) and the other node set attribute is zero, then the zero attribute means NO degrees of freedom are considered. For example, if DA1=2 and DA2=0, then only the Y-translational degree of freedom modes are calculated.

***CONTROL_IMPLICIT_SOLUTION**

Purpose: These optional cards apply to implicit calculations. Use these cards to specify whether a linear or nonlinear solution is desired. Parameters are also available to control the implicit nonlinear and arc length solution methods (see also *CONTROL_IMPLICIT_GENERAL).

Card 1 Format

Card 1 2 3 4 5 6 7 8

Variable	NSOLVR	ILIMIT	MAXREF	DCTOL	ECTOL	RCTOL	LSTOL	ABSTOL
Type	I	I	I	F	F	F	F	F
Default	2	11	15	0.001	0.01	1.0e+10	0.90	1.e-10

Optional Card 2

Card 1 2 3 4 5 6 7 8

Variable	DNORM	DIVERG	ISTIF	NLPRINT	NLNORM	D3ITCTL		
Type	I	I	I	I	I	I		
Default	2	1	1	0	2	0		

Optional Card 3 (if card 3 is used, then card 2 above must also be used)

Card 1 2 3 4 5 6 7 8

Variable	ARCCTL	ARCDIR	ARCLEN	ARCMTH	ARCDMP			
Type	I	I	F	I	I			
Default	0	none	0	1	2			

*CONTROL

*CONTROL_IMPLICIT_SOLUTION

Optional Card 4 (if card 4 is used, then cards 2 and 3 above must also be used)

Card	1	2	3	4	5	6	7	8
Variable	LSMTD	LSDIR	IRAD	SRAD	AWGT	SRED		
Type	I	I	F	F	F	F		
Default	1	2	0.0	0.0	0.0	0.0		

(See remarks below)

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NSOLVR	Solution method for implicit analysis: EQ.1: Linear EQ.2: Nonlinear with BFGS updates (default) EQ.3: Nonlinear with Broyden updates EQ.4: Nonlinear with DFP updates EQ.5: Nonlinear with Davidon updates EQ.6: Nonlinear with BFGS updates + arclength EQ.7: Nonlinear with Broyden updates + arclength EQ.8: Nonlinear with DFP updates + arclength EQ.9: Nonlinear with Davidon updates + arclength
ILIMIT	Iteration limit between automatic stiffness reformations
MAXREF	Stiffness reformation limit per time step
DCTOL	Displacement relative convergence tolerance
ECTOL	Energy relative convergence tolerance
RCTOL	Residual (force) relative convergence tolerance (DEFAULT=inactive)
LSTOL	Line search convergence tolerance
ABSTOL	Absolute convergence tolerance.
DNORM	Displacement norm for convergence test EQ.1: Increment vs. displacement over current step EQ.2: Increment vs. total displacement (default)
DIVERG	Divergence flag (force imbalance increase during equilibrium iterations) EQ.1: reform stiffness if divergence detected (default) EQ.2: ignore divergence

VARIABLE	DESCRIPTION
ISTIF	Initial stiffness formation flag EQ.1: reform stiffness at start of each step (default) EQ.n: reform stiffness at start of every "n"th step
NLPRINT	Nonlinear solver print flag EQ.0: no nonlinear iteration information printed (new v970 default) EQ.1: print iteration information to screen, message, d3hsp files EQ.2: print extra norm information (NLNORM=1) NOTE: during execution, interactive commands can be used: <u>interactive command</u> <u>response</u> <ctrl-c> nlprint toggle NLPRINT between 0 and 1 <ctrl-c> diagnostic toggle NLPRINT between 0 and 2 <ctrl-c> information set NLPRINT=2 for one iteration
NLNORM	Nonlinear convergence norm type EQ.1: consider translational and rotational degrees of freedom EQ.2: consider translational degrees of freedom only (default)
D3ITCTL	Control D3ITER database. If nonzero, the search directions for the nonlinear implicit solution are written to the D3ITER database. To reduce the size of the D3ITER database the database is reset every n time steps where n=D3ITCTL.
<i>The following 5 parameters are for use with arc length methods only ($6 \leq \text{NSOLVR} \leq 9$):</i>	
ARCCTL	Arc length controlling node ID EQ.0: generalized arc length method
ARCDIR	Arc length controlling node direction (ignored if ARCCTL=0 above) EQ.1: global X-translation EQ.2: global Y-translation EQ.3: global Z-translation
ARCLEN	Relative arc length size. See remarks below. LE.0.0: use automatic size, GT.0.0: use ARCLEN*automatic step size.
ARCMTH	Arc length method EQ.1: Crisfield (default) EQ.2: Ramm
ARCDMP	Arc length damping option EQ.2: off (default) EQ.1: on, oscillations in static solution are suppressed

VARIABLE	DESCRIPTION
LSMTD	Line search convergence method: EQ.1: Energy method using only translational variables (default) EQ.2: Residual method EQ.3: Energy method using both translational and rotational variables
LSDIR	Line search direction method: EQ.1: Search on all variables (traditional approach used in versions prior to 971) EQ.2: Search only on the independent (unconstrained) variables EQ.3: Use adaptive line search (see AWGT, SRED) EQ.4: Use curved line search (see IRAD, SRAD)
IRAD	Normalized curvature factor for curved line search, where 0 indicates a straight line search and 1 indicates full curved line search.
SRAD	Radius of influence for determining curve in curved line search. For each independent node, all nodes within this radius are used for determining the curve. If 0, then all nodes connected to the same element as the independent node are used.
AWGT	Adaptive line search weight factor between 0 and 1. A high value tends to restrict the motion of oscillating nodes during the implicit process.
SRED	Initial step reduction between 0 and 1 for adaptive line search, use large number for conservative start in implicit procedure.

Remarks:

NSOLVR If a linear analysis is selected, equilibrium checking and iterations are not performed.

The Full Newton nonlinear solution method can be invoked by using the default BFGS solver, and selecting ILIMIT=1 to form a new stiffness matrix every iteration.

In the neighborhood of limit points the Newton based iteration schemes often fail. The arc length method of Riks and Wempner (combined here with the BFGS method) adds a constraint equation to limit the load step to a constant "arc length" in load-displacement space. This method is frequently used to solve snap through buckling problems. When applying the arc-length method, the curves that define the loading should contain only two points, and the first point should be at the origin (0,0). LS-DYNA will extrapolate, if necessary, to determine the load. In this way, time and load magnitude are related by a constant. It is possible that time can become negative in case of load reversal. The arc length method cannot be used in a dynamic analysis.

ILIMIT	In the default BFGS method, the global stiffness matrix is only reformed every ILIMIT iterations. Otherwise, an inexpensive stiffness update is applied. By setting ILIMIT=1, a stiffness reformation is performed every iteration. This is equivalent to the Full Newton method (with line search). A higher value of ILIMIT (20-25) can reduce the number of stiffness matrix reformations and factorizations which may lead to a significant reduction in cost. Note that the storage requirements for implicit include storing 2 vectors per iteration. Large values of ILIMIT will cause substantial increase in storage requirements.
MAXREF	The nonlinear equilibrium search will continue until the stiffness matrix has been reformed MAXREF times, with ILIMIT iterations between each reformation. If equilibrium has not been found, control will be passed to the automatic time step controller if it is activated. Otherwise, error termination will result. When the auto time step controller is active, it is often efficient to choose MAXREF=5 and try another stepsize quickly, rather than wasting too many iterations on a difficult step.
DCTOL	When the displacement norm ratio is reduced below DCTOL, this condition is satisfied. Smaller numbers lead to more accurate determination of equilibrium and, on the negative side, result in more iterations and higher costs. Use NLPRINT to display norm data each iteration.
ECTOL	When the energy norm ratio is reduced below ECTOL, this condition is satisfied. Smaller numbers lead to more strict determination of equilibrium and, on the negative side, result in more iterations and higher costs. Use NLPRINT to display norm data each iteration.
RCTOL	When the residual norm ratio is reduced below RCTOL, this condition is satisfied. Smaller numbers lead to more strict determination of equilibrium and, on the negative side, result in more iterations and higher costs. By default this convergence criterion is effectively disabled using RCTOL=1.e10. Use NLPRINT to display norm data each iteration.
LSTOL	A line search is performed on stiffening systems to guard against divergence of Newton-based nonlinear solvers. With the Full Newton method, it is sometimes helpful to define a large value (LSTOL=9999.0) to effectively disable line search.
DNORM	When computing the displacement ratio, the norm of the incremental displacement vector is divided by the norm of "total" displacement. This "total" displacement may be either the total over the current step, or the total over the entire simulation. The latter tends to be more lax, and can be poor at the end of simulations where large motions develop. For these problems, an effective combination is DNORM=1, and DCTOL=0.01 or larger.
DIVERG	By default, a new stiffness matrix is formed whenever divergence (growing out-of-balance force) is detected. This flag can be used to suppress this stiffness reformation.

ISTIF	By default, a new stiffness matrix is formed at the start of every time step. Suppressing this stiffness reformation can decrease the cost of simulations which have many tiny steps that are mostly linear, such as transient dynamics.
NLPRINT	This flag controls printing of displacement and energy convergence measures during the nonlinear equilibrium search. If convergence difficulty occurs, this information is helpful in determining the problem.
NLNORM	By default, only translational degrees of freedom are used in evaluating convergence norms. Use this flag to include rotational degrees of freedom, or to make additional data available for diagnosing convergence problems. This additional data includes the worst offending node and degree of freedom contributing to each norm.
ARCCTL	The arc length method can be controlled based on the displacement of a single node in the model. For example, in dome reversal problems the node at the center of the dome can be used. By default, the generalized arc length method is used, where the norm of the global displacement vector controls the solution. This includes all nodes.
ARCLLEN	In many cases the arc length method has difficulty tracking the load displacement curve through critical regions. Using $0 < \text{ARCLLEN} < 1$ will reduce the step size to assist tracking the load-displacement curve with more accuracy. Use of $\text{ARCLLEN} < 1$ will cause more steps to be taken. Suggested values are 1.0 (the default), 0.5, 0.25, and 0.10.
ARCDMP	Some static problems exhibit oscillatory response near instability points. This option numerically suppresses these oscillations, and may improve the convergence behavior of the post-buckling solution.
LSMTD	The default method for determining convergence of the nonlinear line search is to find the minimum of the energy. This parameter allows choosing the energy on only the translational variables, energy of both the translational and rotational variables, or for minimizing the residual (forces). The effect of using a residual based line search is not always positive, sometimes it is too restrictive and stops convergence. However, it is a more conservative approach than using the energy based method since it explicitly controls the norm of the residual. It should not be seen as a better strategy than the energy method but as an alternative to try in cases when the default method seems to be working poorly.
LSDIR	In Version 971 of LS-DYNA new line search options were added. The traditional approach ($\text{LSDIR}=1$) computes the line search direction using all variables. The new (default) approach of $\text{LSIDR}=2$ computes the line search direction only on the unconstrained variables. It has proven to be both robust and more efficient. We have also included two new approaches to try for problems where the default and traditional approach fail and the user is using Full Newton ($\text{ILIMIT}=1$). See the next two remarks for more information on those methods.

- IRAD, SRAD The parameters IRAD and SRAD are for the curved line search (LSDIR=4). The first parameter is a switch (0 or 1) to invoke this line search, an intermediate value is interpreted as weighted combination of a straight and curved line search (the curvature radius is decreased with increasing IRAD). A value of unit is recommended in situations with rather smooth responses, e.g. springback and similar problems. Also, IRAD=1 seems to work best with full Newton iterations. The SRAD parameter should be equal to 0 for most cases, this means that the search curve for a node is determined from the search direction of nodes connected to the same elements as that node. SRAD>0 is interpreted as a radius of influence, meaning that the search curve for a node is determined from the search direction of nodes within a distance SRAD of this node. This option was introduced as an experiment to see if this had a smoothing and stabilizing effect. A value of 0.0 is currently recommended.
- AWGT, SRED The parameters AWGT and SRED are for the adaptive line search. The intention is to improve robustness for problems that have tendencies to oscillate or diverge, indicated by the dnorm and enorm parameter outputs in the iterations (stdout). A value of 0.5 is recommended for AWGT as a starting point. With a nonzero value the motions of individual nodes are tracked. For nodes that are oscillating (going back and forth in space), the maximum step size for the next iteration is reduced in proportion to the parameter AWGT, and for nodes that are not oscillating but going nicely along a straight path, the maximum step size for the next iteration is increased in proportion to 1-AWGT. In test problems, the introduction of the adaptive line search has stabilized the implicit procedure in the sense that the dnorm and enorm values are more monotonically decreasing until convergence with virtually no oscillations. If a problem is still oscillating or diverging, the user should try to increase the AWGT parameter since this is a more restrictive approach but probably gives a slower convergence rate. An option for nasty problems is also to use SRED>0 which is the initial step reduction factor (less than 1). This means that the initial step size is reduced by this value but the maximum step size will increase by an amount that is determined by the success in the iterative procedure, eventually it will reach unity. It can never decrease. Also here, it is intended to be used with full Newton method.

*CONTROL

*CONTROL_IMPLICIT_SOLVER

*CONTROL_IMPLICIT_SOLVER

Purpose: These optional cards apply to implicit calculations. The linear equation solver performs the CPU-intensive stiffness matrix inversion (see also *CONTROL_IMPLICIT_GENERAL).

Card	1	2	3	4	5	6	7	8
Variable	LSOLVR	LPRINT	NEGEV	ORDER	DRCM	DRCPRM	AUTOSPC	AUTOTOL
Type	I	I	I	I	I	F	I	F
Default	4	0	2	0	1	see below	1	see below

Optional Card 2

Card	1	2	3	4	5	6	7	8
Variable	LCPACK	MTXDMP						
Type	I	I						
Default	2	0						

(See remarks below)

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LSOLVR	Linear equation solver method EQ.4: SMP parallel multi-frontal sparse solver (default). EQ.5: SMP parallel multi-frontal sparse solver, double precision EQ.6: BCSLIB-EXT, direct, sparse, double precision EQ.10: iterative, best of currently available iterative methods EQ.11: iterative, Conjugate Gradient method EQ.12: iterative, CG with Jacobi preconditioner EQ.13: iterative, CG with Incomplete Choleski preconditioner EQ.14: iterative, Lanczos method EQ.15: iterative, Lanczos with Jacobi preconditioner EQ.16: iterative, Lanczos with Incomplete Choleski preconditioner

VARIABLE	DESCRIPTION
LPRINT	Linear solver print flag controls screen and message file output EQ.0: no printing EQ.1: output summary statistics on memory, cpu requirements EQ.2: more statistics EQ.3: even more statistics and debug checking (NOTE: during execution, use the interactive command "<ctrl-c> lprint" to toggle this print flag between 0 and 1.)
NEGEV	Negative eigenvalue flag. Selects procedure when negative eigenvalues are detected during stiffness matrix inversion. EQ.1: stop, or retry step if auto step control is active EQ.2: print warning message, try to continue (default)
ORDER	Ordering option EQ.0: method set automatically by LS-DYNA EQ.1: MMD, Multiple Minimum Degree. EQ.2: Metis
DRCM	Drilling rotation constraint method: EQ.1: add stiffness (old default in Version 970) EQ.2: add no stiffness EQ.3: add no stiffness EQ.4: add stiffness (new default in Version 971 Release 2)
DRCPRM	Drilling rotation constraint parameter, DRCPRM. If adding stiffness, DRCM=1, then, for linear problems, DRCPRM=1.0; for nonlinear problems, DRCPRM=100.0; and for eigenvalue problems either 1.E-12 or 1.E-8 is used depending on the shell element type. In the latter case, the input value for DRCPRM is ignored.
AUTOSPC	Automatic Constraint Scan flag EQ.1: scan the assembled stiffness matrix looking for unconstrained, unattached degrees of freedom. Generate additional constraints as necessary to avoid negative eigenvalues. EQ.2: do not add constraints.
AUTOTOL	AUTOSPC tolerance. The test for singularity is the ratio of the smallest singular value and the largest singular value. If this ratio is less than AUTOTOL, then the triple of columns are declared singular and a constraint is generated. Default value in single precision is 1.E-4 and in double precision, 1.E-8.
LCPACK	Matrix assembly package. EQ.2: Use v970's LCPACK (default, only available option in 971)

<u>VARIABLE</u>	<u>DESCRIPTION</u>
MTXDMP	Matrix and right-hand-side dumping. To assist in evaluating other linear equation solution package, LS-DYNA has the option of dumping the globally assembled stiffness matrix and right-hand-side vectors to ASCII files in Harwell-Boeing sparse matrix format. EQ.0: No dumping EQ.1: Dump all matrices and right-hand-side vectors every MTXDMP time steps. Filenames of the form K.matrix.xxxx, M.matrix.xxxx, and RHS.vector.xxxx are used.

Remarks:

LSOLVR The linear solver is used to compute the inverse of the global stiffness matrix, which is a costly procedure both in memory and cpu time. Direct solvers apply Gaussian elimination, while iterative solvers successively improve “guesses” at the correct solution. Iterative solvers require far less memory than direct solvers, but may suffer from convergence problems. Generally, iterative solvers are poor for automotive applications, but can be superior for large brick element soil models in civil engineering.

Solvers 5 and 6 promote the global matrix to double precision before factoring to reduce numerical truncation error. Solvers 4 and 5 are equivalent if a double precision executable is used.

Solver 6 is the direct linear equation solver from BCSLIB-EXT, Boeing's Extreme Mathematical Library. This option should be used whenever the factorization is too large to fit into memory. It has extensive capabilities for out-of-core solution and can solve larger problems than any of the other direct factorization methods. Solver 6 also includes a sophisticated pivoting strategy which can be superior for nearly singular matrices.

LPRINT Select printing of the timing and storage information (LPRINT = 1) if you are comparing performance of linear equation solvers, or if you are running out of memory for large models. Minimum memory requirements for in-core and out-of-core solution are printed. This flag can also be toggled using sense switch "<ctrl-c> lprint". *For best performance, increase available memory using “memory=” on the command line until an IN-CORE solution is indicated.*

When using solver option 6, LPRINT = 2 and 3 will cause increased printed output of statistics and performance information.

NEGEV Negative eigenvalues result from underconstrained models (rigid body modes), severely deformed elements, or non-physical material properties. This flag allows control to be passed directly to the automatic time step controller when negative eigenvalues are detected. Otherwise, significant numerical roundoff error is likely to occur during factorization, and equilibrium iterations may fail (see *CONTROL_IMPLICIT_AUTO).

ORDER The system of linear equations is reordered to optimize the sparsity of the factorization when using direct methods. Metis is a ordering method from University of Minnesota which is very effective for larger problems and for 3D solid problems, but also very expensive. MMD is inexpensive, but may not produce an optimum reordering, leading to higher cost during numeric factorization. MMD is usually best for smaller problems (less than 100,000 degrees of freedom).

Reordering cost is included in the symbolic factorization phase of the linear solver (LPRINT.ge.1). For large models, if this cost exceeds 20% of the numeric factorization cost, it may be more efficient to select the MMD method.

Note that the values of LPRINT and ORDER also affect the eigensolution software. That is LPRINT and ORDER from this keyword card is applicable to eigensolution.

*CONTROL

*CONTROL_IMPLICIT_STABILIZATION

*CONTROL_IMPLICIT_STABILIZATION

Purpose: This optional card applies to implicit calculations. Artificial stabilization is required for multi-step unloading in implicit springback analysis (see also *CONTROL_IMPLICIT_GENERAL).

Card	1	2	3	4	5	6	7	8
Variable	IAS	SCALE	TSTART	TEND				
Type	I	F	F	F				
Default	2	1.0	(see below)	(see below)				

VARIABLE	DESCRIPTION
IAS	Artificial Stabilization flag EQ.1: active EQ.2: inactive (default)
SCALE	Scale factor for artificial stabilization. For flexible parts with large springback, like outer body panels, a value of 0.001 may be required. EQ.-n: curve ID = n gives SCALE as a function of time
TSTART	Start time. (Default: immediately upon entering implicit mode)
TEND	End time. (Default: termination time)

Remarks:

Artificial stabilization allows springback to occur over several steps. This is often necessary to obtain convergence during equilibrium iterations on problems with large springback deformation. Stabilization is introduced at the start time TSTART, and slowly removed as the end time TEND is approached. Intermediate results are not accurate representations of the fully unloaded state. The end time TEND must be reached exactly for total springback to be predicted accurately.

IAS The default for IAS depends on the analysis type in *CONTROL_IMPLICIT_GENERAL. For “seamless” springback analysis, automatic time step control and artificial stabilization are activated by default. Otherwise, IAS is inactive by default.

SCALE This is a penalty scale factor similar to that used in contact interfaces. If modified, it should be changed in order-of-magnitude increments at first. Large values suppress springback deformation until very near the termination time, making convergence during the first few steps easy. Small values may not stabilize the solution enough to allow equilibrium iterations to converge.

*CONTROL

*CONTROL_IMPLICIT_TERMINATION

*CONTROL_IMPLICIT_TERMINATION

Purpose: Specify termination criteria for implicit transient simulations.

Card 1 2 3 4 5 6 7 8

Variable	DELTAU							
Type	F							
Default	0.0							

VARIABLE

DESCRIPTION

DELTAU

Alternate termination criteria for implicit transient simulation.
EQ.0.0: terminate based on ENDTIM (default)
NE.0.0: terminate when displacement for last time step relative to the total displacement is less than DELTAU.

Remarks:

For some implicit applications it is useful to terminate when there is no change in displacement. This keyword provides the ability to specify such a stopping criteria to terminate the simulation prior to ENDTIM.

***CONTROL_MPP_DECOMPOSITION_AUTOMATIC**

Purpose: Instructs the program to apply a simple heuristic to try to determine the proper decomposition for the simulation.

There are no input parameters. The existence of this keyword triggers the automated decomposition. This option should not be used if there is more than one occurrence of any of the following options in the model:

- *INITIAL_VELOCITY
- *CHANGE_VELOCITY
- *BOUNDARY_PRESCRIBED_MOTION

And the following control card must not be used:

- *CONTROL_MPP_DECOMPOSITION_TRANSFORMATION

For the general case, it is recommended that you specify the proper decomposition using the command *CONTROL_MPP_DECOMPOSITION_TRANSFORMATION instead.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_CHECK_SPEED**

***CONTROL_MPP_DECOMPOSITION_CHECK_SPEED**

Purpose: Modifies the decomposition depending on the relative speed of the processors involved.

There are no input parameters. The existence of this keyword causes a short floating point timing routine to be executed on each processor. The information gathered is used during the decomposition, with faster processors being given a relatively larger portion of the problem. This option is not recommended on homogeneous systems.

***CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE *CONTROL**

***CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE**

Purpose: Ensures that the indicated contact interfaces are distributed across all processors, which can lead to better load balance for large contact interfaces.

Card 1 2 3 4 5 6 7 8

Variable	ID1	ID2	ID3	ID4	ID5			
Type	I	I	I	I	I			
Default	None	None	None	None	None			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID1	First contact interface ID to distribute. If no contact ID's are specified, the number given here corresponds to the order of the interfaces as they appear in the input, with the first being 1.
ID2, ID3, ID4, ID5	Remaining interfaces ID's to distribute.

Remarks:

Up to 5 contact interface ID's can be specified. The decomposition is modified as follows: First, all the elements involved in the first contact interface are decomposed across all the processors. Then all the elements involved in the second contact interface (excluding any already assigned to processors) are distributed, and so on. After all the contact interfaces given are processed, the rest of the input is decomposed in the normal manner. This will result in each processor having possibly several disjoint portions of the input assigned to it, which will increase communications somewhat. However, this can be offset by improved load balance in the contact. It is generally recommended that at most one or two interfaces be specified, and then only if they are of substantial size relative to the whole problem.

***CONTROL *CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE**

***CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE**

Purpose: Ensures that the indicated contact interfaces are isolated on a single processor, which can lead to decreased communication.

Card	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5			
Type	I	I	I	I	I			
Default	None	None	None	None	None			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID1	First contact interface ID to isolate. If no contact ID's are specified, the number given here corresponds to the order of the interfaces as they appear in the input, with the first being 1.
ID2, ID3, ID4, ID5	Remaining interfaces ID's to isolate.

Remarks:

Up to 5 contact interfaces can be specified. The decomposition is modified as follows: First, all the elements involved in the first contact interface ID are assigned to the first processor. Then all the elements involved in the second contact interface ID (excluding any already assigned to processors) are assigned to the next processor, and so on. After all the contact interfaces given are processed, the rest of the input is decomposed in the normal manner. This will result in each of the interfaces being processed on a single processor. For small contact interfaces this can result in better parallelism and decreased communication.

***CONTROL_MPP_DECOMPOSITION_FILE**

Purpose: Allow for pre-decomposition and a subsequent run or runs without having to do the decomposition.

Card 1 2 3 4 5 6 7 8

Variable	Name							
Type	A80							
Default	None							

VARIABLE

DESCRIPTION

NAME

Name of a file containing (or to contain) a decomposition record.

Remarks:

If the indicated file does not exist, it is created with a copy of the decomposition information from this run. If the file exists, it is read and the decomposition steps can be skipped. The original run that created the file must be for a number of processors that is a multiple of the number of processors currently being used. Thus, a problem can be decomposed once for, say, 48 processors. Subsequent runs are then possible on any number that divides 48: 1, 2, 3, 4, 6, etc. Since the decomposition phase generally requires more memory than execution, this allows large models to be decomposed on one system and run on another (provided the systems have compatible binary formats). The file extension “.pre” is added automatically.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_METHOD**

***CONTROL_MPP_DECOMPOSITION_METHOD**

Purpose: Specify the decomposition method to use.

Card 1 2 3 4 5 6 7 8

Variable	Name							
Type	A80							
Default	RCB							

VARIABLE

DESCRIPTION

NAME

Name of the decomposition method to use. There are currently two options:

RCB = recursive coordinate bisection

GREEDY = a simple heuristic method

In almost all cases the RCB method is superior and should be used.

***CONTROL_MPP_DECOMPOSITION_NUMPROC**

Purpose: Specify the number of processors for decomposition.

Card 1 2 3 4 5 6 7 8

Variable	N								
Type	I								
Default	None								

VARIABLE

DESCRIPTION

N

Number of processors for decomposition.

Remarks:

This is used in conjunction with the CONTROL_MPP_DECOMPOSITION_FILE command to allow for later runs on different numbers of processors. By default, the decomposition is performed for the number of processors currently being used. However, a different value can be specified here. If N > 1 and only one processor is currently being used, the decomposition is done and then the program terminates. Similarly, if N is NOT a multiple of the current number of processors, the execution terminates after decomposition. Otherwise, the decomposition is performed for N processors, and the execution continues.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_SHOW**

***CONTROL_MPP_DECOMPOSITION_SHOW**

Purpose: Allows display of the final decomposition. There are no input parameters. The existence of this keyword causes the d3plot file to be modified so that all elements belonging to the first processor have material type 1, those on the second processor type 2, and so on. Execution terminates immediately after the decomposition phase, and no simulation is performed. This can be used in conjunction with the CONTROL_MPP_DECOMPOSITION_NUMPROC command to run on 1 processor and produce a d3plot file to visualize the resulting decomposition.

***CONTROL_MPP_DECOMPOSITION_TRANSFORMATION**

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_TRANSFORMATION**

Purpose: Specifies transformations to apply to modify the decomposition.

There are 10 different transformations that can be applied. The input is described here. For a detailed description of each decomposition transformation, see the description in the Appendix for the "pfile".

Any number of transformations can appear with no need for further *CONTROL cards – all non-comment cards up the next control card are expected to be decomposition transformations.

The first 6 transformations each take one parameter:

Card 1 2 3 4 5 6 7 8

Variable	TYPE	VAL						
Type	A10	F						
Default	None	0.0						

VARIABLE

DESCRIPTION

TYPE Which transformation to apply. The possible values are: RX, RY, RZ, SX, SY, SZ

VAL The amount of scaling/rotation to apply.

***CONTROL** ***CONTROL_MPP_DECOMPOSITION_TRANSFORMATION**

The remaining 4 transformations each take 9 parameters:

Card 1 1 2 3 4 5 6 7 8

Variable	TYPE	V1	V2	V3	V4	V5	V6	
Type	A10	F	F	F	F	F	F	
Default	None	0.0	0.0	0.0	0.0	0.0	0.0	

Card 2

Variable	V7	V8	V9					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE

DESCRIPTION

TYPE Which transformation to apply. The possible values are: VEC3, C2R, S2R, MAT

V1-V9 Parameters to the transformation.

***CONTROL_MPP_IO_NOD3DUMP**

Purpose: Suppresses the output of all dump files.

There are no input parameters. The existence of this keyword causes the d3dump and runrsf file output routines to be skipped.

***CONTROL**

***CONTROL_MPP_IO_NODUMP**

***CONTROL_MPP_IO_NODUMP**

Purpose: Suppresses the output of all dump files and full deck restart files.

There are no input parameters. The existence of this keyword causes the d3dump and runrsf file output routines to be skipped. It also suppresses output of the full deck restart file d3full.

***CONTROL_MPP_IO_NOFULL**

Purpose: Suppresses the output of the full deck restart files.

There are no input parameters. The existence of this keyword suppresses the output of the full deck restart file "d3full".

***CONTROL**

***CONTROL_MPP_IO_SWAPBYTES**

***CONTROL_MPP_IO_SWAPBYTES**

Purpose: Swap bytes on some of the output files.

There are no input parameters. The existence of this keyword causes the d3plot file and the “interface component analysis” file to be output with bytes swapped. This is to allow further processing of data on a different machine that has big endian vs. little endian incompatibilities compared to the system on which the analysis is running.

***CONTROL_NONLOCAL**

Purpose: Allocate additional memory for *MAT_NONLOCAL option.

Card 1 2 3 4 5 6 7 8

Variable	MEM							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

MEM

Percentage increase of memory allocated for MAT_NONLOCAL option over that required initially. This is for additional storage that may be required due to geometry changes as the calculation proceeds. Generally, a value of 10 should be sufficient.

*CONTROL

*CONTROL_OUTPUT

*CONTROL_OUTPUT

Purpose: Set miscellaneous output parameters. This keyword does not control the information, such as the stress and strain tensors, which is written into the binary databases. For the latter, see the keyword *DATABASE_EXTENT_BINARY.

Card 1 2 3 4 5 6 7 8

Variable	NPOPT	NEECHO	NREFUP	IACCOP	OPIFS	IPNINT	IKEDIT	IFLUSH
Type	I	I	I	I	F	I	I	I
Default	0	0	0	0	0.	0	100	5000

Optional

Card 1 2 3 4 5 6 7 8

Variable	IPRTF	IERODE	TET10	MSGMAX	IPCURV			
Type	I	I	I	I	I			
Default	0	0	2	50	0			

VARIABLE

DESCRIPTION

- NPOPT Print suppression during input phase flag for the printed output file:
EQ.0: no suppression,
EQ.1: nodal coordinates, element connectivities, rigid wall definitions and initial velocities are not printed.
- NEECHO Print suppression during input phase flag for echo file:
EQ.0: all data printed,
EQ.1: nodal printing is suppressed,
EQ.2: element printing is suppressed,
EQ.3: both node and element printing is suppressed.

VARIABLE	DESCRIPTION
NREFUP	Flag to update reference node coordinates for beam elements. This option requires that each reference node is unique to the beam: EQ.0: no update, EQ.1: update.
IACCOP	Flag to average or filter nodal accelerations output to file “nodout” and the time history database “d3thdt”: EQ.0: no average (default), EQ.1: averaged between output intervals, EQ.2: accelerations for each time step are stored internally and then filtered over each output interval using a filter from General Motors [Sala, Neal, and Wang, 2004] based on a low-pass Butterworth frequency filter. See also [Neal, Lin, and Wang, 2004]. DT2MS in *CONTROL_TIMESTEP must be set to a negative value when IACCOP=2 so that the maximum possible number of time steps for an output interval is known and adequate memory can be allocated. See Figure 8.9.
OPIFS	Output interval for interface file (Δt), see INTRODUCTION, Execution syntax.
IPNINT	Print initial time step sizes for all elements on the first cycle: EQ.0: 100 elements with the smallest time step sizes are printed. EQ.1: the governing time step sizes for each element are printed.
IKEDIT	Problem status report interval steps to the D3HSP (printed output) file. This flag is ignored if the GLSTAT file is written, see *DATABASE_GLSTAT.
IFLUSH	Number of time steps interval for flushing I/O buffers. The default value is 5000. If the I/O buffers are not emptied and an abnormal termination occurs, the output files can be incomplete. The I/O buffers for restart files are emptied automatically whenever a restart file is written so these files are not affected by this option.
IPRTF	Default print flag for RBDOUT and MATSUM files. This flag defines the default value for the print flag which can be defined in the part definition section, see *PART. This option is meant to reduce the file sizes by eliminating data which is not of interest. EQ.0: write part data into both MATSUM and RBDOUT EQ.1: write data into RBDOUT file only EQ.2: write data into MATSUM file only EQ.3: do not write data into RBDOUT and MATSUM

VARIABLE	DESCRIPTION
IERODE	Output eroded internal and kinetic energy into the MATSUM file. Also, output to the MATSUM file under the heading of part ID 0 is the kinetic energy from nonstructural mass, lumped mass elements and lumped inertia elements. EQ.0: do not output extra data EQ.1: output the eroded internal and kinetic energy
TET10	Output ten connectivity nodes into D3PLOT database. The current default is set to 2 since this change in the database may make the data unreadable for many popular post-processors and older versions of Ls-prepost. The default will change to 1 later. EQ.1: write the full ten node connectivity into the D3PLOT database EQ.2: write the four corner nodes of the ten node connectivity into the D3PLOT database
MSGMAX	Maximum number of each error/warning message
IPCURV	Flag to output digitized curve data to d3msg and d3hsp files. EQ.0: off EQ.1: on

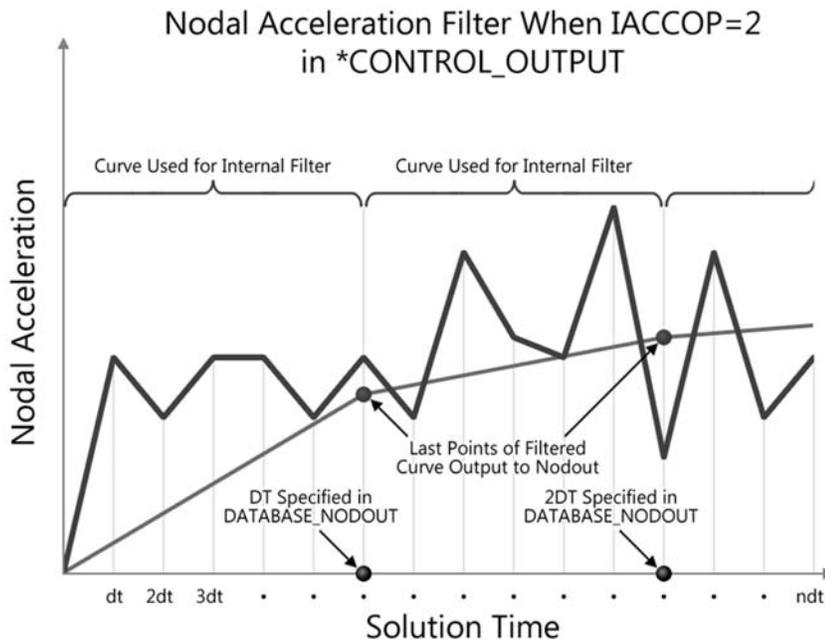


Figure 8.9. Nodal Acceleration Filter for IACOOP=2.

***CONTROL_PARALLEL**

Purpose: Control parallel processing usage for shared memory computers by defining the number of processors and invoking the optional consistency of the global vector assembly.

Card 1 2 3 4 5 6 7 8

Variable	NCPU	NUMRHS	CONST	PARA				
Type	I	I	I	I				
Default	1	0	2	0				
Remarks		1	2	3				

VARIABLE

DESCRIPTION

- NCPU Number of cpus used.
(If runs are submitted through a batch queue or a job scheduler see the Execution Syntax section of Getting Started and the *KEYWORD command for alternative ways to set the number of CPUS.)
- NUMRHS Number of right-hand sides allocated in memory:
EQ.0: same as NCPU, always recommended,
EQ.1: allocate only one.
- CONST Consistency flag for parallel solution (NCPU >1).
Option 2 is recommended for metal forming applications.
EQ.1: on
EQ.2: off, for a faster solution (default).
- PARA Flag for parallel force assembly if CONST=1.
EQ.0: off
EQ.1: on

Remarks:

1. It is recommended to always set NUMRHS=NCPU since great improvements in the parallel performance are obtained since the force assembly is then done in parallel. Setting NUMRHS to one reduces storage by one right hand side vector for each additional processor after the first. If the consistency flag is active, i.e., CONST=1, NUMRHS defaults to unity.
2. For any given problem with the consistency option off, i.e., CONST=2, slight differences in results are seen when running the same job multiple times with the same number of processors and also when varying the number of processors. Comparisons of nodal accelerations often show wide discrepancies; however, it is worth noting that the results of accelerometers often show insignificant variations due to the smoothing effect of the accelerometers which are generally attached to nodal rigid bodies. The accuracy issues are not new and are inherent in numerical simulations of automotive crash and impact problems where structural bifurcations under compressive loads are common. This problem can be easily demonstrated by using a perfectly square thin-walled tubular beam of uniform cross section under a compressive load. Typically, every run on one processor that includes a minor input change (i.e., element or hourglass formulation) will produce dramatically different results in terms of the final shape, and, likewise, if the same problem is again run on a different brand of computer. If the same problem is run on multiple processors the results can vary dramatically from run to run WITH NO INPUT CHANGE. The problem here is due to the randomness of numerical round-off which acts as a trigger in a “perfect” beam. Since summations with (CONST=2) occur in a different order from run to run, the round-off is also random. The consistency flag, CONST=1, provides for identical results (or nearly so) whether one, two, or more processors are used while running in the shared memory parallel (SMP) mode. This is done by requiring that all contributions to global vectors be summed in a precise order independently of the number of processors used. When checking for consistent results, nodal displacements or element stresses should be compared. The NODOUT and ELOUT files should be digit to digit identical. However, the GLSTAT, SECFORC, and many of the other ASCII files will not be identical since the quantities in these files are summed in parallel for efficiency reasons and the ordering of summation operations are not enforced. The biggest drawback of this option is the CPU cost penalty which is at least 15 percent if PARA=0 and is much less if PARA=1 and 2 or more processors are used. Unless the PARA flag is on (for non-vector processors), parallel scaling is adversely affected. The consistency flag does not apply to MPP parallel.
3. The PARA flag will cause the force assembly for the consistency option to be performed in parallel for the shared memory parallel option. Better scaling will be obtained with the consistency option, but with more memory usage. However, the single processing speed is slightly diminished. The logic for parallelization cannot be efficiently vectorized and is not recommended for vector computers since it will degrade CPU performance. This option does not apply to MPP parallel. If PARA=CONST=0 and NUMRHS=NCPU the force assembly by default is done in parallel.

***CONTROL_REMESHING**

Purpose: Provide control over the remeshing of solids which are meshed with the solid tetrahedron element type 13. The element size for three-dimensional adaptivity can be set on the surface mesh of the solid part, and adaptivity can be activated based on the criterions of volume loss, mass increase, or minimum time step size.

Card 1 2 3 4 5 6 7 8

Variable	RMIN	RMAX	VF_LOSS	MFRAC	DT_MIN			
Type	F	F	F	F	F			
Default	none	none	1.0	0.0	0.			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
RMIN	Minimum edge length for the surface mesh surrounding the parts which should be remeshed.
RMAX	Maximum edge length for the surface mesh surrounding the parts which should be remeshed.
VF_LOSS	Volume fraction loss required in a type 13 solid elements to trigger a remesh. In the type 13 solid elements, the pressures are computed at the nodal points; therefore, it is possible for volume to be conserved but for individual tetrahedrons to experience a significant volume loss or gain. The volume loss can lead to numerical problems. Recommended values for VF_LOSS in the range of 0.10 to 0.30 may be reasonable.
MFRAC	Mass ratio gain during mass scaling required for triggering a remesh. For a one percent increase in mass, set MFAC=0.010.
DT_MIN	Time step size required for triggering a remesh. This option is checked before mass scaling is applied and the time step size reset.

Remarks:

1. The value of RMIN and RMAX should be of the same order. The value of RMAX can be set to 2-5 times greater than RMIN.

*CONTROL

*CONTROL_RIGID

*CONTROL_RIGID

Purpose: Special control options related to rigid bodies and the rigid-flexible bodies, see *PART_MODES.

Card	1	2	3	4	5	6	7	8
Variable	LMF	JNTF	ORTHMD	PARTM	SPARSE	METALF		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LMF	Switch the explicit rigid body joint treatment to an implicit formulation which uses Lagrange multipliers to impose prescribed kinematic boundary conditions and joint constraints. This is a new option which is under development in version 970. There is a slight cost overhead due to the assembly of sparse matrix equations which are solved using standard procedures for nonlinear problems in rigid multi-body dynamics. Lagrange multiplier flag: EQ.0: explicit penalty formulation, EQ.1: implicit formulation with Lagrange multipliers.
JNTF	Generalized joint stiffness formulation; see remark 1 below: EQ.0: incremental update, EQ.1: total formulation (exact).
ORTHMD	Orthogonalize modes with respect to each other: EQ.0: true. EQ.1: false, the modes are already orthogonalized.
PARTM	Use global mass matrix to determine part mass distribution. This mass matrix may contain mass from other parts that share nodes. See remark 2 below. EQ.0: true, EQ.1: false.
SPARSE	Use sparse matrix multiply subroutines for the modal stiffness and damping matrices. See Remark 3. EQ.0: false, do full matrix multiplies (frequently faster), EQ.1: true.

VARIABLE	DESCRIPTION
MATELF	<p>Metalforming option, which should not be used for crash and other applications involving rigid bodies. Use fast update of rigid body nodes. If this option is active the rotational motion of all rigid bodies should be suppressed.</p> <p>EQ.0: full treatment is used EQ.1: fast update for metalforming applications</p>

Remarks:

1. As the default, the calculation of the relative angles between two coordinate systems is done incrementally. This is an approximation, in contrast to the total formulation where the angular offsets are computed exactly. The disadvantage of the latter approach is that a singularity exists when an offset angle equals 180 degrees. For most applications, the stop angles prevent this occurrence and JNTF=1 should not cause a problem.
2. If the determination of the normal modes included the mass from both connected bodies and discrete masses, or if there are no connected bodies, then the default is preferred. When the mass of a given part ID is computed, the resulting mass vector includes the mass of all rigid bodies that are merged to the given part ID, but does not include discrete masses. See the keyword: *CONSTRAINED_RIGID_BODIES. A lumped mass matrix is always assumed.
3. Sparse matrix multipliers save a substantial number of operations if the matrix is truly sparse. However, the overhead will slow the multipliers for densely populated matrices.

*CONTROL

*CONTROL_SHELL

*CONTROL_SHELL

Purpose: Provide controls for computing shell response.

Card 1 2 3 4 5 6 7 8

Variable	WRPANG	ESORT	IRNXX	ISTUPD	THEORY	BWC	MITER	PROJ
Type	F	I	I	I	I	I	I	I
Default	20.	0	-1	0	2	2	1	0

First optional Card

Card 1 2 3 4 5 6 7 8

Variable	ROTASCL	INTGRD	LAMSHT	CSTYP6	TSHELL	NFAIL1	NFAIL4	PSNFAIL
Type	F	I	I	I	I	I	I	I
Default	1..	0	0	1	0	inactive	inactive	0

Second optional Card (The first card must also be defined)

Card 1 2 3 4 5 6 7 8

Variable	PSSTUPD	IRQUAD						
Type	I	I						
Default	1..	0						

VARIABLE

DESCRIPTION

WRPANG

Shell element warpage angle in degrees. If a warpage greater than this angle is found, a warning message is printed. Default is 20 degrees.

VARIABLE	DESCRIPTION
ESORT	<p>Automatic sorting of triangular shell elements to treat degenerate quadrilateral shell elements as C0 or DKT triangular shells, see option THEORY below:</p> <p>EQ.0: no sorting required (default), EQ.1: full sorting (C0 triangular shells), EQ.2: full sorting (DKT triangular shells) 971 version R3 and later.</p>
IRNXX	<p>Shell normal update option. This option affects the Hughes-Liu, Belytschko-Wong-Chiang, and the Belytschko-Tsay shell formulations. The latter is affected if and only if the warping stiffness option is active, i.e., BWC=1.</p> <p>EQ.-2: unique nodal fibers which are incrementally updated based on the nodal rotation at the location of the fiber, EQ.-1: recomputed fiber directions each cycle, EQ.0: default set to -1, EQ.1: compute on restarts, EQ.n: compute every n cycles (Hughes-Liu shells only).</p>
ISTUPD	<p>Shell thickness change option for deformable shells. The parameter, PSSTUPD, on the second optional card allows this option to be applied by part ID. For crash analysis, neglecting the elastic component of the strains, ISTUPD=4, may improve energy conservation and stability.</p> <p>EQ.0: no thickness change. EQ.1: membrane straining causes thickness change in 3 and 4 node shell elements. This option is very important in sheet metal forming or whenever membrane stretching is important. EQ.2: membrane straining causes thickness change in 8 node thick shell elements, types 1 and 2. This option is not recommended for implicit or explicit solutions which use the fully integrated type 2 element. The type 3 thick shell is a continuum based shell and thickness changes are always considered. EQ.3: options 1 and 2 apply. EQ.4: option 1 applies, but the elastic strains are neglected for the thickness update. This option only applies to the most common elastic-plastic materials for which the elastic response is isotropic.</p>
THEORY	<p>Default shell theory:</p> <p>EQ.1: Hughes-Liu, EQ.2: Belytschko-Tsay (default), EQ.3: BCIZ triangular shell (not recommended), EQ.4: C⁰ triangular shell, EQ.5: Belytschko-Tsay membrane, EQ.6: S/R Hughes Liu, EQ.7: S/R co-rotational Hughes Liu, EQ.8: Belytschko-Leviathan shell,</p>

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>EQ.9: fully integrated Belytschko-Tsay membrane, EQ.10: Belytschko-Wong-Chiang, EQ.11: Fast (co-rotational) Hughes-Liu. EQ.12: Plane stress (x-y plane), EQ.13: Plane strain (x-y plane), EQ.14: Axisymmetric solid (y-axis of symmetry) - area weighted, EQ.15: Axisymmetric solid (y-axis of symmetry) – volume weighted EQ.16: Fully integrated shell element (very fast) EQ.17: Discrete Kirchhoff triangular shell (DKT) EQ.18: Discrete Kirchhoff linear shell either quadrilateral or Triangular with 6DOF per node, EQ.20: C^0 linear shell element with 6 DOF per node EQ.21: C^0 linear shell element with 5 DOF per node with the Pian-Sumihara membrane hybrid quadrilateral membrane. EQ.25: Belytschko-Tsay shell with thickness stretch. EQ.26: Fully integrated shell with thickness stretch. EQ.27: C^0 triangular shell with thickness stretch.</p>
	<p>For the 2D axisymmetric solid elements, high explosive applications work best with the area weighted approach and structural applications work best with the volume weighted approach. The volume weighted approach can lead to problems along the axis of symmetry under very large deformations. Often the symmetry condition is not obeyed, and the elements will kink along the axis. The volume weighted approach must be used if 2D shell elements are used in the mesh. Type 14 and 15 elements cannot be mixed in the same calculation.</p>
BWC	<p>Warping stiffness for Belytschko-Tsay shells: EQ.1: Belytschko-Wong-Chiang warping stiffness added. EQ.2: Belytschko-Tsay (default).</p>
MITER	<p>Plane stress plasticity option (applies to materials 3, 18, 19, and 24): EQ.1: iterative plasticity with 3 secant iterations (default), EQ.2: full iterative plasticity, EQ.3: radial return noniterative plasticity. May lead to false results and has to be used with great care.</p>
PROJ	<p>Projection method for the warping stiffness in the Belytschko-Tsay shell (the BWC option above) and the Belytschko-Wong-Chiang elements (see remarks below). This parameter applies to explicit calculations since the full projection method is always used if the solution is implicit and this input parameter is ignored. EQ.0: drill projection, EQ.1: full projection.</p>

VARIABLE	DESCRIPTION
ROTASCL	Define a scale factor for the rotary shell mass. This option is not for general use. The rotary inertia for shells is automatically scaled to permit a larger time step size. A scale factor other than the default, i.e., unity, is not recommended.
INTGRD	Default shell through thickness numerical integration rule: EQ.0: Gauss integration. If 1-10 integration points are specified, the default rule is Gauss integration. EQ.1: Lobatto integration. If 3-10 integration points are specified, the default rule is Lobatto. For 2 point integration, the Lobatto rule is very inaccurate, so Gauss integration is used instead. Lobatto integration has an advantage in that the inner and outer integration points are on the shell surfaces.
LAMSHT	For composite shells with material types *MAT_COMPOSITE_DAMAGE and *MAT_ENHANCED_COMPOSITE_DAMAGE. If this flag is set laminated shell theory is used. Lamination theory is applied to correct for the assumption of a uniform constant shear strain through the thickness of the shell. Unless this correction is applied, the stiffness of the shell can be grossly incorrect if there are drastic differences in the elastic constants from ply to ply, especially for sandwich type shells. Generally, without this correction the results are too stiff. For the discrete Kirchhoff shell elements, which do not consider transverse shear, this option is ignored. EQ.0: do not update shear corrections, EQ.1: activate laminated shell theory.
CSTYP6	Coordinate system for the type 6 shell element. The default system computes a unique local system at each in plane point. The uniform local system computes just one system used throughout the shell element. This involves fewer calculations and is therefore more efficient. The change of systems has a slight effect on results; therefore, the older, less efficient method is the default. EQ.1: variable local coordinate system (default), EQ.2: uniform local system.
TSHELL	Thermal shell option. Four node shells are treated internally as twelve node brick elements to allow heat conduction through the thickness of the shell.
NFAIL1	Flag to check for highly distorted under-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is not needed for one point elements that do not use the warping stiffness. A distorted element is one where a negative Jacobian exist within the domain of the shell, not just at integration points. The checks are made away from the CPU requirements for one point elements. If nonzero, NFAIL1 can be changed in a restart.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.1: print message and delete element. EQ.2: print message, write D3DUMP file, and terminate GT.2: print message and delete element. When NFAIL1 elements are deleted then write D3DUMP file and terminate. These NFAIL1 failed elements also include all shell elements that failed for other reasons than distortion. Before the D3DUMP file is written, NFAIL1 is doubled, so the run can immediately be continued if desired.
NFAIL4	Flag to check for highly distorted fully-integrated shell elements, print a message and delete the element or terminate. Generally, this flag is recommended. A distorted element is one where a negative Jacobian exist within the domain of the shell, not just at integration points. The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs. If nonzero, NFAIL1 can be changed in a restart. EQ.1: print message and delete element. EQ.2: print message, write D3DUMP file, and terminate GT.2: print message and delete element. When NFAIL4 elements are deleted then write D3DUMP file and terminate. These NFAIL4 failed elements also include all shell elements that failed for other reasons than distortion. Before the D3DUMP file is written, NFAIL4 is doubled, so the run can immediately be continued if desired.
PSNFAIL	Optional shell part set ID specifying which part ID's are checked by the NFAIL1 and NFAIL4 options. If zero, all shell part ID's are included.
PSSTUPD	PSSTUPD is the optional shell part set ID specifying which part ID's have or do not have their thickness updated. The shell thickness update by default applies to all shell elements in the mesh. Generally, this part set ID is not needed. LT.0: these shell parts are excluded from the shell thickness update EQ.0: all deformable shells have their thickness updated GT.0: these shell parts are included in the shell thickness update
IRQUAD	In plane integration rule for the 8 node shell element: EQ.2: 2 x 2 Gauss quadrature, EQ.3: 3 x 3 Gauss quadrature.

Remarks:

1. The drill projection is used in the addition of warping stiffness to the Belytschko-Tsay and the Belytschko-Wong-Chiang shell elements. This projection generally works well and is very efficient, but to quote Belytschko and Leviathan:

"The shortcoming of the drill projection is that even elements that are invariant to rigid body rotation will strain under rigid body rotation if the drill projection is applied. On one hand, the excessive flexibility rendered by the 1-point quadrature shell element is corrected by the drill projection, but on the other hand the element becomes too stiff due to loss of the rigid body rotation invariance under the same drill projection".

They later went on to add in the conclusions:

"The projection of only the drill rotations is very efficient and hardly increases the computation time, so it is recommended for most cases. However, it should be noted that the drill projection can result in a loss of invariance to rigid body motion when the elements are highly warped. For moderately warped configurations the drill projection appears quite accurate".

In crashworthiness and impact analysis, elements that have little or no warpage in the reference configuration can become highly warped in the deformed configuration and may affect rigid body rotations if the drill projection is used, i.e., **DO NOT USE THE DRILL PROJECTION**. Of course it is difficult to define what is meant by "moderately warped". The full projection circumvents these problems but at a significant cost. The cost increase of the drill projection versus no projection as reported by Belytschko and Leviathan is 12 percent and by timings in LS-DYNA, 7 percent, but for the full projection they report a 110 percent increase and in LS-DYNA an increase closer to 50 percent is observed.

In Version 940.xx of LS-DYNA the drill projection was used exclusively, but in one problem the lack of invariance was observed and reported; consequently, the drill projection was replaced in the Belytschko-Leviathan shell with the full projection and the full projection is now optional for the warping stiffness in the Belytschko-Tsay and Belytschko-Wong-Chiang elements. Until this problem occurred, the drill projection seemed okay. In version 950.xx and later versions of LS-DYNA the Belytschko-Leviathan shell is somewhat slower than previously. In general in light of these problems, the drill projection cannot be recommended. For implicit problems, the full projection method is used in the development of the stiffness matrix.

*CONTROL

*CONTROL_SOLID

*CONTROL_SOLID

Purpose: Provide controls for solid element response.

Card 1 2 3 4 5 6 7 8

Variable	ESORT	FMATRX	NIPTETS	SWLOCL				
Type	I	I	I	I				
Default	0	0	4	2				

This card is optional. Card Format (10I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable	PM1	PM2	PM3	PM4	PM5	PM6	PM7	PM8	PM9	PM10
Type	I	I	I	I	I	I	I	I	I	I
Default	none									

VARIABLE

DESCRIPTION

ESORT	Automatic sorting of tetrahedron and pentahedron elements to treat degenerate tetrahedron and pentahedron elements as tetrahedron (formulation 10) and pentahedron (formulation 15) solids, respective. See *SECTION_SOLID. EQ.0: no sorting required (default). EQ.1: full sorting,
FMATRX	Default method used in the calculation of the deformation gradient matrix. EQ.1: Update incrementally in time. This is the default for explicit. EQ.2: Directly compute F. This is the default for implicit and implicit/explicit switching.
NIPTETS	Number of integration points used in the quadratic tetrahedron elements. Either 4 or 5 can be specified. This option applies to the types 4, 16, and 17 tetrahedron elements.

VARIABLE	DESCRIPTION
SWLOCL	Output option for stresses in solid elements used as spot welds with material *MAT_SPOTWELD. EQ.1: Local, EQ.2: Global (default)
PM1-PM10	Components of a permutation vector for nodes that define the 10-node tetrahedron. The nodal numbering of 10-node tetrahedron elements is somewhat arbitrary. The permutation vector allows other numbering schemes to be used. Unless defined, this permutation vector is not used. PM1-PM10 are unique numbers between 1 to 10 inclusive that reorders the input node ID's for a 10-node tetrahedron into the order used by LS-DYNA.

*CONTROL

*CONTROL_SOLUTION

*CONTROL_SOLUTION

Purpose: To specify the analysis solution procedure if thermal only or coupled thermal analysis is performed. Other solutions parameters including the vector length and NaN (not a number) checking can be set.

Card	1	2	3	4	5	6	7	8
Variable	SOLN	NLQ	ISNAN	LCINT				
Type	I	I	I	I				
Default	0	0	0	100				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SOLN	Analysis solution procedure: EQ.0: Structural analysis only, EQ.1: Thermal analysis only, EQ.2: Coupled structural thermal analysis.
NLQ	Define the vector length used in solution. This value must not exceed the vector length of the system which varies based on the machine manufacturer. The default vector length is printed at termination in the MESSAG file.
ISNAN	Flag to check for a NaN in the force and moment arrays after the assembly of these arrays is completed. This option can be useful for debugging purposes. A cost overhead of approximately 2% is incurred when this option is active. EQ.0: No checking, EQ.1: Checking is active.
LCINT	Number of equally spaced intervals used in the load curve discretization.

***CONTROL_SPH**

Purpose: Provide controls for computing SPH particles

Card 1 2 3 4 5 6 7 8

Variable	NCBS	BOXID	DT	IDIM	MEMORY	FORM	START	MAXV
Type	I	I	F	I	I	I	F	F
Default	1	0	1.e20	none	150	0	0.0	1.e15

Optional Card

Card 1 2 3 4 5 6 7 8

Variable	CONT	DERIV	INI					
Type	I	I	I					
Default	0	0	0					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NCBS	Number of cycles between particle sorting
BOXID	SPH approximations are computed inside a specified BOX. When a particle has gone outside the BOX, it is deactivated. This will save computational time by eliminating particles that no longer interact with the structure.
DT	Death time. Determines when the SPH calculations are stopped.
IDIM	Space dimension for SPH particles: 3 for 3D problems 2 for 2D plane strain problems When a value is not specified LS-DYNA determines the space dimension automatically by checking the use of 3D, 2D or 2D axisymmetric elements.

VARIABLE	DESCRIPTION
MEMORY	Defines the initial number of neighbors per particle (see Remark 1 below).
FORM	Particle approximation theory: EQ.0: default formulation, EQ.1: renormalization approximation EQ.2: symmetric formulation, EQ.3: symmetric renormalized approximation EQ.4: tensor formulation, EQ.5: fluid particle approximation EQ.6: fluid particle with renormalization approximation,
START	Start time for particle approximation. Particle approximations will be computed when time of the analysis has reached the value defined in START.
MAXV	Maximum value for velocity for the SPH particles. Particles with a velocity greater than MAXV are deactivated.
CONT	Defines the computation of the particle approximation between two different SPH parts: EQ.0: Particle approximation is defined (default) EQ.1: Particle approximation is not computed. Two different SPH materials will not interact with each others and penetration is allowed.
DERIV	Time integration type for the smoothing length: EQ.0: $\frac{d}{dt}(h(t)) = \frac{1}{d} h(t) \text{div}(v)$ (default), EQ.1: $\frac{d}{dt}(h(t)) = \frac{1}{d} h(t) (\text{div}(v))^{1/3}$
INI	Computation of the smoothing length during the initialization: EQ.0: Bucket sort based algorithm (default, very fast) EQ.1: Global computation on all the particles of the model.

Remark:

1. This variable is for memory allocation of arrays during the initialization phase. It can be positive or negative. If this value is positive, memory allocation is dynamic. During the calculation, some particles can request more neighbors and LS-DYNA will automatically adapt the size of that variable. Default value should apply for most applications. If this value is negative, memory allocation is static. During the calculation only the closest SPH elements will be considered as neighbors. Using this option can avoid memory allocation problems.

***CONTROL_SPOTWELD_BEAM**

Purpose: Provides factors for scaling the failure force resultants of beam spot welds as a function of their parametric location on the contact segment and the size of the segment. Also, an option is provided to replace beam welds with solid hexahedron element clusters.

Card 1 2 3 4 5 6 7 8

Variable	LCT	LCS	T_ORT	PRTFLG	T_ORO	RPBHX		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCT	Load curve ID for scaling the response in tension based on the shell element size.
LCS	Load curve ID for scaling the response in shear based on the shell element size.
T_ORO	Table ID for scaling the tension response (and shear response if T_ORO=0) based on the location of the beam node relative to the centroid of the shell.
PRTFLG	Set this flag to 1 to print for each spot weld attachment: the beam, node, and shell ID's, the parametric coordinates that define the constraint location, the angle used in the table lookup, and the three scale factors obtained from the load curves and table lookup. See Figure 8.10.
T_ORO	Optional table ID for scaling the shear response based on the location of the beam node relative to the centroid of the shell.
RPBHX	Replace each spot weld beam element with a cluster of RPBHX solid elements. RPBHX may be set to 1, 4, or 8. When RPBHX is set to 4 or 8, a table is generated to output the force and moment resultants into the SWFORC file, if this file is active. This table is described by the keyword: *DEFINE_HEX_SPOTWELD_ASSEMBLY. The ID's of the beam elements are used as the cluster spot weld ID's so the ID's in the SWFORC file are unchanged. The beam elements are automatically deleted from the calculation, and the section and material data is automatically changed to be used with solid elements. See Figure 11.8.

The load curves and table provide a means of scaling the response of the beam spot welds to reduce any mesh dependencies for failure model 6 in *MAT_SPOTWELD. Figure 8.11 shows such dependencies that can lead to premature spot weld failure. Separate scale factors are calculated for each of the beam's nodes. The scale factors s_T , s_S , s_{OT} , and s_{OS} are calculated using the load curves LCT, LCS, table T_ORT, and table T_ORS, respectively, and are introduced in the failure criteria,

$$\left(\frac{s_T s_{OT} \sigma_{rr}}{\sigma_{rr}^F(\dot{\epsilon}_{eff})} \right)^2 + \left(\frac{s_S s_{OS} \tau}{\tau^F(\dot{\epsilon}_{eff})} \right)^2 - 1 = 0$$

If a curve or table is given an ID of 0, its scale factor is set to 1.0. The load curves LCT and LCS are functions of the characteristic size of the shell element used in the time step calculation at the start of the calculation. The orientation table is a function of the spot weld's isoparametric coordinate location on the shell element. A vector $V=(s,t)$ is defined from the centroid of the shell to the contact point of the beam's node. The arguments for the orientation table are the angle:

$$\Theta = \tan^{-1} \left(\frac{\min(|s|, |t|)}{\max(|s|, |t|)} \right),$$

and the normalized distance $\bar{d} = d/D = \max(|s|, |t|)$. See Figure 8.10. The table is periodic over a range of 0 (V aligned with either the s or t axis) to 45 degrees (V is along the diagonal of the element). The table is specified by the angle of V in degrees, ranging from 0 to 45, and the individual curves give the scale factor as a function of the normalized distance of the beam node, \bar{d} , for a constant angle.

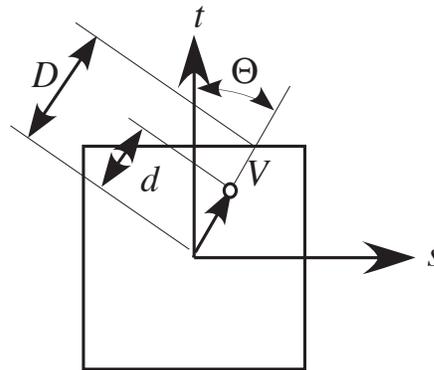


Figure 8.10. Definition of parameters for table definition.

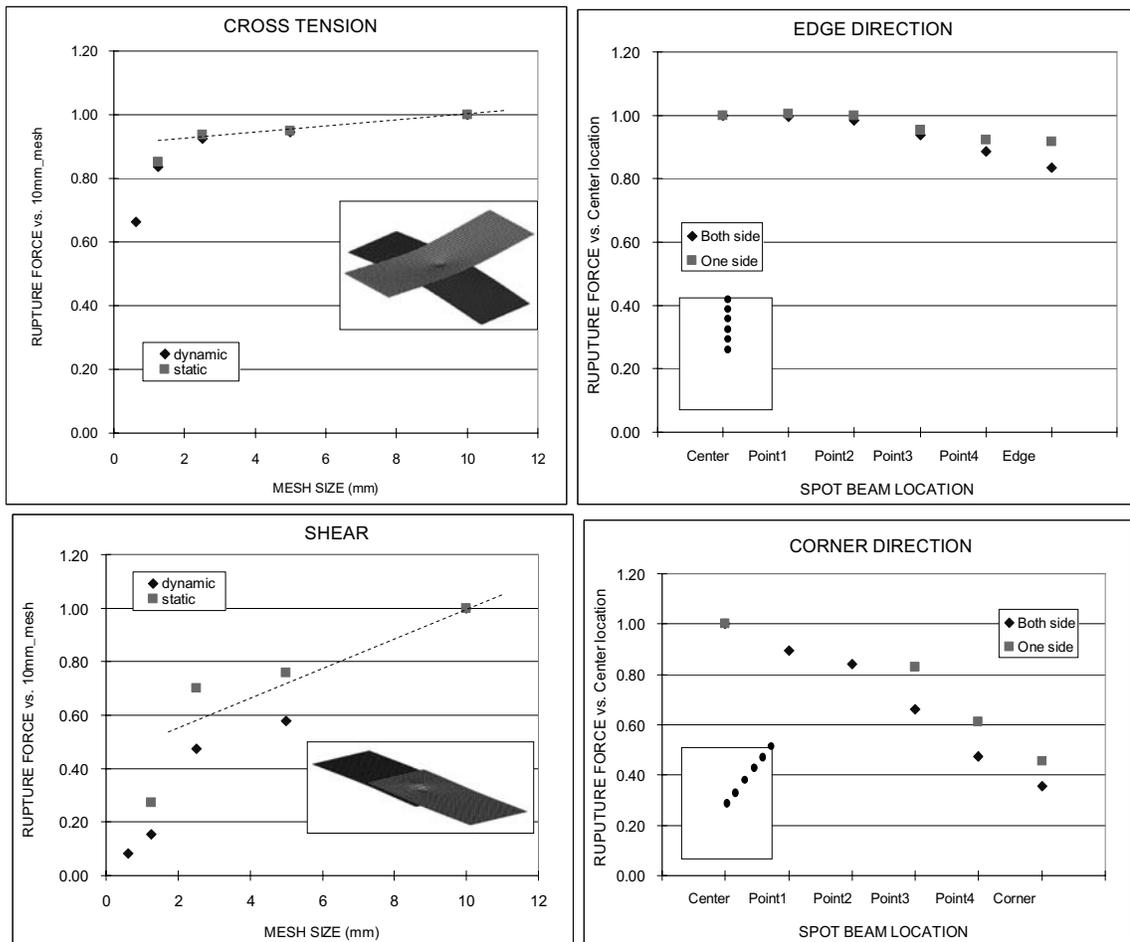


Figure 8.11. The failure force resultants can depend both on mesh size and the location of weld relative to the center of the contact segment.

*CONTROL

*CONTROL_STAGED_CONSTRUCTION

*CONTROL_STAGED_CONSTRUCTION

This control card is used to help break down analyses of construction processes into stages.

Note: This keyword card will be available starting in release 3 of version 971.

Card	1	2	3	4	5	6	7	8
Variable	TSTART	STGS	STGE	ACCEL	FACT	STREF		
Type	F	I	I	F	F	I		
Default	0	0	0	0.0	1.e-6	0		

VARIABLE

DESCRIPTION

TSTART	Time at start of analysis (normally leave blank)
STGS	Construction stage at start of analysis
STGE	Construction stage at end of analysis
ACCEL	Default acceleration for gravity loading
FACT	Default stiffness and gravity factor for parts before they are added
STREF	Reference stage for displacements in d3plot file

Remarks:

See also *DEFINE_CONSTRUCTION_STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.

The staged construction options offer flexibility to carry out the whole construction simulation in one analysis, or to run it stage by stage. Provided that at least one construction stage is defined (*DEFINE_CONSTRUCTION_STAGES), a dynain file will be written at the end of each stage (file names are end_stage001_dynain, etc). These contain node and element definitions and the stress state; the individual stages can then be re-run without re-running the whole analysis. To do this, make a new input file as follows:

- Copy the original input file, containing *DEFINE_CONSTRUCTION_STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.

- Delete node and element definitions as these will be present in the dynain file (*NODE, *ELEMENT_SOLID, *ELEMENT_SHELL, and *ELEMENT_BEAM).
- Delete any *INITIAL cards; the initial stresses in the new analysis will be taken from the dynain file.
- On *CONTROL_STAGED_CONSTRUCTION set STGS to start at the desired stage
- Add an *INCLUDE statement referencing, for example, end_stage002_dynain if starting the new analysis from Stage 3.
- Move or copy the dynain file into the same directory as the new input file.

When STGS is >1 the analysis starts at a non-zero time (the start of stage STGS). In this case a dynain file must be included to start the analysis from the stress state at the end of the previous stage. The end time for stage STGE overrides the termination time on *CONTROL_TERMINATION. A new dynain file will be written at the end of all stages from STGS to STGE.

ACCEL and FACT are used with *STAGED_CONSTRUCTION_PART for simpler input definition of the parts present at different construction stages.

If STGS>1 and elements have been deleted in a previous stage, these elements will be absent from the new analysis and should not be referred to (e.g. *DATABASE_HISTORY_SOLID) in the new input file.

TSTART can be used to set a non-zero start time (again, assuming a compatible dynain file is included). This option is used only if construction stages have not been defined.

STREF allows the user to set a construction stage at the start of which displacements are considered to be zero – e.g. so that initial analysis stages that achieve a pre-construction equilibrium do not contribute to contour plots of displacement. The current coordinates are not modified, only the “initial geometry” coordinates in the d3plot file. If this analysis starts from a stage later than STREF, the reference geometry will be taken from the dynain file that was written at the end of the stage previous to STREF – this dynain file must be in the same directory as the current model for this process to occur.

***CONTROL_STRUCTURED_{OPTION}**

Available options include:

<BLANK>

TERM

Purpose: Write out a LS-DYNA structured input deck for Version 970. The name of this structured file is "dyna.str". This input deck will not support all capabilities that are available in Version 970. As a result some data such as load curve numbers will be output in an internal numbering system. If the TERM option is activated termination will occur after the structured input file is written. This option is useful in debugging especially if problems occur in reading the input file.

***CONTROL_SUBCYCLE**

Purpose: Control time step subcycling. This feature is described in the LS-DYNA Theory Manual, Section 21.2, and its use may be detrimental in cases of vectorized computation. This keyword activates subcycling. The use of mass scaling to preserve a reasonable time step size often works better than subcycling. To use mass scaling set the input parameter, DT2MS, to the negative value of the minimum acceptable time step size. See the keyword, *CONTROL_TIMESTEP.

*CONTROL

*CONTROL_TERMINATION

*CONTROL_TERMINATION

Purpose: Stop the job.

Card 1 2 3 4 5 6 7 8

Variable	ENDTIM	ENDCYC	DTMIN	ENDENG	ENDMAS			
Type	F	I	F	F	F			
Default	0.0	0	0.0	0.0	0.0			
Remarks	1		2					

VARIABLE

DESCRIPTION

ENDTIM	Termination time. Mandatory.
ENDCYC	Termination cycle. The termination cycle is optional and will be used if the specified cycle is reached before the termination time. Cycle number is identical with the time step number.
DTMIN	Reduction (or scale) factor for initial time step size to determine minimum time step, TSMIN. $TSMIN = DTSTART * DTMIN$ where DTSTART is the initial step size determined by LS-DYNA. When TSMIN is reached, LS-DYNA terminates with a restart dump.
ENDENG	Percent change in energy ratio for termination of calculation. If undefined, this option is inactive.
ENDMAS	Percent change in the total mass for termination of calculation. This option is relevant if and only if mass scaling is used to limit the minimum time step size, see *CONTROL_TIMESTEP variable name "DT2MS".

Remarks:

1. Termination by displacement may be defined in the *TERMINATION section.
2. If the erosion flag on *CONTROL_TIMESTEP is set (ERODE=1), then the shell elements and solid elements with time steps falling below TSMIN will be eroded.

***CONTROL_THERMAL_NONLINEAR**

Purpose: Set parameters for a nonlinear thermal or coupled structural/thermal analysis. The control card, *CONTROL_SOLUTION, is also required.

Card 1 2 3 4 5 6 7 8

Variable	REFMAX	TOL	DCP	LUMPBC	THLSTL	NLTHPR	PHCHPN	
Type	I	F	F	I	F	I	F	
Default	10	1.e-04	1.0 / 0.5	0	0.	0	0.	

VARIABLE	DESCRIPTION
REFMAX	Maximum number of matrix reformations per time step: EQ.0: set to 10 reformations.
TOL	Convergence tolerance for temperature: EQ.0.0: set to 1000 * machine roundoff.
DCP	Divergence control parameter: steady state problems $0.3 \leq DCP \leq 1.0$ default 1.0 transient problems $0.0 < DCP \leq 1.0$ default 0.5
LUMPBC	Lump enclosure radiation boundary condition: EQ.0: off (default) EQ.1: on
THLSTL	Line search convergence tolerance: EQ.0.0: No line search GT.0.0: Line search convergence tolerance
NLTHPR	Thermal nonlinear print out level: EQ.0: No print out EQ.1: 1 Print convergence parameters during solution of nonlinear system
PHCHPN	Phase change penalty parameter: EQ.0.0: Penalty formulation not activated GT.0.0: Penalty to enforce constant phase change temperature

*CONTROL

*CONTROL_THERMAL_SOLVER

*CONTROL_THERMAL_SOLVER

Purpose: Set options for the thermal solution in a thermal only or coupled structural-thermal analysis. The control card, *CONTROL_SOLUTION, is also required.

Card 1 1 2 3 4 5 6 7 8

Variable	ATYPE	PTYPE	SOLVER	CGTOL	GPT	EQHEAT	FWORK	SBC
Type	I	I	I	F	I	F	F	F
Default	0	0	3	1.0e-04	8	1.	1.	0.

Optional Card (Define if SOLVER = 11, 12, 13, 14, 15 or 16)

Card 2 1 2 3 4 5 6 7 8

Variable	MSGLVL	MAXITR	ABSTOL	RELTOL	OMEGA			
Type	I	I	F	F	F			
Default	0	500	1.0e-10	1.0e-04	1.0 or 0.			

VARIABLE

DESCRIPTION

ATYPE	Thermal analysis type: EQ.0: Steady state analysis, EQ.1: transient analysis.
PTYPE	Thermal problem type: (see *CONTROL_THERMAL_NONLINEAR if no-zero) EQ.0: linear problem, EQ.1: nonlinear problem with material properties evaluated at gauss point temperature. EQ.2: nonlinear problem with material properties evaluated at element average temperature.

VARIABLE	DESCRIPTION
SOLVER	Thermal analysis solver type: EQ.1:symmetric direct solver, EQ.2:nonsymmetric direct solver, EQ.3:diagonal scaled conjugate gradient iterative (default), EQ.4:incomplete choleski conjugate gradient iterative, EQ.5:nonsymmetric diagonal scaled bi-conjugate gradient EQ.11:symmetric direct solver (recommended over #1), For MPP executions: EQ.11:symmetric direct solver, EQ.12:diagonal scaling (default for mpp) conjugate gradient iterative, EQ.13:symmetric Gauss-Siedel conjugate gradient iterative, EQ.14:SSOR conjugate gradient iterative, EQ.15: ILDLT0 (incomplete factorization) conjugate gradient iterative, EQ.16:modified ILDLT0 (incomplete factorization) conjugate gradient iterative.
CGTOL	Convergence tolerance for SOLVER = 3 and 4. EQ.0.0:use default value 1.e-04
GPT	Number of Gauss points to be used in the solid elements: EQ.0.0:use default value 8, EQ.1.0:one point quadrature is used.
EQHEAT	Mechanical equivalent of heat (e.g., 1 J / N m). EQ.0.0:use default value 1.0, LT.0.0:designates a load curve number for EQHEAT versus time.
FWORK	Fraction of mechanical work converted into heat. EQ.0.0:use default value 1.0.
SBC	Stefan Boltzmann constant. Value is used with enclosure radiation surfaces, see *BOUNDARY_RADIATION_..... LT.0.0:use a smoothing algorithm when calculating view factors to force the row sum=1.
MSGLVL	Output message level (For SOLVER > 10) EQ.0:no output (default), EQ.1:summary information, EQ.2:detailed information, use only for debugging.
MAXITR	Maximum number of iterations. For SOLVER >11. EQ.0:use default value 500,
ABSTOL	Absolute convergence tolerance. For SOLVER >11. EQ.0.0:use default value 1.e-10,

<u>VARIABLE</u>	<u>DESCRIPTION</u>
RELTOL	Relative convergence tolerance. Replaces CGTOL for SOLVER >11. EQ.0.0:use default value 1.e-04,
OMEGA	Relaxation parameter omega for SOLVER 14 and 16. EQ.0.0:use default value 1.0 for Solver 14, use default value 0.0 for Solver 16.

Remarks:

1. Solvers 1, 2, 3 and 4 are only for SMP environments. Solvers 11, 12, 13, 14, 15 and 16 are for SMP and MPP.
2. Solver 11 is the preferred direct solver. Solver 11 uses sparse matrix storage and requires much less memory than Solver 1.
3. Use of a direct solver (e.g., SOLVER = 1, 2 or 11) is usually less efficient than using an iterative solver (SOLVER = 3, 4, 12, 13, 14, 15 or 16). Consider using a direct solver to get the model running and then switch to an iterative solver to decrease execution time (particularly for large models). Direct solvers should be used when experiencing slow or no convergence.
4. For transient problems, diagonal scaling conjugate gradient (SOLVER = 3 or 12) should be adequate.
5. For steady state problems, convergence may be slow or unacceptable, so consider using direct solver (SOLVER = 1, 2 or 11) or a more powerful preconditioner (SOLVER = 4, 13, 14, 15 or 16).
6. Solver 13 (symmetric Gauss-Seidel) and solver 14 (SSOR) are related. When OMEGA = 1, solver 14 is equivalent to solver 13. The optimal omega value for SSOR is problem dependent but lies between 1 and 2.
7. Solver 15 (incomplete LDLT0) and solver 16 (modified incomplete LDLT0) are related. Both are no-fill factorizations that require one extra n-vector of storage. The sparsity pattern of the preconditioner is exactly the same as that of the thermal stiffness matrix. Solver 16 uses the relaxation parameter OMEGA. The optimal OMEGA value is problem dependent, but lies between 0 and 1.
8. Solvers 12, 13, 14, 15 and 16 terminate the iterative solution process when (1) the number of iterations exceeds MAXITR or (2) the 2-norm of the residual drops below ABSTOL + RELTOL*2-norm of the initial residual.

CONTROL_THERMAL_TIMESTEP**CONTROL*****CONTROL_THERMAL_TIMESTEP**

Purpose: Set time step controls for the thermal solution in a thermal only or coupled structural/thermal analysis. Also *CONTROL_SOLUTION, *CONTROL_THERMAL_SOLVER needed.

Card 1 2 3 4 5 6 7 8

Variable	TS	TIP	ITS	TMIN	TMAX	DTEMP	TSCP	LCTS
Type	I	F	F	F	F	F	F	I
Default	0	0.5	none	-	-	1.0	0.5	0

VARIABLE**DESCRIPTION**

TS	Time step control: EQ.0: fixed time step, EQ.1: variable time step (may increase or decrease).
TIP	Time integration parameter: EQ.0.0: set to 0.5 - Crank-Nicholson scheme, EQ 1.0: fully implicit.
ITS	Initial thermal time step
TMIN	Minimum thermal time step: EQ.0.0: set to structural explicit time step.
TMAX	Maximum thermal time step: EQ.0.0: set to 100 * structural explicit time step.
DTEMP	Maximum temperature change in each time step above which the thermal time step will be decreased: EQ.0.0: set to a temperature change of 1.0.
TSCP	Time step control parameter. The thermal time step is decreased by this factor if convergence is not obtained. 0. < TSCP < 1.0: EQ.0.0: set to a factor of 0.5.
LCTS	LCTS designates a load curve number which defines the thermal time step as a function of time. If LCTS is defined, then the other time step control parameters on this keyword are ignored.

*CONTROL

*CONTROL_TIMESTEP

*CONTROL_TIMESTEP

Purpose: Set structural time step size control using different options.

Card 1 1 2 3 4 5 6 7 8

Variable	DTINIT	TSSFAC	ISDO	TSLIMIT	DT2MS	LCTM	ERODE	MS1ST
Type	F	F	I	F	F	I	I	I
Default	-	0.9/0.67	0	0.0	0.0	0	0	0

(This card is optional).

Card 2 1 2 3 4 5 6 7 8

Variable	DT2MSF	DT2MSLC	IMSCL					
Type	F	I	I					
Default	not used	not used	0					

VARIABLE

DESCRIPTION

DTINIT	Initial time step size: EQ.0.0: LS-DYNA determines initial step size.
TSSFAC	Scale factor for computed time step (old name SCFT). See Remark 1 below. (Default = .90; if high explosives are used, the default is lowered to .67).
ISDO	Basis of time size calculation for 4-node shell elements. 3-node shells use the shortest altitude for options 0,1 and the shortest side for option 2. This option has no relevance to solid elements, which use a length based on the element volume divided by the largest surface area. EQ.0: characteristic length=area/(minimum of the longest side or the longest diagonal).

VARIABLE	DESCRIPTION
	<p>EQ.1: characteristic length=area/(longest diagonal).</p> <p>EQ.2: based on bar wave speed and MAX [shortest side, area/(minimum of the longest side or the longest diagonal)]. THIS LAST OPTION CAN GIVE A MUCH LARGER TIME STEP SIZE THAT CAN LEAD TO INSTABILITIES IN SOME APPLICATIONS, ESPECIALLY WHEN TRIANGULAR ELEMENTS ARE USED.</p> <p>EQ.3: time step size is based on the maximum eigenvalue. This option is okay for structural applications where the material sound speed changes slowly. The cost related to determining the maximum eigenvalue is significant, but the increase in the time step size often allows for significantly shorter run times without using mass scaling.</p>
TSLIMIT	<p>Shell element minimum time step assignment, TSLIMIT. When a shell controls the time step, element material properties (moduli <u>not</u> masses) will be modified such that the time step does not fall below the assigned step size. This option is applicable only to shell elements using material models: *MAT_PLASTIC_KINEMATIC, *MAT_POWER_LAW_PLASTICITY, *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY, *MAT_PIECEWISE_LINEAR_PLASTICITY. This so-called stiffness scaling option is NOT recommended. The DT2MS option below applies to all materials and element classes and is preferred. If both TSLIMIT and DT2MS below are active and if TSLIMIT is input as a positive number, then TSLIMIT is set to 1.E-18, which makes it inactive. If TSLIMIT is negative and less than DT2MS , then TSLIMIT is applied prior to the mass being scaled. If DT2MS exceeds the magnitude of TSLIMIT, then TSLIMIT is set to 1.E-18.</p>
DT2MS	<p>Time step size for mass scaled solutions, DT2MS. Positive values are for quasi-static analyses or time history analyses where the inertial effects are insignificant. Default = 0.0. If negative, TSSFAC* DT2MS is the minimum time step size permitted and mass scaling is done if and only if it is necessary to meet the Courant time step size criterion. This latter option can be used in transient analyses if the mass increases remain insignificant. See *CONTROL_TERMINATION variable name "ENDMAS". WARNING: Superelements, *ELEMENT_DIRECT_MATRIX_INPUT, are not mass scaled; consequently, DT2MS does not affect their time step size. In this case an error termination will occur, and DT2MS will need to be reset to a smaller value.</p>
LCTM	<p>Load curve ID that limits the maximum time step size (optional). This load curve defines the maximum time step size permitted versus time. If the solution time exceeds the final time value defined by the curve the computed step size is used. If the time step size from the load curve is exactly zero, the computed time step size is also used.</p>

VARIABLE	DESCRIPTION
ERODE	<p>Erosion flag for solid and t-shell elements when TSMIN (see *CONTROL_TERMINATION) is reached. If this flag is not set the calculation will terminate:</p> <p>EQ.0: no, EQ.1: yes.</p> <p>If ERODE=1, and TSMIN>0 (See *CONTROL_TERMINATION), all solid elements are checked at the beginning of element processing to check for negative volumes. The solid elements, which are found with negative volumes, are eroded and the calculation continues.</p>
MS1ST	<p>Limit mass scaling to the first step and fix the mass vector according to the time steps once. The time step will not be fixed but may drop during the calculation from the specified minimum:</p> <p>EQ.0: no, EQ.1: yes.</p>
DT2MSF	<p>Reduction (or scale) factor for initial time step size to determine the minimum time step size permitted. Mass scaling is done if it is necessary to meet the Courant time step size criterion. If this option is used DT2MS= -DT2MSF multiplied by the initial time step size, Δt, before Δt is scaled by TSSFAC. This option is active if and only if DT2MS=0 above.</p>
DT2MSLC	<p>Load curve specifying DT2MS as a function of time during the explicit solutions phase. The load curve can only be used for increasing the magnitude of DT2MS. Consequently, the magnitude of DT2MS is taken as the maximum of the current value and the value from the load curve.</p>
IMSCL	<p>Flag for selective mass scaling if and only if mass scaling active. Selective mass scaling does not scale the rigid body mass and is therefore more accurate. Since it is memory and CPU intensive, it should be applied only to small finely meshed parts. This option is available starting with the third revision of version 971.</p> <p>EQ.0: no selective mass scaling. EQ.1: all parts undergo selective mass scaling. LT.0: recommended. IMSCL is the part set ID of the parts that undergo selective mass scaling; all other parts are mass scaled the usual way.</p>

Remarks:

1. During the solution we loop through the elements and determine a new time step size by taking the minimum value over all elements.

$$\Delta t^{n+1} = TSSFAC \cdot \min\{\Delta t_1, \Delta t_2, \dots, \Delta t_N\}$$

where N is the number of elements. The time step size roughly corresponds to the transient time of an acoustic wave through an element using the shortest characteristic distance. For stability reasons the scale factor TSSFAC is typically set to a value of .90 (default) or some smaller value. To decrease solution time we desire to use the largest possible stable time step size. Values larger than .90 will often lead to instabilities. Some comments follow:

- The sound speed in steel and aluminum is approximately 5mm per microsecond; therefore, if a steel structure is modeled with element sizes of 5mm, the computed time step size would be 1 microsecond. Elements made from materials with lower sound speeds, such as foams, will give larger time step sizes. Avoid excessively small elements and be aware of the effect of rotational inertia on the time step size in the Belytschko beam element. Sound speeds differ for each material, for example, consider:

AIR	331 m/s
WATER	1478
STEEL	5240
TITANIUM	5220
PLEXIGLAS	2598

- Model stiff components with rigid bodies, not by scaling Young's modulus which can substantially reduce the time step size.
- The altitude of the triangular element should be used to compute the time step size. Using the shortest side is okay only if the calculation is closely examined for possible instabilities. This is controlled by parameter ISDO.

***CONTROL**

***CONTROL_TIMESTEP**

***DAMPING**

The Keyword options in this section in alphabetical order are:

***DAMPING_FREQUENCY_RANGE**

***DAMPING_GLOBAL**

***DAMPING_PART_MASS**

***DAMPING_PART_STIFFNESS**

***DAMPING_RELATIVE**

*DAMPING

*DAMPING_FREQUENCY_RANGE

*DAMPING_FREQUENCY_RANGE

Purpose: This feature provides approximately constant damping (i.e. frequency-independent) over a range of frequencies.

Card	1	2	3	4	5	6	7	8
Variable	CDAMP	FLOW	FHIGH	PSID				
Type	F	F	F	I				
Default	0.0	0.0	0.0	0				

VARIABLE	DESCRIPTION
CDAMP	Damping in fraction of critical.
FLOW	Lowest frequency in range of interest (cycles per unit time, e.g. Hz if time unit is seconds)
FHIGH	Highest frequency in range of interest (cycles per unit time, e.g. Hz if time unit is seconds)
PSID	Part set ID. The requested damping is applied only to the parts in the set. If PSID = 0, the damping is applied to all parts except those referred to by other *DAMPING_FREQUENCY_RANGE cards.

This feature provides approximately constant damping (i.e. frequency-independent) over a range of frequencies $F_{low} < F < F_{high}$. It is intended for small damping ratios (e.g. < 0.05) and frequency ranges such that F_{high}/F_{low} is in the range 10-300. The drawback to this method of damping is that it reduces the dynamic stiffness of the model, especially at low frequencies. This effect is predictable: the natural frequencies of modes close to F_{low} are reduced by 3% for a damping ratio of 0.01 and F_{high}/F_{low} in the range 10-30. Near F_{high} the error is between zero and one third of the error at F_{low} . Estimated frequency errors are shown in the table below.

Frequency	F_{high}/F_{low}			
		3 to 30	30 to 300	300 to 3000
Error at F_{low}				
Damping Ratio	0.01	3%	4.5%	6%
	0.02	6%	9%	12%
	0.04	12%	18%	24%

It is recommended that the elastic stiffnesses in the model be increased slightly to account for this, e.g. for 0.01 damping across a frequency range of 30 to 600Hz, the average error across the frequency range is about 2%. Increase the stiffness by $(1.02)^2$, i.e. by 4%.

*DAMPING

*DAMPING_GLOBAL

*DAMPING_GLOBAL

Purpose: Define mass weighted nodal damping that applies globally to the nodes of deformable bodies and to the mass center of the rigid bodies.

Card	1	2	3	4	5	6	7	8
Variable	LCID	VALDMP	STX	STY	STZ	SRX	SRY	SRZ
Type	I	F	F	F	F	F	F	F
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Remarks	1		2	2	2	2	2	2

VARIABLE

DESCRIPTION

LCID Load curve ID which specifies the system damping constant:
EQ.0: a constant damping factor as defined by VALDMP is used,
EQ.n: system damping is given by load curve n. The damping force applied to each node is $f=-d(t)mv$, where $d(t)$ is defined by load curve n.

VALDMP System damping constant, D_s (this option is bypassed if the load curve number defined above is non zero).

STX Scale factor on global x translational damping forces.

STY Scale factor on global y translational damping forces.

STZ Scale factor on global z translational damping forces.

SRX Scale factor on global x rotational damping moments.

SRY Scale factor on global y rotational damping moments.

SRZ Scale factor on global z rotational damping moments.

Remarks:

1. This keyword is also used for the restart, see *RESTART.
2. If STX=STY=STZ=SRX=SRY=SRZ=0.0 in the input above, all six values are defaulted to unity.

With mass proportional system damping the acceleration is computed as:

$$a^n = M^{-1} \left(P^n - F^n - F_{damp}^n \right)$$

where, M is the diagonal mass matrix, P^n is the external load vector, F^n is the internal load vector, and F_{damp}^n is the force vector due to system damping. This latter vector is defined as:

$$F_{damp}^n = D_s m v$$

The best damping constant for the system is usually some value approaching the critical damping factor for the lowest frequency mode of interest.

$$(D_s)_{critical} = 2\omega_{min}$$

The natural frequency ω_{min} (given in radians per unit time) is generally taken as the fundamental frequency of the structure. This frequency can be determined from an eigenvalue analysis or from an undamped transient analysis. Note that this damping applies to both translational and rotational degrees of freedom. Also note that mass proportional damping will damp rigid body motion as well as vibration.

Energy dissipated by through mass weighted damping is reported as system damping energy in the ASCII file GLSTAT. This energy is computed whenever system damping is active.

*DAMPING

*DAMPING_PART_MASS

*DAMPING_PART_MASS

Purpose: Define mass weighted damping by part ID. Parts may be either rigid or deformable. In rigid bodies the damping forces and moments act at the center of mass.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	LCID	SF	FLAG				
Type	I	I	F	I				
Default	0	0	1.0	0				

(This card is optional and is read if and only if FLAG=1. If this card is not read STX, STY, STZ, SRX, SRY, and SRZ default to unity.)

Card 2 1 2 3 4 5 6 7 8

Variable	STX	STY	STZ	SRX	SRY	SRZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE

DESCRIPTION

PID	Part ID, see *PART.
LCID	Load curve ID which specifies system damping for parts.
SF	Scale factor for load curve. This allows a simple modification of the load curve values.
FLAG	Set this flag to unity if the global components of the damping forces require separate scale factors.
STX	Scale factor on global x translational damping forces.
STY	Scale factor on global y translational damping forces.
STZ	Scale factor on global z translational damping forces.

VARIABLE	DESCRIPTION
SRX	Scale factor on global x rotational damping moments.
SRY	Scale factor on global y rotational damping moments.
SRZ	Scale factor on global z rotational damping moments.

Remarks:

Mass weighted damping damps all motions including rigid body motions. For high frequency oscillatory motion stiffness weighted damping may be preferred. With mass proportional system damping the acceleration is computed as:

$$a^n = M^{-1} (P^n - F^n - F_{damp}^n)$$

where, M is the diagonal mass matrix, P^n is the external load vector, F^n is the internal load vector, and F_{damp}^n is the force vector due to system damping. This latter vector is defined as:

$$F_{damp}^n = D_s m v$$

The best damping constant for the system is usually based on the critical damping factor for the lowest frequency mode of interest. Therefore,

$$D_s = 2 \omega_{min}$$

is recommended where the natural frequency (given in radians per unit time) is generally taken as the fundamental frequency of the structure. The damping is applied to both translational and rotational degrees of freedom. The component scale factors can be used to limit which global components see damping forces.

Energy dissipated by through mass weighted damping is reported as system damping energy in the ASCII file GLSTAT. This energy is computed whenever system damping is active.

*DAMPING

*DAMPING_PART_STIFFNESS

*DAMPING_PART_STIFFNESS

Purpose: Assign Rayleigh stiffness damping coefficient by part ID.

Card	1	2	3	4	5	6	7	8
Variable	PID	COEF						
Type	I	F						
Default	none	0.0						

VARIABLE

DESCRIPTION

PID	Part ID, see *PART.
COEF	Rayleigh damping coefficient. Two methods are now available: LT.0.0: Rayleigh damping coefficient is set based on a given frequency and applied uniformly to each element in the part ID. This approach is used in versions of LS-DYNA prior to version 960. See notes below. EQ.0.0: Inactive. GT.0.0: Rayleigh damping coefficient for stiffness weighted damping. Values between 0.01 and 0.25 are recommended. Higher values are strongly discouraged, and values less than 0.01 may have little effect. The damping coefficient is uniquely defined for each element of the part ID.

Remarks:

The damping matrix in Rayleigh damping is defined as:

$$C = \alpha M + \beta K$$

where C, M, and K are the damping, mass, and stiffness matrices, respectively. The constants α and β are the mass and stiffness proportional damping constants. The mass proportional damping can be treated by system damping, see keywords: *DAMPING_GLOBAL and DAMPING_PART_MASS. Transforming C with the *i*th eigenvector ϕ_i gives:

$$\phi_i^T C \phi_i = \phi_i^T (\alpha M + \beta K) \phi_i = \alpha + \beta \omega_i^2 = 2\omega_i \xi_i \delta_{ij}$$

where ω_i is the *i*th frequency (radians/unit time) and ξ_i is the corresponding modal damping parameter.

Generally, the stiffness proportional damping is effective for high frequencies and is orthogonal to rigid body motion. Mass proportional damping is more effective for low frequencies and will damp rigid body motion. If a large value of the stiffness based damping coefficient is used, it may be necessary to lower the time step size significantly. This must be done manually by reducing the time step scale factor on the *CONTROL_TIMESTEP control card. Since a good value of β is not easily identified, the coefficient, COEF, is defined such that a value of .10 roughly corresponds to 10% damping in the high frequency domain.

In versions prior to 960, one damping coefficient is defined that applies to all elements of the entire part. With this older approach if 10% of critical damping is sought in the i th mode then set:

$$\beta = \frac{.20}{\omega_i}$$

and input β as a negative number. Typically, β is some fraction of the time step size.

Energy dissipated by Rayleigh damping is computed if and only if the flag, RYLEN, on the control card, *CONTROL_ENERGY is set to 2. This energy is accumulated as element internal energy and is included in the energy balance. In the GLSTAT file this energy will be lumped in with the internal energy.

*DAMPING

*DAMPING_RELATIVE

*DAMPING_RELATIVE

Purpose: Apply damping relative to the motion of a rigid body.

Card	1	2	3	4	5	6	7	8
Variable	CDAMP	FREQ	PIDRB	PSID				
Type	F	F	F	I				
Default	0	0	0	0				

VARIABLE

DESCRIPTION

CDAMP	Fraction of critical damping.
FREQ	Frequency at which CDAMP is to apply (cycles per unit time, e.g. Hz if time unit is seconds).
PIDRB	Part ID of rigid body, see *PART. Motion relative to this rigid body will be damped.
PSID	Part set ID. The requested damping is applied only to the parts in the set.

Remarks:

1. This feature provides damping of vibrations for objects that are moving through space. The vibrations are damped, but not the rigid body motion. This is achieved by calculating the velocity of each node relative to that of a rigid body, and applying a damping force proportional to that velocity. The forces are reacted onto the rigid body such that overall momentum is conserved. It is intended that the rigid body is embedded within the moving object.
2. Vibrations at frequencies below FREQ are damped by more than CDAMP, while those at frequencies above FREQ are damped by less than CDAMP. It is recommended that FREQ be set to the frequency of the lowest mode of vibration.

***DATABASE**

The database definitions are optional, but are necessary to obtain output files containing results information. In this section the database keywords are defined in alphabetical order:

- *DATABASE_OPTION**
- *DATABASE_ADAMS**
- *DATABASE_BINARY_OPTION**
- *DATABASE_CROSS_SECTION_OPTION1_{OPTION2}**
- *DATABASE_EXTENT_OPTION**
- *DATABASE_FORMAT**
- *DATABASE_FSI**
- *DATABASE_FSI_SENSOR**
- *DATABASE_HISTORY_OPTION**
- *DATABASE_NODAL_FORCE_GROUP**
- *DATABASE_SPRING_FORWARD**
- *DATABASE_SUPERPLASTIC_FORMING**
- *DATABASE_TRACER**

The ordering of the database definition cards in the input file is completely arbitrary.

***DATABASE_OPTION**

Options for ASCII files include (if the file is not specified it will not be created):

ABSTAT	Airbag statistics.
AVSFLT	AVS database. See <i>*DATABASE_EXTENT_OPTION</i> .
BNDOUT	Boundary condition forces and energy
DEFGEO	Deformed geometry file. (Note that to output this file in Chrysler format insert the following line in your <i>.cshrc</i> file: “setenv LSTC_DEFGEO chrysler”) The NASBDF file (NASTRAN Bulk Data) is created whenever the DEFGEO file is requested.
DCFAIL	Failure function data for <i>*MAT_SPOTWELD_DAIMLERCHRYSLER</i>
DEFORC	Discrete elements.
ELOUT	Element data. See <i>*DATABASE_HISTORY_OPTION</i> .
GCEOUT	Geometric contact entities.
GLSTAT	Global data. Always obtained if SSSTAT file is activated.
H3OUT	HybridIII rigid body dummies.
JNTFORC	Joint force file
MATSUM	Material energies. See Remarks 1 and 2 below.
MOVIE	MOVIE. See <i>*DATABASE_EXTENT_OPTION</i> .
MPGS	MPGS. See <i>*DATABASE_EXTENT_OPTION</i> .
NCFORC	Nodal interface forces. See <i>*CONTACT</i> - Card 1 (SPR and MPR)
NODFOR	Nodal force groups. See <i>*DATABASE_NODAL_FORCE_GROUP</i> .
NODOUT(HF)	Nodal point data. See <i>*DATABASE_HISTORY_OPTION</i> .
RBDOUT	Rigid body data. See Remark 2 below.
RCFORC	Resultant interface forces.
RWFORC	Wall forces.
SBTOUT	Seat belt output file
SECFORC	Cross section forces. See <i>*DATABASE_CROSS_SECTION_OPTION</i> .
SLEOUT	Sliding interface energy. See <i>*CONTROL_ENERGY</i>
SPCFORC	SPC reaction forces.
SPHOUT	SPH data. See <i>*DATABASE_HISTORY_OPTION</i> .
SSSTAT	Subsystem data. See <i>*DATABASE_EXTENT_SSSTAT</i> .
SWFORC	Nodal constraint reaction forces (spot welds and rivets).
TPRINT	Thermal output from a coupled structural/thermal or thermal only analysis.
TRHIST	Tracer particle history information. See <i>*DATABASE_TRACER</i> .

To include global and subsystem mass and inertial properties in the GLSTAT and SSSTAT files add the option *_MASS_PROPERTIES* as show below. If this option is active the current mass and inertia properties are output including the principle inertias and their axes. Mass of deleted nodes and rigid bodies are not included in the calculated properties.

GLSTAT_MASS_PROPERTIES	This is an option for the glstat file to include mass and inertial properties.
SSSTAT_MASS_PROPERTIES	This is an option for the ssstat file to include mass and inertial properties for the subsystems.

Card	1	2	3	4	5	6	7	8
Variable	DT	BINARY	LCUR	IOOPT	DTHFF	BINHF		
Type	F	I	I	I	F	I		
Default	0.	1 or 2	none	0.	0.	1 or 2		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DT	Time interval between outputs. If DT is zero, no output is printed.
BINARY	<p>Flag for binary file</p> <p>EQ.1: ASCII file is written. This is the default on serial and shared memory computers.</p> <p>EQ.2: Data written to a binary database, which contains data that would otherwise be output to the ASCII file. The ASCII file in this case is not created. This is the default on distributed memory computers.</p> <p>EQ.3: ASCII file is written and the data is also written to the binary database (NOTE: this option is only valid for serial and shared memory computers – distributed memory computers will only produce the binary database).</p>
LCUR	Optional load curve ID specifying time interval between dumps.
IOOPT	<p>Flag to govern behavior of the plot frequency load curve defined by LCUR:</p> <p>EQ.1: At the time each plot is generated, the load curve value is added to the current time to determine the next plot time.(this is the default behavior)</p> <p>EQ.2: At the time each plot is generated, the next plot time T is computed so that T = the current time plus the load curve value at time T.</p> <p>EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.</p>
DTHF	Optional input for the NODOUT file option only. Time interval between outputs for the high frequency file, NODOUTHF. If DTHF is zero, no output is printed. Nodal points that are to be output at a higher frequency are flagged in the DATABASE_HISTORY input.
BINHF	Optional input for the NODOUTHF file only. Flag for binary file for the high frequency NODOUTHF file. See BINARY above.

The file names and corresponding unit numbers are:

	<u>I/O UNIT #</u>	<u>FILE NAME</u>
Airbag statistics	i/o unit #43	ABSTAT
ASCII database	i/o unit #44	AVSFLT
Boundary conditions	i/o unit #46	BNDOUT (nodal forces and energies)
Smug animator database	i/o unit#40	DEFGEO
Discrete elements	i/o unit#36	DEFORC
Element data	i/o unit#34	ELOUT
Contact entities	i/o unit #48	GCEOUT
Global data	i/o unit#35	GLSTAT
Joint forces	i/o unit #53	JNTFORC
Material energies	i/o unit#37	MATSUM
MOVIE file family	i/o unit #50	MOVIE _{nnn} .xxx where _{nnn} =001-999
MPGS file family	i/o unit #50	MPGS _{nnn} .xxx where _{nnn} =001-999
Nastran/BDF file	i/o unit#49	NASBDF (see comment below)
Nodal interface forces	i/o unit#38	NCFORC
Nodal force group	i/o unit #45	NODFOR
Nodal point data	i/o unit#33	NODOUT
Rigid body data	i/o unit #47	RBDOUT
Resultant interface forces	i/o unit#39	RCFORC
Rigidwall forces	i/o unit#32	RWFORC
Seat belts	i/o unit #52	SBTOUT
Cross-section forces	i/o unit#31	SECFORC
Interface energies	i/o unit #51	SLEOUT
SPC reaction forces	i/o unit#41	SPCFORC
SPH element data	i/o unit#68	SPHOUT
Subsystems statistics	i/o unit#58	SSSTAT
Nodal constraint resultants	i/o unit #42	SWFORC (spot welds/rivets)
Thermal output	i/o unit #73	TPRINT
Tracer particles	i/o unit #70	TRHIST

Output Components for ASCII Files

ABSTAT	BNDOUT	DCFAIL	DEFORC
volume	x, y, z force	failure function	x, y, z force
pressure		normal term	
internal energy		bending term	
input mass flow rate		shear term	
output mass flow rate		weld area	
mass		effective strain rate	
temperature			
density			

ELOUT			
Beam	Stress Shell	Brick	Strain Shell
axial force resultant	xx, yy, zz stress	xx, yy, zz stress	xx, yy, zz strain
s shear resultant	xy, yz, zx stress	xy, yz, zx stress	xy, yz, zx strain
t shear resultant	plastic strain	effective stress	lower surface strain
s moment resultant		yield function	upper surface strain
t moment resultant			
torsional resultant			

GCEOUT	
x, y, z force	x, y, z moment

GLSTAT	
time step	total energy
kinetic energy	external work
internal energy	total energy / initial energy
spring & damper energy	energy ratio w/o eroded energy
hourglass energy	element id controlling time step
system damping energy	global x, y, z velocity
sliding interface energy	time per zone cycle
eroded kinetic energy	joint internal energy
eroded internal energy	stonewall energy
eroded hourglass energy	rigid body stopper energy

JNTFORC	
x, y, z force	x, y, z moment

MATSUM	
kinetic energy	x, y, z rigid body velocity
internal energy	eroded internal energy
hourglass energy	eroded kinetic energy
x, y, z momentum	added mass

NCFORC	NODOUT	NODFOR
x force	x, y, z displacement	x, y, z force
y force	x, y, z velocity	
z force	x, y, z acceleration	
	x, y, z rotation	
	x, y, z rotational velocity	
	x, y, z rotational acceleration	

RBDOUT	RCFORC	RWFORC
x, y, z displacement	x, y, z force	normal
x, y, z velocity		x, y, z force
x, y, z acceleration		

SECFORC	SLEOUT	SPCFORC	SWFORC
x, y, z force	slave energy	x, y, z force	axial force
x, y, z moment	master energy	x, y, z moment	shear force
x, y, z center	frictional energy		
area			
resultant force			

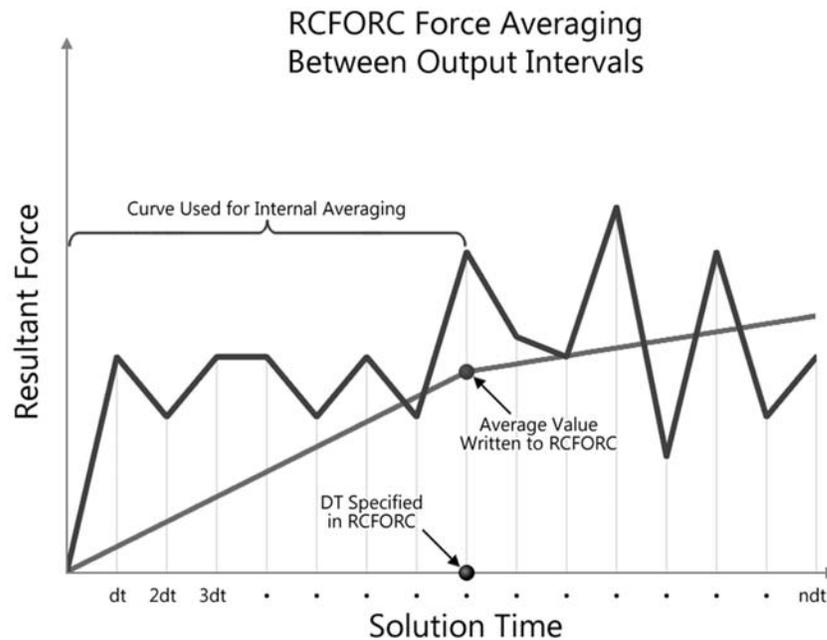
Remarks:

1. The kinetic energy quantities in the MATSUM and GLSTAT files may differ slightly in values for several reasons. First, the energy associated with added mass (from mass-scaling) is included in the GLSTAT calculation, but is not included in MATSUM. Secondly, the energies are computed element by element in MATSUM for the deformable materials and, consequently, nodes which are merged with rigid bodies will also have their kinetic energy included in the rigid body total. Furthermore, kinetic energy is computed from nodal velocities in GLSTAT and from element midpoint velocities in MATSUM.
2. The PRINT option in the part definition allows some control over the extent of the data that is written into the MATSUM and RBDOUT files. If the print option is used the variable PRBF can be defined such that the following numbers take on the meanings:
EQ.0: default is taken from the keyword *CONTROL_OUTPUT,

- EQ.1: write data into RBDOUT file only,
- EQ.2: write data into MATSUM file only,
- EQ.3: do not write data into RBDOUT and MATSUM.

Also see CONTROL_OUTPUT and PART_PRINT.

- 3. This keyword is also used in the restart phase, see *RESTART. Thus, the output interval can be changed when restarting.
- 4. All information in the files except in AVSFLT, MOVIE, AND MPGS can also be plotted using the post-processor LS-PREPOST. Arbitrary cross plotting of results between ASCII files is easily handled.
- 5. Resultant contact forces reported in RCFORC are averaged over the preceding output interval.



- 6. “Spring and damper energy” reported in GLSTAT is a subset of “Internal energy”. The “Spring and damper energy” includes internal energy of discrete elements, seatbelt elements, and that associated with joint stiffness (see *CONSTRAINED_JOINT_STIFFNESS_...).

***DATABASE_ADAMS**

Purpose: Request output of an MDI Modal Neutral File for later use in the ADAMS software.

Card 1 2 3 4 5 6 7 8

Variable	IFLAG	M_UNITS	L_UNITS	T_UNITS				
Type	I	F	F	F				
Default	0	none	none	none				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IFLAG	Flag controlling write of modal neutral file after eigenvalue analysis EQ.0: do not write (default), EQ.1: write to file "d3mnf"
M_UNITS	Mass units of measure used in this model. EQ.-1: kilogram EQ.-2: gram EQ.-3: megagram (metric ton) EQ.-4: lbf*sec**2/in (psi-compatible) EQ.-5: slug EQ.-6: pound-mass
L_UNITS	Length units of measure used in this model. EQ.-1: meter EQ.-2: centimeter EQ.-3: millimeter EQ.-4: inch EQ.-5: foot
T_UNITS	Time units of measure used in this model. EQ.-1: second EQ.-2: millisecond EQ.-3: minute EQ.-4: hour

Remarks:

1. This option is not available for every platform. Check LS-DYNA Banner upon execution of the program to see if this feature is enabled.
2. Models must be created using a combination of the above units.

***DATABASE_BINARY_OPTION**

Options for binary output files with the default names given include:

D3DRLF	Dynamic relaxation database.
D3DUMP	Binary output restart files. Define output frequency in cycles.
D3PART	Dt for partial output states See also *DATABASE_EXTENT_BINARY.
D3PLOT	Dt for complete output states. See also *DATABASE_EXTENT_BINARY.
D3PROP	Output property data.
D3THDT	Dt for time history data of element subsets. See *DATABASE_HISTORY.
FSIFOR	ALE interface force database (please see Remark 1).
RUNRSF	Binary output restart file. Define output frequency in cycles.
INTFOR	Dt for output of contact interface data (file name must be given on the execution line using "S="). Also see *CONTACT variables mpr and spr.
XTFILE	Flag to specify output of extra time history data to XTFILE at same time as D3THDT file. The following card is left blank for this option.
D3CRACK	Dt for output of crack data file for the Winfrith concrete model (file name must be given on the execution line using "q="). This file can be used with the D3PLOT file to show crack formation of the deformed concrete materials.

The D3DUMP and the RUNRSF options create complete databases which are necessary for restarts, see *RESTART. When RUNRSF is specified, the same file is overwritten after each interval, an option allows a series of files to be overwritten in a cyclic order. When D3DUMP is specified, a new restart file is created after each interval, thus a "family" of files is created numbered sequentially D3DUMP01, D3DUMP02, etc. The default file names are RUNRSF and D3DUMP unless other names are specified on the execution line, see the INTRODUCTION, EXECUTION SYNTAX. Since all data held in memory is written into the restart files, these files can be quite large and care should be taken with the D3DUMP files not to create too many. If *DATABASE_BINARY_D3PLOT is not specified in the keyword deck then a complete output state will be written ever time step.

The D3PLOT, D3PART, D3DRLF, and the INTFOR files contain plotting information to plot data over the three dimensional geometry of the model. These databases can be plotted with LS-PREPOST. The D3THDT file contains time history data for element subsets as well as global information, see *DATABASE_HISTORY. This data can be plotted with LS-PREPOST. The default names for the D3PLOT, D3PART, D3DRLF, and the D3THDT files are D3PLOT, D3PART, D3DRLF, and D3THDT. For INTFOR a unique name must be specified on the execution line with S=iff, (iff=file name), for FSIFOR a unique name must be specified on the execution line with h=iff, (iff=file name), see the INTRODUCTION, EXECUTION SYNTAX. The file structure is such that each file contains the full geometry at the beginning, followed by the analysis generated output data at the specified time intervals. For the contents of the D3PLOT, D3PART and D3THDT files see also the *DATABASE_EXTENT_BINARY definition. It is possible to severely restrict the information that is dumped and consequently reduce the size of the databases. The contents of the D3THDT file are also specified with the *DATABASE_HISTORY definition. It should also be noted in particular that the databases can be considerably reduced for models with rigid bodies containing many elements.

DATABASE**DATABASE_BINARY**

Card 1 2 3 4 5 6 7 8

Variable	DT/CYCL	LCDT/NR	BEAM	NPLTC	PSETID			
Type	F	I	I	I	I			
Default	-	-	-	-	-			
Remarks								

Optional Card that only applies to the D3PLOT database

Card 1 2 3 4 5 6 7 8

Variable	IOOPT							
Type	I							
Default	0							
Remarks								

Use only for D3PROP option (No other cards are necessary)

Card 1 2 3 4 5 6 7 8

Variable	IFILE	IMATL	IWALL					
Type	I	I	I					
Default	1	0	0					

VARIABLE	DESCRIPTION
DT	Time interval between outputs.
CYCL	Output interval in time steps (a time step is a cycle). For the D3DRFL file a positive number 'n' will cause plot dumps to be written at every n'th convergence check interval specified on the *CONTROL_DYNAMIC_RELAXATION card.
NR	Number of Running Restart Files, RUNRSF, written in a cyclical fashion. The default number is one, i.e. the same file is overwritten each time.
LCDT	Optional load curve ID specifying time interval between dumps. This option is only available for the D3PLOT, D3PART, D3THDT and INTFOR files.
BEAM	Option flag for *DATABASE_BINARY_D3PLOT or D3PART. EQ.0: Discrete spring and damper elements are added to the D3PLOT or D3PART database where they are display as beam elements. The element global X, global Y, global Z and resultant forces are written to the database, EQ.1: No discrete spring and damper elements are added to the D3PLOT or D3PART database. This option is useful when translating old LS-DYNA input decks to KEYWORD input. In older input decks there is no requirement that beam and spring elements have unique ID's, and beam elements may be created for the spring and dampers with identical ID's to existing beam elements causing a fatal error. Contact interfaces which are based on part IDs of seatbelt elements will not be properly generated if this option is used. EQ.2: Discrete spring and damper elements are added to the D3PLOT or D3PART database where they are displayed as beam elements (similar to option 0). In this option the element resultant force is written to its first database position allowing beam axial forces and spring resultant forces to be plotted at the same time. This can be useful during some post-processing applications.
NPLTC	$DT=ENDTIME/NPLTC$ applies to D3PLOT and D3PART only. This overrides the DT specified in the first field.
PSETID	SET_PART ID for D3PART only.
IOOPT	This option applies to the D3PLOT file only. Flag to govern behavior of the plot frequency load curve defined by LCDT: EQ.1: At the time each plot is generated, the load curve value is added to the current time to determine the next plot time (this is the default behavior).

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.2: At the time each plot is generated, the next plot time T is computed so that T = the current time plus the load curve value at time T. EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.
IFILE	Specify file for D3PROP output. (This can also be defined on the command line by adding d3prop = 1 or d3prop = 2 which also sets IMATL = IWALL = 1) EQ.1: Output data at the end of the first d3plot file. EQ.2: Output data to the file d3prop.
IMATL	Output *EOS, *HOURGLASS, *MAT, *PART and *SECTION data. EQ.0: No EQ.1: Yes
IWALL	Output *RIGIDWALL data. EQ.0: No EQ.1: Yes

Remarks:

1. When *DATABASE_FSI is defined, a few pieces of coupling information of some Lagrangian surface entities interacting with the ALE materials may be output as history parameters into a file called "dbfsi". Coupling pressure is one of the output variables. This coupling pressure is averaged over each surface entity. To obtain coupling pressure contour variations over each segment, use *DATABASE_BINARY_FSIFOR. To use it, three things must be done:

- 1) The INTFORC parameter (*CONSTRAINED_LAGRANGE_IN_SOLID, 4th row, 3rd column) must be turned ON (INTFORC=1).
- 2) A *DATABASE_BINARY_FSIFOR card is defined controlling the output interval. The time interval between output is defined by the parameter DT in this card.
- 3) This interface force file is activated by executing ls970 as follow:

```
ls970 i=inputfilename.k ... h=interfaceforcefilename
```

LSDYNA will then writes out the segment coupling pressure and forces to a binary interface force file for contour plotting over the whole simulation interval.

To plot the binary data in this file, type: lsprepost interfaceforcefilename.

For example, when all 3 of the above actions are taken, and let's assume we define the interfaceforcefilename =fsifor → a series of "fsifor##" binary files are output for contour plotting. To plot this, type "lsprepost fsifor" (without the double quotes).

Card (1 of 2) for the PLANE option

Card	1	2	3	4	5	6	7	8
Variable	PSID	XCT	YCT	ZCT	XCH	YCH	ZCH	RADIUS
Type	I	F	F	F	F	F	F	F
Default	0	0.	0.	0.	0.	0.	0.	0.

Card (2 of 2) for the PLANE option

Card	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM	ID	ITYPE	
Type	F	F	F	F	F	I	I	
Default	0.	0.	0.	infinity	infinity	global	0	

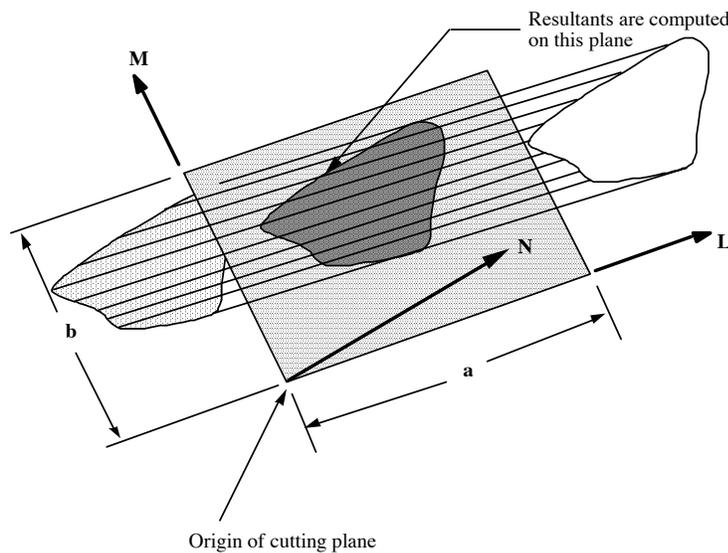


Figure 10.1. Definition of cutting plane for automatic definition of interface for cross-sectional forces. The automatic definition does not check for springs and dampers in the section. For best results the cutting plane should cleanly pass through the middle of the elements, distributing them equally on either side. Elements that intersect the edges of the cutting plane are deleted from the cross-section.

The set option requires that the equivalent of the automatically generated input via the cutting plane be identified manually and defined in sets. All nodes in the cross-section and their related elements that contribute to the cross-sectional force resultants should be defined.

Card (1 of 1) for the SET option

Card 1 2 3 4 5 6 7 8

Variable	NSID	HSID	BSID	SSID	TSID	DSID	ID	ITYPE
Type	I	I	I	I	I	I	I	I
Default	required	0	0	0	0	0	global	0

VARIABLE**DESCRIPTION**

CSID	Optional ID for cross section. If not specified cross section ID is taken to be the cross section order in the input deck.
PSID	Part set ID. If zero all parts are included.
XCT	x-coordinate of tail of any outward drawn normal vector, N , originating on wall (tail) and terminating in space (head), see Figure 10.1.
YCT	y-coordinate of tail of normal vector, N .
ZCT	z-coordinate of tail of normal vector, N .
XCH	x-coordinate of head of normal vector, N .
YCH	y-coordinate of head of normal vector, N .
ZCH	z-coordinate of head of normal vector, N .
RADIUS	Optional radius. If RADIUS > 0., a circular cut plane centered at (XCT, YCT, ZCT) of radius=RADIUS, with the normal vector originating at (XCT, YCT, ZCT) and pointing towards (XCH, YCH, ZCH) will be created. In this case the variables XHEV, YHEV, ZHEV, LENL, and LENM, which are defined on the 2 nd card will be ignored.
XHEV	x-coordinate of head of edge vector, L .
YHEV	y-coordinate of head of edge vector, L .
ZHEV	z-coordinate of head of edge vector, L .

VARIABLE	DESCRIPTION
LENL	Length of edge a, in L direction.
LENM	Length of edge b, in M direction.
NSID	Nodal set ID, see *SET_NODE_OPTION.
HSID	Solid element set ID, see *SET_SOLID.
BSID	Beam element set ID, see *SET_BEAM.
SSID	Shell element set ID, see *SET_SHELL_OPTION.
TSID	Thick shell element set ID, see *SET_TSHELL.
DSID	Discrete element set ID, see *SET_DISCRETE.
ID	Rigid body (see *MAT_RIGID, type 20), accelerometer ID (see *ELEMENT_SEATBELT_ACCELEROMETER) or coordinate ID, see *DEFINE_COORDINATE_NODES. The force resultants are output in the <u>updated</u> local system of the rigid body, accelerometer, or coordinate system.
ITYPE	Flag for local system type: EQ.0: rigid body, EQ.1: accelerometer, EQ.2: coordinate ID.

***DATABASE_EXTENT_OPTION**

Available options include:

- AVS**
- BINARY**
- MOVIE**
- MPGS**
- SSSTAT**

Purpose: Specify output database to be written. Binary applies to the data written to the D3PLOT, D3PART, and D3THDT files. See *DATABASE_BINARY_OPTION.

For the AVS, MPGS, and MOVIE options the following cards apply:

Define as many cards as necessary. The created MPGS and MOVIE databases consist of a geometry file and one file for each output database.

Card	1	2	3	4	5	6	7	8
------	---	---	---	---	---	---	---	---

Variable	VTYPE	COMP						
Type	I	I						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
VTYPE	Variable type: EQ.0: node, EQ.1: brick, EQ.2: beam, EQ.3: shell, EQ.4: thick shell.
COMP	Component ID. For the corresponding VTYPE, integer components from the following tables can be chosen: VTYPE.EQ.0: Table 10.1, VTYPE.EQ.1: Table 10.2, VTYPE.EQ.2: not supported, VTYPE.EQ.3: Table 10.3, VTYPE.EQ.4: not supported.

Remarks:

The AVS database consists of a title card, then a control card defining the number of nodes, brick-like elements, beam elements, shell elements, and the number of nodal vectors, NV, written for each output interval. The next NV lines consist of character strings that describe the nodal vectors. Nodal coordinates and element connectivity follow. For each state the solution time is written, followed by the data requested below. The last word in the file is the number of states. We recommend creating this file and examining its contents, since the organization is relatively transparent. The MOVIE and MPGS database are widely used and will be familiar with users who are currently using these databases.

Table 10.1. Nodal Quantities

Component ID	Quantity
1	x, y, z-displacements
2	x, y, z-velocities
3	x, y, z-accelerations

Table 10.2. Brick Element Quantities

Component ID	Quantity
1	x-stress
2	y-stress
3	z-stress
4	xy-stress
5	yz-stress
6	zx-stress
7	effective plastic strain

Table 10.3. Shell and Thick Shell Element Quantities

Component ID	Quantity
1	midsurface x-stress
2	midsurface y-stress
3	midsurface z-stress
4	midsurface xy-stress
5	midsurface yz-stress
6	midsurface xz-stress
7	midsurface effective plastic strain
8	inner surface x-stress
9	inner surface y-stress
10	inner surface z-stress
11	inner surface xy-stress
12	inner surface yz-stress
13	inner surface zx-stress
14	inner surface effective plastic strain
15	outer surface x-stress
16	outer surface y-stress

Table 10.3. Shell and Thick Shell Element Quantities (cont.).

Component ID	Quantity
17	outer surface z-stress
18	outer surface xy-stress
19	outer surface yz-stress
20	outer surface zx-stress
21	outer surface effective plastic strain
22	bending moment-mxx (4-node shell)
23	bending moment-myy (4-node shell)
24	bending moment-mxy (4-node shell)
25	shear resultant-qxx (4-node shell)
26	shear resultant-qyy (4-node shell)
27	normal resultant-nxx (4-node shell)
28	normal resultant-nyy (4-node shell)
29	normal resultant-nzz (4-node shell)
30	thickness (4-node shell)
31	element dependent variable
32	element dependent variable
33	inner surface x-strain
34	inner surface y-strain
35	inner surface z-strain
36	inner surface xy-strain
37	inner surface yz-strain
38	inner surface zx-strain
39	outer surface x-strain
40	outer surface y-strain
41	outer surface z-strain
42	outer surface xy-strain
43	outer surface yz-strain
44	outer surface zx-strain
45	internal energy
46	midsurface effective stress
47	inner surface effective stress
48	outer surface effective stress
49	midsurface max. principal strain
50	through thickness strain
51	midsurface min. principal strain
52	lower surface effective strain
53	lower surface max. principal strain
54	through thickness strain
55	lower surface min. principal strain
56	lower surface effective strain
57	upper surface max. principal strain
58	through thickness strain
59	upper surface min. principal strain
60	upper surface effective strain

Table 10.4. Beam Element Quantities

Component ID	Quantity
1	x-force resultant
2	y-force resultant
3	z-force resultant
4	x-moment resultant
5	y-moment resultant
6	z-moment resultant

For the BINARY option the following cards apply (Card 3 is optional):

Card 1 1 2 3 4 5 6 7 8

Variable	NEIPH	NEIPS	MAXINT	STRFLG	SIGFLG	EPSFLG	RLTFLG	ENGFLG
Type	I	I	I	I	I	I	I	I
Default	0	0	3	0	1	1	1	1
Remarks			1					

Card 2

Variable	CMPFLG	IEVERP	BEAMIP	DCOMP	SHGE	STSSZ	N3THDT	IALEMAT
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	2	1
Remarks			2					

Card 3

Variable	NINTSLD	PKP_SEN	SCLP		MSSCL	THERM		
Type	I	I	F		I	I		
Default	1	0	1.0		0	0		
Remarks								

VARIABLE	DESCRIPTION
NEIPH	Number of additional integration point history variables written to the binary database for solid elements. The integration point data is written in the same order that it is stored in memory-each material model has its own history variables that are stored. For user defined materials it is important to store the history data that is needed for plotting before the data which is not of interest.
NEIPS	Number of additional integration point history variables written to the binary database for both shell and thick shell elements for each integration point, see NEIPH above.
MAXINT	Number of shell integration points written to the binary database, see also *INTEGRATION_SHELL. If the default value of 3 is used then results are output for the outermost (top) and innermost (bottom) integration points together with results for the neutral axis. If MAXINT is set to 3 and the element has 1 integration point then all three results will be the same. If a value other than 3 is used then results for the first MAXINT integration points in the element will be output. Note: If the element has an even number of integration points and MAXINT is not set to 3 then you will not get mid-surface results. See Remarks below.
STRFLG	Set to 1 to dump strain tensors for solid, shell and thick shell elements for plotting by LS-PREPOST and ASCII file ELOUT. For shell and thick shell elements two tensors are written, one at the innermost and one at the outermost integration point. For solid elements a single strain tensor is written.
SIGFLG	Flag for including stress tensor in the shell LS-DYNA database: EQ.1: include (default), EQ.2: exclude.
EPSFLG	Flag for including the effective plastic strains in the shell LS-DYNA database: EQ.1: include (default), EQ.2: exclude.
RLTFLG	Flag for including stress resultants in the shell LS-DYNA database: EQ.1: include (default), EQ.2: exclude.
ENGFLG	Flag for including shell internal energy density and thickness in the LS-DYNA database: EQ.1: include (default), EQ.2: exclude.

VARIABLE	DESCRIPTION
CMPFLG	Orthotropic and anisotropic material stress and strain output in local material coordinate system for solids, shells and thick shells. EQ.0: global, EQ.1: local.
IEVERP	Every plot state for “d3plot” database is written to a separate file. This option will limit the database to 1000 states: EQ.0: more than one state can be on each plotfile, EQ.1: one state only on each plotfile.
BEAMIP	Number of beam integration points for output. This option does not apply to beams that use a resultant formulation.
DCOMP	Data compression to eliminate rigid body data: EQ.1: off (default), no rigid body data compression, EQ.2: on, rigid body data compression active, EQ.3: off, no rigid body data compression, but nodal velocities and accelerations are eliminated from the database. EQ.4: on, rigid body data compression active and nodal velocities and accelerations are eliminated from the database.
SHGE	Output shell hourglass energy density: EQ.1: off (default), no hourglass energy written, EQ.2: on.
STSSZ	Output shell element time step, mass, or added mass: EQ.1: off (default), EQ.2: output time step size, EQ.3: output mass, added mass, or time step size. See remark 3 below.
N3THDT	Material energy write option for D3THDT database EQ.1: off, energy is NOT written to D3THDT database, EQ.2: on (default), energy is written to D3THDT database.
IALEMAT	Output solid part ID list containing ale materials. EQ.1: on (default)
NINTSLD	Number of solid element integration points written to the LS-DYNA database. The default value is 1. For solids with multiple integration points NINTSLD may be set to 8. Currently, no other values for NINTSLD are allowed. For solids with multiple integration points, an average value is output if NINTSLD is set to 1.

VARIABLE	DESCRIPTION
PKP_SEN	<p>Flag to output the peak pressure and surface energy computed by each contact interface into the interface force database. To obtain the surface energy, FRCENG, must be sent to 1 on the control contact card. When PKP_SEN=1, it is possible to identify the energies generated on the upper and lower shell surfaces, which is important in metal forming applications. This data is mapped after each H-adaptive remeshing.</p> <p>EQ.0: No data is written EQ.1: Output the peak pressures and surface energy by contact interface</p>
SCLP	<p>A scaling parameter used in the computation of the peak pressure. This parameter is generally set to unity (the default), but it must be greater than 0.</p>
MSSCL	<p>Output nodal information related to mass scaling into the D3PLOT database. This option can be activated if and only if DT2MS < 0.0, see control card *CONTROL_Timestep. This option is available starting with the second release of Version 971.</p> <p>EQ.0: No data is written EQ.1: Output incremental nodal mass EQ.2: Output percentage increase in nodal mass</p>
THERM	<p>Output of thermal data to d3plot. The use of this option (THERM>0) may make the database incompatible with other 3rd party software.</p> <p>EQ.0: (default) output temperature EQ.1: output temperature EQ.2: output temperature and flux EQ.3: output temperature, flux, and shell bottom and top surface temperature</p>

Remarks:

1. If MAXINT is set to 3 then mid-surface, inner-surface and outer-surface stresses are output at the center of the element to the LS-DYNA database. For an even number of integration points, the points closest to the center are averaged to obtain the midsurface values. If multiple integration points are used in the shell plane, the stresses at the center of the element are found by computing the average of these points. For MAXINT equal to 3 LS-DYNA assumes that the data for the user defined integration rules are ordered from bottom to top even if this is not the case. If MAXINT is not equal to 3, then the stresses at the center of the element are output in the order that they are stored for the selected integration rule. If multiple points are used in plane the stresses are first averaged.
2. Beam stresses are output to the LS-DYNA database if and only if BEAMIP is greater than zero. In this latter case the data that is output is written in the same order that the integration points are defined. The data at each integration point consists of the following

five values for elastic-plastic Hughes-Liu beams: the normal stress, σ_{rr} ; the transverse shear stresses, σ_{rs} and σ_{tr} ; the effective plastic strain, and the axial strain which is logarithmic. For beams that are not elastic-plastic, the first history variable, if any, is output instead of the plastic strain. For the beam elements of Belytschko and his co-workers, the transverse shear stress components are not used in the formulation. No data is output for the Belytschko-Schwer resultant beam.

3. If mass scaling is active, the output of the time step size reveals little information about the calculation. If global mass scaling is used for a constant time step, the total element mass is output; however, if the mass is increased so that a minimum time step size is maintained (DT2MS is negative), the added mass is output. Also, see the control card *CONTROL_TIMESTEP.

***DATABASE**

***DATABASE_EXTENT**

For the SSSTAT option the following card(s) apply:

Define as many cards as necessary.

(Define one part set ID for each subsystem. Use as many cards as necessary.)

Card 1 2 3 4 5 6 7 8

Variable	PSID1	PSID2	PSID3	PSID4	PSID5	PSID6	PSID7	PSID8
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

PSIDn

Part set ID for subsystem n.; see *SET_PART.

***DATABASE_FORMAT**

Purpose: Define the output format for binary files.

Card 1 2 3 4 5 6 7 8

Variable	IFORM	IBINARY						
Type	I	I						
Default	0	0						
Remarks	1	2						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IFORM	Output format for D3PLOT and D3THDT files EQ.0: LS-DYNA database format (default), EQ.1: ANSYS database format, EQ.2: Both LS-DYNA and ANSYS database formats.
IBINARY	Word size of the binary output files (D3PLOT, D3THDT, D3DRLF and interface files for 64 bit computer such as CRAY and NEC. EQ.0: default 64 bit format, EQ.1: 32 bit IEEE format

Remarks:

1. This option is not available for every platform. Check LS-DYNA Banner upon execution of the program
2. By using this option one can reduce the size of the binary output files which are created by 64 bits computer such as CRAY and NEC.

*DATABASE

*DATABASE_FSI

*DATABASE_FSI

Purpose: This card may be used to output information about certain coupled Lagrangian surfaces. The Lagrangian shell/segment entity to be monitored must be included in a *CONSTRAINED_LAGRANGE_IN_SOLID (CLIS) card. *DATABASE_FSI activates the output of an ASCII file called "dbfsi". This file contains some coupling information (force, pressure, accumulated mass flowing over some surfaces, etc.) for the coupled Lagrangian surface.

Card 1 1 2 3 4 5 6 7 8

Variable	DT							
Type	F							

Define one surface per card

Card 2,3,... 1 2 3 4 5 6 7 8

Variable	DBFSI_ID	SID	SIDTYPE	SWID	CONVID			
Type	I	I	I	I	I			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DT	Output interval
DBFSI_ID	Surface ID (for reference purposes only) or a DATABASE_FSI entity ID. It consists of a geometric entity defined by the set ID below.
SID	Set ID defining the geometrical surface(s) through/upon which some data is to be tracked and output to an ASCII file called "dbfsi". This set ID can be a (1) PID or (2) PSID or (3) SGSID. This Lagrangian SID must be contained in a Lagrangian slave SID defined in a corresponding coupling card, *CONSTRAINED_LAGRANGE_IN_SOLID.
SIDTYPE	Set type: EQ.0: Part set, EQ.1: Part, EQ.2: Segment set.

VARIABLE	DESCRIPTION
SWID	Switch ID from a corresponding *ALE_FSI_SWITCH_MMG_ID card. If defined, the accumulative mass of the “switched” ALE multi-material group (AMMG) is written out under the “pleak” parameter in the “dbfsi” file.
CONVID	For airbag application only: Convection ID from a corresponding *LOAD_ALE_CONVECTION_ID card (which computes the heat transfer between inflator gas and the inflator canister). If defined, the temperature of the Lagrangian part having heat transfer with the gas, and its change in temperature as function of time in the “dbfsi” file.

Remarks:

- When a Lagrangian mesh overlaps with an Eulerian or ALE mesh, the fluid-structure (or ALE-Lagrangian) interaction may be modeled via a *CONSTRAINED_LAGRANGE_IN_SOLID (CLIS) card. This database command allows for the tracking of certain coupling information related to the flow across, and the load on some selected Lagrangian surfaces defined in corresponding CLIS card.
- The output parameters in the dbfsi ASCII file are:
 - p = Averaged pressure on the surface being tracked (Pa)
 - fx,fy,fz = Total force components (N) over the entity(ies) defined (acting at centroid of each surface)
 - pleak = Under LS-Prepost ASCII plotting, this is labeled “POROSITY”. See remark 3 below.
 - Mflux = If a shell part or part set is included in the coupling, and the normal of this shell structure points away from the fluid to be coupled to, then the fluid will flow across this surface since it is not detected by this surface. For example, this may be done for an airbag vent hole. Then, the amount of accumulated mass flowing across this surface may be output via the “mflux” parameter in the “dbfsi” ASCII output file. Under LS-Prepost ASCII plotting, it is labeled “Outlet_mass_Flux”. This value is only an approximation as the relative velocity between the shell and the fluid group is used to compute the mass, but not precisely at the coupling points. The relative velocity vector dots with the normal vector of the Lagrangian surface to give the “sign” of the mass variable.
 - {fx-lc,fy-lc,fz-lc} = Average x|y|z leakage control force component over the surface entity. This is used for debugging only. Too high leakage control forces (relative to coupling forces) may indicate that alternate coupling approach should be considered since the main coupling force is putting out too little resistance to leakage.
 - Ptemp = Lagrangian part Temperature (Activated only when the *LOAD_ALE_CONVECTION card is used).

Pdtemp = Lagrangian part Temperature increase (Activated only when the *LOAD_ALE_CONVECTION card is used).

3. **PLEAK** parameter in the “dbfsi” ASCII output file from this keyword contains the accumulated **mass** (for example, Kg) for 4 different cases:
 - a) When LCIDPOR is defined in the coupling card (CLIS), porous flow across a Lagrangian shell surface may be monitored and output in PLEAK.
 - b) Porous flow across Lagrangian shell may also be defined via a load curve in the *MAT_FABRIC card, and similar result will be tracked and output.
 - c) When NVENT in the CLIS card is defined (isentropic venting), the flow across the isentropic vent hole may be output in PLEAK.
 - d) When an *ALE_FSI_SWITCH_MMG_ID card is defined, and the SWID parameter specifies this ID to be tracked, then the amount of accumulated mass that has been switched when flowing across a monitoring surface is output.

Example:

Consider a model with a Lagrangian mesh overlaps with an Eulerian or ALE mesh. On the Lagrangian mesh, there are 3 Lagrangian surface sets over which some data is to be written out.

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ INPUT:
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*DATABASE_FSI
$      dt
      2.97E-06
$ DBFSI_ID      SID      STYPE      swid      convid [STYPE: 0=PSID;1=PID;2=SGSID]
      11          1          2
      12          2          2
      13          3          1
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ This reads:
$ DBFSI_ID 11 is defined by a SID=1: a SGSID = as specified by STYPE=2
$ DBFSI_ID 12 is defined by a SID=2: a SGSID = as specified by STYPE=2
$ DBFSI_ID 13 is defined by a SID=3: a PID = as specified by STYPE=1
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ An OUTPUT file called "dbfsi" looks like the following:
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
      Fluid-structure interaction output
      Number of surfaces:          3

      id          p          fx          fy          fz          pleak
      mflux      fx-lc      fy-lc      fz-lc      Ptemp
PDtmp
      time= 0.00000E+00
      11  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
      0.0000E+00
      12  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
      0.0000E+00
      13  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
      0.0000E+00
      time= 0.29709E-05
      11  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
      0.0000E+00
      12  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
      0.0000E+00
      13  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
      0.1832E-06
      0.0000E+00
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

***DATABASE_FSI_SENSOR**

Purpose: This card activates the output of an ASCII file called “dbsensor”. Its input defines the pressure sensors’ locations which follow the positions of some Lagrangian segments during the simulation. Its ASCII output file, dbsensor, contains the spatial position of the sensor and its recorded pressure from the ALE elements containing the sensors. This card is activated when a *CONSTRAINED_LAGRANGE_IN_SOLID card is used and the Lagrangian shell elements defining the locations of the sensors must be included in the slave or structure coupling set.

Card 1 Format

Card 1	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							

Card(s) 2+ Format: Define one surface per line

Card 2,3,...	1	2	3	4	5	6	7	8
Variable	DBFSI_ID	NID	SEGMID	OFFSET				
Type	I	I	I	F				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DT	Output interval
DBFSI_ID	Pressure-Sensor ID.
NID	An optional Lagrangian node ID defining an approximate pressure sensor location with respect to a Lagrangian shell element. This is not a required input.
SEGMID	A required Lagrangian shell element ID for locating the pressure sensor. If NID=0 or blank, the sensor will be automatically placed in the center of this SEGMID, accounting for the offset distance.
OFFSET	Offset distance between the pressure sensor and the Lagrangian segment surface. If it is positive, it is on the side pointed to by the segment normal vector and vice versa.

Remarks:

1. The output parameters in the “dbsensor” ASCII file are:

- ID = Sensor ID.
- x,y,z = Sensor spatial location.
- P = Sensor recorded pressure (Pa) from the ALE fluid element containing the sensor.

For example to plot the sensor pressure in LS-Prepost, select:
ASCII → dbsensor → LOAD → (select sensor ID) → Pressure → PLOT

Example 1:

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ INPUT:
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*DATABASE_FSI_SENSOR
  0.01
$ DBFSI_ID      NID SEGMENTID  OFFSET
   10          360      355      -0.5
   20          396      388      -0.5
   30          324      332      -0.5
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ The 1st line reads:
$ SENSOR_ID 10 is located by segment-ID=355. Node-ID=360 precisely locate this
$ sensor (if NID=0, then the sensor is located at the segment center). This
$ sensor is located 0.5 length unit away from the segment surface. Negative
$ sign indicates a direction opposite to the segment normal vector.
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ An OUTPUT file called "dbsensor" looks like the following:
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
ALE sensors output
Number of sensors: 3

      id          x          y          z          p
time= 0.17861E-02
  10  0.0000E+00  0.0000E+00 -0.3900E+00  0.1085E-03
  20 -0.2250E+02  0.2250E+02 -0.3900E+00  0.1085E-03
  30  0.2250E+02 -0.2250E+02 -0.3900E+00  0.1085E-03
time= 0.20081E-02
  10  0.0000E+00  0.0000E+00 -0.3900E+00  0.1066E-03
  20 -0.2250E+02  0.2250E+02 -0.3900E+00  0.1066E-03
  30  0.2250E+02 -0.2250E+02 -0.3900E+00  0.1066E-03
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ ID
= DBFSI_ID
$ x,y,z = Sensor location (defined based on a Lagrangian segment)
$ p      = Sensor pressure as taken from the fluid element containing the sensor.
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

***DATABASE_HISTORY_OPTION**

Available options include:

BEAM
BEAM_SET
BEAM_ID
DISCRETE
DISCRETE_ID
DISCRETE_SET
NODE
NODE
NODE_ID
NODE_LOCAL
NODE_LOCAL_ID
NODE_SET
NODE_SET_LOCAL
SEATBELT
SEATBELT_ID
SHELL
SHELL_ID
SHELL_SET
SOLID
SOLID_ID
SOLID_SET
SPH
SPH_SET
TSHELL
TSHELL_ID
TSHELL_SET

Purpose: Control which nodes or elements are output into the binary history file, D3THDT, the ASCII file NODOUT, the ASCII file ELOUT and the ASCII file SPHOUT. Define as many cards as necessary. The next "*" card terminates the input. See also *DATABASE_BINARY_OPTION and *DATABASE_OPTION.

***DATABASE**

***DATABASE_HISTORY**

For options NODE_LOCAL, NODE_LOCAL_ID, and NODE_SET_LOCAL

Cards 1,.. 1 2 3 4 5 6 7 8

Variable	ID	CID	REF	HFO				
Type	I	I	I	I				

Read the second card for NODE_LOCAL_ID, otherwise, skip.

Card 1,2,... 1-7 8

Variable	HEADING							
Type	A70							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID	NODE/NODE_SET set ID. The contents of the files are given in Table 9.1 for nodes. See the remark below concerning accelerometer nodes.
CID	Coordinate system ID for nodal output. See DEFINE_COORDINATE options.
REF	Output reference: EQ.0: Output is in the local system fixed for all time from the beginning of the calculation. EQ.1: Output is in the local system which is defined by the DEFINE_COORDINATE_NODES. The local system can change orientation depending on the movement of the three defining nodes. The defining nodes can belong to either deformable or rigid parts. EQ.2: Output is <u>relative</u> to the local system which is defined by the DEFINE_COORDINATE_NODES option. The local system can change orientation depending on the movement of the three defining nodes. If dynamic relaxation is used, the reference location is reset when convergence is achieved.
HFO	Flag for high frequency output into NODOUTHF EQ.0: Nodal data written to NODOUT file only EQ.1: Nodal data also written NODOUTHF at the higher frequency
HEADING	A description of the nodal point. It is suggested that unique description be used. This description is written into the D3HSP file and into the ASCII database NODOUT.

Remarks:

1. If a node belongs to an accelerometer, see *ELEMENT_SEATBELT_ACCELEROMETER, and if it also appears as an active node in the NODE_LOCAL or NODE_SET_LOCAL keyword, the coordinate system, CID, transformations will be skipped and the LOCAL option will have no effect.

*DATABASE

*DATABASE_NODAL_FORCE_GROUP

*DATABASE_NODAL_FORCE_GROUP

Purpose: Define a nodal force group for output into ASCII file NODFOR and the binary file XTFILE. See also *DATABASE_OPTION and *DATABASE_BINARY_OPTION.

Card 1 2 3 4 5 6 7 8

Variable	NSID	CID						
Type	I	I						
Default	none	none						

VARIABLE

DESCRIPTION

NSID	Nodal set ID, see *SET_NODE_OPTION.
CID	Coordinate system ID for output of data in local system, see *DEFINE_COORDINATE_OPTION.

Remarks:

1. The nodal reaction forces in the global or local (if CID is defined above) x, y, and z directions are printed into the NODFOR ascii file along with the external work which is a result of these reaction forces. The resultant force vector found by summing the reaction forces over the nodes is also written into this file. These forces can be a result of applied boundary forces such as nodal point forces and pressure boundary conditions, body forces, and contact interface forces. In the absence of body forces, interior nodes would always yield a null force resultant vector. In general this option would be used for surface nodes.

***DATABASE_SPRING_FORWARD**

Purpose: Create spring forward nodal force file. This option is to output resultant nodal force components of sheet metal at the end of the forming simulation into an ASCII file, "SPRING-FORWARD", for spring forward and die corrective simulations.

Card 1 2 3 4 5 6 7 8

Variable	IFLAG							
Type	I							

VARIABLE

DESCRIPTION

IFLAG

Output type:
EQ.0: off,
EQ.1: output element nodal force vector for deformable nodes.

***DATABASE**

***DATABASE_SUPERPLASTIC_FORMING**

***DATABASE_SUPERPLASTIC_FORMING**

Purpose: Specify the output intervals to the superplastic forming output files. The option *LOAD_SUPERPLASTIC_FORMING must be active.

Card 1 2 3 4 5 6 7 8

Variable	DTOUT							
Type	F							

VARIABLE

DESCRIPTION

DTOUT

Output time interval for output to “pressure”, “curve1” and “curve2” files. The “pressure” file contains general information from the analysis and the files “curve1” and “curve2” contain pressure versus time from phases 1 and 2 of the analysis. The data in the pressure and curve files may be plotted using ASCII > superpl in LS-Prepost.

***DATABASE_TRACER**

Purpose: Tracer particles will save a history of either a material point or a spatial point into an ASCII file, TRHIST. This history includes positions, velocities, and stress components. The option *DATABASE_TRHIST must be active. This option applies to ALE and SPH problems.

Card 1 2 3 4 5 6 7 8

Variable	TIME	TRACK	X	Y	Z			
Type	F	I	F	F	F			
Default	0.0	Lagrangian	0	0	0			

VARIABLE	DESCRIPTION
TIME	Start time for tracer particle
TRACK	Tracking option: EQ.0: particle follows material, EQ.1: particle is fixed in space.
X	Initial x-coordinate
Y	Initial y-coordinate
Z	Initial z-coordinate

***DEFINE**

The keyword ***DEFINE** provides a way of defining boxes, coordinate systems, load curves, tables, and orientation vectors for various uses. The keyword cards in this section are defined in alphabetical order:

- *DEFINE_ALEBAG_BAG**
- *DEFINE_ALEBAG_HOLE**
- *DEFINE_ALEBAG_INFLATOR**
- *DEFINE_BOX**
- *DEFINE_BOX_ADAPTIVE**
- *DEFINE_BOX_COARSEN**
- *DEFINE_BOX_DRAWBEAD**
- *DEFINE_BOX_SPH**
- *DEFINE_CONNECTION_PROPERTIES_{OPTION}**
- *DEFINE_CONSTRUCTION_STAGES**
- *DEFINE_CONTACT_VOLUME**
- *DEFINE_COORDINATE_NODES**
- *DEFINE_COORDINATE_SYSTEM**
- *DEFINE_COORDINATE_VECTOR**
- *DEFINE_CURVE_{OPTION}**
- *DEFINE_CURVE_COMPENSATION**
- *DEFINE_CURVE_DRAWBEAD**
- *DEFINE_CURVE_ENTITY**
- *DEFINE_CURVE_FEEDBACK**
- *DEFINE_CURVE_FUNCTION**
- *DEFINE_CURVE_SMOOTH**
- *DEFINE_CURVE_TRIM_{OPTION}**
- *DEFINE_DEATH_TIMES_{OPTION}**
- *DEFINE_FRICTION**
- *DEFINE_HEX_SPOTWELD_ASSEMBLY_{OPTION}**
- *DEFINE_SD_ORIENTATION**
- *DEFINE_SET_ADAPTIVE**
- *DEFINE_SPOTWELD_FAILURE_RESULTANTS**

***DEFINE**

***DEFINE_SPOTWELD_RUPTURE_PARAMETER**

***DEFINE_SPOTWELD_RUPTURE_STRESS**

***DEFINE_STAGED_CONSTRUCTION_PART**

***DEFINE_TABLE**

***DEFINE_TRANSFORMATION**

***DEFINE_VECTOR**

An additional option **_TITLE** may be appended to all the ***DEFINE** keywords. If this option is used then an addition line is read for each section in 80a format which can be used to describe the defined curve, table, etc. At present LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

Examples for the ***DEFINE** keyword can be found at the end of this section.

***DEFINE_ALEBAG_BAG**

Purpose: This card defines information about the Lagrangian airbag structure that is required as complementary definition to the *AIRBAG_ADVANCED_ALE (AAA) card. It contains information about (a) one Lagrangian shell structure representing an airbag (or part of one), (b) venting characteristics of this airbag, and (c) its main coupling control features (Remark 1). The AAA card is used to model the airbag-to-inflator-gas interaction via the ALE method (Remark 2).

Card 1 Format

Card 1 1 2 3 4 5 6 7 8

Variable	BAGID	SID	SIDTYPE	CVBAG	IBLOCK	VTCOEF	VENTSID	VENTYP
Type	I	I	I	I	I	F	I	I
Default	none	none	1	none	none	none	0	none
Remarks	1			3	4	5	6	

Card 2 Format (All coupling parameters are also defined similarly under the *CONSTRAINED_LAGRANGE_IN_SOLID card)

Card 2 1 2 3 4 5 6 7 8

Variable	NQUAD	CTYPE	PFAC	FRIC	FRCMIN	NORMTYP	ILEAK	PLEAK
Type	I	I	F	F	F	I	I	F
Default	none	none	0.1	none	none	2	0.1	none
Remarks	7		8				9	

*DEFINE

*DEFINE_ALEBAG_BAG

Card 3 Format (All coupling parameters are also defined similarly under the *CONSTRAINED_LAGRANGE_IN_SOLID card)

Card 3	1	2	3	4	5	6	7	8
Variable	NORM	START	END					
Type	I	F	F					
Default	1	0.0	1.0E10					
Remarks	10							

VARIABLE	DESCRIPTION
BAGID	An ID associated with the airbag mesh definition defined to be used in a corresponding *AIRBAG_ADVANCED_ALE card (See Remark 1).
SID	A set ID defining the Lagrangian airbag structure that is to be coupled to the inflator gas in a corresponding *AIRBAG_ADVANCED_ALE card. BAGID points to SID.
SIDTYP	Set type for the SID above: EQ.0: for a part set ID (PSID) EQ.1: for a part ID (PID)
CVBAG	Flag for including the SID above in the control volume (CV) analysis phase (See Remark 3): EQ.0: Do not include the above SID in subsequent CV analysis EQ.1: Include the above SID in subsequent CV analysis
IBLOCK	Flag for considering contact blockage for venting holes and porous flow (See Remark 4): EQ.0: Do not consider contact blockage EQ.1: Consider contact blockage
VTCOEF	Flow coefficient for each vent surface area defined by VENTSID (See Remark 5).
VENTSID	Set ID defining the vent hole surface (shape, See Remark 6). This venting estimate is sometimes referred to as “isentropic venting” as the flow is estimated via isentropic flow correlations. The amount of mass that escapes out of the vent hole is simply subtracted (deleted) from the mass inside the airbag, reducing the inflating potential of the inflator gas. The vented flow AMMG cannot be visualized in LS-PrePost.

VARIABLE	DESCRIPTION
VENTYP	Set ID type of the vent surface area defined by VENTSID: EQ.0: Part set ID (PSID). EQ.1: Part ID (PID). EQ.2: Segment set ID (SGSID).
NQUAD	Number of (quadrature) coupling points distributed over each coupled Lagrangian surface segment (See Remark 7). EQ.0: NQUAD will be set by default to 4, EQ.n: An NQUAD*NQUAD coupling points distribution over each Lagrangian segment is defined, EQ.-n: NQUAD is reset to a positive value. Coupling at nodes is obsolete.
CTYPE	Fluid-Structure coupling method: EQ.4: penalty coupling for shell (with or without erosion) and solid elements (without erosion). DIREC is set to 2 (default). EQ.6: penalty coupling designed for airbag modeling which automatically controls the DIREC parameter internally. It is equivalent to setting {CTYPE=4; DIREC=1} for unfolded region; and {CTYPE=4; DIREC=2}; in folded region. For both cases: {ILEAK=2; FRCMIN=0.3}.
PFAC	Penalty factor (CTYPE 4 and 6). PFAC is a scale factor for scaling the estimated stiffness of the interacting (coupling) system. It is used to compute the coupling forces to be distributed on the slave and master parts. If positive real: Fraction of estimated critical stiffness. If negative integer, -n: Refers to load curve ID n. The curve defines the coupling pressure (y-axis) as a function of the penetration (x-axis). (See Remark 8).
FRIC	Coefficient of friction (used with DIREC 2 only).
FRCMIN	Minimum volume fraction of a coupled ALE multi-material group (AMMG), or fluid, in a multi-material ALE element to activate coupling. Default value is 0.5. Reducing FRCMIN (typically, between 0.1 and 0.3) would turn on coupling earlier to prevent leakage in hypervelocity impact cases.
NORMTYP	Penalty coupling spring (or force) direction (DIREC 1 and 2): EQ.0: normal vectors are interpolated from nodal normals (default). EQ.1: normal vectors are interpolated from segment normals. This is sometimes a little more robust for sharp Lagrangian corners, and folds.

VARIABLE	DESCRIPTION
ILEAK	Coupling leakage control flag (See Remark 9): EQ.0: none (default), EQ.1: weak, leakage control is turned off if the penetrating volfrac > FRCMIN+0.1. EQ.2: strong, with improved energy consideration. Leakage control is turned off if the penetrating volfrac > FRCMIN+0.3.
PLEAK	Leakage control penalty factor, $0 < \text{PLEAK} < 0.2$. This factor influences the additional coupling force magnitude to prevent leakage. It is conceptually similar to PFAC. Most of the time, the default value (0.1) is adequate.
NORM	A flag indicating the rule for defining which side of the Lagrangian segment the fluid is supposed to be coupled to. By default (NORM=0) the fluid on the side pointed to by the Lagrangian segment normal (head-side) is coupled to. To couple to the fluid on the side not pointed to by the segment normals (tail-side), set NORM=1 (See Remark 10). "Head" and "tail" refer to the normal vector of the coupling segment. EQ.0: Couple fluid to head-side of Lagrangian segment. EQ.1: Couple fluid to tail-side of Lagrangian segment.
START	Start time for coupling.
END	End time for coupling.

Remarks:

1. This command provides supplemental information for the main *AIRBAG_ADVANCED_ALE (AAA) card. Specifically, it defines (a) a Lagrangian airbag structure that couples with the inflator gas, (b) its venting characteristics, and (c) some of the main coupling parameters. The information input for the AAA set is translated into the ALE keywords internally in LS-DYNA. This input approach does not include all the control features available in the regular ALE modeling approach, especially for couplings. Therefore highly complex fluid-structure interactions may require going back to the traditional ALE input approach. For more details on all coupling parameters, please see *CONSTRAINED_LAGRANGE_IN_SOLID card.
2. The airbag inflation process may be modeled in 2 stages. In the 1st stage the bag-gas interaction is modeled via ALE method where a variable pressure field inside the airbag may be simulated. In the 2nd stage, after the airbag has opened up sufficiently, the uniform pressure inflation method (also called control volume, or CV, approach) is used to inflate the bag. Please refer to the AAA card for more information.
3. If the Lagrangian structure is to be included in the 1st stage only (deployment using ALE method) but to be excluded from the 2nd stage analysis (uniform pressure analysis), then set CVBAG=0. One such example may be the inner bag of a bag-in-bag model. If the

defined Lagrangian structure is to be included in both computational phases, set CVBAG=1.

4. Fabric venting and porous flow behaviors for ALE and CV phases (see *CONSTRAINED_LAGRANGIAN_IN_SOLID (CLIS) and *AIRBAG_HYBRID) is defined under the *MAT_FABRIC card {FLC(t), FAC(P), FVOPT}. The FLC(t) is the orifice flow coefficient (typically is close to 1.0). FAC(P) is a relative porous gas speed curve as a function of absolute upstream pressure. For AAA application, FVOPT must be either 7 or 8. FVOPT 7 and 8 will be used for both ALE and CV phases (*AIRBAG_HYBRID). Blockage consideration for both venting and porous flows is accounted for by the IBLOCK flag. If IBLOCK=0 then in the CV phase FVOPT is set to 7, no contact blockage consideration. If IBLOCK=1 then in the CV phase FVOPT is set to 8, with contact blockage consideration. IBLOCK overwrites FVOPT.
5. VTCOEF will be used to scale the vent area for ALE venting. Upon switching to the CV phase, this coefficient will be used in place of the vent coefficient "C23" of the *AIRBAG_HYBRID card.
6. VENTSID defines the isentropic venting area definition which is used for ALE venting. Upon switching to the CV phase, the venting area will be used for venting in place of parameter "A23" of the *AIRBAG_HYBRID card.
7. See remark 2 under *CONSTRAINED_LAGRANGIAN_IN_SOLID.
8. See remark 6 under *CONSTRAINED_LAGRANGIAN_IN_SOLID
9. See remark 10 under *CONSTRAINED_LAGRANGIAN_IN_SOLID.
10. See remark 7 under *CONSTRAINED_LAGRANGIAN_IN_SOLID. Typically, if the airbag shell elements have their normal vectors pointing outward. Then to couple to the inflator gas from the inside of the bag NORM is set to 1 (the default value for NORM in this card).

*DEFINE

*DEFINE_ALEBAG_HOLE

*DEFINE_ALEBAG_HOLE

Purpose: This optional card defines information about a physical vent hole structure of a Lagrangian airbag. This card is only used with an associated *AIRBAG_ADVANCED_ALE (AAA) card to provide supplemental venting flow information for an airbag deployment simulation (Remark 1). This card defines (a) geometry of a vent hole, (b) information for switching the ALE multi-material group (AMMG) ID of the inflator gas when it passes through a vent hole, and (c) type of vent hole. The information input here is translated into a *ALE_FSI_SWITCH_MMG_ID card (Remark 2).

Card 1 Format

Card	1	2	3	4	5	6	7	8
Variable	HOLEID	SID	SIDTYPE	NQUAD	XOFF	NFOLD	XCLEN	
Type	I	I	I	I	F	I	F	
Default	none	none	none	none	none	none	none	
Remarks	1						3	

VARIABLE

DESCRIPTION

HOLEID	An ID associated with a vent hole surface in an airbag structure definition. It is defined to be used in a corresponding *AIRBAG_ADVANCED_ALE card. (See Remark 1.)
SID	A set ID defining the physical geometry a vent hole in the Lagrangian airbag structure. The inflator gas is supposed to pass through this hole.
SIDTYP	Set type for the SID above: EQ.0: for a part set ID (PSID) EQ.1: for a part ID (PID)
NQUAD	The number of flow-sensor points to be distributed over each monitoring surface or segment (defined by SID). There should be enough sensor points, distributed in each ALE element, to monitor the flow across this monitoring surface (see remark 3). At least 1 or 2 sensor points are needed in each ALE element to monitor the flow through that element. An alternate method for defining flow-sensor points is by defining XCLEN parameter below.

VARIABLE	DESCRIPTION
XOFF	An offset distance away from the monitoring surface, beyond which the AMMGID switching occurs. A positive value of XOFF means the offset distance is along the normal vector direction of the monitoring segment, and vice versa. This offset distance, in general, should be about 1.5 to 2 times the ALE element widths where the vent flow is occurring (default=0.0).
NFOLD	Flag for checking folding logic (default=0=off). If NFOLD=1=on, then LS-DYNA will check if the shell elements defining the monitoring surface are in the folded region or not. If the monitoring segment is still located within a folded region, then no AMMG switching is allowed yet until it has unfolded.
XCLEN	This is an absolute distance for distributing the flow sensor points over each monitoring segment (surface). To make sure that at least 1 or 2 sensor points are present in each ALE element to track the flow of an AMMG, XCLEN may be roughly estimated as a third or one-half the width of the smallest ALE element in the mesh (see Remark 3).

Remarks:

1. A vent hole associated with an airbag structure may be defined for modeling physical venting (actual flow of material across the hole surface may be monitored, in contrast to isentropic venting where vented material is simply deleted). The primary function of this card is to define the vent hole geometry and the AMMG switching as the gas passes through it.

This card provides supplemental information for an associated *AIRBAG_ADVANCED_ALE (AAA) card. The vent hole structure is associated with and airbag ID under AAA so that the coupling and AMMGID switching may be performed. This is equivalent to modeling “physical venting holes” via the *CONSTRAINED_LAGRANGE_IN_SOLID card. It is used only when users define fine enough ALE mesh to resolve the flow across the vent (physical venting, see remark 12 under *CONSTRAINED_LAGRANGE_IN_SOLID card).

The information input for the AAA card and all its supplemental commands are converted into the ALE keywords internally in LS-DYNA. The corresponding ALE keywords are written out to a file called “advalebag.kw” for checking. For an in-depth understanding of the interaction modeling, it is recommended that the users understand how to set up the coupling with the traditional ALE method.

2. For more detailed information, the user may review the *ALE_FSI_SWITCH_MMG_ID card since it is the actions of this card that are being executed. The AMMGIDs of the gases involved in the switching, upstream and downstream of the hole are defined automatically.
3. When both NQUAD and XCLEN are defined, whichever gives smaller distance between sensor-points will be used. XCLEN may give better control as in the case of a null shell acting as the monitoring surface. As this null shell is stretched, NQUAD distribution of sensor-points may not be adequate, but XCLEN would be.

*DEFINE

*DEFINE_ALEBAG_INFLATOR

*DEFINE_ALEBAG_INFLATOR

Purpose: This card defines supplemental information about an airbag inflator that is required as complementary definition to the *AIRBAG_ADVANCED_ALE (AAA) card. It defines the (a) orifices, (b) gas properties, and (c) inlet conditions (Remark 1). This command is only used with an associated *AIRBAG_ADVANCED_ALE card. This card contains basic information that is required by the ALE command *SECTION_POINT_SOURCE_MIXTURES. It is only used for modeling airbag deployment process (Remark 2) via the ALE method.

Card 1 1 2 3 4 5 6 7 8

Variable	INFLAID	unused	unused	unused	NGAS	NORIF	LCIDVEL	LCIDT
Type	I				I	I	I	I
Default	none				none	none	0	none
Remarks	1							

**Repeat this card "NGAS" times, one for each species in the mixture.
See *AIRBAG_HYBRID**

Card 2 1 2 3 4 5 6 7 8

Variable	LCIDMD	unused	unused	MWGAS	unused	A	B	C
Type	I			F		F	F	F
Default	none			none		none	0.0	0.0
Remarks						3	3	3

Repeat this card "NORIF" times.

See *SECTION_POINT_SOURCE_MIXTURE

Card 3	1	2	3	4	5	6	7	8
Variable	NODEID	VECID	ORIFAREA	unused	unused	unused	unused	unused
Type	I	I	F					
Default	0	0	0.0					
Remarks	4							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
INFLAID	An inflator ID. It is defined to be used in a corresponding *AIRBAG_ADVANCED_ALE card. (See Remark 1).
NGAS	Number of thermally equilibrated ideal gas species making up one inflator gas mixture.
NORIF	Number of inflator inlet orifices through which the inflator gas mixture is injected into the airbag. Each orifice requires a node ID to define its location (see NODEID below).
LCIDVEL	User-estimated inflator gas mixture average velocity load curve ID. If LCIDVEL=0 or blank, LSDYNA will estimate the inlet gas velocity.
LCIDT	Inflator gas mixture thermally equilibrated stagnation temperature load curve ID. All species of the mixture are assumed to have the same average stagnation temperature.
LCIDMD	The inlet mass flow rate load curve ID of this species of the inflator gas mixture.
MWGAS	Molecular weight of this species (e.g., kg/mole).
A	Nominal constant-pressure heat capacity (per-mole unit) at STP (see Remark 3).
B	1 st order (linear) coefficient of temperature dependent heat capacity of the inflator gas. (see Remark 3).
C	2 nd order (quadratic) coefficient of temperature dependent heat capacity of the inflator gas. (see Remark 3).

VARIABLE	DESCRIPTION
NODEID	A node ID defining the location of a point source (see Remark 4).
VECID	A vector ID defining the direction of flow at each point source.
ORIFAREA	The orifice area at each point source.

Remarks:

1. This card defines the inflator inflow information for one inflator referred to by an *AIRBAG_ADVANCED_ALE (AAA) card. The basic information includes (a) gas properties, (b) orifices, and (c) inlet conditions. The gas properties are defined similarly to that of the *AIRBAG_HYBRID card. The orifices, and inlet conditions (see *SECTION_POINT_SOURCE_MIXTURE), consist of $\dot{m}(t), T_{stag}(t)$, and maybe an estimated inlet gas velocity curve, $\tilde{v}_{gas}(t)$, if available. The information input for the AAA card and all its associates are translated into ALE keywords internally in LS-DYNA. The ALE keywords translated by AAA are output to the “advalebak.kw” file for review. For in depth understanding of the interaction modeling, it is recommended that the users understand how to set up the coupling with the traditional ALE method. The AAA card requires 2 additional cards for complementary definitions, *DEFINE_ALEBAG_BAG and *DEFINE_ALEBAG_INFLATOR. One optional card, *DEFINE_ALEBAG_HOLE, may be used to define the vent hole of the airbag.
2. The airbag inflation process may be modeled in 2 stages. In the 1st stage the bag-gas interaction is modeled via ALE method where a variable pressure field inside the airbag may be simulated. In the 2nd stage, after the airbag has opened up sufficiently, the uniform pressure inflation method (also called control volume, or CV, approach) is used to inflate the bag.
3. The per-mass-unit, temperature-dependent, constant-pressure heat capacity is

$$C_p(T) = \frac{A + B * T + C * T^2}{MW} \sim \frac{J}{kg * K} \quad \begin{array}{l} B \sim J / (mole * K^2) \\ C \sim J / (mole * K^3) \end{array}$$

$$A = \tilde{C}_{p0} \sim J / (mole * K)$$

The units shown are only for demonstration of the equation. Please see the *MAT_GAS_MIXTURE card definition.

4. In general, it is best to locate a point source near the center of an ALE element. Associated with each point source is an area and a vector indicating flow direction. Each point source should occupy 1 ALE element by itself, and there should be at least 2 empty ALE elements between any 2 point-sources. A point source should be located at least 2 or 3 elements away from the free surface of an ALE mesh or a Lagrangian surface to prevent interaction with the mesh boundary or coupling boundary (see *SECTION_POINT_SOURCE_MIXTURE).

***DEFINE_BOX**

Purpose: Define a box-shaped volume. Two diagonally opposite corner points of a box are specified in global coordinates. The box volume is then used for various specifications, e.g., velocities, contact, etc.

Card 1 2 3 4 5 6 7 8

Variable	BOXID	XMN	XXM	YMN	YMX	ZMN	ZMX	
Type	I	F	F	F	F	F	F	
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	
Remarks								

VARIABLE**DESCRIPTION**

BOXID	Box ID. Define unique numbers.
XMN	Minimum x-coordinate.
XXM	Maximum x-coordinate.
YMN	Minimum y-coordinate.
YMX	Maximum y-coordinate.
ZMN	Minimum z-coordinate.
ZMX	Maximum z-coordinate.

*DEFINE

*DEFINE_BOX_ADAPTIVE

*DEFINE_BOX_ADAPTIVE

Purpose: Define a box-shaped volume enclosing the elements where the adaptive level is to be specified. If the midpoint of the element falls within the box the adaptive level is reset. Elements falling outside of this volume use the value, MAXLVL, on the *CONTROL_ADAPTIVE control cards.

Card 1 1 2 3 4 5 6 7 8

Variable	BOXID	XMN	XMN	YMN	YMX	ZMN	ZMX	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

Card 2

Variable	PID	LEVEL						
Type	I	I						
Default	0	none						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
BOXID	Box ID. Define unique numbers.
XMN	Minimum x-coordinate.
XMN	Maximum x-coordinate.
YMN	Minimum y-coordinate.
YMX	Maximum y-coordinate.
ZMN	Minimum z-coordinate.
ZMX	Maximum z-coordinate.
PID	Part ID. If zero, all active elements within box are considered.
LEVEL	Maximum number of refinement levels for elements that are contained in the box. Values of 1, 2, 3, 4,... allow a maximum of 1, 4, 16, 64, ... elements, respectively, to be created for each original element.

***DEFINE_BOX_COARSEN**

Purpose: Define a specific box-shaped volume indicating elements which are protected from mesh coarsening. See also *CONTROL_COARSEN.

Card 1 2 3 4 5 6 7 8

Variable	BOXID	XMN	XXM	YMN	YMX	ZMN	ZMX	IFLAG
Type	I	F	F	F	F	F	F	I
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0

VARIABLE**DESCRIPTION**

BOXID	Box ID. Define unique numbers.
XMN	Minimum x-coordinate.
XXM	Maximum x-coordinate.
YMN	Minimum y-coordinate.
YMX	Maximum y-coordinate.
ZMN	Minimum z-coordinate.
ZMX	Maximum z-coordinate.
IFLAG	Flag for protecting elements inside or outside of box. EQ.0: elements inside the box cannot be coarsened EQ.1: elements outside the box cannot be coarsened

Remarks:

1. Many boxes may be defined. If an element is protected by any box then it may not be coarsened.

*DEFINE

*DEFINE_BOX_DRAWBEAD

*DEFINE_BOX_DRAWBEAD

Purpose: Define a specific box or tube shaped volume around a draw bead. This option is useful for the draw bead contact. If box shaped, the volume will contain the draw bead nodes and elements between the bead and the outer edge of the blank. If tubular, the tube is centered around the draw bead. All elements within the tubular volume are included in the contact definition.

Card	1	2	3	4	5	6	7	8
Variable	BOXID	PID	SID	IDIR	STYPE	RADIUS	CID	
Type	I	F	F	F	I	F	I	
Default	0	0.0	0.0	0.0	4	0.0	0	
Remarks						optional	optional	

VARIABLE

DESCRIPTION

BOXID	Box ID. Define unique numbers.
PID	Part ID of blank.
SID	Set ID that defines the nodal points that lie along the draw bead. If a node set is defined, the nodes in the set must be consecutive along the draw bead. If a part or part set is defined, the set must consist of beam or truss elements. Within the part set, no ordering of the elements is assumed, but the number of nodes must equal the number of beam elements plus 1.
IDIR	Direction of tooling movement. The movement is in the global coordinate direction unless the tubular box option is active and CID is nonzero. In this latter case, the movement is in the local coordinate direction. EQ.1: tooling moves in x-direction, EQ.2: tooling moves in y-direction, EQ.3: tooling moves in z-direction.
STYPE	Set type: EQ.2: part set ID, EQ.3: part ID, EQ.4: node set ID.

VARIABLE	DESCRIPTION
RADIUS	The radius of the tube, which is centered around the draw bead. Elements of part ID, PID, that lie within the tube will be included in the contact. If the radius is not defined, a rectangular box is used instead. This option is recommended for curved draw beads and for draw beads that are not aligned with the global axes.
CID	Optional coordinate system ID. <i>This option is only available for the tubular drawbead. This option is available starting in the third release of version 971.</i>

*DEFINE

*DEFINE_BOX_SPH

*DEFINE_BOX_SPH

Purpose: Define a box-shaped volume. Two diagonally opposite corner points of a box are specified in global coordinates. Particle approximations of SPH elements are computed when particles are located inside the box. The load curve describes the motion of the maximum and minimum coordinates of the box.

Card 1 1 2 3 4 5 6 7 8

Variable	BOXID	XMN	XMN	YMN	YMX	ZMN	ZMX	VID
Type	I	F	F	F	F	F	F	I
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0

Card 2

Variable	LCID	VD						
Type	I	I						
Default	0	0						

VARIABLE

DESCRIPTION

BOXID	Box ID. Define unique numbers.
XMN	Minimum x-coordinate.
XMN	Maximum x-coordinate.
YMN	Minimum y-coordinate.
YMX	Maximum y-coordinate.
ZMN	Minimum z-coordinate.
ZMX	Maximum z-coordinate.
VID	Vector ID for DOF, see *DEFINE_VECTOR.

VARIABLE	DESCRIPTION
LCID	Load curve ID to describe motion value versus time, see *DEFINE_CURVE
VD	Velocity/Displacement flag: EQ.0: velocity, EQ.1: displacement

*DEFINE

*DEFINE_CONNECTION_PROPERTIES

*DEFINE_CONNECTION_PROPERTIES_{OPTION}

Available options include:

<BLANK>

ADD

Purpose: Define failure related parameters for solid element spot weld failure by *MAT_SPOTWELD_DAIMLERCHRYSLER. For each connection identifier, CON_ID, a separate *DEFINE_CONNECTION_PROPERTIES section must be included. The **ADD** option allows material specific properties to be added to an existing connection ID. See remark 2.

:

Card 1 1 2 3 4 5 6 7 8

Variable	CON_ID	PROPRUL	AREAEQ		DG_TYP			
Type	F	I	I		I			
Default	0	0	0		0	.		

Card 2

Variable		D_SIGY	D_ETAN	D_DG_PR	D_RANK	D_SN	D_SB	D_SS
Type		F	F	F	F	F	F	F
Default		none	none	1.0e+10	none	none	none	none

Card 3

Variable	D_EXSN	D_EXSB	D_EXSS	D_LCSN	D_LCSB	D_LCSS		
Type	F	F	F	I	I	I		
Default	1.0	1.0	1.0	0	0	0		

Define the following 2 cards for each shell material that will have material specific data defined for this CON_ID. The input is terminated by the next “*” keyword card.

Card 4 1 2 3 4 5 6 7 8

Variable	MID	SGIY	ETAN	DG_PR	RANK	SN	SB	SS
Type	A8	F	F	F	F	F	F	F
Default				1.0e+10				

Card 5

Variable	EXSN	EXSB	EXSS	LCSN	LCSB	LCSS		
Type	F	F	F	I	I	I		
Default								

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CON_ID	Connection ID, referenced on *MAT_SPOTWELD_DAIMLERCHRYSLER. Multiple sets of connection data may be used by assigning different connection IDs.
PROPRUL	The failure rule number for this connection.
AREAEQ	Area equation number for the connection area calculation. EQ.0: (default) area_true=area_modeled EQ.1: millimeter form; see Remark 4 EQ.-1: meter form; see Remark 4
DG_TYP	Damage type EQ.0: no damage function is used EQ.1: strain based damage EQ.2: failure function based damage
D_SIGY	Default yield stress for the spot weld element.
D_ETAN	Default tangent modulus for the spot weld element.
D_DG_PR	Default damage parameter for hyperbolic based damage function.
D_RANK	Default rank value.

DEFINE**DEFINE_CONNECTION_PROPERTIES**

VARIABLE	DESCRIPTION
D_SN	Default normal strength.
D_SB	Default bending strength.
D_SS	Default shear strength.
D_EXSN	Default exponent on normal stress term.
D_EXSB	Default exponent on bending stress term.
D_EXSS	Default exponent on shear stress term.
D_LCSN	Default curve ID for normal strength scale factor as a function of strain rate.
D_LCSB	Default curve ID for bending strength scale factor as a function of strain rate.
D_LCSS	Default curve ID for shear strength scale factor as a function of strain rate.
MID	Material ID of the shell material for which properties are defined.
SIGY	Yield stress to be used in the spot weld element calculation.
ETAN	Tangent modulus to be used in the spot weld element calculation.
DG_PR	Damage parameter for hyperbolic based damage function.
RANK	Rank value. See Remark 4.
SN	Normal strength.
SB	Bending strength.
SS	Shear strength.
EXSN	Exponent on normal stress term.
EXSB	Exponent on bending stress term.
EXSS	Exponent on shear stress term.
LCSN	Curve ID for normal strength scale factor as a function of strain rate.
LCSB	Curve ID for bending strength scale factor as a function of strain rate.
LCSS	Curve ID for shear strength scale factor as a function of strain rate.

Remarks:

1. This keyword is used only with *MAT_SPOTWELD_DAIMLERCHRYSLER. The data input is used in a 3 parameter failure model. Each solid spot weld element connects shell elements that may have the same or different materials. The failure model assumes that failure of the spot weld depends on the properties of the welded materials, so this keyword allows shell material specific data to be input for the connection. The default data will be used for any spot weld connected to a shell material that does not have material specific data defined, so it is not necessary to define material specific data for all welded shell materials.
2. To simplify data input, the ADD keyword option allows material specific data to be added to an existing *DEFINE_CONNECTION_PROPERTIES table. To use the ADD option, omit cards 2 and 3, and input only CON_ID on card 1. Then use cards 4 and 5 to input material specific data. For each unique CON_ID, control parameters and default values must be input in one set of *DEFINE_CONNECTION_PROPERTIES data. The same CON_ID may be used for any number of sets of material specific data input with the ADD option.

3. The three parameter failure function is

$$f = \left(\frac{\sigma_n}{\sigma_n^F}\right)^{m_n} + \left(\frac{\sigma_b}{\sigma_b^F}\right)^{m_b} + \left(\frac{\tau}{\tau^F}\right)^{m_\tau} - 1$$

where the three strength terms are SN, SB, and SS, and the three exponents are EXSN, EXSB, and EXSS. The strengths may be a function of strain rate by using the load curves, LCSN, LCSB, and LCSS. The peak stresses in the numerators are calculated from force resultants and simple beam theory.

$$\sigma_n = \frac{N_{rr}}{A} \quad \sigma_b = \frac{\sqrt{M_{rs}^2 + M_{rt}^2}}{Z} \quad \tau = \frac{M_{rr}}{2Z} + \frac{\sqrt{N_{rs}^2 + N_{rt}^2}}{A}$$

where the area is the cross section area of the weld element and Z is given by:

$$Z = \pi \frac{d^3}{32}$$

where *d* is the equivalent diameter of the solid spot weld element assuming a circular cross section.

4. There are three control parameters that define how the table data will be used for the connection, PROPRUL, AREA_EQ, and DG_TYP. PROPRUL determines how the parameters will be used. Because each weld connects two shell surfaces, one weld can have two sets of failure data as well as two values for ETAN and SIGY. At present, a single rule is implemented and the data with the lower RANK will be used.

The second control parameter is AREA_EQ which specifies a rule for calculating a true weld cross section area, A_{true} to be used in the failure function in place of the modeled solid element area, A . For AREA_EQ=1, A_{true} is calculated by

$$A_{true} = \frac{\pi}{4} \left(5 \sqrt{t_{min_shell}} \right)^2$$

where t_{min_shell} is the thickness of the welded shell surface that has the smaller thickness. For AREA_EQ=-1, A_{true} is calculated by

$$A_{true} = \frac{\pi}{4} \left(\frac{5}{1000} \sqrt{1000 * t_{min_shell}} \right)^2$$

The equation for AREA_EQ=1 is valid only for a length unit of millimeters, and AREA_EQ=-1 is valid only for a length unit of meters.

The third control parameter, DG_TYP, chooses from two available damage types. For DG_TYP=0, damage is turned off and the weld fails immediately when $f \geq 0$. For DG_TYP>0, damage is initiated when $f \geq 0$ and complete failure occurs when $\omega \geq 1$. For DG_TYP=1, damage growth is a function of plastic strain:

$$\omega = \frac{\mathcal{E}_{eff}^p - \mathcal{E}_{failure}^p}{\mathcal{E}_{rupture}^p - \mathcal{E}_{failure}^p} \quad \text{if} \quad \mathcal{E}_{failure}^p \leq \mathcal{E}_{eff}^p \leq \mathcal{E}_{rupture}^p$$

where \mathcal{E}_{eff}^p is the effective plastic strain in the weld material. When the value of the failure function first exceeds zero, the plastic strain at failure $\mathcal{E}_{failure}^p$ is set to the current plastic strain, and the rupture strain is offset from the plastic strain at failure by

$$\mathcal{E}_{rupture}^p = \mathcal{E}_{failure}^p + RS - EFAIL$$

where RS and EFAIL are the rupture strain and plastic strain at failure which are input on the *MAT_SPOTWELD_DAIMLERCHRYSLER card. If failure occurs when the plastic strain is zero, the weld material yield stress is reduced to the current effective stress such that damage can progress.

For DG_TYP=2, damage is a function of the failure function, f :

$$\omega = \frac{f}{f_{rupture}} \quad \text{if} \quad f \geq 0$$

where $f_{rupture}$ is the value of the failure function at rupture which is defined by

$$f_{rupture} = RS-EFAIL$$

and RS and EFAIL are input on the *MAT_SPOTWELD_DAIMLERCHRYSLER card.

Because the DG_TYP=1 damage function is scaled by plastic strain, it will monotonically increase in time. The DG_TYP=2 damage function is forced to be a monotonically increasing function in time by using the maximum of the current value and the maximum previous value. For both DG_TYP=1 and DG_TYP=2, the stress scale factor is then calculated by

$$\bar{\sigma} = \frac{DG_PR(1-\omega)}{\omega\left(\frac{1}{2} + \sqrt{\frac{1}{4} + DG_PR}\right) + DG_PR} \sigma$$

This equation becomes nearly linear at the default value of DG_PR which is 1.0e+10.

***DEFINE**

***DEFINE_ CONSTRUCTION_STAGES**

***DEFINE_ CONSTRUCTION_STAGES**

Purpose: Define times and durations of construction stages.

Note: This keyword card will be available starting in release 3 of version 971.

Card	1	2	3	4	5	6	7	8
Variable	ISTAGE	ATS	ATE	ATR	RTS	RTE		
Type	I	F	F	F	F	F		
Default	none	0.0	0.0	none	=ATS	=ATE		

VARIABLE

DESCRIPTION

ISTAGE	Stage ID
ATS	Analysis time at start of stage
ATE	Analysis time at end of stage
ATR	Analysis time duration of ramp
RTS	Real time at start of stage
RTE	Real time at end of stage

Remarks:

See also *CONTROL_ CONSTRUCTION_STAGES and *DEFINE_ STAGED_ CONSTRUCTION_ PART.

The first stage should start at time zero. There must be no gaps between stages, i.e. ATS for each stage must be the same as ATE for the previous stage.

The ramp time allows gravity loading and part stiffening/removal to be applied gradually during the first time period ATR of the construction stage.

The analysis always runs in “analysis time” – typically measured in seconds. The “real time” is used only as a number to appear on output plots and graphs, and is completely arbitrary. A dynain file is written at the end of each stage.

***DEFINE_CONTACT_VOLUME**

Purpose: Define a rectangular, a cylindrical, or a spherical volume in a local coordinate system. Nodes and segments which belong to specified part ID's and lie inside of the defined volume are used in the treatment of contact.

Card 1 1 2 3 4 5 6 7 8

Variable	CVID	CID	TYPE	XC	YC	ZC		
Type	I	I	I	F	F	F		
Default	0	0	0	0.	0.	0.		

For type=0, rectangular prism

Card 2 1 2 3 4 5 6 7 8

Variable	XMN	XXM	YMN	YMX	ZMN	ZMX		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

For type=1, cylindrical volume

Card 2 1 2 3 4 5 6 7 8

Variable	LENGTH	RINNER	ROUTER	D_ANGC				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

DEFINE**DEFINE_CONTACT_VOLUME****For type=3, spherical volume**

Card 2 1 2 3 4 5 6 7 8

Variable	RINNER	ROUTER	D_ANGS					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE**DESCRIPTION**

CVID	Contact volume ID
CID	Coordinate system ID. Required for rectangular and cylindrical volumes
TYPE	Volume type. Set to 0 for rectangular, 1 for cylindrical, and 2 for spherical.
XC	x-coordinate which defines the origin of coordinate system or the center of the sphere for type=3 referenced to the global coordinate system.
YC	y-coordinate which defines the origin of coordinate system or the center of the sphere for type=3 referenced to the global coordinate system.
ZC	z-coordinate which defines the origin of coordinate system or the center of the sphere for type=3 referenced to the global coordinate system.
XMN	Minimum x-coordinate in local coordinate system.
XMN	Maximum x-coordinate in local coordinate system.
YMN	Minimum y-coordinate in local coordinate system.
YMN	Maximum y-coordinate in local coordinate system.
ZMN	Minimum z-coordinate in local coordinate system.
ZMN	Maximum z-coordinate in local coordinate system.
LENGTH	Length of cylinder originating at (XC,YC,ZC) and revolving around the local x-axis.
RINNER	Inner radius of cylinder or sphere.
ROUTER	Outer radius of cylinder or sphere.

VARIABLE	DESCRIPTION
D_ANGC	If the included angle between the axis of the cylinder and the normal vector to the contact segment is <i>less</i> than this angle, the segment is deleted.
D_ANGS	If the included angle between a line draw from the center of the sphere to the centroid of the segment, and the normal vector to the contact segment is <i>greater</i> than this angle, the segment is deleted.

*DEFINE

*DEFINE_COORDINATE_NODES

*DEFINE_COORDINATE_NODES

Purpose: Define a local coordinate system with three node numbers. The local cartesian coordinate system is defined in the following steps. If the primary direction is along the x-axis, then the z-axis is computed from the cross product of x and \bar{y} , (see Figure 11.2), $z = x \times \bar{y}$, then the y-axis is computed via $y = z \times x$. A similar procedure applies if the local axis is along the y or z axes. The DIR option below applies to the third release of 971 and later versions.

Card	1	2	3	4	5	6	7	8
Variable	CID	N1	N2	N3	FLAG	DIR		
Type	I	I	I	I	I	A		
Default	0	0	0	0	0	X		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CID	Coordinate system ID. A unique number has to be defined.
N1	ID of node located at local origin.
N2	ID of node located along local x-axis if DIR=X, the y-axis if DIR=Y, and along the z axis if DIR=Z.
N3	ID of node located in local x-y plane if DIR=X, the local y-z plane if DIR=Y, and the local z-x plane if DIR=Z.
FLAG	Set to unity, 1, if the local system is to be updated each time step for the BOUNDARY_SPC nodal constraints and ELEMENT_BEAM type 6, the discrete beam element. Generally, this option when used with nodal SPC's is <i>not recommended</i> since it can cause excursions in the energy balance because the constraint forces at the node may go through a displacement if the node is partially constrained
DIR	Axis defined by node N2 moving from the origin node N1. The default direction is the x-axis.

Remarks:

1. The nodes N1, N2, and N3 must be separated by a reasonable distance and not colinear to avoid numerical inaccuracies.

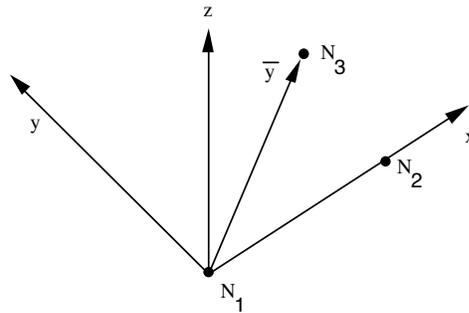


Figure 11.1. Definition of local coordinate system using three nodes when the node N2 lies along the x-axis.

*DEFINE

*DEFINE_COORDINATE_SYSTEM

*DEFINE_COORDINATE_SYSTEM

Purpose: Define a local coordinate system with three points. The same procedure as described in Figure 11.1, see *DEFINE_COORDINATE_NODES, is used. The coordinates of the nodes are given instead. N_1 is defined by (X_0, Y_0, Z_0) , N_2 is defined by (X_L, Y_L, Z_L) , and N_3 by (X_P, Y_P, Z_P) .

Card 1 of 2 - Required.

Card 1 1 2 3 4 5 6 7 8

Variable	CID	XO	YO	ZO	XL	YL	ZL	
Type	I	F	F	F	F	F	F	
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	
Remarks								

Card 2 of 2 - Required.

Card 2 1 2 3 4 5 6 7 8

Variable	XP	YP	ZP					
Type	F	F	F					
Default	0.0	0.0	0.0					
Remarks								

VARIABLE

DESCRIPTION

CID	Coordinate system ID. A unique number has to be defined.
XO	X-coordinate of origin
YO	Y-coordinate of origin
ZO	Z-coordinate of origin
XL	X-coordinate of point on local x-axis

<u>VARIABLE</u>	<u>DESCRIPTION</u>
YL	Y-coordinate of point on local x-axis
ZL	Z-coordinate of point on local x-axis
XP	X-coordinate of point in local x-y plane
YP	Y-coordinate of point in local x-y plane
ZP	Z-coordinate of point in local x-y plane

Remarks:

1. The coordinates of the points must be separated by a reasonable distance and not colinear to avoid numerical inaccuracies.

*DEFINE

*DEFINE_COORDINATE_VECTOR

*DEFINE_COORDINATE_VECTOR

Purpose: Define a local coordinate system with two vectors, see Figure 11.2. The vector cross product, $z = x \times y$, determines the z-axis. The y-axis is then given by $y = z \times x$. If this coordinate system is assigned to a nodal point, then at each time step during the calculation, the coordinate system is incrementally rotated using the angular velocity of the nodal point to which it is assigned.

Card	1	2	3	4	5	6	7	8
Variable	CID	XX	YX	ZX	XV	YV	ZV	NID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0.

VARIABLE

DESCRIPTION

CID	Coordinate system ID. A unique number has to be defined.
XX	X-coordinate on local x-axis. Origin lies at (0,0,0).
YX	Y-coordinate on local x-axis
ZX	Z-coordinate on local x-axis
XV	X-coordinate of local x-y vector
YV	Y-coordinate of local x-y vector
ZV	Z-coordinate of local x-y vector
NID	Optional nodal point ID. The coordinate system rotates with the rotation of this node. If the node is not defined, the coordinate system is stationary.

Remarks:

1. These vectors should be separated by a reasonable included angle to avoid numerical inaccuracies.
2. Ideally, this nodal point should be attached to a rigid body or a structural part where the nodal point angular velocities are meaningful. It should be noted that angular velocities of nodes may not be meaningful if the nodal point is attached only to solid elements and

even to shell elements where the drilling degree of freedom may be singular, which is likely in flat geometries.

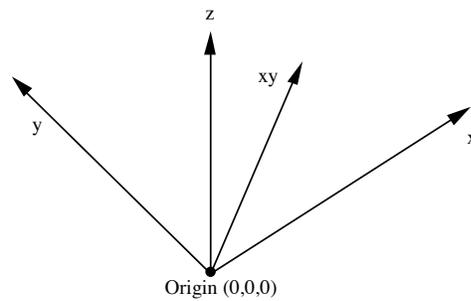


Figure 11.2. Definition of the coordinate system with two vectors.

*DEFINE

*DEFINE_CURVE

*DEFINE_CURVE_{OPTION}

Purpose: Define a curve [for example, load (ordinate value) versus time (abscissa value)], often referred to as a load curve.

Curves are discretized internally with equal intervals along the abscissa for fast evaluation in constitutive models. Discretized curves are not used for evaluating loading conditions. Also, see remark 1 below. To improve the accuracy of the discretized curves in later releases of version 970 changes were made in the discretization process. These changes had the unexpected effect of changing the results generated with validated models such as barriers and occupants. Consequently, *OPTION* was added to make available the old discretization if needed for the validated models.

Available options include:

<OPTION>

3858

5434a

which correspond to the first releases of version 970 and the 2005 release, respectively.

Since input errors and wrong results are sometimes related to load curve usage, a “*Load curve usage*” table is printed in the D3HSP file after all the input is read. This table should be checked to insure that each curve ID is referenced by the option for which the curve is intended.

Card 1 1 2 3 4 5 6 7 8

Variable	LCID	SIDR	SFA	SFO	OFFA	OFFO	DATTYP	
Type	I	I	F	F	F	F	I	
Default	none	0	1.	1.	0.	0.	0	

Card 2, 3, 4, etc. Put one pair of points per card (2E20.0). Input is terminated when a “*” card is found. (Use only two points for applying loads if the implicit arc-length method is active.)

Card 2... 1 2 3 4 5 6 7 8

Variable	A1	O1		
Type	F	F		
Default	0.0	0.0		

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.
SIDR	Stress initialization by dynamic relaxation: EQ.0: load curve used in transient analysis only or for other applications, EQ.1: load curve used in stress initialization but not transient analysis, EQ.2: load curve applies to both initialization and transient analysis.
SFA	Scale factor for abscissa value. This is useful for simple modifications. EQ.0.0: default set to 1.0.
SFO	Scale factor for ordinate value (function). This is useful for simple modifications. EQ.0.0: default set to 1.0.
OFFA	Offset for abscissa values, see explanation below.
OFFO	Offset for ordinate values (function), see explanation below.
DATTYP	Data type. Usually 0, set to 1 <u>only</u> for general xy data. This affects how offsets are applied. General xy data curves refer to curves whose abscissa values do not increase monotonically. Generally, DATTYP=0 for time dependent curves, force versus displacement curves, and stress strain curves.
A1, A2,...	Abscissa values. Only pairs have to be defined, see remarks below.
O1, O2,...	Ordinate (function) values. Only pairs have to be defined, see remarks below.

Remarks:

- Warning:** In the definition of Load Curves used in the constitutive models, reasonable spacing of the points should always be observed, i.e., never set a single point off to a value approaching infinity. LS-DYNA uses internally discretized curves to improve efficiency in the constitutive models. Also, since the constitutive models extrapolate the curves, it is important to ensure that extrapolation does not lead to physically meaningless values, such as a negative flow stress.
- The load curve values are scaled after the offsets are applied, i.e.:

$$Abscissa\ value = SFA \cdot (Defined\ value + OFFA)$$

$$Ordinate\ value = SFO \cdot (Defined\ value + OFFO)$$

3. Positive offsets for the load curves (DATTYP=0) are intended for time versus function curves since two additional points are generated automatically at time zero and at time $.999*OFFA$ with the function values set to zero. If DATTYP=1, then the offsets do not create these additional points. Negative offsets for the abscissa simply shifts the abscissa values without creating additional points.
4. Load curves are not extrapolated by LS-DYNA for applied loads such as pressures, concentrated forces, displacement boundary conditions, etc. Function values are set to zero if the time, etc., goes off scale. Therefore, extreme care must be observed when defining load curves. In the constitutive models, extrapolation is employed if the values on the abscissa go off scale.
5. The load curve offsets and scale factors are ignored during restarts if the curve is redefined. See *CHANGE_CURVE_DEFINITION in the restart section.

***DEFINE_CURVE_COMPENSATION**

Purpose: To define a curve for local compensation. All elements inside or outside of a curve can be compensated locally with a transitional region. This keyword must be used with *INTERFACE_COMPENSATION_NEW.

Card 1 1 2 3 4 5 6 7 8

Variable	INOUT							
Type	I							
Default	none							

Card 2

Variable	X	Y	Z					
Type	F	F	F					
Default	none	none	none					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
INOUT	EQ.1: elements inside the curve will be compensated EQ.2: elements outside the curve will be compensated
X,Y,Z	Coordinates of curve points

*DEFINE

*DEFINE_CURVE_DRAWBEAD

*DEFINE_CURVE_DRAWBEAD

Purpose: To facilitate the definition of drawbead.

Card 1 2 3 4 5 6 7 8

Variable	CID	TCTYPE	VID	PID	BLKID	PERCT		
Type	I	I	I	I	I			
Default								

VARIABLE

DESCRIPTION

CID	Curve ID
TYPE	Bead date type EQ.1: x,y,z data EQ.2: IGES data
VID	Vector ID, See DEFINE_VECTOR. This vector is used to project the bead to the rigid part (PID)
PID	Part ID to attach the drawbead
BLKID	Blank ID
PERCT	Percentage of restraining force (the ratio of restraining force over Lock force). The value should be between 0 and 100.

***DEFINE_CURVE_ENTITY**

Purpose: Define a curve of straight line segments and circular arcs that defines an axisymmetric surface. This curve can only be used with the keyword, *CONTACT_ENTITY for the load curve entity, GEOTYP=11. This option is in the third release.

Card 1 2 3 4 5 6 7 8

Variable	LCID	SFA	SFO	SFR	OFFA	OFFO	OFFR	
Type	I	F	F	F	F	F	F	
Default	none	1.	1.	1.	0.	0.	0.	

Card 2, 3, 4, etc. Put one pair of points per card (3E20.0,I20). Input is terminated when a “*” card is found.

Card 1 2 3 4 5 6 7 8

Variable	Ai	Oi	Ri	IFLAG
Type	F	F	F	I
Default	0.0	0.0	optional	Required if Ri >0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.
SFA	Scale factor for axis value. This is useful for simple modifications. EQ.0.0: default set to 1.0.
SFO	Scale factor for radius values. This is useful for simple modifications. EQ.0.0: default set to 1.0.
SFR	Scale factor for circular radius. This is useful for simple modifications. EQ.0.0: default set to 1.0.
OFFA	Offset for axis values, see explanation below.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
OFFO	Offset for radius values, see explanation below.
OFFR	Offset for circular radius, see explanation below.
Ai	Z-axis coordinates along the axis of rotation.
Oi	Radial coordinates from the axis of rotation
Ri	Radius of arc between points (Ai,Oi) and (Ai+1,Oi+1). If zero, a straight line segment is assumed.
IFLAG	Defined if Ri >0. Set to 1 if center of arc is inside axisymmetric surface and to -1 if the center is outside the axisymmetric surface.

Remarks:

1. The load curve values are scaled after the offsets are applied, i.e.:

$$\text{Axis value} = SFA \cdot (\text{Defined value} + OFFA)$$

$$\text{Radius value} = SFO \cdot (\text{Defined value} + OFFO)$$

$$\text{Circular radius} = SFR \cdot (\text{Defined value} + OFFR)$$

DEFINE_CURVE_FEEDBACK**DEFINE*****DEFINE_CURVE_FEEDBACK**

Purpose: Define information that is used as the solution evolves to scale the ordinate values of the specified load curve ID. One application for this capability is in sheet metal stamping.

Card 1 1 2 3 4 5 6 7 8

Variable	LCID	PID	BOXID	FLDID				
Type	I	I	I	I				
Default	none	none	0	none				

Card 2

Variable	FSL	TSL	SFF	SFT	BIAS			
Type	F	F	F	F	F			
Default	none	none	1.0	1.0	0.0			

VARIABLE**DESCRIPTION**

LCID	ID number for load curve to be scaled.
PID	Active part ID for load curve control
BOXID	Box ID. Elements of specified part ID contained in box are checked. If the box ID is set to zero the all elements of the active part are checked.
FLDID	Load curve ID which defines the flow limit diagram as shown in Figure 11.3.
FSL	If the strain ratio, $\epsilon_{major,workpiece} / \epsilon_{major,fl}$ exceeds <i>FSL</i> , the scale factor for flow, <i>SF</i> , is active.
TSL	Thickness strain limit. If the through thickness strain is exceeded the scale factor for thickening, <i>ST</i> , is active.
SFF	Scale factor for the flow limit diagram, <i>SF</i> (Default=1.0).

VARIABLE	DESCRIPTION
SFT	Scale factor for thickening, ST (Default=1.0).
BIAS	Bias for combined flow and thickening, S , $-1 \leq S \leq 1$.

Remarks:

The scale factor for the load curve ordinate value is updated as:

$$S_{load\ curve}^{n+1} = S_{load\ curve}^n \cdot S_{final}$$

where S_{final} is equal to SF if the strain ratio is exceeded or to ST if the thickness strain limit is exceeded. The bias value determines the final scale factor, S_{final} , in the event that the thickness and flow limit diagram criteria both satisfied. In this case the scale factor for the load curve is given by:

$$S_{final} = \frac{1}{2}(1 - S) \cdot SF + \frac{1}{2}(1 + S)ST$$

Generally, SF is slightly less than unity and ST is slightly greater than unity so that $S_{load\ curve}$ changes insignificantly from time step to time step.

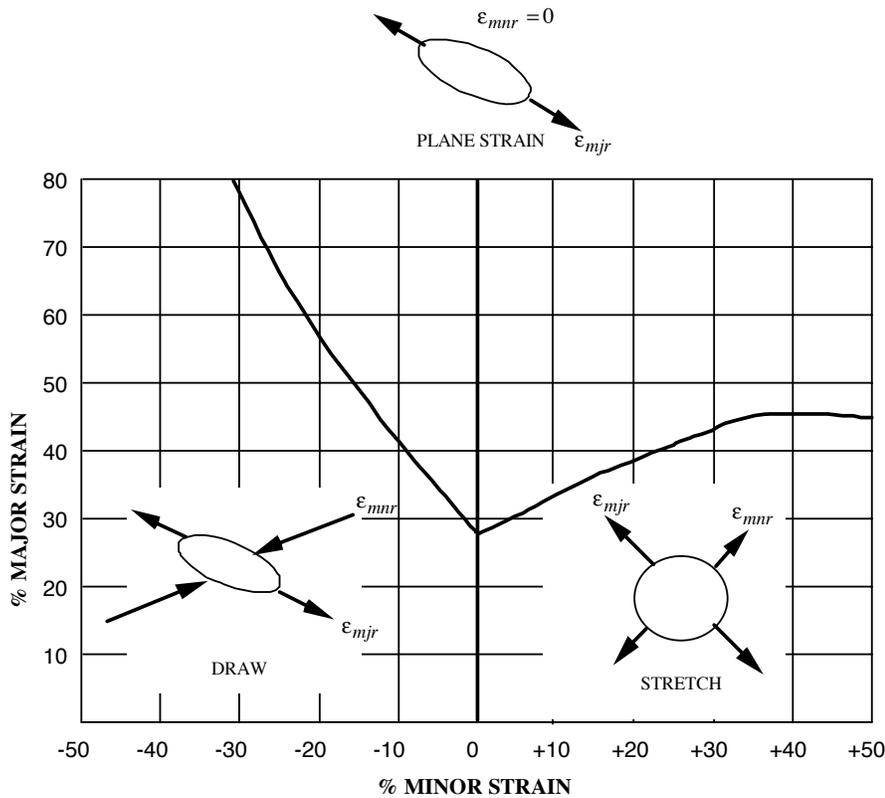


Figure 11.3. Flow limit diagram

***DEFINE_CURVE_FUNCTION**

Purpose: Define a curve [for example, load (ordinate value) versus time (abscissa value)] where the ordinate is given by a function expression. The function can reference other curve definition, kinematical quantities, forces, interpolating polynomials, intrinsic functions, and combinations thereof. Please note that many functions require the definition of a local coordinate system (see Remark 1 below).

Card 1 2 3 4 5 6 7 8

Variable	LCID	SIDR	SFA	SFO	OFFA	OFFO	DATTYP	
Type	I	I	F	F	F	F	I	
Default	none	0	1.	1.	0.	0.	0	

Card 2, 3, 4, etc. (not to exceed 10). These cards are combined to form a single line of input. The next “*” terminates the input.

Card

Variable	FUNCTION
Type	C
Remarks	1

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.
SIDR	Stress initialization by dynamic relaxation: EQ.0: load curve used in transient analysis only or for other applications, EQ.1: load curve used in stress initialization but not transient analysis, EQ.2: load curve applies to both initialization and transient analysis.

***DEFINE**

***DEFINE_CURVE_FUNCTION**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SFA	Scale factor for abscissa value. This is useful for simple modifications. EQ.0.0: default set to 1.0.
SFO	Scale factor for ordinate value (function). This is useful for simple modifications. EQ.0.0: default set to 1.0.
OFFA	Offset for abscissa values, see explanation below.
OFFO	Offset for ordinate values (function), see explanation below.
DATTYP	Data type. Usually 0, set to 1 <u>only</u> for general xy data. This affects how offsets are applied. General xy data curves refer to curves whose abscissa values do not increase monotonically. Generally, DATTYP=0 for time dependent curves, force versus displacement curves, and stress strain curves.
FUNCTION	Arithmetic expression involving a combination of the following possibilities.

Constants and Variables

<u>FUNCTION NAME</u>	<u>DESCRIPTION</u>
TIME	Current simulation time
PI	Proportionality constant relating the circumference of a circle to its diameter
DTOR	Degrees to radians conversion factor (PI/180.)
RTOD	Radians to degrees conversion factor (180./PI)

Intrinsic Functions

<u>FUNCTION NAME</u>	<u>DESCRIPTION</u>
ABS(a)	Absolute value of (a)
AINT(a)	Nearest integer whose magnitude is not larger than (a)
ANINT(a)	Nearest whole number to (a)
MOD(a1,a2)	Remainder when a1 is divided by a2
SIGN(a1,a2)	Transfer sign of a2 to magnitude of a1

MAX(a1,a2)	Maximum of a1 and a2
MIN(a1,a2)	Minimum of a1 and a2
SQRT(a)	Square root of (a)
EXP(a)	e raised to the power of (a)
LOG(a)	Natural logarithm of (a)
LOG10(a)	Log base 10 of (a)
SIN(a)	Sine of (a)
COS(a)	Cosine of (a)
TAN(a)	Tangent of (a)
ASIN(a)	Arc sine of (a)
ACOS(a)	Arc cosine of (a)
ATAN(a)	Arc tangent of (a)
ATAN2(a1,a2)	Arc tangent of (a1/a2)
SINH(a)	Hyperbolic sine of (a)
COSH(a)	Hyperbolic cosine of (a)
TANH(a)	Hyperbolic tangent of (a)

Load Curves

<u>FUNCTION NAME</u>	<u>DESCRIPTION</u>
LCn	Ordinate value of curve n defined elsewhere (see *DEFINE_CURVE)

Coordinate Functions

<u>FUNCTION NAME</u>	<u>DESCRIPTION</u>
CX(n1)	Value of x-coordinate for node n1.
CY(n1)	Value of y-coordinate for node n1.
CZ(n1)	Value of z-coordinate for node n1.

Displacement Functions

FUNCTION NAME	DESCRIPTION
DM(n1[,n2])	Magnitude of translational displacement of node n1 relative to node n2. Node n2 is optional and if omitted the displacement is computed relative to ground.
DX(n1[,n2,n3])	x-translational displacement of node n1 relative to node n2 expressed in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. If node n3 is not specified the displacement is reported in the global coordinate system.
DY(n1[,n2,n3])	y-translational displacement of node n1 relative to node n2 expressed in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. If node n3 is not specified the displacement is reported in the global coordinate system.
DZ(n1[,n2,n3])	z-translational displacement of node n1 relative to node n2 expressed in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. If node n3 is not specified the displacement is reported in the global coordinate system.
AX(n1[,n2])	Rotation displacement of node n1 about the local x-axis of node n2. If n2 is not specified then it defaults to ground. In computing this value it is assumed the rotation about the other two axes (y-, z-axes) of node n2 is zero.
AY(n1[,n2])	Rotation displacement of node n1 about the local y-axis of node n2. If n2 is not specified then it defaults to ground. In computing this value it is assumed the rotation about the other two axes (x-, z-axes) of node n2 is zero.
AZ(n1[,n2])	Rotation displacement of node n1 about the local z-axis of node n2. If n2 is not specified then it defaults to ground. In computing this value it is assumed the rotation about the other two axes (x-, y-axes) of node n2 is zero.
PSI(n1[,n2])	First angle in the body2:313 Euler rotation sequence which orients node n1 in the frame of node n2. If n2 is omitted it defaults to ground.
THETA(n1[,n2])	Second angle in the body2:313 Euler rotation sequence which orients node n1 in the frame of node n2. If n2 is omitted it defaults to ground.
PHI(n1[,n2])	Third angle in the body2:313 Euler rotation sequence which orients node n1 in the frame of node n2. If n2 is omitted it defaults to ground.
YAW(n1[,n2])	First angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n1 in the frame of node n2. If n2 is omitted it defaults to ground.

PITCH(n1[,n2])	Second angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n1 in the frame of node n2. If n2 is omitted it defaults to ground.
ROLL(n1[,n2])	Third angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n1 in the frame of node n2. If n2 is omitted it defaults to ground.

Velocity Functions

FUNCTION NAME	DESCRIPTION
VM(n1[,n2])	Magnitude of translational velocity of node n1 relative to node n2. Node n2 is optional and if omitted the velocity is computed relative to ground.
VR(n1[,n2])	Relative radial translational velocity of node n1 relative to node. If node n2 is omitted it defaults to ground.
VX(n1[,n2,n3])	x-component of the difference between the translational velocity vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
VY(n1[,n2,n3])	y-component of the difference between the translational velocity vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
VZ(n1[,n2,n3])	z-component of the difference between the translational velocity vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
WM(n1[,n2])	Magnitude of angular velocity of node n1 relative to node n2. Node n2 is optional and if omitted the angular velocity is computed relative to ground.
WX(n1[,n2,n3])	x-component of the difference between the angular velocity vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
WY(n1[,n2,n3])	y-component of the difference between the angular velocity vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted it defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
WZ(n1[,n2,n3])	z-component of the difference between the angular velocity vectors of node n1 and node n2 in the local coordinate system of node n3. If node

n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.

Acceleration Functions

FUNCTION NAME	DESCRIPTION
ACCM(n1[,n2])	Magnitude of translational acceleration of node n1 relative to node n2. Node n2 is optional and if omitted the acceleration is computed relative to ground.
ACCX(n1[,n2,n3])	x-component of the difference between the translational acceleration vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
ACCY(n1[,n2,n3])	y-component of the difference between the translational acceleration vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
ACCZ(n1[,n2,n3])	z-component of the difference between the translational acceleration vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
WDTM(n1[,n2])	Magnitude of angular acceleration of node n1 relative to node n2. Node n2 is optional and if omitted the angular acceleration is computed relative to ground.
WDTX(n1[,n2,n3])	x-component of the difference between the angular acceleration vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
WDTY(n1[,n2,n3])	y-component of the difference between the angular acceleration vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.
WDTZ(n1[,n2,n3])	z-component of the difference between the angular acceleration vectors of node n1 and node n2 in the local coordinate system of node n3. If node n2 is omitted if defaults to ground. Node n3 is optional and if not specified the global coordinate system is used.

Generic Force Functions

FUNCTION NAME	DESCRIPTION
FM(n1[,n2])	Magnitude of net translational force acting between node n1 and n2. Node n2 is optional and if omitted the force that acting only on n1.
FX(n1[,n2,n3])	x-component of the net translational force acting at node n1 as computed in the optional local system of node n3. If n2 is specified then all forces acting between n1 and n2 are included in the force computation.
FY(n1[,n2,n3])	y-component of the of the net translational force acting at node n1 as computed in the optional local system of node n3. If n2 is specified then all forces acting between n1 and n2 are included in the force computation.
FZ(n1[,n2,n3])	z-component of the of the net translational force acting at node n1 as computed in the optional local system of node n3. If n2 is specified then all forces acting between n1 and n2 are included in the force computation.
TM(n1[,n2])	Magnitude of net torque acting between node n1 and n2. Node n2 is optional and if omitted the torque that acting only on n1.
TX(n1[,n2,n3])	x-component of the net torque acting at node n1 as computed in the optional local system of node n3. If n2 is specified then all forces acting between n1 and n2 are included in the torque computation.
TY(n1[,n2,n3])	y-component of the net torque acting at node n1 as computed in the optional local system of node n3. If n2 is specified then all forces acting between n1 and n2 are included in the torque computation.
TZ(n1[,n2,n3])	z-component of the net torque acting at node n1 as computed in the optional local system of node n3. If n2 is specified then all forces acting between n1 and n2 are included in the torque computation.

Element Specific Force Functions

FUNCTION NAME	DESCRIPTION
BEAM(id,jflag,comp,rm)	Returns the force component <i>comp</i> (see description below) of beam <i>id</i> as calculated in the local coordinate system <i>rm</i> . Forces are reported in the global coordinate system is <i>rm</i> is zero. If <i>rm</i> equals -1 the beam's r, s, and t force/moment is returned. If <i>jflag</i> is set to zero then the force/torque acting on n1 end of the beam is returned, else if <i>jflag</i> is set to unity the force/torque on the n2 end of the beam is returned. See *ELEMENT_BEAM for the nodal connectivity rule defining n1 and n2.

Admissible values of *comp* are 1-8 and correspond to the following components.

- 1: force magnitude
- 2: x force (axial r-force, *rm*=-1)
- 3: y force (s-shear force, *rm*=-1)
- 4: z force (t-shear force, *rm*=-1)
- 5: torque magnitude
- 6: x torque (torsion, *rm*=-1)
- 7: y torque (s-moment, *rm*=-1)
- 8: z torque (t-moment, *rm*=-1)

JOINT(*id,jflag,comp,rm*) Returns the force component *comp* (see description below) due to rigid body joint *id* as calculated in the local coordinate system *rm*. If *jflag* is set to zero then the force/torque acting on n1 end of the joint is returned. The force/torque on the n2 end of the joint is returned if *jflag* is set to 1. See *CONSTRAINED_JOINT for the rule defining n1 and n2.

General Functions

FUNCTION NAME	DESCRIPTION
CHEBY(<i>x,x0,a0,...,a30</i>)	<p>Evaluates a Chebyshev polynomial at the user specified value <i>x</i>. The parameters <i>x0</i>, <i>a0</i>, <i>a1</i>, ..., <i>a30</i> are used to define the constants for the polynomial defined by:</p> $C(x) = \sum a_j T_j(x - x_0)$ <p>where the functions T_j is defined recursively as</p> $T_j(x - x_0) = 2 \cdot (x - x_0) \cdot T_{j-1}(x - x_0) - T_{j-2}(x - x_0)$ <p>where</p> $T_0(x - x_0) = 1$ $T_1(x - x_0) = x - x_0$
FORCOS(<i>x,x0,ω,a0,...,a30</i>)	<p>Evaluates a Fourier cosine series at the user specified value <i>x</i>. The parameters <i>x0</i>, <i>a0</i>, <i>a1</i>, ..., <i>a30</i> are used to define the constants for the series defined by:</p> $F(x) = \sum a_j T_j(x - x_0)$ <p>where</p> $T_j(x - x_0) = \cos[j \cdot \omega \cdot (x - x_0)]$

FORSIN(x,x0,ω,a0,...,a30) Evaluates a Fourier sine series at the user specified value x. The parameters x0, a0, a1, ..., a30 are used to define the constants for the series defined by:

$$F(x) = \sum a_j T_j(x - x_0)$$

where

$$T_j(x - x_0) = \sin[j \cdot \omega \cdot (x - x_0)]$$

POLYL(x,x0,a0,...,a30) Evaluates a standard polynomial at the user specified value x. The parameters x0, a0, a1, ..., a30 are used to define the constants for the polynomial defined by:

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \dots a_n(x - x_0)^n$$

SHF(x,x0,a,ω,phi,b) Evaluates a Fourier sine series at the user specified value x. The parameters x0, a0, a1, ..., a30 are used to define the constants for the series defined by:

$$SHF = a \cdot \sin[\omega \cdot (x - x_0) - phi] + b$$

STEP(x,x0,h0,x1,h1) Approximates the Heavyside function with a cubic polynomial using the equation:

$$STEP = \begin{cases} h_0 & x \leq x_0 \\ h_0 + (h_1 - h_0) \cdot [(x - x_0)/(x_1 - x_0)]^2 \cdot \{3 - 2 \cdot [(x - x_0)/(x_1 - x_0)]\}; x_0 < x < x_1 & x_0 < x < x_1 \\ h_1 & x \geq x_1 \end{cases}$$

Remarks:

1. A local coordinate system must be attached to nodes if they are referenced by functions involving rotational motion, for example, angular displacement or angular velocity. The local coordinate system is attached to the node using *DEFINE_COORDINATE_NODES where FLAG must be set equal to unity. Similarly, a local coordinate system must also be attached to node n3 if n3 is referenced in functions: DX, DY, DZ, VX, VY, VZ, WX, WY, WZ, ACCX, ACCY, ACCZ, WDTX, WDTY, WDTZ, FX, FY, FZ, TX, TY, or TZ.

*DEFINE

*DEFINE_CURVE_SMOOTH

*DEFINE_CURVE_SMOOTH

Purpose: Define a smoothly varying curve using few parameters. This shape is useful for velocity control of tools in metal forming applications.

Card 1 2 3 4 5 6 7 8

Variable	LCID	SIDR	DIST	TSTART	TEND	TRISE	V0	
Type	I	I	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE

DESCRIPTION

LCID	Load curve ID, must be unique.
SIDR	Stress initialization by dynamic relaxation: EQ.0: load curve used in transient analysis only or for other applications, EQ.1: load curve used in stress initialization but not transient analysis, EQ.2: load curve applies to both initialization and transient analysis.
DIST	Total distance tool will travel (area under curve).
TSTART	Time curve starts to rise
TEND	Time curve returns to zero. If TEND is nonzero, VMAX will be computed automatically to satisfy required travel distance DIST. Input either TEND or VMAX.
TRISE	Rise time
VMAX	Maximum velocity (maximum value of curve). If VMAX is nonzero, TEND will be computed automatically to satisfy required travel distance DIST. Input either TEND or VMAX.

Remarks:

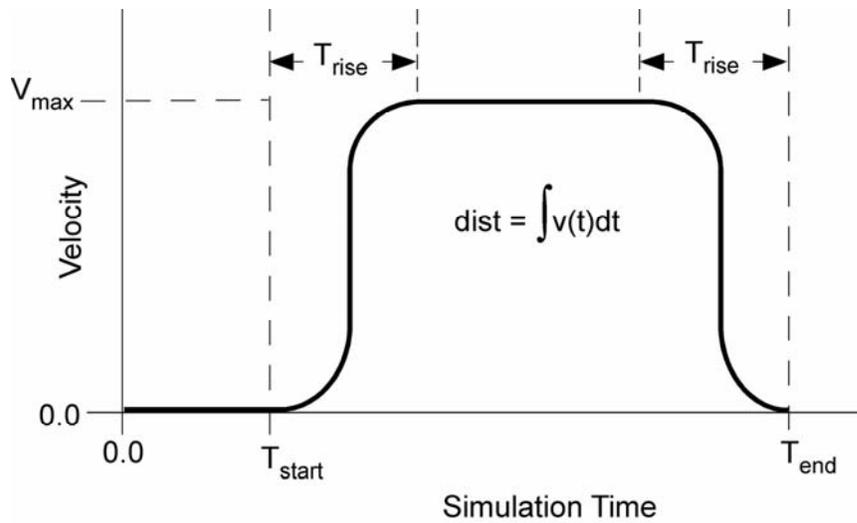


Figure 11.4. Smooth curve created automatically using *DEFINE_CURVE_SMOOTH. This shape is commonly used to control velocity of tools in metal forming applications as shown in the above graph, but can be used for other applications in place of any standard load curve.

*DEFINE

*DEFINE_CURVE_TRIM

*DEFINE_CURVE_TRIM_{OPTION}

Available options include:

<BLANK>

3D

Purpose: Define a curve for trimming. Also, see *INTERFACE_SPRINGBACK. When option 3D is used, the trimming will be processed based on the element normal rather than the vector

Card 1 1 2 3 4 5 6 7 8

Variable	TCID	TCTYPE	TFLG	TDIR	TCTOL	TOLN/IGB	NSEED	
Type	I	I	I	I	F	F	I	
Default	none	none	none	none	0.25	2.0	NONE	
Remarks	1,2,3			figure 11.5	4			

Card 2, 3, 4, etc. defined if and only if TCTYPE=1. Put one pair of points per card (2E20.0) Input is terminated when a “*” card is found.

Card 2... 1 2 3 4 5 6 7 8

Variable	CX	CY		
Type	F	F		
Default	0.0	0.0		
Type	C			

Defined if and only if TCTYPE=2.

Card 2 1 2 3 4 5 6 7 8

Variable	FILENAME								
Type	C								

VARIABLE	DESCRIPTION
TCID	ID number for trim curve.
TCTYPE	Trim curve type: EQ.1: digitized curve provided, EQ.2: IGES trim curve.
TFLG	Element removal option: EQ. -1: remove material outside curve, EQ. 1: remove material inside curve.
TDIR	ID of vector (*DEFINE_VECTOR) giving direction of projection for trim curve (see Figure 11.5). EQ. 0: default vector (0,0,1) is used. Curve is defined in global XY plane, and projected onto mesh in global Z-direction to define trim line.
TCTOL	Tolerance limiting size of small elements created during trimming (see Figure 11.6). LT.0: "simple" trimming, producing jagged edge mesh
TOLN	The maximum gap between the trimming curve and the mesh. If the gap is bigger than this value, this section in the curve will not be used. Used only option 3D is chosen, If option 3D is not used, then IGB.EQ.0: trimming curve is defined in local coordinate system IGB.EQ.1: trimming curve is defined in global coordinate system
NSEED	Any node in the side which will be kept after trimming. Used only when option 3D is chosen.
CX	x-coordinate of trim curve Defined if and only if TCTYPE=1.
CY	y-coordinate of trim curve Defined if and only if TCTYPE=1.
FILENAME	Name of IGES database containing trim curve(s). Defined if and only if TCTYPE=2.

Remarks:

1. This command in combination with *ELEMENT_TRIM trims the requested parts before the job starts.
2. If the command *ELEMENT_TRIM does not exist the parts are trimmed after the job is terminated.
3. Pre-trimming (*ELEMENT_TRIM + *DEFINE_CURVE_TRIM) can handle adaptive mesh and post-trimming. The keyword *DEFINE_CURVE_TRIM by itself cannot deal with an adaptive mesh. See the detailed procedure outlined in the Remarks in the Section *INTERFACE_SPRINGBACK.
4. The trimming tolerance TCTOL limits the size of the smallest element created during trimming. A value of 0.0 places no limit on element size. A value of 0.5 restricts new elements to be at least half of the size of the parent element. A value of 1.0 allows no new elements to be generated, only repositioning of existing nodes to lie on the trim curve. A negative tolerance value activates "simple" trimming, where entire elements are removed, leaving a jagged edge.

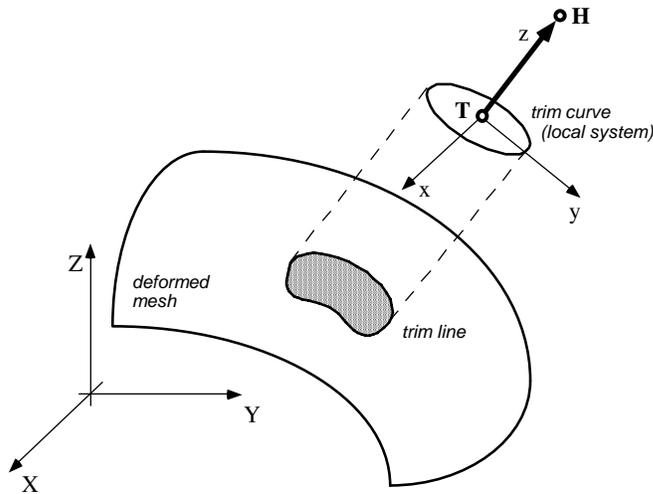


Figure 11.5. Trimming Orientation Vector. The tail (T) and head (H) points define a local coordinate system (x,y,z). The global coordinate system is named (X,Y,Z). The local x-direction is constructed in the Xz plane. If X and z nearly coincide ($|X \cdot z| > 0.95$), then the local x-direction is instead constructed in the Yz plane. Trim curve data is input in the x-y plane, and projected in the z-direction onto the deformed mesh to obtain the trim line.

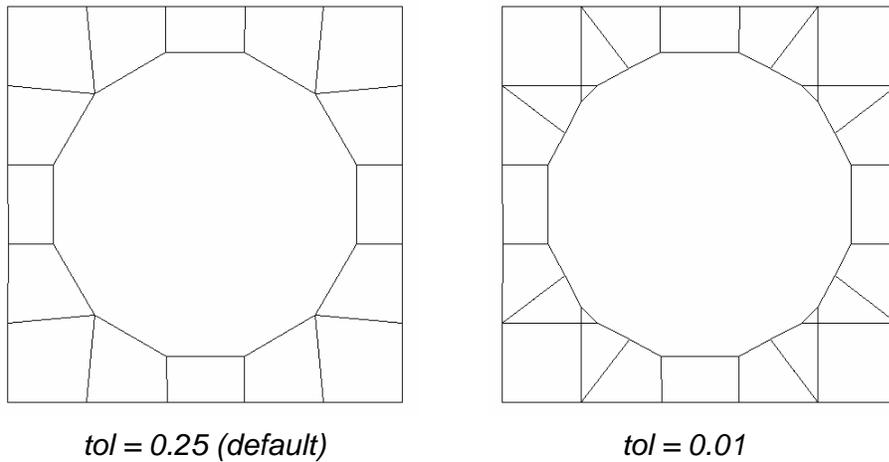


Figure 11.6 Trimming Tolerance. The tolerance limits the size of the small elements generated during trimming. The default tolerance (left) produces large elements. Using a tolerance of 0.01 (right) allows smaller elements, and more detail in the trim line.

*DEFINE

*DEFINE_DEATH_TIMES

*DEFINE_DEATH_TIMES_OPTION

Available options include:

NODES

SET

RIGID

Purpose: To dynamically define the death times for *BOUNDARY_PRESCRIBED_MOTION based on the locations of nodes and rigid bodies. Once a node or rigid body moves past a plane or a geometric entity, the death time is set to the current time. The input in this section continues until the next '*' card is detected.

Card 1 1 2 3 4 5 6 7 8

Variable	GEO	N1	N2	N3				
Type	I	I	I	I				
Default		0	0	0				

Card 2

Variable	X_T	Y_T	Z_T	X_H	Y_H	Z_H	R	FLAG
Type	F	F	F	F	F	F	F	
Default								1

Cards 3, ..., The next "*" card terminates the friction definition.

Card 3... 1 2 3 4 5 6 7 8

Variable	NSID1	NSID2	NSID3	NSID4	NSID5	NSID6	NSID7	NSID8
Type	I	I	I	I	I	I	I	I
Default								

VARIABLE	DESCRIPTION
GEO	Geometric entity type. =1 plane, =2 infinite cylinder, =3 sphere
N1	Node defining the origin of the geometric entity (optional).
N2	Node defining the tail of the orientation vector (optional).
N3	Node defining the head of the orientation vector (optional).
X_T	X coordinate of the origin of the geometric entity and the tail of the orientation vector.
Y_T	Y coordinate of the origin of the geometric entity and the tail of the orientation vector.
Z_T	Z coordinate of the origin of the geometric entity and the tail of the orientation vector.
X_H	X coordinate of the head of the orientation vector.
Y_H	Y coordinate of the head of the orientation vector.
Z_H	Z coordinate of the head of the orientation vector.
R	Radius of cylinder or sphere.
FLAG	+1 for killing motion when the node is outside of the geometric entity or on the positive side of the plane as defined by the normal direction, or -1 for the inside.
NSIDi	i-th node, node set, or rigid body

Remarks:

1. Either N1 or X_T, Y_T, and Z_T should be specified, but not both.
2. Either N2 and N3 or X_H, Y_H, and Z_H should be specified, but not both. If N2 and N3. Specifying N2 and N3 is equivalent of setting the head of the vector equal to the tail of the vector (X_T, Y_T, and Z_T) plus the vector from N2 to N3.

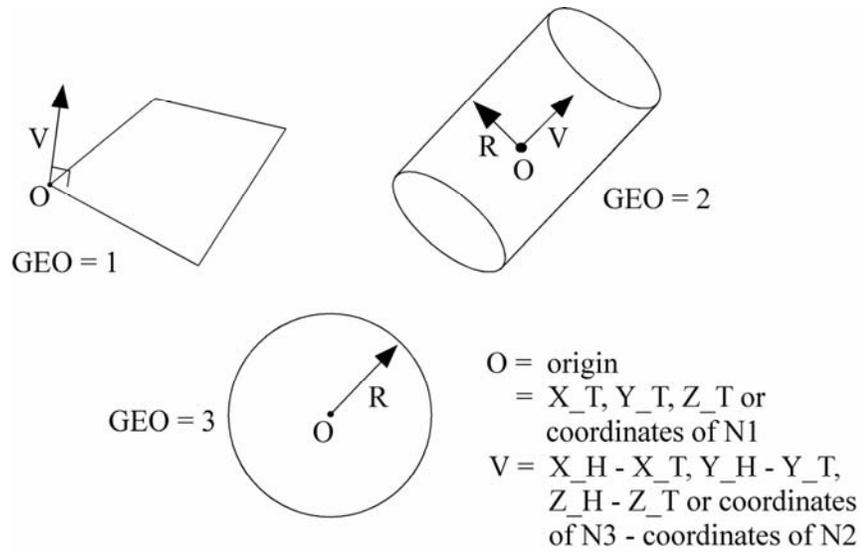


Figure 11.7.

***DEFINE_FRICTION**

Purpose: Define friction coefficients between parts for use in the contact options:

SINGLE_SURFACE,
AUTOMATIC_GENERAL,
AUTOMATIC_SINGLE_SURFACE,
AUTOMATIC_NODES_TO_SURFACE,
AUTOMATIC_SURFACE_TO_SURFACE,
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,
ERODING_SINGLE_SURFACE.

The input in this section continues until then next “*” card is encountered. Default friction values are used for any part ID pair that is not defined. Only one table can defined. The table is used if FS=-2.0 on the second card of the *CONTACT input definition. If FS=-2.0, this table will override the coefficients defined in *PART_CONTACT, which is activated by setting FS=-1.0.

Card 1 1 2 3 4 5 6 7 8

Variable	ID	FS_D	FD_D	DC_D	VC_D			
Type	I	F	F	F	F			
Default	0	0.0	0.0	0.0	0.0			

Cards 2, 3, ..., The next “*” card terminates the friction definition.

Card 2... 1 2 3 4 5 6 7 8

Variable	PID_I	PID_J	FS_IJ	FD_IJ	DC_IJ	VC_IJ		
Type	I	I	F	F	F	F		
Default			0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Identification number. Only one table is allowed.
FS_D	Default value of the static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$. Default values are used when part pair are undefined.
FD_D	Default value of the dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$. Default values are used when part pair are undefined.
DC_D	Default value of the exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$. Default values are used when part pair are undefined.
VC_D	Default value of the coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{lim} = VC \cdot A_{cont}$. A_{cont} being the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_o is the yield stress of the contacted material. Default values are used when part pair are undefined.
PID_I	Part ID I.
PID_J	Part ID J.
FS_IJ	Static coefficient of friction between parts I and J.
FD_IJ	Dynamic coefficient of friction between parts I and J.
DC_IJ	Exponential decay coefficient between parts I and J.
VC_IJ	Viscous friction between parts I and J.

***DEFINE_HEX_SPOTWELD_ASSEMBLY_{OPTION}**

Available options include the number of solid hexahedron elements that are used in the spot weld patch:

<BLANK>

N

Purpose: Define a list of hexahedral solid elements clusters that make up a single spot weld for computing the force and moment resultants that are written into the SWFORC output file. A maximum of a 16 element cluster may be used to define a single spot weld. See Fig. 11.8. This table is generated automatically when beam elements are converted to solid elements. See the input parameter, RPBHX, which is described in the control section: *CONTROL_SPOTWELD_BEAM.

Card 1 1 2 3 4 5 6 7 8

Variable	ID_SW								
Type	I								
Default	0								

Card 2

Variable	EID1	EID2	EID3	EID4	EID5	EID6	EID7	EID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

*DEFINE

*DEFINE_HEX_SPOTWELD_ASSEMBLY

Define the following card if and only if N>8

Optional	1	2	3	4	5	6	7	8
Variable	EID9	EID10	EID11	EID12	EID13	EID14	EID15	EID16
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID_SW	Spot weld ID. A unique ID number must be used.
EID n	Element ID n for up to 16 solid hexahedron elements.

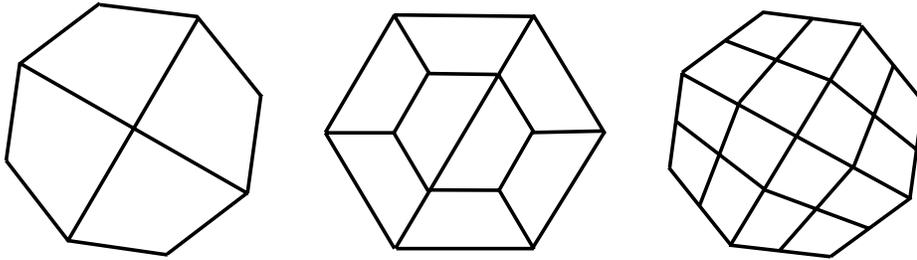


Figure 11.8. Sample four, eight, and sixteen element spot weld clusters comprised of solid hexahedron elements.

***DEFINE_SD_ORIENTATION**

Purpose: Define orientation vectors for discrete springs and dampers. These orientation vectors are optional for this element class. Four alternative options are possible. With the first two options, IOP= 0 or 1, the vector is defined by coordinates and is fixed permanently in space. The third and fourth option orients the vector based on the motion of two nodes, so that the direction can change as the line defined by the nodes rotates.

Card	1	2	3	4	5	6	7	8
Variable	VID	IOP	XT	YT	ZT	NID1	NID2	
Type	I	I	F	F	F	I	I	
Default	0	0	0.0	0.0	0.0	0	0	
Remarks	none	1	IOP=0,1	IOP=0,1	IOP=0,1	IOP=2,3	IOP=2,3	

VARIABLE**DESCRIPTION**

VID	Orientation vector ID. A unique ID number must be used.
IOP	Option: EQ.0: deflections/rotations are measured and forces/moments applied along the following orientation vector. EQ.1: deflections/rotations are measured and forces/moments applied along the axis between the two spring/damper nodes projected onto the plane normal to the following orientation vector. EQ.2: deflections/rotations are measured and forces/moments applied along a vector defined by the following two nodes. EQ.3: deflections/rotations are measured and forces/moments applied along the axis between the two spring/damper nodes projected onto the plane normal to the a vector defined by the following two nodes.
XT	x-value of orientation vector. Define if IOP=0,1.
YT	y-value of orientation vector. Define if IOP=0,1.
ZT	z-value of orientation vector. Define if IOP=0,1.
NID1	Node 1 ID. Define if IOP=2,3.
NID2	Node 2 ID. Define if IOP=2, 3.

Remarks:

1. The orientation vectors defined by options 0 and 1 are fixed in space for the duration of the simulation. Options 2 and 3 allow the orientation vector to change with the motion of the nodes. Generally, the nodes should be members of rigid bodies, but this is not mandatory. When using nodes of deformable parts to define the orientation vector, care must be taken to ensure that these nodes will not move past each other. If this happens, the direction of the orientation vector will immediately change with the result that initiate severe instabilities can develop.

***DEFINE_SET_ADAPTIVE**

Purpose: To control the adaptive refinement level by element or part set.

Card 1 1 2 3 4 5 6 7 8

Variable	SETID	STYPE	ADPLVL	ADPSIZE				
Type	I	I	I	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
SETID	Element set ID or part set ID
STYPE	Set type for SETID: 1-element set 2-part set
ADPLVL	Adaptive refinement level for all elements in SETID set.
ADSIZE	Minimum element size to be adapted based on element edge length for all elements in SETID set.

Remarks:

1. This option is for 3D-shell h-adaptivity only at the present time.
2. The order of defining refinement level for any elements is *CONTROL_ADAPTIVITY and *DEFINE_BOX_ADAPTIVE.
3. If there are multiple definitions of refinement level or element size for any elements, the latter one will be used.

*DEFINE

*DEFINE_SPOTWELD_FAILURE_RESULTANTS

*DEFINE_SPOTWELD_FAILURE_RESULTANTS

Purpose: Define failure criteria between part pairs for predicting spot weld failure. This table is implemented for *solid* element spot welds, which are used with the tied, constraint based, contact option: *CONTACT_TIED_SURFACE_TO_SURFACE. *Note that other tied contact types cannot be used.* The input in this section continues until then next “*” card is encountered. Default values are used for any part ID pair that is not defined. Only one table can defined. See *MAT_SPOTWELD where this option is used whenever *OPT=7*.

Card 1 1 2 3 4 5 6 7 8

Variable	ID	DSN	DSS	DLCIDSN	DLCIDSS			
Type	I	F	F	I	I			
Default	0	0.0	0.0	0	0			

Cards 2, 3, ..., The next “*” card terminates the table definition.

Card 2... 1 2 3 4 5 6 7 8

Variable	PID_I	PID_J	SNIJ	SSIJ	LCIDSNIJ	LCIDSSIJ		
Type	I	I	F	F	I	I		
Default	none	none	0.0	0.0	0	0		

VARIABLE

DESCRIPTION

ID	Identification number. Only one table is allowed.
DSN	Default value of the normal static stress at failure.
DSS	Default value of the transverse static stress at failure.
DLCIDSN	Load curve ID defining a scale factor for the normal stress as a function of strain rate. This factor multiplies DSN to obtain the failure value at a given strain rate.

VARIABLE	DESCRIPTION
DLCIDSS	Load curve ID defining a scale factor for static shear stress as a function of strain rate. This factor multiplies DSN to obtain the failure value at a given strain rate.
PID_I	Part ID I.
PID_J	Part ID J.
SNIJ	The maximum axial stress at failure between parts I and J. The axial stress is computed from the solid element stress resultants, which are based on the nodal point forces of the solid element.
DSSIJ	The maximum shear stress at failure between parts I and J. The shear stress is computed from the solid element stress resultants, which are based on the nodal point forces of the solid element.
LCIDSNIJ	Load curve ID defining a scale factor for the normal stress as a function of strain rate. This factor multiplies SNIJ to obtain the failure value at a given strain rate.
LCIDSSIJ	Load curve ID defining a scale factor for static shear stress as a function of strain rate. This factor multiplies SSIJ to obtain the failure value at a given strain rate.

Remarks:

The stress based failure model, which was developed by *Toyota Motor Corporation*, is a function of the peak axial and transverse shear stresses. The entire weld fails if the stresses are outside of the failure surface defined by:

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{\tau^F}\right)^2 - 1 = 0$$

where σ_{rr}^F and τ^F are specified in the above table by part ID pairs. LS-DYNA automatically identifies the part ID of the attached shell element for each node of the spot weld solid and checks for failure. If failure is detected the solid element is deleted from the calculation.

If the effects of strain rate are considered, then the failure criteria becomes:

$$\left(\frac{\sigma_{rr}}{f_{dsn}(\dot{\epsilon}^p)\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{f_{dss}(\dot{\epsilon}^p)\tau^F}\right)^2 - 1 = 0$$

*DEFINE

*DEFINE_SPOTWELD RUPTURE_PARAMETER

*DEFINE_SPOTWELD RUPTURE_PARAMETER

Purpose: Define a parameter by part ID for shell elements attached to spot weld *beam* elements using the constrained contact option: *CONTACT_SPOTWELD. *This table will not work with other contact types.* Only one table is permitted in the problem definition. Data, which is defined in this table, is used by the stress based spot weld failure model developed by *Toyota Motor Corporation*. See *MAT_SPOTWELD where this option is activated by setting the parameter *OPT* to a value of 9. This spot weld failure model is a development of *Toyota Motor Corporation*.

Card 1 1 2 3 4 5 6 7 8

Variable	PID							
Type	I							
Default								

Card 2

Variable	C11	C12	C13	N11	N12	N13		SIG_PF
Type	F	F	F	F	F	F		F
Default								

Card 3

Variable	C21	C22	C23	N2				SIG_NF
Type	F	F	F	F				
Default								

Card 4 1 2 3 4 5 6 7 8

Variable	LCDPA	LCDPM	LCDPS	LCDNA	LCDNM	LCDNS		NSMT
Type	F	F	F	F	F	F		F
Default	0	0	0	0	0	0		0

VARIABLE**DESCRIPTION**

PID	Part ID for the attached shell.
C11-N2	Parameters for model, see Remarks below.
SIG_PF	Nugget pull-out stress, σ_p .
SIG_NF	Nugget fracture stress, σ_f .
LCDPA	Curve ID defining dynamic scale factor of spot weld axial load rate for nugget pull-out mode.
LCDPM	Curve ID defining dynamic scale factor of spot weld moment load rate for nugget pull-out mode.
LCDPS	Curve ID defining dynamic scale factor of spot weld shear load rate for nugget pull-out mode.
LCDNA	Curve ID defining dynamic scale factor of spot weld axial load rate for nugget fracture mode.
LCDNM	Curve ID defining dynamic scale factor of spot weld moment load rate for nugget fracture mode.
LCDNS	Curve ID defining dynamic scale factor of spot weld shear load rate for nugget fracture mode.
NSMT	The number of time steps used for averaging the resultant rates for the dynamic scale factors.

*DEFINE

*DEFINE_SPOTWELD RUPTURE_PARAMETER

Remarks:

This failure model incorporates two failure functions, one for nugget pull-out and the other for nugget fracture. The nugget pull-out failure function is

$$F_p = \frac{C11 \cdot A / D^{N11} + C12 \cdot M / D^{N12} + C13 \cdot S / D^{N13}}{\sigma_p \left[1 + \left(\frac{\dot{\epsilon}^p}{C} \right)^{1/p} \right]}$$

where A , M , and S are the axial force, moment, and shear resultants respectively, D is the spot weld diameter, and the Cowper-Symonds coefficients are from the attached shell material model. If the Cowper-Symonds coefficients aren't specified, the term within the square brackets, [], is 1.0. The fracture failure function is

$$F_n = \frac{\sqrt{(C21 \cdot A + C22 \cdot M)^2 + 3(C23 \cdot S)^2}}{D^{N2} \cdot \sigma_F \left[1 + \left(\frac{\dot{\epsilon}^p}{C} \right)^{1/p} \right]}$$

When the load curves for the rate effects are specified, the failure criteria are

$$F_p = \frac{C11 \cdot f_{dpa}(\dot{A}) \cdot A / D^{N11} + C12 \cdot f_{dpm}(\dot{M}) \cdot M / D^{N12} + C13 \cdot f_{dps}(\dot{S}) \cdot S / D^{N13}}{\sigma_p}$$

$$F_n = \frac{\sqrt{(C21 \cdot f_{dna}(\dot{A}) \cdot A + C22 \cdot f_{dnm}(\dot{M}) \cdot M)^2 + 3(C23 \cdot f_{dns}(\dot{S}) \cdot S)^2}}{D^{N2} \cdot \sigma_F}$$

where f is the appropriate load curve scale factor. The scale factor for each term is set to 1.0 for when no load curve is specified. No extrapolation is performed if the rates fall outside of the range specified in the load curve to avoid negative scale factors. A negative load curve ID designates that the curve abscissa is the \log_{10} of the resultant rate. This option is recommended when the curve data covers several orders of magnitude in the resultant rate. Note that the load curve dynamic scaling replaces the Cowper-Symonds model for rate effects.

Failure occurs when either of the failure functions is greater than 1.0.

***DEFINE_SPOTWELD RUPTURE STRESS**

Purpose: Define a static stress rupture table by part ID for shell elements connected to spot weld beam elements using the constrained contact option: *CONTACT_SPOTWELD. This table will not work with other contact types. Only one table is permitted in the problem definition. Data, which is defined in this table, is used by the stress based spot weld failure model developed by Toyota Motor Corporation. See *MAT_SPOTWELD where this option is activated by setting the parameter OPT to a value of 6.

Define rupture stresses part by part. The next “*” card terminates this input.

Card 1 2 3 4 5 6 7 8

Variable	PID	SRSIG	SIGTAU	ALPHA				
Type	I	F	F	F				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID for the attached shell.
SRSIG	Axial (normal) rupture stress, σ_{rr}^F .
SRTAU	Transverse (shear) rupture stress, τ^F .
ALPHA	Scaling factor for the axial stress as defined by Toyota. The default value is 1.0.

Remarks:

The stress based failure model, which was developed by Toyota Motor Corporation, is a function of the peak axial and transverse shear stresses. The entire weld fails if the stresses are outside of the failure surface defined by:

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{\tau^F}\right)^2 - 1 = 0$$

where σ_{rr}^F and τ^F are specified in the above table by part ID. LS-DYNA automatically identifies the part ID of the attached shell element for each node of the spot weld beam and independently checks each end for failure. If failure is detected in the end attached to the shell with the greatest plastic strain, the beam element is deleted from the calculation.

If the effects of strain rate are considered, then the failure criteria becomes:

DEFINE**DEFINE_SPOTWELD RUPTURE_STRESS**

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F(\dot{\epsilon}^p)} \right)^2 + \left(\frac{\tau}{\tau^F(\dot{\epsilon}^p)} \right)^2 - 1 = 0$$

where $\sigma_{rr}^F(\dot{\epsilon}^p)$ and $\tau^F(\dot{\epsilon}^p)$ are found by using the Cowper and Symonds model which scales the static failure stresses:

$$\sigma_{rr}^F(\dot{\epsilon}^p) = \sigma_{rr}^F \cdot \left[1 + \left(\frac{\dot{\epsilon}^p}{C} \right)^{1/p} \right]$$

$$\tau^F(\dot{\epsilon}^p) = \tau^F \cdot \left[1 + \left(\frac{\dot{\epsilon}^p}{C} \right)^{1/p} \right]$$

where $\dot{\epsilon}^p$ is the average plastic strain rate which is integrated over the domain of the attached shell element, and the constants p and C are uniquely defined at each end of the beam element by the constitutive data of the attached shell. The constitutive model is described in the material section under keyword: *MAT_PIECEWISE_LINEAR_PLASTICITY.

The peak stresses are calculated from the resultants using simple beam theory.

$$\sigma_{rr} = \frac{N_{rr}}{A} + \frac{\sqrt{M_{rs}^2 + M_{rt}^2}}{\alpha Z} \quad \tau = \frac{M_{rr}}{2Z} + \frac{\sqrt{N_{rs}^2 + N_{rt}^2}}{A}$$

where the area and section modulus are given by:

$$A = \pi \frac{d^2}{4}$$

$$Z = \pi \frac{d^3}{32}$$

and d is the diameter of the spot weld beam.

***DEFINE_STAGED_CONSTRUCTION_PART**

Purpose: Staged construction. This keyword offers a simple way to define parts that are removed (e.g., during excavation), added (e.g., new construction) and used temporarily (e.g., props) during the analysis. Available for solid, shell, and beam element parts.

Note: This keyword card will be available starting in release 3 of version 971.

Card 1 2 3 4 5 6 7 8

Variable	PID	STGA	STGR					
Type	I	I	I					
Default	none	See Remarks	See Remarks					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID
STGA	Construction stage at which part is added
STGR	Construction stage at which part is removed

Remarks:

Used with *DEFINE_CONSTRUCTION_STAGES (defines the meaning of stages STGA and STGR) and *CONTROL_STAGED_CONSTRUCTION. If STGA=0, the part is present at the start of the analysis. If STGR=0, the part is still present at the end of the analysis. Examples:

1. Soil that is excavated would have STGA=0 but STGR>0
2. New construction would have STGA>0 and STGR=0
3. Temporary works would have STGA>0, STGR>STGA.

This is a convenience feature that reduces the amount of input data needed for many typical construction models. Internally, LS-DYNA checks for *LOAD_REMOVE_PART, *LOAD_GRAVITY_PART and *LOAD_STIFFEN_PART referencing the same PID. Generally, these will not be present and LS-DYNA creates the data using STGA and STGR, and default gravity and pre-construction stiffness factor from *CONTROL_STAGED_CONSTRUCTION. If existing cards are found, STGA and STGR are inserted into the existing data. During the analysis, any load curves entered on those existing cards will override STGA and STGR.

*DEFINE

*DEFINE_TABLE

*DEFINE_TABLE

Purpose: Define a table. This input section is somewhat unique in that another keyword, ***DEFINE_CURVE**, is used as part of the input in this section. A table consists of a ***DEFINE_TABLE** card followed by n lines of input. Each of the n additional lines define a numerical value in ascending order corresponding to a ***DEFINE_CURVE** input which follows the ***DEFINE_TABLE** keyword and the related input. For example, to define strain rate dependency where it is desired to provide a stress versus strain curve for each strain rate, n strain rates would be defined following the ***DEFINE_TABLE** keyword. The curves then follow which make up the table. Each curve may have unique spacing and an arbitrary number of points in their definition. (Load curve ID's defined for the table may be referenced elsewhere in the input.) *However, the curves must not cross except at the origin and the curves must share the same origin and end point.* This rather awkward input is done for efficiency reasons related to the desire to avoid indirect addressing in the inner loops used in the constitutive model stress evaluation.

Card 1 1 2 3 4 5 6 7 8

Variable	TBID							
Type	I							
Default	none							

Card 2, 3, 4, etc. Put one point per card (E20.0). Input is terminated when a “*DEFINE_CURVE” card is found.

Card 2... 1 2 3 4 5 6 7 8

Variable	VALUE			
Type	F			
Default	0.0			

Insert one ***DEFINE_CURVE** input section here for each point defined above.

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
VALUE	Load curve will be defined corresponding to this value, e.g., this value could be a strain rate, see purpose above.

Remarks:

1. If for example, 10 stress-strain curves for 10 different strain rates are given, 10 cards with the ascending values of strain rate then follow the first card. Afterwards, 10 corresponding *DEFINE_CURVE specifications have to follow.

***DEFINE**

***DEFINE_TRANSFORMATION**

***DEFINE_TRANSFORMATION**

Purpose: Define a transformation for the INCLUDE_TRANSFORM keyword option. The *DEFINE_TRANSFORMATION command must be defined before the *INCLUDE_TRANSFORM command can be used.

Cards 1, 2, 3, 4, ... (The next “*” card terminates the input.) This set is a combination of a series of options listed in the table defined below.

Card 1 1 2 3 4 5 6 7 8

Variable	TRANID								
Type	I								
Default	none								

Card 2

Variable	OPTION	A1	A2	A3	A4	A5	A6	A7
Type	A	F	F	F	F	F	F	F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TRANID	Transform ID.
OPTION	For the available options see the table below.
A1-A7	Specified entity. Each card must have an option specified. See Table 11.1 below for the available options.

Table 11.1

FORMAT (A10,7F10.0)		
OPTION	ENTITIES + ATTRIBUTES	FUNCTION
SCALE	a1, a2, a3	Scale the global x, y, and z coordinates of a point by a1, a2, and a3, respectively. If zero, a default of unity is set.
ROTATE	a1, a2, a3, a4, a5, a6, a7	Rotate through an angle, a7, about a line with direction cosines a1, a2, and a3 passing through the point with coordinates a4, a5, and a6. If a4 through a7 are zero, then a1 and a2 are the ID's of two POINTs and a3 defines the rotation angle. The axis of rotation is defined by a vector going from point with ID a1 to point with ID a2.
TRANSL	a1, a2, a3	Translate the x, y, and z coordinates of a point by a1, a2, and a3, respectively.
POINT	a1,a2,a3,a4	Define a point with ID, a1, with the initial coordinates a2, a3, and a4.

The ordering of the SCALE, ROTATE, and TRANSL commands is important. It is generally recommend to first scale, then rotate, and finally translate the model.

The POINT option in ROTATE provides a means of defining rotations about axes defined by the previous transformations. The coordinates of the two POINTs are transformed by all the transformations up to the transformation where they are referenced. The POINTs must be defined before they are referenced, and their identification numbers are local to each *DEFINE_TRANSFORMATION. The coordinates of a POINT are transformed using all the transformations before it is referenced, not just the transformations between its definition and its reference. To put it another way, while the ordering of the transformations is important, the ordering between the POINTs and the transformations is not important.

In the following example, the *DEFINE_TRANSFORMATION command is used 3 times to input the same dummy model and position it as follows:

1. Transformation id 1000 imports the dummy model (dummy.k) and rotates it 45 degrees about z-axis at the point (0.0,0.0,0.0). Transformation id 1001 performs the same transformation using the POINT option.
2. Transformation id 2000 imports the same dummy model (dummy.k) and translates 1000 units in the x direction.
3. Transformation id 3000 imports the same dummy model (dummy.k) and translates 2000 units in the x direction. For each *DEFINE_TRANSFORMATION, the commands TRANSL, SCALE, and ROTATE are available. The transformations are applied in the

*DEFINE

*DEFINE_TRANSFORMATION

order in which they are defined in the file, e.g., transformation id 1000 in this example would translate, scale and then rotate the model. *INCLUDE_TRANSFORM uses a transformation id defined by a *DEFINE_TRANSFORMATION command to import a model and perform the associated transformations. It also allows the user upon importing the model to apply offsets to the various entity ids and perform unit conversion of the imported model.

```
*KEYWORD
*DEFINE_TRANSFORMATION
  1000
$ option &      dx&      dy&      dz&
TRANSL          0000.0    0.0      0.0
$ option &      dx&      dy&      dz&
SCALE           1.00     1.0      1.0
$ option &      dx&      dy&      dz&      px&      py&      pz&
angle&
ROTATE          0.00     0.0      1.0      0.00     0.00     0.0
45.00
*DEFINE_TRANSFORMATION
  1001
POINT           1        0.0      0.0      0.0
POINT           2        0.0      0.0      1.0
ROTATE          1        2        45.0
*DEFINE_TRANSFORMATION
  2000
$ option &      dx&      dy&      dz&
TRANSL          1000.0    0.0      0.0
*DEFINE_TRANSFORMATION
$ traniid &
  3000
$ option &      dx&      dy&      dz&
TRANSL          2000.0    0.0      0.0
*INCLUDE_TRANSFORM
dummy.k
$ idnoff &    ideoff&    idpoff& idmoff &    idsoff &    iddofff&    iddofff &
  0          0          0          0          0          0          0
$ idrofff&    ilctmf&
  0          0
$ fctmas&    fcttim&    fctlen&    fcttem &    incout&
  1.0000    1.0000    1.00     1.0      1
$ traniid &
  1000
*INCLUDE_TRANSFORM
dummy.k
$ idnoff &    ideoff&    idpoff& idmoff &    idsoff &    iddofff&    iddofff &
  1000000    1000000    1000000    1000000    1000000    1000000    1000000
$ idrofff&    ilctmf&
  1000000    1000000
$ fctmas&    fcttim&    fctlen&    fcttem &    incout&
  1.0000    1.0000    1.00     1.0      1
$ traniid &
  2000
*INCLUDE_TRANSFORM
dummy.k
$ idnoff &    ideoff&    idpoff& idmoff &    idsoff &    iddofff&    iddofff &
  2000000    2000000    2000000    2000000    2000000    2000000    2000000
$ idrofff&    ilctmf&
  2000000    2000000
$ fctmas&    fcttim&    fctlen&    fcttem &    incout&
  1.0000    1.0000    1.00     1.0      1
$ traniid &
  3000
*END
```

***DEFINE_VECTOR**

Purpose: Define a vector by defining the coordinates of two points.

Card	1	2	3	4	5	6	7	8
Variable	VID	XT	YT	ZT	XH	YH	ZH	CID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0
Remarks								

VARIABLE**DESCRIPTION**

VID	Vector ID
XT	X-coordinate of tail of vector
YT	Y-coordinate of tail of vector
ZT	Z-coordinate of tail of vector
XH	X-coordinate of head of vector
YH	Y-coordinate of head of vector
ZH	Z-coordinate of head of vector
CID	Coordinate system ID to define vector in local coordinate system. All coordinates, XT, YT, ZT, XH, YH, and ZH are in respect to CID. EQ.0: global (default).

Remarks:

1. The coordinates should differ by a certain margin to avoid numerical inaccuracies.

EXAMPLES

The following examples demonstrate the input for these options:

- *DEFINE_BOX
- *DEFINE_COORDINATE_NODES,
- *DEFINE_COORDINATE_SYSTEM,
- *DEFINE_COORDINATE_VECTOR
- *DEFINE_CURVE
- *DEFINE_SD_ORIENTATION
- *DEFINE_VECTOR commands.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *DEFINE_BOX
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define box number eight which encloses a volume defined by two corner
$ points: (-20.0, -39.0, 0.0) and (20.0, 39.0, 51.0). As an example, this
$ box can be used as an input for the *INITIAL_VELOCITY keyword in which
$ all nodes within this box are given a specific initial velocity.
$
*DEFINE_BOX
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   boxid      xmm      xmx      ymn      ymx      zmn      zmx
$           8      -20.0      20.0      -39.0      39.0      0.0      51.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *DEFINE_COORDINATE_NODES
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define local coordinate system number 5 using three nodes: 10, 11 and 20.
$ Nodes 10 and 11 define the local x-direction. Nodes 10 and 20 define
$ the local x-y plane.
$
$ For example, this coordinate system (or any coordinate system defined using
$ a *DEFINE_COORDINATE_option keyword) can be used to define the local
$ coordinate system of a joint, which is required in order to define joint
$ stiffness using the *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED keyword.
$
*DEFINE_COORDINATE_NODES
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   cid      n1      n2      n3
$           5      10      11      20
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *DEFINE_COORDINATE_SYSTEM
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define local coordinate system number 3 using three points. The origin of
$ local coordinate system is at (35.0, 0.0, 0.0). The x-direction is defined
$ from the local origin to (35.0, 5.0, 0.0). The x-y plane is defined using
$ the vector from the local origin to (20.0, 0.0, 20.0) along with the local
$ x-direction definition.
$
*DEFINE_COORDINATE_SYSTEM
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   cid      Xo      Yo      Zo      Xl      Yl      Zl
$     3      35.0    0.0    0.0    35.0    5.0    0.0
$
$   Xp      Yp      Zp
$   20.0    0.0    20.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ *DEFINE_COORDINATE_VECTOR
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define local coordinate system number 4 using two vectors.
$ Vector 1 is defined from (0.0, 0.0, 0.0) to (1.0, 1.0, 0.0)
$ Vector 2 is defined from (0.0, 0.0, 0.0) to (1.0, 1.0, 1.0)
$ See the corresponding keyword command for a description.
$
*DEFINE_COORDINATE_VECTOR
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   cid      Xx      Yx      Zx      Xv      Yv      Zv
$     4       1.0     1.0     0.0     1.0     1.0     1.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$  *DEFINE_CURVE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define curve number 517. This particular curve is used to define the
$ force-deflection properties of a spring defined by a *MAT_SPRING_INELASTIC
$ keyword. The abscissa value is offset 25.0 as a means of modeling a gap
$ at the front of the spring. This type of spring would be a compression
$ only spring.
$
*DEFINE_CURVE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      lcid      sidr      scla      sclo      offa      offo
$          517
$
$      abscissa      ordinate
$          0.0          0.0
$          80.0         58.0
$          95.0         35.0
$         150.0         44.5
$         350.0         45.5
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$  *DEFINE_SD_ORIENTATION
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ A discrete spring is defined with two nodes in 3-D space. However, it is
$ desired to have the force of that spring to act only in the z-direction.
$ The following definition makes this happen. Additionally, vid = 7
$ must be specified in the *ELEMENT_DISCRETE keyword for this spring.
$
*DEFINE_SD_ORIENTATION
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      vid      iop      xt      yt      zt      nid1      nid2
$          7          0          0.0      0.0      1.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *DEFINE_VECTOR
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define vector number 5 from (0,0,0) to (0,1,1). As an example, this vector
$ can be used to define the direction of the prescribed velocity of a node
$ using the *BOUNDARY_PRESCRIBED_MOTION_NODE keyword.
$
*DEFINE_VECTOR
$
$.>...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$ vid xt yt zt xh yh zh
$ 3 0.0 0.0 0.0 0.0 1.0 1.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
    
```


***DEFORMABLE_TO_RIGID**

The cards in this section are defined in alphabetical order and are as follows:

***DEFORMABLE_TO_RIGID**

***DEFORMABLE_TO_RIGID_AUTOMATIC**

***DEFORMABLE_TO_RIGID_INERTIA**

If one of these cards is defined, then any deformable part defined in the model may be switched to rigid during the calculation. Parts that are defined as rigid (*MAT_RIGID) in the input are permanently rigid and cannot be changed to deformable.

Deformable parts may be switched to rigid at the start of the calculation by specifying them on the *DEFORMABLE_TO_RIGID card.

Part switching may be specified on a restart (see RESTART section of this manual) or it may be performed automatically by use of the *DEFORMABLE_TO_RIGID_AUTOMATIC cards.

The *DEFORMABLE_TO_RIGID_INERTIA cards allow inertial properties to be defined for deformable parts that are to be swapped to rigid at a later stage.

It is not possible to perform part material switching on a restart if it was not flagged in the initial analysis. The reason for this is that extra memory needs to be set up internally to allow the switching to take place. If part switching is to take place on a restart, but no parts are to be switched at the start of the calculation, no inertia properties for switching and no automatic switching sets are to be defined, then just define one *DEFORMABLE_TO_RIGID card without further input.

***DEFORMABLE_TO_RIGID**

***DEFORMABLE_TO_RIGID**

***DEFORMABLE_TO_RIGID**

Purpose: Define materials to be switched to rigid at the start of the calculation.

Card 1 2 3 4 5 6 7 8

Variable	PID	MRB						
Type	I	I						
Default	none	0						

VARIABLE

DESCRIPTION

- PID Part ID of the part which is switched to a rigid material, also see *PART.
- MRB Part ID of the master rigid body to which the part is merged. If zero, the part becomes either an independent or master rigid body.

***DEFORMABLE_TO_RIGID_AUTOMATIC *DEFORMABLE_TO_RIGID**

***DEFORMABLE_TO_RIGID_AUTOMATIC**

Purpose: Define a set of parts to be switched to rigid or to deformable at some stage in the calculation.

Card 1 1 2 3 4 5 6 7 8

Variable	SWSET	CODE	TIME 1	TIME 2	TIME 3	ENTNO	RELSW	PAIRED
Type	I	I	F	F	F	I	I	I
Default	none	0	0.	1.0E20	0.	0.	0	0
Remark		1				1,2		3

Card 2

Variable	NRBF	NCSF	RWF	DTMAX	D2R	R2D		
Type	I	I	I	F	I	I		
Default	0	0	0	0.	0	0		
Remark	4	4	4					

VARIABLE

DESCRIPTION

SWSET	Set number for this automatic switch set. Must be unique.
CODE	Activation switch code. Defines the test to activate the automatic material switch of the part: EQ.0: switch takes place at time 1, EQ.1: switch takes place between time 1 and time 2 if rigid wall force (specified below) is zero, EQ.2: switch takes place between time 1 and time 2 if contact surface force (specified below) is zero, EQ.3: switch takes place between time 1 and time 2 if rigid wall force (specified below) is non-zero, EQ.4: switch takes place between time 1 and time 2 if contact surface force (specified below) is non-zero.

***DEFORMABLE_TO_RIGID *DEFORMABLE_TO_RIGID_AUTOMATIC**

VARIABLE	DESCRIPTION
TIME 1	Switch will not take place before this time.
TIME 2	Switch will not take place after this time: EQ.0 Time 2 set to 1.0e20.
TIME 3	Delay period. After this part switch has taken place, another automatic switch will not take place for the duration of the delay period. If set to zero a part switch may take place immediately after this switch.
ENTNO	Rigid wall/contact surface number for switch codes 1, 2, 3, 4.
RELSW	Related switch set. The related switch set is another automatic switch set that must be activated before this part switch can take place: EQ.0: no related switch set.
PAIRED	Define a pair of related switches. EQ. 0: not paired EQ. 1: paired with switch set RELSW and is the Master switch. EQ.-1: paired with switch set RELSW and is the Slave switch.
NRBF	Flag to delete or activate nodal rigid bodies. If nodal rigid bodies or generalized, weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities: EQ.0: no change, EQ.1: delete, EQ.2: activate.
NCSF	Flag to delete or activate nodal constraint set. If nodal constraint/spot weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities: EQ.0: no change, EQ.1: delete, EQ.2: activate.
RWF	Flag to delete or activate rigid walls: EQ.0: no change, EQ.1: delete, EQ.2: activate.
DTMAX	Maximum permitted time step size after switch.
D2R	Number of deformable parts to be switched to rigid plus number of rigid parts for which new master/slave rigid body combinations will be defined: EQ.0: no parts defined.
R2D	Number of rigid parts to be switched to deformable: EQ.0: no parts defined.

*DEFORMABLE_TO_RIGID_AUTOMATIC *DEFORMABLE_TO_RIGID

Remarks:

1. Only surface to surface and node to surface contacts can be used to activate an automatic part switch.
2. Rigid wall numbers are the order in which they are defined in the deck. The first rigid wall and the first contact surface encountered in the input deck will have an entity number of 1. The contact surface id is that as defined on the *CONTACT_....._ID card.
3. Switch sets may be paired together to allow a pair of switches to be activated more than once. Each pair of switches should use consistent values for CODE, i.e. 1&3 or 2&4. Within each pair of switches the related switch, RELSW, should be set to the ID of the other switch in the pair. The Master switch (PAIRED = 1) will be activated before the Slave switch (PAIRED = -1). Pairing allows the multiple switches to take place as for example when contact is made and lost several times during an analysis.
4. If the delete switch is activated, ALL corresponding constraints are deactivated regardless of their relationship to a switched part. By default, constraints which are directly associated with a switched part are deactivated/activated as necessary.

```
$ Define a pair or related switches that will be activated by (no)force on
$ Contact 3. To start with switch set 20 will be activated (PAIRED=1) swapping
$ the PARTS to RIGID. When the contact force is none zero switch set 10 will be
$ activated swapping the PARTS to DEFORMABLE. If the contact force returns to
$ zero switch set 20 will be activated again making the PARTS RIGID.
$
*DEFORMABLE_TO_RIGID_AUTOMATIC
$.>...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$  swset   code   time 1   time 2   time 3   entno   relsw   paired
$    20      2      1      2      3      3      10      1
$  nrbf    ncsf    rwf    dtmax    D2R     R2D
$                                1

*DEFORMABLE_TO_RIGID_AUTOMATIC
$.>...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$  swset   code   time 1   time 2   time 3   entno   relsw   paired
$    10      4      1      2      3      3      20     -1
$  nrbf    ncsf    rwf    dtmax    D2R     R2D
$                                1
```

***DEFORMABLE_TO_RIGID *DEFORMABLE_TO_RIGID_AUTOMATIC**

Define D2R cards below:

Card 1 2 3 4 5 6 7 8

Variable	PID	MRB						
Type	I	I						
Default	none	0						

VARIABLE

DESCRIPTION

PID Part ID of the part which is switched to a rigid material.

MRB Part ID of the master rigid body to which the part is merged. If zero, the part becomes either an independent or master rigid body.

Define R2D cards below:

Card 1 2 3 4 5 6 7 8

Variable	PID							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

PID Part ID of the part which is switched to a deformable material.

*DEFORMABLE_TO_RIGID_INERTIA *DEFORMABLE_TO_RIGID

***DEFORMABLE_TO_RIGID_INERTIA**

Purpose: Inertial properties can be defined for the new rigid bodies that are created when the deformable parts are switched. These can only be defined in the initial input if they are needed in a later restart. Unless these properties are defined, LS-DYNA will recompute the new rigid body properties from the finite element mesh. The latter requires an accurate mesh description. **When rigid bodies are merged to a master rigid body, the inertial properties defined for the master rigid body apply to all members of the merged set.**

Card 1 1 2 3 4 5 6 7 8

Variable	PID							
Type	I							
Default	none							

Card 2

Variable	XC	YC	ZC	TM				
Type	F	F	F	F				

Card 3

Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		
Default	none	0.0	0.0	none	0.0	none		

***DEFORMABLE_TO_RIGID *DEFORMABLE_TO_RIGID_INERTIA**

VARIABLE	DESCRIPTION
PID	Part ID, see *PART.
XC	x-coordinate of center of mass
YC	y-coordinate of center of mass
ZC	z-coordinate of center of mass
TM	Translational mass
IXX	I_{xx} , xx component of inertia tensor
IXY	I_{xy}
IXZ	I_{xz}
IYY	I_{yy}
IYZ	I_{yz}
IZZ	I_{zz}

***EF**

Exchange factors characterize radiative heat transfer between collections of flat surfaces, the union of which is a closed surface (an enclosure). LS-DYNA can calculate exchange factors and then use them as boundary conditions for thermal runs. The (i, j) th element of an exchange factor matrix, E_{ij} , is the fraction of the Stefan-Boltzman surface energy radiated from surface i that is absorbed by surface j . LS-DYNA employs a Monte Carlo algorithm to calculate these exchange factors. For each surface, LS-DYNA simulates photon emission one photon at a time. For each photon, LS-DYNA generates a random initial position on the emitting surfaces as well as a random initial direction that points into the enclosure. LS-DYNA ray traces each photon until it is absorbed. The path of a simulated photon can be complex involving multiple diffuse and specular reflections as well as multiple diffuse and specular transmissions. The results of this Monte Carlo algorithm are used to assemble a matrix that is related to the exchange factor matrix, for which, the (i, j) th entry contains the number of photons emitted from surface i that are absorbed by surface j . From this matrix LS-DYNA then assembles the exchange factor matrix.

Limitations

The exchange factor algorithm is used to model heat transfer across an enclosure containing a non-participating media. The media within the enclosure is assumed to be transparent to the electromagnetic radiation. For modeling heat transfer across enclosures that are made entirely of diffusively reflecting grey-body surfaces, LS-DYNA features a simpler and faster-running algorithm than the Monte Carlo algorithm, that is, the view factor method (see *BOUNDARY_RADATION_VF, type=2). The exchange factor calculation cannot be used concurrently with view factors. A further limitation is that LS-DYNA is capable of including only one exchange factor enclosure per simulation.

Output

The file “exchfl” is a text file containing the exchange factors. This file is written when using the *BOUNDARY_RADIATION_SET_VF_CALCULATE keyword and read when using the *BOUNDARY_RADIATION_SET_VF_READ keyword.

EF Cards

LS-DYNA requires that the user supply all of the cards listed below unless noted as optional.

- ***EF_CONTROL**
- ***EF_GRID** (optional)
- ***EF_MATERIAL**
- ***EF_TOGGLES** (optional)

***EF_CONTROL**

Purpose: This card allows the user to set the parameters for the Monte Carlo algorithm.

This keyword should be used only once.

Card (1 of 2)

Card 1 1 2 3 4 5 6 7 8

Variable	NPHTON	NREFS	NWARNS	NLOST	NLOOPS	ERRDEF	INSEED	
Type	I	I	I	I	I	F	I	
Default	None	100	100	100	1	.01	0	

Card (2 of 2)

Card 2 1 2 3 4 5 6 7 8

Variable	DELT	SPLTOL	AREATOL	NINCR				
Type	F	F	F	I				
Default	0.01	0.0001	0.0001	0				

VARIABLE**DESCRIPTION**

NPHTON	The base number of photons emitted per band per surface per convergence loop. Note that NPHT from *BOUNDARY_RADIATION_SET_EF_CALCULATE also effects the number of photons emitted per surface per band per convergence loop.
NREFS	The maximum number of reflections allowed per photon before LS-DYNA issues a warning.
NWARNS	The maximum number of warnings allowed per surface before the run is aborted
NLOST	The maximum number of lost photons allowed per surface. Round off error often causes the loss of photons, so this number ought not to be set too small (usually the default is reasonable).

VARIABLE	DESCRIPTION
NLOOPS	This specifies the maximum number of convergence loops. If the relative error obtained upon the completion of a run is not within the specified tolerances, LS-DYNA will rerun the model combining the results of all previous runs together with the results of the present run to obtain a more accurate result. LS-DYNA will rerun the problem NLOOPS times to achieve error margins within the specified tolerances. If the desired level of convergence is not obtained within NLOOPS iterations LS-DYNA error terminates.
ERRDEF	Specifies that tolerance for convergence of the surface exchange fractions. This may be overridden on a surface by surface basis with the ERRMAX setting. (see *BOUNDARY_RADIATION_SET_EF_CALCULATE)
INSEED	Tells LS-DYNA how to obtain an initial seed for the Monte Carlo random number generator. <pre>if(INSEED.eq.0) then [use date and time] if(INSEED.gt.0) then [use INSEED as seed] if(INSEED.lt.0) then [use a default seed]</pre>
DELT	The cone angle interval used to numerically integrate material properties.
SPLTOL	To calculate exchange factors, LS-DYNA splits all of the enclosure's quadrilateral surfaces into two triangular surfaces. SPLTOL specifies the amount by which the dot product of the unit normal vectors of the two triangular surfaces can differ from unity.
AREATOL	LS-DYNA splits quadrilateral surfaces in the enclosures along the line connecting the first and third nodes. Quadrilaterals could, just as well, be split along the line connecting the second and fourth nodes. For numerical stability it is important the areas of the triangles created by either splitting be almost identical. AREATOL specifies the largest allowable difference in area.
NINCR	Controls restart-related behavior of LS-DYNA's exchange factor solver. <pre>if(NINCR.eq.0) then [Run normal, no restart files output] if(NINCR.gt.0) then [Write restart file after every NINC surfaces]</pre>

***EF_GRID**

Purpose: This card allows the user to specify grid parameters.

This keyword should appear only once.

Card (1 of 1)

Card 1 1 2 3 4 5 6 7 8

Variable	NGX	NGY	NGZ					
Type	I	I	I					
Default	None	None	None					

VARIABLE**DESCRIPTION**

NGX	The mathematical algorithm underlying the ray tracer, involves gridding the enclosure. NGX specify the number of grid divisions along the x axis. This parameter does not affect LS-DYNA's ability to obtain a solution, but it does effect the amount of CPU time consumed to process each photon. There is no fixed rule for picking NGX, NGY, and NGZ, however for large geometries involving 1,000 to 15,000 surfaces NGX = NGY = NGZ = 25 is often optimal. For smaller geometries smaller values are recommended.
NGY	Specifies the number of grid divisions along the y-axis.
NGZ	Specifies the number of grid divisions along the z-axis.

***EF_MATERIAL**

Purpose: This keyword defines exchange factor material IDs. To define multiple materials use this keyword more than once.

(Material Specifier)

Card 1 1 2 3 4 5 6 7 8

Variable	NMAT	NAME						
Type	F	A70						
Default	none	none						

These next two cards specify the properties of the material.

Card 2 1 2 3 4 5 6 7 8

Variable	MTYP	EXE	EYE	EZE				
Type	I	F	F	F				
Default	0	0	0	0				

Card 3

Variable	RHOS	RHOD	TAUS	TAUD	RDIFFR	RDIFFT		
Type	F	F	F	F	F	F		
Default	0	0	0	0	1	1		
Remark	1	1	1	1	2	2		

VARIABLE	DESCRIPTION												
NMAT	Specifies the material ID, of the exchange factor material.												
NAME	Specifies the material's name. This parameter is used only to make the output file easier to read.												
MTYP	Specifies if and how emission occurs: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">MTYP</th> <th style="text-align: left;">Material Emission</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">-2</td> <td>There is to be no emission and $F_{ij} = 1$ is written to the output file for this surface.</td> </tr> <tr> <td style="text-align: center;">-1</td> <td>There is to be no emission and $F_{ij} = 0$ is written to the output file for this surface.</td> </tr> <tr> <td style="text-align: center;">0</td> <td>Emission is to be distributed in θ according to: $\varepsilon(\theta) = \cos^r(\theta)$</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Beam emission is to occur in the direction $\{E_x, E_y, E_z\}$</td> </tr> <tr> <td style="text-align: center;">2</td> <td>This specifies that emission according to user specified function.</td> </tr> </tbody> </table>	MTYP	Material Emission	-2	There is to be no emission and $F_{ij} = 1$ is written to the output file for this surface.	-1	There is to be no emission and $F_{ij} = 0$ is written to the output file for this surface.	0	Emission is to be distributed in θ according to: $\varepsilon(\theta) = \cos^r(\theta)$	1	Beam emission is to occur in the direction $\{E_x, E_y, E_z\}$	2	This specifies that emission according to user specified function.
MTYP	Material Emission												
-2	There is to be no emission and $F_{ij} = 1$ is written to the output file for this surface.												
-1	There is to be no emission and $F_{ij} = 0$ is written to the output file for this surface.												
0	Emission is to be distributed in θ according to: $\varepsilon(\theta) = \cos^r(\theta)$												
1	Beam emission is to occur in the direction $\{E_x, E_y, E_z\}$												
2	This specifies that emission according to user specified function.												
EXE	Specifies the x component of emission for a type 1 material.												
EYE	Specifies the y component of emission for a type 1 material.												
EZE	Specifies the z component of emission for a type 1 material.												
RHOS	Specifies the specular reflectance.												
RHOD	Specifies the diffuse reflectance.												
TAUS	Specifies the specular transmittance.												
TAUD	Specifies the diffuse transmittance.												
RDIFFR	LS-DYNA simulates diffuse reflection according to the equation: $\varepsilon(\theta) = \cos^r(\theta)$. The user specifies the value for r with RDIFFR.												
RDIFFT	LS-DYNA simulates diffuse transmittance according to the equation: $\varepsilon(\theta) = \cos^r(\theta)$. The user specifies the value for r with RDIFFT.												

Remarks:

1. The standard cosine dependent probability function can be replaced with user-defined probability functions. Negative values of this parameter are taken to be material curve Ids that identify such user-defined probability functions. The range of the defined curve is 0 to 90 degrees.
2. Values different from 1 have been observed to result in errors in reciprocity, so the user is strongly encouraged to consider this when selecting values for RDIFFR and RDIFFT different from 1. Lambertian behavior is achieved by using a value of 1. Values greater than 1 result in biasing the distribution toward the normal, whereas values less than one result in biasing the distribution toward the grazing angle.

***EF_TOGGLES**

Purpose: This card allows the user to set output options.

This keyword should be used only once.

Card 1 1 2 3 4 5 6 7 8

Variable	IPRINT1	IPRINT2	IPRINT3	IPRINT4	IDATA	ITRACES	IRSTRT	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

VARIABLE

DESCRIPTION

IPRINT1	Controls output of exchange fractions to the d3hsp file. In almost all situations this should be set to 0 because the “exchange factors” are written to the file exchfl. EQ.0: do not write exchange fractions EQ.1: write exchange fraction.
IPRINT2	Controls output of a list of lost photons to the d3hsp file. This is useful for debugging. EQ.0: do not write lost photon list EQ.1: write lost photon list
IPRINT3	Controls output about the grid algorithm to the d3hsp file. EQ.0: do not write grid algorithm information EQ.1: write grid algorithm information
IPRINT4	Controls output about material information pertaining to exchange factors to the d3hsp file. EQ.0: do not write material information EQ.1: write material information
IDATA	Controls execution EQ.0: run proceeds EQ.1: terminate after input parameter check
ITRACES	ITRACES Controls output of photon trajectories. EQ.0: do not write trajectory information

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.1: write trajectory information. This file becomes large quickly and is only useful for debugging.
IRESTART	IRESTART should be set either to 1 or 0. If IRESTART is set to 1 then LS-DYNA restarts the exchange factor solver. If IRESTART is set to 1 and a .crh file exists, the Monte Carlo solver will pick up where it left off prior to a crash. If there is a .nij file but no .crh file, then LS-DYNA will recycle the results of the previous exchange factor running emitting more photons to increase accuracy.

***ELEMENT**

The element cards in this section are defined in alphabetical order:

***ELEMENT_BEAM_{OPTION}_{OPTION}**

***ELEMENT_DIRECT_MATRIX_INPUT**

***ELEMENT_DISCRETE**

***ELEMENT_INERTIA_{OPTION}**

***ELEMENT_MASS**

***ELEMENT_MASS_PART**

***ELEMENT_PLOTEL**

***ELEMENT_SEATBELT**

***ELEMENT_SEATBELT_ACCELEROMETER**

***ELEMENT_SEATBELT_PRETENSIONER**

***ELEMENT_SEATBELT_RETRACTOR**

***ELEMENT_SEATBELT_SENSOR**

***ELEMENT_SEATBELT_SLIPRING**

***ELEMENT_SHELL_{OPTION}**

***ELEMENT_SHELL_SOURCE_SINK**

***ELEMENT_SOLID_{OPTION}**

***ELEMENT_SPH**

***ELEMENT_TRIM**

***ELEMENT_TSHELL**

The ordering of the element cards in the input file is completely arbitrary. An arbitrary number of element blocks can be defined preceded by a keyword control card.

***ELEMENT_BEAM_{OPTION}_{OPTION}**

Available options include:

<BLANK>

THICKNESS, SCALAR, SCALR or SECTION

PID

OFFSET

ORIENTATION

WARPAGE

Purpose: Define two node elements including 3D beams, trusses, 2D axisymmetric shells, and 2D plane strain beam elements. The type of the element and its formulation is specified through the part ID (see *PART) and the section ID (see *SECTION_BEAM).

Two alternative methods are available for defining the cross sectional property data. The THICKNESS and SECTION options are provided for the user to override the *SECTION_BEAM data which is taken as the default if the THICKNESS or SECTION option is not used. End release conditions are imposed using constraint equations, and caution must be used with this option as discussed in remark 2 below. The SCALAR/SCALR options applies only to material model type 146, *MAT_1DOF_GENERALIZED_SPRING.

The PID option is used by the type 9 spot weld element only and is ignored for all other beam types. When the PID option is active an additional card is read that gives two part ID's that are tied by the spot weld element. If the PID option is inactive for the type 9 element the nodal points of the spot weld are located to the two nearest master segments. In either case, *CONTACT_SPOTWELD must be defined with the spot weld beam part as slave and the shell parts (including parts PID1 and PID2) as master. The surface of each segment should project to the other and in the most typical case the node defining the weld, assuming only one node is used, should lie in the middle; however, this is not a requirement. Note that with the spot weld elements only one node is needed to define the weld, and two nodes are optional.

Card Format (10I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N ₁	N ₂	N ₃	RT1	RR1	RT2	RR2	LOCAL
Type	I	I	I	I	I	I	I	I	I	I
Default	none	None	none	none	none	0	0	0	0	2
Remarks					1	2,3	2,3	2,3	2,3	2,3

Optional Card (Required if THICKNESS is specified after the keyword)

Card 1 2 3 4 5 6 7 8 9 10

Variable	PARM1	PARM2	PARM3	PARM4	PARM5
Type	F	F	F	F	F
Remarks	4	5	5	5	6

Optional Card (Required if SECTION is specified after the keyword)

Card 1 2 3 4 5 6 7 8

Variable	STYPE	D1	D2	D3	D4	D5	D6	
Type	A	F	F	F	F	F	F	
Remarks								

ELEMENT**ELEMENT_BEAM****Optional Card (Required if SCALAR is specified after the keyword)**

Card 1 2 3 4 5 6 7 8 9 10

Variable	VOL	INER	CID	DOFN1	DOFN2
Type	F	F	F	F	F

Optional Card (Required if SCALR is specified after the keyword)

Card 1 2 3 4 5 6 7 8 9 10

Variable	VOL	INER	CID1	CID2	DOFNS
Type	F	F	F	F	F

Optional Card (Required if PID is specified after the keyword)

Card 1 2 3 4 5 6 7 8 9 10

Variable	PID1	PID2								
Type	I	I								
Default	none	none								
Remarks										

Optional Card (Required if OFFSET is specified after the keyword)

Card 1 2 3 4 5 6 7 8

Variable	WX1	WY1	WZ1	WX2	WY2	WZ2		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		
Remarks	8	8	8	8	8	8		

Optional Card (Required if ORIENTATION is specified after the keyword)

Card 1 2 3 4 5 6 7 8

Variable	VX	VY	VZ					
Type	F	F	F					
Default	0.0	0.0	0.0					
Remarks								

Optional Card (Required if WARPAGE is specified after the keyword)

Card 1 2 3 4 5 6 7 8

Variable	SN ₁	SN ₂						
Type	I	I						
Default	none	none						
Remarks								

VARIABLE	DESCRIPTION
EID	Element ID. A unique ID is generally required, i.e., EID must be different from the element ID's also defined under *ELEMENT_DISCRETE and *ELEMENT_SEATBELT. If the parameter, BEAM, is set to 1 on the keyword input for *DATABASE_BINARY_D3PLOT, the null beams used for visualization are not created for the latter two types, and the ID's used for the discrete elements and the seatbelt elements can be identical to those defined here.
PID	Part ID, see *PART.
N1	Nodal point (end) 1.
N2	Nodal point (end) 2. This node is optional for the spot weld, beam type 9, since if it not defined it will be created automatically and given a non-conflicting nodal point ID. Nodes N1 and N2 are automatically positioned for the spot weld beam element. For the zero length discrete beam elements where one end is attached to ground, set N2=-N1. In this case, a fully constrained nodal point will be created with a unique ID for node N2.
N3	Nodal point 3 for orientation. The third node, N3, is optional for beam types 3, 6, 7, 8, and 9 if the latter, type 9, has a circular cross section. The third node is used for the discrete beam, type 6, if and only if SCOOR is set to 2.0 in the *SECTION_BEAM input, but even in this case it is optional. An orientation vector can be defined directly by using the option, ORIENTATION. In this case N3 can be defined as zero.
RT1, RT2	Release conditions for translations at nodes N1 and N2, respectively: EQ.0: no translational degrees-of-freedom are released EQ.1: x-translational degree-of-freedom EQ.2: y-translational degree-of-freedom EQ.3: z-translational degree-of-freedom EQ.4: x and y-translational degrees-of-freedom EQ.5: y and z-translational degrees-of-freedom EQ.6: z and x-translational degrees-of-freedom EQ.7: x, y, and z-translational degrees-of-freedom (3DOF) This option does not apply to the spot weld, beam type 9.
RR1, RR2	Release conditions for rotations at nodes N1 and N2, respectively: EQ.0: no rotational degrees-of-freedom are released EQ.1: x-rotational degree-of-freedom EQ.2: y-rotational degree-of-freedom EQ.3: z-rotational degree-of-freedom EQ.4: x and y-rotational degrees-of-freedom EQ.5: y and z-rotational degrees-of-freedom

VARIABLE	DESCRIPTION
	EQ.6: z and x-rotational degrees-of-freedom EQ.7: x, y, and z-rotational degrees-of-freedom (3DOF) This option does not apply to the spot weld, beam type 9.
LOCAL	Coordinate system option: EQ.1: global coordinate system EQ.2: local coordinate system (default)
PARM1	Based on beam type: Type.EQ.1: beam thickness, s direction at node 1 Type.EQ.2: area Type.EQ.3: area Type.EQ.4: beam thickness, s direction at node 1 Type.EQ.5: beam thickness, s direction at node 1 Type.EQ.6: volume, see description for VOL below. Type.EQ.7: beam thickness, s direction at node 1 Type.EQ.8: beam thickness, s direction at node 1 Type.EQ.9: beam thickness, s direction at node 1
PARM2	Based on beam type: Type.EQ.1: beam thickness, s direction at node 2 Type.EQ.2: I_{ss} Type.EQ.3: ramp-up time for dynamic relaxation Type.EQ.4: beam thickness, s direction at node 2 Type.EQ.5: beam thickness, s direction at node 2 Type.EQ.6: geometric inertia Type.EQ.6: Inertia, see description for INER below. Type.EQ.7: beam thickness, s direction at node 2 Type.EQ.8: beam thickness, s direction at node 2 Type.EQ.9: beam thickness, s direction at node 2
PARM3	Based on beam type: Type.EQ.1: beam thickness, t direction at node 1 Type.EQ.2: I_{tt} Type.EQ.3: initial stress for dynamic relaxation Type.EQ.4: beam thickness, t direction at node 1 Type.EQ.5: beam thickness, t direction at node 1 Type.EQ.6: local coordinate ID Type.EQ.7: not used. Type.EQ.8: not used. Type.EQ.9: beam thickness, t direction at node 1
PARM4	Based on beam type: Type.EQ.1: beam thickness, t direction at node 2 Type.EQ.2: I_{tt} Type.EQ.3: not used

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	Type.EQ.4: beam thickness, t direction at node 2 Type.EQ.5: beam thickness, t direction at node 2 Type.EQ.6: area Type.EQ.7: not used. Type.EQ.8: not used. Type.EQ.9: beam thickness, t direction at node 2
PARM5	Based on beam type: Type.EQ.1: not used Type.EQ.2: shear area Type.EQ.3: not used Type.EQ.4: not used Type.EQ.5: not used Type.EQ.6: offset Type.EQ.7: not used. Type.EQ.8: not used. Type.EQ.9: print flag to SWFORC file. The default is taken from the SECTION_BEAM input. To override set PARM5 to 1.0 to suppress printing, and to 2.0 to print.
STYPE	Section type (A format): EQ.SECTION_01: Circular EQ.SECTION_11: T-shape1 EQ.SECTION_02: Tubular EQ.SECTION_12: I-shape1 EQ.SECTION_03: L-shape EQ.SECTION_13: Channel1 EQ.SECTION_04: I-shape EQ.SECTION_14: Z-shape EQ.SECTION_05: Channel EQ.SECTION_15: Channel2 EQ.SECTION_06: T-shape EQ.SECTION_16: T-shape2 EQ.SECTION_07: Box-shape EQ.SECTION_17: Box-shape1 EQ.SECTION_08: Square EQ.SECTION_18: Hexagon EQ.SECTION_09: Cross EQ.SECTION_19: Hat-shape EQ.SECTION_10: H-shape EQ.SECTION_20: Hat-shape1
D1-D6	Input parameters for section option using STYPE above.
PID1	Optional part ID for spot weld element type 9.
PID2	Optional part ID for spot weld element type 9.
VOL	Volume of discrete beam and scalar beam. If the mass density of the material model for the discrete beam is set to unity, the magnitude of the lumped mass can be defined here instead. This lumped mass is partitioned to the two nodes of the beam element. The translational time step size for the type 6 beam is dependent on the volume, mass density, and the translational stiffness values, so it is important to define this parameter. Defining the volume is also essential for mass scaling if the type 6 beam controls the time step size.

VARIABLE	DESCRIPTION
INER	Mass moment of inertia for the six degree of freedom discrete beam and scalar beam. This lumped inertia is partitioned to the two nodes of the beam element. The rotational time step size for the type 6 beam is dependent on the lumped inertia and the rotational stiffness values, so it is important to define this parameter if the rotational springs are active. Defining the rotational inertia is also essential for mass scaling if the type 6 beam rotational stiffness controls the time step size.
CID	Coordinate system ID for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID=0, a default coordinate system is defined in the global system.
DOFN1	Active degree-of-freedom at node 1, a number between 1 to 6 where 1, 2, and 3 are the x, y, and z-translations and 4, 5, and 6 are the x, y, and z-rotations. This degree-of-freedom acts in the local system given by CID above. This input applies to material model type 146.
DOFN2	Active degree-of-freedom at node 2, a number between 1 to 6. This degree-of-freedom acts in the local system given by CID above. This input applies to material model type 146.
CID1	Coordinate system ID at node 1 for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID1=0, a default coordinate system is defined in the global system.
CID2	Coordinate system ID at node 2 for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID2=0, a default coordinate system is defined in the global system.
DOFNS	Active degrees-of-freedom at node 1 and node 2. A two-digit number, the first for node 1 and the second for node 2, between 11 to 66 is expected where 1, 2, and 3 are the x, y, and z-translations and 4, 5, and 6 are the x, y, and z-rotations. These degrees-of-freedom acts in the local system given by CID1 and CID2 above. This input applies to material model type 146. If DOFNS=12 the node one has an x-translation and node 2 has a y translation.
WX1-WZ1	Offset vector at nodal point N1. See Remark 8.
WX2-WZ2	Offset vector at nodal point N2. Set Remark 8.
VX,VY, VZ	Orientation vector at node N1. In this case the orientation nodal point N3, is defined as zero.
SN1	Scalar nodal point (end) 1. This node is required for the WARPAGE option.
SN2	Scalar nodal point (end) 2. This node is required for the WARPAGE option.

Remarks:

1. A plane through N_1 , N_2 , and N_3 defines the orientation of the principal r-s plane of the beam, see Figure 14.1.
2. This option applies to all three-dimensional beam elements. The released degrees-of-freedom can be either global, or local relative to the local beam coordinate system, see Figure 14.1. A local coordinate system is stored for each node of the beam element and the orientation of the local coordinate systems rotates with the node. To properly track the response, the nodal points with a released resultant are automatically replaced with new nodes to accommodate the added degrees-of-freedom. Then constraint equations are used to join the nodal points together with the proper release conditions imposed. **Consequently, nodal points which belong to beam elements which have release conditions applied cannot be subjected to other constraints such as applied displacement /velocity/acceleration boundary conditions, nodal rigid bodies, nodal constraint sets, or any of the constraint type contact definitions.** Force type loading conditions and penalty based contact algorithms may be used with this option.
3. Please note that this option may lead to nonphysical constraints if the translational degrees-of-freedom are released, but this should not be a problem if the displacements are infinitesimal.
4. If the second card is not defined for the resultant beam or if the area, A , is not defined the properties are taken from the cross section cards, see *SECTION_BEAM.
5. Do not define for discrete beams (beam type 6), see *SECTION_BEAM.
6. Define for resultant beam elements only, see *SECTION_BEAM.
7. The stress resultants are output in local coordinate system for the beam. Stress information is optional and is also output in the local system for the beam.
8. Beam offsets are sometimes necessary for correctly modeling beams that act compositely with other elements such as shells or other beams. When the OFFSET option is specified, global X, Y, and Z components of two offset vectors are given, one vector for each of the two beam nodes. The offset vector extends from the beam node (N_1 or N_2) to the reference axis of the beam. The beam reference axis lies at the origin of the local s and t axes, i.e., halfway between the outermost surfaces of the beam cross-section. Note that for cross-sections that are not doubly symmetric, e.g, a T-section, the reference axis does not pass through the centroid of the cross-section. Beam offsets can only be defined for Hughes-Liu beams (ELFORM=1).

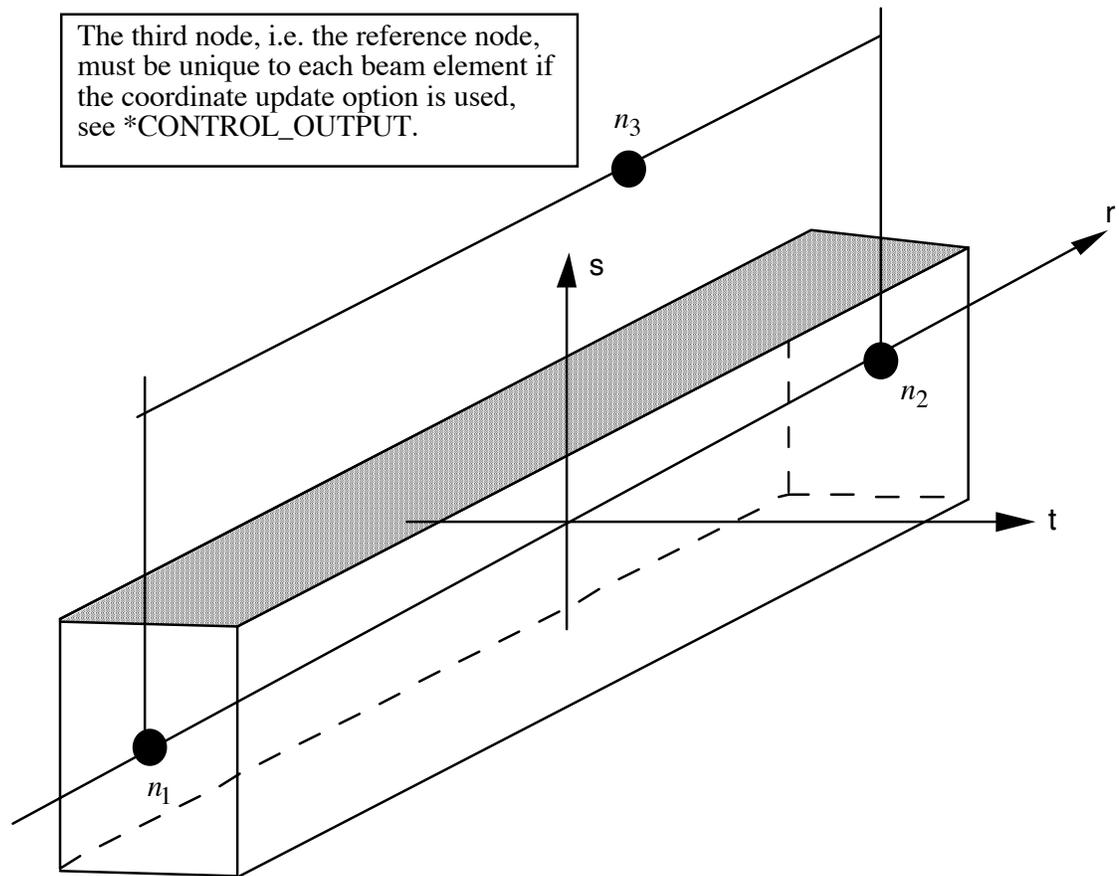


Figure 14.1. LS-DYNA beam elements. Node n_3 determines the initial orientation of the cross section.

***ELEMENT**

***ELEMENT_DIRECT_MATRIX_INPUT**

***ELEMENT_DIRECT_MATRIX_INPUT**

Purpose: Define an element consisting of mass, damping, stiffness, and inertia matrices in a specified file which follows the format used in the direct matrix input, DMIG, of NASTRAN. The supported format is the type 6 symmetric matrix in real double precision. LS-DYNA supports both the standard and the extended precision formats. The mass and stiffness matrices are required. The inertia matrix is required when using *LOAD_BODY_OPTION to correctly compute the action of a prescribed base acceleration on the superelement, otherwise the inertia matrix is unused. The damping matrix is optional. The combination of these matrices is referred to as a superelement. Three input cards are required for each superelement.

The degrees-of-freedom for this superelement may consist of generalized coordinates as well as nodal point quantities. Degrees-of-freedom, defined using *NODE input, are called attachment nodes. Only attachment nodes are included in the output to the ASCII and binary databases.

The matrices for a given superelement can be of different order. However, the explicit integration scheme requires the inversion of the union of the element mass matrix and nodal masses associated with attachment nodes. Any degree of freedom included in the other (stiffness, damping, inertia) matrices but without nonzero columns in the combined mass matrix will be viewed as massless and constrained not to move. After deleting zero rows and columns the combined mass matrix is required to be positive definite.

The inertia matrix is required to have 3 columns which corresponds to the 3 global coordinates. It is used to compute the forces acting on the superelement by multiplying the inertia matrix times the gravitational acceleration specified via *LOAD_BODY_OPTION.

There is no assumption made on the order of the matrices nor the sparse matrix structure of the element matrices except that they are symmetric and the combined mass matrix is invertible as described above.

Multiple elements may be input using *ELEMENT_DIRECT_MATRIX_INPUT. They may share attachment nodes with other direct matrix input elements. Only *BOUNDARY_PRESCRIBED_MOTION and global constraints imposed via *NODE or *BOUNDARY_SPC on attachment nodes can be applied in explicit applications. Implicit applications can have additional constraints on attachment nodes.

Card Format (I10)

Card 1 1 2 3 4 5 6 7 8

Variable	EID	IFRMT						
Type	I	0						

Card Format (A80)

Card 2

Variable	FILENAME
Type	C

Card Format (4A10)

Card 3 1 2 3 4 5 6 7 8

Variable	MASS	DAMP	STIF	INERT				
Type	C	C	C	C				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Super element ID.
IFRMT	Format: EQ.0: standard format NE.0: extended precision format
MASS	Name of mass matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN.
DAMP	Name of damping matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN.
STIF	Name of stiffness matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN.
INERT	Name of inertia matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN. This file must be present when *LOAD_BODY is used to put gravitational forces on the model.

*ELEMENT

*ELEMENT_DISCRETE

*ELEMENT_DISCRETE

Purpose: Define a discrete (spring or damper) element between two nodes or a node and ground. It is recommended that beam type 6, see *ELEMENT_BEAM and SECTION_BEAM, be used whenever possible, especially if orientation is specified. The latter option tends to be more accurate and cost effective. The *ELEMENT_DISCRETE option is no longer being developed and extended.

Note: These elements enter into the time step calculations. Care must be taken to ensure that the nodal masses connected by the springs and dampers are defined and unrealistically high stiffness and damping values must be avoided. **All rotations are in radians.**

Card Format (5I8,E16.0,I8,E16.0)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N1	N2	VID	S	PF	OFFSET
Type	I	I	I	I	I	F	I	F
Default	none	none	none	none	0	1.	0	0

VARIABLE

DESCRIPTION

EID	Element ID. A unique number is required. Since null beams are created for visualization, this element ID should not be identical to element ID's defined for ELEMENT_BEAM and ELEMENT_SEATBELT.
PID	Part ID, see *PART.
N1	Nodal point 1.
N2	Nodal point 2. If zero, the spring/damper connects node N1 to ground.
VID	Orientation option. The orientation option should be used cautiously since forces, which are generated as the nodal points displace, are not orthogonal to rigid body rotation unless the nodes are coincident.. The type 6, 3D beam element, is recommended when orientation is required with the absolute value of the parameter SCOR set to 2 or 3, since this option avoids rotational constraints. EQ.0: the spring/damper acts along the axis from node N1 to N2, NE.0: the spring/damper acts along the axis defined by the orientation vector, VID defined in the *DEFINE_SD_ORIENTATION section.

VARIABLE	DESCRIPTION
S	Scale factor on forces.
PF	Print flag: EQ.0: forces are printed in DEFORC file, (see *DATABASE_OPTION), EQ.1: forces are not printed DEFORC file.
OFFSET	Initial offset. The initial offset is a displacement or rotation at time zero. For example, a positive offset on a translational spring will lead to a tensile force being developed at time zero.

*ELEMENT

*ELEMENT_INERTIA

*ELEMENT_INERTIA_{OPTION}

Available options include:

<BLANK>

OFFSET

to allow the lumped mass and inertia tensor to be offset from the nodal point. The nodal point can belong to either a deformable or rigid node.

Purpose: Define a lumped inertia element assigned to a nodal point.

Card Format (3I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	NID	CSID							
Type	I	I	I							
Default	none	none	none							
Remarks			1							

Card Format (7E10.0)

Card 1 2 3 4 5 6 7 8

Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ	MASS	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Remarks		2	2		2			

Define if and only if the **OFFSET** option is active. Card Format (3E10.0)

Card 1 2 3 4 5 6 7 8

Variable	X-OFF	Y-OFF	Z-OFF					
Type	F	F	F					
Default	0.	0.	0.					
Remarks		2	2					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Element ID. A unique number must be used.
NID	Node ID. Node to which the mass is assigned.
CSID	Coordinate set ID EQ.0: global inertia tensor GE.1: principal moments of inertias with orientation vectors defined by Coordinate set CSID. See *DEFINE_COORDINATE_SYSTEM and *DEFINE_COORDINATE_VECTOR.
IXX	XX component of inertia tensor.
IXY	XY component of inertia tensor.
IXZ	XZ component of inertia tensor.
IYY	YY component of inertia tensor.
IYZ	YZ component of inertia tensor.
IZZ	ZZ component of inertia tensor.
MASS	Lumped mass
X-OFF	x-offset from nodal point.
Y-OFF	y-offset from nodal point.
Z-OFF	z-offset from nodal point.

Remarks:

1. The coordinate system cannot be defined for this element using the option *DEFINE_COORDINATE_NODE.
2. If CSID is defined then IXY, IXZ and IYZ are set to zero. The nodal inertia tensor must be positive definite, i.e., its determinant must be greater than zero, since its inverse is required. This check is done after the nodal inertia is added to the defined inertia tensor.

***ELEMENT_MASS_{OPTION}**

Available options include:

<BLANK>

NODE_SET

Purpose: Define a lumped mass element assigned to a nodal point or equally distributed to the nodes of a node set.

(Note: NODE_SET option is available starting with the R3 release of Version 971.)

Card Format (2I8,E16.0)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	ID	MASS	PID						
Type	I	I	F	I						
Default	none	None	0.	none						

VARIABLE	DESCRIPTION
EID	Element ID. A unique number is recommended. The nodes in a node set share the same element ID.
ID	Node ID or node set ID if the NODE_SET option is active. This is the node or node set to which the mass is assigned.
MASS	Mass value. When the NODE_SET option is active, the mass is equally distributed to all nodes in a node set.
PID	Part ID. This input is optional.

*ELEMENT

*ELEMENT_MASS_PART

*ELEMENT_MASS_PART_{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define additional non-structural mass to be distributed by an area weighted distribution to all nodes of a given part ID. As an option, the total mass can be defined and the additional non-structural mass is computed. This option applies to all part ID's defined by shell elements.

(Note: SET option is available starting with the R3 release of Version 971.)

Card Format (I8,2E16.0)

Card 1 2 3 4 5 6 7 8 9 10

Variable	ID	ADDMASS	FINMASS						
Type	I	F	F						
Default	none	0.	0.						

VARIABLE

DESCRIPTION

ID Part or part set ID if the SET option is active. A unique number must be used.

ADDMASS Added translational mass to be distributed to the nodes of the part ID or part set ID. Set to zero if TOTMASS is nonzero. Since the additional mass is not included in the time step calculation of the elements in the PID or SID, ADDMASS must be greater than zero.

FINMASS Final translational mass of the part ID or part set ID. The total mass of the PID or SID is computed and subtracted from the final mass of the part or part set to obtain the added translational mass, which must exceed zero. Set FINMASS to zero if ADDMASS is nonzero. FINMASS is available in the R3 release of version 971.

***ELEMENT_PLOTEL**

Purpose: Define a null beam element for visualization.

Card Format (3I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	N ₁	N ₂							
Type	I	I	I							
Default	none	none	None							
Remarks	1									

VARIABLE

DESCRIPTION

- | | |
|-----|---|
| EID | Element ID. A unique number must be used. |
| N1 | Nodal point (end) 1. |
| N2 | Nodal point (end) 2. |

Remarks:

1. Part ID, 10000000, is assigned to PLOTEL elements.
2. PLOTEL element ID's must be unique with respect to other beam elements.

*ELEMENT

*ELEMENT_SEATBELT

*ELEMENT_SEATBELT

Purpose: Define a seat belt element.

Card Format (5I8,E16.0,2I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N ₁	N ₂	SBRID	SLEN	N ₃	N ₄	
Type	I	I	I	I	I	F	I	I	
Default	none	none	none	none	none	0.0	0	0	
Remarks					1	2	3		

VARIABLE

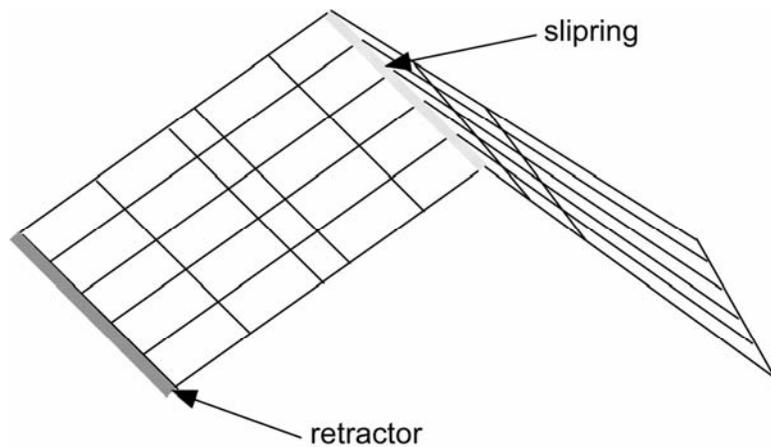
DESCRIPTION

EID	Element ID. A unique number is required. Since null beams are created for visualization, this element ID should not be identical to element ID's defined for ELEMENT_BEAM and ELEMENT_DISCRETE.
PID	Part ID
N1	Node 1 ID
N2	Node 2 ID
SBRID	Retractor ID, see *ELEMENT_SEATBELT_RETRACTOR.
SLEN	Initial slack length
N3	Optional node 3 ID. When N3>0 and N4>0, the elements becomes a shell seat belt element. The thickness of the shell seatbelt is defined in *SECTION_SHELL, not in *SECTION_SEATBELT. The shell-type seatbelt must be of a rectangular shape as shown in Figure 14.2 and contained in a logically regular mesh.
N4	Node 4 ID, which is required if and only if N3 is defined.

Remarks:

1. The retractor ID should be defined only if the element is initially **inside** a retractor, see *ELEMENT_SEATBELT_RETRACTOR.

2. Belt elements are single degree of freedom elements connecting two nodes. When the strain in an element is positive (i.e. the current length is greater than the unstretched length), a tension force is calculated from the material characteristics and is applied along the current axis of the element to oppose further stretching. The unstretched length of the belt is taken as the initial distance between the two nodes defining the position of the element plus the initial slack length.
3. Seatbelt shell elements are a new feature in version 971 and must be used with caution. The seatbelt shells distribute the loading on the surface of the dummy more realistically than the two node belt elements. For the seatbelt shells to work with slippings and retractors it is necessary to use a logically regular mesh of quadrilateral elements. A seatbelt defined by a part ID must not be disjoint.



Top view:

RN5				SN5			
	RE4			SRE14	SRE24		
RN4				SN4			
	RE3			SRE13	SRE23		
RN3				SN3			
	RE2			SRE12	SRE22		
RN2				SN2			
	RE1			SRE11	SRE21		
RN1				SN1			

Figure 14.2. Definition of seatbelt shell elements. The ordering of the nodes and elements are important for seatbelt shells. See the input descriptions for SECTION_SHELL, ELEMENT_SEATBELT_RETRACTOR and ELEMENT_SEATBELT_SLIPRING.

*ELEMENT

*ELEMENT_SEATBELT_ACCELEROMETER

*ELEMENT_SEATBELT_ACCELEROMETER

Purpose: Define seat belt accelerometer. The accelerometer is fixed to a rigid body containing the three nodes defined below. Whenever computed accelerations are compared to experimental results or whenever computed accelerations are compared between different runs, accelerometers are essential. Raw nodal accelerations contain considerable numerical noise and their comparisons are generally meaningless and, therefore, misleading.

Card	1	2	3	4	5	6	7	8
Variable	SBACID	NID1	NID2	NID3	IGRAV	INTOPT	MASS	
Type	I	I	I	I	I	I	F	
Default	0	0	0	0	0	0	0.	
Remarks								

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SBACID	Accelerometer ID. A unique number must be used.
NID1	Node 1 ID
NID2	Node 2 ID
NID3	Node 3 ID
IGRAV	Gravitational accelerations due to body force loads. EQ.0: included in acceleration output EQ.1: removed from acceleration output
INTOPT	Integration option. If the accelerometer undergoes rigid body translation without rotation this option has no effect; however, if rotation occurs, option 1 may provide better agreement with the output of the accelerometer. EQ.0: velocities are integrated from the global accelerations and transformed into the local system of the accelerometer EQ.1: velocities are integrated directly from the local accelerations of the accelerometer.

VARIABLE	DESCRIPTION
MASS	Optional added mass for accelerometer. This mass is equally distributed to nodal points NID1, NID2, and NID3. This option avoids the need to use the *ELEMENT_MASS keyword input if additional mass is required.

Remarks:

The presence of the accelerometer means that the accelerations and velocities of node 1 will be output to **all** output files in local instead of global coordinates.

The local coordinate system is defined by the three nodes as follows:

- local **x** from node 1 to node 2,
- local **z** perpendicular to the plane containing nodes, 1, 2, and 3 ($\mathbf{z} = \mathbf{x} \times \mathbf{a}$), where **a** is from node 1 to node 3),
- local **y** = $\mathbf{z} \times \mathbf{x}$.

The three nodes should all be part of the same rigid body. The local axis then rotates with the body.

*ELEMENT

*ELEMENT_SEATBELT_PRETENSIONER

*ELEMENT_SEATBELT_PRETENSIONER

Purpose: Define seat belt pretensioner. A combination with sensors and retractors is also possible.

Card 1 2 3 4 5 6 7 8

Variable	SBPRID	SBPRTY	SBSID1	SBSID2	SBSID3	SBSID4		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		
Remarks			1					

Card

Variable	SBRID	TIME	PTLCID	LMTFRC				
Type	I	F	I	F				
Default	0	0.0	0	0				
Remarks								

VARIABLE

DESCRIPTION

- SBPRID Pretensioner ID. A unique number has to be used.
- SBPRTY Pretensioner type (see Remark 2 below):
EQ.1: pyrotechnic retractor with force limits,
EQ.2: pre-loaded spring becomes active,
EQ.3: lock spring removed,
EQ.4: force versus time retractor.
EQ.5: pyrotechnic retractor (old type in version 950) but with optional force limiter, LMTFRC.
EQ.6: combination of types 4 and 5 as described in the notes below.
EQ.7: independent pretensioner/retractor.

VARIABLE	DESCRIPTION
SBSID1	Sensor 1, see *ELEMENT_SEATBELT_SENSOR.
SBSID2	Sensor 2, see *ELEMENT_SEATBELT_SENSOR.
SBSID3	Sensor 3, see *ELEMENT_SEATBELT_SENSOR.
SBSID4	Sensor 4, see *ELEMENT_SEATBELT_SENSOR.
SBRID	Retractor number (SBPRTY = 1, 4, 5, or 6) or spring element number (SBPRTY = 2 or 3).
TIME	Time between sensor triggering and pretensioner acting.
PTLCID	Load curve for pretensioner (Time after activation, Pull-in) (SBPRTY = 1, 4, 5 or 6).
LMTFRC	Optional limiting force for retractor type 5. If zero, this option is ignored.

Remarks:

1. At least one sensor should be defined.

Pretensioners allow modeling of seven types of active devices which tighten the belt during the initial stages of a crash. Types 1 and 5 represent a pyrotechnic device which spins the spool of a retractor, causing the belt to be reeled in. The user defines a pull-in versus time curve which applies once the pretensioner activates. Types 2 and 3 represent preloaded springs or torsion bars which move the buckle when released. The pretensioner is associated with any type of spring element including rotational. Note that the preloaded spring, locking spring and any restraints on the motion of the associated nodes are defined in the normal way; the action of the pretensioner is merely to cancel the force in one spring until (or after) it fires. With the second type, the force in the spring element is canceled out until the pretensioner is activated. In this case the spring in question is normally a stiff, linear spring which acts as a locking mechanism, preventing motion of the seat belt buckle relative to the vehicle. A preloaded spring is defined in parallel with the locking spring. This type avoids the problem of the buckle being free to 'drift' before the pretensioner is activated. Types 4, 6, and 7, force types, are described below.

To activate the pretensioner, the following sequence of events must occur:

1. Any one of up to four sensors must be triggered.
 2. Then a user-defined time delay occurs.
 3. Then the pretensioner acts.
2. In the 950 version of LS-DYNA, there are three types of seatbelt pretensioners that can be simulated. Types 2 and 3 are simple triggers for activating or deactivating springs, which

then pull on the buckle. No changes have been made to these, and they are not discussed here. The type 1 pretensioner is intended to simulate a pyrotechnic retractor. The user inputs a load curve describing the pull-in of the pretensioner as a function of time. This pretensioner type interacts with the retractor, forcing it to pull in the amount of belt indicated. It works well, and does exactly what it says it will do, but it can be difficult to use in practice. The reason for this is that it has no regard for the forces being exerted on the belt. If a pull-in of 20mm is specified at a particular time, then 20mm of belt will be pulled in, even if this results in unrealistic forces in the seatbelt. Furthermore, there was no explicit way to turn this pretensioner off. Once defined, it overrode the retractor completely, and the amount of belt passing into or out of the retractor depended solely on the load curve specified.

In the 970 version of LS-DYNA, the behavior of the type 1 pretensioner was changed due to user feedback regarding these shortcomings. The behavior now is fundamentally simpler, though a bit confusing to explain. Each retractor has a loading (and optional unloading) curve that describes the force on the belt element as a function of the amount of belt that has been pulled out of the retractor since the retractor locked. The new type 1 pretensioner acts as a shift of this retractor load curve. An example will make this clear. Suppose at a particular time that 5mm of belt material has left the retractor. The retractor will respond with a force corresponding to 5mm pull-out on its loading curve. But suppose this retractor has a type 1 pretensioner defined, and at this instant of time the pretensioner specifies a pull-in of 20mm. The retractor will then respond with a force that corresponds to (5mm + 20mm) on its loading curve. This results in a much larger force. The effect can be that belt material will be pulled in, but unlike in the 950 version, there is no guarantee. The benefit of this implementation is that the force vs. pull-in load curve for the retractor is followed and no unrealistic forces are generated. Still, it may be difficult to produce realistic models using this option, so two new types of pretensioners have been added. These are available in 970 versions 1300 and later.

The type 4 pretensioner takes a force vs. time curve, See Figure 14.3. Each time step, the retractor computes the desired force without regard to the pretensioner. If the resulting force is less than that specified by the pretensioner load curve, then the pretensioner value is used instead. As time goes on, the pretensioner load curve should drop below the forces generated by the retractor, and the pretensioner is then essentially inactive. This provides for good control of the actual forces, so no unrealistic values are generated. The actual direction and amount of belt movement is unspecified, and will depend on the other forces being exerted on the belt. This is suitable when the force the pretensioner exerts over time is known.

The type 5 pretensioner is essentially the same as the old type 1 pretensioner, but with the addition of a force limiting value. The pull-in is given as a function of time, and the belt is drawn into the retractor exactly as desired. However, if at any point the forces generated in the belt exceed the pretensioner force limit, then the pretensioner is deactivated and the retractor takes over. In order to prevent a large discontinuity in the force at this point, the loading curve for the retractor is shifted (in the abscissa) by the amount required to put the current (pull-out, force) on the load curve. For example, suppose the current force is 1000, and the current pull-out is -10 (10mm of belt has been pulled IN by the pretensioner). If the retractor would normally generate a force of 1000 after 25mm of belt had been pulled OUT, then the load curve is shifted to the left by 35,

and remains that way for the duration of the calculation. So that at the current pull-in of 10, it will generate the force normally associated with a pull out of 25. If the belt reaches a pull out of 5, the force will be as if it were pulled out 40 (5 + the shift of 35), and so on. This option is included for those who liked the general behavior of the old type 1 pretensioner, but has the added feature of the force limit to prevent unrealistic behavior.

The type 6 pretensioner is a variation of the type 4 pretensioner, with features of the type 5 pretensioner. A force vs. time curve is input and the pretensioner force is computed each cycle. The retractor linked to this pretensioner should specify a positive value for PULL, which is the distance the belt pulls out before it locks. As the pretensioner pulls the belt into the retractor, the amount of pull-in is tracked. As the pretensioner force decreases and drops below the belt tension, belt will begin to move back out of the retractor. Once PULL amount of belt has moved out of the retractor (relative to the maximum pull in encountered), the retractor will lock. At this time, the pretensioner is disabled, and the retractor force curve is shifted to match the current belt tension. This shifting is done just like the type 5 pretensioner. It is important that a positive value of PULL be specified to prevent premature retractor locking which could occur due to small outward belt movements generated by noise in the simulation.

The type 7 pretensioner is a simple combination of retractor and pretensioner. It is similar to the type 6 except for the following changes: when the retractor locks, the pretensioner is NOT disabled – it continues to exert force according to the force vs. time curve until the end of the simulation. (The force vs. time curve should probably drop to 0 at some time.) Furthermore, the retractor load curve is not shifted – the retractor begins to exert force according to the force vs. pull-out curve. These two forces are added together and applied to the belt. Thus, the pretensioner and retractor are essentially independent.

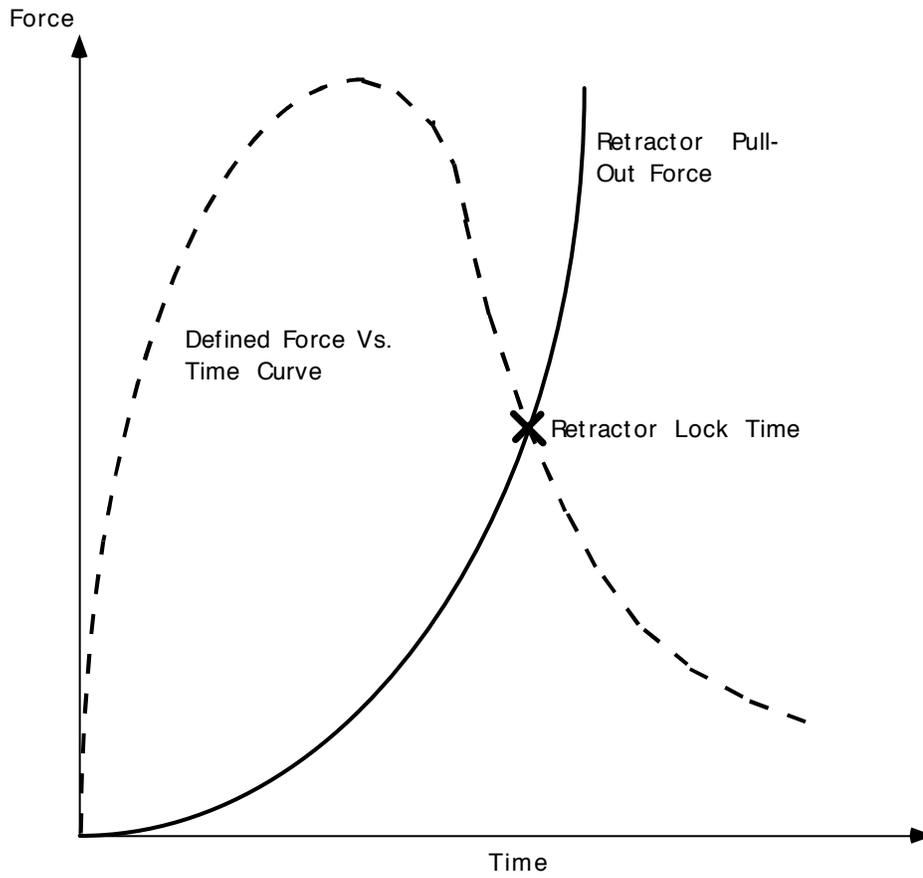


Figure 14.3. Force versus time pretensioner. At the intersection, the retractor locks.

ELEMENT_SEATBELT_RETRACTOR**ELEMENT*****ELEMENT_SEATBELT_RETRACTOR**

Purpose: Define seat belt retractor. See remarks below for seatbelt shell elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SBRID	SBRNID	SBID	SID1	SID2	SID3	SID4	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	
Remarks		1,2	2	3				

Card 2

Variable	TDEL	PULL	LLCID	ULCID	LFED			
Type	F	F	I	I	F			
Default	0.0	0.0	0	0	0.0			
Remarks			4	5				

VARIABLE**DESCRIPTION**

SBRID	Retractor ID. A unique number has to be used.
SBRNID	Retractor node ID
SBID	Seat belt element ID
SID1	Sensor ID 1
SID2	Sensor ID 2
SID3	Sensor ID 3
SID4	Sensor ID 4

*ELEMENT

*ELEMENT_SEATBELT_RETRACTOR

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TDEL	Time delay after sensor triggers.
PULL	Amount of pull-out between time delay ending and retractor locking, a length value.
LLCID	Load curve for loading (Pull-out, Force), see Figure 14.6.
ULCID	Load curve for unloading (Pull-out, Force), see Figure 14.6.
LFED	Fed length, see explanation below.

Remarks:

1. The retractor node should not be on any belt elements. The element defined should have one node coincident with the retractor node but should not be inside the retractor.
2. When $SBRNID < 0$, this retractor is for shell-type seatbelt, $-SBRNID$ is the *SET_NODE containing RN1, RN2, ...RN5. SBID is then *SET_SHELL_LIST. Note that the numbering of $-SBRNID$, SBID has to be consistent in the direction of numbering. For example, if *SET_NODE for SBRNID has nodes of (RN1, RN2, RN3, RN4, RN5) then *SET_SHELL_LIST for SBID should have elem. of (RE1, RE2, RE3, RE4). See Figure 14.2.
3. At least one sensor should be defined.
4. The first point of the load curve should be $(0, T_{min})$. T_{min} is the minimum tension. All subsequent tension values should be greater than T_{min} .
5. The unloading curve should start at zero tension and increase monotonically (i.e., no segments of negative or zero slope).

Retractors allow belt material to be paid out into a belt element. Retractors operate in one of two regimes: unlocked when the belt material is paid out, or reeled in under constant tension and locked when a user defined force-pullout relationship applies.

The retractor is initially unlocked, and the following sequence of events must occur for it to become locked:

1. Any one of up to four sensors must be triggered. (The sensors are described below.)
2. Then a user-defined time delay occurs.
3. Then a user-defined length of belt must be paid out (optional).
4. Then the retractor locks and once locked, it remains locked.

In the unlocked regime, the retractor attempts to apply a constant tension to the belt. This feature allows an initial tightening of the belt and takes up any slack whenever it occurs. The tension value is taken from the first point on the force-pullout load curve. The maximum rate of pull out or pull in is given by $0.01 \times \text{fed length}$ per time step. Because of this, the constant tension value is not always achieved.

In the locked regime, a user-defined curve describes the relationship between the force in the attached element and the amount of belt material paid out. If the tension in the belt subsequently relaxes, a different user-defined curve applies for unloading. The unloading curve is followed until the minimum tension is reached.

The curves are defined in terms of initial length of belt. For example, if a belt is marked at 10mm intervals and then wound onto a retractor, and the force required to make each mark emerge from the (locked) retractor is recorded, the curves used for input would be as follows:

- 0 Minimum tension (should be > zero)
- 10mm Force to emergence of first mark
- 20mm Force to emergence of second mark
- ..
- ..
- ..

Pyrotechnic pretensions may be defined which cause the retractor to pull in the belt at a predetermined rate. This overrides the retractor force-pullout relationship from the moment when the pretensioner activates.

If desired, belt elements may be defined which are initially inside the retractor. These will emerge as belt material is paid out, and may return into the retractor if sufficient material is reeled in during unloading.

Elements e2, e3 and e4 are initially inside the retractor, which is paying out material into element e1. When the retractor has fed L_{crit} into e1, where

$$L_{crit} = \text{fed length} - 1.1 \times \text{minimum length}$$

(minimum length defined on belt material input)

(fed length defined on retractor input)

Element e2 emerges with an unstretched length of $1.1 \times \text{minimum length}$; the unstretched length of element e1 is reduced by the same amount. The force and strain in e1 are unchanged; in e2, they are set equal to those in e1. The retractor now pays out material into e2.

If no elements are inside the retractor, e2 can continue to extend as more material is fed into it.

As the retractor pulls in the belt (for example, during initial tightening), if the unstretched length of the mouth element becomes less than the minimum length, the element is taken into the retractor.

To define a retractor, the user enters the retractor node, the 'mouth' element (into which belt material will be fed), e1 in Figure 12.3, up to 4 sensors which can trigger unlocking, a time delay, a payout delay (optional), load and unload curve numbers, and the fed length. The retractor node is typically part of the vehicle structure; belt elements should not be connected to this node directly, but any other feature can be attached including rigid bodies. The mouth element should have a node coincident with the retractor but should not be inside the retractor. The fed length would typically be set either to a typical element initial length, for the distance between painted marks on a real belt for comparisons with high speed film. The fed length should be at least three times the minimum length.

If there are elements initially inside the retractor (e2, e3 and e4 in the Figure) they should not be referred to on the retractor input, but the retractor should be identified on the element input for these elements. Their nodes should all be coincident with the retractor node and should not be restrained or constrained. Initial slack will automatically be set to $1.1 \times$ minimum length for these elements; this overrides any user-defined value.

Weblockers can be included within the retractor representation simply by entering a 'locking up' characteristic in the force pullout curve, see Figure 14.5. The final section can be very steep (but must have a finite slope).

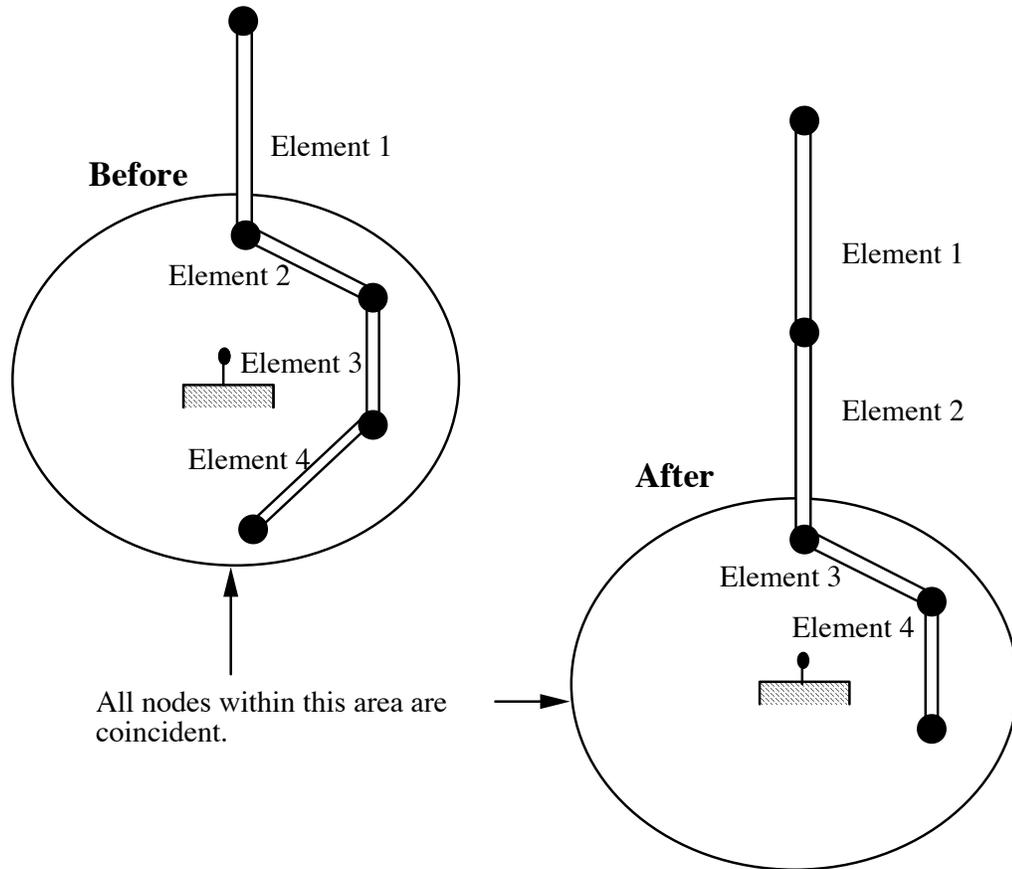


Figure 14.4. Elements in a retractor.

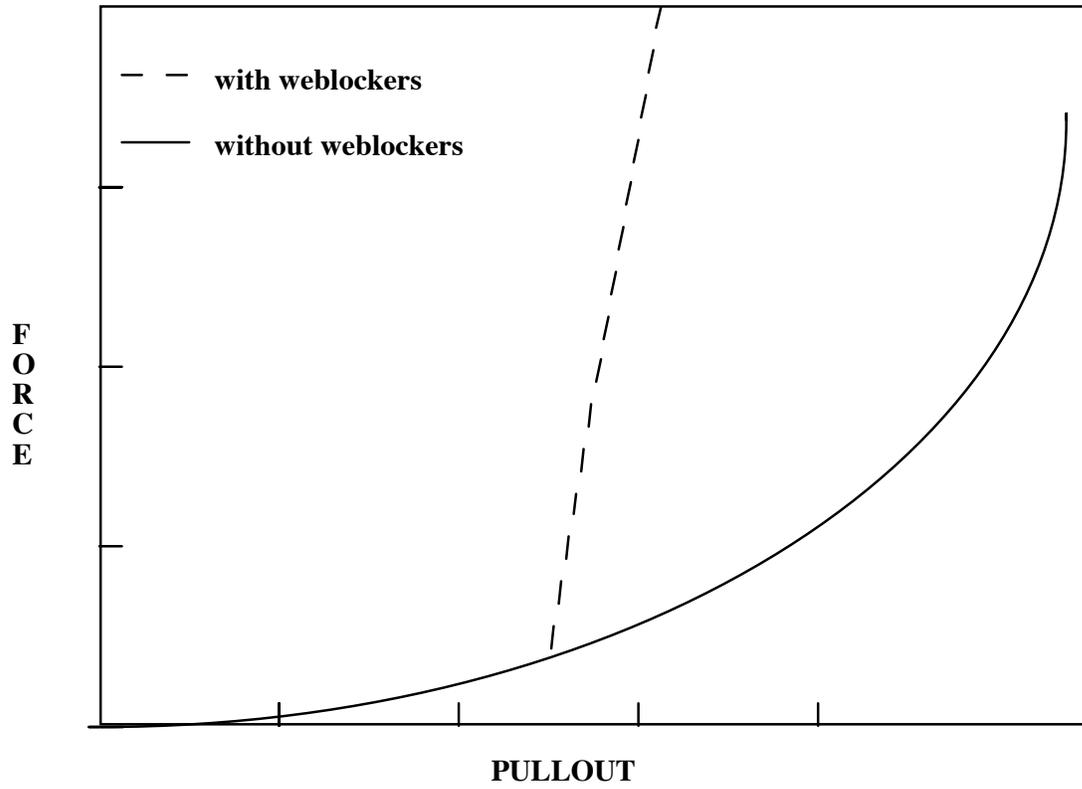


Figure 14.5. Retractor force pull characteristics.

***ELEMENT_SEATBELT_SENSOR**

***ELEMENT**

***ELEMENT_SEATBELT_SENSOR**

Purpose: Define seat belt sensor. Four types are possible, see explanation below.

Card 1 1 2 3 4 5 6 7 8

Variable	SBSID	SBSTYP	SBSFL					
Type	I	I	I					
Default	0	0	0					
Remarks								

Card 2 if SBSTYP=1

Card 2 1 2 3 4 5 6 7 8

Variable	NID	DOF	ACC	ATIME				
Type	I	I	F	F				
Default	0	0	0.0	0.0				
Remarks	1							

ELEMENT**ELEMENT_SEATBELT_SENSOR****Card 2 if SBSTYP=2**

Card 2 1 2 3 4 5 6 7 8

Variable	SBRID	PULRAT	PULTIM					
Type	I	F	F					
Default	0	0.0	0.0					
Remarks								

Card 2 if SBSTYP=3

Card 2 1 2 3 4 5 6 7 8

Variable	TIME							
Type	F							
Default	0.0							
Remarks								

Card 2 if SBSTYP=4

Card 2 1 2 3 4 5 6 7 8

Variable	NID1	NID2	DMX	DMN				
Type	I	I	F	F				
Default	0	0	0.0	0.0				
Remarks			2	2				

VARIABLE	DESCRIPTION
SBSID	Sensor ID. A unique number has to be used.
SBSTYP	Sensor type: EQ.1: acceleration of node, EQ.2: retractor pull-out rate, EQ.3: time, EQ.4: distance between nodes.
SBSFL	Sensor flag: EQ.0: sensor active during dynamic relaxation, EQ.1: sensor can be triggered during dynamic relaxation.
NID	Node ID of sensor
DOF	Degree of freedom: EQ.1: x, EQ.2: y, EQ.3: z.
ACC	Activating acceleration
ATIME	Time over which acceleration must be exceeded
SBRID	Retractor ID, see *ELEMENT_SEATBELT_RETRACTOR.
PULRAT	Rate of pull-out (length/time units)
PULTIM	Time over which rate of pull-out must be exceeded
TIME	Time at which sensor triggers
NID1	Node 1 ID
NID2	Node 2 ID
DMX	Maximum distance
DMN	Minimum distance

Remarks:

1. Node should not be on rigid body, velocity boundary condition, or other 'imposed motion' feature.
2. Sensor triggers when the distance between the two nodes is $d \geq d_{\max}$ or $d \leq d_{\min}$.

Sensors are used to trigger locking of retractors and activate pretensioners. Four types of sensors are available which trigger according to the following criteria:

Type 1 – When the magnitude of x-, y-, or z- acceleration of a given node has remained above a given level continuously for a given time, the sensor triggers. This does not work with nodes on rigid bodies.

Type 2 – When the rate of belt payout from a given retractor has remained above a given level continuously for a given time, the sensor triggers.

Type 3 – The sensor triggers at a given time.

Type 4 – The sensor triggers when the distance between two nodes exceeds a given maximum or becomes less than a given minimum. This type of sensor is intended for use with an explicit mass/spring representation of the sensor mechanism.

By default, the sensors are inactive during dynamic relaxation. This allows initial tightening of the belt and positioning of the occupant on the seat without locking the retractor or firing any pretensioners. However, a flag can be set in the sensor input to make the sensors active during the dynamic relaxation phase.

***ELEMENT_SEATBELT_SLIPRING**

Purpose: Define seat belt slip ring.

Card 1 2 3 4 5 6 7 8

Variable	SBSRID	SBID1	SBID2	FC	SBRNID	LTIME	FCS	
Type	I	I	I	F	I	F	F	
Default	0	0	0	0.0	0	1.0E20	0.0	
Remarks	1	1	1	yes	yes			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SBSRID	Slipring ID. A unique number has to be used. See remarks below for the treatment of sliprings for shell belt elements.
SBID1	Seat belt element 1 ID
SBID2	Seat belt element 2 ID
FC	Coulomb dynamic friction coefficient
SBRNID	Slip ring node, NID
LTIME	Slip ring lockup time. After this time no material is moved from one side of the slip ring to the other. This option is not active during dynamic relaxation.
FCS	Optional static Coulomb friction coefficient.

Remarks:

When SBRNID<0, this slipring is for shell-type seatbelt, -SBRNID is the *SET_NODE containing SN1, SN2, ...SN5. SBID1 and SBID2 are then *SET_SHELL_LIST. Note that the numbering of -SBRNID, SBID1 and SBID2 has to be consistent in the direction of numbering. For example if, *SET_NODE for SBRNID has nodes of (SN1, SN2, SN3, SN4, SN5) then *SET_SHELL_LIST for SBID1 should have elem. of (SRE11, SRE12, SRE13, SRE14) and *SET_SHELL_LIST for SBID2 should have elem. of (SRE21, SRE22, SRE23, SRE24). See Figure 14.2.

Elements 1 and 2 should share a node which is coincident with the slip ring node. The slip ring node should not be on any belt elements.

Sliprings allow continuous sliding of a belt through a sharp change of angle. Two elements (1 & 2 in Figure 14.6) meet at the slipring. Node B in the belt material remains attached to the slipring node, but belt material (in the form of unstretched length) is passed from element 1 to element 2 to achieve slip. The amount of slip at each timestep is calculated from the ratio of forces in elements 1 and 2. The ratio of forces is determined by the relative angle between elements 1 and 2 and the coefficient of friction, μ . The tension in the belts are taken as T_1 and T_2 , where T_2 is on the high tension side and T_1 is the force on the low tension side. Thus, if T_2 is sufficiently close to T_1 , no slip occurs; otherwise, slip is just sufficient to reduce the ratio T_2/T_1 to $e^{\mu\theta}$. No slip occurs if both elements are slack. The out-of-balance force at node B is reacted on the slipring node; the motion of node B follows that of slipring node.

If, due to slip through the slipring, the unstretched length of an element becomes less than the minimum length (as entered on the belt material card), the belt is remeshed locally: the short element passes through the slipring and reappears on the other side (see Figure 14.6). The new unstretched length of e1 is $1.1 \times$ minimum length. Force and strain in e2 and e3 are unchanged; force and strain in e1 are now equal to those in e2. Subsequent slip will pass material from e3 to e1. This process can continue with several elements passing in turn through the slipring.

To define a slipring, the user identifies the two belt elements which meet at the slipring, the friction coefficient, and the slipring node. The two elements must have a common node coincident with the slipring node. No attempt should be made to restrain or constrain the common node for its motion will automatically be constrained to follow the slipring node. Typically, the slipring node is part of the vehicle body structure and, therefore, belt elements should not be connected to this node directly, but any other feature can be attached, including rigid bodies.

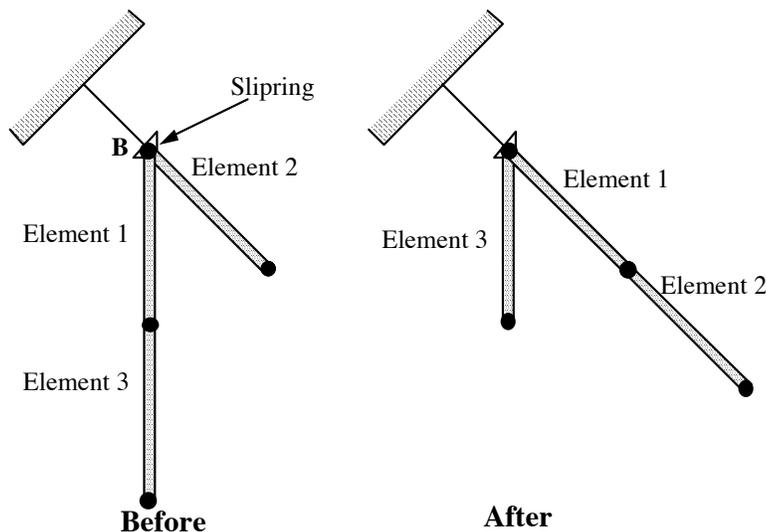


Figure 14.6. Elements passing through slipring.

***ELEMENT_SHELL_{OPTION}**

Available options include:

<BLANK>

THICKNESS

BETA or MCID

OFFSET

DOF

Purpose: Define three, four, six, and eight node elements including 3D shells, membranes, 2D plane stress, plane strain, and axisymmetric solids. The type of the element and its formulation is specified through the part ID (see *PART) and the section ID (see *SECTION_SHELL). Also, the thickness of each element can be specified when applicable on the element cards or else a default thickness value is used from the section definition. For orthotropic and anisotropic materials a local material angle (variable PSI) can be defined which is cumulative with the integration point angles specified in *SECTION_SHELL. Alternatively, the angle PSI can be determined by defining a local coordinate system, MCID. An offset option is available for moving the shell reference surface from the nodal points that define the shell.

For the shell formulation that uses additional nodal degrees-of-freedom, the option DOF is available to connect the nodes of the shell to corresponding scalar nodes. Four scalar nodes are required for element type 25 to model the thickness changes that require 2 additional degrees-of-freedom per shell node. Defining these nodes is optional, if left undefined, they will be automatically created.

Card Format (10I8)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	0	0	0	0
Remarks			3	3	3	3				

ELEMENT**ELEMENT_SHELL****Optional Card (Required if THICKNESS or BETA is specified after the keyword) (5E16.0)**

Card 1 2 3 4 5 6 7 8 9 10

Variable	THIC1	THIC2	THIC3	THIC4	BETA or MCID
Type	F	F	F	F	F
Default	0.	0.	0.	0.	0.
Remarks	1				2

Optional Card (Required if OFFSET is specified after the keyword) (E16.0)

Card 1 2 3 4 5 6 7 8 9 10

Variable	OFFSET				
Type	F				
Default	0.				
Remarks	7				

Optional Card for scalar nodes (Required if DOF is specified after the keyword) (10I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable			NS1	NS2	NS3	NS4				
Type			I	I	I	I				
Default										
Remarks			8	8	8	8				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Element ID. Chose a unique number with respect to other elements.
PID	Part ID, see *PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
N4	Nodal point 4
N5-N8	Mid-side nodes for eight node shell
THIC1	Shell thickness at node 1
THIC2	Shell thickness at node 2
THIC3	Shell thickness at node 3
THIC4	Shell thickness at node 4
BETA	Orthotropic material base offset angle offset (see remarks 2 and 6 below). The angle is given in degrees. If blank the default is set to zero.
MCID	Material coordinate system ID. The angel BETA is automatically determined by the projection of the x-axis of the local coordinate system, MCID, onto the surface of the shell element.
OFFSET	The offset distance from the plane of the nodal points to the reference surface of the shell in the direction of the normal vector to the shell.
NS1	Scalar node 1, parameter NDOF on the *NODE_SCALAR is normally set to 2. If the thickness is constrained, set NDOF=0.
NS2	Scalar node 2
NS3	Scalar node 3
NS4	Scalar node 4

Remarks:

1. Default values in place of zero shell thicknesses are taken from the cross-section property definition of the PID, see *SECTION_SHELL.
2. BETA is defined only for orthotropic and anisotropic materials.

3. Counterclockwise node numbering determines the top surface, see Figure 14.7.
4. Stresses and strain output in the binary databases are by default given in the global coordinate system. Stress resultants are output in the local coordinate system for the shell element.
5. Interior angles must be less than 180 degrees.
6. To allow for an arbitrary orientation of the shell elements within the finite element mesh, each ply in the composite can have a unique material orientation angle which measures the offset from a reference system in the element. The reference system is determined by the AOPT and associated parameters in the *MAT input. The total offset for the i 'th integration point through the element thickness consists of two parts, the base offset and the layer offset. We write this as

$$\theta_i = \beta + \beta_i$$

where β is the base offset and β_i is the layer offset. The element BETA input here overrides the BETA on *MAT input. The β_i angles are input either by *PART_COMPOSITE, or by using the ICOMP option on *SECTION_SHELL. Figures 14.8 and 14.9 depict these angles.

7. The parameter OFFSET gives the offset from the nodal points of the shell to the reference surface. This option applies to most shell formulations excluding two-dimensional elements, membrane elements, and quadratic shell elements. The reference surface offset given by OFFSET is not taken into account in the contact subroutines.
8. The scalar nodes specified on the optional card refer to the scalar nodes defined by the user to hold additional degrees of freedom for shells with this capability. Scalar nodes are used with shell element type 25 and 26.

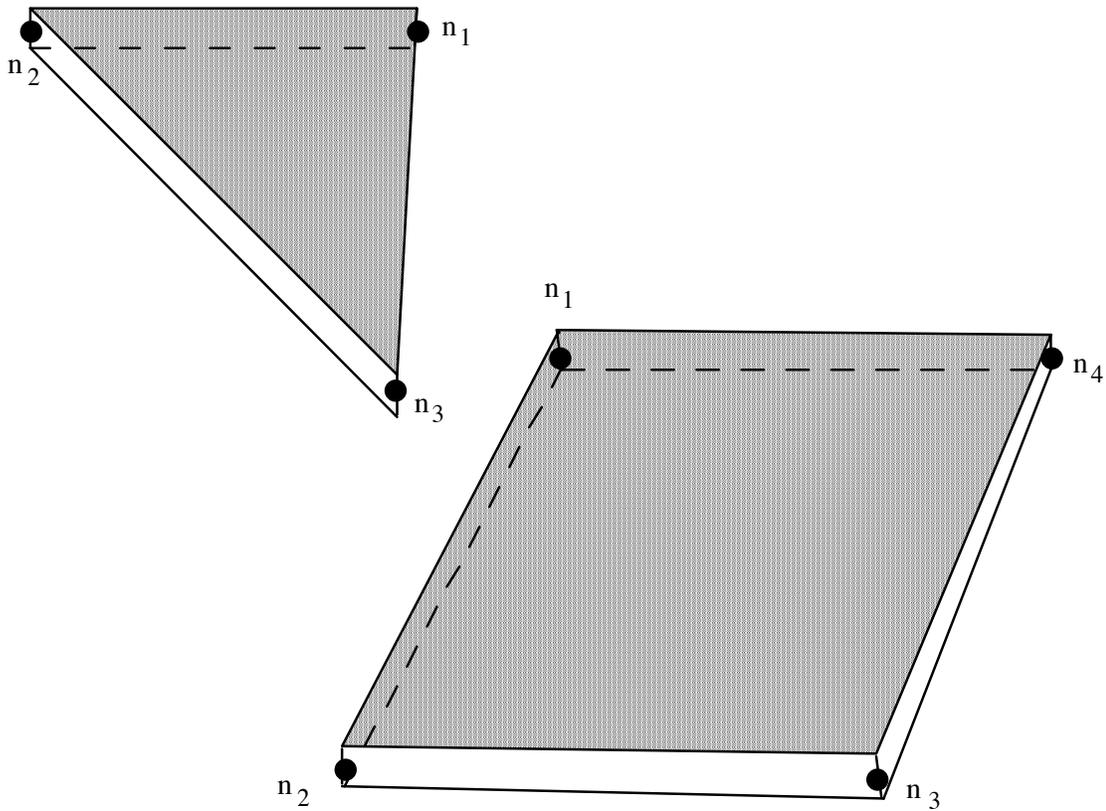


Figure 14.7. LS-DYNA shell elements. Counterclockwise node numbering determines the top surface.

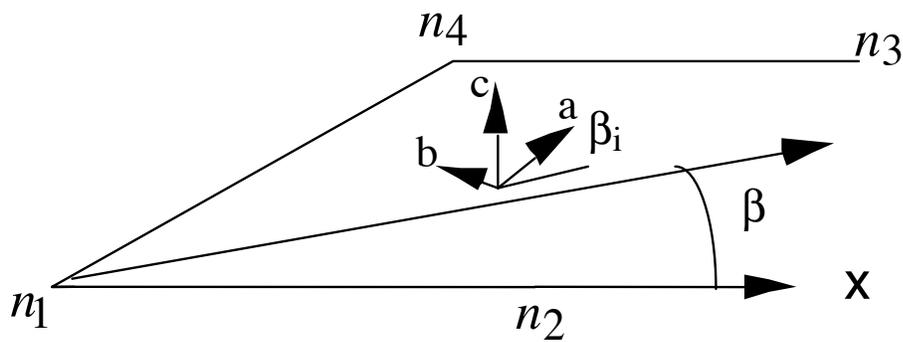


Figure 14.8. Orientation of material directions (shown relative to the 1-2 side as when AOPT=0 in *MAT).

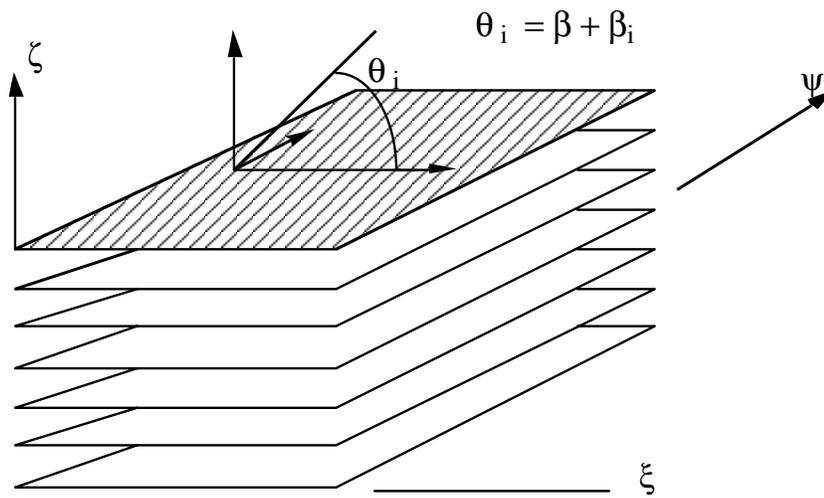


Figure 14.9. A multi-layer laminate can be defined. The angle β_i is defined for the i 'th lamina (integration point), see *SECTION_SHELL.

***ELEMENT_SHELL_SOURCE_SINK**

Purpose: Define a strip of shell elements of a single part ID to simulate a continuous forming operation. This option requires logical regular meshing of rectangular elements, which implies that the number of nodal points across the strip is constant along the length. Elements are created at the source and disappear at the sink. The advantage of this approach is that it is not necessary to define an enormous number of elements to simulate a continuous forming operation. Currently, only one source-sink definition is allowed. The boundary conditions at the source are discrete nodal point forces to keep the work piece in tension. At the sink, displacement boundary conditions are applied.

Card 1 2 3 4 5 6 7 8

Variable	NSSR	NSSK	PID					
Type	I	I	I					
Default	none	none	none					

VARIABLE**DESCRIPTION**

NSSR	Node set at source. Provide an ordered set of nodes between corner nodes, which include the corner nodes.
NSSK	Node set at sink. Provide an ordered set of nodes between corner nodes, which include the corner nodes.
PID	Part ID of work piece.

*ELEMENT

*ELEMENT_SOLID

*ELEMENT_SOLID_{OPTION}

Available options include:

<BLANK>

ORTHO

DOF

TET4TOTET10

Purpose: Define three-dimensional solid elements including 4 noded tetrahedrons and 8-noded hexahedrons. The type of solid element and its formulation is specified through the part ID (see *PART) and the section ID (see *SECTION_SOLID_OPTION). Also, a local coordinate system for orthotropic and anisotropic materials can be defined by using the ORTHO option. If extra degrees of freedom are needed, the DOF option should be used. The option TET4TOTET10 converts 4 node tetrahedrons to 10 node tetrahedrons. See remarks below.

Card Format (2I8)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID								
Type	I	I								
Default	None	none								
Remarks	2									

Card 2

Variable	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10
Type	I	I	I	I	I	I	I	I	I	I
Default	None									

Optional Cards (Required if ORTHO is specified after the keyword)

Optional card 1 1 2 3 4 5 6 7 8 9 10

Variable	A1	A2	A3		
Type	F	F	F		
Default	0.	0.	0.		
Remarks	3				

Optional card 2

Variable	D1	D2	D3		
Type	F	F	F		
Default	0.	0.	0.		
Remarks	3				

Optional Card (Required if DOF is specified after the keyword)

Card 2 1 2 3 4 5 6 7 8 9 10

Variable			NS1	NS2	NS3	NS4	NS5	NS6	NS7	NS8
Type			I	I	I	I	I	I	I	I
Default			none							
Remarks										

ELEMENT**ELEMENT_SOLID**

VARIABLE	DESCRIPTION
EID	Element ID. A unique number has to be chosen.
PID	Part ID, see *PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
.	.
.	.
N10	Nodal point 10
A1	x-component of local material direction a, or else rotation angle in degrees (see remark 4).
A2	y-component of local material direction a.
A3	z-component of local material direction a.
D1	x-component of vector in the plane of the material vectors a and b.
D2	y-component of vector in the plane of the material vectors a and b.
D3	z-component of vector in the plane of the material vectors a and b.
NS1	Scalar node 1
NS2	Scalar node 2
NS3	Scalar node 3
NS4	Scalar node 4
NS5	Scalar node 5
NS6	Scalar node 6
NS7	Scalar node 7
NS8	Scalar node 8

Remarks:

1. The option TET4TOTET10 automatically converts 4 node tetrahedron solids to 10 node quadratic tetrahedron solids. Additional mid-side nodes are created which are shared by all tetrahedron elements that contain the edge. The user node ID's for these generated nodes are offset after the largest user node ID defined in the input file. When defining the *SECTION_SOLID keyword, the element type must be specified as either 16 or 17 which are the 10-noded tetrahedrons in LS-DYNA.
2. Four, six, and eight node elements are depicted in Figure 14.10 where the ordering of the nodal points is shown. This ordering must be followed or code termination will occur during the initialization phase with a negative volume message. The input of nodes on the element cards for the tetrahedron and pentahedron elements is given by:

4-noded tetrahedron N1, N2, N3, N4, N4, N4, N4, 0, 0

6-noded pentahedron N1, N2, N3, N4, N5, N5, N6, N6, 0, 0

If hexahedrons are mixed with tetrahedrons and pentahedrons in the input under the same part ID, degenerate tetrahedrons and pentahedrons are used. One problem with degenerate elements is related to an uneven mass distribution (node 4 of the tetrahedron has five times the mass of nodes 1-3) which can make these elements somewhat unstable with the default time step size. By using the control flag under the keyword, *CONTROL_SOLID, automatic sorting can be invoked to treat the degenerate elements as type 10 and type 15 tetrahedron and pentahedron elements, respectively.

For elements with 4-8 nodes the card formats of LS-DYNA versions 940-970 are still valid. Card 2 is not defined in the older format.

Card Format (10I8)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I

3. For the orthotropic and anisotropic material models the local directions may be defined on the second card following the element connectivity definition. The local directions are then computed from the two vectors such that (see Figure 14.11):

$$\underline{c} = \underline{a} \times \underline{d} \text{ and } \underline{b} = \underline{c} \times \underline{a} .$$

These vectors are internally normalized within LS-DYNA. If the material model uses AOPT=3, the *a* and *b* axes will be rotated about the *c* axis by the BETA angle on the material card.

- 4. Stress output for solid elements is in the global coordinate system by default.
- 5. If vector **d** is input as a zero length vector, then A1 is interpreted as an offset rotation angle in degrees which describes a rotation about the **c**-axis of the **a-b-c** coordinate system that is defined by AOPT and associated parameters on the *MAT input. This angle overrides the offset angle defined by BETA on the *MAT input.
- 6. The scalar nodes specified on the optional card refer to the scalar nodes defined by the user to hold additional degrees of freedom for solids with this capability. This option is primarily to be used with user defined solids.

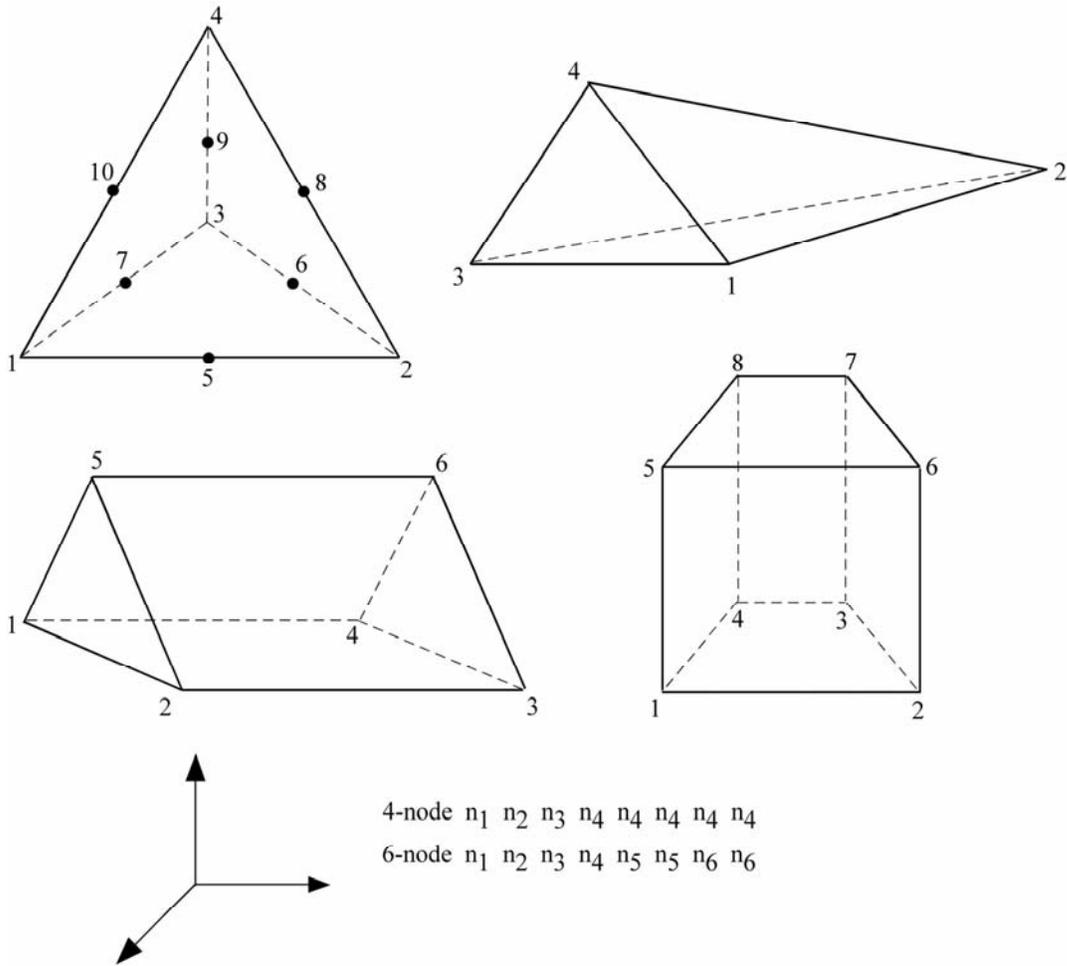


Figure 14.10. Four, six, and eight node solid elements. Nodes 1-4 are on the bottom surface.

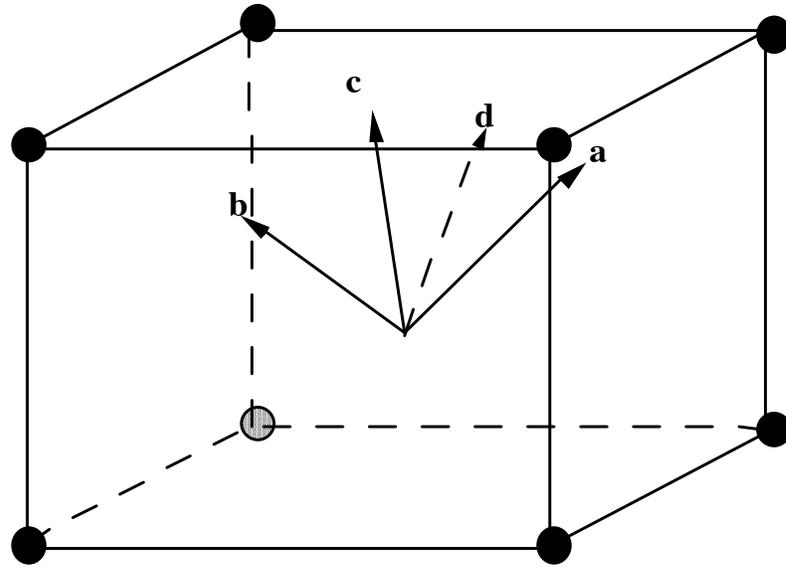


Figure 14.11. Two vectors **a** and **d** are defined and the triad is computed and stored.

*ELEMENT

*ELEMENT_SPH

*ELEMENT_SPH

Purpose: Define a lumped mass element assigned to a nodal point.

Card Format (2I8,E16.0)

Card 1 2 3 4 5 6 7 8 9 10

Variable	NID	PID	MASS							
Type	I	I	F							
Default	none	none	0.							
Remarks										

VARIABLE

DESCRIPTION

NID	Node ID and Element ID are the same for the SPH option.
PID	Part ID to which this node (element) belongs.
MASS	Mass value

ELEMENT_TRIM**ELEMENT*****ELEMENT_TRIM**

Purpose: Define a part subset to be trimmed by *DEFINE_CURVE_TRIM.

Card Format (8I10)

Card 1 1 2 3 4 5 6 7 8

Variable	PSID								
Type	I								
Default	none								

VARIABLE**DESCRIPTION**

PSID

Part set ID for trimming, see *SET_PART.

Remarks:

1. This keyword is used in combination with *DEFINE_CURVE_TRIM to trim the parts in PSID at time=0, i.e. before the simulation begins.

*ELEMENT

*ELEMENT_TSHELL

*ELEMENT_TSHELL

Purpose: Define an eight node thick shell element which is available with either fully reduced or selectively reduced integration rules. This plane stress element can be used as an alternative to the 4 node shell elements in cases where an 8-noded element is desired. Care must be taken in defining the element connectivity as N1 to N4 define the lower surface of the thick shell. The number of through-thickness integration points is defined by the user. The definition is completed by the *PART and *SECTION_TSHELL cards.

Card Format (10I8)

Card 1 2 3 4 5 6 7 8 9 10

Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I
Default	none									
Remarks			1							

VARIABLE

DESCRIPTION

EID	Element ID. Unique numbers have to be used.
PID	Part ID, see *PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
.	.
.	.
N8	Nodal point 8

Remarks:

1. The correct numbering of the nodes is essential for correct use. Nodes n_1 to n_4 define the lower surface, and nodes n_5 to n_8 define the upper surface. If one point integration is used (see *SECTION_TSHELL), the integration points then lie along the t -axis as depicted in Figure 14.12. Two by two selective reduced integration is also available. Extreme care must be used in defining the connectivity to insure proper orientation.
2. The stresses for this shell element are output in the global coordinate system.
3. To define a thick shell wedge element nodal pairs n_3 & n_4 and n_7 & n_8 are repeated. The ordering is then $n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8$, where nodes n_1, n_2, n_3 form the lower triangular face and nodes n_4, n_5, n_6 for the upper triangular face of the wedge.

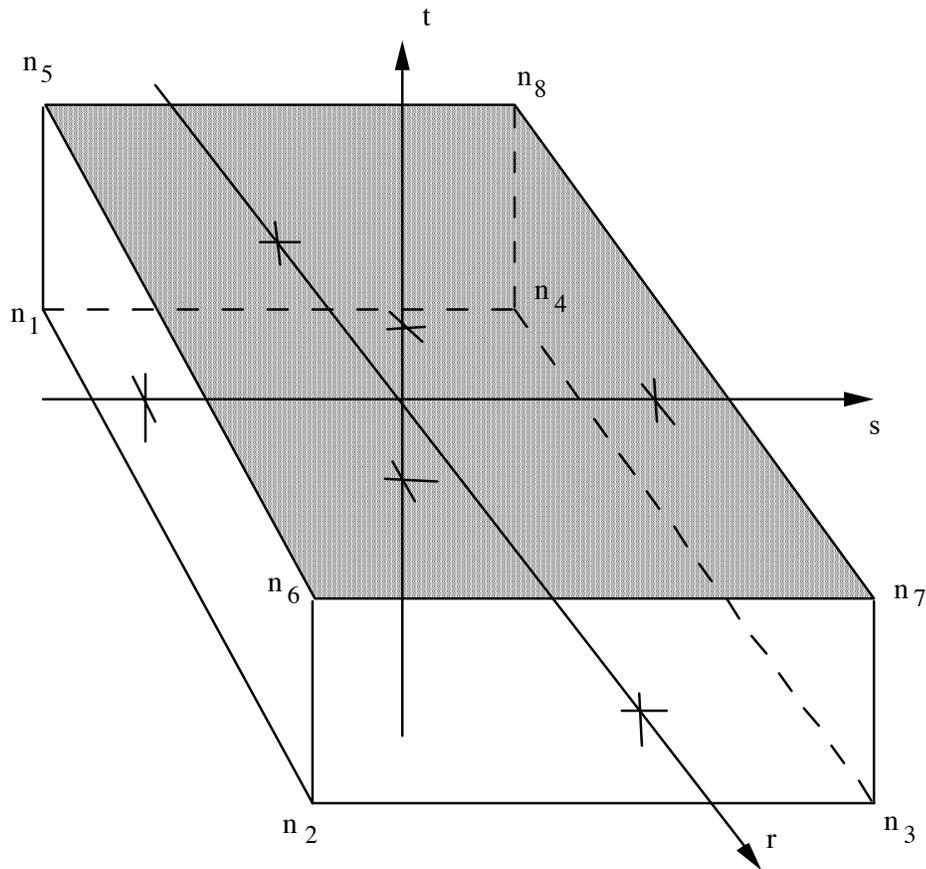


Figure 14.12. Solid 8-node Shell Element.

***ELEMENT**

***ELEMENT_TSHELL**

***EOS**

LS-DYNA has historically referenced equations of state by type identifiers. Below these identifiers are given with the corresponding keyword name in the order that they appear in the manual. The equations of state can be used with a subset of the materials that are available for solid elements. Type 15 is linked to the type 2 thick shell element and can be used to model engine gaskets.

- TYPE 1: *EOS_LINEAR_POLYNOMIAL**
- TYPE 2: *EOS_JWL**
- TYPE 3: *EOS_SACK_TUESDAY**
- TYPE 4: *EOS_GRUNEISEN**
- TYPE 5: *EOS_RATIO_OF_POLYNOMIALS**
- TYPE 6: *EOS_LINEAR_POLYNOMIAL_WITH_ENERGY_LEAK**
- TYPE 7: *EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE**
- TYPE 8: *EOS_TABULATED_COMPACTION**
- TYPE 9: *EOS_TABULATED**
- TYPE 10: *EOS_PROPELLANT_DEFLAGRATION**
- TYPE 11: *EOS_TENSOR_PORE_COLLAPSE**
- TYPE 12: *EOS_IDEAL_GAS**
- TYPE 14: *EOS_JWLB**
- TYPE 15: *EOS_GASKET**
- TYPE 21-30: *EOS_USER_DEFINED**

An additional option **_TITLE** may be appended to all the ***EOS** keywords. If this option is used then an additional line is read for each section in 80a format which can be used to describe the equation of state. At present LS-DYNA does not make use of the title. Inclusion of title simply gives greater clarity to input decks.

*EOS

DEFINITIONS & NOTES ON SOME COMMONLY USED PARAMETERS:

In order to prescribe the boundary and/or initial thermodynamic condition, manual computations are often necessary. Conventions or definitions must be established to simplify this process. Some basic variables are defined in the following. Since many of these variables have already been denoted by different symbols, the notations used here are unique in this section only! They are presented to only clarify their usage. A corresponding SI unit set is also presented as an example.

First consider a few volumetric parameters since they are a measure of compression (or expansion).

$$\text{Volume} = V \approx (m^3)$$

$$\text{Mass} = M \approx (Kg)$$

$$\text{Current specific volume (per mass)} = v = \frac{V}{M} = \frac{1}{\rho} \approx \left(\frac{m^3}{Kg} \right)$$

$$\text{Reference specific volume} = v_0 = \frac{V_0}{M} = \frac{1}{\rho_0} \approx \left(\frac{m^3}{Kg} \right)$$

$$\text{Relative volume} = v_r = \frac{V}{V_0} = \frac{(V/M)}{(V_0/M)} = \frac{v}{v_0} = \frac{\rho_0}{\rho}$$

$$\text{Current normalized volume increment} = \frac{dv}{v} = \frac{v-v_0}{v} = 1 - \frac{1}{v_r} = 1 - \frac{\rho}{\rho_0}$$

$$\text{A frequently used volumetric parameter is } \mu = \frac{1}{v_r} - 1 = \frac{v_0 - v}{v} = -\frac{dv}{v} = \frac{\rho}{\rho_0} - 1$$

$$\text{Sometimes another volumetric parameter is used: } \eta = \frac{v_0}{v} = \frac{\rho}{\rho_0}$$

$$\text{Thus the relation between } \mu \text{ and } \eta \text{ is } \mu = \frac{v_0 - v}{v} = \eta - 1$$

The following table summarizes these volumetric parameters.

VARIABLES	COMPRESSION	NO LOAD	EXPANSION
$v_r = \frac{v}{v_0} = \frac{\rho_0}{\rho}$	< 1	1	> 1
$\eta = \frac{1}{v_r} = \frac{v_0}{v} = \frac{\rho}{\rho_0}$	> 1	1	< 1
$\mu = \frac{1}{v_r} - 1 = \eta - 1$	> 0	0	< 0

V0 – INITIAL REALTIVE VOLUME

There are 3 definitions of density that must be distinguished from each other:

$\rho_0 = \rho_{ref}$ = Density at nominal/reference state, usually non-stress or non-deformed state.
 $\rho|_{t=0}$ = Density at $t = 0$
 ρ = Current density

Recalling the current relative volume, $v_r = \frac{\rho_0}{\rho} = \frac{v}{v_0}$, and the relative volume at time=0 is then

$v_{r0} = v_r|_{t=0} = \frac{\rho_0}{\rho|_{t=0}} = \frac{v|_{t=0}}{v_0}$. Generally, the V0 input parameter in an *EOS_ card, refers to this

v_{r0} . ρ_0 is generally the density defined in the *MAT_ card. Hence, if a material is mechanically compressed at $t=0$, V0, or v_{r0} , the initial relative volume, may be computed and input accordingly ($v_0 \neq V0$).

The “reference” state is a unique state with respect to which the material stress tensor is computed. Therefore v_0 is very critical in computing the pressure level in a material. Incorrect choice of v_0 would lead to incorrect pressure computed. In general, v_0 is chosen such that at zero compression or expansion, the material should be in equilibrium with its ambient surrounding. In many of the equations shown in the EOS section, μ is frequently used as a measure of compression (or expansion). However, the users must clearly distinguish between μ and v_{r0} .

E0 - INTERNAL ENERGY

Internal energy represents the thermal energy state (temperature dependent component) of a system. One definition for internal energy is

$$E = MC_v T \approx (\text{Joule})$$

Note that the capital “E” here is the absolute internal energy. It is not the same as that used in the subsequent *EOS keyword input, or some equations shown for each *EOS_ card. This internal energy is often defined with respect to a mass or volume unit.

Internal energy per unit mass (also called specific internal energy):

$$e = \frac{E}{M} = C_v T \approx \left(\frac{\text{Joule}}{\text{Kg}} \right)$$

Internal energy per unit current volume:

*EOS

$$e_v = \frac{M}{V} C_v T = \rho C_v T = \frac{C_v T}{v} \approx \left(\frac{\text{Joule}}{\text{m}^3} = \frac{\text{N}}{\text{m}^2} \right)$$

Internal energy per unit reference volume:

$$e_{v_0} = \frac{M}{V_0} C_v T = \rho_0 C_v T = \frac{C_v T}{v_0} \approx \left(\frac{\text{Joule}}{\text{m}^3} = \frac{\text{N}}{\text{m}^2} \right).$$

e_{v_0} typically refers to the capital “E” shown in some equations under this “EOS” section. Hence the initial “internal energy per unit reference volume”, E0, a keyword input parameter in the *EOS section can be computed from

$$e_{v_0}|_{t=0} = \rho_0 C_v T|_{t=0}$$

To convert from e_{v_0} to e_v , simply divide e_{v_0} by v_r

$$e_v = \rho C_v T = [\rho_0 C_v T] \frac{\rho}{\rho_0} = \frac{e_{v_0}}{v_r}$$

EQUATION OF STATE (EOS)

A thermodynamic state of a homogeneous material, not undergoing any chemical reactions or phase changes, may be defined by two state variables. This relation is generally called an equation of state. For example, a few possible forms relating pressure to two other state variables are

$$P = P(\rho, T) = P(v, e) = P(v_r, e_v) = P(\mu, e_{v_0})$$

The last equation form is frequently used to compute pressure. The EOS for solid phase materials is sometimes partitioned into 2 terms, a cold pressure and a thermal pressure

$$P = P_c(\mu) + P_T(\mu, e_{v_0})$$

$P_c(\mu)$ is the cold pressure hypothetically evaluated along a 0-degree-Kelvin isotherm. This is sometimes called a 0-K pressure-volume relation or cold compression curve. $P_T(\mu, e_{v_0})$ is the thermal pressure component that depends on both volumetric compression and thermal state of the material.

Different forms of the EOS describe different types of materials and how their volumetric compression (or expansion) behaviors. The coefficients for each EOS model come from data-fitting, phenomenological descriptions, or derivations based on classical thermodynamics, etc.

LINEAR COMPRESSION

In low pressure processes, pressure is not significantly affected by temperature. When volumetric compression is within an elastic linear deformation range, a linear bulk modulus may be used to relate volume changes to pressure changes. Recalling the definition of an isotropic bulk modulus is [Fung 1965] $\frac{\Delta v}{v} = -\frac{P}{K}$. This may be rewritten as $P = K \left[-\frac{\Delta v}{v} \right] = K\mu$. The bulk modulus, K , thus is equivalent to C_1 in *EOS_LINEAR_POLYNOMIAL when all other coefficients are zero. This is a simplest form of an EOS. To initialize a pressure for such a material, only v_{r_0} must be defined.

INITIAL CONDITION SETTING

In general, a thermodynamic state must be defined by two state variables. The need to specify v_{r_0} and/or $e_{v_0}|_{t=0}$ depends on the form of the EOS chosen. The user should review the equation term-by-term to establish what parameters to be initialized.

For many of the EOS available, pressure is specified (given), and the user must make an assumption on either $e_{v_0}|_{t=0}$ or v_{r_0} . Consider two possibilities (a) $T|_{t=0}$ is defined or assumed from which $e_{v_0}|_{t=0}$ may be computed, or (2) $\rho|_{t=0}$ is defined or assumed from which v_{r_0} may be obtained.

WHEN TO USE THE EOS

For small strains considerations, a total stress tensor may be partitioned into a deviatoric stress component and a mechanical pressure.

$$\sigma_{ij} = \sigma'_{ij} + \frac{\sigma_{kk}}{3} \delta_{ij} = \sigma'_{ij} - P \delta_{ij}$$

$$P = -\frac{\sigma_{kk}}{3} \Leftrightarrow \frac{\sigma_{kk}}{3} = -P$$

The pressure component may be written from the diagonal stress components.

Note that $\frac{\sigma_{kk}}{3} = \frac{[\sigma_{11} + \sigma_{22} + \sigma_{33}]}{3}$ is positive in tension while P is positive in compression.

Similarly the total strain tensor may be partitioned into a deviatoric strain component (volume-preserving deformation) and a volumetric deformation.

$$\epsilon_{ij} = \epsilon'_{ij} + \frac{\epsilon_{kk}}{3} \delta_{ij}$$

*EOS

where $\frac{\epsilon_{kk}}{3}$ is called the mean normal strain, and ϵ_{kk} is called the dilatation or volume strain (change in volume per unit initial volume)

$$\epsilon_{kk} = \frac{V - V_0}{V_0}$$

Roughly speaking, a typical convention may refer to the relation $\sigma'_{ij} = f(\epsilon'_{ij})$ as a “constitutive equation”, and $P = f(\mu, e_{v0})$ as an EOS. The use of an EOS may be omitted only when volumetric deformation is very small, and $|P| \ll |\sigma'_{ij}|$.

***EOS_LINEAR_POLYNOMIAL**

Purpose: Define coefficients for linear polynomial EOS and initialize the initial thermodynamic state of the material. This is done by defining E0 and V0 below.

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	C0	C1	C2	C3	C 4	C5	C6
Type	A8	F	F	F	F	F	F	F

Card 2

Variable	E0	V0						
Type	F	F						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
C0	The 0 th polynomial equation coefficient.
C1	The 1 st polynomial equation coefficient (when used by itself, this is the <u>elastic</u> bulk modulus, i.e. it cannot be used for deformation that is beyond the elastic regime).
...	...
C6	The 6 th polynomial equation coefficient.
E0	Initial internal energy per unit reference specific volume (see the beginning of the *EOS section).
V0	Initial relative volume (see the beginning of the *EOS section).

Remarks:

1. The linear polynomial equation of state is linear in internal energy. The pressure is given by:

$$P = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2)E.$$

where terms $C_2\mu^2$ and $C_6\mu^2$ are set to zero if $\mu < 0$, $\mu = \frac{\rho}{\rho_0} - 1$, and $\frac{\rho}{\rho_0}$ is the ratio of current density to reference density. ρ_0 is a nominal or reference density defined in the *MAT_NULL card.

The linear polynomial equation of state may be used to model gas with the gamma law equation of state. This may be achieved by setting:

$$C_0 = C_1 = C_2 = C_3 = C_6 = 0$$

and

$$C_4 = C_5 = \gamma - 1$$

where

$$\gamma = \frac{C_p}{C_v}$$

is the ratio of specific heats. Pressure for a perfect gas is then given by:

$$P = (\gamma - 1) \left\{ \frac{\rho}{\rho_0} E \right\} = (\gamma - 1) \left\{ \frac{e_{v0}}{v_r} \right\}$$

E has the unit of pressure (where $E = e_{v0}$ and $v_r = \rho_0 / \rho$)

2. When $C_0 \neq 0$, it does not necessarily mean that the initial pressure is zero, $P_0 \neq C_0$! The initial pressure depends the values of all the coefficients and on $\mu|_{t=0}$ and $E|_{t=0}$. The pressure in a material is computed from the whole equation above, $P = P(\mu, E)$. It is always preferable to initialize the initial condition based on $\mu|_{t=0}$ and $E|_{t=0}$. The use of $C_0 \neq 0$ must be done with caution as it may change the form and behavior of the material. The safest way is to use the whole EOS equation to manually check for the pressure value. For example, for ideal gas, it is wrong to define $C_4 = C_5 = \gamma - 1$ and $C_0 \neq 0$ at the same time.
3. V0 and E0 defined in this card must be the same as the time-zero ordinates for the 2 load curves defined in the *BOUNDARY_AMBIENT_EOS card, if it is used. This is so that they would both consistently define the same initial state for a material.

*EOS_JWL

This is Equation of state Form 2.

Card	1	2	3	4	5	6	7	8
Variable	EOSID	A	B	R1	R2	OMEG	E0	VO
Type	A8	F	F	F	F	F	F	F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
A	
B	
R1	
R2	
OMEG	
E0	
VO	Initial relative volume.

Remarks:

The JWL equation of state defines the pressure as

$$p = A \left(1 - \frac{\omega}{R_1 V} \right) e^{-R_1 V} + B \left(1 - \frac{\omega}{R_2 V} \right) e^{-R_2 V} + \frac{\omega E}{V},$$

and is usually used for detonation products of high explosives.

*EOS

*EOS_SACK_TUESDAY

*EOS_SACK_TUESDAY

This is Equation of state Form 3.

Card	1	2	3	4	5	6	7	8
Variable	EOSID	A1	A2	A3	B1	B2	E0	V0
Type	A8	F	F	F	F	F	F	F

VARIABLE

DESCRIPTION

EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
A1	
A2	
A3	
B1	
B2	
E0	Initial internal energy
V0	Initial relative volume

Remarks:

The Sack equation of state defines pressure as

$$p = \frac{A_3}{V^{A_1}} e^{-A_2 V} \left(1 - \frac{B_1}{V} \right) + \frac{B_2}{V} E$$

and is used for detonation products of high explosives.

***EOS_GRUNEISEN**

This is Equation of state Form 4.

Card 1 2 3 4 5 6 7 8

Variable	EOSID	C	S1	S2	S3	GAMAO	A	E0
Type	A8	F	F	F	F	F	F	F

Card 2

Variable	V0							
Type	F							

VARIABLE

DESCRIPTION

EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
C	
S1	
S2	
S3	
GAMAO	
A	
E0	Initial internal energy
V0	Initial relative volume

Remarks:

The Gruneisen equation of state with cubic shock velocity-particle velocity defines pressure for compressed materials as

$$p = \frac{\rho_0 C^2 \mu \left[1 + \left(1 - \frac{\gamma_0}{2} \right) \mu - \frac{a}{2} \mu^2 \right]}{\left[1 - (S_1 - 1) \mu - S_2 \frac{\mu^2}{\mu+1} - S_3 \frac{\mu^3}{(\mu+1)^2} \right]^2} + (\gamma_0 + a \mu) E.$$

and for expanded materials as

$$p = \rho_0 C^2 \mu + (\gamma_0 + a \mu) E.$$

where C is the intercept of the v_s - v_p curve; S_1 , S_2 , and S_3 are the coefficients of the slope of the v_s - v_p curve; γ_0 is the Gruneisen gamma; a is the first order volume correction to γ_0 ; and

$$\mu = \frac{\rho}{\rho_0} - 1.$$

***EOS_RATIO_OF_POLYNOMIALS**

This is Equation of state Form 5.

Card Format (I10) for card 1, **(4E20.0)** all following cards.

Card 1 1

Variable	EOSID
Type	A8

Card 2 1 2 3 4

Variable	A10	A11	A12	A13
Type	F	F	F	F

Card 3

Variable	A20	A21	A22	A23
Type	F	F	F	F

Card 4

Variable	A30	A31	A32	A33
Type	F	F	F	F

Card 5

Variable	A40	A41	A42	A43
Type	F	F	F	F

EOS**EOS_RATIO_OF_POLYNOMIALS**

Card 6 1 2 3 4

Variable	A50	A51	A52	A53
Type	F	F	F	F

Card 7

Variable	A60	A61	A62	A63
Type	F	F	F	F

Card 8

Variable	A70	A71	A72	A73
Type	F	F	F	F

Card 9 1 2 3 4

Variable	A14	A24		
Type	F	F		

Card 10

Variable	ALPH	BETA	E0	V0
Type	F	F	F	F

VARIABLE	DESCRIPTION
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
A10	
A11	
A12	
A13	
A20	
A21	
A22	
A23	
A30	
A31	
A32	
A33	
A40	
A41	
A42	
A43	
A50	
A51	
A52	
A53	
A60	
A61	
A62	
A63	
A70	

VARIABLE	DESCRIPTION
A71	
A72	
A73	
A14	
A24	
ALPHA	α
BETA	β
E0	Initial internal energy
V0	Initial relative volume

Remarks:

The ratio of polynomials equation of state defines the pressure as

$$p = \frac{F_1 + F_2 E + F_3 E^2 + F_4 E^3}{F_5 + F_6 E + F_7 E^2} (1 + \alpha \mu)$$

where

$$F_i = \sum_{j=0}^n A_{ij} \mu^j \quad n = 4 \text{ if } i < 3$$

$$\mu = \frac{\rho}{\rho_0} - 1 \quad n = 3 \text{ if } i \geq 3$$

In expanded elements F_1 is replaced by $F_1' = F_1 + \beta \mu^2$. By setting coefficient $A_{10} = 1.0$, the delta-phase pressure modeling for this material will be initiated. The code will reset it to 0.0 after setting flags.

***EOS_LINEAR_POLYNOMIAL_WITH_ENERGY_LEAK**

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	C0	C1	C2	C3	C4	C5	C6
Type	A8	F	F	F	F	F	F	F

Card 2

Variable	E0	V0	LCID					
Type	F	F	I					

VARIABLE	DESCRIPTION
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
C0	
C1	
C2	
C3	
C4	
C5	
C6	
E0	Initial internal energy
V0	Initial relative volume
LCID	Load curve ID defining the energy deposition rate

***EOS**

***EOS_LINEAR_POLYNOMIAL_WITH_ENERGY_LEAK**

Remarks:

This polynomial equation of state, linear in the internal energy per initial volume, E , is given by

$$p = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2)E$$

in which C_0 , C_1 , C_2 , C_3 , C_4 , C_5 , and C_6 are user defined constants and

$$\mu = \frac{1}{V} - 1.$$

where V is the relative volume. In expanded elements, we set the coefficients of μ^2 to zero, i.e.,

$$C_2 = C_6 = 0$$

Internal energy, E , is increased according to an energy deposition rate versus time curve whose ID is defined in the input.

***EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE**

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	A	B	XP1	XP2	FRER	G	R1
Type	A8	F	F	F	F	F	F	F

Card 2

Variable	R2	R3	R5	R6	FMXIG	FREQ	GROW1	EM
Type	F	F	F	F	F	F	F	F

Card 3

Variable	AR1	ES1	CVP	CVR	EETAL	CCRIT	ENQ	TMP0
Type	F	F	F	F	F	F	F	F

Card 4

Variable	GROW2	AR2	ES2	EN	FMXGR	FMNGR		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
A	
B	
XP1	

EOS**EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE**

VARIABLE	DESCRIPTION
XP2	
FRER	
G	
R1	
R2	
R3	
R5	
R6	
FMXIG	
FREQ	
GROW1	
EM	
AR1	
ES1	
CVP	Heat capacity of reaction products
CVR	Heat capacity of unreacted HE
EETAL	
CCRIT	
ENQ	Heat of reaction
TMP0	Initial temperature (°K)
GROW2	
AR2	
ES2	
EN	
FMXGR	
FMNGR	

Remarks:

Equation of State Form 7 is used to calculate the shock initiation (or failure to initiate) and detonation wave propagation of solid high explosives. It should be used instead of the ideal HE burn options whenever there is a question whether the HE will react, there is a finite time required for a shock wave to build up to detonation, and/or there is a finite thickness of the chemical reaction zone in a detonation wave. At relatively low initial pressures (<2-3 GPa), this equation of state should be used with material type 10 for accurate calculations of the unreacted HE behavior. At higher initial pressures, material type 9 can be used. A JWL equation of state defines the pressure in the unreacted explosive as

$$P_e = r_1 e^{-r_5 V_e} + r_2 e^{-r_6 V_e} + r_3 \frac{T_e}{V_e} \quad (r_3 = \omega_e c v r)$$

where V_e and T_e are the relative volume and temperature, respectively, of the unreacted explosive. Another JWL equation of state defines the pressure in the reaction products as

$$P_p = a e^{-x p 1 V_p} + b e^{-x p 2 V_p} + \frac{g T_p}{V_p} \quad (g = \omega_p c v p)$$

where V_p and T_p are the relative volume and temperature, respectively, of the reaction products. As the chemical reaction converts unreacted explosive to reaction products, these JWL equations of state are used to calculate the mixture of unreacted explosive and reaction products defined by the fraction reacted F ($F=0$ implies no reaction, $F=1$ implies complete reaction). The temperatures and pressures are assumed to be equal ($T_e = T_p, P_e = P_p$) and the relative volumes are additive, i.e.,

$$V = (1 - F) V_e + V_p$$

The chemical reaction rate for conversion of unreacted explosive to reaction products consists of three physically realistic terms: an ignition term in which a small amount of explosive reacts soon after the shock wave compresses it; a slow growth of reaction as this initial reaction spreads; and a rapid completion of reaction at high pressure and temperature. The form of the reaction rate equation is

$$\frac{\partial F}{\partial t} = freq (1 - F)^{f r e r} (V e^{-1} - 1 - c c r i t)^{e e t a l} \quad (\text{Ignition})$$

$$+ grow1 (1 - F)^{e s 1} F^{a r 1} p^{e m} \quad (\text{Growth})$$

$$+ grow2 (1 - F)^{e s 2} f^{a r 2} p^{e n} \quad (\text{Completion})$$

The ignition rate is set equal to zero when $F \geq f m x i g$, the growth rate is set equal to zero when $F \geq f m x g r$, and the completion rate is set equal to zero when $F \leq f m n g r$.

***EOS**

***EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE**

Details of the computational methods and many examples of one and two dimensional shock initiation and detonation wave calculation can be found in the references. Unfortunately, sufficient experimental data has been obtained for only two solid explosives to develop very reliable shock initiation models: PBX-9504 (and the related HMX-based explosives LX-14,LX-10,LX-04, etc.) and LX-17 (the insensitive TATB-based explosive). Reactive flow models have been developed for other explosives (TNT, PETN, Composition B, propellants, etc.) but are based on very limited experimental data.

History variables 5 and 8 are temperature and burn fraction, respectively. See NEIPH in *DATABASE_EXTENT_BINARY if these output variables are desired in the databases for post-processing.

***EOS_TABULATED_COMPACTION**

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	GAMA	E0	V0				
Type	A8	F	F	F				

Card Format (5E16.0)

Card 2 1 2 3 4 5

Variable	EV1	EV2	EV3	EV4	EV5
Type	F	F	F	F	F

Card 3

Variable	EV6	EV7	EV8	EV9	EV10
Type	F	F	F	F	F

Repeat Cards 2 and 3 for C_i , T_i , and K_i . A total of 9 cards must be defined.

VARIABLE	DESCRIPTION
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
$\epsilon V1, \epsilon V2, \dots, \epsilon VN$	Volumetric strain, $\ln V$
$C1, C2, \dots, CN$	
$T1, T2, \dots, TN$	
$K1, K2, \dots, KN$	Bulk unloading modulus
GAMA	γ
E0	Initial internal energy
V0	Initial relative volume

Remarks:

The tabulated compaction model is linear in internal energy. Pressure is defined by

$$p = C(\epsilon_v) + \gamma T(\epsilon_v)E$$

in the loading phase. The volumetric strain, ϵ_v is given by the natural logarithm of the relative volume V . Unloading occurs along the unloading bulk modulus to the pressure cutoff. Reloading always follows the unloading path to the point where unloading began, and continues on the loading path, see Figure 15.1. Up to 10 points and as few as 2 may be used when defining the tabulated functions. LS-DYNA will extrapolate to find the pressure if necessary.

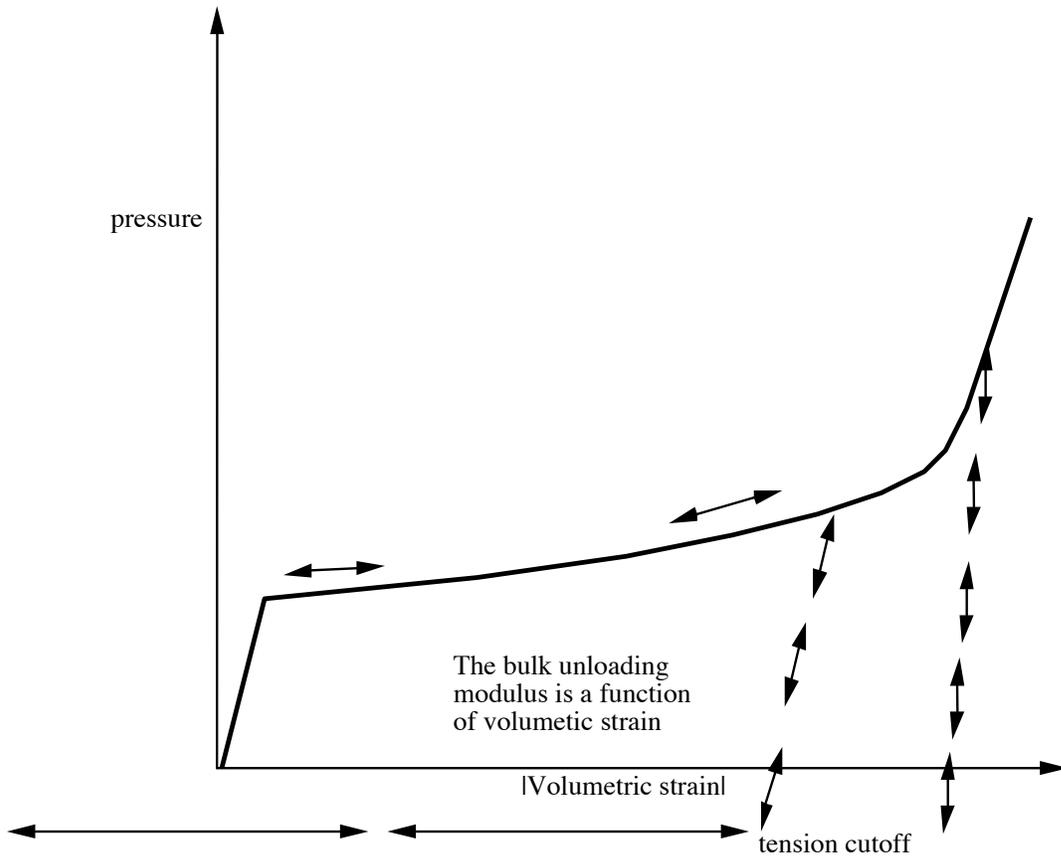


Figure 15.1. Pressure versus volumetric strain curve for Equation of state Form 8 with compaction. In the compacted states the bulk unloading modulus depends on the peak volumetric strain. Volumetric strain values should be input with correct sign (negative in compression) and in descending order. Pressure is positive in compression.

***EOS_TABULATED**

This is Equation of state Form 9.

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	GAMA	E0	V0	LCC	LCT		
Type	A8	F	F	F	I	I		

Define 6 additional cards if and only if LCC and LCT equal zero. Card Format (5E16.0)

Card 2 1 2 3 4 5

Variable	EV1	EV2	EV3	EV4	EV5
Type	F	F	F	F	F

Card 3

Variable	EV6	EV7	EV8	EV9	EV10
Type	F	F	F	F	F

Repeat Cards 2 and 3 to define C_i and T_i.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
GAMA	γ
E0	Initial internal energy
V0	Initial relative volume
LCC	Load curve defining tabulated function C. See equation in Remarks. The abscissa values of LCC and LCT must <i>increase</i> monotonically. The definition can extend into the tensile regime.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCT	Load curve defining tabulated function T. See equation in Remarks.
EV1, EV2, ...EVN	Volumetric strain, $\ln(V)$, where V is the relative volume. The first abscissa point, EV1, must be 0.0 or positive if the curve extends into the tensile regime with subsequent points <i>decreasing</i> monotonically.
C1,C2,..CN	Tabulated points for function C.
T1,T2,..TN	Tabulated points for function T.

Remarks:

The tabulated equation of state model is linear in internal energy. Pressure is defined by

$$P = C(\epsilon_v) + \gamma T(\epsilon_v) E$$

The volumetric strain, ϵ_v is given by the natural logarithm of the relative volume V . Up to 10 points and as few as 2 may be used when defining the tabulated functions. LS-DYNA will extrapolate to find the pressure if necessary.

***EOS_PROPELLANT_DEFLAGRATION**

This Equation of state (10) has been added to model airbag propellants.

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	A	B	XP1	XP2	FRER		
Type	A8	F	F	F	F	F		

Card 2

Variable	g	R1	R2	R3	R5			
Type	F	F	F	F	F			

Card 3

Variable	R6	FMXIG	FREQ	GROW1	EM			
Type	F	F	F	F	F			

Card 4

Variable	AR1	ES1	CVP	CVR	EETAL	CCRIT	ENQ	TMP0
Type	F	F	F	F	F			

Card 5

Variable	GROW2	AR2	ES2	EN	FMXGR	FMNGR		
Type	F	F	F	F	F	F		

EOS**EOS_PROPELLANT_DEFLAGRATION**

VARIABLE	DESCRIPTION
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
A	Product JWL coefficient
B	Product JWL coefficient
XP1	Product JWL coefficient
XP2	Product JWL coefficient
FRER	Unreacted Co-volume
G	Product wC_V
R1	Unreacted JWL coefficient
R2	Unreacted JWL coefficient
R3	Unreacted wC_V
R5	Unreacted JWL coefficient
R6	Unreacted JWL coefficient
FMXIG	Initial Fraction Reacted F_0
FREQ	Initial Pressure P_0
GROW1	First burn rate coefficient
EM	Pressure Exponent (1 st term)
AR1	Exponent on F (1 st term)
ES1	Exponent on (1-F) (1 st term)
CVP	Heat capacity products
CVR	Heat capacity unreacted
EETAL	Extra, not presently used
CCRIT	Product co-volume
ENQ	Heat of Reaction

VARIABLE	DESCRIPTION
TMP0	Initial Temperature (298°K)
GROW2	Second burn rate coefficient
AR2	Exponent on F (2 nd term)
ES2	Exponent on (1-F) (2 nd term)
EN	Pressure Exponent (2 nd term)
FMXGR	Maximum F for 1 st term
FMNGR	Minimum F for 2 nd term

Remarks:

A deflagration (burn rate) reactive flow model requires an unreacted solid equation of state, a reaction product equation of state, a reaction rate law and a mixture rule for the two (or more) species. The mixture rule for the standard ignition and growth model [Lee and Tarver 1980] assumes that both pressures and temperatures are completely equilibrated as the reaction proceeds. However, the mixture rule can be modified to allow no thermal conduction or partial heating of the solid by the reaction product gases. For this relatively slow process of airbag propellant burn, the thermal and pressure equilibrium assumptions are valid. The equations of state currently used in the burn model are the JWL, Gruneisen, the van der Waals co-volume, and the perfect gas law, but other equations of state can be easily implemented. In this propellant burn, the gaseous nitrogen produced by the burning sodium azide obeys the perfect gas law as it fills the airbag but may have to be modeled as a van der Waal's gas at the high pressures and temperatures produced in the propellant chamber. The chemical reaction rate law is pressure, particle geometry and surface area dependant, as are most high-pressure burn processes. When the temperature profile of the reacting system is well known, temperature dependent Arrhenius chemical kinetics can be used.

Since the airbag propellant composition and performance data are company private information, it is very difficult to obtain the required information for burn rate modeling. However, Imperial Chemical Industries (ICI) Corporation supplied pressure exponent, particle geometry, packing density, heat of reaction, and atmospheric pressure burn rate data which allowed us to develop the numerical model presented here for their $\text{NaN}_3 + \text{Fe}_2\text{O}_3$ driver airbag propellant. The deflagration model, its implementation, and the results for the ICI propellant are presented in [Hallquist, et.al., 1990].

The unreacted propellant and the reaction product equations of state are both of the form:

$$p = A e^{-R_1 V} + B e^{-R_2 V} + \frac{\omega C_v T}{V - d}$$

where p is pressure (in Mbars), V is the relative specific volume (inverse of relative density), ω is the Gruneisen coefficient, C_v is heat capacity (in Mbars -cc/cc°K), T is temperature in °K, d is

the co-volume, and A , B , R_1 and R_2 are constants. Setting $A=B=0$ yields the van der Waal's co-volume equation of state. The JWL equation of state is generally useful at pressures above several kilobars, while the van der Waal's is useful at pressures below that range and above the range for which the perfect gas law holds. Of course, setting $A=B=d=0$ yields the perfect gas law. If accurate values of ω and C_v plus the correct distribution between "cold" compression and internal energies are used, the calculated temperatures are very reasonable and thus can be used to check propellant performance.

The reaction rate used for the propellant deflagration process is of the form:

$$\frac{\partial F}{\partial t} = Z(1-F)^y F^x p^w + V(1-F)^u Frp^s$$

for $0 < F < F_{limit1}$ for $F_{limit2} < F < 1$

where F is the fraction reacted ($F = 0$ implies no reaction, $F = 1$ is complete reaction), t is time, and p is pressure (in Mbars), r,s,u,w,x,y , F_{limit1} and F_{limit2} are constants used to describe the pressure dependence and surface area dependence of the reaction rates. Two (or more) pressure dependant reaction rates are included in case the propellant is a mixture or exhibited a sharp change in reaction rate at some pressure or temperature. Burning surface area dependencies can be approximated using the $(1-F)^y F^x$ terms. Other forms of the reaction rate law, such as Arrhenius temperature dependent $e^{-E/RT}$ type rates, can be used, but these require very accurate temperatures calculations. Although the theoretical justification of pressure dependent burn rates at kilobar type pressures is not complete, a vast amount of experimental burn rate versus pressure data does demonstrate this effect and hydrodynamic calculations using pressure dependent burn accurately simulate such experiments.

The deflagration reactive flow model is activated by any pressure or particle velocity increase on one or more zone boundaries in the reactive material. Such an increase creates pressure in those zones and the decomposition begins. If the pressure is relieved, the reaction rate decreases and can go to zero. This feature is important for short duration, partial decomposition reactions. If the pressure is maintained, the fraction reacted eventually reaches one and the material is completely converted to product molecules. The deflagration front rates of advance through the propellant calculated by this model for several propellants are quite close to the experimentally observed burn rate versus pressure curves.

To obtain good agreement with experimental deflagration data, the model requires an accurate description of the unreacted propellant equation of state, either an analytical fit to experimental compression data or an estimated fit based on previous experience with similar materials. This is also true for the reaction products equation of state. The more experimental burn rate, pressure production and energy delivery data available, the better the form and constants in the reaction rate equation can be determined.

Therefore, the equations used in the burn subroutine for the pressure in the unreacted propellant

$$P_u = R1 \cdot e^{-R5 \cdot V_u} + R2 \cdot e^{-R6 \cdot V_u} + \frac{R3 \cdot T_u}{V_u - FRER}$$

where V_u and T_u are the relative volume and temperature respectively of the unreacted propellant. The relative density is obviously the inverse of the relative volume. The pressure P_p in the reaction products is given by:

$$P_p = A \cdot e^{-XP1 \cdot V_p} + B \cdot e^{-XP2 \cdot V_p} + \frac{G \cdot T_p}{V_p - CCRIT}$$

As the reaction proceeds, the unreacted and product pressures and temperatures are assumed to be equilibrated ($T_u = T_p = T$, $p = P_u = P_p$) and the relative volumes are additive:

$$V = (1 - F) \cdot V_u + F \cdot V_p$$

where V is the total relative volume. Other mixture assumptions can and have been used in different versions of DYNA2D/3D. The reaction rate law has the form:

$$\frac{\partial F}{\partial t} = \text{GROW1}(\text{P} + \text{FREQ})^{\text{EM}} (\text{F} + \text{FMXIG})^{\text{AR1}} (1 - \text{F} + \text{FMXIG})^{\text{ES1}} \\ + \text{GROW2}(\text{P} + \text{FREQ})^{\text{EN}} (\text{F} + \text{FMXIG})^{\text{AR2}} (1 - \text{F} + \text{FMXIG})^{\text{ES2}}$$

If F exceeds FMXGR , the GROW1 term is set equal to zero, and, if F is less than FMNGR , the GROW2 term is zero. Thus, two separate (or overlapping) burn rates can be used to describe the rate at which the propellant decomposes.

This equation of state subroutine is used together with a material model to describe the propellant. In the airbag propellant case, a null material model (type #10) can be used. Material type #10 is usually used for a solid propellant or explosive when the shear modulus and yield strength are defined. The propellant material is defined by the material model and the unreacted equation of state until the reaction begins. The calculated mixture states are used until the reaction is complete and then the reaction product equation of state is used. The heat of reaction, ENQ , is assumed to be a constant and the same at all values of F but more complex energy release laws could be implemented.

*EOS

*EOS_TENSOR_PORE_COLLAPSE

*EOS_TENSOR_PORE_COLLAPSE

This is Equation of state Form 11.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	NLD	NCR	MU1	MU2	IE0	EC0	
Type	A8	F	F	F	F	F	F	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
NLD	Virgin loading load curve ID
NCR	Completely crushed load curve ID
MU1	Excess Compression required before any pores can collapse
MU2	Excess Compression point where the Virgin Loading Curve and the Completely Crushed Curve intersect
IE0	Initial Internal Energy
EC0	Initial Excess Compression

Remarks:

The pore collapse model described in the TENSOR manual [23] is no longer valid and has been replaced by a much simpler method. This is due in part to the lack of experimental data required for the more complex model. It is desired to have a close approximation of the TENSOR model in the DYNA code to enable a quality link between them. The TENSOR model defines two curves, the virgin loading curve and the completely crushed curve as shown in Figure 15.2. It also defines the excess compression point required for pore collapse to begin (μ_1), and the excess compression point required to completely crush the material (μ_2). From this data and the maximum excess compression the material has attained (μ_{max}), the pressure for any excess compression (μ) can be determined.

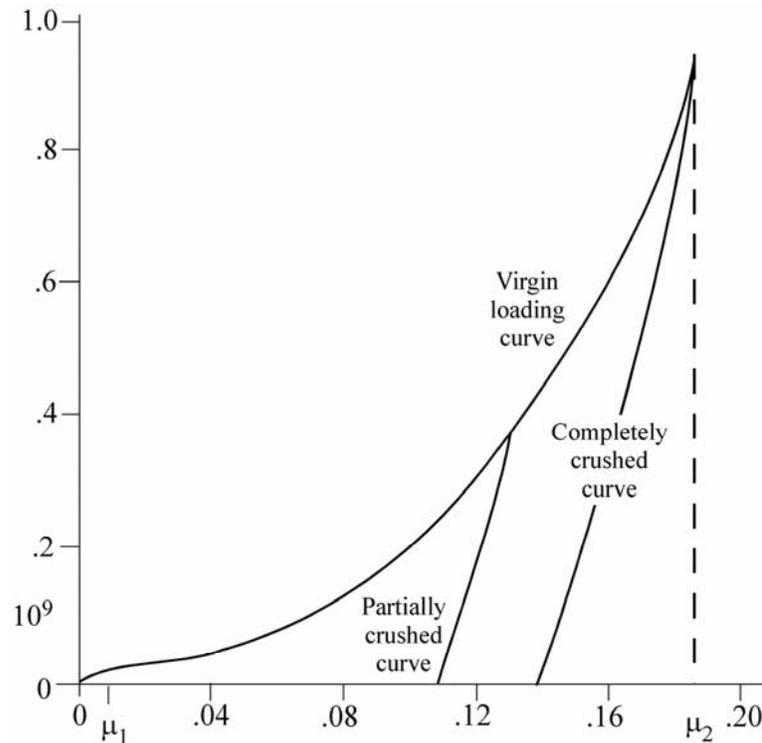


Figure 15.2. Pressure versus compaction curve.

Unloading occurs along the virgin loading curve until the excess compression surpasses μ_1 . After that, the unloading follows a path between the completely crushed curve and the virgin loading curve. Reloading will follow this curve back up to the virgin loading curve. Once the excess compression exceeds μ_2 , then all unloading will follow the completely crushed curve.

For unloading between μ_1 and μ_2 a partially crushed curve is determined by the relationship:

$$P_{pc}(\mu) = P_{cc} \left(\frac{(1 + \mu_B)(1 + \mu)}{1 + \mu_{\max}} - 1 \right).$$

where

$$\mu_B = P_{cc}^{-1}(P_{\max})$$

and the subscripts pc and cc refer to the partially crushed and completely crushed states, respectively. This is more readily understood in terms of the relative volume (V).

$$V = \frac{1}{1 + \mu}$$

$$P_{pc}(V) = P_{cc} \left(\frac{V_B}{V_{\min}} V \right)$$

This representation suggests that for a fixed $V_{\min} \left(= \frac{1}{\mu_{\max} + 1} \right)$ the partially crushed curve will separate linearly from the completely crushed curve as V increases to account for pore recovery in the material.

The bulk modulus K is determined to be the slope of the current curve times one plus the excess compression

$$K = \frac{\partial P}{\partial \mu} (1 + \mu)$$

The slope $\frac{\partial P}{\partial \mu}$ for the partially crushed curve is obtained by differentiation as:

$$\frac{\partial P}{\partial \mu} = \frac{\partial P_{cc} \left(\frac{(1 + \mu_B)(1 + \mu)}{(1 + \mu_{\max})} \right) (1 + \mu_B)}{\partial \mu (1 + \mu_{\max})}$$

Simplifying,

$$K = \frac{\partial P_{cc}(\mu_a)}{\partial \mu} (1 + \mu_a)$$

where

$$\mu_a = \frac{(1 + \mu_B)(1 + \mu)}{(1 + \mu_{\max})} - 1.$$

The bulk sound speed is determined from the slope of the completely crushed curve at the current pressure to avoid instabilities in the time step.

The virgin loading and completely crushed curves are modeled with monotonic cubic-splines. An optimized vector interpolation scheme is then used to evaluate the cubic-splines. The bulk modulus and sound speed are derived from a linear interpolation on the derivatives of the cubic-splines.

***EOS_IDEAL_GAS**

Purpose: This is equation of state form 12 for modeling ideal gas. It is an alternate approach to using *EOS_LINEAR_POLYNOMIAL with $C4 = C5 = (\gamma-1)$ to model ideal gas. This has a slightly improved energy accounting algorithm.

Card 1	1	2	3	4	5	6	7	8
Variable	EOSID	CV0	CP0	CL	CQ	T0	V0	
Type	A8	F	F	F	F	F	F	

VARIABLE**DESCRIPTION**

EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
CV0	Nominal constant-volume specific heat coefficient (at STP)
CP0	Nominal constant-pressure specific heat coefficient (at STP)
CL	Linear coefficient for the variations of Cv and Cp versus T
CQ	Quadratic coefficient for the variations of Cv and Cp versus T
T0	Initial temperature
V0	Initial relative volume (see the beginning of the *EOS section)

Remarks:

- The pressure in the ideal gas law is defined as

$$p = \rho (C_p - C_v) T$$

$$C_p = C_{p0} + C_L T + C_Q T^2$$

$$C_v = C_{v0} + C_L T + C_Q T^2$$

where C_p and C_v are the specific heat capacities at constant pressure and at constant volume, respectively. ρ is the density. The relative volume, V0 parameter in the input, is defined as

$$v_r = \frac{V}{V_0} = \frac{(V/M)}{(V_0/M)} = \frac{v}{v_0} = \frac{\rho_0}{\rho}$$

where ρ_0 is a nominal or reference density defined in the *MAT_NULL card. The initial pressure can then be manually computed as

$$P|_{t=0} = \rho|_{t=0} (C_P - C_V) T|_{t=0}$$

$$\rho|_{t=0} = \left\{ \frac{\rho_0}{v_r|_{t=0}} \right\}$$

$$P|_{t=0} = \left\{ \frac{\rho_0}{v_r|_{t=0}} \right\} (C_P - C_V) T|_{t=0}$$

The initial relative volume, $v_r|_{t=0}$ (V0), initial temperature, $T|_{t=0}$ (T0), and heat capacity information are defined in the *EOS_IDEAL_GAS input. Note that the “reference” density is typically a density at a non-stressed or nominal stress state. The initial pressure should always be checked manually against simulation result.

2. When dealing with Eulerian/ALE models, the ideal gas model is implemented to preserve the adiabatic state during advection. The adiabatic state is conserved on the expense of a perfect internal energy conservation.
3. The ideal gas model is good for low density gas only. Deviation from the ideal gas behavior may be indicated by the compressibility factor defined as

$$Z = \frac{Pv}{RT}$$

When Z deviates from 1, the gas behavior deviates from ideal.

4. V0 and T0 defined in this card must be the same as the time-zero ordinates for the 2 load curves defined in the *BOUNDARY_AMBIENT_EOS card, if it is used. This is so that they both would consistently define the same initial state for a material.

***EOS_JWL**

This is Equation of state Form 14. The JWL (Jones-Wilkens-Lee-Baker) equation of state, developed by Baker [1991] and further described by Baker and Orosz [1991], describes the high pressure regime produced by overdriven detonations while retaining the low pressure expansion behavior required for standard acceleration modeling. The derived form of the equation of state is based on the JWL form due to its computational robustness and asymptotic approach to an ideal gas at high expansions. Additional exponential terms and a variable Gruneisen parameter have been added to adequately describe the high-pressure region above the Chapman-Jouguet state.

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	A1	A2	A3	A4	A5		
Type	A8	F	F	F	F	F		

Card 2

Variable	R1	R2	R3	R4	R5			
Type	F	F	F	F	F			

Card 3

Variable	AL1	AL2	AL3	AL4	AL5			
Type	F	F	F	F	F			

Card 4

Variable	BL1	BL2	BL3	BL4	BL5			
Type	F	F	F	F	F			

Card 5 1 2 3 4 5 6 7 8

Variable	RL1	RL2	RL3	RL4	RL5			
Type	F	F	F	F	F			

Card 6

Variable	C	OMEGA	E	V0				
Type	I	F	F	F				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
A1	Equation of state coefficient, see below.
A2	Equation of state coefficient, see below.
A3	Equation of state coefficient, see below.
A4	Equation of state coefficient, see below.
A5	Equation of state coefficient, see below.
R1	Equation of state coefficient, see below.
R2	Equation of state coefficient, see below.
R3	Equation of state coefficient, see below.
R4	Equation of state coefficient, see below.
R5	Equation of state coefficient, see below.
AL1	$A_{\lambda 1}$, equation of state coefficient, see below.
AL2	$A_{\lambda 2}$, equation of state coefficient, see below.
AL3	$A_{\lambda 3}$, equation of state coefficient, see below.

VARIABLE	DESCRIPTION
AL4	$A_{\lambda 4}$, equation of state coefficient, see below.
AL5	$A_{\lambda 5}$, equation of state coefficient, see below.
BL1	$B_{\lambda 1}$, equation of state coefficient, see below.
BL2	$B_{\lambda 2}$, equation of state coefficient, see below.
BL3	$B_{\lambda 3}$, equation of state coefficient, see below.
BL4	$B_{\lambda 4}$, equation of state coefficient, see below.
BL5	$B_{\lambda 5}$, equation of state coefficient, see below.
RL1	$R_{\lambda 1}$, equation of state coefficient, see below.
RL2	$R_{\lambda 2}$, equation of state coefficient, see below.
RL3	$R_{\lambda 3}$, equation of state coefficient, see below.
RL4	$R_{\lambda 4}$, equation of state coefficient, see below.
RL5	$R_{\lambda 5}$, equation of state coefficient, see below.
C	Equation of state coefficient, see below.
OMEGA	Equation of state coefficient, see below.
E	Energy density per unit initial volume
V0	Initial relative volume.

Remarks:

The JWL equation-of-state defines the pressure as

$$p = \sum_{i=1}^5 A_i \left(1 - \frac{\lambda}{R_i V} \right) e^{-R_i V} + \frac{\lambda E}{V} + C \left(1 - \frac{\lambda}{\omega} \right) V^{-(\omega+1)}$$

$$\lambda = \sum_{i=1}^5 A_i (A_{\lambda i} V + B_{\lambda i}) e^{-R_{\lambda i} V} + \omega$$

where V is the relative volume, E is the energy per unit initial volume, and A_i , R_i , $A_{\lambda i}$, $B_{\lambda i}$, $R_{\lambda i}$, C , and ω are input constants defined above.

JWLB input constants for some common explosives as found in Baker and Stiel [1997] are given in the following table.

	TATB	LX-14	PETN	TNT	Octol 70/30
ρ_0 (g/cc)	1.800	1.821	1.765	1.631	1.803
E0 (Mbar)	.07040	.10205	.10910	.06656	.09590
DCJ (cm/ μ s)	.76794	.86619	.83041	.67174	.82994
PCJ (Mbar)	.23740	.31717	.29076	.18503	.29369
A1 (Mbar)	550.06	549.60	521.96	490.07	526.83
A2 (Mbar)	22.051	64.066	71.104	56.868	60.579
A3 (Mbar)	.42788	2.0972	4.4774	.82426	.91248
A4 (Mbar)	.28094	.88940	.97725	.00093	.00159
R1	16.688	34.636	44.169	40.713	52.106
R2	6.8050	8.2176	8.7877	9.6754	8.3998
R3	2.0737	20.401	25.072	2.4350	2.1339
R4	2.9754	2.0616	2.2251	.15564	.18592
C (Mbar)	.00776	.01251	.01570	.00710	.00968
ω	.27952	.38375	.32357	.30270	.39023
A λ 1	1423.9	18307.	12.257	.00000	.011929
B λ 1	14387.	1390.1	52.404	1098.0	18466.
R λ 1	19.780	19.309	43.932	15.614	20.029
A λ 2	5.0364	4.4882	8.6351	11.468	5.4192
B λ 2	-2.6332	-2.6181	-4.9176	-6.5011	-3.2394
R λ 2	1.7062	1.5076	2.1303	2.1593	1.5868

***EOS_GASKET**

This model works with solid elements and the thick shell using selective reduced 2 x 2 integration (ELFORM=2 on SECTION_TSHHELL) to model the response of gaskets. For the thick shell only, it is completely decoupled from the shell material, i.e., in the local coordinate system of the shell, this model defines the normal stress, σ_{zz} , and doesn't change any of the other stress components. The model is a reduction of the *MAT_GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM.

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	LCID1	LCID2	LCID3	LCID4			
Type	A8	I	I	I	I			

Card 2

Variable	UNLOAD	ET	DMPF	TFS	CFS	LOFFSET	IVS	
Type	F	F	F	F	F	F	F	

VARIABLE	DESCRIPTION
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
LCID1	Load curve for loading.
LCID2	Load curve for unloading.
LCID3	Load curve for damping as a function of volumetric strain rate.
LCID4	Load curve for scaling the damping as a function of the volumetric strain.
UNLOAD	Unloading option (See Volume II, Figure 119.1.): EQ.0.0: Loading and unloading follow loading curve EQ.1.0: Loading follows loading curve, unloading follows unloading curve. The unloading curve ID if undefined is taken as the loading curve. EQ.2.0: Loading follows loading curve, unloading follows unloading stiffness, KT or KR, to the unloading curve. The loading and unloading curves may only intersect at the origin of the axes.

VARIABLE	DESCRIPTION
	EQ.3.0: Quadratic unloading from peak displacement value to a permanent offset.
DMPF	Damping factor for stability. Values in the neighborhood of unity are recommended. The damping factor is properly scaled to eliminate time step size dependency. Also, it is active if and only if ET is defined.
TFS	Tensile failure strain.
CFS	Compressive failure strain.
OFFSET	Offset factor between 0 and 1.0 to determine permanent set upon unloading if the UNLOAD=3.0. The permanent sets in compression and tension are equal to the product of this offset value and the maximum compressive and tensile displacements, respectively.
IVS	Initial volume strain.

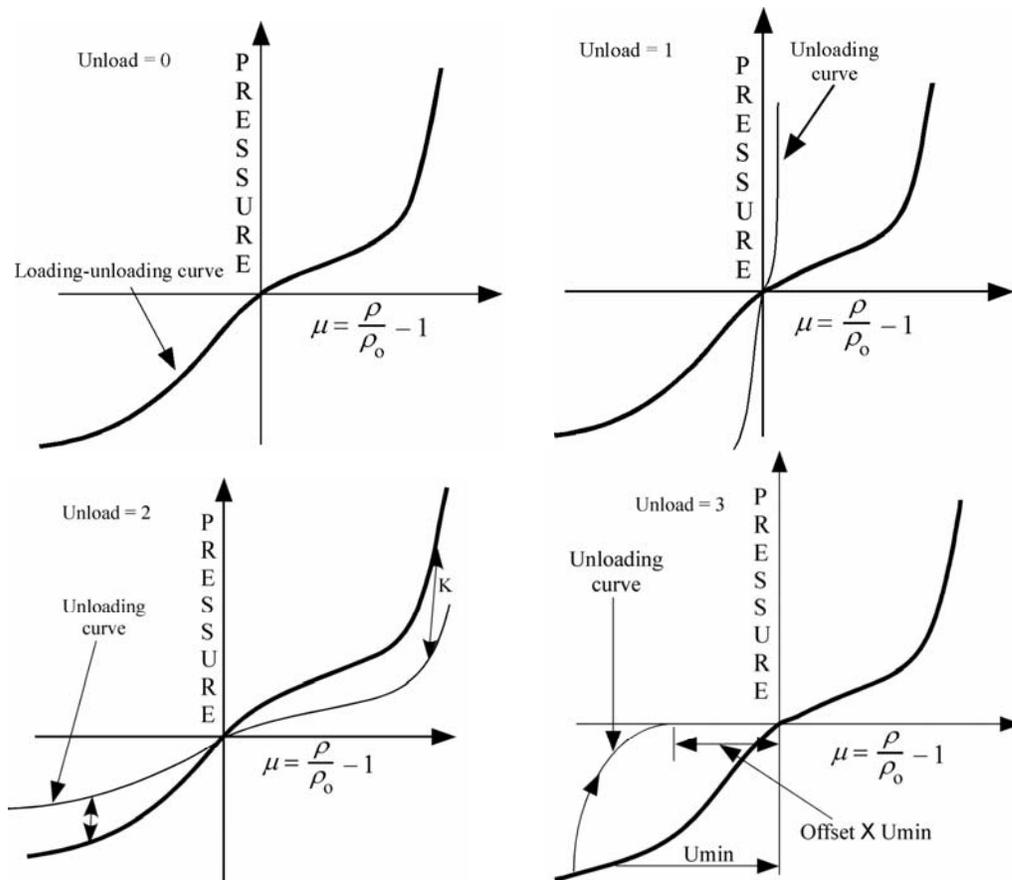


Figure 15.3. Load and Unloading behavior.

***EOS_USER_DEFINED**

These are equations of state 21-30. The user can supply his own subroutines. See also Appendix B. The keyword input has to be used for the user interface with data.

Card 1 1 2 3 4 5 6 7 8

Variable	EOSID	EOST	LMC	NHV	IVECT	EO	VO	BULK
Type	A8	I	I	I	I	F	F	F

Define LMC material parameters using 8 parameters per card.

Card 2 1 2 3 4 5 6 7 8

Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
EOSID	Equation of state ID, a unique number or label not exceeding 8 characters must be specified.
EOST	User equation of state type (21-30 inclusive). A number between 21 and 30 has to be chosen.
LMC	Length of material constant array which is equal to the number of material constants to be input. ($LMC \leq 48$)
NHV	Number of history variables to be stored, see Appendix D.
IVECT	Vectorization flag (on=1). A vectorized user subroutine must be supplied.
EO	Initial internal energy.
VO	Initial relative volume.
BULK	Bulk modulus. This value is used in the calculation of the contact surface stiffness.

***EOS**

***EOS_USER_DEFINED**

VARIABLE	DESCRIPTION
P1	First material parameter.
P2	Second material parameter.
P3	Third material parameter.
P4	Fourth material parameter.
.	.
.	.
.	.
PLMC	LMCth material parameter.

***HOURGLASS**

***HOURGLASS**

Purpose: Define hourglass and bulk viscosity properties which are referenced via HGID in the *PART command. Properties specified here, when invoked for a particular part, override those in *CONTROL_HOURGLASS and *CONTROL_BULK_VISCOSITY.

An additional option **_TITLE** may be appended to ***HOURGLASS** keywords. If this option is used then an additional line is read for each section in 80a format which can be used to describe the section. At present LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

Card 1 1 2 3 4 5 6 7 8

Variable	HGID	IHQ	QM	IBQ	Q1	Q2	QB/VDC	QW
Type	A8	I	F	I	F	F	F	F
Default	0	1	.10	0	1.5	0.06	QM/0.	QM
Remark		1,6	2,4,7	3	3	3	5	5

VARIABLE

DESCRIPTION

HGID Hourglass ID. A unique number or label not exceeding 8 characters must be specified. This ID is referenced by HGID in the *PART command.

IHQ Hourglass control type. For solid elements six options are available. For quadrilateral shell and membrane elements the hourglass control is based on the formulation of Belytschko and Tsay, i.e., options 1-3 are identical, and options 4-6 are identical:

- EQ.0: default=1 regardless of IHQ in *CONTROL_HOURGLASS,
- EQ.1: standard LS-DYNA viscous form,
- EQ.2: Flanagan-Belytschko viscous form,
- EQ.3: Flanagan-Belytschko viscous form with exact volume integration for solid elements,
- EQ.4: Flanagan-Belytschko stiffness form,

*HOURGLASS

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>EQ.5: Flanagan-Belytschko stiffness form with exact volume integration for solid elements.</p> <p>EQ.6: Belytschko-Bindeman [1993] assumed strain co-rotational stiffness form for 2D and 3D solid elements only. This form is available for explicit and IMPLICIT solution methods. In fact, type 6 or 7 is mandatory for the implicit options.</p> <p>EQ.7: Linear total strain form of type 6 hourglass control. This form is available for explicit and implicit solution method (See remark 6 below).</p> <p>EQ.8: Applicable to the type 16 fully integrated shell element. IHQ=8 activates the full projection warping stiffness for accurate solutions. A speed penalty of 25% is common for this option.</p> <p>EQ.9: Puso [2000] enhanced assumed strain stiffness form for 3D hexahedral elements. This form is available for explicit and implicit solution methods, hence it is an alternative to the Belytschko-Bindeman hourglass type 6 for implicit simulations.</p>
	<p>A discussion of the viscous and stiffness hourglass control for shell elements follows at the end of this section.</p>
QM	<p>Hourglass coefficient. Values of QM that exceed .15 may cause instabilities. The recommended default applies to all options except for IHQ=6. The stiffness forms, however, can stiffen the response especially if deformations are large and therefore should be used with care. For the shell and membrane elements QM is taken as the membrane hourglass coefficient, the bending as QB, and warping as QW. These coefficients can be specified independently, but generally, QM=QB=QW, is adequate. For type 6 solid element hourglass control, see remark 4 below. For hourglass type 9, see Remark 8.</p>
IBQ	<p>Bulk viscosity type (See Remark 3 below.): EQ.1: standard LS-DYNA.</p>
Q1	<p>Quadratic bulk viscosity coefficient.</p>
Q2	<p>Linear bulk viscosity coefficient.</p>
QB	<p>Hourglass coefficient for shell bending. The default: QB=QM. (See Remark 4).</p>
VDC	<p>Viscous damping coefficient for types 6 and 7 hourglass control.</p>
QW	<p>Hourglass coefficient for shell warping. The default: QB=QW.</p>

Remarks:

1. Viscous hourglass control is recommended for problems deforming with high velocities. Stiffness control is often preferable for lower velocities, especially if the number of time

steps are large. For solid elements the exact volume integration provides some advantage for highly distorted elements.

2. For automotive crash the stiffness form of the hourglass control with a coefficient of 0.05 is preferred by many users.
3. Bulk viscosity is necessary to propagate shock waves in solid materials and therefore applies only to solid elements. Generally, the default values are okay except in problems where pressures are very high, larger values may be desirable. In low density foams, it may be necessary to reduce the viscosity values since the viscous stress can be significant. It is not advisable to reduce it by more than an order of magnitude.
4. Type 6 hourglass control is for 2D and 3D solid elements only. Based on elastic constants and an assumed strain field, it produces accurate coarse mesh bending results for elastic material when QM=1.0. For plasticity models with a yield stress tangent modulus that is much smaller than the elastic modulus, a smaller value of QM (0.001 to 0.1) may produce better results. For any material, keep in mind that the stiffness is based on the elastic constants, so if the material softens, a QM value smaller than 1.0 may work better. For anisotropic materials, an average of the elastic constants is used. For fluids modeled with null material, type 6 hourglass control is viscous and is scaled to the viscosity coefficient of the material (see *MAT_NULL).
5. In part, the computational efficiency of the Belytschko-Lin-Tsay and the under integrated Hughes-Liu shell elements are derived from their use of one-point quadrature in the plane of the element. To suppress the hourglass deformation modes that accompany one-point quadrature, hourglass viscous or stiffness based stresses are added to the physical stresses at the local element level. The discussion of the hourglass control that follows pertains to all one point quadrilateral shell and membrane elements in LS-DYNA.

The hourglass shape vector τ_I is defined as

$$\tau_I = h_I - (h_J \hat{x}_{aJ}) B_{aI}$$

where, \hat{x}_{aJ} are the element coordinates in the local system at the Ith element node, B_{aI} is the strain displacement matrix, and hourglass basis vector is:

$$h = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

is the basis vector that generates the deformation mode that is neglected by one-point quadrature. In the above equations and the remainder of this subsection, the Greek subscripts have a range of 2, e.g., $\hat{x}_{aI} = (\hat{x}_{1I}, \hat{x}_{2I}) = (\hat{x}_I, \hat{y}_I)$.

The hourglass shape vector then operates on the generalized displacements to produce the generalized hourglass strain rates

*HOURGLASS

$$\dot{q}_\alpha^M = \tau_l \hat{v}_{\alpha l}$$

$$\dot{q}_\alpha^B = \tau_l \hat{\theta}_{\alpha l}$$

$$\dot{q}_3^W = \tau_l \hat{v}_{3l}$$

where the superscripts M, B, and W denote membrane, bending, and warping modes, respectively. The corresponding hourglass stress rates are then given by

$$\dot{Q}_\alpha^M = \frac{QM \cdot EtA}{8} B_{\beta 1} B_{\beta 1} \dot{q}_\alpha^M$$

$$\dot{Q}_\alpha^B = \frac{QB \cdot Et^3 A}{192} B_{\beta 1} B_{\beta 1} \dot{q}_\alpha^B$$

$$\dot{Q}_3^W = \frac{QW \cdot \kappa G t^3 A}{12} B_{\beta 1} B_{\beta 1} \dot{q}_3^W$$

where t is the shell thickness. The hourglass coefficients: QM, QB, and QW are generally assigned values between 0.05 and 0.10.

Finally, the hourglass stresses which are updated using the time step, Δt , from the stress rates in the usual way, i.e.,

$$Q^{n+1} = Q^n + \Delta t \dot{Q}$$

and the hourglass resultant forces are then

$$\hat{f}_{\alpha l}^H = \tau_l Q_\alpha^M$$

$$\hat{m}_{\alpha l}^H = \tau_l Q_\alpha^B$$

$$\hat{f}_{3l}^H = \tau_l Q_3^W$$

where the superscript H emphasizes that these are internal force contributions from the hourglass deformations.

6. IHG=7 is a linear total strain formulation of the Belytschko-Bindeman [1993] stiffness form for 2D and 3D solid elements. This linear form was developed for visco-elastic material and guarantees that an element will spring back to its initial shape regardless of the severity of deformation.
7. The default value for QM is 0.1 unless superseded by a nonzero value of QH in *CONTROL_HOURGLASS. A nonzero value of QM supersedes QH.

8. Hourglass type 9 is available for hexahedral elements and is based on physical stabilization using an enhanced assumed strain method. In performance it is similar to the Belytschko-Bindeman hourglass formulation (type 6) but gives more accurate results for distorted meshes, e.g., for skewed elements. If $QH=1.0$, it produces accurate coarse bending results for elastic materials. The hourglass stiffness is by default based on elastic properties, hence the QH parameter should be reduced to about 0.1 for plastic materials in order not to stiffen the structure during plastic deformation. For materials 3, 18 and 24 there is the option to use a negative value of QH . With this option, the hourglass stiffness is based on the current material properties, i.e., the plastic tangent modulus, and scaled by $|QH|$.

***HOURGLASS**

***INCLUDE**

Purpose: The keyword ***INCLUDE** provides a means of reading independent input files containing model data. The file contents are placed directly at the location of the ***INCLUDE** line.

***INCLUDE_{OPTION}**

***INCLUDE_COMPENSATION_OPTION**

***INCLUDE_{OPTION}**

Available options include:

<BLANK>

BINARY

NASTRAN

PATH

STAMPED_PART_{OPTION1}_{OPTION2}_{OPTION3}

OPTION1: **_SET**

OPTION2: **_MATRIX**

OPTION3: **_INVERSE**

STAMPED_SET

TRANSFORM

TRANSFORM_BINARY

The **BINARY** and **TRANSFORM_BINARY** options specify that the initial stress file, DYNAIN, is written in a binary format. See the keyword ***INTERFACE_SPRINGBACK**.

The **PATH** option defines a directory in which to look for the include files. The program always searches the local directory first. If an include file is not found and the filename has no path, the program will search for it in all the directories defined by ***INCLUDE_PATH**.

The **STAMPED_PART** option allows the plastic strain and thickness distribution of the stamping simulation to be mapped onto a part in the crash model.

1. When option 1, **_SET** is used, the PID will be part set ID. All the parts included in this set will be considered in this mapping.
2. When option 2, **_MATRIX** is used, translation matrix will be read directly and the orientation nodes will be ignored.
3. When option 3, **_INVERSE** (must be used with **_MATRIX**) is used, the matrix will be reversed first.

When **STAMPED_SET** is used, the target is a part set ID. Between the stamped part and the crash part, note the following points:

1. The outer boundaries of the parts do not need to match since only the regions of the crash part which overlap the stamped part are initialized.
2. Arbitrary mesh patterns are assumed.
3. Element formulations can change.
4. Three nodes on each part are used to reorient the stamped part for the mapping of the strain and thickness distributions. After reorientation, the three nodes on each part should approximately coincide.
5. The number of in plane integrations points can change.

- 6. The number of through thickness integration points can change. Full interpolation is used.
- 7. The node and element ID's between the stamped part and the crash part do not need to be unique.

The TRANSFORM option allows for node, element, and set ID's to be offset and for coordinates and constitutive parameters to be transformed and scaled.

The card is required.

Card 1 1

Variable	FILENAME
Type	C

If the NASTRAN option is active then define the following input line.

NASTRAN option

Card 2 1 2 3 4 5 6 7 8

Variable	BEAMDF	SHELLDF	SOLIDDF					
Type	I	I	I					
Default	2	21	18					

If the STAMPED_PART option is active then define the following input.

STAMPED_PART option

Card 2 1 2 3 4 5 6 7 8

Variable	PID	THICK	PSTRN	STRAIN	STRESS	INCOUT		RMAX
Type	I	I	I	I	I	I		F
Default	none	0	0	0	0	0		10.0

*INCLUDE

*INCLUDE_{OPTION}

Card 3 1 2 3 4 5 6 7 8

Variable	N1S	N2S	N3S	N1C	N2C	N3C	TENSOR	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	
Remarks	2	2	2	2	2	2	4	

Optional Cards

Card 4 1 2 3 4 5 6 7 8

Variable	ISYM	IAFTER						
Type	I	I						

Card 5

Variable	X01	Y01	Z01					
Type	F	F	F					

Card 6

Variable	X02	Y02	Z02	X03	Y03	Z03		
Type	F	F	F	F	F	F		

If the TRANSFORM option is active then define the following input

TRANSFORM option

Card 2 1 2 3 4 5 6 7 8

Variable	IDNOFF	IDEOFF	IDPOFF	IDMOFF	IDSOFF	IDFOFF	IDDOFF	
Type	I	I	I	I	I	I	I	

Card 3

Variable	IDROFF							
Type	I							

Card 4

Variable	FCTMAS	FCTTIM	FCTLEN	FCTTEM	INCOUT1			
Type	F	F	F	A	I			

Card 5

Variable	TRANID							
Type	I							
Default	0							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
FILENAME	File name of file to be included in this keyword file, 80 characters maximum. If the STAMPED_PART option is active, this is the DYNAIN file containing the results from metal stamping.
BEAMDF	LS-DYNA beam element type. Defaults to type 2.
SHELLDF	LS-DYNA shell element type. Defaults to type 21.

VARIABLE	DESCRIPTION
SOLIDDF	LS-DYNA solid element type. Defaults to type 18.
PID	Part ID of crash part for remapping.
THICK	Thickness remap: EQ.0: map thickness EQ.1: do not map thickness EQ.2: average value inside a circle defined by RMAX
PSTRN	Plastic strain remap: EQ.0: map plastic strain EQ.1: do not plastic strain EQ.2: average value inside a circle defined by RMAX
STRAIN	Strain remap: EQ.0: map strains EQ.1: do not map strains
STRESS	Stress tensor remap: EQ.0: map stress tensor EQ.1: do not map stress tensor
TENSOR	Tensor remap: EQ.0: map tensor data from history variables. (See Remark 4.) EQ.1: do not map tensor data from history variables
INCOUT	EQ.1: to save the mapped data to a file called dyna.inc, which contains the mapped data for the part that is being mapped. This option is useful to do mapping using INCLUDE_STAMPED_PART and then save the mapped data for future use. When INCOUT is set to 2, the output file is in dynain format and the file name is dynain_xx (xx is the part or part set id); and when INCOUT is set to 3, the output file is in NASTRAN format, and the file name is: nastran_xx. EQ.2: to save the mapped data for the specified part (PID) to a file called dynain_PID. EQ.3: to save the mapped data for the specified part (PID) to a file called nastran_PID (in nastran format)
RMAX	Search radius
N1S	First of 3 nodes needed to reorient the stamped part.
N2S	Second of 3 nodes needed to reorient the stamped part.
N3S	Third of 3 nodes needed to reorient the stamped part.
N1C	First of 3 nodes needed to reorient the crash model part.

VARIABLE	DESCRIPTION
N2C	Second of 3 nodes needed to reorient the crash model part.
N3C	Third of 3 nodes needed to reorient the crash model part.
ISYM	Symmetric switch EQ.0: no symmetric mapping EQ.1: yz plane symmetric mapping EQ.2: zx plane symmetric mapping EQ.3: zx and yz planes symmetric mapping EQ.4: user defined symmetric plane mapping
IAFTER	Mirroring sequence switch EQ.0: generate a symmetric part before transformation EQ.1: generate a symmetric part after transformation
X01, Y01, Z01	First point in the symmetric plane (required if ISYM.NE.0)
X02, Y02, Z02	Second point in the symmetric plane
X03, Y03, Z03	Third point in the symmetric plane
IDNOFF	Offset to node ID.
IDEOFF	Offset to element ID.
IDPOFF	Offset to part ID, nodal rigid body ID, and constrained nodal set ID.
IDMOFF	Offset to material ID and equation of state ID.
IDSOFF	Offset to set ID.
IDFOFF	Offset to function ID or table ID.
IDDOFF	Offset to any ID defined through DEFINE except the FUNCTION option.
IDROFF	Offset to section ID and hourglass ID.
FCTMAS	Mass transformation factor. For example, FCTMAS=1000. When the original mass units are in tons and the new unit is kg.
FCTTIM	Time transformation factor. For example, FCTTIM=.001 when the original time units are in milliseconds and the new time unit is seconds.
FCTLEN	Length transformation factor.
FCTTEM	Temperature transformation factor consisting of a four character flag: FtoC (Fahrenheit to Centigrade), CtoF, FtoK, KtoF, KtoC, and CtoK.

*INCLUDE

*INCLUDE_{OPTION}

VARIABLE	DESCRIPTION
INCOUT1	Set to 1 for the creation of a file, DYNA.INC, which contains the transformed data. The data in this file can be used in future include files and should be checked to ensure that all the data was transformed correctly.
TRANID	Transformation ID, if 0 no transformation will be applied. See the input DEFINE_TRANSFORMATION.

Remarks:

1. To make the input file easy to maintain, this keyword allows the input file to be split into subfiles. Each subfile can again be split into sub-subfiles and so on. This option is beneficial when the input data deck is very large. Consider the following example:

```
*TITLE
full car model
*INCLUDE
carfront.k
*INCLUDE
carback.k
*INCLUDE
occupantcompartment.k
*INCLUDE
dummy.k
*INCLUDE
bag.k
*CONTACT
...
*END
```

Note that the command *END terminates the include file.

The carfront.k file can again be subdivided into rightrail.k, leftrail.k, battery.k, wheelhouse.k, shotgun.k, etc.. Each *.k file can include nodes, elements, boundary conditions, initial conditions, and so on.

```
*INCLUDE
rightrail.k
*INCLUDE
leftrail.k
*INCLUDE
battery.k
*INCLUDE
wheelhouse.k
*INCLUDE
shotgun.k
...
...
*END
```

2. When defining ***INCLUDE_STAMPED_PART** the target mesh must be read in before the include stamped part.

n1s, n2s, n3s, n1c, n2c, n3c are used for transforming the stamped part to the crashed part, such that it is in the same position as the crashed part. If the stamped part is in the same position as the crashed part then n1s, n2s, n3s, n4s, n1c, n2c, n3c can all be set to 0. Note: If these 6 nodes are input as 0, LS-DYNA will not transform the stamped part.

When symmetric mapping is used (**ISYM** is not zero), the three points should not be in one line.

If **ISYM** = 0, 1, 2, or 3, only the first point (X01, Y01, Z01) is needed

If **ISYM** = 4, all the three points are needed

3. All filenames and paths are limited to a total of 240 characters in length, and to a limit of 80 characters per line. To continue a filename or path over more than one line use '+' (note space before plus sign) on the end of the line.
4. Certain material models (notably Material 190) have tensor data stored within the history variables. Within material subroutines this data is typically stored in element local coordinate systems. In order to properly map this information between models it is necessary to have the tensor data present on the ***INITIAL_STRESS_SHELL** card and have it stored in global coordinates. During mapping the data is then converted into the local coordinate system of the crash mesh. This data can be dumped into the dynain file that is created at termination time if the parameter **FTENSOR** is set to 0 on the ***INTERFACE_SPRINGBACK_DYNA3D** card. Currently, the only material model that supports mapping of element history tensor data is Material 190.

***INCLUDE**

***INCLUDE_COMPENSATION**

***INCLUDE_COMPENSATION_OPTION**

Purpose: To include geometry information for springback compensation. This keyword must be used with *INTERFACE_COMPENSATION_NEW.

Options available include:

BLANK_BEFORE_SPRINGBACK

BLANK_AFTER_SPRINGBACK

DESIRED_BLANK_SHAPE

COMPENSATED_SHAPE

CURRENT_TOOLS

***INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK**

Card 1

1

Variable	FILENAME
Type	C
Default	./blank0.tmp

***INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK**

Card 2

1

Variable	FILENAME
Type	C
Default	./spbk.tmp

INCLUDE_COMPENSATION**INCLUDE*****INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE**

Card 3

1

Variable	FILENAME
Type	C
Default	./reference0.dat

INCLUDE_COMPENSATION_COMPENSATED_SHAPE

Card 4

1

Variable	FILENAME
Type	C
Default	./reference1.dat

***INCLUDE_COMPENSATION_CURRENT_TOOLS**

Card 5

1

Variable	FILENAME
Type	C
Default	./rigid.tmp

Remarks:

When the option BLANK_BEFORE_SPRINGBACK is used, the included file is the 'dynain' file just before springback prediction. For the first iteration, it might be the same as reference0.dat. It can also be the mesh after coarsening and before springback.

When the option BLANK_AFTER_SPRINBACK is used, the included file is the 'dynain' file just after springback prediction.

When the option DESIRED_BLANK_SHAPE is used, the included file is the 'dynain' file after trimming in the first iteration.

When the option COMPENSATED_SHAPE is used, the included file is the 'dynain' file. For the first iteration, it is the same as reference0.dat; and for the following iterations, this file is obtained from the 'disp.tmp' which is generated as an output file during the previous compensation iteration.

When the option CURRENT_TOOLS is used, the included file is the file containing the tool mesh. The tool mesh after each forming simulation. The draw bead nodes have to be included in this file so that they will be modified with the rigid tools. If the file is named rigid0.tmp the elements of the tools get refined along the outline of the part.

***INITIAL**

The keyword ***INITIAL** provides a way of initializing velocities and detonation points. The keyword control cards in this section are defined in alphabetical order:

- *INITIAL_DETONATION**
- *INITIAL_FOAM_REFERENCE_GEOMETRY**
- *INITIAL_GAS_MIXTURE**
- *INITIAL_MOMENTUM**
- *INITIAL_STRAIN_SHELL**
- *INITIAL_STRAIN_SOLID**
- *INITIAL_STRESS_BEAM**
- *INITIAL_STRESS_DEPTH**
- *INITIAL_STRESS_SECTION**
- *INITIAL_STRESS_SHELL_{OPTION}**
- *INITIAL_STRESS_SOLID**
- *INITIAL_STRESS_TSHELL**
- *INITIAL_TEMPERATURE_OPTION**
- *INITIAL_VEHICLE_KINEMATICS**

Two mutually exclusive methods are available for initial velocity generation:

- *INITIAL_VELOCITY**
- *INITIAL_VELOCITY_NODE**
- *INITIAL_VELOCITY_RIGID_BODY**

and:

- *INITIAL_VELOCITY_GENERATION**

The latter is convenient for specifying initial rotational velocities about arbitrary axes. These methods for velocity generation must not be mixed in a single input deck.

- *INITIAL_VOID_OPTION**
- *INITIAL_VOLUME_FRACTION**
- *INITIAL_VOLUME_FRACTION_GEOMETRY**

*INITIAL

*INITIAL_DETONATION

*INITIAL_DETONATION

Purpose: Define points to initiate the location of high explosive detonations in part ID's which use the material (type 8) *MAT_HIGH_EXPLOSIVE_BURN. Also see *CONTROL_EXPLOSIVE_SHADOW.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	X	Y	Z	LT			
Type	I	F	F	F	F			
Default	all HE	0.	0.	0.	0,			

Optional card required if and only if PID=-1.

Card 2 1 2 3 4 5 6 7 8

Variable	PEAK	DECAY	XS	YS	ZS	NID		
Type	F	F	F	F	F	I		
Remark	1	1						

VARIABLE

DESCRIPTION

PID	Part ID of high explosive material to be lit, see *PART. However, two other options are available: EQ.-1: an acoustic boundary, also, *BOUNDARY_USA_SURFACE, EQ. 0: all high explosive materials are considered.
X	x-coordinate of detonation point, see Figure 18.1.
Y	y-coordinate of detonation point.
Z	z-coordinate of detonation point.

VARIABLE	DESCRIPTION
LT	Lighting time for detonation point. This time is ignored for an acoustic boundary.
PEAK	Peak pressure, p_o , of incident pressure pulse, see remark below.
DECAY	Decay constant, τ
XS	x-coordinate of standoff point, see Figure 18.1.
YS	y-coordinate of standoff point
ZS	z-coordinate of standoff point
NID	Reference node ID near structure

Remarks:

For solid elements (not acoustic) two options are available. If the control card option, *CONTROL_EXPLOSIVE_SHADOW, is not used the lighting time for an explosive element is computed using the distance from the center of the element to the nearest detonation point, L_d ; the detonation velocity, D ; and the lighting time for the detonator, t_d :

$$t_L = t_d + \frac{L_d}{D}$$

The detonation velocity for this default option is taken from the element whose lighting time is computed and does not account for the possibilities that the detonation wave may travel through other explosives with different detonation velocities or that the line of sight may pass outside of the explosive material.

If the control card option, *CONTROL_EXPLOSIVE_SHADOW, is defined, the lighting time is based on the shortest distance through the explosive material. If inert obstacles exist within the explosive material, the lighting time will account for the extra time required for the detonation wave to travel around the obstacles. The lighting times also automatically accounts for variations in the detonation velocity if different explosives are used. No additional input is required for this option but care must be taken when setting up the input. This option works for two and three-dimensional solid elements. It is recommended that for best results:

1. Keep the explosive mesh as uniform as possible with elements of roughly the same dimensions.
2. Inert obstacle such as wave shapers within the explosive must be somewhat larger than the characteristic element dimension for the automatic tracking to function properly. Generally, a factor of two should suffice. The characteristic element dimension is found by checking all explosive elements for the largest diagonal.

3. The detonation points should be either within or on the boundary of the explosive. Offset points may fail to initiate the explosive.
4. Check the computed lighting times in the post processor LS-PREPOST. The lighting times may be displayed at time=0., state 1, by plotting component 7 (a component normally reserved for plastic strain) for the explosive material. The lighting times are stored as negative numbers. The negative lighting time is replaced by the burn fraction when the element ignites.

Line detonations may be approximated by using a sufficient number of detonation points to define the line. Too many detonation points may result in significant initialization cost.

The pressure versus time curve for the acoustic option is defined by:

$$p(t) = p_o e^{-\frac{t}{\tau}}$$

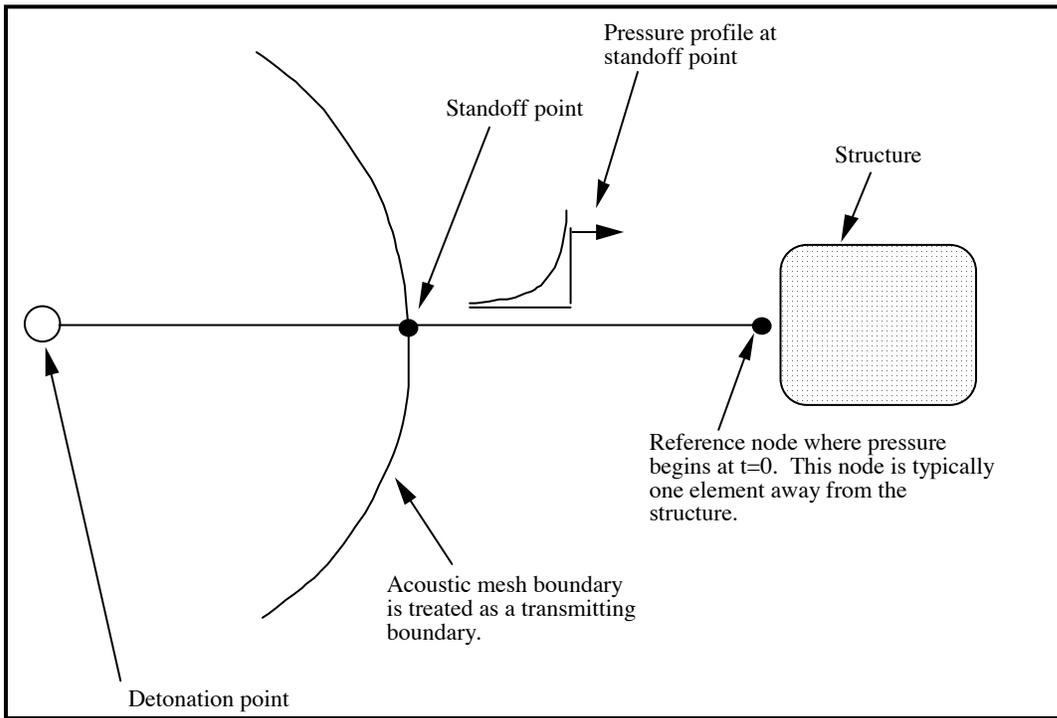


Figure 18.1 Initialization of the initial pressures due to an explosive disturbance is performed in the acoustic media. LS-DYNA automatically determines the acoustic mesh boundary and applies the pressure time history to the boundary. This option is only applicable to the acoustic element formulation, see *SECTION_SOLID.

***INITIAL_FOAM_REFERENCE_GEOMETRY**

Purpose: The reference configuration allows stresses to be initialized in the following hyperelastic material models: 2, 7, 21, 23, 27, 31, 38, 57, 73, 83, 132, 179, and 181. Supported solid elements are the constant stress hexahedron (#1), the fully integrated S/R hexahedron (#2), the tetrahedron (#10), and the pentahedron (#15).

To use this option, the geometry of the foam material is defined in a deformed configuration. The stresses in the low density foam then depend only on the deformation gradient matrix F_{ij} :

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

where x_i is the deformed configuration and X_j is the undeformed configuration. By using this option, dynamic relaxation can be avoided once a deformed configuration is obtained usually on the first run of a particular problem.

Card Format (I8,3E16.0)

Card 1,... 1 2 3 4 5 6 7 8 9 10

Variable	NID	X	Y	Z						
Type	I	F	F	F						
Default	none	0.	0.	0.						
Remarks										

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NID	Node number
X	x coordinate in reference configuration
Y	y coordinate in reference configuration
Z	z coordinate in reference configuration

*INITIAL

*INITIAL_GAS_MIXTURE

*INITIAL_GAS_MIXTURE

Purpose: This command is used to specify (a) which ALE multi-material groups may be present inside an ALE mesh set at time zero, and (b) the corresponding reference gas temperature and density which define the initial thermodynamic state of the gases. The order of the species in the gas mixture corresponds to the order of different gas species defined in the associated *MAT_GAS_MIXTURE card. This card must be used together with a *MAT_GAS_MIXTURE (or equivalently, a *MAT_ALE_GAS_MIXTURE) card.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	STYPE	MMGID	TEMP				
Type	I	I	I	F				
Default	none	0	none	none				

Card 2

Variable	RO1	RO2	RO3	RO4	RO5	RO6	RO7	RO8
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID for initialization. This SID defines the ALE mesh within which certain ALE multi-material group(s) may be present at t=0.
STYPE	Set type for the SID above: EQ.0: SID is a part set ID EQ.1: SID is a part ID
MMGID	ALE Multi-material group ID of the material that may be present at t=0 in the ALE mesh set defined by SID.
TEMP	Initial static temperature of the gas species occupying the ALE mesh. Note that all species in the mixture are assumed to be in thermal equilibrium (having the same T).

VARIABLE	DESCRIPTION
RO1-RO8	Initial densities of the ALE material(s) which may be occupying some region (or all) of the aforementioned ALE mesh, for up to eight different gas species. The order of the density input corresponds to the order of the materials defined in associated *MAT_GAS_MIXTURE card.

Remarks:

1. Please see the example under the *MAT_GAS_MIXTURE card definition for an application of the *INITIAL_GAS_MIXTURE card.
2. The temperature is assumed to be the initial temperature which together with the gas density, will define the initial pressure of the gas species via the perfect gas law $(P|_{t=0} = \rho|_{t=0} (C_P - C_V) T|_{t=0})$. The user should manually check the initial pressure for consistency.
3. Given an ALE mesh, this mesh may initially be occupied by one or more ALE multi-material groups (AMMG). For example, a background ALE mesh (H1) containing AMMG 1 may be partially filled with AMMG 2 via the volume filling command *INITIAL_VOLUME_FRACTION_GEOMETRY. Then there are 2 AMMGs to be initialized for this mesh H1. The commands look like the following.

```

$-----
$ One card is defined for each AMMG that will occupy some elements of a mesh set
*INITIAL_GAS_MIXTURE
$   SID   STYPE   MMGID   T0
$     1     1     (1)   298.15
$   RHO1   RHO2   RHO3   RHO4   RHO5   RHO6   RHO7   RHO8
$ 1.0E-9
*INITIAL_GAS_MIXTURE
$   SID   STYPE   MMGID   T0
$     1     1     (2)   298.15
$   RHO1   RHO2   RHO3   RHO4   RHO5   RHO6   RHO7   RHO8
$ 1.2E-9
$-----

```

*INITIAL

*INITIAL_MOMENTUM

*INITIAL_MOMENTUM

Purpose: Define initial momentum to be deposited in solid elements. This option is to crudely simulate an impulsive type of loading.

Card 1 2 3 4 5 6 7 8

Variable	EID	MX	MY	MZ	DEPT			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			

VARIABLE

DESCRIPTION

EID	Element ID
MX	Initial x-momentum
MY	Initial y-momentum
MZ	Initial z-momentum
DEPT	Deposition time

***INITIAL_STRAIN_SHELL**

Purpose: Initialize strain tensor and inner and outer through thickness integration points at element center. This option is primarily for multi-stage metal forming operations where the accumulated strain is of interest.

Define as many shell elements in this section as desired. The input is assumed to terminate when a new keyword is detected. These strain tensors are defined at the inner and outer integration points and are used for post-processing only. There is no interpolation with this option and the strains are defined in the global cartesian coordinate system. The *DATABASE_EXTENT_BINARY flag STRFLG must be set to unity for this option to work.

Card 1 1 2 3 4 5 6 7 8

Variable	EID							
Type	I							
Default	none							

Define two cards below, one for the inner integration point and the other for the outer integration point, respectively.

Card 2,... 1 2 3 4 5 6 7 8

Variable	EPSxx	EPSyy	EPSzz	EPSxy	EPSyz	EPSzx		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Element ID
EPSij	Define the ij strain component. The strains are defined in the GLOBAL cartesian system.

*INITIAL

*INITIAL_STRAIN_SOLID

*INITIAL_STRAIN_SOLID

Purpose: Initialize strain tensor at element center. This option can be used for multi-stage metal forming operations where the accumulated strain is of interest. This option is available starting in Release 3 of version 971.

Define as many solid elements in this section as desired. The input is assumed to terminate when a new keyword is detected. These strain tensors are defined at the element center and are used for post-processing only. The strains are defined in the global cartesian coordinate system. The *DATABASE_EXTENT_BINARY flag STRFLG must be set to unity for this option to work.

Card 1 1 2 3 4 5 6 7 8

Variable	EID								
Type	I								
Default	none								

Define one card below.

Card 2,... 1 2 3 4 5 6 7 8

Variable	EPSxx	EPSyy	EPSzz	EPSxy	EPSyz	EPSzx		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Element ID
EPSij	Define the ij strain component. The strains are defined in the GLOBAL cartesian system.

***INITIAL_STRESS_BEAM**

Purpose: Initialize stresses and plastic strains in the Hughes-Liu beam elements.

Define as many beams in this section as desired. The input is assumed to terminate when a new keyword is detected.

Card 1 1 2 3 4 5 6 7 8

Variable	EID	RULE	NPTS	LOCAL				
Type	I	I	I	I				
Default	none	none	none	0				

Define NTPS cards below, one per integration point.

Card 2,... 1 2 3 4 5 6 7 8

Variable	SIG11	SIG22	SIG33	SIG12	SIG23	SIG31	EPS	
Type	F	F	F	F	F	F	F	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EID	Element ID
RULE	Integration rule type number: EQ.1.0: 1 × 1 Gauss quadrature EQ.2.0: 2 × 2 Gauss quadrature (default beam), EQ.3.0: 3 × 3 Gauss quadrature, EQ.4.0: 3 × 3 Lobatto quadrature, EQ.5.0: 4 × 4 Gauss quadrature.
NPTS	Number of integration points output.
LOCAL	Coordinate system for stresses: EQ.0: stress components are defined in the global coordinate system. EQ.1: stress components are defined in the local beam system. In the local system components SIG22, SIG33, and SIG23 are set to 0.0.

***INITIAL**

***INITIAL_STRESS_BEAM**

VARIABLE	DESCRIPTION
SIGIJ	Define the IJ stress component.
EPS	Effective plastic strain

***INITIAL_STRESS_DEPTH**

Purpose: Initialize solid element stresses where stress is a function of depth.

Note: This keyword card will be available starting in release 3 of version 971.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	RO_G	ZDATUM	KFACT	LC	LCH		
Type	I	F	F	F	I	I		
Default	none	none	none	0.0	none	none		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID
RO_G	Stress per unit elevation above datum (usually = density x gravity)
ZDATUM	Z-coordinate of datum
KFACT	X- and Y-stress = KFACT x Z-stress
LC	Optional curve of stress vs z-coordinate (ZDATUM is ignored with this option)
LCH	Optional curve of horizontal stress versus z-coordinate (KFACT is ignored with this option)

Remarks:

$Z\text{-stress} = RO_G \times (Z_{\text{element}} - ZDATUM)$. To generate compressive stresses, the datum should be above the highest element – usually at the surface of the soil in geotechnics simulations. If the curve is present, it overrides RO_G and ZDATUM. Note that the points in the curve should be ordered with most negative z-coordinate first – this order looks “upside-down” on the page.

*INITIAL

*INITIAL_STRESS_SECTION

*INITIAL_STRESS_SECTION

Purpose: Initialize the stress in solid elements that are part of a section definition to create a preload. The stress component in the direction normal to the cross-section plane is initialized. This option works with a subset of materials that are incrementally updated including the elastic, viscoelastic, and elastoplastic materials. Rubbers, foams, and materials that are combined with equations-of-state cannot be initialized by this approach. NEW: Hyperelastic materials # 57, 73 and 83 can be initialized with this approach.

Card 1 1 2 3 4 5 6 7 8

Variable	ISSID	CSID	LCID	PSID	VID			
Type	I	A8	I	I	I			
Default	none	none	none	none	none			

VARIABLE

DESCRIPTION

ISSID	Section stress initialization ID.
CSID	Cross-section ID. See *DATABASE_CROSS_SECTION.
LCID	Load curve ID defining preload stress versus time. When the load curve ends or goes to zero, the initialization is assumed to be completed. See remark 2 below.
PSID	Part set ID.
VID	Vector ID defining the direction normal to the cross section. This vector must be defined if *DATABASE_CROSS_SECTION_SET is used to define the cross section. If the cross section is defined using the PLANE option, the normal used in the definition of the plane is used if VID is left undefined.

Remarks:

1. To achieve convergence during explicit dynamic relaxation, the application of the damping options is very important. If contact is active, contact damping is recommended with a value between 10-20 percent. Additional damping, via the option DAMPING_PART_STIFFNESS also speeds convergence where a coefficient of 0.10 is effective. If damping is not used, convergence may not be possible.

2. When defining the load curve, LCID, a ramp starting at the origin should be used to increase the stress to the desired value. The time duration of the ramp should produce a quasi-static response. When the end of the load curve is reached, or when the value of the load decreases from its maximum value, the initialization stops. If the load curve begins at the desired stress value, i.e., no ramp, convergence will take much longer, since the impulsive like load created by the initial stress can excite nearly every frequency in the structural system where stress is initialized.
3. This option currently applies only to materials that are incrementally updated. Hyperelastic materials and materials that require an equation-of-state are not currently supported.
4. Solid elements types 1, 2, 3, 4, 9, 10, 13, 15, 16, 17, and 18 are supported. ALE elements are not supported.

*INITIAL

*INITIAL_STRESS_SHELL

*INITIAL_STRESS_SHELL_{OPTION}

Available options include:

<BLANK>

SET

Purpose: Initialize stresses, history variables, and the effective plastic strain for shell elements.

Define as many shell elements or shell element sets in this section as desired. The input is assumed to terminate when a new keyword is detected. It is not necessary for the location of the through thickness integration points to match those used in the elements which are initialized. The data will be interpolated by LS-DYNA.

Card 1 1 2 3 4 5 6 7 8

Variable	EID/SID	NPLANE	NTHICK	NHISV	NTENSR			
Type	I	I	I	I	I			
Default	none	none	none	0	0			

Define NPLANE X NTHICK cards below, one (if NHISV=0) per integration point. Include optional cards as necessary to define the NHISV history variable. For each through thickness point define NPLANE points. NPLANE should be either 1 or 4 corresponding to either 1 or 4 Gauss integration points. If four integration points are specified, they should be ordered such that their in plane parametric coordinates are at:

$$\left(-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), \left(-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right),$$

respectively.

Card 2... 1 2 3 4 5 6 7 8

Variable	T	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS
Type	F	F	F	F	F	F	F	F

Optional 1 2 3 4 5 6 7 8

Variable	HISV1	HISV2	HISV3	HISV4	HISV5	HISV6	HISV7	HISV8
Type	F	F	F	F	F	F	F	F

Optional

Variable	TENXX	TENYY	TENZZ	TENXY	TENYZ	TENZX		
Type	F	F	F	F	F	F		

VARIABLE**DESCRIPTION**

EID/SID	Element ID or shell set ID, see *SET_SHELL_....
NPLANE	Number of in plane integration points being output.
NTHICK	Number of integration points through the thickness.
NHISV	Number of additional history variables.
NTENSR	Number of components of tensor data taken from the element history variables stored.
T	Parametric coordinate of through thickness integration point between -1 and 1 inclusive.
SIG _{ij}	Define the <i>ij</i> stress component. The stresses are defined in the GLOBAL cartesian system.
EPS	Effective plastic strain
HISV _n	Define the <i>n</i> th history variable.
TEN _{ij}	Define the <i>ij</i> th component of the tensor taken from the history variables. The tensor is defined in the GLOBAL Cartesian system. Define enough lines to provide a total of NTENSOR components, stored six components per line. This applies to material 190 only.

*INITIAL

*INITIAL_STRESS_SOLID

*INITIAL_STRESS_SOLID

Purpose: Initialize stresses and plastic strains for solid elements. This command is not applicable to hyperelastic materials or any material model based on a Total Lagrangian formulation. Furthermore, for *mat_005, *mat_014, and any material that requires an equation-of-state (*EOS), the initialized stresses are deviatoric stresses, not total stresses.

Define as many solid elements in this section as desired. The input is assumed to terminate when a new keyword is detected. If eight points are defined for 1 point LS-DYNA solid elements, the average value will be taken.

Card 1 1 2 3 4 5 6 7 8

Variable	EID	NINT	NHISV	LARGE	IVFLG			
Type	I	I	I	I	I			
Default	none	none	0	0	0			

Define NINT cards below, one per integration point. NINT should be either 1 or 8. If eight Gauss integration points are specified, they should be ordered such that their parametric coordinates are located at:

$$\left(-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right),$$

$$\left(-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), \left(-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right),$$

respectively.

If LARGE=0, then define the following cards, no history variables are allowed.

Card 2... 1 2 3 4 5 6 7 8

Variable	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS	
Type	F	F	F	F	F	F	F	

If **LARGE=1**, then define the following card sets for each integration point. If **NHISV>3** define as many additional cards as necessary. (5E16.0)

Card 2	1	2	3	4	5
Variable	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ
Type	F	F	F	F	F

Card 3...

Variable	SIGZX	EPS	HISV1	HISV2	HISV3
Type	F	F	F	F	F

Card 4...

Variable	HISV...	HISV...	HISV _{n-1}	INITVOL	
Type	F	F	F	F	

VARIABLE	DESCRIPTION
EID	Element ID
NINT	Number of integration points either 1 or 8.
NHISV	Number of additional history variables. If NHISV exceeds the number of integration point history variables required by the constitutive model, only the number required is output; therefore, if in doubt, set NHISV to a large number.
LARGE	Format size, if zero, NHISV must also be set to zero (this is the format used by LS-DYNA versions 970 and earlier) and, if set to 1, a larger format is used and NHISV is used.
IVFLG	Initial Volume flag (only used in large format) EQ.0:last history variable is used as normal, EQ.1:last history variable is used as the initial volume of the element. At least one history variable is required otherwise it may read as zero.

INITIAL**INITIAL_STRESS_SOLID**

VARIABLE	DESCRIPTION
SIGij	Define the <i>ij</i> th stress component. Stresses are defined in the GLOBAL Cartesian system.
EPS	Effective plastic strain
HISV1...n-1	Define n-1 history variables.
INITVOL	Initial volume (HISVn)

***INITIAL_STRESS_TSHELL**

Purpose: Initialize stresses and plastic strains for thick shell elements.

Define as many thick shell elements in this section as desired. The input is assumed to terminate when a new keyword is detected. It is not necessary for the location of the through thickness integration points to match those used in the elements which are initialized. The data will be interpolated by LS-DYNA.

Card 1 1 2 3 4 5 6 7 8

Variable	EID	NPLANE	NTHICK					
Type	I	I	I					
Default	none	none	none					

Define NPLANE X NTHICK cards below, one per integration point. For each through thickness point define NPLANE points. NPLANE should be either 1 or 4 corresponding to either 1 or 4 Gauss integration points. If four integration points are specified, they should be ordered such that their in plane parametric coordinates are at:

$$\left(-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}\right), \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), \left(-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right),$$

respectively.

Card 2 1 2 3 4 5 6 7 8

Variable	T	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS
Type	F	F	F	F	F	F	F	F

INITIAL**INITIAL_STRESS_TSHELL**

VARIABLE	DESCRIPTION
EID	Element ID
NPLANE	Number of in plane integration points being output.
NTHICK	Number of integration points through the thickness.
T	Parametric coordinate of through thickness integration point. between -1 and 1 inclusive.
SIGij	Define the ij stress component. The stresses are defined in the GLOBAL cartesian system.
EPS	Effective plastic strain

***INITIAL_TEMPERATURE_OPTION**

Available options include:

NODE

SET

Purpose: Define initial nodal point temperatures using nodal set ID's or node numbers. These initial temperatures are used in a thermal only analysis or a coupled thermal/structural analysis. See also *CONTROL_THERMAL_SOLVER, *CONTROL_THERMAL_TIMESTEP, and CONTROL_THERMAL_NONLINEAR.

For thermal loading in a structural only analysis, see *LOAD_THERMAL_OPTION.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID/NID	TEMP	LOC					
Type	I	I	I					
Default	none	0.	0					
Remark	1							

VARIABLE

DESCRIPTION

NSID/NID	Nodal set ID or nodal point ID, see also *SET_NODES: EQ.0: all nodes are included (set option only).
TEMP	Temperature at node or node set.
LOC	Application of surface for thermal shell elements, see parameter, TSHELL, in the *CONTROL_SHELL input: EQ.-1: lower surface of thermal shell element EQ. 0: middle surface of thermal shell element EQ. 1: upper surface of thermal shell element

Remarks:

1. If a nodal temperature is specified on more than one input card, then the last set input will determine its temperature unless it is specified on a *INITIAL_TEMPERATURE_NODE card.

*INITIAL

*INITIAL_VEHICLE_KINEMATICS

*INITIAL_VEHICLE_KINEMATICS

Purpose: Define initial kinematical information for a vehicle. In its initial orientation, the vehicle's yaw, pitch, and roll axes must be aligned with the global axes. Successive simple rotations are taken about these body fixed axes.

Card 1 1 2 3 4 5 6 7 8

Variable	GRAV	PSID	XO	YO	ZO	XF	YF	ZF
Type	I	I	F	F	F	F	F	F
Default	none	none	0.	0.	0.	0.	0.	0.

Card 2

Variable	VX	VY	VZ	AAXIS	BAXIS	CAXIS		
Type	F	F	F	I	I	I		
Default	0.	0.	0.	0	0	0		

Card 3

Variable	AANG	BANG	CANG	WA	WB	WC		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
GRAV	Gravity direction code. EQ. 1: Global +x direction. EQ.-1: Global -x direction. EQ. 2: Global +y direction. EQ.-2: Global -y direction. EQ. 3: Global +z direction. EQ.-3: Global -z direction. Note: this must be the same for all vehicles present in the model.
PSID	Part set ID.
XO	x-coordinate of initial position of mass center.
YO	y-coordinate of initial position of mass center.
ZO	z-coordinate of initial position of mass center.
XF	x-coordinate of final position of mass center.
YF	y-coordinate of final position of mass center.
ZF	z-coordinate of final position of mass center.
VX	x-component of mass center velocity.
VY	y-component of mass center velocity.
VZ	z-component of mass center velocity.
AAXIS	First rotation axis code. EQ.1: Initially aligned with global x-axis. EQ.2: Initially aligned with global y-axis. EQ.3: Initially aligned with global z-axis.
BAXIS	Second rotation axis code.
CAXIS	Third rotation axis code.
AANG	Rotation angle about the first rotation axis (degrees).
BANG	Rotation angle about the second rotation axis (degrees).
CANG	Rotation angle about the third rotation axis (degrees).
WA	Angular velocity component for the first axis (radian/second).

VARIABLE	DESCRIPTION
WB	Angular velocity component for the second axis (radian/second).
WC	Angular velocity component for the third axis (radian/second).

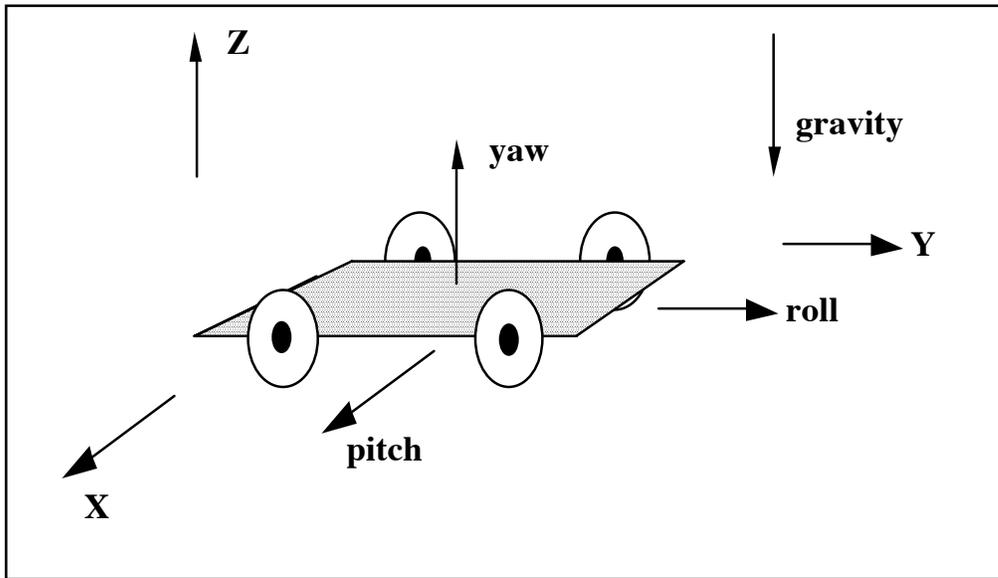


Figure 18.2. The vehicle pictured is to be oriented with a successive rotation sequence about the yaw, pitch, and roll axes, respectively. Accordingly, AAXIS=3, BAXIS=1, and CAXIS=2. The direction of gravity is given by GRAV=-3.

INITIAL_VELOCITY**INITIAL*****INITIAL_VELOCITY**

Purpose: Define initial nodal point translational velocities using nodal set ID's. This may also be used for sets in which some nodes have other velocities. See NSIDEX below.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	NSIDEX	BOXID	IRIGID				
Type	I	I	I	I				
Default	none	0	0	0				
Remark	1							

Card 2

Variable	VX	VY	VZ	VXR	VYR	VZR		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Define the following card if and only if NSIDEX>0.

Card 3 1 2 3 4 5 6 7 8

Variable	VXE	VYE	VZE	VXRE	VYRE	VZRE		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
NSID	Nodal set ID, see *SET_NODES, containing nodes for initial velocity: If NSID = 0 the initial velocity is applied to all nodes.
NSIDEX	Nodal set I, see *SET_NODES, containing nodes that are exempted from the imposed velocities and may have other initial velocities.
BOXID	All nodes in box which belong to NSID are initialized. Nodes outside the box are not initialized. Exempted nodes are initialized to velocities defined by VXE, VYE, and VZE below regardless of their location relative to the box.
IRIGID	Option to overwrite rigid body velocities defined on *PART_INERTIA and *CONSTRAINED_NODAL_RIGID_BODY_INERTIA cards. GE.1: part set ID, containing ID of parts to overwrite. Center of gravity of part must lie within box BOXID. If BOXID is not defined then all parts defined in the set are overwritten. EQ.-1: Overwrite velocities for all *PART_INERTIA's and *CONSTRAINED_NODAL_RIGID_BODY_INERTIA's with a center of gravity within box BOXID. If BOXID is not defined then all are overwritten. EQ.-2: Overwrite velocities for all *PART_INERTIA's and *CONSTRAINED_NODAL_RIGID_BODY_INERTIA's.
VX	Initial velocity in x-direction
VY	Initial velocity in y-direction
VZ	Initial velocity in z-direction
VXR	Initial rotational velocity about the x-axis
VYR	Initial rotational velocity about the y-axis
VZR	Initial rotational velocity about the z-axis
VXE	Initial velocity in x-direction of exempted nodes
VYE	Initial velocity in y-direction of exempted nodes
VZE	Initial velocity in z-direction of exempted nodes
VXRE	Initial rotational velocity in x-direction of exempted nodes
VYRE	Initial rotational velocity in y-direction of exempted nodes
VZRE	Initial rotational velocity in z-direction of exempted nodes

Remarks:

1. This generation input must not be used with *INITIAL_VELOCITY_GENERATION keyword.
2. If a node is initialized on more than one input card set, then the last set input will determine its velocity. However, if the nodal velocity is also specified on a *INITIAL_VELOCITY_NODE card, then the velocity specification on this card will be used.
3. Unless the option IRIGID is specified rigid bodies, initial velocities given in *PART_INERTIA will overwrite generated initial velocities. The IRIGID option will cause the rigid body velocities specified on the *PART_INERTIA input to be overwritten. To directly specify the motion of a rigid body without using the keyword, *PART_INERTIA, which also requires the definition of the mass properties, use the keyword option, *INITIAL_VELOCITY_RIGID_BODY.
4. Nodes which belong to rigid bodies must have motion consistent with the translational and rotational velocity of the center of gravity (c.g.) of the rigid body. During initialization the rigid body translational and rotational rigid body momentum's are computed based on the prescribed nodal velocity field. From this rigid body momentum, the translational and rotational velocities of the nodal points are computed and reset to the new values. These new values may or may not be the same as the values prescribed for the nodes that make up the rigid body. Sometimes this occurs in single precision due to numerical round-off. If a problem like this occurs specify the velocity using the keyword: *INITIAL_VELOCITY_RIGID_BODY.

***INITIAL**

***INITIAL_VELOCITY_NODE**

***INITIAL_VELOCITY_NODE**

Purpose: Define initial nodal point velocities for a node.

Card	1	2	3	4	5	6	7	8
Variable	NID	VX	VY	VZ	VXR	VYR	VZR	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NID	Node ID
VX	Initial translational velocity in x-direction
VY	Initial translational velocity in y-direction
VZ	Initial translational velocity in z-direction
VXR	Initial rotational velocity about the x-axis
VYR	Initial rotational velocity about the y-axis
VZR	Initial rotational velocity about the z-axis

See Remarks on *INITIAL_VELOCITY card.

***INITIAL_VELOCITY_RIGID_BODY**

Purpose: Define the initial translational and rotational velocities at the center of gravity (c.g.) for a rigid body or a nodal rigid body. This input overrides all other velocity input for the rigid body and the nodes which define the rigid body.

Card 1 2 3 4 5 6 7 8

Variable	PID	VX	VY	VZ	VXR	VYR	VZR	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

VARIABLE**DESCRIPTION**

PID	Part ID of the rigid body or the nodal rigid body.
VX	Initial translational velocity at the c.g. in global x-direction.
VY	Initial translational velocity at the c.g. in global y-direction.
VZ	Initial translational velocity at the c.g. in global z-direction.
VXR	Initial rotational velocity at the c.g. about the global x-axis.
VYR	Initial rotational velocity at the c.g. about the global y-axis.
VZR	Initial rotational velocity at the c.g. about the global z-axis.

See remarks 3 and 4 on the *INITIAL_VELOCITY input description.

*INITIAL

*INITIAL_VELOCITY_GENERATION

*INITIAL_VELOCITY_GENERATION

Purpose: Define initial velocities for rotating and translating bodies. Caution: Rigid body velocities cannot be reinitialized after dynamic relaxation by setting PHASE=1 since rigid body velocities are always restored to the values that existed prior to dynamic relaxation. Reinitialization of velocities after dynamic relaxation is only for nodal points of deformable bodies; therefore, if rigid bodies are present in the part set ID, this input should be defined twice, once for IPHASE=0 and again for IPHASE=1.

Card 1 1 2 3 4 5 6 7 8

Variable	ID	STYP	OMEGA	VX	VY	VZ	IVATN	
Type	I	I	F	F	F	F	I	
Default	none	none	0.	0.	0.	0.	0	

Card 2

Variable	XC	YC	ZC	NX	NY	NZ	PHASE	
Type	F	F	F	F	F	F	I	
Default	0.	0.	0.	0.	0.	0.	0	

VARIABLE

DESCRIPTION

ID Part ID, part set ID, or node set ID if zero STYP is ignored and all velocities are set. WARNING if IVATN=0: If a part ID of a rigid body is specified only the nodes that belong to elements of the rigid body are initialized. Nodes defined under the keyword. *CONSTRAINED_EXTRA_NODES are not initialized. Set IVATN=1 to initialize velocities of slaved nodes and parts.

STYP Set type:
 EQ.1: part set ID, see *SET_PART,
 EQ.2: part ID, see *PART,
 EQ.3: node set ID, see *SET_NODE.

OMEGA Angular velocity about the rotational axis.

VARIABLE	DESCRIPTION
VX	Initial translational velocity in global x-direction.
VY	Initial translational velocity in global y-direction.
VZ	Initial translational velocity in global z-direction.
IVATN	Flag for setting the initial velocities of slave nodes and parts: EQ.0: slaved parts are ignored. EQ.1: slaved parts and slaved nodes of the master parts will be assigned initial velocities like the master part.
XC	x-coordinate on rotational axis.
YC	y-coordinate on rotational axis.
ZC	z-coordinate on rotational axis.
NX	x-direction cosine.
NY	y-direction cosine.
NZ	z-direction cosine.
PHASE	Flag specifying phase of the analysis the velocities apply to: EQ.0. Velocities applied immediately, EQ.1. Velocities applied after dynamic relaxation.

Remarks:

1. This generation input must not be used with *INITIAL_VELOCITY or *INITIAL_VELOCITY_NODE options.
2. The velocities are initialized in the order the *INITIAL_VELOCITY_GENERATION input is defined. Later input via the *INITIAL_VELOCITY_GENERATION keyword may overwrite the velocities previously set.
3. For rigid bodies, initial velocities given in *PART_INERTIA will overwrite generated initial velocities.
4. Nodes which belong to rigid bodies must have motion consistent with the translational and rotational velocity of the rigid body. During initialization the rigid body translational and rotational rigid body momentum's are computed based on the prescribed nodal velocities. From this rigid body motion the velocities of the nodal points are computed and reset to the new values. These new values may or may not be the same as the values prescribed for the node.
5. SPH elements can be initialized using the STYP=3 option only.

*INITIAL

*INITIAL_VOID

*INITIAL_VOID_OPTION

Available options include:

PART

SET

Purpose: Define initial voided part set ID's or part numbers. Void materials cannot be created during the calculation. Fluid elements which are evacuated, e.g., by a projectile moving through the fluid, during the calculation are approximated as fluid elements with very low densities. The constitutive properties of fluid materials used as voids must be identical to those of the materials which will fill the voided elements during the calculation. Mixing of two fluids with different properties is not permitted with this option.

Card 1 1 2 3 4 5 6 7 8

Variable	PSID/PID							
Type	I							
Default	none							
Remark	1							

VARIABLE

DESCRIPTION

PSID/PID

Part set ID or part ID, see also *SET_PART:

Remarks:

This void option and multiple materials per element, see *ALE_MULTI-MATERIAL_GROUP are incompatible and cannot be used together in the same run.

***INITIAL_VOLUME_FRACTION**

Purpose: Define initial volume fractions of different materials in multi-material ALE elements.

Card 1 1 2 3 4 5 6 7 8

Variable	EID	VF1	VF2	VF3	VF4	VF5	VF6	VF7
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE**DESCRIPTION**

EID	Element ID.
VF1	Volume fraction of multi-material group 1, AMMGID=1.
VF2	Volume fraction of multi-material group 2. Only needed in simulations with 3 material groups. Otherwise VF2=1-VF1.
VF3	Volume fraction of multi-material group 3, AMMGID=3.
VF4	Volume fraction of multi-material group 4, AMMGID=4.
VF5	Volume fraction of multi-material group 5, AMMGID=5.
VF6	Volume fraction of multi-material group 6, AMMGID=6.
VF7	Volume fraction of multi-material group 7, AMMGID=7.

*INITIAL

*INITIAL_VOLUME_FRACTION_GEOMETRY

*INITIAL_VOLUME_FRACTION_GEOMETRY

Purpose: This is a volume-filling command for defining the volume fractions of various ALE multi-material groups (AMMG) that can occupy certain regions in some specified ALE mesh set. It is applied only for multi-material ALE model. See Remark 5.

Defines the background ALE mesh set & an AMMGID that initially fills it.

Card 1 1 2 3 4 5 6 7 8

Variable	FMSID	FMIDTYP	BAMMG	NTRACE				
Type	I	I	I	I				
Default	none	0	0	3				

VARIABLE

DESCRIPTION

FMSID	A background ALE (fluid) mesh SID to be initialized or filled with various AMMG's. This set ID refers to one or more ALE parts.
FMIDTYP	ALE mesh set ID type: EQ.0: FMSID is an ALE part set ID (PSID). EQ.1: FMSID is an ALE part ID (PID).
BAMMG	The background fluid group ID or ALE Multi-Material group ID (AMMGID) that initially fills all ALE mesh region defined by FMSID.
NTRACE	Number of sampling points for volume filling detection. Typically NTRACE ranges from 3 to maybe 10 (or more). The higher it is, the finer the ALE element is divided so that small gaps between 2 Lagrangian shells may be filled in. See Remark 6.

Defines the container type and the AMMGID that fills inside or outside it.

Card a 1 2 3 4 5 6 7 8

Variable	CONTTYP	FILLOPT	FAMMG					
Type	I	I	I					
Default	none	0	none					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CONTTYP	<p>A “container” defines a Lagrangian surface boundary of a spatial region, inside (or outside) of which, an AMMG would fill up. CONTTYP defines the container geometry type of this surface boundary (or shell structure).</p> <p>EQ.1: The container geometry is defined by a part ID (PID) or a part set ID (PSID), where the parts should be defined by shell elements (see *PART or *SET_PART).</p> <p>EQ.2: The container geometry is defined by a segment set (SGSID).</p> <p>EQ.3: The container geometry is defined by a plane: a point and a normal vector.</p> <p>EQ.4: The container geometry is defined by a conical surface: 2 end points and 2 corresponding radii.</p> <p>EQ.5: The container geometry is defined by a cuboid or rectangular box: 2 opposing end points, minimum to maximum coordinates.</p> <p>EQ.6: The container geometry is defined by a sphere: 1 center point, and a radius.</p>
FILLOPT	<p>A flag to indicate which side of the container surface the AMMG is supposed to fill. The “head” side of a container surface/segment is defined as the side pointed to by the heads of the normal vectors of the segments (“tail” side refers to opposite direction to “head”). See Remark 7.</p> <p>EQ.0: The “head” side of the geometry defined above will be filled with fluid (default).</p> <p>EQ.1: The “tail” side of the geometry defined above will be filled with fluid.</p>
FAMMG	<p>This defines the fluid group ID or ALE Multi-Material group ID (AMMGID) which will fill up the interior (or exterior) of the space defined by the “container”. <u>The order of AMMGIDs are defined by the order in which they are listed under *ALE MULTI-MATERIAL GROUP card.</u> For example, that card defines AMMGID=1 on its first line, and AMMGID=2 on its second line, etc.</p>

*INITIAL

*INITIAL_VOLUME_FRACTION_GEOMETRY

CONTTYP = 1 for container defined by a shell PID or PSID

Card b-1 1 2 3 4 5 6 7 8

Variable	SID	STYPE	NORMDIR	XOFFSET				
Type	I	I	I	F				
Default	none	0	0	0.0				
Remark			obsolete					

VARIABLE

DESCRIPTION

SID A Set ID pointing to a part ID (PID) or part set ID (PSID) of the Lagrangian shell element structure defining the “container” geometry to be filled (see *PART or *SET_PART).

SSTYPE Set ID type:
EQ.0: Container SID is a Lagrangian part set ID (PSID).
EQ.1: Container SID is a Lagrangian part ID (PID).

NORMDIR Obsolete (see Remark 7).

XOFFSET Absolute length unit for offsetting the fluid interface from the nominal fluid interface LS-DYNA would otherwise define by default. This parameter only applies to GEOTYPE=1 (4th column) and GEOTYPE=2 (3rd column). This is applicable to cases in which high pressure fluid is contained within a container. The offset allows LS-DYNA time to prevent leakage. In general, this may be set to roughly 5-10% of the ALE elm width. It may be important only for when ILEAK is turned ON to give the code time to "catch" the leakage. If ILEAK is not ON, this may not be necessary.

CONTTYP = 2 for container defined by a SGSID

Card b-2 1 2 3 4 5 6 7 8

Variable	SGSID	NORMDIR	XOFFSET					
Type	I	I	F					
Default	none	0	0.0					
Remark		obsolete						

VARIABLE

DESCRIPTION

SGSID	Segment Set ID defining the “container”, see *SET_SEGMENT.
NORMDIR	Obsolete (see Remark 7).
XOFFSET	Absolute length unit for offsetting the fluid interface from the nominal fluid interface LSDYNA would otherwise define by default. This parameter only applies to GEOTYPE=1 (4 th column) and GEOTYPE=2 (3 rd column). This is applicable to cases in which high pressure fluid is contained within a container. The offset allows LS-DYNA time to prevent leakage. In general, this may be set to roughly 5-10% of the ALE elm width. It may be important only for when ILEAK is turned ON to give the code time to "catch" the leakage. If ILEAK is not ON, this may not be necessary.

CONTTYP = 3 for container defined by a PLANE

Card b-3 1 2 3 4 5 6 7 8

Variable	X0	Y0	Z0	XCOS	YCOS	ZCOS		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

INITIAL**INITIAL_VOLUME_FRACTION_GEOMETRY**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
X0, Y0, Z0	X, Y and Z coordinate of a spatial point on the plane.
X1, Y1, Z1	X, Y and Z direction cosines of the plane normal vector. The filling will occur on the side pointed to by the plane normal vector (or “head” side).

CONTTYP = 4 for container defined by a Cylinder and Cone.

Card b-4 1 2 3 4 5 6 7 8

Variable	X0	Y0	Z0	X1	Y1	Z1	R1	R2
Type	F	F	F	F	F	F	F	F
Default	none							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
X0, Y0, Z0	X, Y and Z coordinate of the center of the 1 st base of the cone.
X1, Y1, Z1	X, Y and Z coordinate of the center of the 2 nd base of the cone.
R1	Radius of the 1 st base of the cone
R2	Radius of the 2 nd base of the cone

CONTTYP = 5 for container defined by a Rectangular Box

Card b-5 1 2 3 4 5 6 7 8

Variable	X0	Y0	Z0	X1	Y1	Z1		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
X0, Y0, Z0	X, Y and Z coordinate of the maximum coordinate of the box.
X1, Y1, Z1	X, Y and Z coordinate of the minimum coordinate of the box.

CONTTYP = 6 for container defined by a Sphere

Card b-6 1 2 3 4 5 6 7 8

Variable	X0	Y0	Z0	R0				
Type	F	F	F	F				
Default	none	none	none	none				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
X0, Y0, Z0	X, Y and Z coordinate of the center of the sphere.
R0	Radius of the sphere

Remarks:

1. After card **1** defining the basic mesh filled by certain fluid group (AMMGID), each “filling action” will require 2 additional lines of input (cards **a** and **b-#**, where **#** is the CONTTYP value). At the minimum there will be 3 cards required for this command (**1**, **a** and **b-#**) for 1 “filling action”.
2. There can be one or more “filling actions” prescribed for each definition of this command. The “filling actions” take place in the prescribed order and the effects are accumulative. The latter filling actions over-write the previous ones. Therefore any complex filling logics will require some planning. For example, the following card sequence, with 2 “filing actions”, is allowable:

1

a (CONTTYP=1)

b-1

a (CONTTYP=3)

b-3

This sequence of cards prescribes a system of background ALE mesh with 2 “filing actions” to be executed. The 1st is a filling of a CONTTYP=1, and the 2nd of CONTTYP=3.

3. Card **a** is required for all container geometry types (CONTTYP). Card **b-#** defines the container actual geometry and corresponds to each of the CONTTYP choice.
4. If ELFORM=12 (in *SECTION_SOLID) for the ALE mesh to be filled, i.e. single-material-and-void element formulation is used, then the non-void material is automatically referred to as AMMG 1 and the void as AMMG 2. The multi-material groups are implied even though no *ALE_MULTI-MATERIAL_GROUP card is required.
5. A simple ALE background mesh (for example, a cuboid mesh) can be constructed enveloping some Lagrangian shell structure (or container). The ALE region inside this Lagrangian shell container may be filled with one multi-material group (AMMG1), and the outside region with another (AMMG2). This approach simplifies the mesh generation requirements for ALE material parts with complex geometries.
6. Default is NTRACE=3 → total number is $(2*NTRACE+1)^3 = (7X7X7)$. This means an ALE element is subdivided into 7X7X7 regions. Each is to be filled in with the appropriate AMMG. An example of this application would be the filling of initial gas between multiple layers of Lagrangian airbag shell elements sharing the same ALE element.
7. The simplest approach is to:
 - (a) Set "NORMDIR=0" (default).
 - (b) Define the shell (or segment) container with inward normals.
 - (c) Using only FILLOPT in card **a** to control which side of the container to fill.

Example:

Consider a simple ALE model with ALE parts H1-H5 (5 AMMGs possible) and 1 Lagrangian shell (container) part S6. Only parts H1 and S6 initially have their meshes defined. We will perform 4 "filling actions". The volume filling results after each step will be shown below to clarify the concept used. The input for the volume filling looks like this.

```

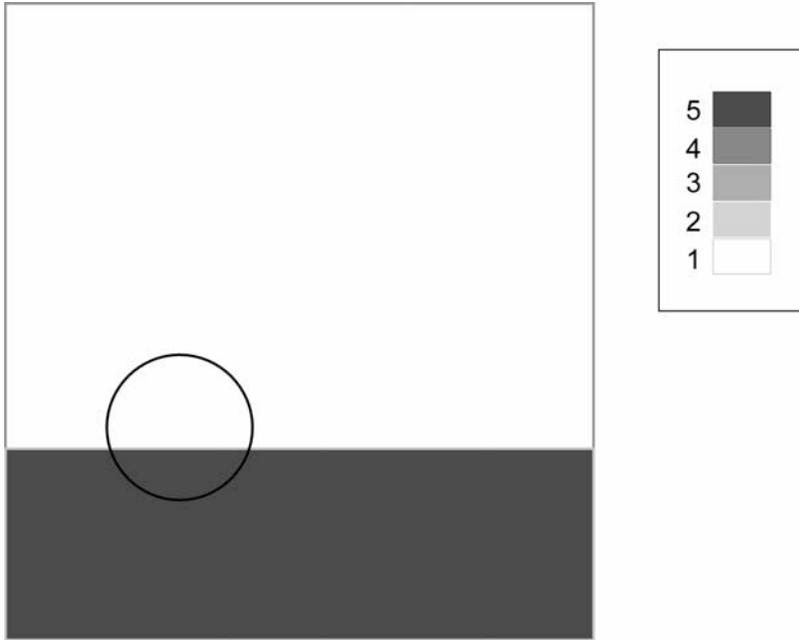
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ H1 = AMMG 1 = fluid 1 initially occupying whole ALE mesh= background mesh
$ H5 = AMMG 5 = fluid 5 fills below a plane = filling action 1 = CONTTYP=3
$ H2 = AMMG 2 = fluid 2 fills outside S5 = filling action 2 = CONTTYP=1
$ H3 = AMMG 3 = fluid 3 fills inside a cone = filling action 3 = CONTTYP=4
$ H4 = AMMG 4 = fluid 4 fills inside a box = filling action 4 = CONTTYP=5
$ S6 = Lagrangian shell container
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_MULTI-MATERIAL_GROUP
    1      1
    2      1
    3      1
    4      1
    5      1
*INITIAL_VOLUME_FRACTION_GEOMETRY
$ The 1st card fills the whole pid H1 with AMMG 1=background ALE mesh
$ FMSID  FMIDTYP  BAMMG  <=== card 1: background fluid
    1      1      1
$ filling action 1 = AMMG 5 fill all elms below a plane
$ CONTTYP  FILLOPT  FILAMMGID  <=== card a : container: CONTTYP=3=plane
    3      0      5
$ X0, Y0, Z0, NX, NY, NZ <=== card b-3: details on container =plane
25.0,20.0, 0.0, 0.0,-1.0,0.0
$ filling action 2: AMMG 2 fills OUTSIDE (FILLOPT=1) shell S6 (inward normals);
$ CONTTYP  FILLOPT  FAMMG  <== card a : container #1; FILLOPT=1=fill tail
    1      1      2
$ SETID  SETTYPE  NORMDIR  <== card b-1: details on container #1
    6      1      0
$ filling action 3 = AMMG 3 fill all elms inside a CONICAL region
$ CONTTYP  FILLOPT  FAMMG  CONTTYP = 4 = Container = conical region
    4      0      3
$ X1 Y1 Z1 X2 Y2 Z2 R1 R2
    25.0 75.0 0.0 25.0 75.0 1.0 8.0 8.0
$ filling action 4 = AMMG 4 fill all elms inside a BOX region
$ CONTTYP  FILLOPT  FFLUIDID : CONTTYP=5 = "BOX"
    5      0      4
$ XMIN YMIN ZMIN XMAX YMAX ZMAX
    65.0 35.0 0.0 85.0 65.0 1.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8

```

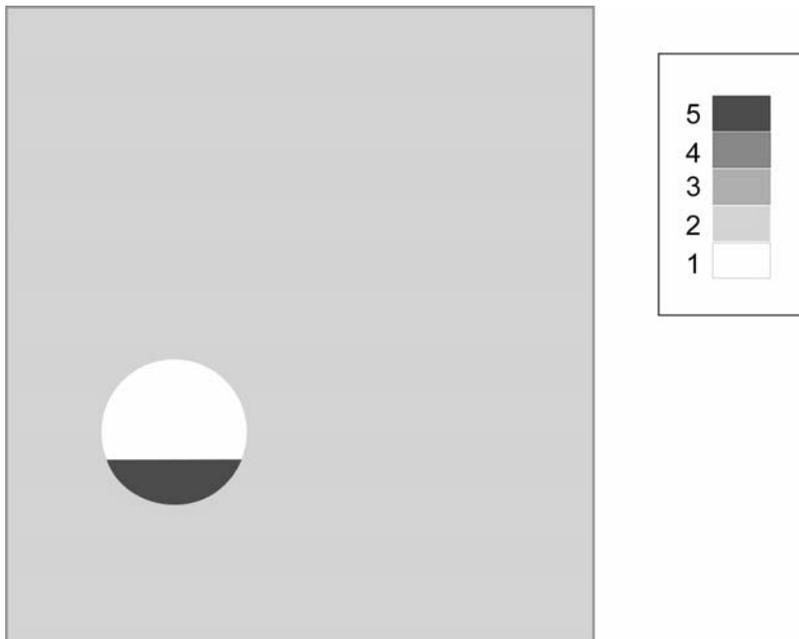
*INITIAL

*INITIAL_VOLUME_FRACTION_GEOMETRY

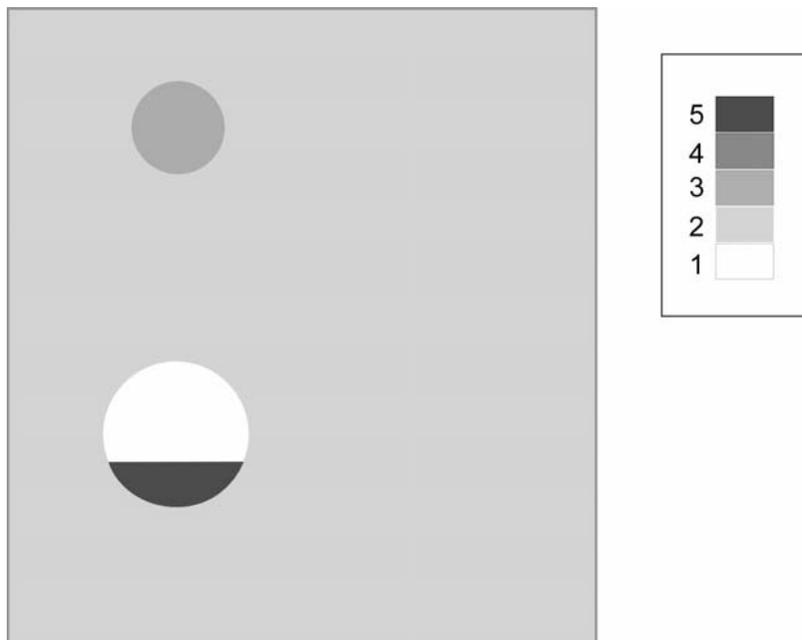
Before the 1st “filling action” the whole ALE mesh of part H1 is filled with AMMG 1 (white).
After the 1st “filling action”, AMMG 5 fills below the specified plane.



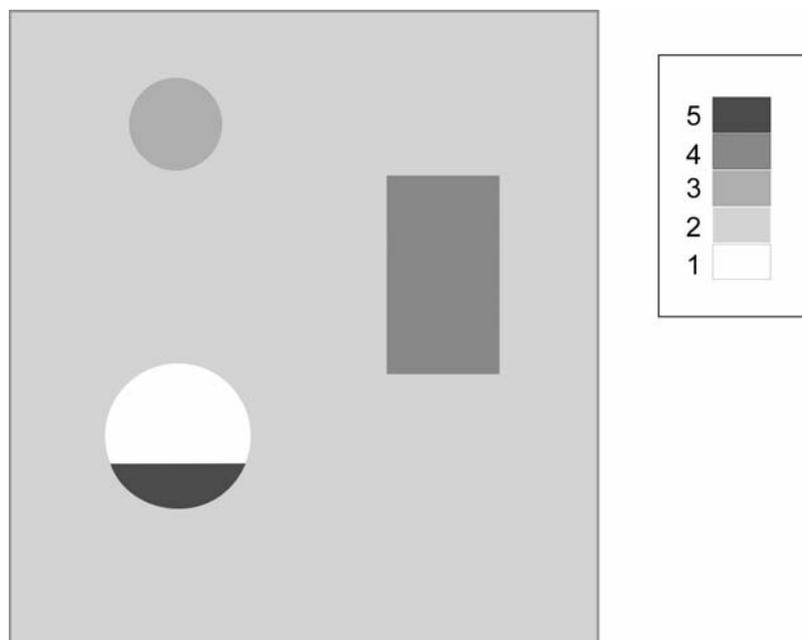
After the 1st and 2nd “filling actions”, it fills outside the shell (S6) with AMMG 2.



After the 1st, 2nd and 3rd “filling actions”, it fills in the analytical sphere with AMMG 3.



After the 1st, 2nd, 3rd and 4th “filling actions”, it fills in the analytical box with AMMG 4.



***INITIAL**

***INITIAL_VOLUME_FRACTION_GEOMETRY**

***INTEGRATION**

In this section the user defined integration rules for beam and shell elements are specified. IRID refers to integration rule identification number on *SECTION_BEAM and *SECTION_SHELL cards respectively. Quadrature rules in the *SECTION_SHELL and *SECTION_BEAM cards need to be specified as a negative number. The absolute value of the negative number refers to user defined integration rule number. Positive rule numbers refer to the built in quadrature rules within LS-DYNA. The keyword cards in this section are:

***INTEGRATION_BEAM**

***INTEGRATION_SHELL**

*INTEGRATION

*INTEGRATION_BEAM

*INTEGRATION_BEAM

Purpose: Define user defined through the thickness integration rules for the beam element.

Card 1 1 2 3 4 5 6 7 8

Variable	IRID	NIP	RA	ICST	K			
Type	I	I	F	I	I			
Default	none	0	0.0	0	0			

Define the following card if and only if ICST>0.

Card 1 2 3 4 5 6 7 8

Variable	D1	D2	D3	D4	SREF	TREF	D5	D6
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	1.0	0.0	none	none

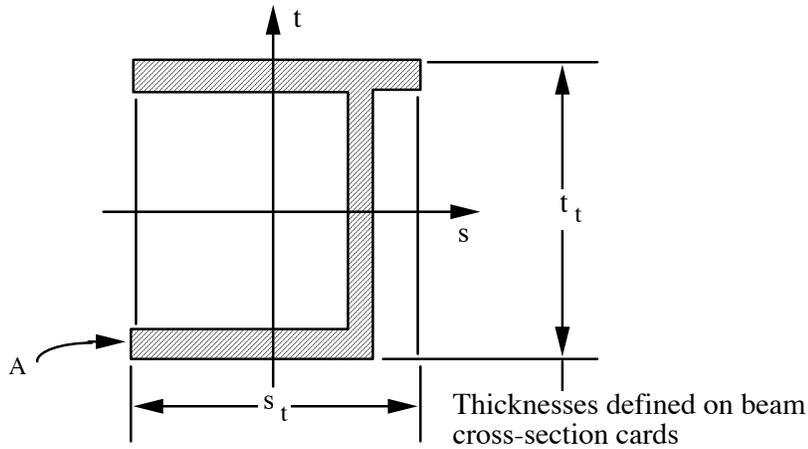
Define NIP cards below (Skip if NIP=0).

Card 1 2 3 4 5 6 7 8

Variable	S	T	WF	PID				
Type	F	F	F	I				

VARIABLE	DESCRIPTION
IRID	Integration rule ID. IRID refers to IRID on *SECTION_BEAM card.
NIP	Number of integration points, see also ICST.
RA	Relative area of cross section, i.e., the actual cross-sectional area divided by the area defined by the product of the specified thickness in the s direction and the thickness in the t direction. See also ICST below and Figure 19.1.
ICST	Standard cross section type, ICST. If this type is nonzero then NIP and the relative area above should be input as zero. See the discussion following the input description Figure 19.3. EQ. 01: I-shape EQ. 12: Cross EQ. 02: Channel EQ. 13: H-shape EQ. 03: L-shape EQ. 14: T-shape1 EQ. 04: T-shape EQ. 15: I-shape2 EQ. 05: Tubular box EQ. 16: Channel1 EQ. 06: Z-shape EQ. 17: Channel2 EQ. 07: Trapezoidal EQ. 18: T-shape2 EQ. 08: Circular EQ. 19: Box-shape1 EQ. 09: Tubular EQ. 20: Hexagon EQ. 10: I-shape1 EQ. 21: Hat-shape EQ. 11: Solid box EQ. 22: Hat-shape1
K	Integration refinement parameter for standard cross section types. Select an integer ≥ 0 . See Figure below.
D1-D6	Cross-section dimensions. See Figure below.
SREF	s_{ref} , location of reference surface normal to s, for the Hughes-Liu beam only. This option is only useful if the beam is connected to a shell or another beam on its outer surface. Overrides NSLOC in *SECTION_BEAM.
TREF	t_{ref} , location of reference surface normal to t, for the Hughes-Liu beam only. This option is only useful if the beam is connected to a shell or another beam on its outer surface. Overrides NTLOC in *SECTION_BEAM.
S	Normalized s coordinate of integration point, $-1 \leq s \leq 1$.
T	Normalized t coordinate of integration point, $-1 \leq t \leq 1$.
WF	Weighting factor, A_{Ti} , i.e., the area associated with the integration point divided by actual cross sectional area $A_{Ti} = A_i/A$, see Figure 19.2.

VARIABLE	DESCRIPTION
PID	Optional PID, used to identify material properties for this integration point. If zero, the “master” PID (referenced on *ELEMENT) will be used. This feature will be available in release 3 of version 971.



$$\text{Relative Area} = \frac{A}{s_t \cdot t_t}$$

Figure 19.1. Definition of relative area for user defined integration rule.

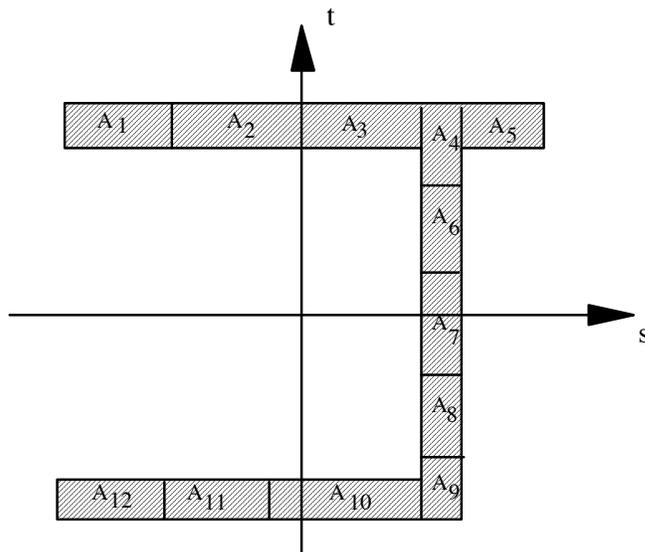


Figure 19.2. Definition of integration points for user defined integration rule.

Remarks:

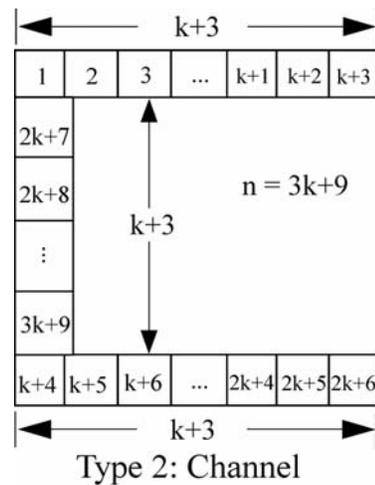
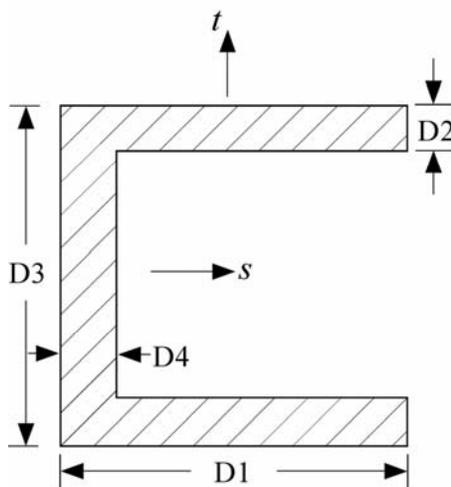
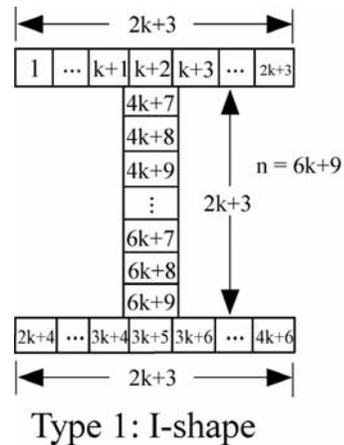
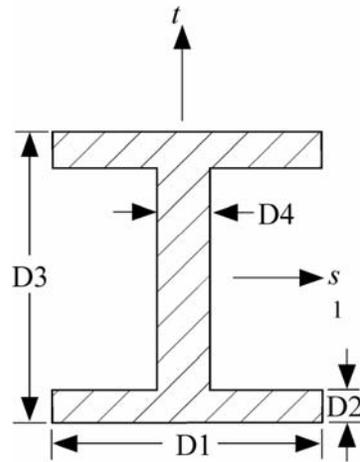
The input for standard beam section types is defined below. In following figures the dimensions are shown on the left and the location of the integration points are shown on the right. If a quantity is not defined in the sketch, then it should be set to zero in the input. The input quantities include:

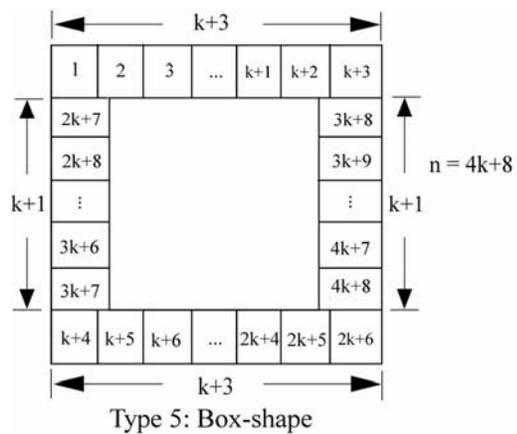
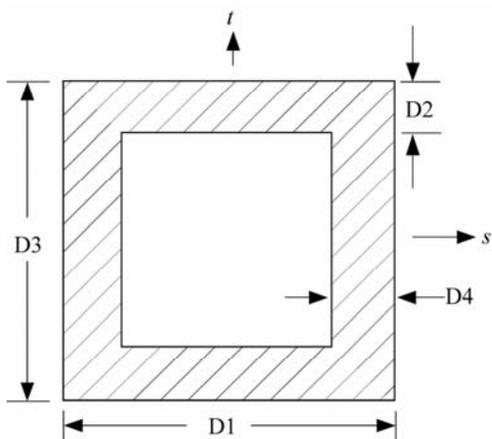
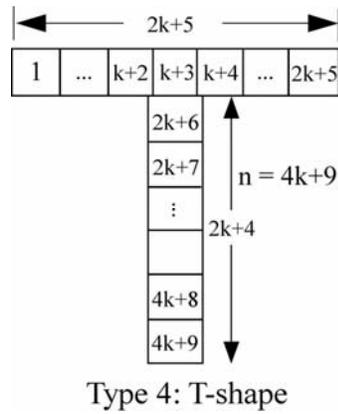
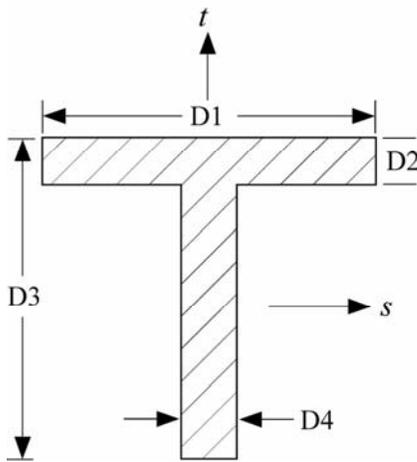
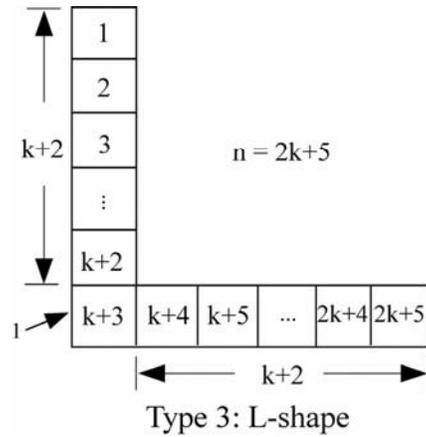
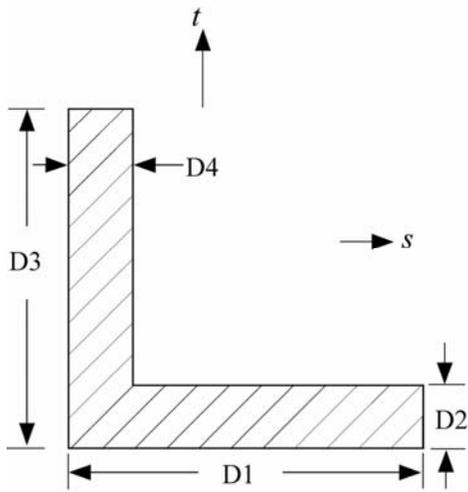
D1-D6 = Dimensions of section

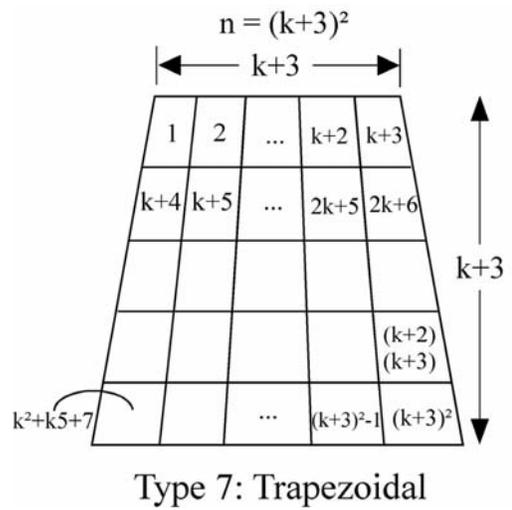
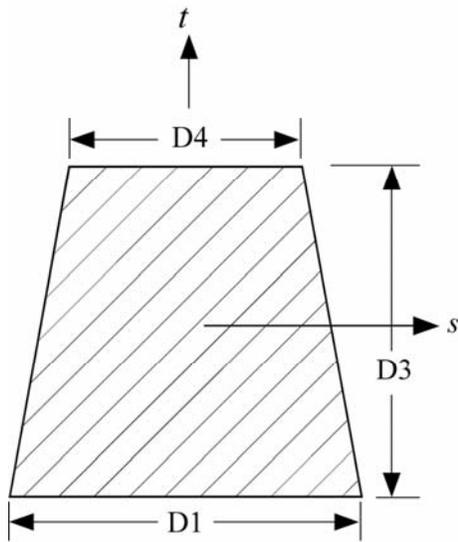
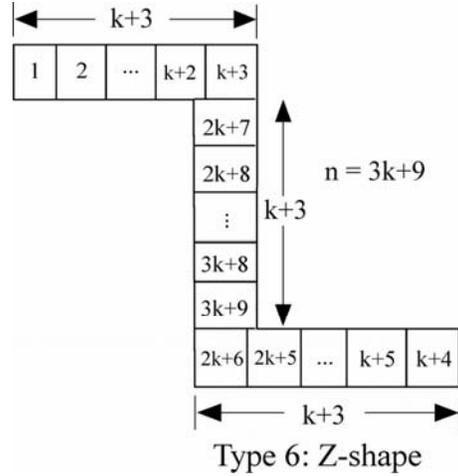
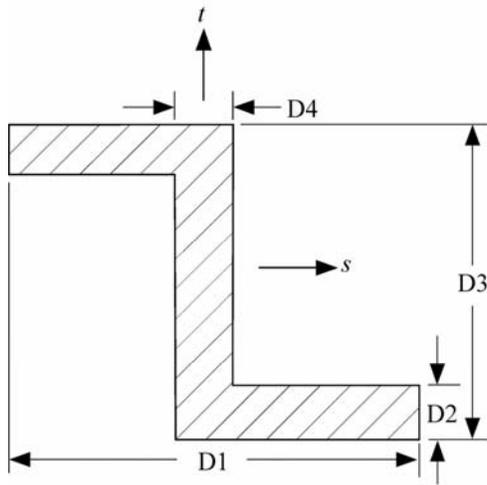
k = Integration refinement parameter (an integer GE. 0)

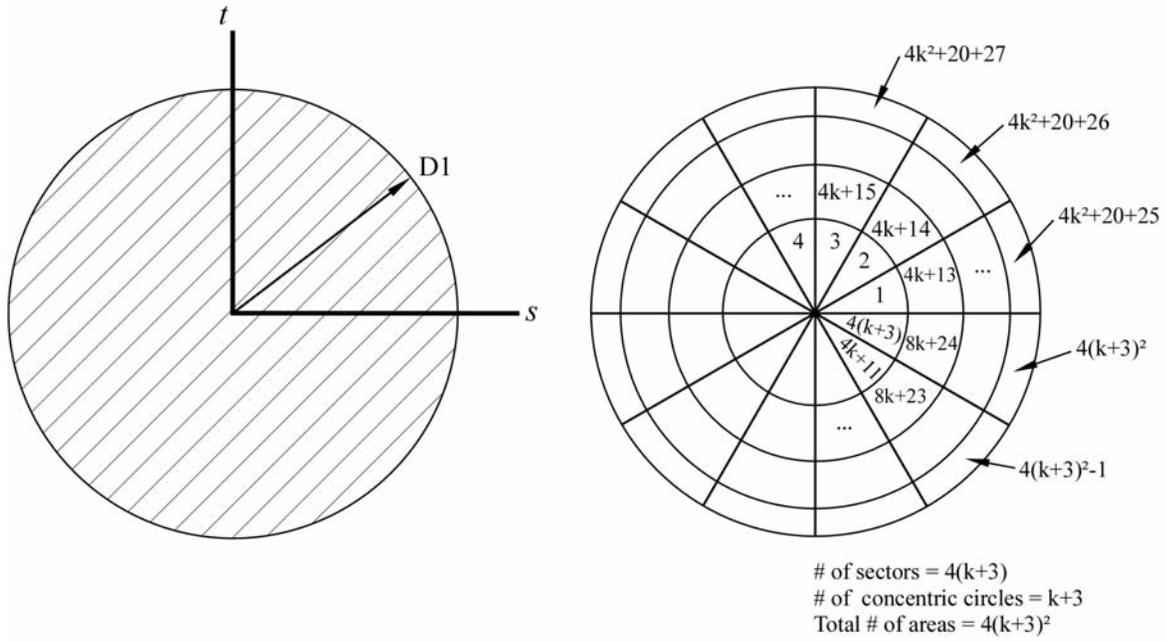
s_{ref} = location of reference surface normal to s, Hughes-Liu beam only

t_{ref} = location of reference surface normal to t, Hughes-Liu beam only

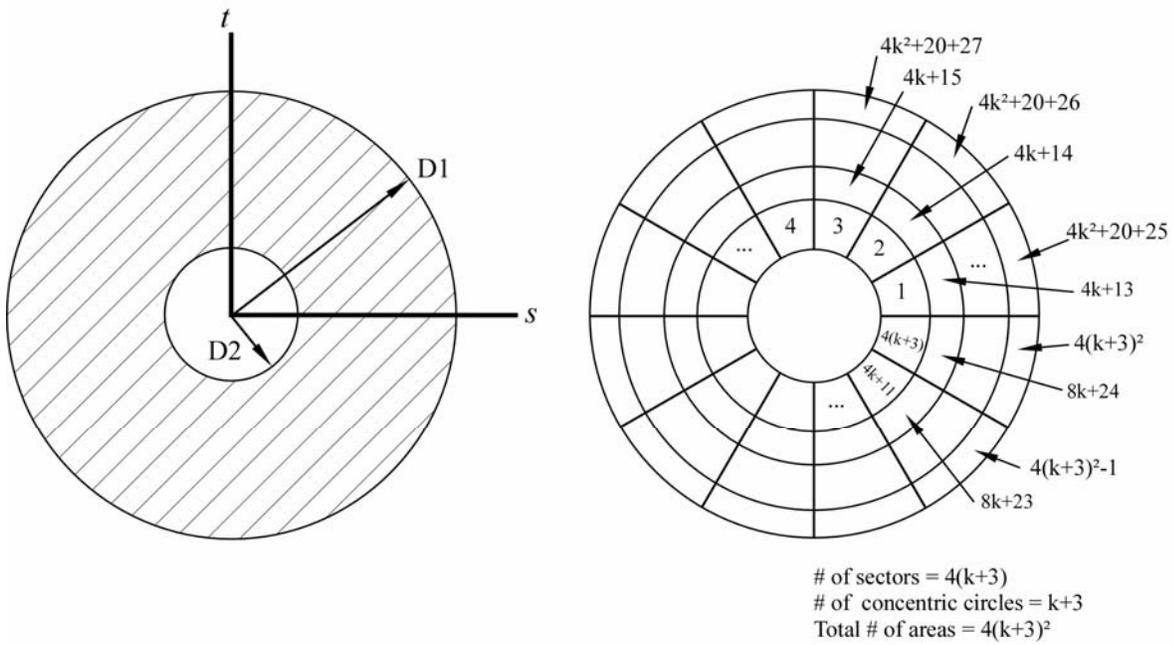




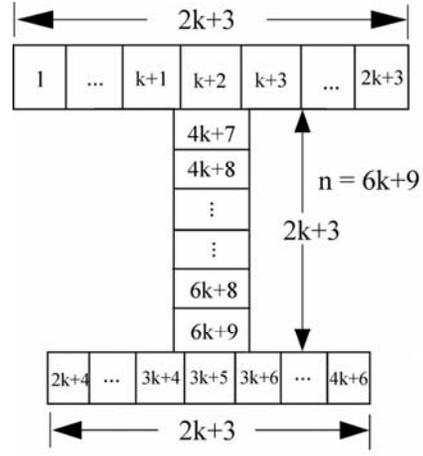
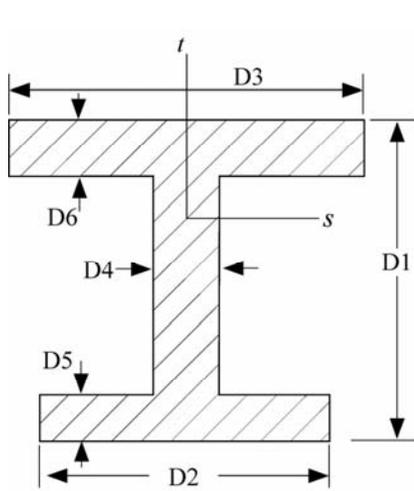




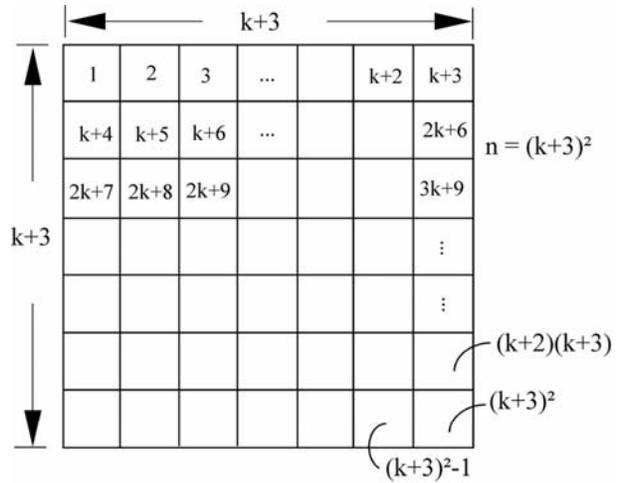
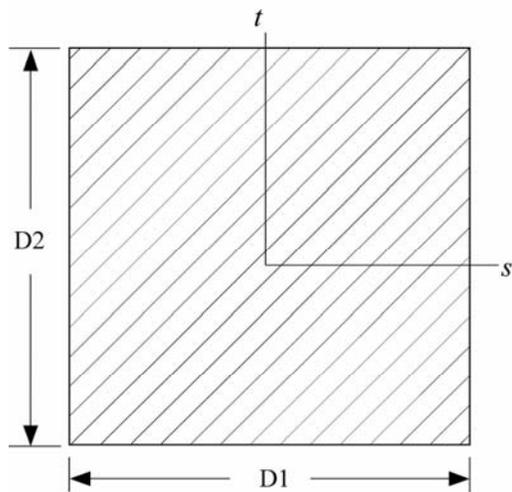
Type 8: Circular



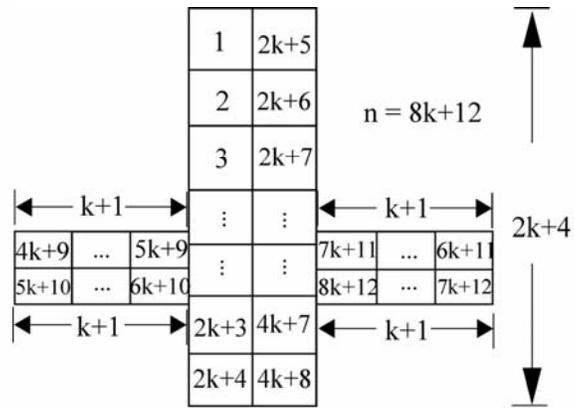
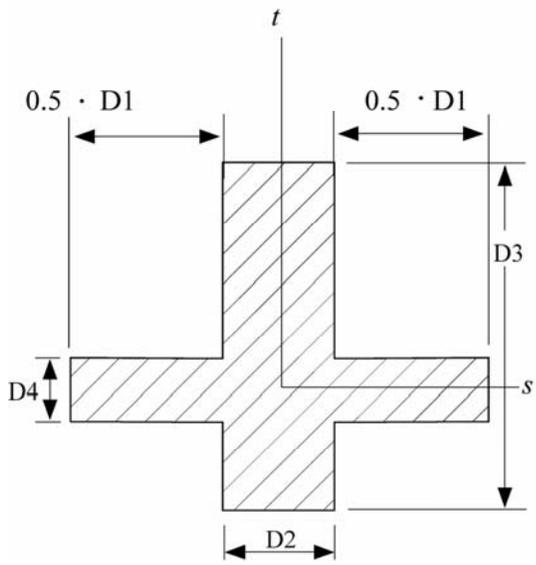
Type 9: Tubular



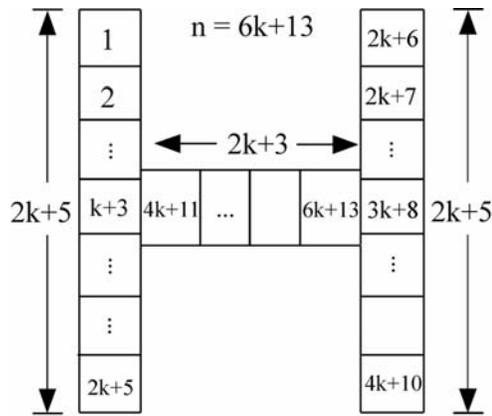
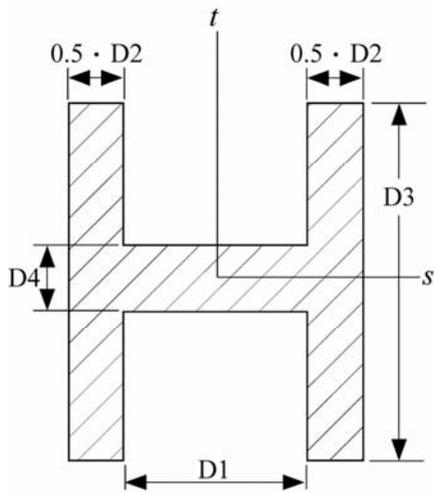
Type 10: I-shape1



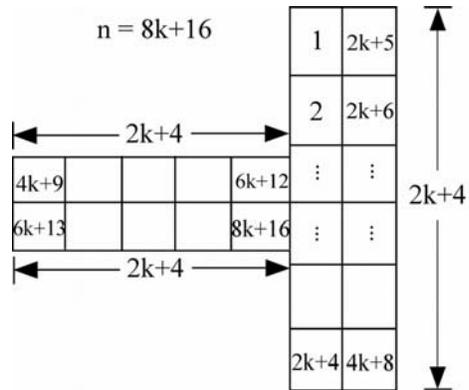
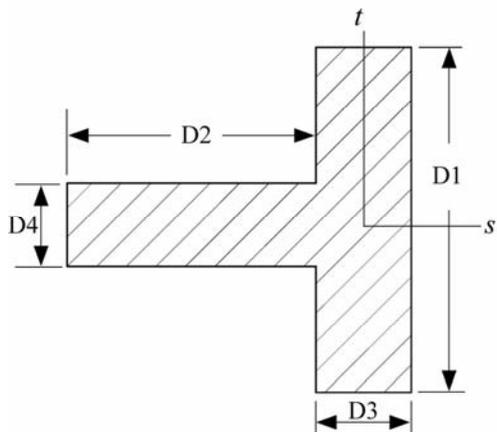
Type 11: Solid Box



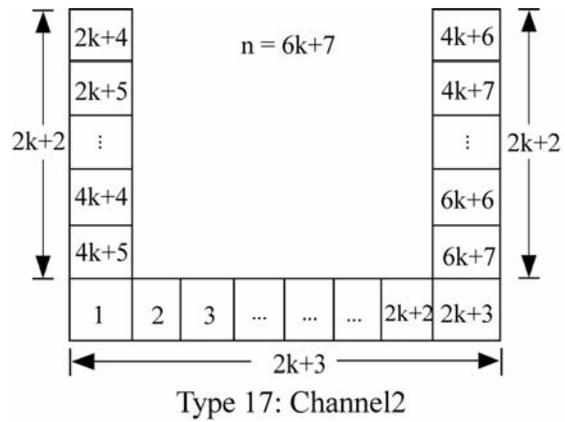
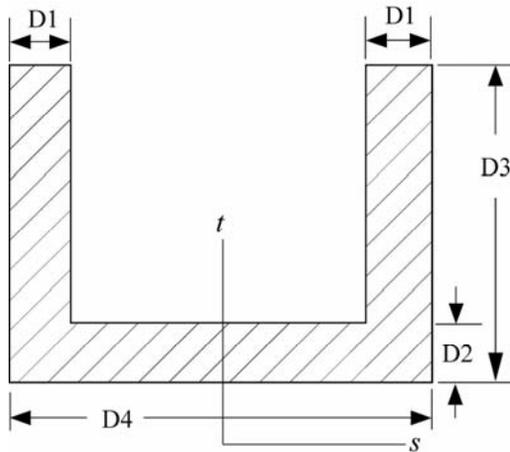
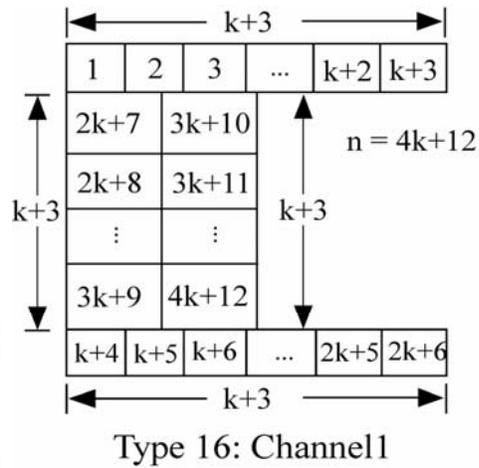
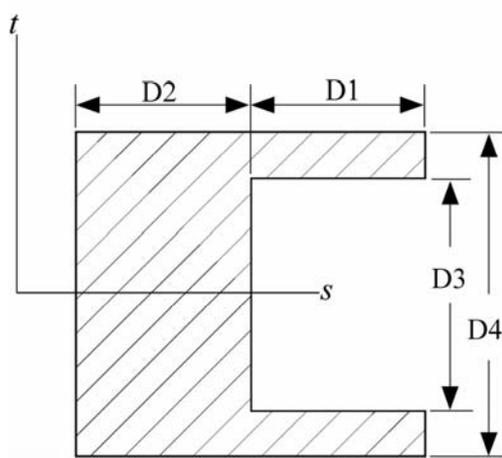
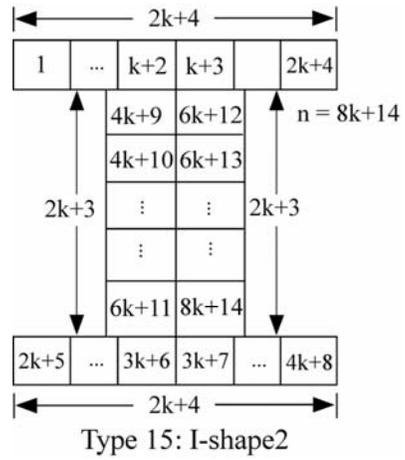
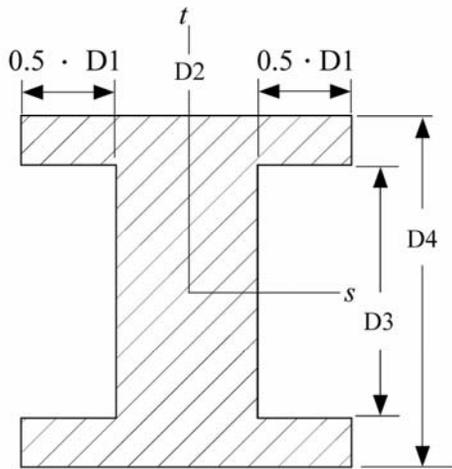
Type 12: Cross

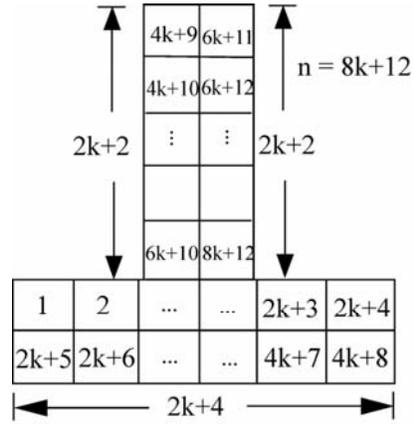
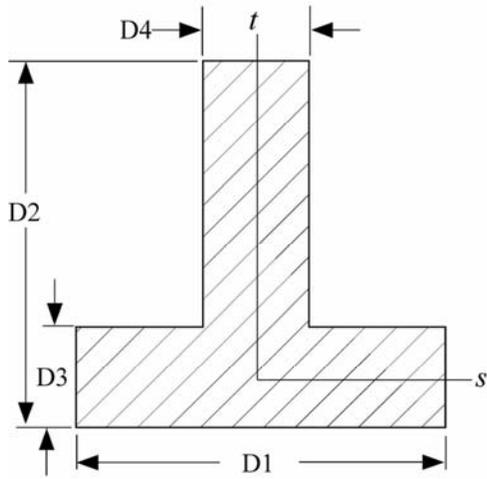


Type 13: H-shape

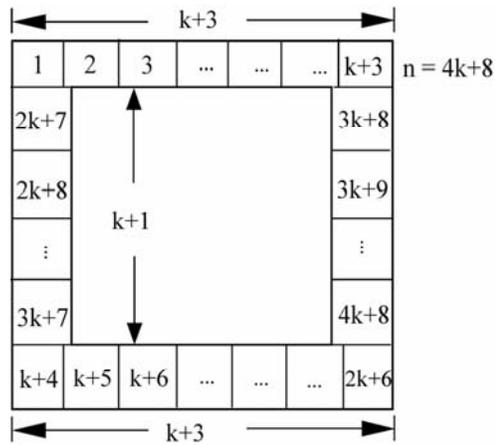
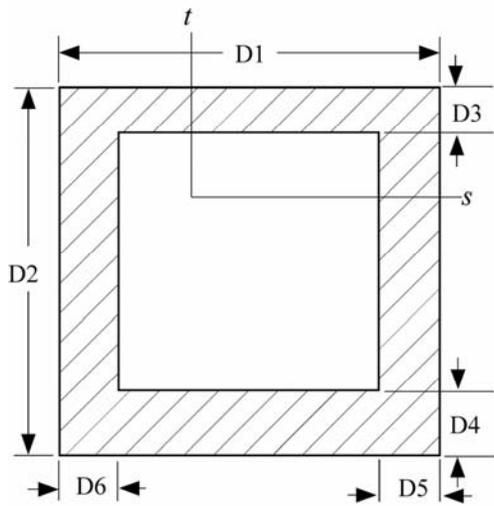


Type 14: T-shape 1

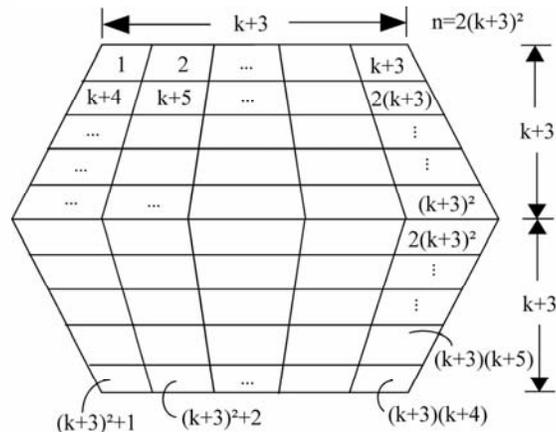
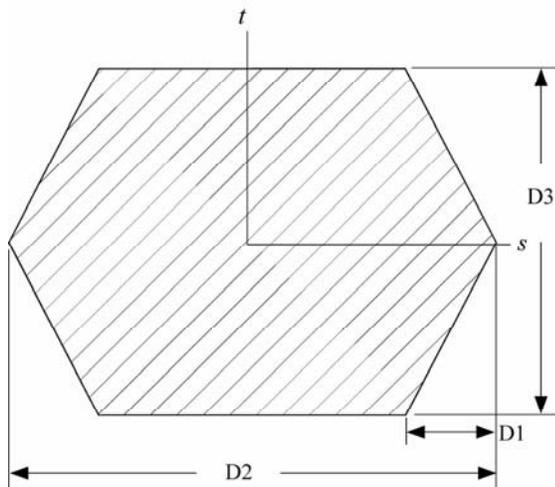




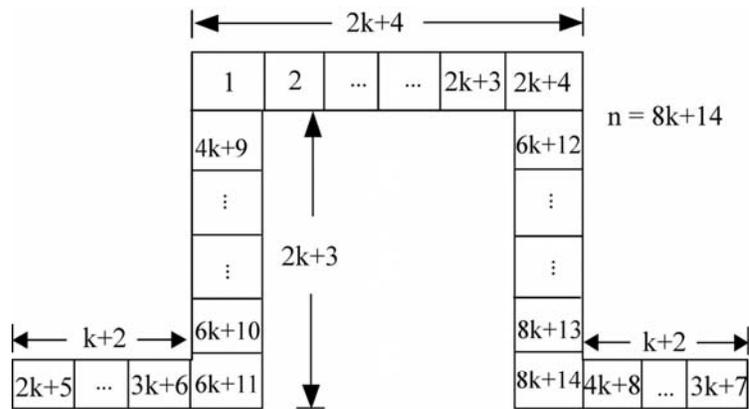
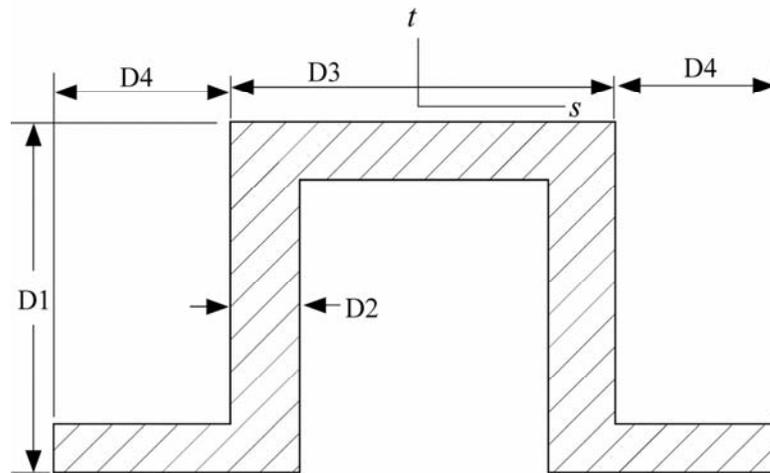
Type 18: T-shape2



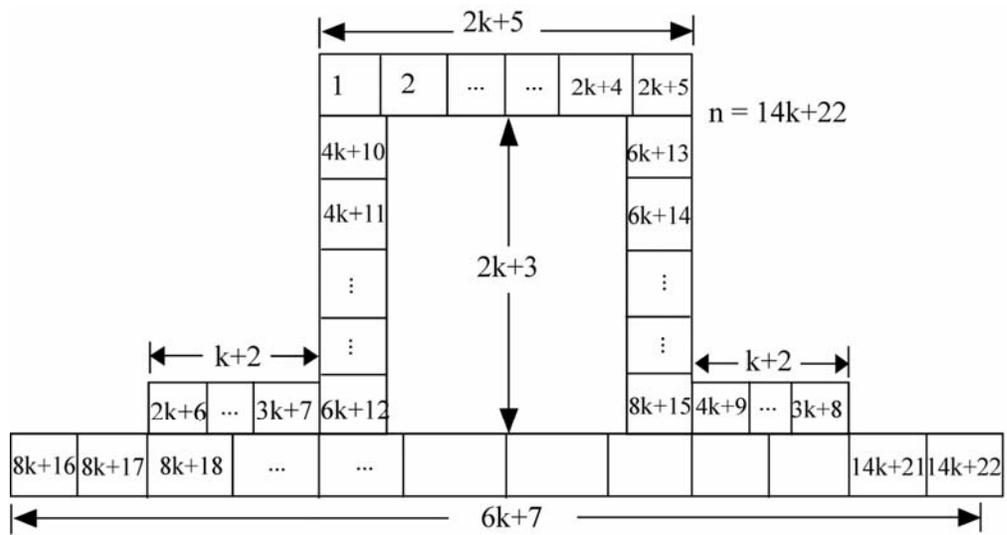
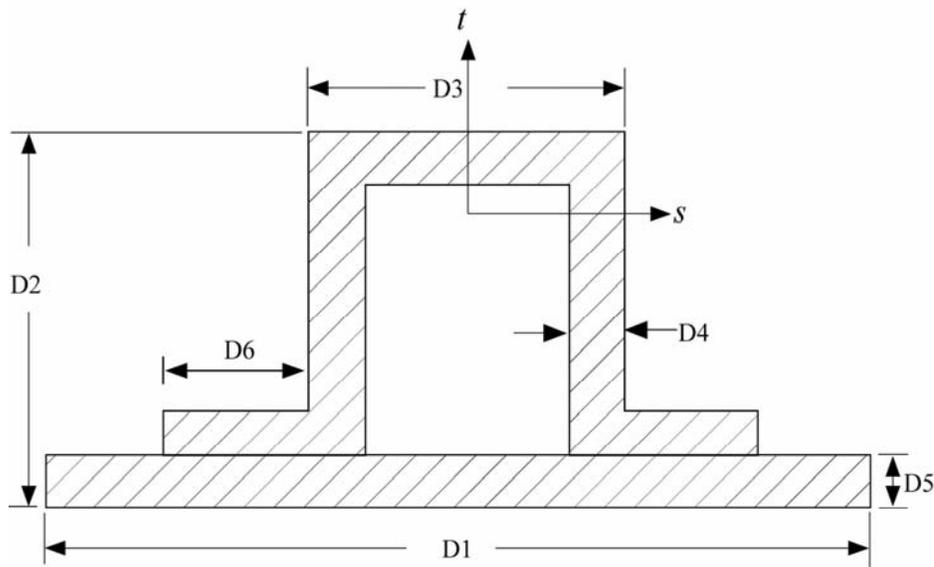
Type 19: Box-shape1



Type 20: Hexagon



Type 21: Hat-shape



Type 22: Hat-shape1

Figure 19.3. Standard beam cross sections.

***INTEGRATION_SHELL**

Purpose: Define user defined through the thickness integration rules for the shell element. This option applies to three dimensional shell elements with three or four nodes (ELEMENT_SHELL types 1-11 and 16) and to the eight node thick shell (ELEMENT_TSHELL).

Card 1 1 2 3 4 5 6 7 8

Variable	IRID	NIP	ESOP	FAILOPT					
Type	I	I	I	I					

Define NIP cards below if ESOP = 0.

Card 1 2 3 4 5 6 7 8

Variable	S	WF	PID						
Type	F	F	I						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IRID	Integration rule ID (IRID refers to IRID on *SECTION_SHELL card).
NIP	Number of integration points
ESOP	Equal spacing of integration points option: EQ.0: integration points are defined below, EQ.1: integration points are equally spaced through thickness such that the shell is subdivided into NIP layers of equal thickness.
FAILOPT	Treatment of failure when mixing different constitutive types, which do and do not include failure models, through the shell thickness. For example, consider the case where a linear viscoelastic material model, which does not have a failure option, is mixed with a composite model, which does have a failure option. Note: If the failure option includes failure based on the time step size of the element, element deletion will occur regardless of the value of FAILOPT. EQ.0: Element is deleted when the layers which include failure, fail. EQ.1: Element failure cannot occur since some layers do not have a failure option.

VARIABLE	DESCRIPTION
S	Coordinate of integration point in range -1 to 1.
WF	Weighting factor. This is typically the thickness associated with the integration point divided by actual shell thickness, i.e., the weighting factor for the i th integration point = $\frac{\Delta t_i}{t}$ as seen in Figure 19.4.
PID	Optional part ID if different from the PID specified on the element card. The average mass density for the shell element is based on a weighted average of the density of each layer that is used through the thickness. When modifying the constitutive constants through the thickness, it is often necessary to defined unique part IDs without elements that are referenced only by the user integration rule. These additional part IDs only provide a density and constitutive constants with local material axes (if used) and orientation angles taken from the PID referenced on the element card. In defining a PID for an integration point, it is okay to reference a solid element PID. The material type through the thickness can change as long as the material type is not hyperelastic (rubber).

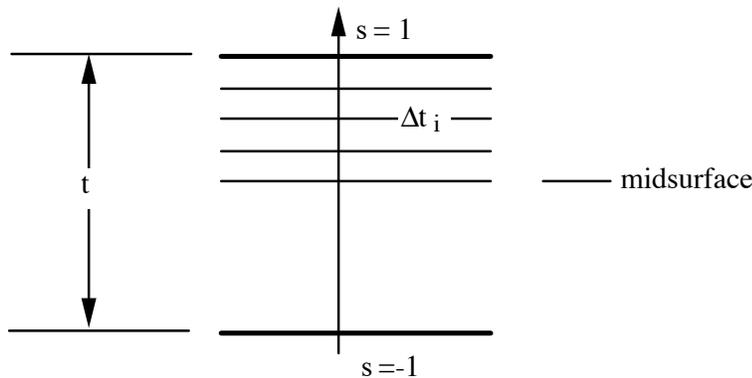


Figure 19.4. In the user defined shell integration rule the ordering of the integration points is arbitrary.

***INTERFACE**

Interface definitions are used to define surfaces, nodal lines, and nodal points for which the displacement and velocity time histories are saved at some user specified frequency. This data may then be used in subsequent analyses as an interface ID in the ***INTERFACE_LINKING_DISCRETE_NODE** as master nodes, in ***INTERFACE_LINKING_SEGMENT** as master segments and in ***INTERFACE_LINKING_EDGE** as the master edge for a series of nodes. This capability is especially useful for studying the detailed response of a small member in a large structure. For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized in the region bounded by the interfaces. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest. When beginning the first analysis, specify a name for the interface segment file using the **Z=**parameter on the LS-DYNA execution line. When starting the second analysis, the name of the interface segment file created in the first run should be specified using the **L=**parameter on the LS-DYNA command line. Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA's analysis capabilities. The keyword cards in this section are:

- *INTERFACE_COMPENSATION_NEW**
- *INTERFACE_COMPONENT_OPTION**
- *INTERFACE_JOY**
- *INTERFACE_LINKING_DISCRETE_NODE_OPTION**
- *INTERFACE_LINKING_EDGE**
- *INTERFACE_LINKING_SEGMENT**
- *INTERFACE_SPRINGBACK_OPTION1_OPTION2**

*INTERFACE

*INTERFACE_COMPENSATION_NEW

*INTERFACE_COMPENSATION_NEW

Purpose: The current method is used to compensate springback behavior. The capabilities of this feature include: (1) To calculate the deviation of the part from its intended design of the stamped part, and automatically compensate the tool to minimize the deviation, (2) the trimming curve can also be modified after the die modification to make sure that the trimming curve follows the modified tool, and (3) automatically detect the undercut problem.

This method is a nonlinear method. If one iteration is not enough to bring down the deviation to less than the acceptable tolerance, it is always advised to use more iterations. Usually, it is found that 2-4 iterations are needed for most of the cases. In addition, this method provides a scale factor, which allows the user to decide the ratio of shape deviation the part is compensated.

Limitation of the current method involves deficiency in handling (eliminating) the undercut problem.

All the target and current geometry must be included by using the keyword:

***INCLUDE_COMPENSATION_OPTION.**

Card	1	2	3	4	5	6	7	8
Variable	METHOD	SL	SF	ELREF	PSIDm	UNDCT		
Type	I	F	F	I	F	F		
Default	6	5.0	0.75	1				

VARIABLE

DESCRIPTION

METHOD	There are seven extrapolation methods. See Remark 1.
SL	The smooth level parameter controls the smoothness of the modified surfaces. A large value makes the surface smoother. The commonly used value is between 5 and 10. If springback is large, the transition region is expected to be large. However, by using a smaller value of SL, the region of transition can be reduced.
SF	This scales how much of the shape deviation is compensated. For example, if 10 mm springback is predicted, and the scale factor is chosen as 0.75, then the compensation in the opposite direction will only be 7.5 mm.

VARIABLE	DESCRIPTION
	<p>Through many parameter studies, it is found that the best scale factor is case dependent. For some cases, a scale factor of 0.75 is best, while for others, larger values are better. Sometimes, the best value can be larger than 1.1.</p> <p>Since it is impossible to choose the best value for each application it is suggested that for a new application, the initial trial is 0.75. If the springback cannot be effectively compensated, more iterations must be used to compensate the remaining shape deviation.</p> <p>For channel with twisting, the scale factor is more important. It was found that a small change of the tool shape might change the twisting mode. If this occurs, using a small value (<0.5) is suggested.</p>
ELREF	EQ.1: special element refinement is used with the tool elements (default) EQ.2: special element refinement is turned off
PSIDm	Define the Part set ID for master parts. It is important to properly choose the parts for the master side. Usually, only one side (master side) of the tool will be chosen as the master side, and the modification of the other side (slave side) depends solely on the change, which occurs in the master side. In this way, the two sides are coupled and a constant gap between the two sides is maintained. If both sides are chosen as master side, the gap between the two sides might change and the gap might become inhomogeneous. <p>The choice of Master side will have effect on the final result for method 7 for three-piece draw. At this time, when the punch and binder are chosen as the master side, the binder region will not be changed. Otherwise, when the die is chosen as Master side the binder will be changed, since the changes extend to the edges of the Master tool.</p>
UNDCt	EQ.0: Default EQ.1: Check and fix undercut

Remarks:

1. After trimming, only a limited part of the tool has direct relationship with the springback of the blank part. The modification of the rigid tool outside the trimming curve has to rely on extrapolation. However, extrapolating is an unstable process, it is easy to have a non-smooth surface. To resolve this problem, seven smoothing algorithms have been proposed. The frequently used methods are: 7, 3, and 6. The others are used occasionally.

Method 7: If the punch is chosen as the master side, the binder will not be changed. The only change occurring involves inside punch opening. Under this option, the smoothing

factor has little effect. The smoothness of the modified tool depends on the magnitude of the springback and the size of the addendum region.

Advantages: The binder will not be changed.

Disadvantages: The change will be limited inside the addendum region, and the modified surface may not be smooth if the springback magnitude is large and the transition is small. This is a non-linear method, and the iterative method is used.

Method 6: The smoothness and the transition region of the modified surface will depend on the springback magnitude and the smoothing factor. If the springback magnitude is large, the transition region will be increased automatically. On the other hand, the transition region will be smaller if the springback magnitude is small. At the same time, a larger smoothing factor will result in a smaller transition region.

Advantages: The smoothness of the modified surfaces can be controlled. This is a non-linear method and the iterative method is used.

Disadvantages: It is impossible to limit the transition region, and the binder surface could change if the springback is large.

Method 3: Similar to Method 6, however, it is a linear method and no iteration is necessary. The other options may be removed in the future; therefore they will not be discussed.

Method used to Prevent Undercut

When the wall is steep, it is very possible that undercut will happen. Since undercut is not accepted in real die manufacturing, it is necessary to prevent it from happening.

The code can automatically detect undercut and issue a warning message. In addition, it will save all the element information into a file so that the user can easily know which elements may be problematic.

If the undercut is only limited to a few elements, it is possible to fix the problem manually. The code provides one more option to handle undercut problem, i.e. to compensate the springback only in the punch moving direction (by using a negative scale factor). It is known that this method is not the premium method to handle undercut problem, and we are still working on it and try to find a better solution.

***INTERFACE_COMPONENT_OPTION**

Available options include:

NODE

SEGMENT

Purpose: Define an interface for linking calculations. This card applies to the first analysis for storing interfaces in the file specified by Z=isf1 on the execution command line. The output interval used to write data to the interface file is controlled by OPIFS on *CONTROL_OUTPUT.

This capability allows the definition of interfaces that isolate critical components. A database is created that records the motion of the interfaces. In later calculations the isolated components can be reanalyzed with arbitrarily refined meshes with the motion of their boundaries specified by the database created by this input. The interfaces defined here become the masters in the tied interface options.

Each definition consists of a set of cards that define the interface. Interfaces may consists of a set of four node segments for moving interfaces of solid elements, a line of nodes for treating interfaces of shells, or a single node for treating beam and spring elements.

Card 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							

VARIABLE

DESCRIPTION

SID

Set ID, see *SET_NODE or *SET_SEGMENT.

***INTERFACE**

***INTERFACE_JOY**

***INTERFACE_JOY**

Purpose: Define an interface for linking calculations by moving a nodal interface.

Card 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							

VARIABLE

DESCRIPTION

SID

Nodal set ID, see *SET_NODE.

***INTERFACE_LINKING_DISCRETE_NODE**

***INTERFACE**

***INTERFACE_LINKING_DISCRETE_NODE_OPTION**

Available options include:

NODE

SET

Purpose: Define an interface for linking discrete nodes to an interface file. This link applies to spring and beam elements only.

Card 1 2 3 4 5 6 7 8

Variable	NID/NSID	IFID						
Type	I	I						

VARIABLE

DESCRIPTION

NID Node ID or Node set ID to be moved by interface file, see *NODE or *SET_NODE.

IFID Interface ID in interface file.

***INTERFACE**

***INTERFACE_LINKING_EDGE**

***INTERFACE_LINKING_EDGE**

Purpose: Define an interface for linking a series of nodes in shell structure to an interface file for the second analysis using L=isf2 on the execution command line. This link applies segments on shell elements only.

Card	1	2	3	4	5	6	7	8
Variable	NSID	IFID						
Type	I	I						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NSID	Node set ID to be moved by interface file.
IFID	Interface ID in interface file.

INTERFACE_LINKING_SEGMENT**INTERFACE*****INTERFACE_LINKING_SEGMENT**

Purpose: Define an interface for linking segments to an interface file for the second analysis using L=isf2 on the execution command line. This applies segments on shell and solid elements.

Card 1 2 3 4 5 6 7 8

Variable	SSID	IFID						
Type	I	I						

VARIABLE**DESCRIPTION**

SSID	Segment set to be moved by interface file.
IFID	Interface ID in interface file.

*INTERFACE

*INTERFACE_SPRINGBACK

*INTERFACE_SPRINGBACK_OPTION1_OPTION2

Available options included for *OPTION1* are:

LSDYNA

NASTRAN

SEAMLESS

and for *OPTION2*:

THICKNESS

NOTHICKNESS

See the remarks below.

Purpose: Define a material subset for an implicit springback calculation in LS-DYNA and any nodal constraints to eliminate rigid body degrees-of-freedom.

Card	1	2	3	4	5	6	7	8
Variable	PSID	NSHV	FTYPE		FTENSR			
Type	I	I	I		I			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PSID	Part set ID for springback, see *SET_PART.
NSHV	Number of shell or solid history variables (beyond the six stresses and effective plastic strain) to be initialized in the interface file. For solids, one additional state variable (initial volume) is also written. If NSHV is nonzero, the element formulations, calculational units, and constitutive models should not change between runs. If NHSV exceeds the number of integration point history variables required by the constitutive model, only the number required is written; therefore, if in doubt, set NHSV to a large number.
FTYPE	File type: EQ.0: ASCII, EQ.1: binary EQ.2: both ASCII and binary.

VARIABLE	DESCRIPTION
FTENSR	Flag for dumping tensor data from the element history variables into the dynain file. EQ.0: Don't dump tensor data from element history variables EQ.1: Dump any tensor data from element history variables into the dynain file in GLOBAL coordinate system Currently, only Material 190 supports this option.

Define a list of nodal points that are constrained for the springback. This section is terminated by an "*" indicating the next input section.

Card	1	2	3	4	5	6	7	8
Variable	NID	TC	RC					
Type	I	F	F					

VARIABLE	DESCRIPTION
NID	Node ID, see *NODE.
TC	Translational Constraint: EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements. EQ.7: constrained x, y, and z displacements.
RC	Rotational constraint: EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

Remarks:

1. The NOTHICKNESS option is available for LS-DYNA and NASTRAN in which case the shell element thickness is not an output. The file name for LS-DYNA is “dynain” and for NASTRAN is “nastin.” The *CONTROL_ADAPTIVITY is available for LS-DYNA.
2. Trimming is available for the adaptive mesh but it requires some steps. To trim an adaptive mesh use the following procedure:
 - (1) Generate the file, “dynain” using the keyword *INTERFACE_SPRINGBACK_LSDYNA.
 - (2) Prepare a new input deck including the file “dynain.”
 - (3) Add the keyword *ELEMENT_TRIM to this new deck.
 - (4) Add the keyword *DEFINE_CURVE_TRIM to this new deck.
 - (5) Run this new input deck with i=input_file_name. The adaptive constraints are eliminated by remeshing and the trimming is performed.
 - (6) In case this new trimmed mesh is needed, run a zero termination time job and output the file generated via the keyword, *INTERFACE_SPRINGBACK_LSDYNA.
3. Temperature – The file new_temp_ic.inc will be created for a thermal solution and a coupled thermal-mechanical solution. The file new_temp_ic.inc is a KEYWORD include file which contains “new temperature initial conditions” for the nodes belonging to the PSID.

Remarks for Seamless Springback:

In seamless springback LS-DYNA automatically and seamlessly switches from explicit or implicit dynamic to implicit static mode at the end of a forming simulation, and continues to run the static springback analysis. Seamless springback can be activated in the original LS-DYNA input deck, or later using a small restart input deck. In this way, the user can decide to continue a previous forming analysis by restarting to add the implicit springback phase. (Another alternative approach to springback simulation is to use the keyword *INTERFACE_SPRINGBACK_LSDYNA to generate a "dynain" file after forming, and then perform a second simulation running LS-DYNA in fully implicit mode for springback. See Appendix P for a description of how to run an implicit analysis using LS-DYNA.

The implicit springback phase begins when the forming simulation termination time ENDTIM is reached, as specified with the keyword *CONTROL_TERMINATION. Since the springback phase is static, its termination time can be chosen arbitrarily (unless material rate effects are included). The default choice is 2.0*ENDTIM, and can be changed using the *CONTROL_IMPLICIT_GENERAL keyword.

Since the springback analysis is a static simulation, a minimum number of essential boundary conditions or Single Point Constraints (SPC's) can be input to prohibit rigid body motion of the part. These boundary conditions can be added for the springback phase using the input option on the *INTERFACE_SPRINGBACK_SEAMLESS keyword above.

If no boundary conditions are added with the SEAMLESS option an eigenvalue computation is automatically performed using the Inertia Relief Option to find any rigid body modes and then automatically constrain them out of the springback simulation (see *CONTROL_IMPLICIT_INERTIA_RELIEF). This approach introduces no artificial deformation and is recommended for many simulations.

Several new *CONTROL_IMPLICIT keywords have been added to control the implicit springback phase. These keywords can also be added to a restart input deck. Generally, default settings can be used, so these keywords need not be included in the input deck.

To obtain accurate springback solutions, a nonlinear springback analysis must be performed. In many simulations, this iterative equilibrium search will converge without difficulty. If the springback simulation is particularly difficult, either due to nonlinear deformation, nonlinear material response, or numerical precision errors, a multi-step springback simulation will be automatically invoked. In this approach, the springback deformation is divided into several smaller, more manageable steps.

Two specialized features in LS-DYNA are used to perform multi-step springback analyses. The addition and gradual removal of artificial springs is performed by the artificial stabilization feature. Simultaneously, the automatic time step control is used to guide the solution to the termination time as quickly as possible, and to persistently retry steps where the equilibrium search has failed. By default, both of these features are active during a seamless springback simulation. However, the default method attempts to solve the springback problem in a single step. If this is successful, the solution will terminate normally. If the single step springback analysis fails to converge, the step size will be reduced, and artificial stabilization will become active. Defaults for these features can be changed using the following keywords: *CONTROL_IMPLICIT_GENERAL, *CONTROL_IMPLICIT_AUTO, and *CONTROL_IMPLICIT_STABILIZATION.

***INTERFACE**

***INTERFACE_SPRINGBACK**

***KEYWORD**

***KEYWORD**_{OPTION} {MEMORY} {NCPU=n}

Available options include:

<BLANK>

ID

Purpose: The keyword, *KEYWORD, flags LS-DYNA that the input deck is a keyword deck rather than the structured format, which has a strictly defined format. This must be the first card in the input file. Alternatively, by typing “keyword” on the execution line, keyword input formats are assumed and this beginning “*KEYWORD” line is not required.

There are 2 optional parameters that can be specified with the *KEYWORD which can be listed in any order. If a number {MEMORY} is specified after the word *KEYWORD, it defines the memory size to be used in words. The memory size can also be set on the command line. Note that the memory size specified on the *KEYWORD card is overridden by the memory specified on the execution line. If the parameter {NCPU=n} is specified it defines the number of CPUs “n” to be used during the analysis. This only applies to the Shared Memory Parallel (SMP) version of LS-DYNA. For the Distributed Memory Version (MPP) the number of CPUs is always defined with the “mpirun” command. Defining the number of CPUs on the execution line overrides what is specified with the *KEYWORD command and both override the number of CPUs specified by *CONTROL_PARALLEL.

One optional card is available for *KEYWORD for naming file names with a prefix. This option allows for multiple simulations in a directory, since the prefix is attached to all output and scratch filenames, i.e., not the input filenames.

Optional if the ID option is active.

Card 1 1 2 3 4 5 6 7 8

Variable	PROJECT	NUM	STAGE
Type	A	A	A
Default	none	none	none
Remarks	see below		

***KEYWORD**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PROJECT	First part of the file name prefix.
NUM	Second part of the file name prefix.
STAGE	Third part of the file name prefix.

Remarks:

1. An example, the prefix with the file name given on the optional card for file D3PLOT would be the following:

PROJECT_NUM_STAGE.D3PLOT

The prefix can also be assigned by the option, jobid on the execution line. The execution line would look like the following:

ls971 i=input.k jobid= PROJECT_NUM_STAGE

2. An example of the {MEMORY} and {NCPU=n} options would be the follows:

*KEYWORD 12000000 NCPU=2

The *KEYWORD command is requesting 12 million words of memory and 2 CPUs to be used for the analysis.

***LOAD**

The keyword ***LOAD** provides a way of defining applied forces. The keyword control cards in this section are defined in alphabetical order:

- *LOAD_ALE_CONVECTION_{OPTION}**
- *LOAD_BEAM_OPTION**
- *LOAD_BLAST**
- *LOAD_BODY_OPTION**
- *LOAD_BODY_GENERALIZED**
- *LOAD_BODY_POROUS**
- *LOAD_BRODE**
- *LOAD_DENSITY_DEPTH**
- *LOAD_GRAVITY_PART**
- *LOAD_HEAT_CONTROLLER**
- *LOAD_HEAT_GENERATION_OPTION**
- *LOAD_MASK**
- *LOAD_MOTION_NODE**
- *LOAD_MOVING_PRESSURE**
- *LOAD_NODE_OPTION**
- *LOAD_REMOVE_PART**
- *LOAD_RIGID_BODY**
- *LOAD_SEGMENT_{OPTION}**
- *LOAD_SEGMENT_NONUNIFORM_{OPTION}**
- *LOAD_SEGMENT_SET_{OPTION}**
- *LOAD_SEGMENT_SET_NONUNIFORM_{OPTION}**
- *LOAD_SHELL_{OPTION1}_{OPTION2}**
- *LOAD_SSA**
- *LOAD_STIFFEN_PART**
- *LOAD_SUPERPLASTIC_FORMING**
- *LOAD_SURFACE_STRESS_OPTION**
- *LOAD_THERMAL_OPTION**
- *LOAD_THERMAL_CONSTANT**
- *LOAD_THERMAL_CONSTANT_NODE**

***LOAD**

***LOAD_THERMAL_LOAD_CURVE**

***LOAD_THERMAL_TOPAZ**

***LOAD_THERMAL_VARIABLE**

***LOAD_THERMAL_VARIABLE_NODE**

***LOAD_THERMAL_VARIABLE_SHELL_{*OPTION*}**

***LOAD_VOLUME_LOSS**

***LOAD_ALE_CONVECTION_{OPTION}**

Purpose: This card is used to define the convection thermal energy transfer from a hot ALE fluid to the surrounding Lagrangian structure (remark 1). It is associated with a corresponding coupling card defining the interaction between the ALE fluid and the Lagrangian structure. It is only used when thermal energy transfer from the ALE fluid to the surrounding Lagrangian structure is significant. This is designed specifically for airbag deployment application where the heat transfer from the inflator gas to the inflator compartment can significantly affect the inflation potential of the inflator.

Available options include:

<BLANK>

ID

To define an ID number for each convection heat transfer computation in an optional card preceding all other cards for this command. This ID number can be used to output the part temperature and temperature change as functions of time in the *DATABASE_FSI card. To do this, set the CONVID parameter in the *DATABASE_FSI card equal to this ID.

Card	1	2	3	4	5	6	7	8
Variable	LAGPID	LAGT	LAGCP	H	LAGMAS			
Type	I	F	F	F	F			
Default	none	none	none	none	none			
Remarks								

VARIABLE	DESCRIPTION
LAGPID	Lagrangian PID (slave PID) from a corresponding coupling card which receives the thermal energy in the convection heat transfer.
LAGT	Initial temperature of this Lagrangian slave part.
LAGCP	Constant-pressure heat capacity of this Lagrangian slave part. It has a per-mass unit (for example, J/[Kg*K]).

VARIABLE	DESCRIPTION
H	Convection heat transfer coefficient on this Lagrangian slave part surface. It is the amount of energy (J) transferred per unit area, per time, and per temperature difference. For example, its units may be J/[m ² *s*K]
LAGMAS	The mass of the Lagrangian slave part receiving the thermal energy. This is in absolute mass unit (for example, Kg).

Remarks:

1. The only application of this card so far has been for the transfer of thermal energy from the ALE hot inflator gas to the surrounding Lagrangian structure (inflator canister and airbag-containing compartment) in an airbag deployment model.
2. The heat transferred is taken out of the inflator gas thermal energy thus reducing its inflating potential.
3. This is not a precise heat transfer modeling attempt. It is simply one mechanism for taking out excessive energy from the inflating potential of the hot inflator gas.
4. The heat transfer formulation may roughly be represented as following. Some representative units are shown just for clarity.

$$\dot{Q} = H * A * \Delta T \sim Watt \sim \frac{J}{s} \sim \left[\frac{J}{m^2 * s * K} \right] * m^2 * K$$

$$\dot{Q} * dt = \dot{M} * C_p * [T_{Lag_new} - T_{Lag_orig}] \sim \frac{kg}{s} * \left[\frac{J}{kg * K} \right] * K \sim \frac{J}{s}$$

***LOAD_BEAM_OPTION**

Available options include:

ELEMENT

SET

Purpose: Apply the distributed traction load along any local axis of beam or a set of beams. The local axes are defined in Figure 22.1, see also *ELEMENT_BEAM.

Card	1	2	3	4	5	6	7	8
Variable	EID/ESID	DAL	LCID	SF				
Type	I	I	I	F				
Default	none	none	none	1.				
Remarks								

VARIABLE	DESCRIPTION
EID/ESID	Beam ID (EID) or beam set ID (ESID), see *ELEMENT_BEAM or *SET_BEAM.
DAL	Direction of applied load: EQ.1: along r-axis of beam, EQ.2: along s-axis of beam, EQ.3: along t-axis of beam.
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor. This is for a simple modification of the function values of the load curve.

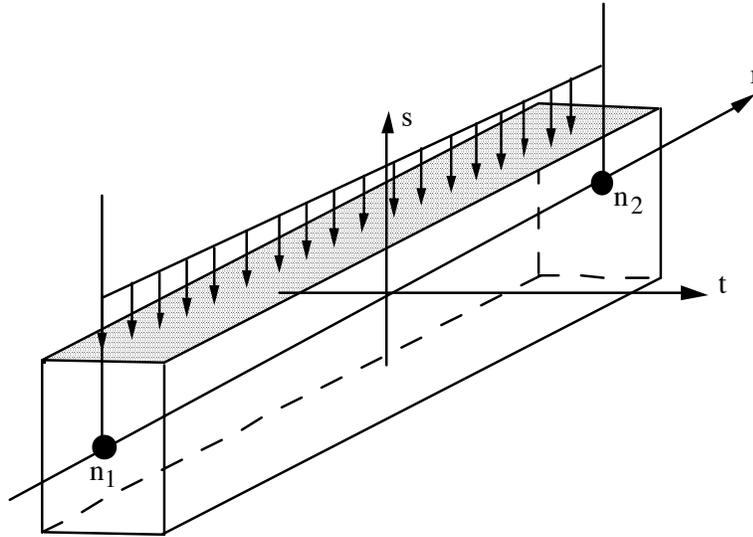


Figure 22.1. Applied traction loads are given in force per unit length. The s and t directions are defined on the *ELEMENT_BEAM keyword.

***LOAD_BLAST**

Purpose: Define an airblast function for the application of pressure loads due to explosives in conventional weapons. The implementation is based on a report by Randers-Pehrson and Bannister [1997] where it is mentioned that this model is adequate for use in engineering studies of vehicle responses due to the blast from land mines. This option determines the pressure values when used in conjunction with the keywords: *LOAD_SEGMENT, *LOAD_SEGMENT_SET, or *LOAD_SHELL.

Card 1 1 2 3 4 5 6 7 8

Variable	WGT	XBO	YBO	ZBO	TBO	IUNIT	ISURF	
Type	F	F	F	F	F	I	I	
Default	none	0.0	0.0	0.0	0.0	2	2	

Card 2

Variable	CFM	CFL	CFT	CFP				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

VARIABLE**DESCRIPTION**

WGT	Equivalent mass of TNT.
XBO	x-coordinate of point of explosion.
YBO	y-coordinate of point of explosion.
ZBO	z-coordinate of point of explosion.
TBO	Time-zero of explosion.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IUNIT	Unit conversion flag. EQ.1: feet, pound-mass, seconds, psi EQ.2: meters, kilograms, seconds, Pascals (default) EQ.3: inch, dozens of slugs, seconds, psi EQ.4: centimeters, grams, microseconds, Megabars EQ.5: user conversions will be supplied (see Card 2)
ISURF	Type of burst. EQ.1: surface burst - hemispherical charge situated on the surface EQ.2: air burst - spherical charge at least one charge diameter away from the surface (default)
CFM	Conversion factor - pounds per LS-DYNA mass unit.
CFL	Conversion factor - feet per LS-DYNA length units.
CFT	Conversion factor - milliseconds per LS-DYNA time unit.
CFP	Conversion factor - psi per LS-DYNA pressure unit.

Remarks:

1. A minimum of two load curves, even if unreferenced, must be present in the model.
2. Segment normals should point toward the charge.

***LOAD_BODY_OPTION**

Available options include for base accelerations:

X

Y

Z

for angular velocities:

RX

RY

RZ

and to specify a part set:

PARTS

Purpose: Define body force loads due to a prescribed base acceleration or angular velocity using global axes directions. This data applies to all nodes in the complete problem unless a part subset is specified via the *LOAD_BODY_PARTS keyword. If a part subset is defined then all nodal points belonging to the subset will have body forces applied. The parts specified via the *LOAD_BODY_PARTS keyword apply to the options X, Y, Z, RX, RY, and RZ above, i.e., different part sets may not apply to different options. Only one part set is expected. **Note: This option applies nodal forces, i.e., it cannot be used to prescribe translational or rotational motion.** Two keyword definitions are needed to apply body loads on a subset of parts: *LOAD_BODY_X and *LOAD_BODY_PARTS.

For options: X, Y, Z, RX, RY, and RZ.

Card 1 2 3 4 5 6 7 8

Variable	LCID	SF	LCIDDR	XC	YC	ZC	CID	
Type	I	F	I	F	F	F	I	
Default	none	1.	0	0.	0.	0.	0	

For option: PARTS.

Card 1 2 3 4 5 6 7 8

Variable	PSID							
Type	I							
Default	none							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor
LCIDDR	Load curve ID for dynamic relaxation phase (optional). This is only needed if dynamic relaxation is defined and a different load curve to LCID is required during the dynamic relaxation phase. Note if LCID is set to zero then no body load will be applied during dynamic relaxation regardless of the value LCIDDR is set to. See *CONTROL_DYNAMIC_RELAXATION
XC	X-center of rotation, define for angular velocities.
YC	Y-center of rotation, define for angular velocities.
ZC	Z-center of rotation, define for angular velocities.
CID	Coordinate system ID to define acceleration in local coordinate system. The accelerations (LCID) are with respect to CID. EQ.0: global
PSID	Part set ID.

Remarks:

1. Translational base accelerations allow body forces loads to be imposed on a structure. Conceptually, base acceleration may be thought of as accelerating the coordinate system in the direction specified, and, thus, the inertial loads acting on the model are of opposite sign. For example, if a cylinder were fixed to the y-z plan and extended in the positive x-direction, then a positive x-direction base acceleration would tend to shorten the cylinder, i.e., create forces acting in the negative x-direction.
2. Base accelerations are frequently used to impose gravitational loads during dynamic relaxation to initialize the stresses and displacements. During the analysis, in this latter

case, the body forces loads are held constant to simulate gravitational loads. When imposing loads during dynamic relaxation, it is recommended that the load curve slowly ramp up to avoid the excitation of a high frequency response.

3. Body force loads due to the angular velocity about an axis are calculated with respect to the deformed configuration and act radially outward from the axis of rotation. Torsional effects which arise from changes in angular velocity are neglected with this option. The angular velocity is assumed to have the units of radians per unit time.
4. The body force density is given at a point P of the body by:

$$b = \rho(\omega \times \omega \times r)$$

where ρ is the mass density, ω is the angular velocity vector, and r is a position vector from the origin to point P. Although the angular velocity may vary with time, the effects of angular acceleration are not included.

5. Angular velocities are useful for studying transient deformation of spinning three-dimensional objects. Typical applications have included stress initialization during dynamic relaxation where the initial rotational velocities are assigned at the completion of the initialization, and this option ceases to be active.

***LOAD_BODY_GENERALIZED**

Purpose: Define body force loads due to a prescribed base acceleration or a prescribed angular velocity over a subset of the complete problem. The subset is defined by using nodes. Warning: This option does not use the mass specified with *PART_INERTIA, which applies to rigid bodies, but instead uses the nodal mass. The preceding option, *LOAD_BODY, can be used with the *PART_INERTIA option.

Card 1 1 2 3 4 5 6 7 8

Variable	N1	N2	LCID	DRLCID	XC	YC	ZC	
Type	I	I	I	I	F	F	F	
Default	none	none	none	0	0.	0.	0.	
Remarks								

Card 2

Variable	AX	AY	AZ	OMX	OMY	OMZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		
Remarks	1, 2	1, 2	1, 2	3, 4, 5	3, 4, 5	3, 4, 5		

VARIABLE**DESCRIPTION**

N1	Beginning node ID for body force load.
N2	Ending node ID for body force load.
LCID	Load curve ID, see *DEFINE_CURVE.
DRLCID	Load curve ID for dynamic relaxation phase. Only necessary if dynamic relaxation is defined. See *CONTROL_DYNAMIC_RELAXATION.

VARIABLE	DESCRIPTION
XC	X-center of rotation. Define only for angular velocity.
YC	Y-center of rotation. Define only for angular velocity.
ZC	Z-center of rotation. Define only for angular velocity.
AX	Scale factor for acceleration in x-direction
AY	Scale factor for acceleration in y-direction
AZ	Scale factor for acceleration in z-direction
OMX	Scale factor for x-angular velocity
OMY	Scale factor for y-angular velocity
OMZ	Scale factor for z-angular velocity

Remarks:

1. Translational base accelerations allow body forces loads to be imposed on a structure. Conceptually, base acceleration may be thought of as accelerating the coordinate system in the direction specified, and, thus, the inertial loads acting on the model are of opposite sign. For example, if a cylinder were fixed to the y-z plane and extended in the positive x-direction, then a positive x-direction base acceleration would tend to shorten the cylinder, i.e., create forces acting in the negative x-direction.
2. Base accelerations are frequently used to impose gravitational loads during dynamic relaxation to initialize the stresses and displacements. During the analysis, in this latter case, the body forces loads are held constant to simulate gravitational loads. When imposing loads during dynamic relaxation, it is recommended that the load curve slowly ramp up to avoid the excitation of a high frequency response.
3. Body force loads due to the angular velocity about an axis are calculated with respect to the deformed configuration and act radially outward from the axis of rotation. Torsional effects which arise from changes in angular velocity are neglected with this option. The angular velocity is assumed to have the units of radians per unit time.

4. The body force density is given at a point P of the body by:

$$b = \rho(\omega \times \omega \times r)$$

where ρ is the mass density, ω is the angular velocity vector, and r is a position vector from the origin to point P. Although the angular velocity may vary with time, the effects of angular acceleration are included.

5. Angular velocities are useful for studying transient deformation of spinning three-dimensional objects. Typical applications have included stress initialization during dynamic relaxation where the initial rotational velocities are assigned at the completion of the initialization, and this option ceases to be active.

***LOAD_BODY_POROUS**

Purpose: Define the effects of porosity on the flow with body-force-like loads applied to the ALE element nodes. Ergun porous flow assumptions are used. This only applies to non-deformable (constant-porosity), fully saturated porous media. This model only works with a non-zero and constant viscosity fluid defined via either *MAT_NULL or *MAT_ALE_VISCOUS card.

For options: X, Y, Z, RX, RY, and RZ.

	1	2	3	4	5	6	7	8
Variable	SID	SIDTYP	AX	AY	AZ	BX	BY	BZ
Type	I	I	F	F	F	F	F	F
Default	0	0	0.0	0.0	0.0	0.0	0.0	0.0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID of the ALE fluid part subjected to porous flow condition.
SIDTYP	Set ID type of the SID above. If SIDTYP=0 (default), then the SID=PSID (part set ID). If SIDTYP=1, then SID=PID (part ID).
AX,AY,AZ	Permeability coefficients for viscous terms in global X, Y and Z directions (please see equation below). If $\{[A_x \neq 0] \& [A_y = 0] \& [A_z = 0]\}$, then an isotropic viscous permeability condition is assumed for the porous medium.
BX,BY,BZ	Passability coefficients for inertia terms in global X, Y and Z directions (please see equation below). If $\{[B_x \neq 0] \& [B_y = 0] \& [B_z = 0]\}$, then an isotropic inertial permeability condition is assumed for the porous medium.

Remarks:

1. Consider the basic general Ergun equation for porous flow in one direction:

$$\frac{\Delta P}{\Delta L} = \frac{\mu}{k_1} V_s + \frac{\rho}{k_2} V_s^2$$

Where

ρ = Fluid density.

μ = Fluid dynamic viscosity.

$V_s = \frac{4Q}{\pi D^2}$ = Superficial fluid velocity.

Q = Overall volume flow rate (m³/s).

D = Porous channel characteristic width (perpendicular to ΔL).

$k_1 = \frac{\varepsilon^3 d_p^2}{150(1-\varepsilon)^2}$ = Viscous parameter

$k_2 = \frac{\varepsilon^3 d_p}{1.75(1-\varepsilon)}$ = Inertial parameter

ε = Porosity = Pore volume / Total media volume

d_p = Particle diameter

2. The above equation can be generalized into 3 dimensional flows where each component may be written as

$$-\frac{dP}{dx_i} = A_i \mu V_i + B_i \rho |V_i| V_i$$

where $i=1,2,3$ refers to the global coordinate directions (no summation intended for repeated indices), μ is the constant dynamic viscosity, ρ is the fluid density, V_i is the fluid velocity components, A_i is analogous to k_1 above, and B_i is analogous to k_2 above.

3. If $B_i = 0$, the equation is reduced to simple Darcy Law for porous flow (may be good for sand-like flow). For coarse grain (rocks) media, the inertia term will be important and the user needs to input these coefficients.

***LOAD_BRODE**

Purpose: Define Brode function for application of pressure loads due to explosion, see Brode [1970], also see *LOAD_SEGMENT, *LOAD_SEGMENT_SET, or *LOAD_SHELL.

Card 1 1 2 3 4 5 6 7 8

Variable	YLD	BHT	XBO	YBO	ZBO	TBO	TALC	SFLC
Type	F	F	F	F	F	F	I	I
Default	0.0	0.0	0.0	0.0	0.0	0.0	0	0
Remarks							1	1

Card 2

Variable	CFL	CFT	CFP					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE**DESCRIPTION**

YLD	Yield (Kt, equivalent tons of TNT).
BHT	Height of burst.
XBO	x-coordinates of Brode origin.
YBO	y-coordinates of Brode origin.
ZBO	z-coordinates of Brode origin.
TBO	Time offset of Brode origin.
TALC	Load curve number giving time of arrival versus range relative to Brode origin (space, time), see *DEFINE_CURVE and remark below.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SFLC	Load curve number giving yield scaling versus scaled time (time relative to Brode origin divided by $[\text{yield}^{(1/3)}]$) origin (space, time), see *DEFINE_CURVE and remark below.
CFL	Conversion factor - kft to LS-DYNA length units.
CFT	Conversion factor - milliseconds to LS-DYNA time units.
CFP	Conversion factor - psi to LS-DYNA pressure units.

Remarks:

1. If these curves are defined a variable yield is assumed. Both load curves must be specified for the variable yield option. If this option is used, the shock time of arrival is found from the time of arrival curve. The yield used in the Brode formulas is computed by taking the value from the yield scaling curve at the current time/ $[\text{yield}^{(1/3)}]$ and multiplying that value by yield.

***LOAD_DENSITY_DEPTH**

Purpose: Define density versus depth for gravity loading. This option has been occasionally used for analyzing underground and submerged structures where the gravitational preload is important. The purpose of this option is to initialize the hydrostatic pressure field at the integration points in the element.

This card should be only defined once in the input deck.

Card 1 2 3 4 5 6 7 8

Variable	PSID	GC	DIR	LCID				
Type	I	F	I	I				
Default	0	0.0	1	none				
Remarks	1,2			3				

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PSID	Part set ID, see *SET_PART. If a PSID of zero is defined then all parts are initialized.
GC	Gravitational acceleration value.
DIR	Direction of loading: EQ.1: global x, EQ.2: global y, EQ.3: global z.
LCID	Load curve ID defining density versus depth, see *DEFINE_CURVE.

Remarks:

1. Density versus depth curves are used to initialize hydrostatic pressure due to gravity acting on an overburden material. The hydrostatic pressure acting at a material point at depth, d, is given by:

$$p = - \int_d^{d_{surface}} p(z)gdz$$

where p is pressure, $d_{surface}$, is the depth of the surface of the material to be initialized (usually zero), $\rho(z)$ is the mass density at depth z , and g is the acceleration of gravity. This integral is evaluated for each integration point. Depth may be measured along any of the global coordinate axes, and the sign convention of the global coordinate system should be respected. The sign convention of gravity also follows that of the global coordinate system. For example, if the positive z axis points "up", then gravitational acceleration should be input as a negative number.

2. For this option there is a limit of 12 parts that can be defined by PSID, unless all parts are initialized.
3. Depth is the ordinate of the curve and is input as a descending x, y, or z coordinate value. Density is the abscissa of the curve and must vary (increase) with depth, i.e., an infinite slope is not allowed.

***LOAD_GRAVITY_PART**

Purpose: Define gravity for individual parts. This feature is intended for use with *LOAD_STIFFEN_PART to simulate staged construction. Available for solids and shells, and also beam element types 1, 2, 6, and 9.

Note: This keyword card will be available starting in release 3 of version 971.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	DOF	LC	ACCEL	LCDR	STGA	STGR	
Type	I	I	I	F	I	I	I	
Default	none	none	none	0	none	0	0	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID for application of gravity load
DOF	Direction: enter 1, 2 or 3 for X, Y or Z
LC	Load curve defining factor vs. time (or zero if STGA, STGR are defined)
ACCEL	Acceleration (will be multiplied by factor from curve)
LCDR	Load curve defining factor vs. time during dynamic relaxation
STGA	Construction stage at which part is added (optional)
STGR	Construction stage at which part is removed (optional)

Remarks:

There are 3 options for defining how the gravity load on a part varies with time.

1. Curve LC gives factor vs time. This overrides the other methods if LC is non-zero.
2. STGA, STGR refer to stages at which part is added and removed – the stages are defined in *DEFINE_CONSTRUCTION_STAGES. If STGA is zero, the gravity load starts at time zero. If not, it ramps up from the small factor FACT (on *CONTROL_STAGED_CONSTRUCTION) up to full value over the ramp time

at the start of stage STGA. If STGR is zero, the gravity load continues until the end of the analysis. If not, it ramps down from full value to FACT over the ramp time at the start of stage STGR.

3. *DEFINE_STAGED_CONSTRUCTION_PART can be used instead of *LOAD_GRAVITY_PART to define this loading. During initialization, a LOAD_GRAVITY_PART card will be created and the effect is the same as using the STGA, STGR method described above; ACCEL is then taken from *CONTROL_STAGED_CONSTRUCTION.

***LOAD_HEAT_CONTROLLER**

Purpose: Used to define a thermostat control function. The thermostat controls the heat generation within a material by monitoring a remote nodal temperature. Control can be specified as on-off, proportional, integral, or proportional + integral.

Card 1 1 2 3 4 5 6 7 8

Variable	NODE	PID	LOAD	TSET	TYPE	GP	GI	
Type	I	I	F	F	I	F	F	
Default	none							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NODE	Sensor is located at this node number.
PID	Part ID assigned to the elements modeling the heater or cooler being controlled.
LOAD	Heater output (q_0). [typical units: W/m ³]
TSET	Controller set point temperature at the location identified by NODE.
TYPE	Type of control function. EQ.1: on-off EQ.2: proportional + integral
GP	Proportional gain.
GI	Integral gain.

Remarks:

The thermostat control function is

$$\dot{q}''' = \dot{q}_0''' + G_P (T_{set} - T_{node}) + G_I \int_{t=0}^t (T_{set} - T_{node}) dt$$

*LOAD

*LOAD_HEAT_GENERATION

*LOAD_HEAT_GENERATION_OPTION

Available options include:

SET

SOLID

Purpose: Define solid elements or solid element set with heat generation.

Card	1	2	3	4	5	6	7	8
Variable	SID	LCID	CMULT	WBLCID	CBLCID	TBLCID		
Type	I	I	F	I	I	I		
Default	none	none	0.	0	0	0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Solid element set ID or solid element ID, see *SET_SOLID or *ELEMENT_SOLID, respectively.
LCID	Load curve ID for volumetric heat generation rate, \dot{q}''' : GT.0: function versus time, EQ.0: use multiplier value CMULT only, LT.0: function versus temperature.
CMULT	Curve multiplier for \dot{q}''' . Depending on the definition of LCID this value is either used for scaling or for constant heat generation.
WBLCID	Load curve ID defining the blood perfusion rate [e.g., kg/m ³ sec] as a function of time.
CBLCID	Load curve ID defining the blood heat capacity [e.g., J/kg C] as a function of the blood temperature.
TBLCID	Load curve ID defining the blood temperature [e.g., C] as a function of time.

Remarks:

$$\text{Rate of heat transfer from blood to tissue} = W_b C_b (T_b - T) \quad [\text{units: J/m}^3 \text{ sec}]$$

***LOAD_MASK**

Purpose: Apply a distributed pressure load over a three-dimensional shell part. The pressure is applied to a subset of elements that are within a fixed global box and lie either outside or inside of a closed curve in space which is projected onto the surface.

Card 1 2 3 4 5 6 7 8

Variable	PID	LCID	VID1	OFF	BOXID	LCIDM	VID2	INOUT
Type	I	I	F	F	I	I	I	I
Default	none	none	1.	0.	0	0	none	0
Remarks	1		2					

Card

Variable	ICYCLE							
Type	I							
Default	200							
Remarks								

VARIABLE

DESCRIPTION

- PID Part ID (PID). This part must consist of 3D shell elements. To use this option with solid element the surface of the solid elements must be covered with null shells. See *MAT_NULL.
- LCID Curve ID defining the pressure time history, see *DEFINE_CURVE.
- VID1 Vector ID normal to the surface on which the applied pressure acts. Positive pressure acts in a direction that is in the opposite direction. This vector may be used if the surface on which the pressure acts is relatively flat. If zero, the pressure load depends on the orientation of the shell elements as shown in Figure 22.3.

VARIABLE	DESCRIPTION
OFF	Pressure loads will be discontinued if $ VID1 \cdot n_{shell} < OFF$ where n_{shell} is the normal vector to the shell element.
BOXID	Only elements inside the box with part ID, PID, are considered. If no ID is given all elements of part ID, PID, are included. When the active list of elements are updated, elements outside the box will no longer have pressure applied, i.e., the current configuration is always used.
LCIDM	Curve ID defining the mask. This curve gives (x,y) pairs of points in a local coordinate system defined by the vector ID, VID2. Generally, the curve should form a closed loop, i.e., the first point is identical to the last point, and the curve should be flagged as a DATTYP=1 curve in the *DEFINE_CURVE section. If no curve ID is given, all elements of part ID, PID, are included with the exception of those deleted by the box. The mask works like the trimming option, i.e., see DEFINE_CURVE_TRIM and Figure 11.5.
VID2	Vector ID used to project the masking curve onto the surface of part ID, PID. The origin of this vector determines the origin of the local system that the coordinates of the PID are transformed into prior to determining the pressure distribution in the local system. This curve must be defined if LCIDM is nonzero. See Figure 11.5.
INOUT	If 0, elements whose center falls inside the projected curve are considered. If 1, elements whose center falls outside the projected curve are considered.
ICYCLE	Number of time steps between updating the list of active elements (default=200). The list update can be quite expensive and should be done at a reasonable interval. The default is not be appropriate for all problems.

Remarks:

1. The part ID must consist of 3D shell elements.

***LOAD_MOTION_NODE**

Purpose: Apply a concentrated nodal force or moment to a node based on the motion of another node.

Card 1 2 3 4 5 6 7 8

Variable	NODE1	DOF1	LCID	SF	CID1	NODE2	DOF2	CID2
Type	I	I	I	F	I	I	I	I
Default	none	none	none	1.	0	0	0	0
Remarks					1			1

VARIABLE**DESCRIPTION**

NODE1	Node ID for the concentrated force.
DOF1	Applicable degrees-of-freedom: EQ.1: x-direction of load action, EQ.2: y-direction of load action, EQ.3: z-direction of load action, EQ.4: moment about the x-axis, EQ.5: moment about the y-axis, EQ.6: moment about the z-axis.
LCID	Load curve ID, see *DEFINE_CURVE. The applied force is a function of the applicable degree-of-freedom of NODE2.
SF	Load curve scale factor.
CID1	Coordinate system ID (optional), see remark 1 on next page.
NODE2	Node ID for calculating the force.
DOF2	Applicable degrees-of-freedom: EQ. 1: x-coordinate EQ. 2: y-coordinate, EQ. 3: z-coordinate, EQ. 4: x-translational displacement, EQ. 5: y-translational displacement,

***LOAD**

***LOAD_MOTION_NODE**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ. 6: z-translational displacement, EQ. 7: rotational displacement about the x-axis, EQ. 8: rotational displacement about the y-axis, EQ. 9: rotational displacement about the z-axis. EQ.10: x-translational velocity, EQ.11: y-translational velocity, EQ.12: z-translational velocity, EQ.13: rotational velocity about the x-axis, EQ.14: rotational velocity about the y-axis, EQ.15: rotational velocity about the z-axis.
CID2	Coordinate system ID (optional), see Remark 1.

Remarks:

1. The global coordinate system is the default. The local coordinate system ID's are defined in the *DEFINE_COORDINATE_SYSTEM section.

LOAD_MOVING_PRESSURE**LOAD*****LOAD_MOVING_PRESSURE**

Purpose: Apply moving pressure loads to a surface. The pressure loads approximate a jet of high velocity fluid impinging on the surface. Multiple surfaces may be defined each acted on by a set of nozzles.

Card 1 1 2 3 4 5 6 7 8

Variable	LOADID								
Type	I								
Default	none								

Define the following cards for each nozzle. Include one card for each nozzle

Cards 2,...,n 1 2 3 4 5 6 7 8

Variable	NODE1	NODE2	LCID	CUTOFF	LCIDT				
Type	I	I	I	F	I				
Default	none	none	none	none	0				

The following card defines the surface where the nozzles act.

Card n+1 1 2 3 4 5 6 7 8

Variable	ID	IDTYPE	NIP						
Type	I	I	I						
Default	none	none	3x3						

*LOAD

*LOAD_MOVING_PRESSURE

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LOADID	Loading ID.
NODE1	Node located at the origin of the nozzle.
NODE2	Node located at the head of the nozzle
ID	Segment set ID, shell element set ID, part set ID, or part ID. See IDT below.
IDT	Slave segment or node set type. The type must correlate with the number specified for SSID: EQ.0: segment set ID for surface-to-surface contact, EQ.1: shell element set ID for surface-to-surface contact, EQ.2: part set ID, EQ.3: part ID,
NIP	Number of integration in segment used to compute pressure loads.
LCID	Load curve ID defining pressure versus radial distance from the center of the jet.
CUTOFF	Outer radius of jet. The pressure acting outside this radius is set to zero.
LCIDT	Load curve ID, which scales the pressure as a function of time. If a load curve isn't specified, the scale factor defaults to 1.0.

***LOAD_NODE_OPTION**

Available options include:

POINT**SET**

Purpose: Apply a concentrated nodal force to a node or a set of nodes.

Card 1 2 3 4 5 6 7 8

Variable	NODE/NSID	DOF	LCID	SF	CID	M1	M2	M3
Type	I	I	I	F	I	I	I	I
Default	none	none	none	1.	0	0	0	0
Remarks					1	2		

VARIABLE**DESCRIPTION**

NODE/NSID

Node ID or nodal set ID (NSID), see **SET_NODE_OPTION*.

DOF

Applicable degrees-of-freedom:

EQ.1: x-direction of load action,

EQ.2: y-direction of load action,

EQ.3: z-direction of load action,

EQ.4: follower force, see remark 2 on next page,

EQ.5: moment about the x-axis,

EQ.6: moment about the y-axis,

EQ.7: moment about the z-axis.

EQ.8: follower moment

LCID

Load curve ID, see **DEFINE_CURVE*.

SF

Load curve scale factor.

CID

Coordinate system ID (optional), see remark 1 on next page.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
M1	Node 1 ID. Only necessary if DOF.EQ.4 or 8, see remark 2 below.
M2	Node 2 ID. Only necessary if DOF.EQ.4 or 8, see remark 2 below.
M3	Node 3 ID. Only necessary if DOF.EQ.4 or 8, see remark 2 below.

Remarks:

1. The global coordinate system is the default. The local coordinate system ID's are defined in the *DEFINE_COORDINATE_SYSTEM section.
2. Nodes M_1 , M_2 , M_3 must be defined for a follower force. A positive follower force acts normal to the plane defined by these nodes, and a positive follower moment puts a counterclockwise torque about the t-axis. These actions are depicted in Figure 22.2.
3. For shell formulations 14 and 15, the axisymmetric solid elements with area and volume weighting, respectively, the specified nodal load is per unit length (type14) and per radian (type 15).

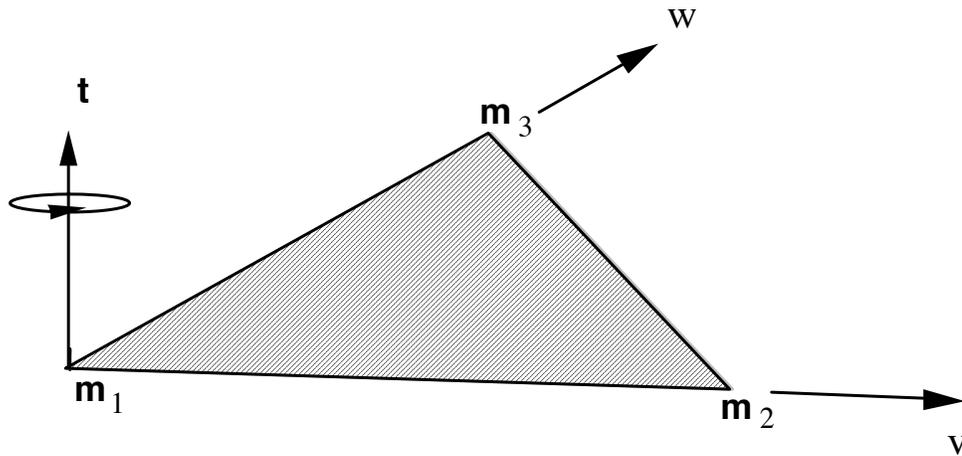


Figure 22.2. Follower force and moment acting on a plane defined by nodes m_1 , m_2 , and m_3 . In this case, the load is applied to node m_1 ; i.e., $m=m_1$. A positive force acts in the positive t -direction, and a positive moment puts a counterclockwise torque about the normal vector. The positive t -direction is found by the cross product $t = v \times w$ where v and w are vectors as shown.

***LOAD_REMOVE_PART**

Purpose: Delete the elements of a part. Shock effects are prevented by gradually reducing the stresses prior to deletion. Available only for solid and shell elements.

Note: This keyword card will be available starting in release 3 of version 971.

Card	1	2	3	4	5	6	7	8
Variable	PID	TIME0	TIME1	STGR				
Type	I	F	F	I				
Default	none	0	0	0				

VARIABLE**DESCRIPTION**

PID	Part ID for deletion
TIME0	Time at which stress reduction starts
TIME1	Time at which stresses become zero and elements are deleted
STGR	Construction stage at which part is removed (optional)

Remarks:

There are 3 methods of defining the part removal time:

1. TIME0, TIME1 override all the other methods if non-zero
2. STGR refers to the stage at which the part is removed – the stages are defined in *DEFINE_CONSTRUCTION_STAGES. This is equivalent to setting TIME0 and TIME1 equal to the start and end of the ramp time at the beginning of stage STGR.
3. *DEFINE_STAGED_CONSTRUCTION_PART can be used instead of *LOAD_REMOVE_PART to define this loading. During initialization, a STIFFEN_PART card will be created and the effect is the same as using the STGA, STGR method described above.

***LOAD_RIGID_BODY**

Purpose: Apply a concentrated nodal force to a rigid body. The force is applied at the center of mass or a moment is applied around a global axis. As an option, local axes can be defined for force or moment directions.

Card 1 2 3 4 5 6 7 8

Variable	PID	DOF	LCID	SF	CID	M1	M2	M3
Type	I	I	I	F	I	I	I	I
Default	none	none	none	1.	0	0	0	0
Remark					1	2		

VARIABLE**DESCRIPTION**

PID	Part ID of the rigid body, see <i>*PART_OPTION</i> .
DOF	Applicable degrees-of-freedom: EQ.1: x-direction of load action, EQ.2: y-direction of load action, EQ.3: z-direction of load action, EQ.4: follower force, see Remark 2, EQ.5: moment about the x-axis, EQ.6: moment about the y-axis, EQ.7: moment about the z-axis. EQ.8: follower moment, see Remark 2.
LCID	Load curve ID, see <i>*DEFINE_CURVE</i> . GT.0: force as a function of time, LT.0: force as a function of the absolute value of the rigid body displacement.
SF	Load curve scale factor
CID	Coordinate system ID
M1	Node 1 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2.
M2	Node 2 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2.
M3	Node 3 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2.

*LOAD

*LOAD_SEGMENT

Extra card if N5 is not zero

Card	1	2	3	4	5	6	7	8
Variable	N6	N7	N8					
Type	I	I	I					
Default	none	none	none					
Remarks								

VARIABLE

DESCRIPTION

ID	Loading ID
HEADING	A description of the loading.
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor
AT	Arrival time for pressure or birth time of pressure.
N1	Node ID
N2	Node ID
N3	Node ID. Repeat N2 for two-dimensional geometries.
N4	Node ID. Repeat N2 for two-dimensional geometries.
N5	Optional mid-side node ID located between nodes 1 and 2.
N6	Optional mid-side node ID located between nodes 2 and 3.
N7	Optional mid-side node ID located between nodes 3 and 4.
N8	Optional mid-side node ID located between nodes 4 and 1. Do not define for six node quadratic surface segments.

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine the pressure for the segments, see *LOAD_BRODE.
2. If LCID is input as -2, then the ConWep function is used to determine the pressure for the segments, see *LOAD_BLAST.
3. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.
4. The activation time, AT, is the time during the solution that the pressure begins to act. Until this time, the pressure is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and AT i.e., (solution time-AT). Relative displacements that occur prior to reaching AT are ignored. Only relative displacements that occur after AT are prescribed.
5. Triangular segments are defined by repeating the third node.

*LOAD

*LOAD_SEGMENT_NONUNIFORM

Card 3 1 2 3 4 5 6 7 8

Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	none							

VARIABLE

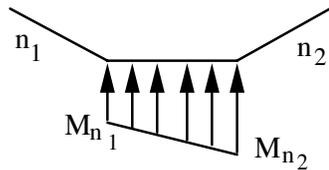
DESCRIPTION

ID	Loading ID
HEADING	A description of the loading.
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor
AT	Arrival/birth time for the traction load.
DT	Death time for the traction load.
CID	Coordinate system ID
V1,V2,V3	Vector direction cosines referenced to coordinate system CID to define the direction of the traction loading.
N1	Node ID
N2	Node ID
N3	Node ID. Repeat N2 for two-dimensional geometries.
N4	Node ID. Repeat N2 for two-dimensional geometries or repeat N3 for triangular segments.
N5	Optional mid-side node ID located between nodes 1 and 2.
N6	Optional mid-side node ID located between nodes 2 and 3.
N7	Optional mid-side node ID located between nodes 3 and 4.
N8	Optional mid-side node ID located between nodes 4 and 1. Do not define for six node quadratic surface segments.

VARIABLE	DESCRIPTION
P1	Scale factor at node ID, N1.
P2	Scale factor at node ID, N2.
P3	Scale factor at node ID, N3.
P4	Scale factor at node ID, N4.
P5	Scale factor at node ID, N5.
P6	Scale factor at node ID, N6.
P7	Scale factor at node ID, N7.
P8	Scale factor at node ID, N8.

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine pressure for the segment set, also see *LOAD_BRODE.
2. If LCID is input as -2, then the ConWep function is used to determine the pressure for the segments, see *LOAD_BLAST.
3. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.
4. The activation time, AT, is the time during the solution that the pressure begins to act. Until this time, the pressure is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and AT i.e., (solution time-AT). Relative displacements that occur prior to reaching AT are ignored. Only relative displacements that occur after AT are prescribed.



2-Dimensional Definition for axisymmetric, plane stress, and plane strain geometries

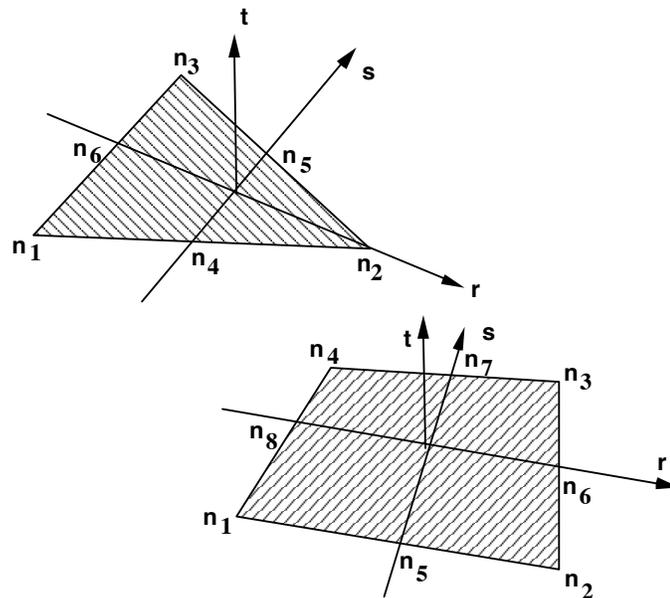


Figure 22.3. Nodal numbering for pressure cards. Positive pressure acts in the negative t -direction. For two-dimensional problems repeat the second node for the third and fourth nodes in the segment definitions.

VARIABLE	DESCRIPTION
ID	Loading ID
HEADING	A description of the loading.
SSID	Segment set ID.
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor
AT	Arrival/birth time for pressure.
DT	Death time for pressure.
CID	Coordinate system ID
V1,V2,V3	Vector direction cosines referenced to coordinate system CID to define the direction of the traction loading.

VARIABLE	DESCRIPTION
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor
AT	Arrival time for pressure or birth time of pressure.

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine the pressure for the segments, see also *LOAD_BRODE.
2. If LCID is input as -2, then the ConWep function is used to determine the pressure for the segments, see *LOAD_BLAST.
3. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.

*LOAD

*LOAD_SHELL

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *LOAD_SHELL_ELEMENT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ From a sheet metal forming example. A blank is hit by a punch, a holder is
$ used to hold the blank on its sides. All shells on the holder are given a
$ pressure boundary condition to clamp down on the blank. The pressure
$ follows load curve 3, but is scaled by -1 so that it applies the load in the
$ correct direction. The load starts at zero, but quickly rises to 5 MPa
$ after 0.001 sec. (Units of this model are in: ton, mm, s, N, MPa, N-mm)
$
*LOAD_SHELL_ELEMENT
$
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$
   eid      lcid      sf      at
30001       3 -1.00E+00   0.0
30002       3 -1.00E+00   0.0
30003       3 -1.00E+00   0.0
30004       3 -1.00E+00   0.0
30005       3 -1.00E+00   0.0
30006       3 -1.00E+00   0.0
30007       3 -1.00E+00   0.0
$
$ Note: Just a subset of all the shell elements of the holder is shown above,
$ in practice this list contained 448 shell element id's.
$
$
*DEFINE_CURVE
$   lcid      sidr      scla      sclo      offa      offo
$       3
$
$       abscissa      ordinate
$               0.000              0.0
$               0.001              5.0
$               0.150              5.0
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

***LOAD_SSA**

Purpose: The Sub-Sea Analysis capability allows a simple way of loading the structure to account for the effects of the primary explosion and the subsequent bubble oscillations.

Define one card.

Card 1 1 2 3 4 5 6 7 8

Variable	VS	DS	REFL	ZB	ZSURF	FPSID	PSID	
Type	F	F	F	F	F	I	I	
Default	none	none	0.	0.	0.	0	0	

Define two cards for each explosive charge. This input is terminated by the next “*” keyword card.

Card 1 1 2 3 4 5 6 7 8

Variable	A	ALPHA	GAMMA	KTHETA	KAPPA			
Type	F	F	F	F	F			
Default	none	none	none	none	none			

Card 2

Variable	XS	YS	ZS	W	TDELY	RAD	CZ	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
VS	Sound speed in fluid
DS	Density of fluid
REFL	Consider reflections from sea floor. EQ.0: off EQ.1: on
ZB	Z coordinate of sea floor if REFL=1, otherwise, not used.
ZSURF	Z coordinate of sea surface
FPSID	Part set ID of parts subject to flood control. Use the *PART_SET_COLUMN option where the parameters A1 and A2 must be defined as follows: Parameter A1: Flooding status: EQ.1.0: Fluid on both sides. EQ.2.0: Fluid outside, air inside. EQ.3.0: Air outside, fluid inside. EQ.4.0: Material or part is ignored. Parameter A2: Tubular outer diameter of beam elements. For shell elements this input must be greater than zero for loading.
PSID	Part IDs of parts defining the wet surface. The elements defining these parts must have their outward normals pointing into the fluid. See Figure 22.4. EQ.0: all parts are included. GT.0: define NPIDS part ID's below.
A	Shock pressure parameter
ALPHA	α , shock pressure parameter
GAMMA	γ , time constant parameter
KTHETA	K_θ , time constant parameter
KAPPA	κ , ratio of specific heat capacities
XS	X coordinate of charge
YS	Y coordinate of charge
ZS	Z coordinate of charge

VARIABLE	DESCRIPTION
W	Weight of charge
TDELY	Time delay before charge detonates
RAD	Charge radius
CZ	Water depth

Remarks:

The pressure history of the primary shockwave at a point in space through which a detonation wave passes is given as:

$$P(t) = P_m e^{-\frac{t}{\theta}}$$

where P_m and the time constant θ below are functions of the type and weight W of the explosive charge and the distance Q from the charge.

$$P_{peak} = A \left[\frac{W^{1/3}}{Q} \right]^\alpha$$

$$\theta = K_\theta W^{1/3} \left[\frac{W^{1/3}}{Q} \right]^\gamma$$

where A , α , γ , and K_θ are constants for the explosive being used.

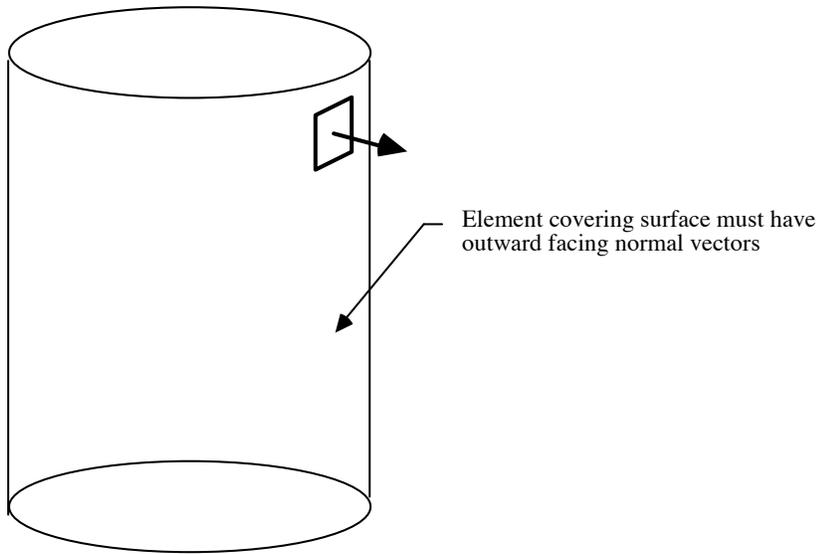


Figure 22.4. The shell elements interacting with the fluid must be numbered such that their outward normal vector points into the fluid media.

***LOAD_STIFFEN_PART**

Purpose: Staged construction. Available for solid, shell, and beam elements.

Note: This keyword card is available starting in release 3 of version 971.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	LC	(blank)	STGA	STGR			
Type	I	I		I	I			
Default	none	0		0	0			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID
LC	Load curve defining factor vs. time
STGA	Construction stage at which part is added (optional)
STGR	Construction stage at which part is removed (optional)

Remarks:

1. In many cases it is more convenient to use *DEFINE_STAGED_CONSTRUCTION_PART – this card creates *LOAD_STIFFEN_PART data automatically.
2. For parts that are initially present but are excavated (removed) during the analysis, the stiffness factor starts at 1.0. During the excavation time, it ramps down to a small value such as 1.0E-6. The excavation time should be sufficiently long to avoid introducing shock or dynamic effects. For parts that are introduced during the construction, e.g. retaining walls, the elements are initially present in the model but the factor is set to a low value such as 1.0e-6. During the construction time the factor should be ramped up to 1.0. The construction time should be sufficiently long to avoid shock or dynamic effects. A factor that ramps up from 1.0E-6 to 1.0, then reduces back to 1.0E-6, can be used for temporary retaining walls, props, etc.
3. When the factor is increasing, it applies only to the stiffness and strength of the material in response to subsequent strain increments, not to any existing stresses.

4. When the factor is decreasing, it applies also to existing stresses as well as to the stiffness and strength.
5. This feature works with all material models when used only to reduce the stiffness (e.g. parts that are excavated, not parts that are added during construction). It works for most material types in all other cases, except those few materials that re-calculate stresses each time step from total strains (elastic, SOIL_BRICK, rubber models, orthotropic elastic, fabric, etc). There is no error check at present to detect STIFFEN_PART being used with an inappropriate material model. Symptoms of resulting problems would include non-physical large stresses when a part stiffens, due to the accumulated strains in the “dormant” material since the start of the analysis.
6. This feature is generally used with *LOAD_GRAVITY_PART. The same curve is often used for the stiffness factor and the gravity factor.
7. There are 3 methods of defining the factor-versus-time:
 1. LC overrides all the other methods if non-zero
 2. STGA, STGR refer to stages at which the part is added and removed – the stages are defined in *DEFINE_CONSTRUCTION_STAGES. If STGA is zero, the part has full stiffness at time zero. If not, it ramps up from the small factor FACT (on *CONTROL_STAGED_CONSTRUCTION) up to 1.0 over the ramp time at the start of stage STGA. If STGR is zero, the stiffness factor continues at 1.0 until the end of the analysis. If not, it ramps down from 1.0 to FACT over the ramp time at the start of stage STGR.
 3. *DEFINE_STAGED_CONSTRUCTION_PART can be used instead of *LOAD_STIFFEN_PART to define this loading. During initialization, a *LOAD_STIFFEN_PART card will be created and the effect is the same as using the STGA, STGR method described above.

***LOAD_SUPERPLASTIC_FORMING**

Purpose: Perform superplastic forming (SPF) analyses. This option can be applied to both solid and shell elements. The pressure loading controlled by the load curve ID given below is scaled to maintain a constant maximum strain rate.

This option must be used with material model 64, *MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY, for strain rate sensitive, powerlaw plasticity. For the output of data, see *DATABASE_SUPERPLASTIC_FORMING. Mass scaling is recommended in SPF applications.

Card 1 2 3 4 5 6 7 8

Variable	LCP1	CSP1	NCP1	LCP2	CSP2	NCP2		
Type	I	I	F	I	I	F		
Default	none	none	none.	none	none	none		
Remarks				1	1	1		

Card

Variable	ERATE	SCMIN	SCMAX	NCYL				
Type	F	F	F	I				
Default	none	none	none.	0				
Remarks				2				

VARIABLE

DESCRIPTION

- LCP1 Load curve number for Phase I pressure loading, (see *DEFINE_CURVE).
- CSP1 Contact surface number to determine completion of Phase 1.
- NCP1 Percent of nodes in contact to terminate Phase I, (see *CONTACT_OPTION).

<u>VARIABLE</u>	<u>DESCRIPTION</u>
LCP2	Load curve number for Phase II pressure loading (reverse), (see *DEFINE_CURVE).
CSP2	Contact surface to determine completion of Phase II, (see *CONTACT_OPTION).
NCP2	Percent of nodes in contact to terminate Phase II.
ERATE	Desired strain rate. This is the time derivative of the logarithmic strain.
SCMIN	Minimum allowable value for load curve scale factor. To maintain a constant strain rate the pressure curve is scaled. In the case of a snap through buckling the pressure may be removed completely. By putting a value here the pressure will continue to act but at a value given by this scale factor multiplying the pressure curve.
SCMAX	Maximum allowable value for load curve scale factor. Generally, it is a good idea to put a value here to keep the pressure from going to unreasonable values after full contact has been attained. When full contact is achieved the strain rates will approach zero and pressure will go to infinity unless it is limited or the calculation terminates.
NCYL	Number of cycles for monotonic pressure after reversal.

Remarks:

1. Optionally, a second phase can be defined. In this second phase a unique set of pressure segments must be defined whose pressure is controlled by load curve 2. During the first phase, the pressure segments of load curve 2 are inactive, and likewise, during the second phase the pressure segments of the first phase are inactive. When shell elements are used the complete set of pressure segments can be repeated in the input with a sign reversal used on the load curve. When solid elements are used the pressure segments for each phase will, in general, be unique.
2. This is an ad hoc parameter which should probably not be used.
3. Data in the output files “pressure”, “curve1”, and “curve2”, may be plotted using ASCII > superpl in LS-PREPOST. The file “curve2” is created only if the second phase is active. See *DATABASE_SUPERPLASTIC_FORMING.
4. The constraint method contact, *CONTACT_CONSTRAINT_NODES_TO_SURFACE, is recommended for superplastic forming simulations since the penalty methods are not as reliable when mass scaling is applied. Generally, in superplastic simulations mass scaling is used to enable the calculation to be carried out in real time.

***LOAD_SURFACE_STRESS_{OPTION}**

Available options include:

<BLANK>

SET

Purpose: Store segment pressures from contact and applied pressure loads on the upper and lower surfaces of the shell surface.

Card 1 1 2 3 4 5 6 7 8

Variable	PID/PSID								
Type	I								

Card 2

Variable	LSCID1	LSCID2	LSCID3	LSCID4	LSCID5	LSCID6	LSCID7	LSCID8
Type	I	I	I	I	I	I	I	I

Card 3

Variable	USCID1	USCID2	USCID3	USCID4	USCID5	USCID6	USCID7	USCID8
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

PID/PSID

Part ID or if option set is active, part set ID.

LSCID_n

Lower surface contact ID's. Up to eight ID's can be defined. These contacts contribute to the pressure acting on the lower surface of the shell. If the pressure on the lower surface is due to applied pressure loads, specify a -1 instead of a contact ID. Only one, -1, may exist in the set of 8.

USCID_n

Upper surface contact ID's. Up to eight ID's can be defined. These contacts contribute to the pressure acting on the upper surface of the shell. . If the pressure on the upper surface is due to applied pressure loads, specify a -1 instead of a contact ID. Only one, -1, may exist in the set of 8.

***LOAD_THERMAL_OPTION**

Available options include:

CONSTANT

CONSTANT_NODE

LOAD_CURVE

TOPAZ

VARIABLE

VARIABLE_NODE

Purpose: To define nodal temperatures that thermally load the structure. Nodal temperatures defined by the **LOAD_THERMAL_OPTION* method are all applied in a structural only analysis. They are ignored in a thermal only or coupled thermal/structural analysis, see **CONTROL_THERMAL_OPTION*.

All the **LOAD_THERMAL* options cannot be used in conjunction with each other. Only those of the same thermal load type, as defined below in column 2, may be used together.

<i>*LOAD_THERMAL_CONSTANT</i>	- Thermal load type 1
<i>*LOAD_THERMAL_CONSTANT_NODE</i>	- Thermal load type 1
<i>*LOAD_THERMAL_LOAD_CURVE</i>	- Thermal load type 2
<i>*LOAD_THERMAL_TOPAZ</i>	- Thermal load type 3
<i>*LOAD_THERMAL_VARIABLE</i>	- Thermal load type 4
<i>*LOAD_THERMAL_VARIABLE_NODE</i>	- Thermal load type 4

***LOAD_THERMAL_CONSTANT**

Purpose: Define nodal sets giving the temperature that remains constant for the duration of the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state, read in above and held constant throughout the analysis, dynamically loads the structure. Thus, the temperature defined can also be seen as a relative temperature to a surrounding or initial temperature.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	NSIDEX	BOXID					
Type	I	I	I					
Default	none	0.	0.					

Card 2

Variable	T	TE					
Type	F	F					
Default	0.	0.					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NSID	Nodal set ID containing nodes for initial temperature (see *SET_NODES): EQ.0: all nodes are included:
NSIDEX	Nodal set ID containing nodes that are exempted from the imposed temperature (optional).
BOXID	All nodes in box which belong to NSID are initialized. Others are excluded (optional).
T	Temperature
TE	Temperature of exempted nodes (optional)

*LOAD

*LOAD_THERMAL_CONSTANT_NODE

*LOAD_THERMAL_CONSTANT_NODE

Purpose: Define nodal temperature that remains constant for the duration of the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state, read in above and held constant throughout the analysis, dynamically loads the structure. Thus, the temperature defined can also be seen as a relative temperature to a surrounding or initial temperature.

Card 1 2 3 4 5 6 7 8

Variable	NID	T						
Type	I	F						
Default	none	0.						

VARIABLE

DESCRIPTION

NID

Node ID

T

Temperature, see remark below.

Remarks:

1. The temperature range for the constitutive constants in the thermal materials must include the reference temperature of zero. If not termination will occur with a temperature out-of-range error immediately after the execution phase is entered.

***LOAD_THERMAL_LOAD_CURVE**

Purpose: Nodal temperatures will be uniform throughout the model and will vary according to a load curve. The temperature at time=0 becomes the reference temperature for the thermal material. The reference temperature is obtained from the optional curve for dynamic relaxation if this curve is used. The load curve option for dynamic relaxation is useful for initializing preloads.

Card 1 2 3 4 5 6 7 8

Variable	LCID	LCIDDR						
Type	I	I						
Default	none	0						

VARIABLE

DESCRIPTION

LCID Load curve ID, see *DEFINE_CURVE, to define temperature versus time.

LCIDDR An optional load curve ID, see *DEFINE_CURVE, to define temperature versus time during the dynamic relaxation phase.

***LOAD**

***LOAD_THERMAL_TOPAZ**

***LOAD_THERMAL_TOPAZ**

Purpose: Nodal temperatures will be read in from the TOPAZ3D database. This file is defined in the EXECUTION SYNTAX, see GETTING STARTED.

***LOAD_THERMAL_VARIABLE**

Purpose: Define nodal sets giving the temperature that is variable in the duration of the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state, read in above and varied according to the load curve, dynamically loads the structure. Thus, the defined temperatures are relative temperatures to an initial reference temperature.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	NSIDEX	BOXID					
Type	I	I	I					
Default	none	0.	0.					

Card 2

Variable	TS	TB	LCID	TSE	TBE	LCIDE		
Type	F	F	I	F	F	I		
Default	0.	0.	none	0.	0.	none		
Remark	1	1	1	1	1			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NSID	Nodal set ID containing nodes (see *SET_NODE_OPTION): EQ.0: all nodes are included.
NSIDEX	Nodal set ID containing nodes that are exempted (optional), (see *SET_NODE_OPTION).
BOXID	All nodes in box which belong to NSID are initialized. Others are excluded.
TS	Scaled temperature.

***LOAD**

***LOAD_THERMAL_VARIABLE**

VARIABLE	DESCRIPTION
TB	Base temperature.
LCID	Load curve ID that multiplies the scaled temperature, (see *DEFINE_CURVE).
TSE	Scaled temperature of the exempted nodes (optional).
TBE	Base temperature of the exempted nodes (optional).
LCIDE	Load curve ID that multiplies the scaled temperature of the exempted nodes (optional), (see *DEFINE_CURVE).

Remarks:

1. The temperature is defined as

$$T = T_{\text{base}} + T_{\text{scale}} f(t)$$

where $f(t)$ is the current value of the load curve, T_{scale} is the scaled temperature, and T_{base} is the base temperature.

***LOAD_THERMAL_VARIABLE_NODE**

Purpose: Define nodal temperature that is variable during the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state read in and varied according to the load curve dynamically loads the structure. Thus, the defined temperatures are relative temperatures to an initial reference temperature.

Card 1 2 3 4 5 6 7 8

Variable	NID	TS	TB	LCID				
Type	I	F	F	I				
Default	none	0.	0.	none				

VARIABLE	DESCRIPTION
NID	Node ID
TS	Scaled temperature
TB	Base temperature
LCID	Load curve ID that multiplies the scaled temperature, (see *DEFINE_CURVE).

Remarks:

1. The temperature is defined as:

$$T = T_{base} + T_{scale} f(t)$$

where $f(t)$ is the current value of the load curve, T_{scale} is the scaled temperature, and T_{base} is the base temperature

*LOAD

*LOAD_THERMAL_VARIABLE_SHELL

*LOAD_THERMAL_VARIABLE_SHELL_{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define a known temperature time history as a function of the through-thickness coordinate for the shell elements.

Card 1 1 2 3 4 5 6 7 8

Variable	ID	EID/SID						
Type	I	I						
Default	none	2						

Card 2, 3, 4, etc. Input is terminated when a “*” card is found.

Card 2... 1 2 3 4 5 6 7 8

Variable	TBASE	TSCALE	TCURVE	TCURDR	ZCO			
Type	F	F	I	I	F			
Default	0	1.0	constant	TCURVE	-1/+1			

VARIABLE

DESCRIPTION

ID	Load case ID
EID/SID	Shell/Shell set ID.
TBASE	Base temperature
TSCALE	Scale factor on temperature from load curve
TCURVE	Load curve ID for temperature vs time

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TCURDR	Load curve ID used during dynamic relaxation
ZCO	Relative coordinate through-thickness (-1.0 to +1.0)

Remarks:

1. The temperature is defined as:

$$T = T_{\text{base}} + T_{\text{scale}} f(t)$$

where $f(t)$ is the current value of the load curve, T_{scale} is the scaled temperature, and T_{base} is the base temperature.

2. If a load curve ID is undefined, unity is used instead of the value from the curve.
3. Through-thickness points must be defined in order of increasing ZCO (-1.0 to +1.0). CZO=+1.0 is the top surface of the element, i.e. the element surface in the positive outward normal vector direction from the mid-plane.
4. At least two points must be defined.
5. If the element has multiple in-plane integration points – the same temperature distribution is used at each in-plane integration point.
6. If a shell's temperature distribution is defined using this card any values defined by *LOAD_THERMAL_NODE are ignored for that shell.

*LOAD

*LOAD_VOLUME_LOSS

*LOAD_VOLUME_LOSS

Purpose: To represent the effect of tunneling on surrounding structures, it is common to assume that a pre-defined fraction (e.g., 2%) of the volume occupied by the tunnel is lost during the construction process. Available for solid elements only. This feature is currently unavailable in MPP.

Note: This keyword card will be available starting in release 3 of version 971.

Card	1	2	3	4	5	6	7	8
Variable	PSID	COORD	LCUR	FX	FY	FZ	PMIN	FACTOR
Type	I	I	I	F	F	F	F	F
Default	none	0	0	1	1	1	-1.e20	.01

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PSID	Part Set ID
COORD	(Leave blank at present)
LCUR	Curve ID containing volume fraction lost vs. time
FX	Fraction of strain occurring in X-direction
FY	Fraction of strain occurring in Y-direction
FZ	Fraction of strain occurring in Z-direction
PMIN	(Leave blank)
FACTOR	Feedback factor

Remarks:

Volume loss is modeled by a process similar to thermal contraction: if the material is unrestrained it will shrink while remaining unstressed; if restrained, stresses will become more tensile. Typically the material surrounding the tunnel offers partial restraint; the volume loss algorithm adjusts the applied “thermal” strains to attempt to achieve the desired volume loss. Optionally, FX, FY and FZ may be defined: these will be treated as ratios for the X, Y and Z strains; this feature can be used to prevent contraction parallel to the tunnel axis.

The total volume of all the parts in the part set is monitored and output at the time-history interval (on *DATABASE_BINARY_D3THDT) to a file named *vloss_output*. This file contains lines of data (*time, volume1, volume2, volume3...*) where *volume1* is the total volume of elements controlled by the first *LOAD_VOLUME_LOSS card, *volume2* is the total volume of elements controlled by the second *LOAD_VOLUME_LOSS card, etc.

***NODE**

The keywords defined in this section include:

***NODE**

***NODE_RIGID_SURFACE**

***NODE_SCALAR_{*OPTION*}**

***NODE_TRANSFORM**

*NODE

*NODE

*NODE

Purpose: Define a node and its coordinates in the global coordinate system. Also, the boundary conditions in global directions can be specified. Generally, nodes are assigned to elements; however, exceptions are possible, see remark 2 below. The nodal point ID must be unique relative to other nodes defined in the *NODE section.

Card Format (I8,3E16.0,2F8.0)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	NID	X	Y	Z	TC	RC	
Type	I	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	
Remarks					1	1	

VARIABLE	DESCRIPTION
NID	Node number
X	x coordinate
Y	y coordinate
Z	z coordinate
TC	Translational Constraint: EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.
RC	Rotational constraint: EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations,

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

Remarks:

1. Boundary conditions can also be defined on nodal points in a local (or global) system by using the keyword `*BOUNDARY_SPC`. For other possibilities also see the `*CONSTRAINED` keyword section of the manual.
2. A node without an element or a mass attached to it will be assigned a very small amount of mass and rotary inertia. Generally, massless nodes should not cause any problems but in rare cases may create stability problems if these massless nodes interact with the structure. Warning messages are printed when massless nodes are found. Also, massless nodes are used with rigid bodies to place joints, see `*CONSTRAINED_EXTRA_NODES_OPTION` and `*CONSTRAINED_NODAL_RIGID_BODY`.

*NODE

*NODE_RIGID_SURFACE

*NODE_RIGID_SURFACE

Purpose: Define a rigid node and its coordinates in the global coordinate system. These nodes are used to define rigid road surfaces and they have no degrees of freedom. The nodal points are used in the definition of the segments that define the rigid surface. See *CONTACT_RIGID_SURFACE. The nodal point ID must be unique relative to other nodes defined in the *NODE section.

Card Format (I8,3E16.0)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	NID	X	Y	Z						
Type	I	F	F	F						
Default	none	0.	0.	0.						
Remarks										

VARIABLE

DESCRIPTION

NID	Node number
X	x coordinate
Y	y coordinate
Z	z coordinate

***NODE_SCALAR_{OPTION}**

Available options include:

<BLANK>

VALUE

Purpose: Define a scalar nodal point which has one degree-of-freedom. The scalar point ID must be unique relative to other nodes defined in the *NODE section.

Define the following card if and if no option is picked

Card Format (2I8)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	NID	NDOF									
Type	I	I									
Default	none	0									
Remarks											

Define the following card for and only for option VALUE

Card Format (I8,3E16.0,I8)

Card 1 1 2 3 4 6 7 9

Variable	NID	X1	X2	X3	NDOF		
Type	I	F	F	F	I		
Default	none	0	0	0	0		
Remarks							

NODE**NODE_SCALAR**

VARIABLE	DESCRIPTION
NID	Scalar node ID.
NDOF	Number of degrees-of-freedom EQ.0: fully constrained EQ.1: one degree-of-freedom EQ.2: two degrees-of-freedom EQ.3: three degrees-of-freedom
XI	Initial value of Ith degree of freedom.

***NODE**

***NODE_TRANSFORM**

***PARAMETER**

Two keywords are used in this section.

***PARAMETER**

***PARAMETER_EXPRESSION**

*PARAMETER

*PARAMETER

*PARAMETER

Purpose: Define the numerical values of parameter names referenced throughout the input file. The parameter definitions, if used, should be placed at the beginning of the input file following *KEYWORD.

Define as many cards as necessary.

Card 1 1 2 3 4 5 6 7 8

Variable	PRMR1	VAL1	PRMR2	VAL2	PRMR3	VAL3	PRMR4	VAL4
Type	A	I or F						
Default	none	none	none	none	none	none	none	none

Card 2...

Variable	PRMRn	VALn	PRMRn+1	VALn+1		
Type	A	I or F	A	I or F				
Default	none	none	none	none				

VARIABLE

DESCRIPTION

PRMRn

Define the nth parameter in a field of 10. Within this field the first character must be either an "R" for a real number or an "I" for an integer. Lower or upper case for "I" or "R" is okay. Following the type designation, define the name of the parameter using up to, but not exceeding seven characters. For example, when defining a shell thickness named, "SHLTHK", both inputs "RSHLTHK" or "R SHLTHK" can be used and placed anywhere in the field of 10. When referencing SHLTHK in the input file see Remark 1 below.

VALn

Define the numerical value of the n parameter as either a real or integer number consistent with preceding definition for PRMRn.

Remarks:

- Parameters can be referenced anywhere in the input by placing an "&" at the first column of its field followed by the name of the parameter without blanks.

***PARAMETER_EXPRESSION**

***PARAMETER**

***PARAMETER_EXPRESSION**

Purpose: Define the numerical values of parameter names referenced throughout the input file. Like the *PARAMETER keyword, but allows for general algebraic expressions, not simply fixed values.

Define as many cards as necessary.

Card 1 1 2 3 4 5 6 7 8

Variable	PRMR1	EXPRESSION1
Type	A	A
Default	none	none

Card 2...

Variable	PRMRn	EXPRESSIONn
Type	A	A
Default	none	none

VARIABLE

DESCRIPTION

PRMRn

Define the nth parameter in a field of 10. Within this field the first character must be either an "R" for a real number or an "I" for an integer. Lower or upper case for "I" or "R" is okay. Following the type designation, define the name of the parameter using up to, but not exceeding seven characters. For example, when defining a shell thickness named, "SHLTHK", both inputs "RSHLTHK" or "R SHLTHK" can be used and placed anywhere in the field of 10. When referencing SHLTHK in the input file see Remark 1 below.

*PARAMETER

*PARAMETER_EXPRESSION

<u>VARIABLE</u>	<u>DESCRIPTION</u>
EXPRESSION _n	General expression which is evaluated, having the result stored in PRMR _n . The following functions are available: sin, cos, tan, csc, sec, ctn, asin, acos, atan, atan2, sinh, cosh, tanh, asinh, acosh, atanh, min, max, sqrt, mod, abs, sign, int, aint, nint, anint, float, exp, log, log10, float, and general arithmetic expressions involving +, -, *, /, and **. The standard rules regarding operator precedence are obeyed, and nested parentheses are allowed. The expression can reference previously defined parameters (with or without the leading &). The expression can be continued on multiple lines simply by leaving the first 10 characters of the continuation line blank.

Remarks:

1. Parameters can be referenced anywhere in the input by placing an "&" at the first column of its field followed by the name of the parameter without blanks. Expressions can be included in the input when placed between brackets "<>" as long as the total line length does not exceed 80 columns.
2. The integer and real properties of constants and parameters are honored when evaluating expressions. So 2/5 becomes 0, but 2.0/5 becomes 0.4.
3. The sign, atan2, min, max, and mod functions all take two arguments. The others all take only 1.
4. Functions that use an angle as their argument, e.g., sin or cos, assume the angle is in radians.

***PART**

The following keywords are used in this section:

***PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}**

***PART_ADAPTIVE_FAILURE**

***PART_COMPOSITE_{OPTION}**

***PART_MODES**

***PART_SENSOR**

***PART_MOVE**

***PART**

***PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}**

***PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}**

For *OPTION1* the available options are

<BLANK>

INERTIA

REPOSITION

For *OPTION2* the available options are

<BLANK>

CONTACT

For *OPTION3* the available options are

<BLANK>

PRINT

For *OPTION4* the available options are

<BLANK>

ATTACHMENT_NODES

Options 1, 2, 3, and 4 may be specified in any order on the *PART card.

Purpose: Define parts, i.e., combine material information, section properties, hourglass type, thermal properties, and a flag for part adaptivity.

The INERTIA option allows the inertial properties and initial conditions to be defined rather than calculated from the finite element mesh. This applies to rigid bodies, see *MAT_RIGID, only. The REPOSITION option applies to deformable materials and is used to reposition deformable materials attached to rigid dummy components whose motion is controlled by either CAL3D or MADYMO. At the beginning of the calculation each component controlled by CAL3D/MADYMO is automatically repositioned to be consistent with the CAL3D/MADYMO input. However, deformable materials attached to these components will not be repositioned unless this option is used.

The CONTACT option allows part based contact parameters to be used with the automatic contact types a3, 4, a5, a10, 13, a13, 15 and 26, that is

***CONTACT_AUTOMATIC_SURFACE_TO_SURFACE**

***CONTACT_SINGLE_SURFACE,**

***CONTACT_AUTOMATIC_NODES_TO_SURFACE,**

*CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,

*CONTACT_AUTOMATIC_SINGLE_SURFACE,

*CONTACT_AIRBAG_SINGLE_SURFACE,

*CONTACT_ERODING_SINGLE_SURFACE,

*CONTACT_AUTOMATIC_GENERAL.

The default values to use for these contact parameters can be specified on the *CONTACT input section card.

The PRINT option allows user control over whether output data is written into the ASCII files MATSUM and RBDOUT. See *DATABASE_ASCII.

Card 1

Variable	HEADING	
Type	C	
Default	none	
Remarks	1	

Card 2

1 2 3 4 5 6 7 8

Variable	PID	SECID	MID	EOSID	HGID	GRAV	ADPOPT	TMID
Type	I	A8	A8	A8	A8	I	I	A8
Default	none	none	none	0	0	0	0	0

PART**PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}****Additional Cards are required for the INERTIA option. See remarks 3 and 4.**

Card 3 1 2 3 4 5 6 7 8

Variable	XC	YC	ZC	TM	IRCS	NODEID		
Type	F	F	F	F	I	I		

Card 4

Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		

Card 5

Variable	VTX	VTY	VTZ	VRX	VRY	VRZ		
Type	F	F	F	F	F	F		

Optional card required for IRCS=1. Define two local vectors or a local coordinate system ID.

Card 6 1 2 3 4 5 6 7 8

Variable	XL	YL	ZL	XLIP	YLIP	ZLIP	CID	
Type	F	F	F	F	F	F	I	
Remark	2	2	2	2	2	2	none	

An additional Card is required for the REPOSITION option.

Optional 1 2 3 4 5 6 7 8

Variable	CMSN	MDEP	MOVOPT					
Type	I	I	I					

Additional Card is required for the CONTACT option.

WARNING: If FS, FD, DC, and VC are specified they will not be used unless FS is set to a negative value (-1.0) in the *CONTACT section. These frictional coefficients apply only to contact types: SINGLE_SURFACE, AUTOMATIC_GENERAL, AUTOMATIC_SINGLE_SURFACE, AUTOMATIC_NODES_TO_..., AUTOMATIC_SURFACE_..., AUTOMATIC_ONE_WAY_..., and ERODING_SINGLE_SURFACE. Default values are input via *CONTROL_CONTACT input.

Optional 1 2 3 4 5 6 7 8

Variable	FS	FD	DC	VC	OPTT	SFT	SSF	
Type	F	F	F	F	F	F	F	

An additional Card is required for the PRINT option. This option applies to rigid bodies and provides a way to turn off ASCII output in files RBDOUT and MATSUM.

Optional 1 2 3 4 5 6 7 8

Variable	PRBF							
Type	I							

*PART

*PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}

An additional Card is required for the ATTACHMENT_NODES option. All nodes are treated as attachment nodes if this option is not used. Attachment nodes apply to rigid bodies only. The motion of these nodes, which must belong to the rigid body, are updated each cycle. Other nodes in the rigid body are updated only for output purposes. Include all nodes in the attachment node set which interact with the structure through joints, contact, merged nodes, applied nodal point loads, and applied pressure. Include all nodes in the attachment node set if their displacements, accelerations, and velocities are to be written into an ASCII output file. Body force loads are applied to the c.g. of the rigid body.

Optional	1	2	3	4	5	6	7	8
Variable	ANSID							
Type	I							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
HEADING	Heading for the part
PID	Part identification
SECID	Section identification defined in the *SECTION section
MID	Material identification defined in the *MAT section
EOSID	Equation of state identification defined in the *EOS section. Nonzero only for solid elements using an equation of state to compute pressure.
HGID	Hourglass/bulk viscosity identification defined in the *HOURLASS Section: EQ.0: default values are used.
GRAV	Part initialization for gravity loading. This option initializes hydrostatic pressure in the part due to gravity acting on an overburden material. This option applies to brick elements only and must be used with the *LOAD_DENSITY_DEPTH option: EQ.0: all parts initialized, EQ.1: only current material initialized.
ADPOPT	Indicate if this part is adapted or not. (See also *CONTROL_ADAPTIVITY): LT.0: R-adaptive remeshing for 2-D solids, ADOPT gives the load curve ID that defines the element size as a function of time. EQ.0: adaptive remeshing is inactive for this part ID, EQ.1: H-adaptive for 3-D shells. EQ.2: R-adaptive remeshing for 2-D solids, 3-D tetrahedrons and 3-D EFG.

VARIABLE	DESCRIPTION
TMID	Thermal material property identification defined in the *MAT_THERMAL Section. Thermal properties must be specified for all solid, shell, and thick shell parts if a thermal or coupled thermal structural/analysis is being performed. Beams and discrete elements are not considered in thermal analyses. EQ.0: defaults to MID
XC	x-coordinate of center of mass. If nodal point, NODEID, is defined XC, YC, and ZC are ignored and the coordinates of the nodal point, NODEID, are taken as the center of mass.
YC	y-coordinate of center of mass
ZC	z-coordinate of center of mass
TM	Translational mass
IRCS	Flag for inertia tensor reference coordinate system: EQ.0: global inertia tensor, EQ.1: local inertia tensor is given in a system defined by the orientation vectors.
NODEID	Nodal point defining the CG of the rigid body. This node should be included as an extra node for the rigid body; however, this is not a requirement. If this node is free, its motion will not be updated to correspond with the rigid body after the calculation begins.
I _{XX}	I _{XX} , xx component of inertia tensor
I _{XY}	I _{XY} , xy component of inertia tensor (see Remark 4)
I _{XZ}	I _{XZ} , xz component of inertia tensor (see Remark 4)
I _{YY}	I _{YY} , yy component of inertia tensor
I _{YZ}	I _{YZ} , yz component of inertia tensor (see Remark 4)
I _{ZZ}	I _{ZZ} , zz component of inertia tensor
VTX	initial translational velocity of rigid body in x direction
VTY	initial translational velocity of rigid body in y direction
VTZ	initial translational velocity of rigid body in z direction
VRX	initial rotational velocity of rigid body about x axis

PART**PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}**

VARIABLE	DESCRIPTION
VRV	initial rotational velocity of rigid body about y axis
VRZ	initial rotational velocity of rigid body about z axis
XL	x-coordinate of local x-axis. Origin lies at (0,0,0).
YL	y-coordinate of local x-axis
ZL	z-coordinate of local x-axis
XLIP	x-coordinate of vector in local x-y plane
YLIP	y-coordinate of vector in local x-y plane
ZLIP	z-coordinate of vector in local x-y plane
CID	Local coordinate system ID, see *DEFINE_COORDINATE_.... With this option leave fields 1-6 blank.
CMSN	CAL3D segment number/MADYMO system number. See the numbering in the corresponding program.
MDEP	MADYMO ellipse/plane number: GT.0: ellipse number, EQ.0: default, LT.0: absolute value is plane number.
MOVOPT	Flag to deactivate moving for merged rigid bodies, see *CONSTRAINED_RIGID_BODIES. This option allows a merged rigid body to be fixed in space while the nodes and elements of the generated CAL3D/MADYMO parts are repositioned: EQ.0: merged rigid body is repositioned, EQ.1: merged rigid body is not repositioned.
FS	Static coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$
FD	Dynamic coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$
DC	Exponential decay coefficient. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$

VARIABLE	DESCRIPTION
VC	<p>Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{lim} = VC \cdot A_{cont}$. A_{cont} being the area of the segment contacted by the node in contact.</p> <p>The suggested value for VC is to use the yield stress in shear $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_o is the yield stress of the contacted material.</p>
OPTT	Optional contact thickness. This applies to shells only.
SFT	Optional thickness scale factor for PART ID in automatic contact (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
SSF	Scale factor on default slave penalty stiffness for this PART ID whenever it appears in the contact definition. If zero, SSF is taken as unity.
PRBF	<p>Print flag for RBDOUT and MATSUM files.</p> <p>EQ.0: default is taken from the keyword *CONTROL_OUTPUT, EQ.1: write data into RBDOUT file only EQ.2: write data into MATSUM file only EQ.3: do not write data into RBDOUT and MATSUM</p>
ANSID	<p>Attachment node set ID. This option should be used very cautiously and applies only to rigid bodies. The attachment point nodes are updated each cycle whereas other nodes in the rigid body are updated only in the output databases. All loads seen by the rigid body must be applied through this nodal subset or directly to the center of gravity of the rigid body. If the rigid body is in contact this set must include all interacting nodes.</p> <p>EQ.0: All nodal updates are skipped for this rigid body. The null option can be used if the rigid body is fixed in space or if the rigid body does not interact with other parts, e.g., the rigid body is only used for some visual purpose.</p>

Remarks:

1. HEADING default is standard material description, e.g. Material Type 1.
2. The local cartesian coordinate system is defined as described in *DEFINE_COORDINATE_VECTOR. The local z-axis vector is the vector cross product of the x-axis and the in plane vector. The local y-axis vector is finally computed as the vector cross product of the z-axis vector and the x-axis vector. The local coordinate system defined by CID has the advantage that the local system can be defined by nodes in the rigid body which makes repositioning of the rigid body in a preprocessor much easier since the local system moves with the nodal points.

***PART**

***PART_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}**

3. When specifying mass properties for a rigid body using the inertia option, the mass contributions of deformable bodies to nodes which are shared by the rigid body should be considered as part of the rigid body.
4. If the inertia option is used, all mass and inertia properties of the body must be specified for there are no default values. Note that the off-diagonal terms of the inertia tensor are opposite in sign from the products of inertia.
5. The initial velocity of the rigid body may be overwritten by the *INITIAL_VELOCITY card.

See parameter IRIGID on this card.

***PART_ADAPTIVE_FAILURE**

Purpose: This is an option for two-dimensional adaptivity to allow a part that is singly connected to split into two parts. This option is under development and will be generalized in the future to allow the splitting of parts that are multiply connected.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	T						
Type	I	F						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID
T	Thickness. When the thickness of the part reaches this minimum value the part is split into two parts. <i>The value for T should be on the order of the element thickness of a typical element.</i>

*PART

*PART_COMPOSITE

*PART_COMPOSITE_{OPTION}

Available options include:

<BLANK>

CONTACT

Purpose: The following input provides a simplified method of defining a composite material model for shell elements that eliminates the need for user defined integration rules and part ID's for each composite layer. The material ID, thickness, and material angle for each through-thickness integration point of a composite shell are provided below (up to two integration points per card). The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. The total thickness of the composite shell is the sum of the integration point thickness THICK_i; consequently, the shell thickness is assumed to be uniform. When *PART_COMPOSITE is used a section definition, *SECTION_SHELL, and integration rule definition, *INTEGRATION_SHELL, are unnecessary.

The CONTACT option allows part based contact parameters to be used with the automatic contact types a3, 4, a5, a10, 13, a13, 15 and 26, which are listed under the *PART definition above.

Card 1

Variable	HEADING	
Type	C	
Default	none	

Card 2

1 2 3 4 5 6 7 8

Variable	PID	ELFORM	SHRF	NLOC	MAREA	HGID	ADPOPT	
Type	I	I	F	F	F	A8	I	
Default	none	0	0.0	0.0	0.0	0	0	

Additional Card is required for the CONTACT option.

WARNING: If FS, FD, DC, and VC are specified they will not be used unless FS is set to a negative value (-1.0) in the *CONTACT section. These frictional coefficients apply only to contact types: SINGLE_SURFACE, AUTOMATIC_GENERAL, AUTOMATIC_SINGLE_SURFACE, AUTOMATIC_NODES_TO_..., AUTOMATIC_SURFACE_..., AUTOMATIC_ONE_WAY_..., and ERODING_SINGLE_SURFACE. Default values are input via *CONTROL_CONTACT input.

Optional 1 2 3 4 5 6 7 8

Variable	FS	FD	DC	VC	OPTT	SFT	SSF	
Type	F	F	F	F	F	F	F	

The material ID, thickness, and material angle for each through-thickness integration point of a composite shell are provided below (up to two integration points per card). The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. The next “*” card terminates this input.

Card 3 1 2 3 4 5 6 7 8

Variable	MID1	THICK1	B1		MID2	THICK2	B2	
Type	I	F	F		I	F	F	

Cards 4... 1 2 3 4 5 6 7 8

Variable	MID3	THICK3	B3		Etc.			
Type	I	F	F		I	F	F	

VARIABLE

DESCRIPTION

HEADING

Heading for the part

PID

Part ID

VARIABLE	DESCRIPTION
ELFORM	Element formulation options, see Remarks 1 and 2 below: EQ.1: Hughes-Liu, EQ.2: Belytschko-Tsay, EQ.3: BCIZ triangular shell, EQ.4: C ⁰ triangular shell, EQ.6: S/R Hughes-Liu, EQ.7: S/R co-rotational Hughes-Liu, EQ.8: Belytschko-Leviathan shell, EQ.9: Fully integrated Belytschko-Tsay membrane, EQ.10: Belytschko-Wong-Chiang, EQ.11: Fast (co-rotational) Hughes-Liu, EQ.16: Fully integrated shell element (very fast),
SHRF	Shear correction factor which scales the transverse shear stress. The shell formulations in LS-DYNA, with the exception of the BCIZ and DK elements, are based on a first order shear deformation theory that yields constant transverse shear strains which violates the condition of zero traction on the top and bottom surfaces of the shell. The shear correction factor is an attempt to compensate for this error.
NLOC	Location of reference surface for three dimensional shell elements. If nonzero, the offset distance from the plane of the nodal points to the reference surface of the shell in the direction of the shell normal vector is a value $offset = -0.50 \times NLOC \times (average\ shell\ thickness)$. This offset is not considered in the contact subroutines. Alternatively, the offset can be specified by using the OFFSET option in the *ELEMENT_SHELL input section. EQ. 1.0: top surface, EQ. 0.0: mid-surface (default), EQ.-1.0: bottom surface.
MAREA	Non-structural mass per unit area. This is additional mass which comes from materials such as carpeting. This mass is not directly included in the time step calculation.
HGID	Hourglass/bulk viscosity identification defined in the *HOURLASS Section: EQ.0: default values are used.
ADPOPT	Indicate if this part is adapted or not. Also see, *CONTROL_ADAPTIVITY: EQ.0: no adaptivity, EQ.1: H-adaptive for 3-D shells.

VARIABLE	DESCRIPTION
FS	Static coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$
FD	Dynamic coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$
DC	Exponential decay coefficient. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = FD + (FS - FD)e^{-DC v_{rel} }$
VC	Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{lim} = VC \cdot A_{cont}$. A_{cont} being the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_o is the yield stress of the contacted material.
OPTT	Optional contact thickness. This applies to shells only.
SFT	Optional thickness scale factor for PART ID in automatic contact (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
SSF	Scale factor on default slave penalty stiffness for this PART ID whenever it appears in the contact definition. If zero, SSF is taken as unity.
MIDi	Material ID of integration point i , see *MAT_.... Section.
THICKi	Thickness of integration point i .
Bi	Material angle of integration point i .

***PART_MODES**

Purpose: Define mode shapes for a flexible rigid body. Currently, flexible bodies cannot be merged into other flexible bodies or rigid bodies; however, interconnections to other rigid/flexible bodies can use the penalty joint option. The flexible rigid bodies are not implemented with the Lagrange multiplier joint option. The deformations are modeled using the modes shapes obtained experimentally or in a finite element analysis, e.g., NASTRAN.pch file or an LSTC eigout file. These modes should include both constraint and attachment modes. For stress recovery in flexible rigid bodies, use of linear element formulations is recommended. A lump mass matrix is assumed in the implementation. Also see the keyword control card: *CONTROL_RIGID.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	NMFB	FORM	ANSID	FORMAT	KMFLAG	NUPDF	SIGREC
Type	I	I	I	I	I	I	I	

Card 2

Variable	FILENAME
Type	C
Default	none

Define the following cards if and only if KMFLAG=1. Use as many cards as necessary to identify the NMFB kept modes. After NMFB modes are defined no further input is expected.

Cards 3... 1 2 3 4 5 6 7 8

Variable	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7	MODE8
Type	I	I	I	I	I	I	I	I
Default	none	nont	none	nont	none	nont	none	nont

Read optional modal damping cards here. A keyword card (with a "*" in column 1) terminates this input.

Card 1 2 3 4 5 6 7 8

Variable	MSTART	MSTOP	DAMPF					
Type	I	I	F					
Default	none	none	none					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part identification. This part must be a rigid body.
NMFB	Number of kept modes in flexible body. The number of modes in the file, FILENAME, must equal or exceed NMFB. If KMFLAG=0 the first NMFB modes in the file are used.
FORM	Flexible body formulation. See remark 5 below. EQ.0: exact EQ.1: fast
ANSID	Attachment node set ID (optional).
FORMAT	Input format of modal information: EQ.0: NASTRAN.pch file. EQ.1: (not supported) EQ.2: NASTRAN.pch file (LS-DYNA binary version). The binary version of this file is automatically created if a NASTRAN.pch file is read. The name of the binary file is the name of the NASTRAN.pch file but with ".bin" appended. The binary file is smaller and can be read much faster. EQ.3: LS-DYNA d3eigv binary eigenvalue database (see *CONTROL_IMPLICIT_EIGENVALUE). EQ.4: LS-DYNA d3mode binary constraint/attachment mode database (see *CONTROL_IMPLICIT_MODE). EQ.5: Both d3eigv and d3mode databases are input. Database names must be "d3eigv" and "d3mode", and FILENAME below is ignored. NMFB above gives the total number of modes in both databases.
KMFLAG	Kept mode flag. Selects method for identifying modes to keep. EQ.0: the first NMFB modes in the file, FILENAME, are used. EQ.1: define NMFB kept modes with additional input.

VARIABLE	DESCRIPTION
NUPDF	Nodal update flag. If active, an attachment node set, ANSID, must be defined. EQ.0: all nodes of the rigid part are updated each cycle. EQ.1: only attachment nodes are fully updated. All nodes in the body are output based on the rigid body motion without the addition of the modal displacements. For maximum benefit an attachment node set can also be defined with the PART_ATTACHMENT_NODES option. The same attachment node set ID should be used here.
SIGREC	Stress recovery flag. EQ.0: no stress recovery EQ.1: stress recovery only EQ.2: stress recovery and then set the recovery stress as initial stress when switching to flexible body. (Shell formulations 16, 18, 20, 21 and Solid formulation 2) EQ.3: only for shell elform=16, the recovery stress is based on elform=21, and then set the recovery stress as initial stress for elform=16 when switching to flexible body.
FILENAME	The path and name of a file which contains the modes for this rigid body.
MODEn	Keep normal mode, MODEn.
MSTART	First mode for damping, ($1 \leq MSTART \leq NMFB$).
MSTOP	Last mode for damping, MSTOP, ($1 \leq MSTOP \leq NMFB$). All modes between MSTART and MSTOP inclusive are subject to the same modal damping coefficient, DAMPF.
DAMPF	Modal damping coefficient, ζ .

Remarks:

1. The format of the file which contains the normal modes follows the file formats of NASTRAN output for modal information.
2. The mode set typically combines both normal modes and attachment modes. The eigenvalues for the attachment modes are computed from the stiffness and mass matrices.
3. The part ID specified must be either a single rigid body or a master rigid body (see *CONSTRAINED_RIGID_BODIES) which can be made up of many rigid parts.
4. The modal damping is defined by the modal damping coefficient ζ ., where a value of 1.0 equals critical damping. For a one degree of freedom model system, the relationship

between the damping and the damping coefficient is $c = 2\zeta\omega_n m$, where c is the damping, m is the mass, and ω_n is the natural frequency, $\sqrt{k/m}$.

5. There are two formulation options. The first is a formulation that contains all the terms of the flexible body equations, and its cost grows approximately as the square of the number of modes. The second formulation ignores most of the second order terms appearing in the exact equations and its cost grows linearly with the number of modes. Users are responsible for determining which formulation is appropriate for their problems. In general, if the angular velocities are small and if the deflections are small with respect to the geometry of the system it is safe to use the second (faster) formulation.

*PART

*PART_SENSOR

*PART_SENSOR

Purpose: Activate and deactivate parts, based on sensor defined in ELEMENT_SEATBELT_SENSOR. This option applies to discrete beam element only.

Define one card. Card Format (3I10)

Card 1 1 2 3 4 5 6 7 8

Variable	PID	SIDA	ACTIVE					
Type	I	I	I					
Default	0	0	0					

VARIABLE

DESCRIPTION

PID	Part ID, which is controlled by sensor
SIDA	Sensor ID to activate or deactivate part.
ACTIVE	Flag. If zero, the part is active from time zero until a signal is received by the part to deactivate. If one, the part is inactive from time zero and becomes active when a signal is received by the part to activate. The history variables for inactive parts are initialized at time zero.

***PART_MOVE**

Purpose: Translate a part by an incremental displacement in either a local or a global coordinate system. This option currently applies to parts defined either by shell and solid elements. All nodal points of the given part ID are moved. Care must be observed since parts that share boundary nodes with the part being moved must also be moved to avoid severe mesh distortions.

Define one card. Card Format (I8,3E16.0)

Card 1 1 2 3 4 5 6 7 8 9 10

Variable	PID	XMOV	YMOV	ZMOV	CID		
Type	I	F	F	F	I		
Default	none	0.	0.	0.	0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part identification
XMOV	Move shell/solid part ID, PID, in the x-direction by the incremental distance, XMOV.
YMOV	Move shell/solid part ID, PID, in the y-direction by the incremental distance, YMOV.
ZMOV	Move shell/solid part ID, PID, in the z-direction by the incremental distance, ZMOV.
CID	Coordinate system ID to define incremental displacement in local coordinate system. All displacements, XMOV, YMOV, and ZMOV, are with respect to CID. EQ.0: global

***PART**

***PART_MOVE**

***PERTURBATION**

The keyword ***PERTURBATION** provides a way of defining deviations from the designed structure such as buckling imperfections. The perturbation cards in this section are defined in alphabetical order:

***PERTURBATION_NODE**

***PERTURBATION_SHELL_THICKNESS**

*PERTURBATION

*PERTURBATION_NODE

*PERTURBATION_NODE

Purpose: Define a perturbation of the nodal locations over a node set or over the whole model.

Card 1 of 2 Required

Card 1 1 2 3 4 5 6 7 8

Variable	TYPE	NID	SCL	CMP				
Type	I	I	F	I				
Default	1	0	1.0	7				

Define if TYPE is 1 (harmonic field). One or many definitions.

Card 2 1 2 3 4 5 6 7 8

Variable	AMPL	XWL	XOFF	YWL	YOFF	ZWL	ZOFF	
Type	F	F	F	F	F	F	F	
Default	1.0	0.0	0.0	0.0	0.0	0.0	0.0	

Define if TYPE is 2 (fade field). One definition only.

Card 2 1 2 3 4 5 6 7 8

Variable	FADE							
Type	F							
Default	1.0							

PERTURBATION_NODE**PERTURBATION**

Define if TYPE is 3 (file field). One definition only.

Card 2 1 2 3 4 5 6 7 8

Variable	FNAME								
Type	A								
Default	None								

VARIABLE	DESCRIPTION
TYPE	Type of perturbation EQ.1: Harmonic Field EQ.2: Fade out all perturbations at this node set EQ.3: Read perturbations from a file
NID	Node set ID. Specify 0 to perturb all the nodes in the model.
SCL	Scale factor
CMP	Component EQ.1: x coordinate EQ.2: y coordinate EQ.3: z coordinate EQ.4: x and y coordinate EQ.5: y and z coordinate EQ.6: z and x coordinate EQ.7: x, y, and z coordinate
AMPL	Amplitude of the harmonic perturbation
XWL	x wavelength of the harmonic field
XOFF	x offset of harmonic field
YWL	y wavelength of the harmonic field
YOFF	y offset of harmonic field
ZWL	z wavelength of the harmonic field
ZOFF	z offset of harmonic field
FADE	Distance over which all *PERTURBATION_NODE are faded to zero
FNAME	Name of file containing the perturbation definitions

Remarks:

1. The perturbation can be viewed in LS-PREPOST. The current version of LS-DYNA creates files named `pert_node_x/y/z/res`, which can be viewed as user-defined fringe plots.

2. The harmonic perturbation is

$$P_{CMP}(x, y, z) = SCL * AMPL \left[\sin \left(2\pi \frac{x + XOFF}{XWL} \right) + \sin \left(2\pi \frac{y + YOFF}{YWL} \right) + \sin \left(2\pi \frac{z + ZOFF}{ZWL} \right) \right]$$

Note that the harmonic perturbations can sum to values greater than $SCL * AMPL$.

3. The fade perturbation is $p'(x, y, z) = SCL \left(1 - \frac{1}{e^{ax'}} \right) p(x, y, z)$ with x' the shortest distance to a node in the node set specified.

4. The file `FNAME` must contain the perturbation in the LS-DYNA keyword format. This file can be created using the LS-PREPOST Output capability. The data must be arranged into two columns with the first column being the node ids. Lines starting with the character `$` will be ignored.

PERTURBATION_SHELL_THICKNESS**PERTURBATION*****PERTURBATION_SHELL_THICKNESS**

Purpose: Define a perturbation of the shell thicknesses over an element set or over the whole model.

Card 1 of 2 Required

Card 1 1 2 3 4 5 6 7 8

Variable	TYPE	EID	SCL					
Type	I	I	F					
Default	1	0	1.0					

Define if TYPE is 1 (harmonic field). One or many definitions.

Card 2 1 2 3 4 5 6 7 8

Variable	AMPL	XWL	XOFF	YWL	YOFF	ZWL	ZOFF	
Type	F	F	F	F	F	F	F	
Default	1.0	0.0	0.0	0.0	0.0	0.0	0.0	

Define if TYPE is 2 (fade field). One definition only.

Card 2 1 2 3 4 5 6 7 8

Variable	FADE							
Type	F							
Default	1.0							

*PERTURBATION

*PERTURBATION_SHELL_THICKNESS

VARIABLE	DESCRIPTION
TYPE	Type of perturbation EQ.1: Harmonic Field EQ.2: Fade out all perturbations at this element set
EID	Element set ID. Specify 0 to perturb all the elements in the model.
SCL	Scale factor
AMPL	Amplitude of the harmonic perturbation
XWL	x wavelength of the harmonic field
XOFF	x offset of harmonic field
YWL	y wavelength of the harmonic field
YOFF	y offset of harmonic field
ZWL	z wavelength of the harmonic field
ZOFF	z offset of harmonic field
FADE	Distance over which all *PERTURBATION_SHELL_THICKNESS are faded to zero

Remarks:

1. The perturbation can be viewed in LS-PREPOST. The current version of LS-DYNA creates a file named *pert_shell_thickness*, which can be viewed as a user-defined fringe plots.

2. The harmonic perturbation is

$$p(x, y, z) = SCL * AMPL \left[\sin \left(2\pi \frac{x + XOFF}{XWL} \right) + \sin \left(2\pi \frac{y + YOFF}{YWL} \right) + \sin \left(2\pi \frac{z + ZOFF}{ZWL} \right) \right]$$

Note that the harmonic perturbations can sum to values greater than $SCL * AMPL$.

3. The fade perturbation is $p'(x, y, z) = SCL \left(1 - \frac{1}{e^{ax'}} \right) p(x, y, z)$ with x' the shortest distance to an element in the element set specified.

***RAIL**

Two keywords are defined in this section.

***RAIL_TRACK**

***RAIL_TRAIN**

*RAIL

*RAIL_TRACK

*RAIL_TRACK

Purpose: Wheel-rail contact algorithm intended for railway applications but can also be used for other purposes. The wheel nodes (defined on *RAIL_TRAIN) represent the contact patch between wheel and rail. A penalty method is used to constrain the wheel nodes to slide along the track. A track consists of two rails, each of which is defined by a set of beam elements.

Card 1 of 2

Card 1 1 2 3 4 5 6 7 8

Variable	ID	BSETID1	NORGN1	LCUR1	OSET1	SF1	GA1	
Type	I	I	I	I	F	F	F	
Default	none	None	None	None	0.0	1.0	0.0	

Card 2

Variable	BLANK	BSETID2	NORGN2	LCUR2	OSET2	SF2	GA2	
Type	-	I	I	I	F	F	F	
Default	-	None	None	None	0.0	1.0	0.0	

VARIABLE

DESCRIPTION

ID	Track ID
BSETID1,2	Beam set ID for rails 1 and 2 containing all beam elements that make up the rail, see *SET_BEAM.
NORGN1,2	Reference node at one end of each rail, used as the origin for the roughness curve. The train will move in a direction away from this node.
LCUR1,2	Load curve ID (see *DEFINE_CURVE) defining track roughness (vertical displacement from line of beam elements) of the rail as a function of distance from the reference node NORIGIN. Distance from reference node on x-axis of curve, roughness on y-axis. Default: no roughness.

VARIABLE	DESCRIPTION
OSET1,2	Origin of curve LCUR is shifted by distance OSET from the reference node.
SF1,2	Roughness values are scaled by SF. Default: 1.0.
GA1,2	Shear stiffness of rail per unit length (used to calculate local rail shear deformation within each beam element). GA = shear modulus x cross-sectional area. Default: local shear deformation is ignored.

Remarks:

*RAIL_TRACK and *RAIL_TRAIN were written by Arup to represent wheel-rail contact. They have been used to generate loading on models of bridges for vibration predictions, stress calculations and for estimating accelerations experienced by passengers. Other non-railway uses are possible: the algorithm causes the “train” nodes to follow the line defined by the “rail” beam elements and transfers forces between them. In some cases (especially vibration modeling), double precision versions of LS-DYNA may give superior results because of the small relative deflections between wheel and rail.

Track modeling

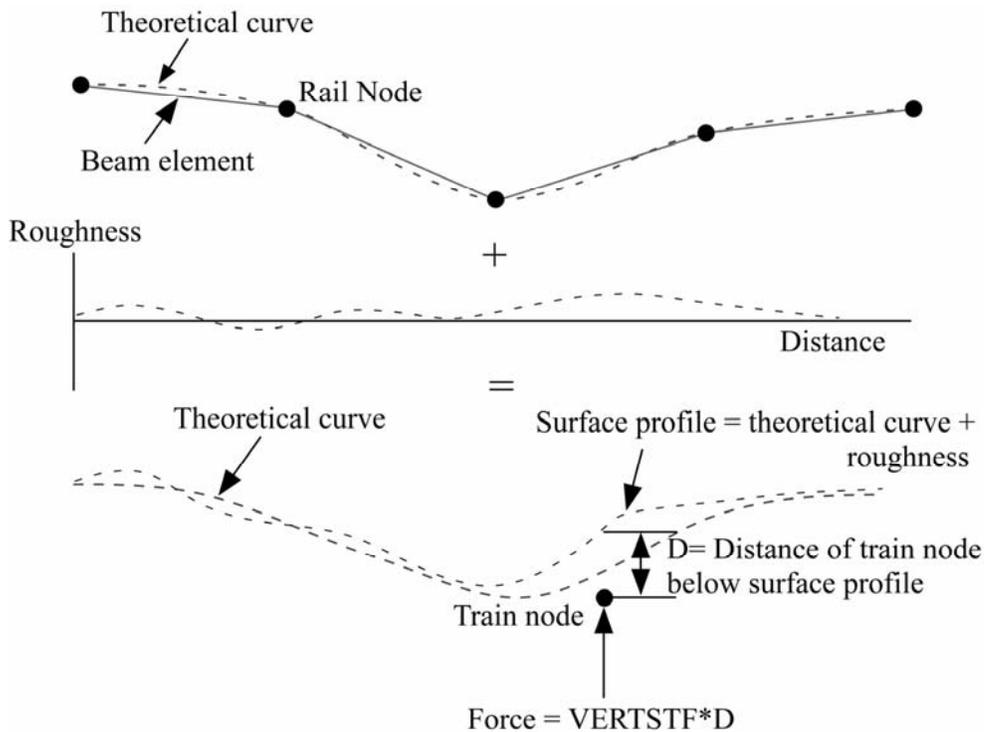
The rails of the track should be modeled by two parallel lines of beam elements. The track can be curved or straight and the rails can be modeled as deformable or rigid. If required, rail pads, sleepers and ballast may also be modeled – typically with spring, damper and beam elements. It is also possible to use this algorithm to control the motion of simple road vehicle models: beam element “rails” made of null material can be embedded in the road surface. It is recommended that the mesh size of the two rails should be similar: LS-DYNA calculates a local coordinate system for each train node based on the alignment of the currently contacted beam element and the nearest node on the other rail.

Because wheel-rail contact stiffness is generally very high, and wheel masses are large, small deviations from a straight line or smooth curve can lead to large transient forces. It is recommended that great care be taken in generating and checking the geometry for the track, especially where the track is curved. Some pre-processors write the coordinates with insufficient precision to the LS-DYNA input file, and this can cause unintended roughness in the geometry. For the same reason, if the line of the track were taken as straight between nodes, spurious forces would be generated when the wheel passes from one rail element to the next. This is avoided because the *RAIL algorithm calculates a theoretical curved centerline for the rail element to achieve continuity of slope from one element to the next. Where the length of the rail elements is similar to or shorter than the maximum section dimension, shear deformation may be significant and it is possible to include this in the theoretical centerline calculation to further reduce spurious forces at the element boundaries (inputs GA1, GA2).

Roughness (small deviations in the vertical profile from a perfect straight line) does exist in real life and is a principal source of vibration. *RAIL allows the roughness to be modeled by a load curve giving the vertical deviation (in length units) of the rail surface from the theoretical centerline of the beam elements as a function of distance along the track from the origin node of the rail. The roughness curve is optional. Ideally, roughness profiles measured from both rails of

the same piece of track should be used so that the relationship between bump and roll modes is correctly captured.

Whether roughness is included or not, it is important to select as the origin nodes (NORIGIN1 and NORIGIN2) the nodes at the end of the rails away from which the train will be traveling. The train can start at any point along the rails but must travel away from the origin nodes.



Train modeling

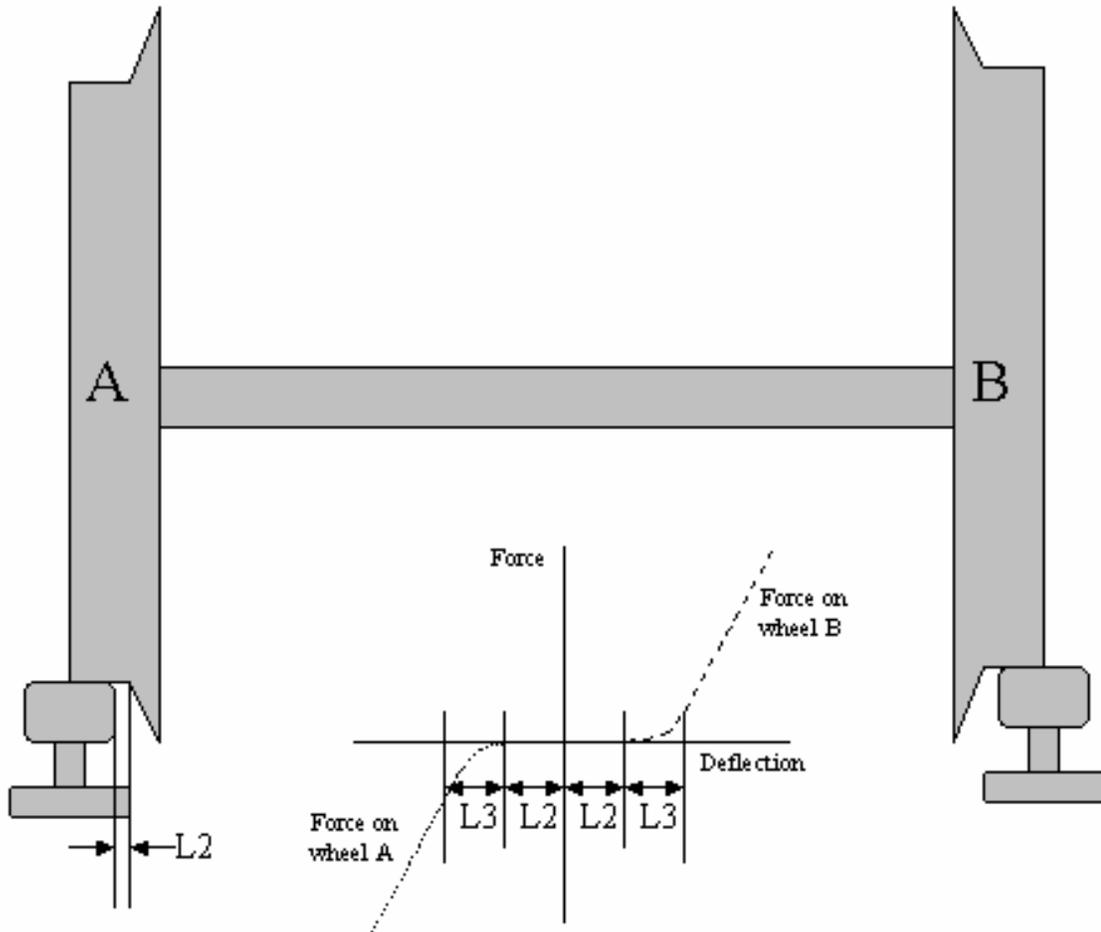
The vehicle models are typically modeled using spring, damper and rigid elements, or simply a point mass at each wheel position. Each node in the set referred to on *RAIL_TRAIN represents the contact patch of one wheel (note: not the center of the wheel). These nodes should be initially on or near the line defined by either of the two rails. LS-DYNA will move the train nodes initially onto the rails to achieve the correct initial wheel-rail forces. If the results are viewed with magnified displacements, the initial movements can appear surprising.

Wheel roughness input is available. This will be applied in addition to track roughness. The input curve must continue for the total rolled distance – it is not assumed to repeat with each wheel rotation. This is to avoid problems associated with ensuring continuity between the start

and end of the profile around the wheel circumference, especially since the profiles might be generated from roughness spectra rather than taken directly from measured data.

Wheel-rail interface

The wheel-rail interface model is a simple penalty function designed to ensure that the train nodes follow the line of the track. It does not attempt to account for the shape of the rail profile. Vertical and lateral loads are treated independently. For this reason, the algorithm is not suitable for rail vehicle dynamics calculations.



Wheel-rail contact stiffness is input on *RAIL_TRAIN. For vertical loads, a linear force-deflection relationship is assumed in compression; no tensile force is generated (this corresponds to the train losing contact with the rail). Typical contact stiffness is 2MN/mm. Lateral deflections away from the theoretical centerline of the rail beams are also penalized by a linear force-deflection relationship. The lateral force is applied only to wheels on the side towards which the train has displaced (corresponding to wheel flanges that run inside the rails). Optionally, a “gap” can be defined (input parameter L2) such that the wheel set can drift laterally by L2 length units before any lateral force is generated. A further option is to allow smooth transition between “gap” and “contact” by means of a transition distance (input parameter L3). Generally, with straight tracks a simple linear stiffness is sufficient. With curved tracks, a

reasonable gap and transition distance should be defined to avoid unrealistic forces being generated in response to small inaccuracies in the distance between the rails. Gravity loading is expected, in order to maintain contact between rail and wheel. This is normally applied by an initial phase of dynamic relaxation. To help achieve convergence quickly, or in some cases avoid the need for dynamic relaxation altogether, the initial force expected on each train node can be input (parameter FINIT on *RAIL_TRAIN). LS-DYNA positions the nodes initially such that the vertical contact force will be FINIT at each node. If the suspension of the rail vehicles is modeled, it is recommended that the input includes carefully calculated precompression of the spring elements; if this is not done, achieving initial equilibrium under gravity loading can be very time consuming.

The *RAIL algorithm ensures that the train follows the rails, but does not provide forward motion. This is generally applied using *INITIAL_VELOCITY, or for straight tracks, *BOUNDARY_PRESCRIBED_MOTION.

Output

LS-DYNA generates an additional ASCII output file **train_force_n**, where *n* is an integer updated to avoid overwriting any existing files. The file contains the forces on each train node, output at the same time intervals as the binary time history file (DT on *DATABASE_BINARY_D3THDT).

Checking

It is recommended that track and train models be tested separately before adding the *RAIL cards. Check that the models respond stably to impulse forces and that they achieve equilibrium under gravity loading. The majority of problems we have encountered have been due to unstable behavior of train or track. Often, these are first detected by the *RAIL algorithm and an error message will result.

***RAIL_TRAIN**

Purpose: Define train properties. A train is defined by a set of nodes in contact with a rail defined by *RAIL_TRACK.

Card 1 of 2

Card 1 1 2 3 4 5 6 7 8

Variable	ID	NSETID	(omit)	FINIT	(omit)	TRID	LCUR	OFFS
Type	I	I	F	F	F	I	I	F
Default	none	None	0.0	0.0	0.0	0	None	0.0

Card 2

Variable	VERTSTF	LATSTF	V2	V3	L2	L3		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID	Train ID
NSETID	Node set ID containing all nodes that are in contact with rails.
(omit)	Unused variable – leave blank.
FINIT	Estimate of initial vertical force on each wheel (optional) – speeds up the process of initial settling down under gravity loading.
(omit)	Unused variable – leave blank.
TRID	ID of track for this train, see *RAIL_TRACK.
LCUR	Load curve ID (see *DEFINE_CURVE) containing wheel roughness (distance of wheel surface away from perfect circle) vs. distance traveled. The curve does not repeat with each rotation of the wheel – the last point should be at a greater distance than the train is expected to travel. Default: no wheel roughness.

VARIABLE	DESCRIPTION
OFFS	Offset distance used to generate different roughness curves for each wheel from the roughness curve LCUR. The curve is offset on the x-axis by a different whole number multiple of OFFS for each wheel.
VERTSTF	Vertical stiffness of rail contact.
LATSTF	Lateral stiffness of rail contact.
V2,V3	Unused variables – leave blank.
L2	Lateral clearance from rail to wheel rim. Lateral force is applied to a wheel only when it has moved more than L2 away from the other rail, i.e. the wheel rims are assumed to be near the inner face of the rail.
L3	Further lateral distance before full lateral stiffness applies (force-deflection curve follows a parabola up to this point).

***RIGIDWALL**

Two keywords are used in this section to define rigid surfaces:

***RIGIDWALL_GEOMETRIC_OPTION_{OPTION}_{OPTION}**

***RIGIDWALL_PLANAR_{OPTION}_{OPTION}_{OPTION}**

The RIGIDWALL option provides a simple way of treating contact between a rigid surface and nodal points of a deformable body, called slave nodes. Slave nodes which belong to rigid parts are not, in general, checked for contact with only one exception. The RIGIDWALL_PLANAR option may be used with nodal points of rigid bodies if the planar wall defined by this option is fixed in space and the RWPNAL parameter is set to a positive nonzero value on the control card, *CONTROL_CONTACT.

When the rigid wall defined in this section moves with a prescribed motion, the equations of rigid body mechanics are not involved. For a general rigid body treatment with arbitrary surfaces and motion, refer to the *CONTACT_ENTITY definition. The *CONTACT_ENTITY option is for treating contact between rigid and deformable surfaces only.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
RWID	Rigid wall ID. This must be a unique number.
HEADING	Rigid wall descriptor. It is suggested that unique descriptions be used.

For GEOMETRIC options:

- Cards 1 and 2 are required for all geometric shapes.
- Card 3 is required, but is dependent upon which shape is specified.
- Optional Card A is required if MOTION is specified.

Card 1 - Required for all shape types

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	NSIDEX	BOXID	BIRTH	DEATH			
Type	I	I	I	F	F			
Default	none	0	0	0.	1.0E+20			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NSID	Nodal set ID containing slave nodes, see <i>*SET_NODE_OPTION</i> : EQ.0: all nodes are slave to rigid wall.
NSIDEX	Nodal set ID containing nodes that exempted as slave nodes, see <i>*SET_NODE_OPTION</i> .
BOXID	If defined, only nodes in box are included as slave nodes to rigid wall.
BIRTH	Birth time of rigid wall. The time values of the load curves that control the motion of the wall are offset by the birth time.
DEATH	Death time of rigid wall. At this time the wall is deleted from the calculation. If dynamic relaxation is active at the beginning of the calculation and if BIRTH=0.0, the death time is ignored during the dynamic relaxation.

*RIGIDWALL

*RIGIDWALL_GEOMETRIC

Card 2 - Required for all shape types.

Card 2	1	2	3	4	5	6	7	8
Variable	XT	YT	ZT	XH	YH	ZH	FRIC	
Type	F	F	F	F	F	F	F	
Default	0.	0.	0.	0.	0.	0.	0.	
Remarks								

VARIABLE

DESCRIPTION

XT	x-coordinate of tail of any outward drawn normal vector, \mathbf{n} , originating on wall (tail) and terminating in space (head), see Figure 28.1.
YT	y-coordinate of tail of normal vector \mathbf{n}
ZT	z-coordinate of tail of normal vector \mathbf{n}
XH	x-coordinate of head of normal vector \mathbf{n}
YH	y-coordinate of head of normal vector \mathbf{n}
ZH	z-coordinate of head of normal vector \mathbf{n}
FRIC	Interface friction: EQ.0.0: frictionless sliding after contact, EQ.1.0: stick condition after contact, 0.<FRIC<1.: Coulomb friction coefficient.

Card 3 - Required if FLAT is specified after the keyword.

A plane with a finite size or with an infinite size can be defined, see Figure 28.1. The vector **m** is computed as the vector cross product **n X l**. The origin, which is the tail of the normal vector, is the corner point of the finite size plane.

Card 3 1 2 3 4 5 6 7 8

Variable	XHEV	YHEV	ZHEV	LENL	LENM			
Type	F	F	F	F	F			
Default	0.	0.	0.	infinity	infinity			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XHEV	x-coordinate of head of edge vector l , see Figure 28.1.
YHEV	y-coordinate of head of edge vector l
ZHEV	z-coordinate of head of edge vector l
LENL	Length of l edge. A zero value defines an infinite size plane.
LENM	Length of m edge. A zero value defines an infinite size plane.

Card 3 - Required if PRISM is specified after the keyword.

The description of the definition of a plane with finite size is enhanced by an additional length in the direction negative to **n**, see Figure 28.1.

Card 3	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM	LENP		
Type	F	F	F	F	F	F		
Default	none	0.	0.	infinity	infinity	infinity		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XHEV	x-coordinate of head of edge vector l , see Figure 28.1.
YHEV	y-coordinate of head of edge vector l
ZHEV	z-coordinate of head of edge vector l
LENL	Length of l edge. A zero value defines an infinite size plane.
LENM	Length of m edge. A zero value defines an infinite size plane.
LENP	Length of prism in the direction negative to n , see Figure 28.1.

Card 3 - Required if CYLINDER is specified after the keyword.

The tail of **n** specifies the top plane of the cylinder. The length is defined in the direction negative to **n**. See Figure 28.1.

Card 3 1 2 3 4 5 6 7 8

Variable	RADCYL	LENCYL						
Type	F	F						
Default	none	infinity						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
RADCYL	Radius of cylinder
LENCYL	Length of cylinder, see Figure 28.1. Only if a value larger than zero is specified is a finite length assumed.

*RIGIDWALL

*RIGIDWALL_GEOMETRIC

Card 3 - Required if SPHERE is specified after the keyword.

The center of the sphere is identical to the tail of **n**, see Figure 28.1.

Card 3 1 2 3 4 5 6 7 8

Variable	RADSPH							
Type	F							
Default	0.							

VARIABLE

DESCRIPTION

RADSPH

Radius of sphere

Optional Card A - Required if MOTION is specified after the keyword.

Optional Card A	1	2	3	4	5	6	7	8
Variable	LCID	OPT	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	none	none	none	none			

VARIABLE**DESCRIPTION**

LCID	Stonewall motion curve number, see *DEFINE_CURVE.
OPT	Type of motion: EQ.0: velocity specified, EQ.1: displacement specified.
VX	x-direction cosine of velocity/displacement vector
VY	y-direction cosine of velocity/displacement vector
VZ	z-direction cosine of velocity/displacement vector

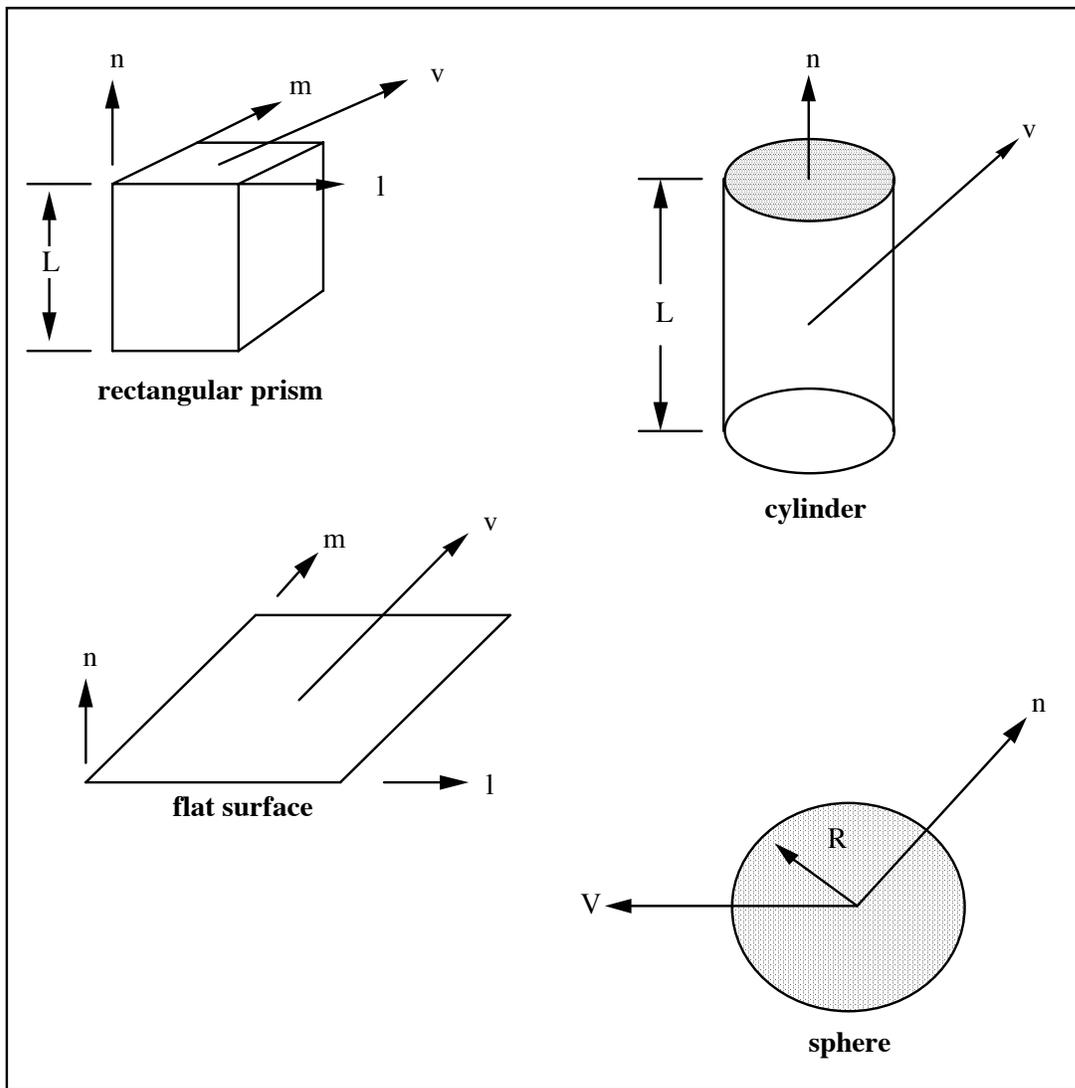


Figure 28.1. Vector n determines the orientation of the generalized stonewalls. For the prescribed motion options the wall can be moved in the direction V as shown.

*RIGIDWALL

*RIGIDWALL_PLANAR

*RIGIDWALL_PLANAR_{OPTION}_{OPTION}_{OPTION}

Available options include:

<BLANK>

ORTHO

FINITE

MOVING

FORCES

The ordering of the options in the input below must be observed but the ordering of the options on the command line is unimportant, i.e.; the **ORTHO** card is first, the **FINITE** definition card below must precede the **MOVING** definition card, and the **FORCES** definition card should be last. The **ORTHO** option does not apply if the **MOVING** option is used. If an ID number is specified the additional option is available:

ID

If active, the ID card is the first card following the keyword.

Purpose: Define planar rigid walls with either finite or infinite size (**FINITE**). Orthotropic friction can be defined (**ORTHO**). Also, the plane can possess a mass and an initial velocity (**MOVING**); otherwise, the wall is assumed to be stationary. The **FORCES** option allows the specification of segments on the rigid walls on which the contact forces are computed. In order to achieve a more physical reaction related to the force versus time curve, the **SOFT** value on the **FORCES** card can be specified.

ID Card Define if and only if ID option is active.

Card 1 1 2 3 4 5 6 7 8

Variable	RWID								
Type	I								
Default	none								

VARIABLE

DESCRIPTION

RWID

Rigid wall ID. Up to 8 characters can be used.

- Cards 1 and 2 are required.
- Optional Cards A and B are required if ORTHO is specified.
- Optional Card C is required if FINITE is specified.
- Optional Card D is required if MOVING is specified.
- Optional Card E is required if FORCES is specified.

Required.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	NSIDEX	BOXID	OFFSET	BIRTH	DEATH	RWKSF	
Type	I	I	I	F	F	F	F	
Default	none	0	0	0.	0.	1.0E+20	1.0	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NSID	Nodal set ID containing slave nodes, see <i>*SET_NODE_OPTION</i> : EQ.0: all nodes are slave to rigid wall.
NSIDEX	Nodal set ID containing nodes that exempted as slave nodes, see <i>*SET_NODE_OPTION</i> .
BOXID	All nodes in box are included as slave nodes to rigid wall, see <i>*DEFINE_BOX</i> . If options NSID or NSIDEX are active then only the subset of nodes activated by these options are checked to see if they are within the box.
OFFSET	All nodes within a normal offset distance, OFFSET, to the rigid wall are included as slave nodes for the rigid wall. If options NSID, NSIDEX, or BOXID are active then only the subset of nodes activated by these options are checked to see if they are within the offset distance. This option applies to the PLANAR wall only.
BIRTH	Birth time of rigid wall. The time values of the load curves that control the motion of the wall are offset by the birth time.
DEATH	Death time of rigid wall. At this time the wall is deleted from the calculation. If dynamic relaxation is active at the beginning of the calculation and if BIRTH=0.0, the death time is ignored during the dynamic relaxation.
RWKSF	Stiffness scaling factor. If RWKSF is also specified in <i>*CONTROL_CONTACT</i> , the stiffness is scaled by the product of the two values.

*RIGIDWALL

*RIGIDWALL_PLANAR

Required.

Card 2 1 2 3 4 5 6 7 8

Variable	XT	YT	ZT	XH	YH	ZH	FRIC	WVEL
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE

DESCRIPTION

XT	x-coordinate of tail of any outward drawn normal vector, n , originating on wall (tail) and terminating in space (head), see Figure 28.3.
YT	y-coordinate of tail of normal vector n
ZT	z-coordinate of tail of normal vector n
XH	x-coordinate of head of normal vector n
YH	y-coordinate of head of normal vector n
ZH	z-coordinate of head of normal vector n
FRIC	Interface friction: EQ.0.0: frictionless sliding after contact, EQ.1.0: no sliding after contact, 0.<FRIC<1.: Coulomb friction coefficient. EQ.2.0: node is welded after contact with frictionless sliding. Welding occurs if and only if the normal value of the impact velocity exceeds the critical value specified by WVEL. EQ.3.0: node is welded after contact with no sliding. Welding occurs if and only if the normal value of the impact velocity exceeds the critical value specified by WVEL.
WVEL	Critical normal velocity at which nodes weld to wall (FRIC = 2 or 3).

Optional Cards A and B - Required if ORTHO is specified after the keyword.

See Figure 28.2 for the definition of orthotropic friction.

Optional Card A	1	2	3	4	5	6	7	8
Variable	SFRICA	SFRICB	DFRICA	DFRICB	DECAYA	DECAYB		
Type	F	F	F	F	F	F		
Default	0.	0.	0	0	0.	0.		

Optional Card B								
Variable	NODE1	NODE2	D1	D2	D3			
Type	I	I	F	F	F			
Default	0.	0.	0	0	0.			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SFRICA	Static friction coefficient in local a-direction, μ_{sa} , see Figure 28.2
SFRICB	Static friction coefficient in local b-direction, μ_{sb}
DFRICA	Dynamic friction coefficient in local a-direction, μ_{ka}
DFRICB	Dynamic friction coefficient in local b-direction, μ_{kb}
DECAYA	Decay constant in local a-direction, d_{va}
DECAYB	Decay constant in local b-direction, d_{vb}
NODE1	Node 1, alternative to definition with vector d below. See Figure 28.2. With the node definition the direction changes if the nodal pair rotates.
NODE2	Node 2
D1	d_1 , x-component of vector, alternative to definition with nodes above. See Figure 28.2. This vector is fixed as a function of time.

VARIABLE	DESCRIPTION
D2	d ₂ , y-component of vector
D3	d ₃ , z-component of vector

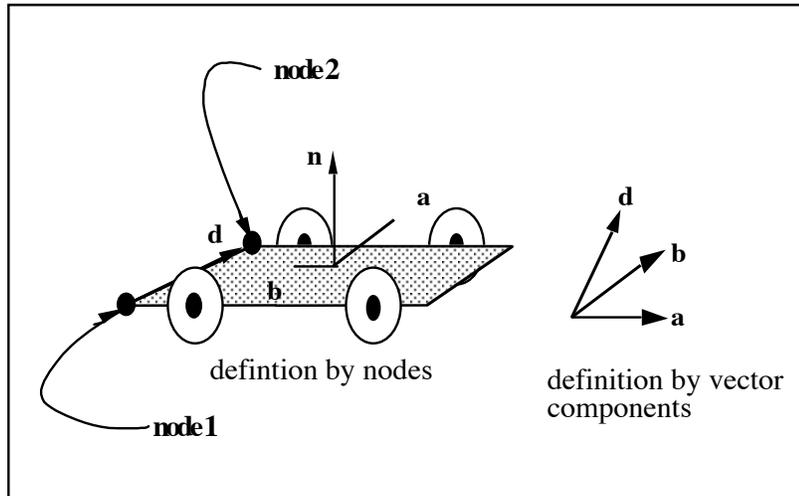


Figure 28.2. Definition of orthotropic friction vectors. The two methods of defining the vector, **d**, are shown. If vector **d** is defined by nodes 1 and 2, the local coordinate system may rotate with the body which contains the nodes; otherwise, **d** is fixed in space, thus on the rigid wall, and the local system is stationary.

Remarks:

1. The coefficients of friction are defined in terms of the static, dynamic and decay coefficients and the relative velocities in the local a and b directions as

$$\mu_a = \mu_{ka} + (\mu_{sa}\mu_{ka})e^{d_{va}V_{relative,a}}$$

$$\mu_b = \mu_{kb} + (\mu_{sb}\mu_{kb})e^{d_{vb}V_{relative,b}}$$

2. Orthotropic rigid walls can be used to model rolling objects on rigid walls where the frictional forces are substantially higher in a direction transverse to the rolling direction. To use this option define a vector **d** to determine the local frictional directions via:

$$\underline{b} = \underline{n} \times \underline{d} \text{ and that } \underline{a} = \underline{b} \times \underline{n}$$

where **n** is the normal vector to the rigid wall. If **d** is in the plane of the rigid wall, then **a** is identical to **d**.

Optional Card C - Required if FINITE is specified after the keyword.

See Figure 28.3. The **m** vector is computed as the vector cross product $\mathbf{m}=\mathbf{n} \times \mathbf{l}$. The origin, the tail of the normal vector, is taken as the corner point of the finite size plane.

Optional Card C	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM			
Type	F	F	F	F	F			
Default	0.	0.	0.	infinity	infinity			

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XHEV	x-coordinate of head of edge vector l , see Figure 28.3.
YHEV	y-coordinate of head of edge vector l
ZHEV	z-coordinate of head of edge vector l
LENL	Length of l edge
LENM	Length of m edge

*RIGIDWALL

*RIGIDWALL_PLANAR

Optional Card D - Required if MOVING is specified after keyword.

Note: The MOVING option is not compatible with the ORTHO option.

Optional Card D	1	2	3	4	5	6	7	8
Variable	MASS	V0						
Type	F	F						
Default	none	0.						

VARIABLE

DESCRIPTION

MASS

Total mass of stonewall

V0

Initial velocity of stonewall in direction of defining vector, **n**

Optional Card E - Required if FORCES is specified after the keyword.

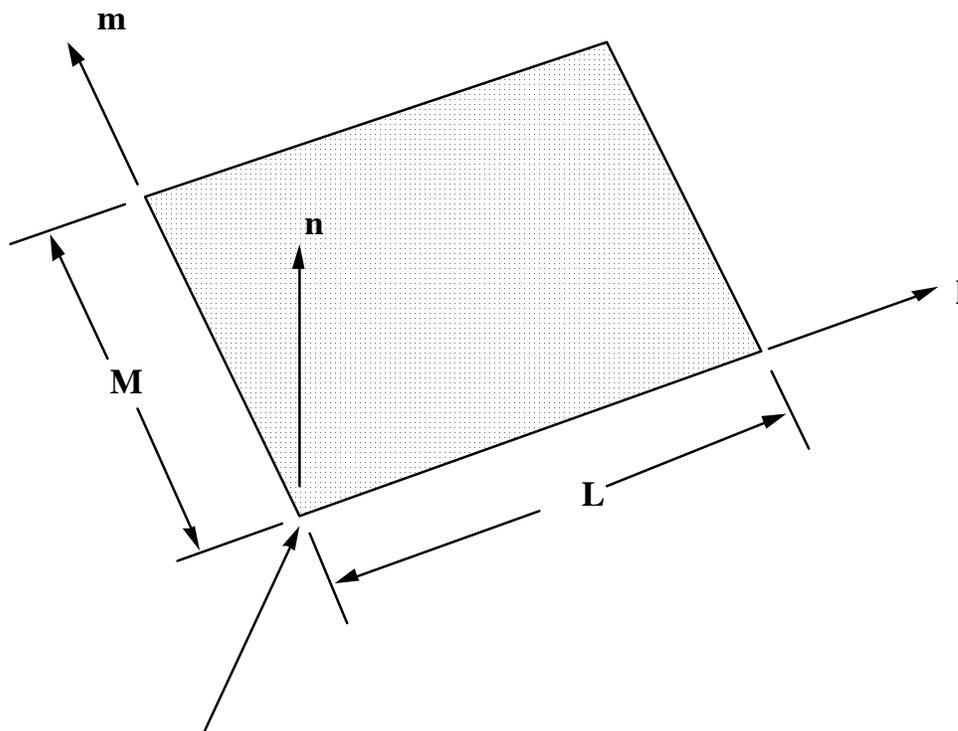
This option allows the force distribution to be monitored on the plane. Also four points can be defined for visualization of the rigid wall. A shell or membrane element must be defined with these four points as the connectivity for viewing in LS-PREPOST.

Optional Card E	1	2	3	4	5	6	7	8
Variable	SOFT	SSID	N ₁	N ₂	N ₃	N ₄		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		
Remarks		1	2					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SOFT	Number of cycles to zero relative velocity to reduce force spike
SSID	Segment set identification number for defining areas for force output, see *SET_SEGMENT and remark 1 below.
N1-N4	Optional node for visualization

Remarks:

1. The segment set defines areas for computing resultant forces. These segments translate with the moving stonewall and allow the forced distribution to be determined. The resultant forces are written in file "RWFORC."
2. These four nodes are for visualizing the movement of the wall, i.e., they move with the wall. To view the wall in LS-PREPOST it is necessary to define a single shell element with these four nodes as its connectivity. The single element must be deformable (non rigid) or else the segment will be treated as a rigid body and the nodes will have their motion modified independently of the stonewall.



Tail of normal vector is the origin and corner point if extent of stonewall is finite.

Figure 28.3. Vector **n** is normal to the stonewall. An optional vector **l** can be defined such that $\mathbf{m}=\mathbf{n} \times \mathbf{l}$. The extent of the stonewall is limited by defining **L** (LENL) and **M** (LENM). A zero value for either of these lengths indicates that the stonewall is infinite in that direction.

***RIGIDWALL_PLANAR**

***RIGIDWALL**

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *RIGIDWALL_PLANAR_MOVING_FORCES
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define a moving planar rigid wall:
$   - that is parallel to the y-z plane starting at x = 250 mm
$   - with an initial velocity of 8.94 mm/ms in the negative z-direction
$   - that has a mass of 800 kg
$   - which prevents all nodes in the model from penetrating the wall
$   - with a friction coefficient for nodes sliding along the wall of 0.1
$   - track the motion of the wall by creating a node (numbered 99999)
$     at the tail of the wall and assigning the node to move with the wall
$
$
$RIGIDWALL_PLANAR_MOVING_FORCES
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   nsid   nsidex   boxid
$       0         0         0
$
$   xt      yt      zt      xh      yh      zh      fric
$   250.0   0.0     0.0     0.0     0.0     0.0     0.1
$
$ SW mass   SW vel
$ 800.00    8.94
$
$   soft     ssid    node1    node2    node3    node4
$     0       0      99999
$
$
$*NODE
$
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   nid      x        y        z        tc     rc
$ 99999      250.0     0.0      0.0      0       0
$
$
$*DATABASE_HISTORY_NODE
$ Define nodes that output into nodout
$   id1      id2      id3
$...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$ 99999
$
$*DATABASE_NODOUT
$   dt
$     0.1
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

***RIGIDWALL**

***RIGIDWALL_PLANAR**

***SECTION**

In this section, the element formulation, integration rule, nodal thicknesses, and cross sectional properties are defined. All section identifiers (SECID's) defined in this section must be unique, i.e., if a number is used as a section ID for a beam element then this number cannot be used again as a section ID for a solid element. The keyword cards in this section are defined in alphabetical order:

***SECTION_BEAM**

***SECTION_DISCRETE**

***SECTION_POINT_SOURCE**

***SECTION_POINT_SOURCE_MIXTURE**

***SECTION_SEATBELT**

***SECTION_SHELL_{OPTION}**

***SECTION_SOLID_{OPTION}**

***SECTION_SPH_{OPTION}**

***SECTION_TSHELL**

The location and order of these cards in the input file are arbitrary.

An additional option **_TITLE** may be appended to all the ***SECTION** keywords. If this option is used then an additional line is read for each section in 80a format which can be used to describe the section. At present LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

SECTION**SECTION_BEAM*****SECTION_BEAM**

Purpose: Define cross sectional properties for beam, truss, discrete beam, and cable elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	ELFORM	SHRF	QR/IRID	CST	SCoor	NSM	
Type	A8	I	F	F	F	F	F	
Default	none	1	1.0	2.0	0.0	0.0	0.0	

Define the appropriate card format depending on the value of ELFORM (1-9) above.

Card 2 1 2 3 4 5 6 7 8

Integrated beam type 1,11	TS1	TS2	TT1	TT2	NSLOC	NTLOC		
Resultant 2	A	ISS	ITT	IRR	SA			
Truss 3	A	RAMPT	STRESS					
Resultant 2,3, and 12 alternative	STYPE	D1	D2	D3	D4	D5	D6	
Integrated beam type 4,5	TS1	TS2	TT1	TT2				
Discrete 6	VOL	INER	CID	CA	OFFSET	RRCON	SRCON	TRCON
Scalar 6	VOL	INER	CID	DOFN1	DOFN2			
2D shells 7,8	TS1	TS2	TT1	TT2				
Spot weld 9	TS1	TS2	TT1	TT2	PRINT			

Card 2
(continued)

	1	2	3	4	5	6	7	8
Resultant 12 1 st card	A	ISS	ITT	J	SA			
Resultant 12 2 nd card	YS	ZS	IYR	IZR	IRR	IW	IWR	
Type	A & F	F	F	F	F	F	F	F

VARIABLE**DESCRIPTION**

SECID

Section ID. SECID is referenced on the *PART card. A unique number or label not exceeding 8 characters must be specified.

ELFORM

Element formulation options:

EQ.1: Hughes-Liu with cross section integration (default),

EQ.2: Belytschko-Schwer resultant beam (resultant),

EQ.3: truss (resultant), see remark 2.

EQ.4: Belytschko-Schwer full cross-section integration,

EQ.5: Belytschko-Schwer tubular beam with cross-section integration,

EQ.6: discrete beam/cable,

EQ.7: 2D plane strain shell element (xy plane),

EQ.8: 2D axisymmetric volume weighted shell element (xy plane),

EQ.9: spotweld beam, see *MAT_SPOTWELD.

EQ.11: integrated warped beam

EQ.12: resultant warped beam

Note that the 2D and 3D element types must not be mixed, and different types of 2D elements must not be used together. For example, the plane strain element type must not be used with the axisymmetric element type. In 3D the different beam elements types, i.e., 1-6 and 9 can be freely mixed together.

SHRF

Shear factor. This factor is not needed for truss, resultant beam, discrete beam, and cable elements. The recommended value for rectangular sections is 5/6, the default is 1.0.

QR/IRID

Quadrature rule or rule number for user defined rule for integrated beams:

EQ.1.0: one integration point,

EQ.2.0: 2×2 Gauss quadrature (default beam),

EQ.3.0: 3×3 Gauss quadrature,

VARIABLE	DESCRIPTION
	<p>EQ.4.0: 3×3 Lobatto quadrature, EQ.5.0: 4×4 Gauss quadrature EQ.-n: where <i>nl</i> is the number of the user defined rule. IRID integration rule <i>n</i> is defined using *INTEGRATION_BEAM card.</p>
CST	<p>Cross section type, not needed for truss, resultant beam, discrete beam, and cable elements: EQ.0.0: rectangular, EQ.1.0: tubular (circular only), EQ.2.0: arbitrary (user defined integration rule).</p>
SCoor	<p>Location of triad for tracking the rotation of the discrete beam element, see the parameter CID below. The force and moment resultants in the output databases are referenced to this triad. The flags -3.0, -1.0, 0.0, 1.0, and 3.0 are inactive if the option to update the local system is active in the CID definition.</p> <p>EQ.-3.0: beam node 1, the angular velocity of node 1 rotates triad, EQ.-2.0: beam node 1, the angular velocity of node 1 rotates triad but the r-axis is adjusted to lie along the line between the two beam nodal points. This option is not recommended for zero length discrete beams., EQ.-1.0: beam node 1, the angular velocity of node 1 rotates triad, EQ. 0.0: centered between beam nodes 1 and 2, the average angular velocity of nodes 1 and 2 is used to rotate the triad, EQ.+1.0: beam node 2, the angular velocity of node 2 rotates triad. EQ.+2.0: beam node 2, the angular velocity of node 2 rotates triad. but the r-axis is adjusted to lie along the line between the two beam nodal points. This option is not recommended for zero length discrete beams. EQ.+3.0: beam node 2, the angular velocity of node 2 rotates triad.</p> <p><i>If the magnitude of SCoor is less than or equal to unity then zero length discrete beams are assumed with infinitesimal separation between the nodes in the deformed state. For large separations or nonzero length beams set SCoor to 2 or 3.</i></p>
NSM	<p>Nonstructural mass per unit length. This option applies to beam types 1-5 and does not apply to discrete, 2D, and spotweld beams, respectively.</p>
TS1	<p>Beam thickness (CST=0.0, 2.0) or outer diameter (CST = 1.0) in <i>s</i> direction at node <i>n</i>₁. Note that the thickness defined on the *ELEMENT_BEAM_THICKNESS card overrides the definition give here.</p>

VARIABLE	DESCRIPTION
TS2	Beam thickness (CST=0.0, 2.0) or outer diameter (CST = 1.0) in s direction at node n_2 . For truss elements only, it is the ramp up time for the stress initialization by dynamic relaxation.
TT1	Beam thickness (CST=0.0, 2.0) or inner diameter (CST = 1.0) in t direction at node n_1 . For truss elements only, it is the stress for the initialization of the stress by dynamic relaxation.
TT2	Beam thickness (CST=0.0, 2.0) or inner diameter (CST = 1.0) in t direction at node n_2 .
NSLOC	Location of reference surface normal to s axis for Hughes-Liu beam elements only. See Remark 5. EQ.1.0: side at $s = 1.0$, EQ.0.0: center, EQ.-1.0: side at $s = -1.0$.
NTLOC	Location of reference surface normal to t axis for Hughes-Liu beam elements only. See Remark 5. EQ.1.0: side at $t = 1.0$, EQ.0.0: center, EQ.-1.0: side at $t = -1.0$.
A	Cross-sectional area. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here, see Figure 29.1.
ISS	I_{ss} , moment of inertia about local s -axis. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here, see Figure 29.1.
ITT	I_{tt} , moment of inertia about local t -axis. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here, see Figure 29.1.
J	J , torsional constant. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here, see Figure 29.1. If J is zero, then J is reset to the sum of $ISS+ITT$ as an approximation for warped beam.
SA	Shear area. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here, see Figure 29.1.
YS	s coordinate of sheer center of cross-section. (The coordinate system is located at the centroid.)
ZS	t coordinate of sheer center of cross-section. (The coordinate system is located at the centroid.)

VARIABLE	DESCRIPTION
IYR	$\int_A s \cdot r^2 dA$, where $r^2 = s^2 + t^2$
IZR	$\int_A t \cdot r^2 dA$, where $r^2 = s^2 + t^2$
IRR	$\int_A r^4 dA$, where $r^2 = s^2 + t^2$
IW	Warping constant. $\int_A \omega^2 dA$, where ω is the sectorial area.
IWR	$\int_A \omega r^2 dA$
RAMPT	Optional ramp-up time for dynamic relaxation. At the end of the ramp-up time, a uniform stress, STRESS, will exist in the truss in the truss element. This option will not work for hyperelastic materials.
STRESS	Optional initial stress for dynamic relaxation. At the end of dynamic relaxation a uniform stress equal to this value should exist in the truss element.
STYPE	Section type (A format): EQ.SECTION_01: I-shape EQ.SECTION_12: Cross EQ.SECTION_02: Channel EQ.SECTION_13: H-shape EQ.SECTION_03: L-shape EQ.SECTION_14: T-shape1 EQ.SECTION_04: T-shape EQ.SECTION_15: I-shape2 EQ.SECTION_05: Tubular box EQ.SECTION_16: Channel1 EQ.SECTION_06: Z-shape EQ.SECTION_17: Channel2 EQ.SECTION_07: Trapezoidal EQ.SECTION_18: T-shape2 EQ.SECTION_08: Circular EQ.SECTION_19: Box-shape1 EQ.SECTION_09: Tubular EQ.SECTION_20: Hexagon EQ.SECTION_10: I-shape1 EQ.SECTION_21: Hat-shape EQ.SECTION_11: Solid box EQ.SECTION_22: Hat-shape1
D1-D6	Input parameters for section option using STYPE above.
VOL	Volume of discrete beam. If the mass density of the material model for the discrete beam is set to unity, the magnitude of the lumped mass can be defined here instead. This lumped mass is partitioned to the two nodes of the beam element. The translational time step size for the type 6 beam is dependent on the volume, mass density, and the translational stiffness values, so it is important to define this parameter. Defining the volume is also essential for mass scaling if the type 6 beam controls the time step size.

VARIABLE	DESCRIPTION
INER	Mass moment of inertia for the six degree of freedom discrete beam. This lumped inertia is partitioned to the two nodes of the beam element. The rotational time step size for the type 6 beam is dependent on the lumped inertia and the rotational stiffness values, so it is important to define this parameter if the rotational springs are active. Defining the rotational inertia is also essential for mass scaling if the type 6 beam rotational stiffness controls the time step size.
CID	Coordinate system ID for orientation (material types 66-69, 93, 95, 97), see *DEFINE_COORDINATE_option. If CID=0, a default coordinate system is defined in the global system or on the third node of the beam, which is used for orientation. This option is not defined for material types than act between two nodal points, such as cable elements. The coordinate system rotates with the discrete beam, see SCOOD above.
CA	Cable area, materials type ID 71, *MAT_CABLE.
OFFSET	Offset for cable. For a definition see materials type ID 71, *MAT_CABLE.
RRCON	r-rotational constraint for local coordinate system EQ.0.0: Coordinate ID rotates about r axis with nodes. EQ.1.0: Rotation is constrained about the r-axis
SRCON	s-rotational constraint for local coordinate system EQ.0.0: Coordinate ID rotates about s axis with nodes. EQ.1.0: Rotation is constrained about the s-axis
TRCON	t-rotational constraint for local coordinate system EQ.0.0: Coordinate ID rotates about t axis with nodes. EQ.1.0: Rotation is constrained about the t-axis
DOFN1	Active degree-of-freedom at node 1, a number between 1 and 6 where 1 in x-translation and 4 is x-rotation.
DOFN2	Active degree-of-freedom at node 2, a number between 1 and 6.
PRINT	Output spot force resultant from spotwelds. EQ.0.0: Data is output to SWFORC file. EQ.1.0: Output is suppressed.

Remarks:

1. For implicit calculations all of the beam element choices are implemented:
2. For the truss element, define the cross-sectional area, A, only.

3. The local coordinate system rotates as the nodal points that define the beam rotate. In some cases this may lead to unexpected results if the nodes undergo significant rotational motions. In the definition of the local coordinate system using `*DEFINE_COORDINATE_NODES`, if the option to update the system each cycle is active then this updated system is used. This latter technique seems to be more stable in some applications.
4. The integrated warped beam (type 11) is a 7 degree of freedom beam that must be used with an integration rule of the open standard cross sections, see `*INTEGRATION_BEAM`. To incorporate the additional degrees of freedom corresponding to the twist rates, the user should declare one scalar node (`*NODE_SCALAR`) for each node attached to a warped beam. This degree of freedom is associated to the beam element using the warpage option on the `*ELEMENT_BEAM` card.
5. Beam offsets are sometimes necessary for correctly modeling beams that act compositely with other elements such as shells or other beams. A beam offset extends from the beam's N1-to-N2 axis to the reference axis of the beam. The beam reference axis lies at the origin of the local s and t axes, i.e., halfway between the outermost surfaces of the beam cross-section. Note that for cross-sections that are not doubly symmetric, e.g, a T-section, the reference axis does not pass through the centroid of the cross-section. The offset in the positive s-direction is $s\text{-offset} = -0.5 * NSLOC *$ (beam cross-section dimension in s-direction). Similarly, the offset in the positive t-direction is $t\text{-offset} = -0.5 * NTLOC *$ (beam cross-section dimension in t-direction). See also `*ELEMENT_BEAM_OFFSET` for an alternate approach to defining beam offsets.

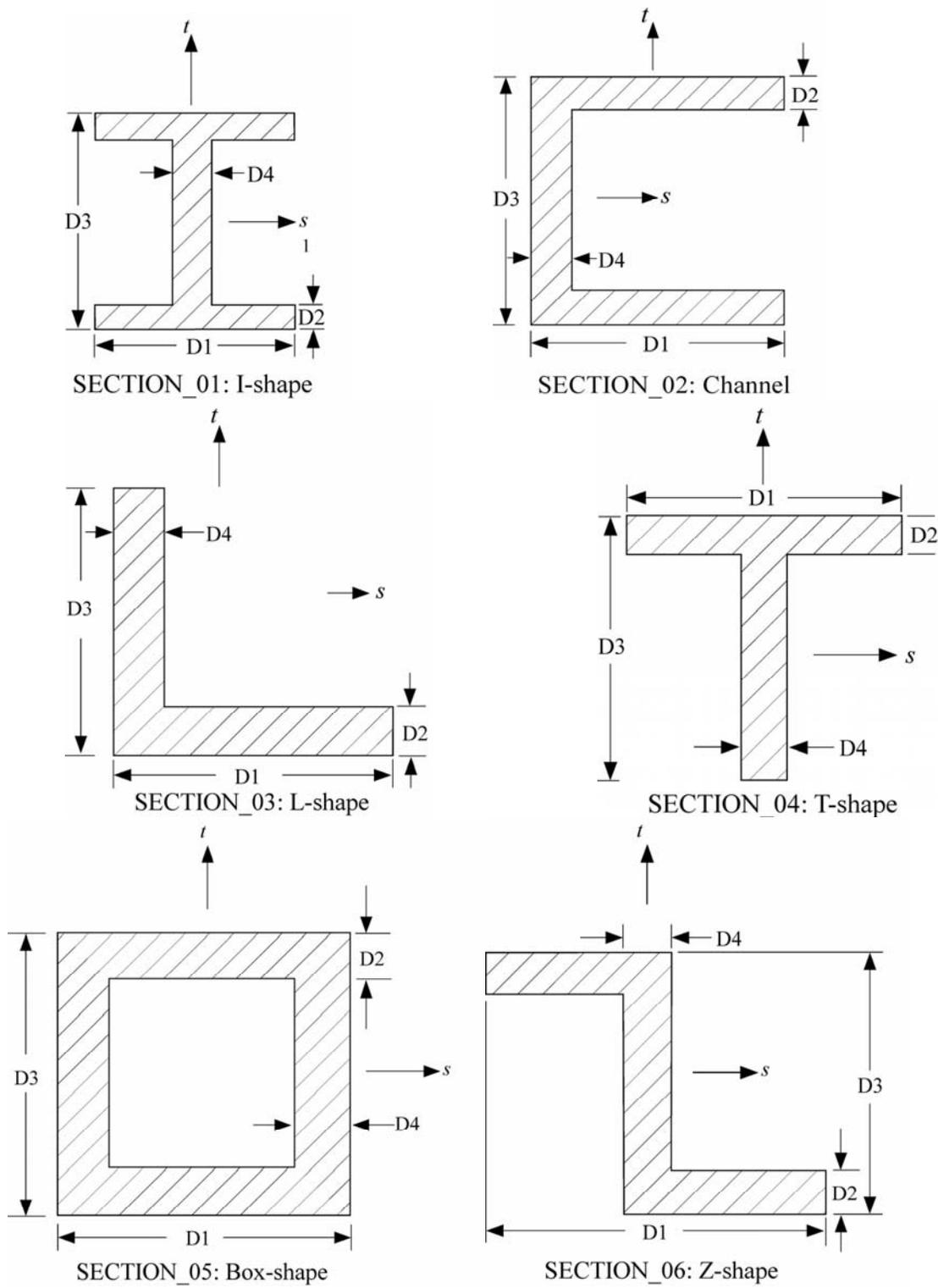


Figure 29.1. Properties of beam cross section for several common cross sections.

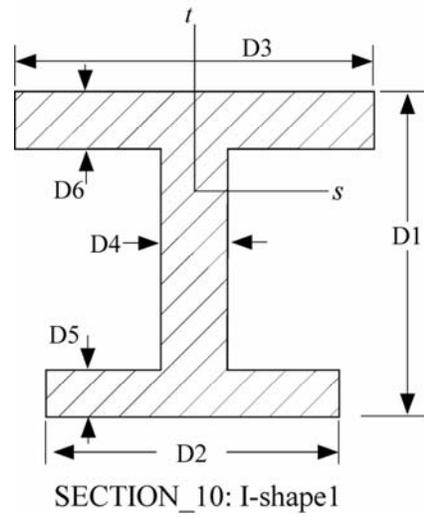
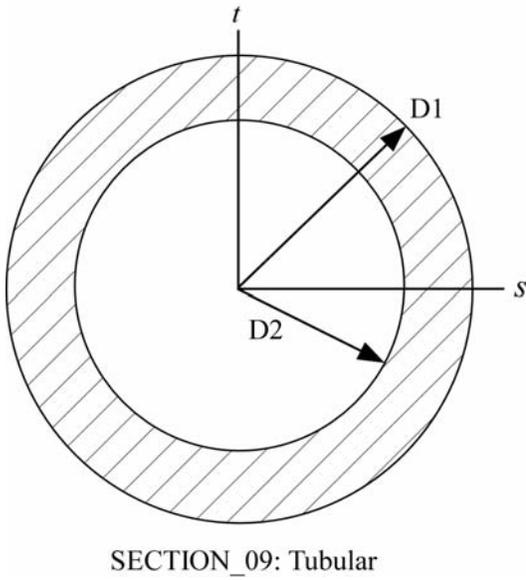
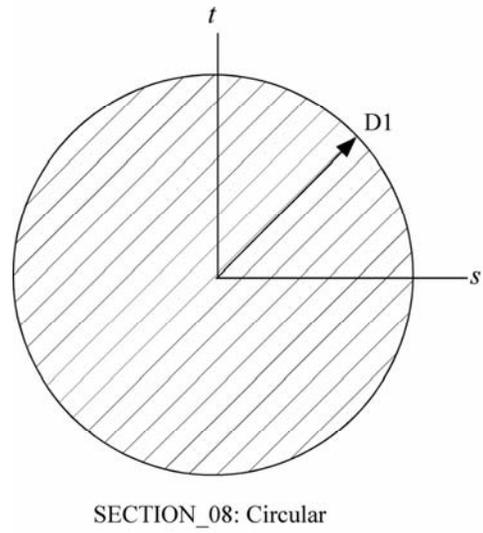
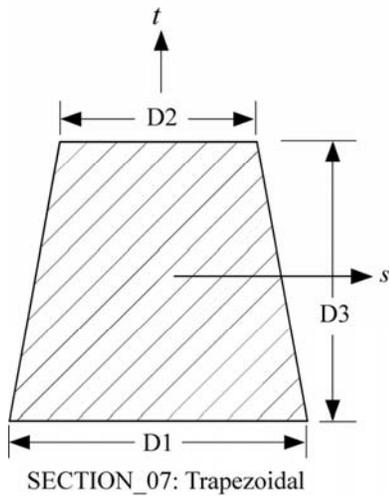


Figure 29.1. Properties of beam cross section for several common cross sections (cont'd).

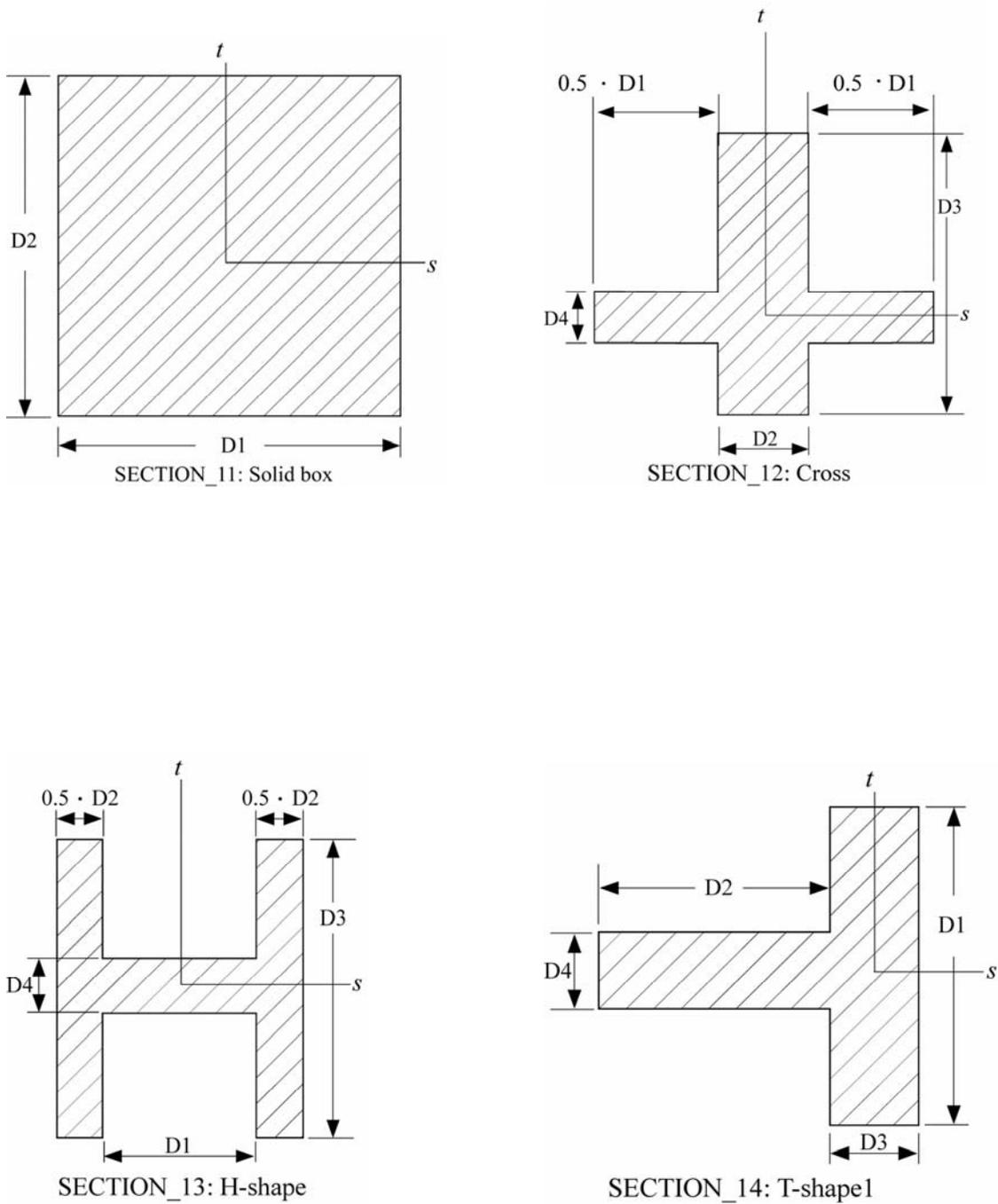


Figure 29.1. Properties of beam cross section for several common cross sections (cont'd).

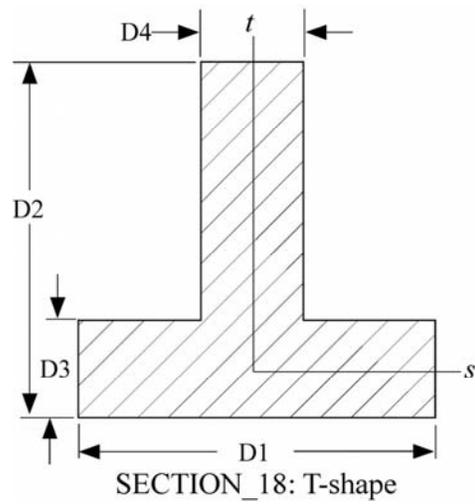
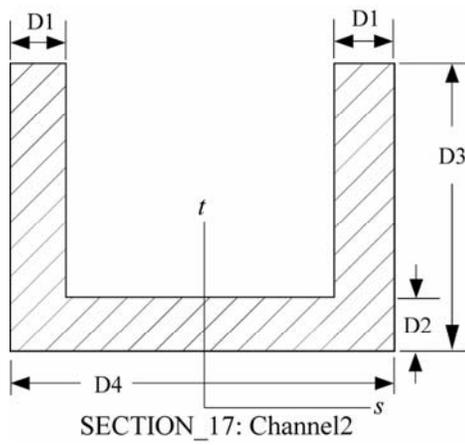
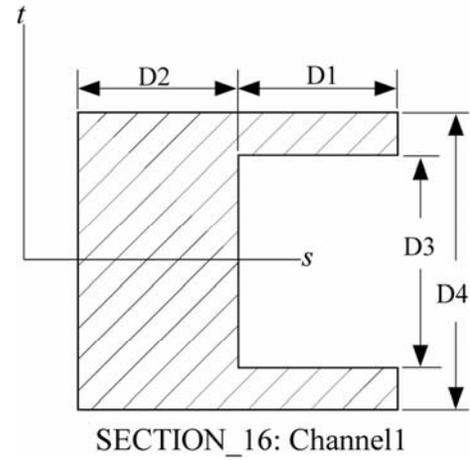
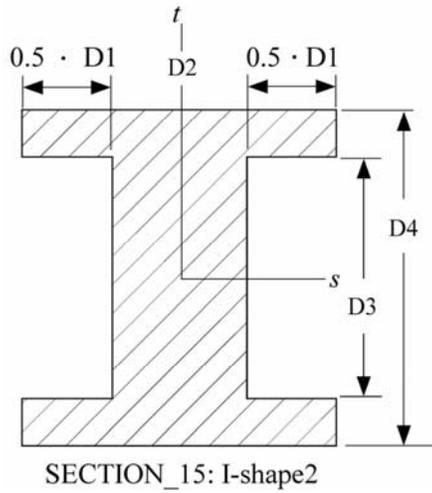


Figure 29.1. Properties of beam cross section for several common cross sections (cont'd).

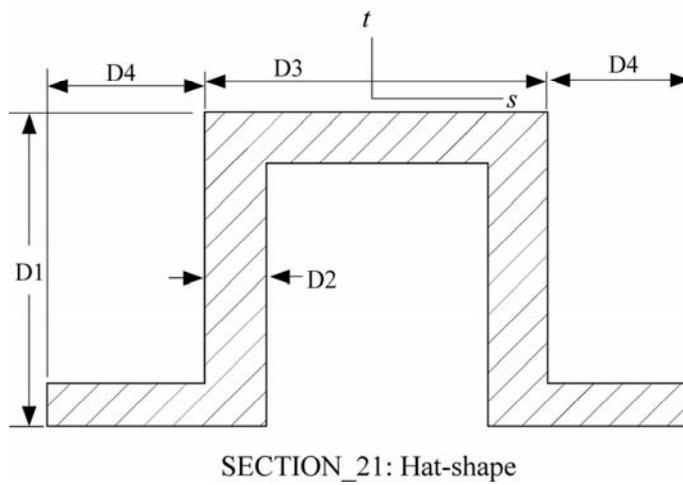
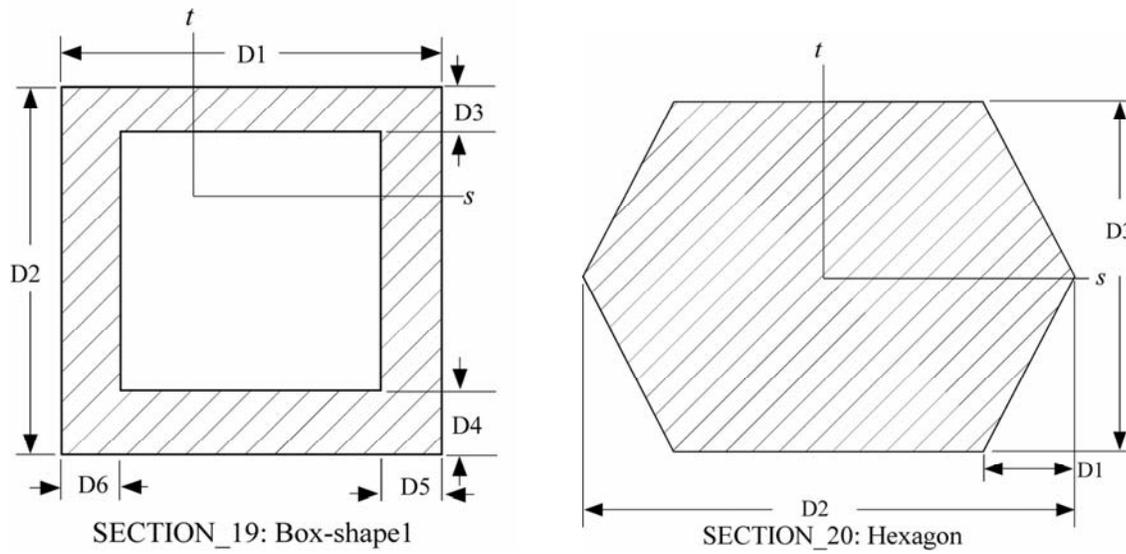


Figure 29.1. Properties of beam cross section for several common cross sections (cont'd).

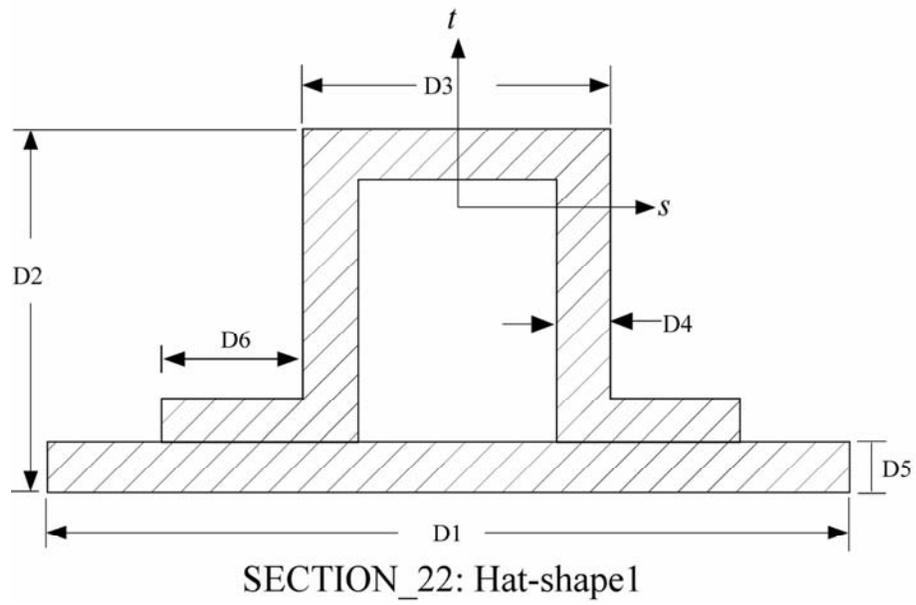


Figure 29.1. Properties of beam cross section for several common cross sections (cont'd).

*SECTION

*SECTION_DISCRETE

*SECTION_DISCRETE

Purpose: Defined spring and damper elements for translation and rotation. These definitions must correspond with the material type selection for the elements, i.e., *MAT_SPRING... and *MAT_DAMPER...

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	DRO	KD	V0	CL	FD		
Type	A8	I	F	F	F	F		

Card 2

Variable	CDL	TDL						
Type	F	F						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label not exceeding 8 characters must be specified.
DRO	Displacement/Rotation Option: EQ.0: the material describes a translational spring/damper, EQ.1: the material describes a torsional spring/damper.
KD	Dynamic magnification factor. See Remarks 1 and 2 below.
V0	Test velocity
CL	Clearance. See Remark 3 below.
FD	Failure deflection (twist for DRO=1). Negative for compression, positive for tension.
CDL	Deflection (twist for DRO=1) limit in compression. See Remark 4 below.
TDL	Deflection (twist for DRO=1) limit in tension. See Remark 4 below.

Remarks:

1. The constants from KD to TDL are optional and do not need to be defined.
2. If k_d is nonzero, the forces computed from the spring elements are assumed to be the static values and are scaled by an amplification factor to obtain the dynamic value:

$$F_{dynamic} = \left(1. + k_d \frac{V}{V_0} \right) F_{static}$$

where

V = absolute value of the relative velocity between the nodes.

V_0 = dynamic test velocity.

For example, if it is known that a component shows a dynamic crush force at 15m/s equal to 2.5 times the static crush force, use $k_d=1.5$ and $V_0=15$.

3. Here, “clearance” defines a compressive displacement which the spring sustains before beginning the force-displacement relation given by the load curve defined in the material selection. If a non-zero clearance is defined, the spring is compressive only.
4. The deflection limit in compression and tension is restricted in its application to no more than one spring per node subject to this limit, and to deformable bodies only. For example in the former case, if three springs are in series, either the center spring or the two end springs may be subject to a limit, but not all three. When the limiting deflection is reached, momentum conservation calculations are performed and a common acceleration is computed in the appropriate direction. An error termination will occur if a rigid body node is used in a spring definition where deflection is limited.

Constrained boundary conditions on the *NODE cards and the BOUNDARY_SPC cards must not be used for nodes of springs with deflection limits.

5. Discrete elements can be included in implicit applications.

***SECTION_POINT_SOURCE**

Purpose: This command provides the inlet boundary condition for single gas in flow (inflation potential) via a set of point source(s). It also provides the inflator orifice geometry information. It requires 3 curves defining the inlet condition for the inflator gas coming into the tank or an airbag as input ($\bar{T}_{gas_corrected}(t)$, $v_r(t)$, and $vel(t)$). Please see also the *ALE_TANK_TEST card for additional information.

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	LCIDT	LCIDVOLR	LCIDVEL	NIDLC001	NIDLC002	NIDLC003	
Type	A8	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Card 2

Variable	NODEID	VECID	ORIFAREA					
Type	I	I	F					
Default	0	0	0.0					

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SECID	Section ID. A unique number or label not exceeding 8 characters must be specified.
LCIDT	Temperature load curve ID
LCIDVOLR	Relative volume load curve ID
LCIDVEL	Inlet flow velocity load curve ID
NIDLC001	The 1 st node ID defining a local coordinate (See Remark 2).
NIDLC002	The 2 nd node ID defining a local coordinate (See Remark 2).
NIDLC003	The 3 rd node ID defining a local coordinate (See Remark 2).

VARIABLE	DESCRIPTION
NODEID	The node ID(s) defining the point source(s).
VECID	The vector ID defining the direction of flow at each point source.
ORIFAREA	The orifice area at each point source.

Remarks:

1. In an airbag inflator tank test, the tank pressure data is measured. This pressure is used to derive $\dot{m}(t)$ and the estimated $\bar{T}_{gas}(t)$, usually via a lumped-parameter method, a system of conservation equations and EOS. Subsequently $\dot{m}(t)$ and $\bar{T}_{gas}(t)$ (stagnation temperature) are used as input to obtain $\bar{T}_{gas_corrected}(t)$ (static temperature), $v_r(t)$, and $vel(t)$. These 3 curves are then used to describe inflator gas inlet condition (see *ALE_TANK_TEST for more information).
2. In a car crash model, the inflator housing may get displaced during the impact. The 3 node IDs defines the local reference coordinate system to which the point sources are attached. These 3 reference nodes may be located on a rigid body which can translate and rotate as the inflator moves during the impact. This allows for the point sources to move in time. These reference nodes may be used as the point sources themselves.
3. If the *ALE_TANK_TEST card is present, please see the Remarks under that card.

Example:

Consider a tank test model which consists of the inflator gas (PID 1) and the air inside the tank (PID 2). The 3 load curves define the thermodynamic and kinetic condition of the incoming gas. The nodes define the center of the orifice, and the vector the direction of flow at each orifice.

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*PART
inflator gas
$  PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   1         1         1         0         0         0         0         0
*SECTION_POINT_SOURCE
$  SECID    LCIDT    LCIDVOLR    LCIDVEL    NIDLCOOR1    NIDLCOOR2    NIDLCOOR3
   1         3         4         5         0         0         0
$  NODEID   VECTID    AREA
   24485    3        15.066
   ...
   24557    3        15.066
*PART
air inside the tank
$  PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   2         2         2         0         0         0         0         0
*SECTION_SOLID
$  SECID    ELFORM    AET
   2         11         0
*ALE_MULTI-MATERIAL GROUP
$  SID      SIDTYPE
   1         1
   2         1
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

***SECTION_POINT_SOURCE_MIXTURE**

***SECTION**

***SECTION_POINT_SOURCE_MIXTURE**

Purpose: This command provides (a) an element formulation for a solid ALE part of the type similar to ELFORM=11 of *SECTION_SOLID, and (b) the inlet gas injection boundary condition for multiple-gas mixture in-flow via a set of point source(s). It also provides the inflator orifice geometry information. This must be used in combination with the *MAT_GAS_MIXTURE and/or *INITIAL_GAS_MIXTURE card (see Remark 1).

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	LCIDT	Not Used	LCIDVEL	NIDLC001	NIDLC002	NIDLC003	IDIR
Type	A8	I		I	I	I	I	I
Default	none	none		none	none	none	none	0

Card 2

Variable	LCMDOT1	LCMDOT2	LCMDOT3	LCMDOT4	LCMDOT5	LCMDOT6	LCMDOT7	LCMDOT8
Type	I	I	I	I	I	I	I	I
Default	none							

Card 3

Variable	NODEID	VECID	ORIFAREA					
Type	I	I	F					
Default	none	none	0.0					

VARIABLE

DESCRIPTION

SECID

Section ID. A unique number or label not exceeding 8 characters must be specified.

VARIABLE	DESCRIPTION
LCIDT	Inflator gas mixture average stagnation temperature load curve ID (all gases of the mixture are assumed to have the same average temperature).
LCIDVEL	User-defined inflator gas mixture average velocity load curve ID. If LCIDVEL=0 or blank, LSDYNA will estimate the inlet gas velocity.
NIDLC001	The 1 st node ID defining a local coordinate (see Remark 2).
NIDLC002	The 2 nd node ID defining a local coordinate (see Remark 2).
NIDLC003	The 3 rd node ID defining a local coordinate (see Remark 2).
IDIR	A flag for constraining the nodal velocity of the nodes of the ALE element containing a point source. If IDIR=0 (default), then the ALE nodes behind the point source (relative position of nodes based on the vector direction of flow of point source) will have zero velocity. If IDIR=1, then all ALE nodes will have velocity distributed based on energy conservation. The latter option seems to be more robust in airbag modeling (see Remark 6).
LCMDOT1	The mass flow rate load curve ID of the 1 st gas in the mixture.
LCMDOTn	The mass flow rate load curve ID of the n th gas in the mixture.
LCMDOT8	The mass flow rate load curve ID of the 8 th gas in the mixture.
NODEID	The node ID(s) defining the point sources (see Remark 6).
VECID	The vector ID defining the direction of flow at each point source.
ORIFAREA	The orifice area at each point source.

Remarks:

1. This command is used to define a part that acts as the ideal gas mixture injection source. The associated ALE material (gas mixture) may not be present at time zero, but can be introduced (injected) into an existing ALE domain. For airbag application, the input from control volume analysis, inlet mass flow rate, $\dot{m}(t)$, and, inlet stagnation gas temperature, $\bar{T}_{gas}(t)$ may be used as direct input for ALE analysis. If available, the user may input a load curve for the gas mixture average inlet velocity. If not, LS-DYNA will estimate the inlet gas velocity.
2. The gas mixture is assumed to have a uniform temperature ($\bar{T} \approx T_i$) and inlet velocity. However, the species in the mixture may each have a different inlet mass flow rate.

3. A brief review of the concept used is presented. The total energy (e_T) is the sum of internal (e_i) and kinetic $\left(\frac{V^2}{2}\right)$ energies, (per unit mass).

$$e_T = e_i + \frac{V^2}{2}$$

$$C_V T_{stag} = C_V T + \frac{V^2}{2}$$

$$T_{stag} = T + \frac{V^2}{2C_V}$$

The distinction between stagnation and static temperatures is shown above. C_V is the constant-volume heat capacity. The gas mixture average internal energy per unit mass in terms of mixture species contribution is

$$e_i = \bar{C}_V \bar{T} = \sum_i \left(\frac{\rho_i}{\rho_{mixture}} \right) C_{V_i} T_i = \left[\sum_i \left(\frac{\rho_i}{\rho_{mixture}} \right) C_{V_i} \right] \bar{T}$$

$$\bar{C}_V = \left[\sum_i \left(\frac{\rho_i}{\rho_{mixture}} \right) C_{V_i} \right]$$

Since we approximate $\bar{T} \approx T_i$, then gas mixture average static temperature is related to the mixture average internal energy per unit mass as following

$$\bar{T} = \frac{e_i}{\left[\sum_i \left(\frac{\rho_i}{\rho_{mixture}} \right) C_{V_i} \right]}$$

Note that the “i” subscript under “e” denotes “internal” energy, while the other “i” subscripts denote the “ith” species in the gas mixture. The total mixture pressure is the sum of the partial pressures of the individual species.

$$\bar{P} = \sum_i p_i$$

The ideal gas EOS applies to each individual species (by default)

$$P_i = \rho_i (C_{P_i} - C_{V_i}) T_i$$

4. Generally, it is not possible to conserve both momentum and kinetic (KE) at the same time. Typically, internal energy (IE) is conserved and KE may not be. This may result in some KE loss (hence, total energy loss). For many analyses this is tolerable, but for

airbag application, this may lead to the reduction of the inflating potential of the inflator gas.

In *MAT_GAS_MIXTURE computation, any kinetic energy not accounted-for during advection is stored in the internal energy. Therefore, there is no kinetic energy loss, and the total energy of the element is conserved over the advection step. This is a simple, ad hoc approach that is not rigorously derived for the whole system based on first principles. Therefore it is not guaranteed to apply universally to all scenarios. It is the user's responsibility to validate the model with data.

5. Since ideal gas is assumed, there is no need to define the EOS for the gases in the mixture.
6. In general, it is best to locate a point source near the center of an ALE element. Associated with each point source is an area and a vector indicating flow direction. Each point source should occupy 1 ALE element by itself, and there should be at least 2 empty ALE elements between any 2 point sources. A point source should be located at least 3 elements away from the free surface of an ALE mesh for stability.

Example 1:

Consider a tank test model without coupling which consists of:

- a background mesh with air (PID 1 = gas 1) initially inside that mesh (tank space),
and
- the inflator gas mixture (PID 2 consisting of inflator gases 2, 3, and 4).

The mixture is represented by one AMMGID and the air by another AMMGID.

The tank internal space is simply modeled with an Eulerian mesh of the same volume. The Tank itself is not modeled thus no coupling is required. The inflator gases fill up this space mixing with the air initially inside the tank.

The background air (gas 1) is included in the gas mixture definition in this case because that air will participate in the mixing process. Only include in the mixture those gases that actually undergo mixing (gases 1, 2, 3 and 4). Note that for an airbag model, the "outside" air should not be included in the mixture (it should be defined independently) since it does not participate in the mixing inside the airbag. This is shown in the next example.

The nodes define the center of the orifices, and the vectors define the directions of flow at these orifices.

***SECTION_POINT_SOURCE_MIXTURE**

***SECTION**

```

$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*PART
Tank background mesh, initially filled with air, allows gas mixture to flow in.
$   PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   1         1         1         0         0         0         0         0
*SECTION_SOLID
$   SECID      ELFORM      AET
   1         11         0
$ The next card defines the properties of the gas species in the mixture.
*MAT_GAS_MIXTURE
$   MID
   1
$   Cv1      Cv2      Cv3      Cv4      Cv5      Cv6      Cv7      Cv8
  654.47    482.00    2038.30    774.64     0.0     0.0     0.0     0.0
$   Cp1      Cp2      Cp3      Cp4      Cp5      Cp6      Cp7      Cp8
  941.32    666.67    2500.00    1071.40     0.0     0.0     0.0     0.0

$ The next card specifies that gas 1 (background air) occupies PID 1 at time 0.
*INITIAL_GAS_MIXTURE
$   SID      STYPE      AMMGID      TEMP0
   1         1         1         293.00
$   RHO1      RHO2      RHO3      RHO4      RHO5      RHO6      RHO7      RHO8
  1.20E-9    0.0      0.0      0.0      0.0      0.0      0.0      0.0
*PART
The gas mixture (inlet) definition (no initial mesh required for this PID)
$   PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   2         2         1         0         0         0         0         0
*SECTION_POINT_SOURCE_MIXTURE
$   SECID      LCIDT      NOTUSED      LCIDVEL      NIDLCOOR1      NIDLCOOR2      NIDLCOOR3      IDIR
   2         1         0         5         0         0         0         0
$   LCMDOT1      LCMDOT2      LCMDOT3      LCMDOT4      LCMDOT5      LCMDOT6      LCMDOT7      LCMDOT8
   0         2         3         4         0         0         0         0
$   NODEID      VECTID      AREA
  24485         1         25.0
...
  24557         1         25.0
*ALE_MULTI-MATERIAL_GROUP
$   SID      SIDTYPE
   1         1
   2         1
*DEFINE_VECTOR
$   VECTID      XTAIL      YTAIL      ZTAIL      XHEAD      YHEAD      ZHEAD
   1         0.0      0.0      0.0      0.0      1.0      0.0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8

```

*SECTION

*SECTION_POINT_SOURCE_MIXTURE

Example 2:

Consider an airbag inflation model which consists of:

- a background Eulerian mesh for air initially outside the airbag (PID 1)
- the inflator gas mixture (PID 2 consisting of inflator gases 1, 2, and 3).

The mixture is represented by one AMMGID and the air by another AMMGID.

The background air (PID 1) is NOT included in the gas mixture definition in this case because that air will NOT participate in the mixing process. Only include in the mixture those gases that actually undergo mixing (gases 1, 2, and 3). Gases 1, 2, and 3 in this example correspond to gases 2, 3, and 4 in example 1. Compare the air properties in PID 1 here to that of example 1. Note that the *INITIAL_GAS_MIXTURE card is not required to initialize the background mesh in this case.

```
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*PART
Tank background mesh, initially filled with air, allows gas mixture to flow in.
$   PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   1         1         1         0         0         0         0         0
*SECTION SOLID
$   SECID     ELFORM      AET
   1         11         0
*MAT_NULL
$   MID      RHO      PCUT      MU      TEROD      CEROD      YM      PR
   1      1.20E-9  -1.0E-6    0.0    0.0      0.0      0.0    0.0
*EOS_IDEAL_GAS
$   EOSID     CV0      CP0      COEF1     COEF2      T0      RELVOL0
   1      654.47  941.32    0.0      0.0      293.00    1.0
$ The next card defines the properties of the gas species in the mixture.
*PART
The gas mixture (inlet) definition (no initial mesh required for this PID)
$   PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
   2         2         2         0         0         0         0         0
*SECTION_POINT_SOURCE_MIXTURE
$   SECID     LCIDT     NOTUSED     LCIDVEL  NIDLCOOR1  NIDLCOOR2  NIDLCOOR3     IDIR
   2         1         0         5         0         0         0         0
$   LCMDOT1  LCMDOT2  LCMDOT3  LCMDOT4  LCMDOT5  LCMDOT6  LCMDOT7  LCMDOT8
   2         3         4         0         0         0         0         0
$   NODEID   VECTID     AREA
   24485     1         25.0
   ...
   24557     1         25.0
*MAT_GAS_MIXTURE
$   MID
   2
$   Cv1     Cv2     Cv3     Cv4     Cv5     Cv6     Cv7     Cv8
   482.00  2038.30  774.64  0.0     0.0     0.0     0.0     0.0
$   Cp1     Cp2     Cp3     Cp4     Cp5     Cp6     Cp7     Cp8
   666.67  2500.00  1071.40  0.0     0.0     0.0     0.0     0.0
$ The next card specifies that gas 1 (background air) occupies PID 1 at time 0.
*ALE_MULTI-MATERIAL_GROUP
$   SID     SIDTYPE
   1         1
   2         1
*DEFINE_VECTOR
$   VECTID   XTAIL   YTAIL   ZTAIL   XHEAD   YHEAD   ZHEAD
   1         0.0    0.0    0.0    0.0    1.0    0.0
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
```

*SECTION_SEATBELT

Purpose: Define section properties for the seat belt elements. This card is required for the *PART Section. Currently, only the ID is required.

Card 1 1 2 3 4 5 6 7 8

Variable	SECID							
Type	A8							

VARIABLE

DESCRIPTION

SECID

Section ID. A unique number or label not exceeding 8 characters must be specified.

Remarks:

1. Seatbelt elements are not implemented for implicit calculations.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *SECTION_SEATBELT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define a seat belt section that is referenced by part 10. Nothing
$ more than the sid is required.
$
*SECTION_SEATBELT
$
$.>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   sid
$   111
$
$
$
*PART
Seatbelt material
$.>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   pid      sid      mid      eosid      hgid      adpopt
$   10       111     220
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

*SECTION

*SECTION_SHELL

*SECTION_SHELL_{OPTION}

Available options include:

<BLANK>

ALE

EFG

such that the keyword cards appear:

*SECTION_SHELL

*SECTION_SHELL_ALE

*SECTION_SHELL_EFG

Purpose: Define section properties for shell elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	ELFORM	SHRF	NIP	PROPT	QR/IRID	ICOMP	SETYP
Type	A8	I	F	F	F	F	I	I
Default	none	0	1.0	2	0.0	0.0	0	1
Remarks		1,7,8						

Card 2

Variable	T1	T2	T3	T4	NLOC	MAREA	IDOF	EDGSET
Type	F	F	F	F	F	F	F	I
Default	0.0	T1	T1	T1	0.0	0.0	0.0	
Remarks							7	8

Optional Section Cards if ICOMP=1. Define NIP angles putting 8 on each card.

Card 1 2 3 4 5 6 7 8

Variable	B1	B2	B3	B4	B5	B6	B7	B8
Type	F	F	F	F	F	F	F	F

Optional Section Card for ALE option.

Also see *CONTROL_ALE and *ALE_SMOOTHING.

Card 1 2 3 4 5 6 7 8

Variable	AFAC	BFAC	CFAC	DFAC	EFAC	START	END	AAFAC
Type	F	F	F	F	F	F	F	F

Optional Section Card for EFG option.

Also see *CONTROL_EFG.

Card 1 2 3 4 5 6 7 8

Variable	DX	DY	ISPLINE	IDILA				
Type	F	F	I	I				
Default	1.01	1.01	0	0				

Define the next 3 cards if and only if ELFORM=101,102,103,104 or 105.

Also see Appendix C

Card 1 2 3 4 5 6 7 8

Variable	NIPP	NXDOF	IUNF	IHGF	ITAJ	LMC	NHSV	ILOC
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Define NIPP cards according to the following format

Also see Appendix C

Card	1	2	3	4	5	6	7	8
Variable	XI	ETA	WGT					
Type	F	F	F					
Default	None	None	None					

Define LMC property parameters using 8 parameters per card.

Also see Appendix C

Card	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	0	0	0	0	0	0	0	0

VARIABLE

DESCRIPTION

SECID

Section ID. SECID is referenced on the *PART card. A unique number or label not exceeding 8 characters must be specified.

ELFORM

Element formulation options, see Remarks 1 and 2 below:

- EQ.1: Hughes-Liu,
- EQ.2: Belytschko-Tsay,
- EQ.3: BCIZ triangular shell,
- EQ.4: C⁰ triangular shell,
- EQ.5: Belytschko-Tsay membrane,
- EQ.6: S/R Hughes-Liu,
- EQ.7: S/R co-rotational Hughes-Liu,
- EQ.8: Belytschko-Leviathan shell,
- EQ.9: Fully integrated Belytschko-Tsay membrane,
- EQ.10: Belytschko-Wong-Chiang,
- EQ.11: Fast (co-rotational) Hughes-Liu,
- EQ.12: Plane stress (x-y plane),
- EQ.13: Plane strain (x-y plane),
- EQ.14: Axisymmetric solid (y-axis of symmetry) - area weighted,

VARIABLE	DESCRIPTION
	EQ.15: Axisymmetric solid (y-axis of symmetry) - volume weighted,
	EQ.16: Fully integrated shell element (very fast),
	EQ.17: Fully integrated DKT, triangular shell element,
	EQ.18: Fully integrated linear DK quadrilateral/triangular shell
	EQ.20: Fully integrated linear assumed strain C ⁰ shell (See Remarks).
	EQ.21: Fully integrated linear assumed strain C ⁰ shell (5 DOF).
	EQ.22: Linear shear panel element (3 DOF per node, see remarks)
	EQ.23: 8-node quadratic quadrilateral shell (under development)
	EQ.24: 6-node quadratic triangular shell (under development)
	EQ.25: Belytschko-Tsay shell with thickness stretch.
	EQ.26: Fully integrated shell with thickness stretch.
	EQ.27: C ⁰ triangular shell with thickness stretch.
	EQ.43: Mesh-free plane strain formulation (x-y plane).
	EQ.44: Mesh-free axisymmetric solid formulation (y-axis of symmetry).
	EQ.99: Simplified linear element for time-domain vibration studies. See remark 5 below.
	GT.100.AND.LT.106: User defined shell

The type 18 element is only for linear static and normal modes. It can also be used for linear springback in sheet metal stamping.

Note that the 2D and 3D element types must not be mixed, and different types of 2D elements must not be used together. For example, 2D axisymmetric calculations can use either element types 14 or 15 but these element types must not be mixed together. Likewise, the plane strain element type must not be used with either the plane stress element or the axisymmetric element types. In 3D, the different shell elements types, i.e., 1-11 and 16, can be freely mixed together.

SHRF Shear correction factor which scales the transverse shear stress. The shell formulations in LS-DYNA, with the exception of the BCIZ and DK elements, are based on a first order shear deformation theory that yields constant transverse shear strains which violates the condition of zero traction on the top and bottom surfaces of the shell. The shear correction factor is attempt to compensate for this error. A suggested value is 5/6 for isotropic materials. This value is incorrect for sandwich or laminated shells; consequently, laminated/sandwich shell theory is now an option in some of the constitutive models, e.g., material types 22, 54, and 55.

NIP Number of through thickness integration points. Either Gauss (default) or Lobatto integration can be used. The flag for Lobatto integration can be set on the control card, *CONTROL_SHELL. The location of the Gauss and Lobatto integration points are tabulated below.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.0.0: set to 2 integration points for shell elements. EQ.1.0: 1 point (no bending) EQ.2.0: 2 point EQ.3.0: 3 point EQ.4.0: 4 point EQ.5.0: 5 point EQ.6.0: 6 point EQ.7.0: 7 point EQ.8.0: 8 point EQ.9.0: 9 point EQ.10.: 10 point GT.10.: trapezoidal or user defined rule
	Through thickness integration for the two-dimensional elements (options 12-15 above) is not meaningful; consequently, the default is equal to 1 integration point. Fully integrated two-dimensional elements are available for options 13 and 15 by setting NIP equal to a value of 4 corresponding to a 2 by 2 Gaussian quadrature. If NIP is 0 or 1 and the *MAT_SIMPLIFIED_JOHNSON_COOK model is used, then a resultant plasticity formulation is activated. NIP is always set to 1 if a constitutive model based on resultants is used.
PROPT	Printout option (**NOT ACTIVE**): EQ.1.0: average resultants and fiber lengths, EQ.2.0: resultants at plan points and fiber lengths, EQ.3.0: resultants, stresses at all points, fiber lengths.
QR/IRID	Quadrature rule or Integration rule ID, see *INTEGRATION_SHELL: LT.0.0: absolute value is specified rule number, EQ.0.0: Gauss/Lobatto (up to 10 points are permitted), EQ.1.0: trapezoidal, <i>not recommend for accuracy reasons</i> .
ICOMP	Flag for orthotropic/anisotropic layered composite material model. This option applies to material types 22, 23, 33, 34, 36, 40, 41-50, 54-56, 58, 59, 103, 116, and 194. EQ.1: a material angle in degrees is defined for each through thickness integration point. Thus, each layer has one integration point.
SETYP	2D solid element type: Defined for ELFORM 13, 14, and 15. EQ.1: Lagrangian EQ.2: Eulerian (single material with voids) EQ.3: ALE
T1	Shell thickness at node n ₁ , unless the thickness is defined on the *ELEMENT_SHELL_OPTION card.

VARIABLE	DESCRIPTION
T2	Shell thickness at node n_2 , see comment for T1 above.
T3	Shell thickness at node n_3 , see comment for T1 above.
T4	Shell thickness at node n_4 , see comment for T1 above.
NLOC	<p>Location of reference surface for three dimensional shell elements. If nonzero, the offset distance from the plane of the nodal points to the reference surface of the shell in the direction of the shell normal vector is a value $offset = -0.50 \times NLOC \times (average\ shell\ thickness)$. This offset is not considered in the contact subroutines. Alternatively, the offset can be specified by using the OFFSET option in the *ELEMENT_SHELL input section.</p> <p>EQ. 1.0: top surface, EQ. 0.0: mid-surface (default), EQ.-1.0: bottom surface.</p>
MAREA	<p>Non-structural mass per unit area. This is additional mass which comes from materials such as carpeting. This mass is not directly included in the time step calculation. Another and often more convenient alternative for defining distributed mass is by the option: *ELEMENT_MASS_PART, which allows additional non-structural mass to be distributed by an area weighted distribution to all nodes of a given part ID).</p>
IDOF	<p>Applies to shell element types 25 and 26.</p> <p>EQ.1: The thickness field is continuous across the element edges for metalforming applications. This is the default.</p> <p>EQ.2: The thickness field is discontinuous across the element edges. This is necessary for crashworthiness simulations due to shell intersections, sharp included angles, and non-smooth deformations.</p>
EDGSET	<p>Edge node set required for shell type seatbelts. Input an ordered set of nodes along one of the transverse edges of a seatbelt. If there is no retractor associated with a belt, the node set can be on either edge. If the retractor exists, the edge must be on the retractor side and input in the same sequence of retractor node set. Therefore, another restriction on the seatbelt usage is that each belt has its own section definition and, therefore, a unique part ID. See Figure 13.2 in the section *ELEMENT_SEATBELT for additional clarification.</p>
B1	β_1 , material angle at first integration point
B2	β_2 , material angle at second integration point
B3	β_3 , material angle at third integration point

VARIABLE	DESCRIPTION
·	·
·	·
B8	β_8 , material angle at eighth integration point
·	·
·	·
Bnip	β_{nip} , material angle at ninth integration point
AFAC	Smoothing weight factor - Simple average: EQ.-1: turn smoothing off.
BFAC	Smoothing weight factor - Volume weighting
CFAC	Smoothing weight factor - Isoparametric
DFAC	Smoothing weight factor - Equipotential
EFAC	Smoothing weight factor - Equilibrium
START	Start time for smoothing
END	End time for smoothing
AAFAC	ALE advection factor
DX,DY	Normalized dilation parameters of the kernel function in X and Y directions. The normalized dilation parameters of the kernel function are introduced to provide the smoothness and compact support properties on the construction of the mesh-free shape functions. Values between 1.0 and 2.0 are recommended. Values smaller than 1.0 are not allowed. Larger values will increase the computation time and will sometimes result in a divergence problem. See Remark 6.
ISPLINE	Replace the choice for the EFG kernel functions definition in *CONTROL_EFG. This allows users to define different ISPLINE in different sections.
IDILA	Replace the choice for the normalized dilation parameter definition in *CONTROL_EFG. This allows users to define different IDILA in different sections.
NIPP	Number of in-plane integration points for user-defined shell (0 if resultant/discrete element)
NXDOF	Number of extra degrees of freedom per node for user-defined shell

VARIABLE	DESCRIPTION
IUNF	Flag for using nodal fiber vectors in user-defined shell: EQ.0: Nodal fiber vectors are not used. EQ.1: Nodal fiber vectors are used.
IHFG	Flag for using hourglass stabilization (NIPP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used EQ.3: Same as 2, but the resultant material tangent moduli are passed
ITAJ	Flag for setting up finite element matrices (NIPP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
ILOC	Coordinate system option: EQ.0: Pass all variables in LS-DYNA local coordinate system EQ.1: Pass all variables in global coordinate system
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

GAUSS INTEGRATION RULE					
NUMBER OF GAUSS POINT	1 POINT	2 POINT	3 POINT	4 POINT	5 POINT
#1	.0	-.5773503	.0	-.8611363	.0
#2		+.5773503	-.7745967	-.3399810	-.9061798
#3			+.7745967	+.3399810	-.5384693
#4				+.8622363	+.5384693
#5					+.9061798
NUMBER OF GAUSS POINT	6 POINT	7 POINT	8 POINT	9 POINT	10 POINT
#1	-.9324695	-.9491080	-.9702896	-.9681602	-.9739066
#2	-.6612094	-.7415312	-.7966665	-.8360311	-.8650634
#3	-.2386192	-.4058452	-.5255324	-.6133714	-.6794096
#4	+.2386192	.0	-.1834346	-.3242534	-.4333954
#5	+.6612094	+.4058452	+.1834346	0.0	-.1488743
#6	+.9324695	+.7415312	+.5255324	+.3242534	+.1488743
#7		+.9491080	+.7966665	+.6133714	+.4333954
#8			+.9702896	+.8360311	+.6794096
#9				+.9681602	+.8650634
#10					+.9739066

Location of through thickness Gauss integration points. The coordinate is referenced to the shell midsurface at location 0. The inner surface of the shell is at -1 and the outer surface is at +1.

LOBATTO INTEGRATION RULE					
NUMBER OF INTEG. POINT	-	-	3 POINT	4 POINT	5 POINT
#1			.0	-1.0	.0
#2			-1.0	-.4472136	-1.0
#3			+1.0	+.4472136	-.6546537
#4				+1.0	+.6546537
#5					+1.0
NUMBER OF INTEG. POINT	6 POINT	7 POINT	8 POINT	9 POINT	10 POINT
#1	-1.0	-1.0	-1.0	-1.0	-1.0
#2	-.7650553	-.8302239	-.8717401	-.8997580	-.9195339
#3	-.2852315	-.4688488	-.5917002	-.6771863	-.7387739
#4	+.2852315	.0	-.2092992	-.3631175	-.4779249
#5	+.7650553	+.4688488	+.2092992	.0	-.1652790
#6	+1.0	+.8302239	+.5917002	+.3631175	+.1652790
#7		+1.0	+.8717401	+.6771863	+.4779249
#8			+1.0	+.8997580	+.7387739
#9				+1.0	+.9195339
#10					+1.0

Location of through thickness Lobatto integration points. The coordinate is referenced to the shell midsurface at location 0. The inner surface of the shell is at -1 and the outer surface is at +1.

Remarks:

- For implicit calculations the following element choices are implemented:

EQ.1: Hughes-Liu,
 EQ.2: Belytschko-Tsay (default),
 EQ.6: S/R Hughes-Liu,
 EQ.10: Belytschko-Wong-Chiang,
 EQ.12: Plane stress (x-y plane),
 EQ.13: Plane strain (x-y plane)
 EQ.15: Axisymmetric solid (y-axis of symmetry) - volume weighted,
 EQ.16: Fully integrated shell element,
 EQ.17: Fully integrated DKT, triangular shell element,
 EQ.18: Taylor 4-node quadrilateral and 3-node triangle (linear only)
 EQ.20: Wilson 3 & 4-node DSE quadrilateral (linear only)
 EQ.21: Fully integrated linear assumed strain C0 shell (5 DOF).
 EQ.22: Linear shear panel element (3 DOF per node)

- EQ.25: Belytschko-Tsay shell with thickness stretch.
EQ.26: Fully integrated shell element with thickness stretch.
EQ.27: Triangle with thickness stretch.

If another element formulation is requested, LS-DYNA will substitute one of the above in place of the one chosen.

2. The linear elements consist of an assembly of membrane and plate elements. The elements have six d.o.f. per node and can therefore be connected to beams, or used in complex shell surface intersections. All elements possess the required zero energy rigid body modes and have exact constant strain and curvature representation, i.e. they pass all the first order patch tests. In addition, the elements have behavior approaching linear bending (cubic displacement) in the plate-bending configuration.
 - a. The membrane component of all elements is based on an 8-node/6-node isoparametric mother element which incorporates nodal in-plane rotations through cubic displacement constraints of the sides [Taylor 1987; Wilson 2000].
 - b. The plate component of element 18 is based on the Discrete Kirchhoff Quadrilateral (DKQ) [Batoz 1982]. Because the Kirchhoff assumption is enforced, the DKQ is transverse shear rigid and can only be used for thin shells. No transverse shear stress information is available. The triangle is based on a degeneration of the DKQ. This element sometimes gives slightly lower eigenvalues when compared with element type 20.
 - c. The plate component of element 20 is based on the 8-node serendipity element. At the mid-side, the parallel rotations and transverse displacements are constrained and the normal rotations are condensed to yield a 4-node element. The element is based on thick plate theory and is recommended for thick and thin plates.
 - d. The quadrilateral elements contain a warpage correction using rigid links.
 - e. The membrane component of element 18 has a zero energy mode associated with the in-plane rotations. This is automatically suppressed in a non-flat shell by the plate stiffness of the adjacent elements. Element 20 has no spurious zero energy modes.
3. The linear shear panel element resist tangential in plane shearing along the four edges and can only be used with the elastic material constants of *MAT_ELASTIC. Membrane forces and out-of-plane loads are not resisted.
4. Element type 99 is intended for vibration studies carried out in the time domain. These models may have very large numbers of elements and may be run for relatively long durations. The purpose of this element is to achieve substantial CPU savings. This is achieved by imposing strict limitations on the range of applicability, thereby simplifying the calculations:
 - Elements must be rectangular; all edges must parallel to the global X-, Y- or Z-axis;
 - Small displacement, small strain, negligible rigid body rotation;
 - Elastic material only

If these conditions are satisfied, the performance of the element is similar to the fully integrated shell (ELFORM=16) but at less CPU cost than the default Belytschko-Tsay shell element (ELFORM=2). Single element torsion and in-plane bending modes are included; meshing guidelines are the same as for fully integrated shell elements.

No damping is included in the element formulation (e.g. volumetric damping). It is strongly recommended that damping be applied, e.g. *DAMPING_PART_MASS or *DAMPING_FREQUENCY_RANGE.

- 5. SHELL_EFG formulation is only available for the explicit analysis.
- 6. Loads, lumped masses, discrete element stiffnesses, etc. in axisymmetric simulations are interpreted as values per unit length (circumferentially) in the case where shell formulation 14 is invoked and per unit radian in the case where shell formulation 15 is used.
- 7. Shell element formulation 25 and 26 are the Belytschko-Tsay element and fully integrated shell element with two additional degrees of freedom that allows a linear variation of strain through the thickness. By default, the thickness field is continuous across the element edges implying that there can be no complex intersections since this would lock up the structure. It assumes a relatively flat surface and is intended primarily for sheets in metal forming. By specifying IDOF=2, the thickness field is decoupled between elements which makes the element suited for crash. If there are any thickness stretch triangles (formulation 27), IDOF must be set to 2.
- 8. Users must input a set of nodes along one of the transverse edges of a seatbelt. If there is no retractor associated with a belt, the node set can be on either edge. If the retractor exists, the edge should be on the retractor side and input in the same sequence of retractor node set. Therefore, another restriction on the seatbelt usage is each belt has its own section definition and a different part.

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$ *SECTION_SHELL
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define a shell section that specifies the following:
$   elform = 10 Belytschko-Wong-Chiang shell element formulation.
$   nip = 3 Three through the shell thickness integration points.
$   t1 - t4 = 2.0 A shell thickness of 2 mm at all nodes.
$
*SECTION_SHELL
$
$.>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
$   sid   elform   shrf   nip   propt   qr/irid   icomp
$       1         10         3.0000
$
$   t1     t2     t3     t4     nloc
$   2.0    2.0    2.0    2.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

*SECTION

*SECTION_SOLID

*SECTION_SOLID_{OPTION}

Available options include:

<BLANK>

ALE

EFG

such that the keyword cards appear:

*SECTION_SOLID

*SECTION_SOLID_ALE

*SECTION_SOLID_EFG

Purpose: Define section properties for solid continuum and fluid elements.

Card 1 define for all options

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	ELFORM	AET					
Type	A8	I	I					
Remark		1, 2						

Card 2 define only for the ALE option.

Also see *ALE_SMOOTHING for the smoothing definition.

Card 2 1 2 3 4 5 6 7 8

Variable	AFAC	BFAC	CFAC	DFAC	START	END	AAFAC	
Type	F	F	F	F	F	F	F	

SECTION_SOLID**SECTION****Define only for the EFG option.**

Also see *CONTROL_EFG. See Remark 7.

Card 2 1 2 3 4 5 6 7 8

Variable	DX	DY	DZ	ISPLINE	IDILA	IEBT	IDIM	TOLDEF
Type	F	F	F	I	I	I	I	F
Default	1.01	1.01	1.01	0	0	1	1	0.01

Define if and only if ELFORM=101,102,103,104 or 105.

Also see Appendix C

Card 1 2 3 4 5 6 7 8

Variable	NIP	NXDOF	IHGF	ITAJ	LMC	NHSV		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Define NIP cards according to the following format.

Also see Appendix C

Card 1 2 3 4 5 6 7 8

Variable	XI	ETA	ZETA	WGT				
Type	F	F	F	F				
Default	None	None	None	None				

Define LMC property parameters using 8 parameters per card.

Also see Appendix C

Card	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	0	0	0	0	0	0	0	0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label not exceeding 8 characters must be specified.
ELFORM	<p>Element formulation options, (see Remark 3 below):</p> <p>EQ.0: 1 point corotational for *MAT_MODIFIED_HONEYCOMB. See remark 4.</p> <p>EQ.1: constant stress solid element (default),</p> <p>EQ.2: fully integrated S/R solid. See remark 5 below,</p> <p>EQ.3: fully integrated quadratic 8 node element with nodal rotations,</p> <p>EQ.4: S/R quadratic tetrahedron element with nodal rotations,</p> <p>EQ.5: 1 point ALE,</p> <p>EQ.6: 1 point Eulerian,</p> <p>EQ.7: 1 point Eulerian ambient,</p> <p>EQ.8: acoustic,</p> <p>EQ.9: 1 point corotational for *MAT_MODIFIED_HONEYCOMB. See remark 4.</p> <p>EQ.10: 1 point tetrahedron.</p> <p>EQ.11: 1 point ALE multi-material element</p> <p>EQ.12: 1 point integration with single material and void.</p> <p>EQ.13: 1 point nodal pressure tetrahedron. See remark 14 below.</p> <p>EQ.14: 8 point acoustic</p> <p>EQ.15: 2 point pentahedron element.</p> <p>EQ.16: 4 or 5 point 10-noded tetrahedron (See Remark 13)</p> <p>EQ.17: 10-noded composite tetrahedron</p> <p>EQ.18: 8 point enhanced strain solid element for linear statics only</p> <p>EQ.19: 4 point cohesive element</p> <p>EQ.20: 4 point cohesive elements with offsets for use with shells</p> <p>EQ.41: Mesh-free solid formulation</p> <p>EQ.99: simplified linear element for time-domain vibration studies. See remarks.</p> <p>GT.100.and.LT.106: User defined solid</p>

VARIABLE	DESCRIPTION
AET	Ambient Element type: Can be defined for ELFORM 7, 11 and 12. EQ.1: temperature (not currently available), EQ.2: pressure and temperature (not currently available), EQ.3: pressure outflow, EQ.4: pressure inflow. (Default for ELFORM 7)
AFAC	Smoothing weight factor - Simple average: EQ.-1: turn smoothing off.
BFAC	Smoothing weight factor - Volume weighting
CFAC	Smoothing weight factor - Isoparametric
DFAC	Smoothing weight factor - Equipotential
START	Start time for smoothing
END	End time for smoothing
AAFAC	ALE advection factor
DX, DY, DZ	Normalized dilation parameters of the kernel function in X, Y and Z directions. The normalized dilation parameters of the kernel function are introduced to provide the smoothness and compact support properties on the construction of the mesh-free shape functions. Values between 1.0 and 1.5 are recommended. Values smaller than 1.0 are not allowed. Larger values will increase the computation time and will sometimes result in a divergence problem. See Remark 7.
ISPLINE	Replace the choice for the EFG kernel functions definition in *CONTROL_EFG. This allows users to define different ISPLINE in different sections. EQ.0: Cubic spline function (default). EQ.1: Quadratic spline function. EQ.2: Cubic spline function with circular shape.
IDILA	Replace the choice for the normalized dilation parameter definition in *CONTROL_EFG. This allows users to define different IDILA in different sections. EQ.0: Maximum distance based on the background elements. EQ.1: Maximum distance based on surrounding nodes.
IEBT	Essential boundary condition treatment: See Remark 10 and 11. EQ. 1: Full transformation method (default) EQ.-1: (w/o transformation) EQ. 2: Mixed transformation method EQ. 3: Coupled FEM/EFG method

VARIABLE	DESCRIPTION
	EQ. 4: Fast transformation method EQ.-4: (w/o transformation) EQ.5: Fluid particle method for E.O.S and *MAT_ELASTIC_FLUID materials
IDIM	Domain integration method: See Remark 12. EQ.1: Local boundary integration (default) EQ.2: Two-point Gauss integration EQ.3: Improved Gauss integration for IEBT=4 or -4
TOLDEF	Deformation tolerance for the activation of adaptive EFG Semi-Lagrangian and Eulerian kernel. See Remark 13. = 0.0: Lagrangian kernel > 0.0: Semi_Lagrangian kernel < 0.0: Eulerian kernel
NIP	Number of integration points for user-defined solid (0 if resultant/discrete element)
NXDOF	Number of extra degrees of freedom per node for user-defined solid
IHGF	Flag for using hourglass stabilization (NIP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used EQ.3: Same as 2, but the resultant material tangent modulus is passed
ITAJ	Flag for setting up finite element matrices (NIP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
ZETA	Third isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

Remarks:

1. The keyword ***CONTROL_SOLID** activates automatic sorting of tetrahedron and pentahedron elements into type 10 and 15 element formulation, respectively. These latter elements are far more stable than the degenerate solid element. The sorting is performed internally and is transparent to the user.
2. For implicit calculations the following element choices are implemented:
 - EQ.1: constant stress solid element,
 - EQ.2: fully integrated S/R solid. See remark 5 below,
 - EQ.3: fully integrated 8 node solid with rotational DOFs,
 - EQ.4: fully integrated S/R 4 node tetrahedron with rotational DOFs,
 - EQ.10: 1 point tetrahedron.
 - EQ.13: 1 point nodal pressure tetrahedron.
 - EQ.15: 2 point pentahedron element.
 - EQ.16: 5 point 10 noded tetrahedron
 - EQ.18: 8 point enhanced strain solid element for linear statics only,

If another element formulation is requested, LS-DYNA will substitute, when possible, one of the above in place of the one chosen. The type 1 element, constant stress, is generally much more accurate than the type 2 element, the selective reduced integrated element for implicit problems.

3. Element formulations 0 and 9, applicable only to ***MAT_MODIFIED_HONEYCOMB**, behave essentially as nonlinear springs so as to permit severe distortions sometimes seen in honeycomb materials. In formulation 0, the local coordinate system follows the element rotation whereas in formulation 9, the local coordinate system is based on axes passing through the centroids of the element faces. Formulation 0 is preferred for severe shear deformation where the barrier is fixed in space. If the barrier is attached to a moving body, which can rotate, then formulation 9 is usually preferred.
4. The selective reduced integrated solid element, element type 2, assumes that pressure is constant throughout the element to avoid pressure locking during nearly incompressible flow. However, if the element aspect ratios are poor, shear locking will lead to an excessively stiff response. A better choice, given poor aspect ratios, is the one point solid element which work well for implicit and explicit calculations. For linear statics, the type 18 enhanced strain element works well with poor aspect ratios. Please note that highly distorted elements should always be avoided since excessive stiffness will still be observed even in the enhanced strain formulations.
5. Element type 99 is intended for vibration studies carried out in the time domain. These models may have very large numbers of elements and may be run for relatively long durations. The purpose of this element is to achieve substantial CPU savings. This is achieved by imposing strict limitations on the range of applicability, thereby simplifying the calculations:
 - Elements must be cubed; all edges must parallel to the global X-, Y- or Z-axis;
 - Small displacement, small strain, negligible rigid body rotation;
 - Elastic material only

If these conditions are satisfied, the performance of the element is similar to the fully integrated S/R solid (ELFORM=2) but at less CPU cost than the default solid element (ELFORM=1). Single element bending and torsion modes are included, so meshing guidelines are the same as for fully integrated solids – e.g. relatively thin structures can be modeled with a single solid element through the thickness if required. Typically, the CPU requirement per element-cycle is roughly two thirds that of the default solid element.

No damping is included in the element formulation (e.g. volumetric damping). It is strongly recommended that damping be applied, e.g. *DAMPING_PART_MASS or *DAMPING_FREQUENCY_RANGE.

6. The current EFG formulation performs automatic sorting for finite element tetrahedral, pentahedron and hexahedral elements as the background mesh to identify the mesh-free geometry and provide the contact surface definition in the computation.
7. Element type 19 is a cohesive element. The tractions on the mid-surface defined as the mid-points between the nodal pairs 1-5, 2-6, 3-7, and 4-8 are functions of the differences of the displacements between nodal pairs interpolated to the four integration points. The initial volume of the cohesive element may be zero, in which case, the density may be defined in terms of the area of nodes 1-2-3-4. See Appendix A and the user material description for additional details.

The tractions are calculated in the local coordinate system defined at the centroid of the element, see the Figure below. Defining the rotation matrix from the local to the global coordinate system at time t as $R(t)$, the initial coordinates as X , and the current coordinates as x , the displacements at an integration point are

$$\begin{aligned}\Delta u &= R^T(t)\Delta x - R^T(0)\Delta X \\ \Delta x &= \sum_{i=1}^4 N_i(s,t)\Delta x_{i+4,i} \\ \Delta X &= \sum_{i=1}^4 N_i(s,t)\Delta X_{i+4,i}\end{aligned}$$

The forces are obtained by integrating the tractions over the midsurface, and rotating them into the global coordinate system.

$$F_i = R(t) \sum_{g=1}^4 t_g N_i(s_g, t_g) \det(J_g) \text{ for } i=1,4 \text{ and } F_{i+4} = -F_i$$

8. Element type 20 is identical to element 19 but with offsets for use with shells. The element is assumed to be centered between two layers of shells on the cohesive element's lower (1-2-3-4) and upper (5-6-7-8) surfaces. The offset distances for both shells are one half the initial thicknesses of the nodal pairs (1-5, 2-6, 3-7, and 4-8) separating the two shells. These offsets are used with the nodal forces to calculate moments that are applied to the shells.

- 9. The mixed transformation method, the coupled FEM/EFG method and the fast transformation method were implemented in EFG 3D solid formulation. These three new features were added to improve the efficiency on the imposition of essential boundary conditions and the transfer of real nodal values and generalized nodal values. The mixed transformation method is equivalent to the full transformation method with improved efficiency. The behavior of the coupled FEM/EFG method is between FEM and EFG. The fast transformation method provides the most efficient and robust results.
- 10. Current fluid particle formulation (IEBT=5) only supports for the 4-noded background element.
- 11. For compressible material like foam and soil, IDIM=1 (default) is recommended. For nearly incompressible material like metal and rubber, IDIM=2 is recommended.
- 12. This parameter is introduced to improve the negative volume problem usually seen in the large deformation analysis. For the same analysis, the larger value of Toldef, the earlier Semi-Lagrangian or Eulerian kernel is introduced into the EFG computation and more cpu time is expected. Value between 0.0 and 0.1 is suggested in the crashworthiness analysis. Semi-Lagrangian kernel is suggested for the solid materials and Eulerian kernel is suggested for the fluid and E.O.S. materials.

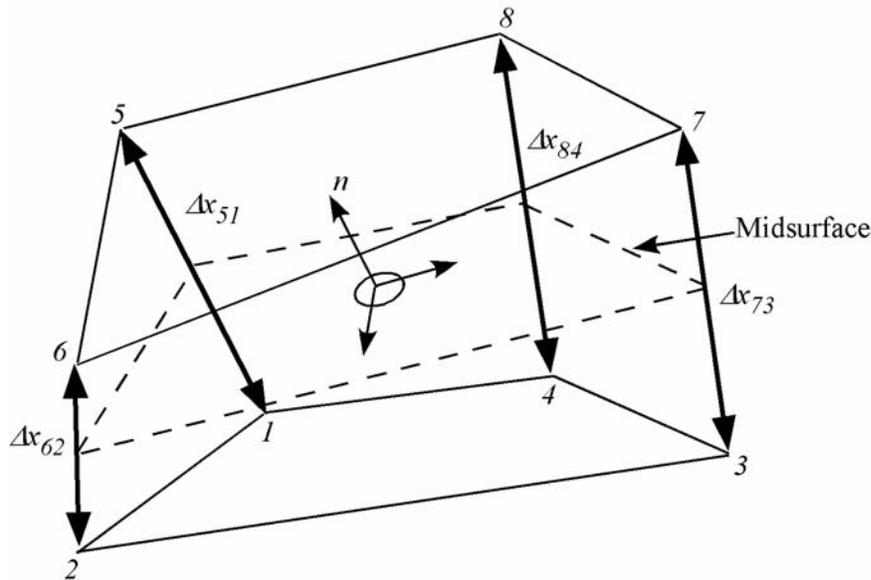


Figure 29.2

- 13. Formulations 16 and 17 are 10-noded, tetrahedral formulations. The parameter NIPTETS in *CONTROL_SOLID controls the number of integration points for these formulations. Formulation 17 is generally preferred over formulation 16 because, unlike 16, the nodal weighting factors are equal and thus nodal forces from contact and applied pressures are distributed correctly.

When applying loads to 10-noded tetrahedrons via segments, no load will be applied to the midside nodes if the segments contain only corner nodes. When defining contact, it is recommended that *CONTACT_AUTOMATIC_... be used and the contact surface of the 10-noded tetrahedral part be specified by its part ID. In this manner, midside nodes receive contact forces.

If the 10-noded element connectivity is not defined in accordance with the figure shown in *ELEMENT_SOLID, the order of the nodes can be quickly changed via a permutation vector specified with *CONTROL_SOLID. If *ELEMENT_SOLID defines 4-noded tetrahedrons, you can easily convert to 10-noded tetrahedrons using the command *ELEMENT_SOLID_TET4TOTET10. Because the characteristic length of a 10-noded tetrahedron is half that of a 4-noded tetrahedron, the time step for the tetrahedrons will be smaller by a factor of 2. The parameter TET10 in 971, when set to 1 in *CONTROL_OUTPUT, causes the full 10-node connectivity to be written to the d3plot and d3part databases.

- 14. Element type 13 is identical with type 10 but with additional averaging of nodal pressures, which significantly lowers volumetric locking. Therefore, it is well suited for applications with incompressible and nearly incompressible material behavior, i.e. rubber materials or ductile metals with isochoric plastic deformations (e.g. bulk forming). Compared to the standard tetrahedron (type 10), a speed penalty of max. 25 % can be observed. Currently, material models 1, 3, 6, 24, 27, 81, 82, 120, 123, 124 and 181 are fully supported (materials 81, 82, 120, 123 and 124 are available starting with the R3 release of Version 971), for other materials this element behaves like type 10.

(Note: NODE_SET option is available starting with the R3 release of Version 971.)

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$ *SECTION_SOLID
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ A bolt modeled with solids was found to have excessive hourglassing.
$ Thus, the section (sid = 116) associated with the bolt part was used
$ to specify that a fully integrated Selectively-Reduced solid element
$ formulation be used to totally eliminate the hourglassing (elform = 2).
$
*SECTION_SOLID
$.>...1.>...2.>...3.>...4.>...5.>...6.>...7.>...8
$   sid   elform
$   116     2
$
*PART
bolts
$   pid     sid     mid     eosid     hgid     adpopt
$   17      116     5
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

```

***SECTION_SPH_{OPTION}**

Available options include:

<BLANK>

TENSOR

USER

such that the keyword cards appear:

***SECTION_SPH**

***SECTION_SPH_TENSOR**

***SECTION_SPH_USER**

Purpose: Define section properties for SPH particles. For the **USER** option, see remark 3.

Card 1 define for all options

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	CSLH	HMIN	HMAX	SPHINI	DEATH	START	
Type	A8	F	F	F	F	F	F	
Default	none	1.2	0.2	2.0	0.0	1.e20	0.0	

Card 2 define only for the TENSOR option.

Card 2 1 2 3 4 5 6 7 8

Variable	HXCSLH	HYCSLH	HZCSLH	HXINI	HYINI	HZINI		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label not exceeding 8 characters must be specified.
CSLH	Constant applied to the smoothing length of the particles. The default value applies for most problems. Values between 1.05 and 1.3 are acceptable. Taking a value less than 1 is inadmissible. Values larger than 1.3 will increase the computational time. The default value is recommended.
HMIN	Scale factor for the minimum smoothing length (See Remark 1)
HMAX	Scale factor for the maximum smoothing length (See Remark 1)
SPHINI	Optional initial smoothing length (overrides true smoothing length). This option applies to avoid LS-DYNA to calculate the smoothing length during initialization. In this case, the variable CSLH doesn't apply.
DEATH	Time imposed SPH approximation is stopped.
START	Time imposed SPH approximation is activated.
HXCSLH	Constant applied for the smoothing length in the X direction for the tensor case.
HYCSLH	Constant applied for the smoothing length in the Y direction for the tensor case.
HZCSLH	Constant applied for the smoothing length in the Z direction for the tensor case.
HXINI	Optional initial smoothing length in the X direction for the tensor case (overrides true smoothing length)
HYINI	Optional initial smoothing length in the Y direction for the tensor case (overrides true smoothing length)
HZINI	Optional initial smoothing length in the Z direction for the tensor case (overrides true smoothing length)

Remarks:

1. The SPH processor in LS-DYNA uses a variable smoothing length. LS-DYNA computes the initial smoothing length, h_0 , for each SPH part by taking the maximum of the minimum distance between every particle. Every particle has its own smoothing length which varies in time according to the following equation:

$$\frac{d}{dt}(h(t)) = h(t)div(v)$$

$h(t)$ is the smoothing length, $div(v)$ is the divergence of the flow. The smoothing length increases when particles separate from each other and reduces when the concentration of particles is important. It varies to keep the same number of particles in the neighborhood. The smoothing length varies between the minimum and maximum values

$$HMIN * h_0 < h(t) < HMAX * h_0$$

Defining a value of 1 for HMIN and 1 for HMAX will result in a constant smoothing length in time and space.

2. SPH is implemented for explicit applications.
3. The USER option allows the definition of customized subroutine for the variation of the smoothing length. A subroutine called *hdot* is defined in the file dyn21.F (Unix/linux) or lsdyna.f (Windows).

*SECTION

*SECTION_TSHELL

*SECTION_TSHELL

Purpose: Define section properties for thick shell elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SECID	ELFORM	SHRF	NIP	PROPT	QR	ICOMP	
Type	A8	I	F	F	F	F	I	
Default	none	1	1.0	2	1	0	0	

Optional Section Cards if ICOMP=1 define NIP angles putting 8 on each card.

Card 2... 1 2 3 4 5 6 7 8

Variable	B1	B2	B3	B4	B5	B6	B7	B8
Type	F	F	F	F	F	F	F	F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label not exceeding 8 characters must be specified.
ELFORM	Element formulation: EQ.1: one point reduced integration (default), EQ.2: selective reduced 2 ∞ 2 in plane integration. EQ.3: assumed strain 2 ∞ 2 in plane integration, see remark below.
SHRF	Shear factor. A value of 5/6 is recommended.
NIP	Number of through shell thickness integration points: EQ.0: set to 2 integration points.
PROPT	Printout option: EQ.1.0: average resultants and fiber lengths, EQ.2.0: resultants at plan points and fiber lengths, EQ.3.0: resultants, stresses at all points, fiber lengths.

VARIABLE	DESCRIPTION
QR	Quadrature rule: LT.0.0: absolute value is specified rule number, EQ.0.0: Gauss (up to five points are permitted), EQ.1.0: trapezoidal, not recommended for accuracy reasons.
ICOMP	Flag for layered composite material mode: EQ.1: a material angle is defined for each through thickness integration point. For each layer one integration point is used.
B1	β_1 , material angle at first integration point. The same procedure for determining material directions is use for thick shells that is used for the 4 node quadrilateral shell.
B2	β_2 , material angle at second integration point
B3	β_3 , material angle at third integration point
.	.
.	.
.	.
B8	β_8 , material angle at eighth integration point
.	.
Bnip	β_{nip} , material angle at niph integration point

Define as many cards as necessary until NIP points are defined.

Remarks:

1. Thick shell formulation type 3 uses a full three-dimensional stress update rather than the two-dimensional plane stress update of types 1 and 2. The type 3 element is distortion sensitive and should not be used in situations where the elements are badly shaped. With element types 1 and 2 a single element through the thickness will capture bending response, but with element type 3 two are recommended to avoid excessive softness.
2. These elements are available for implicit applications.

***SECTION**

***SECTION_TSHELL**

***SENSOR**

The keyword ***SENSOR** provides a convenient way of activating and deactivating boundary conditions, airbags, discrete elements, joints, contact, rigid walls, single point constraints, and constrained nodes. The sensor capability is new in the second release of version 971 and will evolve in later releases to encompass many more LS-DYNA capabilities and replace some of the existing capabilities such as the airbag sensor logic. The keyword control cards in this section are defined below in alphabetical order:

***SENSOR_CONTROL**

***SENSOR_DEFINE_CALC-MATH**

***SENSOR_DEFINE_ELEMENT**

***SENSOR_DEFINE_FORCE**

***SENSOR_DEFINE_NODE**

***SENSOR_SWITCH**

***SENSOR_SWITCH_CALC-LOGIC**

To define a sensor, three categories of sensor keyword cards are needed as shown in Figure 30.1.

1. Sensor definitions using the, ***SENSOR_DEFINE** keywords, which can be combined with the mathematical calculation cards, ***SENSOR_DEFINE_CALC-MATH**, for more complicated definitions. This category of keyword cards yield a numerical value to be referred by ***SENSOR_SWITCH** as a switching criterion.
 - ***SENSOR_DEFINE**
This card defines the sensor location and types by node ID, element ID, or force-type ID.
 - ***SENSOR_DEFINE_CALC-MATH**
This keyword card defines a new sensor ID obtained by performing mathematical calculations on the information from **SENSOR_DEFINE** definitions.
2. Sensor switching criterion definition using the, ***SENSOR_SWITCH**, keyword, which can be combined with the logical calculation cards, ***SENSOR_SWITCH_CALC-LOGIC**, for more complicated definitions. The logic value yielded by this category of cards can be referred by ***SENSOR_CONTROL** to determine if a status switch condition is met.
 - ***SENSOR_SWITCH**
This card compares the numerical value from ***SENSOR_DEFINE** or ***SENSOR_DEFINE_CALC-MATH** with the given criterion to see if a switching condition is met.

*SENSOR

- *SENSOR_SWITCH_CALC-LOGIC
This card performs logical calculation on the information from SENSOR_SWITCH.
3. Sensor control definition, *SENSOR_CONTROL. This category of cards determines how and what to switch based on the logical values from *SENSOR_SWITCH and/or *SENSOR_SWITCH_CALC-LOGIC.

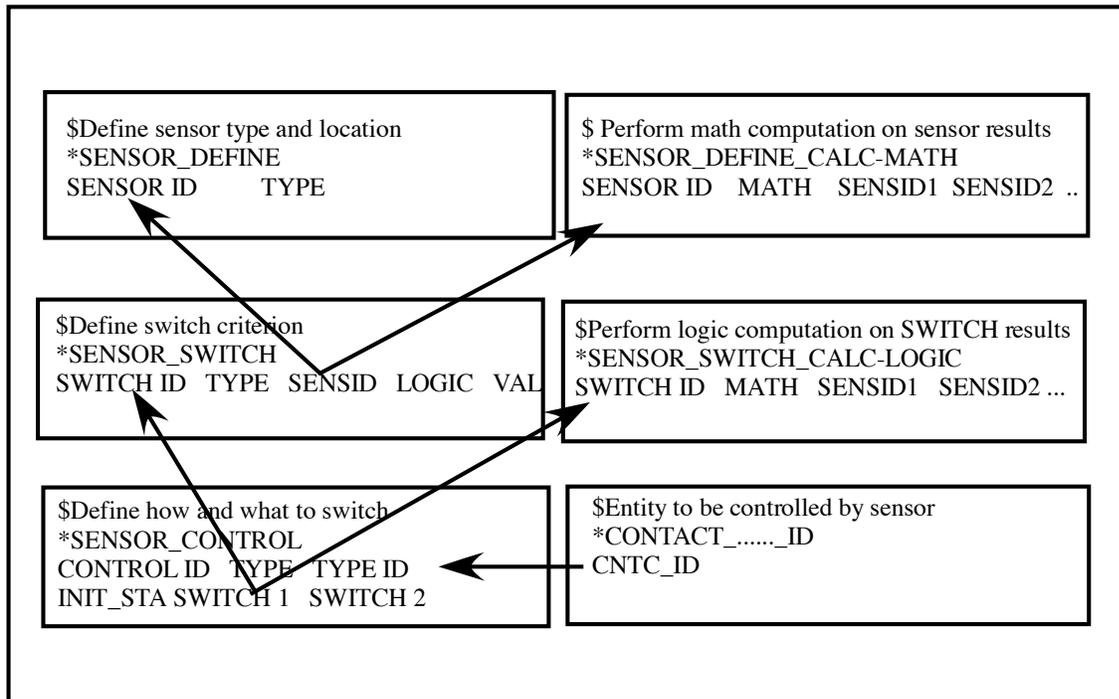


Figure 30.1. Relationship between sensor keyword definitions.

***SENSOR_CONTROL**

Purpose: This command, based on the information of *SENSOR_SWITCH, controls the status, on or off, of an entity like *CONTACT, *AIRBAG.

Card 1 1 2 3 4 5 6 7 8

Variable	CNTLID	TYPE	TYPEID					
Type	I	A	I					

Card 2

Variable	INITSTT	SWIT1	SWIT2	SWIT3	SWIT4	SWIT5	SWIT6	SWIT7
Type	A	I	I	I	I	I	I	I

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CNTLID	Control ID.
TYPE	Entity to be controlled: EQ.AIRBAG: for *AIRBAG EQ.PRESC-MOT: for *BOUNDARY_PRESCRIBED MOTION EQ.DISC-ELE: for *ELEMENT_DISCRETE EQ.JOINT: for *CONSTRAINED_JOINT EQ.JOINTSTIF: for *CONSTRAINED_JOINT_STIFFNESS EQ.CONTACT: for *CONTACT EQ.RWALL: for *RIGID_WALL EQ.SPC: for *BOUNDARY_SPC EQ.SPOTWELD: for *CONSTRAINED_POINTS
TYPEID	ID of entity to be controlled.
INITSTT	Initial status: EQ.On: EQ.Off:
SWIT1	ID of switch which will change the initial status after its condition is met.
SWITn	ID of nth switch which will change the status set by switch n-1 after its condition is met.

*SENSOR

*SENSOR_DEFINE_CALC-MATH

*SENSOR_DEFINE_CALC-MATH

Purpose: Defines a new sensor with a unique ID. The values associated with this sensor are computed by performing mathematical calculations with the information obtained from sensors defined by the *SENSOR_DEFINE_OPTION.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	CALC	SENS1	SENS2	SENS3	SENS4	SENS5	SENS6
Type	I	A	I	I	I	I	I	I

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SENSID	Sensor ID.
CALC	Mathematical calculation, See Table 30.1.
SENSi	<i>i</i> th Sensor ID

CALC	DESCRIPTION	MATHEMATICAL EXPRESSION
ABSSUM	Absolute value of the sum of sensor values	$ \text{sens1}+\text{sens2}+\dots $
MIN	The minimum of sensor values	$\text{Min}(\text{sens1}, \text{sens2}, \dots)$
MAX	The maximum of sensor values	$\text{Max}(\text{sens1}, \text{sens2}, \dots)$
MAXMAG	The maximum of magnitude of sensor values	$\text{Max}(\text{sens1} , \text{sens2} , \dots)$
MINMAG	The minimum of the magnitude of sensor values	$\text{Min}(\text{sens1} , \text{sens2} , \dots)$
MULTIPLY	Multiplication of sensor values; negative for division (performed left to right)	$\text{sens1} * \text{sens2} * \text{sens3} \dots$
SQRE	Summation of squared values of sensor values	$\text{Sens1}^2 + \text{sens2}^2 \dots$
SQRTSQRE	Square root of the sum of squared values	$\text{SQRT}(\text{sens1}^2 + \text{sens2}^2 + \dots)$
SQRT	Summation of square root of sensor values; negative for subtracting values	$(\text{sens1})^{**0.5} + (\text{sens2})^{**0.5} \dots$
SUMABS	Summation of absolute sensor values	$ \text{sens1} + \text{sens2} + \dots$
SUM	Summation of sensor values; negative for subtracting values	$\text{sens1} + \text{sens2} + \dots$

Table 30.1. Available mathematical functions.

*SENSOR

*SENSOR_DEFINE_ELEMENT

*SENSOR_DEFINE_ELEMENT

Purpose: Define a strain gage type element sensor that checks the stress, strain, or resultant force of an element.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	ETYPE	ELEMID	COMP	CTYPE	LAYER		
Type	I	A	I	A	A	A		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SENSID	Sensor ID.
ETYPE	Element type: EQ.BEAM: beam element. EQ.SHELL: shell element EQ.SOLID: solid element EQ.DISC-ELE: discrete element
ELEMID	Element ID
COMP	Element type: EQ.XX: x-normal component for shells and solids EQ.YY: y-normal component for shells and solids EQ.ZZ: z-normal component for shells and solids EQ.XY: xy-shear component for shells and solids EQ.YZ: yz-shear component for shells and solids EQ.ZX: zx-shear component for shells and solids EQ:AXIAL: axial EQ:SHEARS: local s-direction EQ:SHEART: local t-direction EQ: : leave blank for discrete elements
CTYPE	Component type: EQ.STRAIN: strain component for shells and solids EQ.STRESS: stress component for shells and solids EQ.FORCE: force resultants for beams EQ.MOMENT: moment resultants for beams EQ.FORCE: discrete element force EQ.DLEN: change in length for discrete element
LAYER	Layer of integration point in shell element EQ.BOT: component at lower surface EQ.TOP: component at upper surface

***SENSOR_DEFINE_FORCE**

Purpose: Define a force transducer type sensor.

Card 1 2 3 4 5 6 7 8

Variable	SENSID	FTYPE	TYPEID	VID	CRD			
Type	I	A	I	A/I	I			

VARIABLE	DESCRIPTION
SENSID	Sensor ID.
FTYPE	Force type. See Table 30.2.
TYPEID	ID defined in the associated KEYWORD command. See Table 30.2.
VID	Vector along which the forces is measured. EQ.X: x-direction in coordinate system CRD. EQ.Y: y-direction in coordinate system CRD. EQ.Z: z-direction in coordinate system CRD. EQ.XMOMENT: x-direction moment for JOINT. EQ.YMOMENT: y-direction moment for JOINT. EQ.ZMOMENT: z-direction moment for JOINT. EQ.n: vector ID n in coordinate system CRD.
CRD	Coordinate system, defined by *DEFINE_COORDINATE_NODES, to which VECT is attached.

FTYPE	TYPEID (Enter ID defined in following KEYWORD commands)	OUTPUT	ASCII FILE
AIRBAG	*AIRBAG	Airbag pressure	ABSTAT
CONTACT	*CONTACT	Contact force	RCFORC
JOINT	*CONSTRAINED_JOINT	Joint force	JNTFORC
JOINTSTIF	*CONSTRAINED_JOINT_STIFFNESS	Joint stiffness force	NA
PRESC-MOT	*BOUNDARY_PRESCRIBED_MOTION	Prescribed motion force	BNDOUT
RWALL	*RIGIDWALL	Rigid wall force	RWFORC
SPC	*BOUNDARY_SPC	SPC reaction force	SPCFORC
SPOTWELD	*CONSTRAINED_POINTS	Spot weld force	SWFORC
X-SECTION	*DATABASE_CROSS_SECTION	Joint force	SECFORC

Table 30.2. Force transducer type sensor

***SENSOR_DEFINE_NODE**

Purpose: Define an accelerometer type sensor. This command outputs the relative linear acceleration, velocity, or relative coordinate of node-1 with respect to node-2 along vector VID, which is fixed in coordinate-system CRD.

Card 1 2 3 4 5 6 7 8

Variable	SENSID	NODE1	NODE2	VID	CRD	CTYPE		
Type	I	I	I	A/I	I	A		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SENSID	Sensor ID.
NODE1,2	Nodes defining the accelerometer.
VID	Vector along which the forces is measured: EQ.X: x-direction in coordinate system CRD. EQ.Y: y-direction in coordinate system CRD. EQ.Z: z-direction in coordinate system CRD. EQ.n: vector ID n in coordinate system CRD.
CRD	Coordinate system, defined by *DEFINE_COORDINATE_NODES, to which VECT is attached.
CTYPE	Output component type: EQ.ACC: acceleration EQ.VEL: velocity EQ.COORD: displacement

*SENSOR

*SENSOR_SWITCH

*SENSOR_SWITCH

Purpose: This command compares the value of a sensor, *SENSOR_DEFINE or SENSOR_CALC-MATH, to a given criterion to check if the switch condition is met. It outputs a logic value of TRUE or FALSE.

Card	1	2	3	4	5	6	7	8
Variable	SWITID	TYPE	SENSID	LOGIC	VALUE	FILTRID	TIMWIN	
Type	I	A	I	A	F	I	F	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SWITID	Switch ID can be referred directly by *SENSOR_CONTROL to control the status of entities like CONTACT and AIRBAG, or can be referred to by *SENSOR_SWITCH_CALC-LOGIC for logic computation.
TYPE	Type: EQ.Sensor: EQ.Time:
SENSID	ID of the sensor whose value will be compared to the criterion to determine if a switch condition is met.
LOGIC	Logic: EQ.LT: less than EQ.GT: greater than
VALUE	Critical value
FILTER	Filter option. Not yet implemented
TIMWIN	Trigger a status change when the value given by the sensor is less than or greater than (depending on LOGIC) the VALUE for a duration defined by TIMWIN.

***SENSOR_SWITCH_CALC-LOGIC**

Purpose: This command performs a logic calculation for the logic output of up to seven *SENSOR_SWITCH or *SENSOR_SWITCH_CALC-LOGIC definitions. The output is a logic value of either TRUE or FALSE.

Card 1 2 3 4 5 6 7 8

Variable	SWITID	SWIT1	SWIT2	SWIT3	SWIT4	SWIT5	SWIT6	SWIT7
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SWITID	Switch ID can be referred directly by *SENSOR_CONTROL to control the status of entities like CONTACT and AIRBAG, or can be referred to by *SENSOR_SWITCH_CALC-LOGIC for logic computation.
SWITn	Input a positive sensor ID for "AND" and negative ID for "OR".

***SET**

The keyword ***SET** provides a convenient way of defining groups of nodes, parts, elements, and segments. The sets can be used in the definitions of contact interfaces, loading conditions, boundary conditions, and other inputs. Each set type must have a unique numeric identification. The keyword control cards in this section are defined in alphabetical order:

***SET_BEAM_{OPTION}**

***SET_BEAM_ADD**

***SET_DISCRETE_{OPTION}**

***SET_DISCRETE_ADD**

***SET_MULTI-MATERIAL_GROUP_LIST**

***SET_NODE_{OPTION}**

***SET_NODE_ADD_{OPTION}**

***SET_PART_{OPTION}**

***SET_PART_ADD**

***SET_SEGMENT_{OPTION}**

***SET_2D_SEGMENT_{OPTION}**

***SET_SHELL_{OPTION}**

***SET_SHELL_ADD**

***SET_SOLID_{OPTION}**

***SET_SOLID_ADD**

***SET_TSHELL_{OPTION}**

An additional option **_TITLE** may be appended to all the ***SET** keywords. If this option is used then an addition line is read for each section in 80a format which can be used to describe the set. At present LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

The **GENERAL** option is available for set definitions. In this option, the commands are executed in the order defined. For example, the delete option cannot delete a node or element unless the node or element was previously added via a command such as **BOX** or **ALL**.

*SET

*SET_BEAM

*SET_BEAM_{OPTION}

Available options include:

<BLANK>

GENERATE

GENERAL

The last option, GENERATE, will generate a block of beam element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of beam elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							

Cards 2, 3, 4, ... (OPTION=none) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERATE) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERAL) (The next “*” card terminates the input.) This set is a combination of a series of options: ALL, ELEM, DELEM, PART, DPART, BOX, and DBOX.

Card 2... 1 2 3 4 5 6 7 8

Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID
K1	First beam element
K2	Second beam element
.	.
.	.
KNUM	Last beam element
BNBEG	First beam element ID in block N.
BNEND	Last beam element ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not element ID's.
OPTION	Option for GENERAL. See table below.
E1,...,E7	Specified entity. Each card must have the option specified. See table below.

OPTION	ENTITY (define up to 7)	FUNCTION
ALL		All beam elements will be included in the set.
ELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... will be included.
DELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... previously added will be excluded.
PART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... will be included.
DPART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... previously added will be excluded.
BOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... will be included.
DBOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... previously added will be excluded.

***SET_BEAM_ADD**

Purpose: Define a beam set by combining beam sets.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							
Remark								

Card 2, 3, 4 ... (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	BSID1	BSID2	BSID3	BSID4	BSID5	BSID6	BSID7	BSID8
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SID Set ID of new beam set. All beam sets should have a unique set ID.

BSID n The n th beam set ID

*SET

*SET_DISCRETE

*SET_DISCRETE_{OPTION}

Available options include:

<BLANK>

GENERATE

GENERAL

The last option, GENERATE, will generate a block of discrete element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of discrete elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							

Cards 2, 3, 4, ... (OPTION=none) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERATE) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERAL) (The next “*” card terminates the input.) This set is a combination of a series of options: ALL, ELEM, DELEM, PART, DPART, BOX, and DBOX.

Card 2... 1 2 3 4 5 6 7 8

Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

<u>VARIABLE</u>	<u>DESCRIPTION</u>
-----------------	--------------------

SID	Set ID
K1	First discrete element
K2	Second discrete element
·	· ·
·	· ·
KNUM	Last discrete element
BNBEG	First discrete element ID in block N.
BNEND	Last discrete element ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not element ID's.
OPTION	Option for GENERAL. See table below.
E1,....,E7	Specified entity. Each card must have the option specified. See table below.

OPTION	ENTITY (define up to 7)	FUNCTION
ALL		All discrete elements will be included in the set.
ELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... will be included.
DELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... previously added will be excluded.
PART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... will be included.
DPART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... previously added will be excluded.
BOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... will be included.
DBOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... previously added will be excluded.

SET_DISCRETE_ADD**SET*****SET_DISCRETE_ADD**

Purpose: Define a discrete set by combining discrete sets.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	None							
Remark								

Card 2, 3, 4 ... (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	DSID1	DSID2	DSID3	DSID4	DSID5	DSID6	DSID7	DSID8
Type	I	I	I	I	I	I	I	I

VARIABLE**DESCRIPTION**

SID Set ID of new beam set. All beam sets should have a unique set ID.

DSID n The n th discrete set ID

*SET

*SET_MULTI-MATERIAL_GROUP_LIST

*SET_MULTI-MATERIAL_GROUP_LIST

Purpose: This command defines an ALE multi-material set ID (AMMSID) which contains a collection of one or more ALE multi-material group ID(s) (AMMGID). This provides a means for selecting any specific ALE multi-material(s). Application includes, for example, a selection of any particular fluid(s) to be coupled to a fluid-structure interaction.

Card 1 1 2 3 4 5 6 7 8

Variable	AMSID							
Type	I							
Default	0							

Card 2

Variable	AMGID1	AMGID2	AMGID3	AMGID4	AMGID5	AMGID6	AMGID7	AMGID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE

DESCRIPTION

AMSID	An ALE multi-material set ID (AMSID) which contains a collection of one or more ALE multi-material group ID(s) (AMMGID).
AMGID1	The 1st ALE multi-material group ID (AMGID=1) defined by the 1st data line of the *ALE_MULTI-MATERIAL_GROUP card.
...	...
AMGID8	The 8th ALE multi-material group ID (AMGID=8) defined by the 8th data line of the *ALE_MULTI-MATERIAL_GROUP card.

Remarks:

1. Refer to an example in the *CONSTRAINED_LAGRANGE_IN_SOLID section.

***SET_NODE_{OPTION}**

Available options include:

<BLANK>

LIST

COLUMN

LIST_GENERATE

GENERAL

The option, LIST_GENERATE, will generate a block of node ID's between a starting nodal ID number and an ending nodal ID number. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a nodal set with some identical or unique attributes.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			
Remark		1	1	1	1			

Cards 2, 3, 4, ... (OPTION=LIST or <BLANK>) (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I

*SET

*SET_NODE

Cards 2, 3, 4, ... (OPTION=COLUMN) (The next “*” card terminates the input.)

Card 2...	1	2	3	4	5	6	7	8
Variable	NID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Remark		2	2	2	2			

Cards 2, 3, 4, ... (OPTION=LIST_GENERATE) (The next “*” card terminates the input.)

Card 2...	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERAL) (The next “*” card terminates the input.) This set is a combination of a series of options: ALL, NODE, DNODE, PART, DPART, BOX, and DBOX.

Card 2...	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SID	Set identification. All node sets should have a unique set ID.
DA1	First nodal attribute default value, see remark 1 below.
DA2	Second nodal attribute default value
DA3	Third nodal attribute default value
DA4	Fourth nodal attribute default value
NIDN	Node ID n

VARIABLE	DESCRIPTION
NID	Nodal ID
A1	First nodal attribute, see remark 2 below.
A2	Second nodal attribute
A3	Third nodal attribute
A4	Fourth nodal attribute
BNBEG	First node ID in block N.
BNEND	Last node ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the node numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not nodal ID's.
OPTION	Option for GENERAL. See table below.
E1,....,E7	Specified entity. Each card must have the option specified. See table below.

OPTION	ENTITY (define up to 7)	FUNCTION
ALL		All nodes will be included in the set.
NODE	n1, n2, n3, n4, n5, n6, n7	Nodes n1, n2, n3, ... will be included.
DNODE	n1, n2, n3, n4, n5, n6, n7	Nodes n1, n2, n3, ... previously added will be excluded.
PART	p1, p2, p3, p4, p5, p6, p7	Nodes of parts p1, p2, p3, ... will be included.
DPART	p1, p2, p3, p4, p5, p6, p7	Nodes of parts p1, p2, p3, ... previously added will be excluded.
BOX	b1, b2, b3, b4, b5, b6, b7	Nodes inside boxes b1, b2, b3, ... will be included.
DBOX	b1, b2, b3, b4, b5, b6, b7	Nodes inside boxes b1, b2, b3, ... previously added will be excluded.

Remarks:

1. Nodal attributes can be assigned for some input types. For example, for contact option, *CONTACT_TIEBREAK_NODES_TO_SURFACE the attributes are:

DA1=NFLF Normal failure force,

DA2=NSFLF Shear failure force,

DA3=NNEN Exponent for normal force,

DA4=NMES Exponent for shear force.
2. The default nodal attributes can be overridden on these cards; otherwise, A1=DA1, etc.

***SET_NODE_ADD_{OPTION}**

Available options include:

<BLANK>

ADVANCED

Purpose: Define a node set by combining node sets or for the ADVANCED option by combining, NODE, SHELL, SOLID, BEAM, SEGMENT, DISCRETE and THICK SHELL sets.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Default	none	none	none	none	none			
Remark								

If the ADVANCED option is inactive:

Card 2, 3, 4, ... (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	NSID1	NSID2	NSID3	NSID4	NSID5	NSID6	NSID7	NSID8
Type	I	I	I	I	I	I	I	I

*SET

*SET_NODE_ADD

If the **ADVANCED** option is active:

Card 2, 3, 4 ... (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	SID1	TYPE1	SID2	TYPE2	SID3	TYPE3	SID4	TYPE4
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

NSID	Set ID of new node set. All node sets should have a unique set ID.
BSID _{<i>n</i>}	The <i>n</i> th node set ID
SID _{<i>n</i>}	The <i>n</i> th set ID
TYPE _{<i>n</i>}	Type set for SID _{<i>n</i>} : 1: - Node set 2: - Shell set 3: - Beam set 4: - Solid set 5: - Segment set 6: - Discrete set 7: - Thick shell set

***SET_PART_{OPTION}**

Available options include:

<BLANK>

LIST

COLUMN

LIST_GENERATE

The last option will generate a block of part ID's between a starting part ID number and an ending part ID number. An arbitrary number of blocks can be specified to define the part set.

Purpose: Define a set of parts with optional attributes. For the column option, see *AIRBAG or *CONSTRAINED_RIGID_BODY_STOPPERS.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.						
Remark		1	1	1	1			

Card 2, 3, 4, ... (OPTION=LIST or <BLANK>) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I

*SET

*SET_PART

Card 2, 3, 4, ... (OPTION=COLUMN) (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	PID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Remark		1	1	1	1			

Cards 2, 3, 4, ... (OPTION=LIST_GENERATE) (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SID	Set ID. All part sets should have a unique set ID.
DA1	First attribute default value, see remark 1 below.
DA2	Second attribute default value
DA3	Third attribute default value
DA4	Fourth attribute default value
PID	Part ID
PID1	First part ID
PID2	Second part ID
.	.
A1	First part attribute, see remark 2 below.
A2	Second part attribute
A3	Third part attribute
A4	Fourth part attribute

VARIABLE	DESCRIPTION
BNBEG	First part ID in block N.
BNEND	Last part ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the part numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not part ID's.

Remarks:

1. Part attributes can be assigned for some input types. For example, for airbags a time delay, DA1=T1, can be defined before pressure begins to act along with a time delay, DA2=T2, before full pressure is applied, (default T2=T1), and for the constraint option, *CONSTRAINED_RIGID_BODY_STOPPERS one attribute can be defined: DA1, the closure distance which activates the stopper constraint.
2. The default part attributes can be overridden on the part cards; otherwise, A1=DA1, etc.

*SET

*SET_PART_ADD

*SET_PART_ADD

Purpose: Define a part set by combining part sets. The attributes, if any, (see *SET_PART above) will be taken from the part sets that are combined.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							
Remark								

Card 2, 3, 4, ... (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	PSID1	PSID2	PSID3	PSID4	PSID5	PSID6	PSID7	PSID8
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SID Set ID. All part sets should have a unique set ID.

PSID n The n th part set ID

***SET_SEGMENT_{OPTION}**

Available options include:

<BLANK>

GENERAL

Purpose: Define a set of quadrilateral and triangular segments with optional identical or unique attributes.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			
Remarks		1	1	1	1			

Cards 2, 3, 4, ... (No option is specified) (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	N1	N2	N3	N4	A1	A2	A3	A4
Type	I	I	I	I	F	F	F	F
Remarks				2	3	3	3	3

*SET

*SET_SEGMENT

Cards 2, 3, 4, ... (OPTION=GENERAL) (The next “*” card terminates the input.) This set is a combination of a series of options listed in the table defined below.

Card 2...	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I or F	I or F	I or F	I or F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID. All segment sets should have a unique set ID.
DA1	First segment attribute default value, see remark 1 below.
DA2	Second segment attribute default value
DA3	Third segment attribute default value
DA4	Fourth segment attribute default value
N1	Nodal point n ₁
N2	Nodal point n ₂
N3	Nodal point n ₃
N4	Nodal point n ₄ , see remark 2 below.
A1	First segment attribute, see remark 3 below.
A2	Second segment attribute
A3	Third segment attribute
A4	Fourth segment attribute
NFLS	Normal failure stress
SFLS	Shear failure stress. Failure criterion:
OPTION	Option for GENERAL. See table below.
E1,...,E7	Specified entity. Each card must have an option specified. See table below.

FORMAT (A10,3I10, 4F10.0)		
OPTION	ENTITIES + ATTRIBUTES	FUNCTION
BOX	b1, b2, b3, a1, a2, a3, a4	Generate segments inside box ID bi, i=1,,2,3. For shell elements one segment per shell is generated. For solid elements only those segments wrapping the solid part and pointing outward from the part will be generated.
BOX_SHELL	b1, b2, b3, a1, a2, a3, a4	Generate segments inside box ID bi, i=1,,2,3. The segments are only generated for shell elements. One segment per shell is generated.
BOX_SLDIO	b1, b2, b3, a1, a2, a3, a4	Generate segments inside box ID bi, i=1,,2,3. Both exterior segments and inter-element segments are generated.
BOX_SOLID	b1, b2, b3, a1, a2, a3, a4	Generate segments inside box ID bi, i=1,,2,3. The segments are only generated for exterior solid elements
PART	p1, p2, p3, a1, a2, a3, a4	Generate segments of parts p1, p2, p3 with attributes a1-a4. For shell elements one segment per shell is generated. For solid elements only those segments wrapping the solid part and pointing outward from the part will be generated.
PART_IO	p1, p2, p3, a1, a2, a3, a4	Generate segments of parts p1, p2, p3 with attributes a1-a4. Same as the PART option above except that inter-element segments inside parts will be generated as well. This option is sometimes useful for single surface contact of solid elements to prevent negative volumes caused by inversion.

FORMAT (A10,7I10)		
DBOX	b1, b2, b3, b4, b5, b6, b7	Segments inside boxes b1, b2, ... previously added will be excluded.
DBOX_SHELL	b1, b2, b3, b4, b5, b6, b7	Shell related segments inside boxes b1, b2, ... previously added will be excluded.
DBOX_SOLID	b1, b2, b3, b4, b5, b6, b7	Solid related segments inside boxes b1, b2, ... previously added will be excluded.
DPART	p1, p2, p3, p4, p5, p6, p7	Segments of parts p1, p2, p3, ... previously added will be excluded.
DSEG	n1, n2, n3, n4	Segments with node ID's n1,n2, n3, and n4 previously added will be deleted. The numbering sequence is irrelevant.
SEG	n1, n2, n3, n4	Create segment with node ID's n1,n2, n3, and n4.t.

Remarks:

1. Segment attributes can be assigned for some input types. For example, for the contact options, the attributes for the SLAVE surface are:

DA1=NFLS Normal failure stress, *CONTACT_TIEBREAK_SURFACE_contact only,

DA2=SFLS Shear failure stress, *CONTACT_TIEBREAK_SURFACE_contact only,

DA3=FSF Coulomb friction scale factor,

DA4=VSF Viscous friction scale factor,

and the attributes for the MASTER surface are:

DA1=FSF Coulomb friction scale factor,

DA2=VSF Viscous friction scale factor.

For airbags, see *AIRBAG, a time delay, DA1=T1, can be defined before pressure begins to act on a segment along with a time delay, DA2=T2, before full pressure is applied to the segment, (default T2=T1), and for the constraint option,

2. To define a triangular segment make n4 equal to n3.
3. The default segment attributes can be overridden on these cards, otherwise, A1=DA1, etc.

***SET_2D_SEGMENT_{OPTION}**

Available options include:

<BLANK>

SET

Purpose: Define a set of boundary line segments in two dimensional axisymmetric, plane stress, and plane strain geometries with optional identical or unique attributes. This option is recommended for thermal problems which involve adaptivity.

	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			
Remarks		1	1	1	1			

	1	2	3	4	5	6	7	8
Variable	PID/PSID							
Type	I							
Remarks	2							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID. All segment sets should have a unique set ID.
DA1	First segment attribute default value, see remark 1 below.
DA2	Second segment attribute default value
DA3	Third segment attribute default value
DA4	Fourth segment attribute default value

***SET**

***SET_2D_SEGMENT**

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID/PSID	Part ID or part set ID if SET option is specified.

Remarks:

1. The boundary along $r = 0$ isn't included in axisymmetric problems.
2. The common boundary between parts isn't included in the boundary segments.

***SET_SHELL_{OPTION}**

Available options include:

<BLANK>

LIST

COLUMN

LIST_GENERATE

GENERAL

The last option will generate a block of shell ID's between a starting shell ID number and an ending ID number. An arbitrary number of blocks can be specified to define the shell set.

Purpose: Define a set of shell elements with optional identical or unique attributes.

Card 1 1 2 3 4 5 6 7 8

Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			
Remarks		1	1	1	1			

Card 2, 3, 4, ... (OPTION=LIST or <BLANK>) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	EID1	EID2	EID3	EID4	EID5	EID6	EID7	EID8
Type	I	I	I	I	I	I	I	I
Remarks	2	2	2	2	2	2	2	2

*SET

*SET_SHELL

Card 2, 3, 4, ... (*OPTION=COLUMN*) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	EID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Remarks		3	3	3	3			

Cards 2, 3, 4, ... (*OPTION=LIST_GENERATE*) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (*OPTION=GENERAL*) (The next "*" card terminates the input.) This set is a combination of a series of options: ALL, ELEM, DELEM, PART, DPART, BOX, and DBOX.

Card 2... 1 2 3 4 5 6 7 8

Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SID	Set ID. All shell sets should have a unique set ID.
DA1	First attribute default value, see remark 1.
DA2	Second attribute default value
DA3	Third attribute default value
DA4	Fourth attribute default value
EID1	First shell element ID, see remark 2.

VARIABLE	DESCRIPTION
EID2	Second shell element ID
•	•
•	•
EID	Element ID
A1	First attribute
A2	Second attribute
A3	Third attribute
A4	Fourth attribute
BNBEG	First shell ID in shell block N.
BNEND	Last shell ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not element ID's.
OPTION	Option for GENERAL. See table below.
E1,....,E7	Specified entity. Each card must have the option specified. See table below.

OPTION	ENTITY (define up to 7)	FUNCTION
ALL		All shell elements will be included in the set.
ELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... will be included.
DELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... previously added will be excluded.
PART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... will be included.
DPART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... previously added will be excluded.
BOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... will be included.
DBOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... previously added will be excluded.

Remarks:

1. Shell attributes can be assigned for some input types. For example, for the contact options, the attributes for the SLAVE surface are:

DA1=NFLS Normal failure stress, *CONTACT_TIEBREAK_SURFACE_contact only,

DA2=SFLS Shear failure stress, *CONTACT_TIEBREAK_SURFACE_contact only,

DA3=FSF Coulomb friction scale factor,

DA4=VSF Viscous friction scale factor,

and the attributes for the MASTER surface are:

DA1=FSF Coulomb friction scale factor,

DA2=VSF Viscous friction scale factor.

2. The default attributes are taken.
3. The default shell attributes can be overridden on these cards; otherwise, A1=DA1, etc.

***SET_SHELL_ADD**

Purpose: Define a shell set by combining shell sets.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							
Remark								

Card 2, 3, 4 ... (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID of new shell set. All shell sets should have a unique set ID.
BSID n	The n th shell set ID

*SET

*SET_SOLID

*SET_SOLID_{OPTION}

Available options include:

<BLANK>

GENERATE

GENERAL

The last option, GENERATE, will generate a block of solid element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of solid elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							

Cards 2, 3, 4, ... (OPTION=none) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERATE) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERAL) (The next “*” card terminates the input.) This set is a combination of a series of options: ALL, ELEM, DELEM, PART, DPART, BOX, and DBOX.

Card 2... 1 2 3 4 5 6 7 8

Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

<u>VARIABLE</u>	<u>DESCRIPTION</u>
SID	Set ID. All solid sets should have a unique set ID.
K1	First element ID
K2	Second element ID
.	.
.	.
K8	Eighth element ID
BNBEG	First solid element ID in block N.
BNEND	Last solid element ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not element ID's.
OPTION	Option for GENERAL. See table below.
E1,....,E7	Specified entity. Each card must have the option specified. See table below.

OPTION	ENTITY (define up to 7)	FUNCTION
ALL		All solid elements will be included in the set.
ELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... will be included.
DELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... previously added will be excluded.
PART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... will be included.
DPART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... previously added will be excluded.
BOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... will be included.
DBOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... previously added will be excluded.

***SET_SOLID_ADD**

Purpose: Define a solid set by combining solid sets.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							
Remark								

Card 2, 3, 4 ... (The next “*” card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

SID Set ID of new solid set. All solid sets should have a unique set ID.

SSID_n The *n*th solid set ID.

*SET

*SET_TSHELL

*SET_TSHELL_{OPTION}

Available options include:

<BLANK>

GENERATE

GENERAL

The last option, GENERATE, will generate a block of thick shell element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of thick shell elements.

Card 1 1 2 3 4 5 6 7 8

Variable	SID							
Type	I							
Default	none							

Cards 2, 3, 4, ... (OPTION=none) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERATE) (The next "*" card terminates the input.)

Card 2... 1 2 3 4 5 6 7 8

Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Cards 2, 3, 4, ... (OPTION=GENERAL) (The next “*” card terminates the input.) This set is a combination of a series of options: ALL, ELEM, DELEM, PART, DPART, BOX, and DBOX.

Card 2... 1 2 3 4 5 6 7 8

Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE**DESCRIPTION**

SID	Set ID. All tshell sets should have a unique set ID.
K1	First thick shell element ID
K2	Second thick shell element ID
.	.
.	.
K8	Eighth thick shell element ID
.	.
.	.
BNBEG	First thick shell element ID in block N.
BNEND	Last thick shell element ID in block N. All defined ID's between and including BNBEG to BNEND are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. BNBEG and BNEND may simply be limits on the ID's and not element ID's.
OPTION	Option for GENERAL. See table below.
E1,....,E7	Specified entity. Each card must have the option specified. See table below.

OPTION	ENTITY (define up to 7)	FUNCTION
ALL		All thick shell elements will be included in the set.
ELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... will be included.
DELEM	e1, e2, e3, e4, e5, e6, e7	Elements e1, e2, e3, ... previously added will be excluded.
PART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... will be included.
DPART	p1, p2, p3, p4, p5, p6, p7	Elements of parts p1, p2, p3, ... previously added will be excluded.
BOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... will be included.
DBOX	b1, b2, b3, b4, b5, b6, b7	Elements inside boxes b1, b2, ... previously added will be excluded.

***TERMINATION**

The keyword ***TERMINATION** provides an alternative way of stopping the calculation before the termination time is reached. The termination time is specified on the ***CONTROL_TERMINATION** input and will terminate the calculation whether or not the options available in this section are active. Different types of termination may be defined:

***TERMINATION_BODY**

***TERMINATION_CONTACT**

***TERMINATION_CURVE**

***TERMINATION_DELETED_SHELLS_OPTION**

***TERMINATION_NODE**

*TERMINATION

*TERMINATION_BODY

*TERMINATION_BODY

Purpose: Terminate calculation based on rigid body displacements. For *TERMINATION_BODY the analysis terminates when the center of mass displacement of the rigid body specified reaches either the maximum or minimum value (stops 1, 2 or 3) or the displacement magnitude of the center of mass is exceeded (stop 4). If more than one condition is input, the analysis stops when any of the conditions is satisfied. Termination by other means than *TERMINATION input is controlled by the *CONTROL_TERMINATION control card. Note that this type of termination is not active during dynamic relaxation.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	STOP	MAXC	MINC				
Type	I	I	F	F				
Default	none	none	-	-				

VARIABLE

DESCRIPTION

PID	Part ID of rigid body, see *PART_OPTION.
STOP	Stop criterion: EQ.1: global x direction, EQ.2: global y direction, EQ.3: global z direction, EQ.4: stop if displacement magnitude is exceeded.
MAXC	Maximum (most positive) displacement, options 1, 2, 3 and 4: EQ.0.0: MAXC set to 1.0e21.
MINC	Minimum (most negative) displacement, options 1, 2 and 3 above only: EQ.0.0: MINC set to -1.0e21.

***TERMINATION_CONTACT**

Purpose: The analysis terminates when the magnitude of the contact interface resultant force is zero. If more than one contact condition is input, the analysis stops when any of the conditions is satisfied. Termination by other means than *TERMINATION input is controlled by the *CONTROL_TERMINATION control card. Note that this type of termination is not active during dynamic relaxation.

Card 1 1 2 3 4 5 6 7 8

Variable	CID	ACTIM	DUR					
Type	I	I	F					
Default	none	none	-	-				

VARIABLE

DESCRIPTION

CID Contact ID. The contact ID is defined by the ordering of the contact input unless the TITLE option which allows the CID to be defined is used in the *CONTACT section.

ACTIM Activation time.

DUR Time duration of null resultant force prior to termination. This time is tracked only after the activation time is reached and the contact resultant forces are zero.

EQ.0.0: Immediate termination after null force is detected.

*TERMINATION

*TERMINATION_CURVE

*TERMINATION_CURVE

Purpose: Terminate the calculation when the load curve value returns to zero. This termination can be used with the contact option *CONTACT_AUTO_MOVE. In this latter option, the load curve is modified to account for the movement of the master surface.

Card 1 1 2 3 4 5 6 7 8

Variable	LCID	ATIME						
Type	I	F						
Default	none	Remark 1		-				

VARIABLE

DESCRIPTION

LCID Load curve ID governing termination.

ATIME Activation time. After this time the load curve is checked. If zero, see remark 1 below.

Remarks:

1. If ATIME=0.0, termination will occur after the load curve value becomes nonzero and then returns to zero.

***TERMINATION_DELETED_SHELLS_{OPTION}**

Available options include:

<BLANK>

SET

Purpose: Terminate the calculation when the number of deleted shells for a specified part ID exceeds the value defined here. This input has no effect for a part ID that is left undefined. Generally, this option should be used with the NFAIL1 and NFAIL4 parameters that are defined in the *CONTROL_SHELL control information.

Card 1 1 2 3 4 5 6 7 8

Variable	PID/PSID	NDS						
Type	I	I						
Default	none	1						

VARIABLE

DESCRIPTION

PID/PSID

Part ID or if option SET is active, part set ID.

NDS

Number of elements that must be deleted for the specified part ID's, before an error termination occurs.

*TERMINATION

*TERMINATION_NODE

*TERMINATION_NODE

Purpose: Terminate calculation based on nodal point coordinates. The analysis terminates for *TERMINATION_NODE when the current position of the node specified reaches either the maximum or minimum value (stops 1, 2 or 3), or picks up force from any contact surface (stops 4). Termination by other means than *TERMINATION is controlled by the *CONTROL_TERMINATION control card. Note that this type of termination is not active during dynamic relaxation.

Card 1 1 2 3 4 5 6 7 8

Variable	NID	STOP	MAXC	MINC				
Type	I	I	F	F				
Default	none	none	-	-				

VARIABLE

DESCRIPTION

NID	Node ID, see *NODE_OPTION.
STOP	Stop criterion: EQ.1: global x direction, EQ.2: global y direction, EQ.3: global z direction, EQ.4: stop if node touches contact surface.
MAXC	Maximum (most positive) coordinate (options 1, 2 and 3) above only.
MINC	Minimum (most negative) coordinate (options 1, 2 and 3) above only.

***TITLE**

***TITLE**

Purpose: Define job title.

Card 1 2 3 4 5 6 7 8

Variable	TITLE						
Type	C						
Default	LS-DYNA USER INPUT						

VARIABLE

DESCRIPTION

TITLE

Heading to appear on output and in output files.

***TITLE**

***USER**

***USER_INTERFACE_OPTION**

Available options include:

CONTROL

FRICTION

Purpose: Define user defined input and allocate storage for user defined subroutines for the contact algorithms. See also *CONTROL_CONTACT. The **CONTROL** option above allows the user to take information from the contact interface for further action, e.g., stopping the analysis. A sample user subroutine is provided in Appendix F.

The **FRICTION** option may be used to modify the Coulomb friction coefficients in contact types 3, 5, or 10 (*CONTACT_SURFACE_TO_SURFACE, *CONTACT_NODES_TO_SURFACE, or *CONTACT_ONE_WAY_SURFACE_TO_SURFACE) according to contact information or to use a friction coefficient database. A sample user-defined friction subroutine is provided in Appendix G. For the subroutine to be called, the static friction coefficient FS on Card 2 of *CONTACT must be any nonzero value, and shell thickness offsets must be invoked in the contact by setting SHLTHK to 1 or 2 using *CONTROL_CONTACT or Opt. Card B in *CONTACT. The array length USRFRC in *CONTROL_CONTACT should be set to a value no less than the sum of the number of history variables NOC and the number of user-defined input parameters in *USER_INTERFACE_FRICTION.

Card 1 1 2 3 4 5 6 7 8

Variable	IFID	NOC	NOCI	NHSV				
Type	I	I	I	I				
Default	none	none	none	O				

(Use as many cards as necessary to define NOCI variables)

Card 2... 1 2 3 4 5 6 7 8

Variable	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
Type	F	F	F	F	F	F	F	F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IFID	Interface number
NOC	Number of history variables for interface. The number should not exceed the length of the array defined on *CONTROL_CONTACT. See Remarks.
NOCI	Initialize the first NOCI history variables in the input. NOCI must be smaller or equal to NOC.
NHSV	Number of history variables per interface node (only for friction interface).
UC1	First user defined input parameter.
UC2	Second user defined input parameter.
.	.
.	.
.	.
UCn	Last user defined input parameter, where n = NOCI.

Remarks:

The (NOC) interface variables (of which NOCI are initialized) are passed as arguments to the user defined subroutine. See Appendix G for the full list of arguments passed to the subroutine.

***USER_LOADING**

Purpose: Provide a means of applying pressure and force boundary conditions. The keyword *USER_LOADING activates this option. Input here is optional with the input being read until the next “*” keyword appears. The data read here is to be stored in a common block provided in the user subroutine, LOADUD. This data is stored and retrieved from the restart files.

(Insert as many cards as needed. The next * card terminates input.)

Card 1... 1 2 3 4 5 6 7 8

Variable	PARAM1	PARAM2	PARAM3	PARAM4	PARAM5	PARAM6	PARAM7	PARAM8
Type	F	F	F	F	F	F	F	F
Default	none							

VARIABLE

DESCRIPTION

PARAMn

This is the nth user input parameter.

RESTART INPUT DATA

In general three categories of restart actions are possible with LS-DYNA and are outlined in the following discussion:

- a) A simple restart occurs when LS-DYNA was interactively stopped before reaching the termination time. Then simply defining the R=rtf file on the execution line for LS-DYNA restarts the calculation from the termination point and the calculation will continue to the specified termination time-see INTRODUCTION, Execution Syntax. No additional input deck is required.

- b) If minor modifications are desired as, e.g.,
 - reset termination time,
 - reset output printing interval,
 - reset output plotting interval,
 - delete contact surfaces,
 - delete elements and parts,
 - switch deformable bodies to rigid,
 - switch rigid bodies to deformable,
 - change damping options.

This type of restart is called a small restart and the corresponding input deck a “small restart input deck.” All modifications to the problem made with the restart input deck will be reflected in subsequent restart dumps. All the members of the file families are consecutively numbered beginning from the last member. The small input deck replaces the standard input deck on the execution line which has at least the following contents:

LS-DYNA I=*restartinput* R=D3DUMP*n*

where *D3DUMPn* (or whatever name is chosen for the family member) is the *n*th restart file from the last run where the data is taken. LS-DYNA automatically detects that a small input deck is used since the I=*restartinput* file may contain the keywords:

***CHANGE_OPTION**

***CONTROL_DYNAMIC_RELAXATION**

***CONTROL_SHELL**

***RESTART**

***CONTROL_TERMINATION**

***CONTROL_TIMESTEP**

***DAMPING_GLOBAL**

***DATABASE_OPTION**

***DATABASE_BINARY_OPTION**

***DELETE_OPTION**

***INTERFACE_SPRINGBACK**

***RIGID_DEFORMABLE_OPTION**

***STRESS_INITIALIZATION_{OPTION}**

***TERMINATION_OPTION**

***TITLE**

***KEYWORD** (see INTRODUCTION, Execution Syntax)

***CONTROL_CPU**

***DEFINE_OPTION**

***SET_OPTION**

i.e., the keyword ***STRESS_INITIALIZATION** may not be used in the small restart. The user has to take care that nonphysical modifications to the input deck are avoided; otherwise, complete nonsense may be the result.

- c) If many modifications are desired a so-called full restart may be the appropriate choice. Then the keyword ***STRESS_INITIALIZATION** has to be provided in the input. As also outlined in the INTRODUCTION, Restart Analysis, either all parts can be initialized with the restart data or some selection of parts can be made for the stress initialization. See ***STRESS_INITIALIZATION**. In a full deck restart, deleted elements in this section will be deleted in the full deck automatically even though they are defined. Likewise, if it is necessary to change the velocity field, that must also be performed in this section using the **CHANGE_VELOCITY_....** options. The velocity field in the full deck part of the input is ignored.

***CHANGE_OPTION**

Purpose: Change solution options.

Available options include:

BOUNDARY_CONDITION

CONTACT_SMALL_PENETRATION

CURVE_DEFINITION

RIGID_BODY_CONSTRAINT

RIGID_BODY_INERTIA

RIGID_BODY_STOPPER

STATUS_REPORT_FREQUENCY

THERMAL_PARAMETERS

VELOCITY

VELOCITY_NODE

VELOCITY_RIGID_BODY

VELOCITY_ZERO

*RESTART

*CHANGE

For **BOUNDARY_CONDITION** option define an arbitrary number of cards giving the nodal ID and the additional translational displacement boundary condition code. Previous boundary condition codes will continue to be imposed, i.e., a fixed node cannot be freed with this option. This input terminates when the next “*” card is encountered.

Card 1...	1	2	3	4	5	6	7	8
Variable	NID	BCC						
Type	I	I						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NID	Nodal point ID, see also *NODE.
BCC	New translational boundary condition code: EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.

CHANGE**RESTART**

For **CONTACT_SMALL_PENETRATION** option define an arbitrary number of cards giving a list of contact surface ID numbers where the small penetration check is to be turned on. This input terminates when the next “*” card is encountered. See the PENCHK variable on the *CONTACT definition.

Card 1... 1 2 3 4 5 6 7 8

Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I

VARIABLE**DESCRIPTION***ID_n*Contact ID for surface number *n*.

*RESTART

*CHANGE

The **CURVE_DEFINITION** option allows a load curve to be redefined. *The new load curve must contain the same number of points as the curve it replaces.* The curve should be defined in the **DEFINE_CURVE** section of this manual. This input terminates when the next “*” card is encountered. Any offsets and scale factors are ignored.

Card 1 2 3 4 5 6 7 8

Variable	LCID							
Type	I							

VARIABLE

DESCRIPTION

LCID

Load curve ID

The **RIGID_BODY_CONSTRAINT** option allows translational and rotational boundary conditions on a rigid body to be changed. This input terminates when the next “*” card is encountered. Also, see *CONSTRAINED_RIGID_BODIES.

Card	1	2	3	4	5	6	7	8
Variable	PID	TC	RC					
Type	I	I	I					

VARIABLE**DESCRIPTION**

PID	Part ID, see *PART.
TC	Translational constraint: EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.
RC	Rotational constraint: EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

*RESTART

*CHANGE

The **RIGID_BODY_INERTIA** option allows the mass and inertia properties of a rigid body to be changed. This input terminates when the next "*" card is encountered. The inertia tensor is defined in the local system defined in *MAT_RIGID at the start of the calculation. This coordinate system, which is fixed in the rigid body, tracks the rigid body rotation.

Card 1 2 3 4 5 6 7 8

Variable	ID	PID	TM					
Type	I	I	F					

Card 2

Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ID	ID for this change inertia input.
PID	Part ID, see *PART.
TM	Translational mass.
IXX	I_{xx} , xx component of inertia tensor.
IXY	I_{xy}
IXZ	I_{xz}
IYY	I_{yy} , yy component of inertia tensor.
IYZ	I_{yz}
IZZ	I_{zz} , zz component of inertia tensor.

CHANGE**RESTART**

The **RIGID_BODY_STOPPER** option allows existing stoppers to be redefined. This input terminates when the next “*” card is encountered. See *CONSTRAINED_RIGID_BODY_STOPPERS.

New stopper definitions cannot be introduced in this section. Existing stoppers can be modified.

Card 1 1 2 3 4 5 6 7 8

Variable	PID	LCMAX	LCMIN	PSIDMX	PSIDMN	LCVMNX	DIR	VID
Type	I	I	I	I	I	I	I	I
Default	required	0	0	0	0	0	required	0

Card 2

Variable	BIRTH	DEATH						
Type	F	F						
Default	0	10 ²⁸						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID of master rigid body, see *PART.
LCMAX	Load curve ID defining the maximum coordinate as a function of time: EQ.0: no limitation of the maximum displacement. New curves can be defined by the *DEFINE_CURVE within the present restart deck.
LCMIN	Load curve ID defining the minimum coordinate as a function of time: EQ.0: no limitation of the minimum displacement. New curves can be defined by the *DEFINE_CURVE within the present restart deck.
PSIDMX	Optional part set ID of rigid bodies that are slaved in the maximum coordinate direction to the master rigid body. This option requires additional input by the *SET_PART definition.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PSIDMN	Optional part set ID of rigid bodies that are slaved in the minimum coordinate direction to the master rigid body. This option requires additional input by the *SET_PART definition.
LCVMNX	Load curve ID which defines the maximum absolute value of the velocity that is allowed within the stopper: EQ.0: no limitation of the minimum displacement.
DIR	Direction stopper acts in: EQ.1: x-translation, EQ.2: y-translation, EQ.3: z-translation, EQ.4: arbitrary, defined by vector VID, EQ.5: x-axis rotation, EQ.6: y-axis rotation, EQ.7: z-axis rotation, EQ.8: arbitrary, defined by vector VID.
VID	Vector for arbitrary orientation of stopper. The vector must be defined by a *DEFINE_VECTOR within the present restart deck.
BIRTH	Time at which stopper is activated.
DEATH	Time at which stopper is deactivated.

Remarks:

The optional definition of part sets in minimum or maximum coordinate directions allows the motion to be controlled in an arbitrary direction.

***CHANGE**

***RESTART**

The **STATUS_REPORT_FREQUENCY** option allows the output status interval to be changed.

Card 1 2 3 4 5 6 7 8

Variable	IKEDIT							
Type	I							

VARIABLE

DESCRIPTION

IKEDIT

Problem status report interval steps in the D3HSP output file:
EQ.0: interval remains unchanged.

*RESTART

*CHANGE

The **THERMAL_PARAMETERS** option allows parameters used by a thermal or coupled structural/thermal analysis to be changed. These parameters were initially defined on the *CONTROL_THERMAL cards. Two cards are defined for this option.

Card 1 1 2 3 4 5 6 7 8

Variable	TS	DT	TMIN	TMAX	DTEMP	TSCP		
Type	I	F	F	F	F	F		

Card 2

Variable	REFMAX	TOL						
Type	I	F						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TS	Thermal time step code: EQ.0: No change, EQ.1: Fixed time step, EQ.2: variable time step.
DT	Thermal time step on restart: EQ.0: No change.
TMIN	Minimum thermal time step: EQ.0: No change.
TMAX	Maximum thermal time step: EQ.0: No change.
DTEMP	Maximum temperature change in a thermal time step: EQ.0: No change.
TSCP	Time step control parameter (0.0 < TSCP < 1.0): EQ.0: No change.
REFMAX	Maximum number of reformations per thermal time step: EQ.0: No change.
TOL	Non-linear convergence tolerance: EQ.0: No change.

CHANGE**RESTART**

The **VELOCITY_NODE** and the **VELOCITY_NODE_ONLY** options allow the velocity of nodal points to be changed at restart. Termination of this input is when the next “*” card is read. Undefined nodes will have their nodal velocities reset to zero if a ***CHANGE_VELOCITY_NODE** definition is encountered in the restart deck. However, if any of the ***CHANGE_VELOCITY** or **CHANGE_VELOCITY_NODE** definitions have **_ONLY** appended, then only the specified nodes will have their nodal velocities modified.

Card 1 2 3 4 5 6 7 8

Variable	NID	VX	VY	VZ	VXR	VYR	VZR	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

VARIABLE**DESCRIPTION**

NID	Node ID
VX	Translational velocity in x-direction.
VY	Translational velocity in y-direction.
VZ	Translational velocity in z-direction.
VXR	Rotational velocity about the x-axis.
VYR	Rotational velocity about the y-axis.
VZR	Rotational velocity about the z-axis.

Remarks:

1. If a node is initialized on more than one input card set, then the last set input will determine its velocity, unless it is specified on a ***CHANGE_VELOCITY_NODE** card.
2. If both ***CHANGE_VELOCITY** and ***CHANGE_VELOCITY_ZERO** cards are defined then all velocities will be reset to zero.

*RESTART

*CHANGE

The **VELOCITY** and **VELOCITY_ONLY** options allow a new velocity field to be imposed at restart. Termination of this input is when the next “*” card is read. Undefined nodes will have their nodal velocities reset to zero if a *CHANGE_VELOCITY definition is encountered in the restart deck. However, if any of the *CHANGE_VELOCITY definitions have _ONLY appended, then only the specified nodes will have their nodal velocities modified.

Card 1 1 2 3 4 5 6 7 8

Variable	NSID							
Type	I							
Default	none							
Remark	1							

Card 2

Variable	VX	VY	VZ	VXR	VYR	VZR		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE

DESCRIPTION

- NSID Nodal set ID containing nodes for initial velocity.
- VX Velocity in x-direction.
- VY Velocity in y-direction.
- VZ Velocity in z-direction.
- VXR Rotational velocity about the x-axis.
- VYR Rotational velocity about the y-axis.
- VZR Rotational velocity about the z-axis.

Remarks:

1. If a node is initialized on more than one input card set, then the last set input will determine its velocity, unless it is specified on a *CHANGE_VELOCITY_NODE card.
2. Undefined nodes will have their nodal velocities set to zero if a *CHANGE_VELOCITY definition is encountered in the restart deck.
3. If both *CHANGE_VELOCITY and *CHANGE_VELOCITY_ZERO cards are defined then all velocities will be reset to zero.

*RESTART

*CHANGE

The **VELOCITY_RIGID_BODY** option allows the velocity components of a rigid body to be changed at restart. Termination of this input is when the next “*” card is read.

Card	1	2	3	4	5	6	7	8
Variable	PID	VX	VY	VZ	VXR	VYR	VZR	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID of rigid body.
VX	Translational velocity in x-direction.
VY	Translational velocity in y-direction.
VZ	Translational velocity in z-direction.
VXR	Rotational velocity about the x-axis.
VYR	Rotational velocity about the y-axis.
VZR	Rotational velocity about the z-axis.

Remarks:

1. Rotational velocities are defined about the center of mass of the rigid body.
2. Rigid bodies not defined in this section will not have their velocities modified.

The **VELOCITY_ZERO** option resets the velocities to zero at the start of the restart. Only the *CHANGE_VELOCITY_ZERO card is required for this option without any further input.

***CONTROL_DYNAMIC_RELAXATION**

Purpose: Define controls for dynamic relaxation.

Card	1	2	3	4	5	6	7	8
Variable	NRCYCK	DRTOL	DRFCTR	DRTERM	TSSFDR	IRELAL	EDTTL	IDRFLG
Type	I	F	F	F	F	I	F	I
Default	250	0.001	0.995	infinity	TSSFAC	0	0.0	0
Remarks	1	1	1	1	1			1

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NRCYCK	Number of iterations between convergence checks, for dynamic relaxation option (default = 250).
DRTOL	Convergence tolerance for dynamic relaxation option (default = 0.001).
DRFCTR	Dynamic relaxation factor (default = .995).
DRTERM	Optional termination time for dynamic relaxation. Termination occurs at this time or when convergence is attained (default = infinity).
TSSFDR	Scale factor for computed time step during dynamic relaxation. If zero, the value is set to TSSFAC defined on *CONTROL_TERMINATION. After converging, the scale factor is reset to TSSFAC.
IRELAL	Automatic control for dynamic relaxation option based on algorithm of Papadrakakis [1981].
EDTTL	Convergence tolerance on automatic control of dynamic relaxation.
IDRFLG	Dynamic relaxation flag for stress initialization: EQ.0: not active, EQ.1: dynamic relaxation is activated.

Remarks:

1. If a dynamic relaxation analysis is being restarted at a point before convergence was obtained, then NRCYCK, DRTOL, DRFCTR, DRTERM and TSSFDR will default to their previous values, and IDRFLG will be set to 1.
2. If dynamic relaxation is activated after a restart from a normal transient analysis LS-DYNA continues the output of data as it would without the dynamic relaxation being active. This is unlike the dynamic relaxation phase at the beginning of the calculation when a separate database is not used. Only load curves that are flagged for dynamic relaxation are applied after restarting.

***CONTROL_SHELL**

Purpose: Change failure parameters NFAIL1 and NFAIL2 if necessary. These parameters must be nonzero in the initial run.

Card 1 1 2 3 4 5 6 7 8

Variable									
Type									

Card 2

Variable						NFAIL1	NFAIL4	
Type						I	I	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
-----------------	--------------------

NFAIL1	<p>Flag to check for highly distorted under-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is not needed for one point elements that do not use the warping stiffness. A distorted element is one where a negative jacobian exists within the domain of the shell, not just at integration points. The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs. This test will increase CPU requirements for one point elements.</p>
--------	--

EQ.1: print message and delete element.
EQ.2: print message, write D3DUMP file, and terminate
GT.2: print message and delete element. When NFAIL1 elements are deleted then write D3DUMP file and terminate. These NFAIL1 failed elements also include all shell elements that failed for other reasons than distortion. Before the D3DUMP file is written, NFAIL1 is doubled, so the run can immediately be continued if desired.

NFAIL4	<p>Flag to check for highly distorted fully-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is recommended. A distorted element is one where a negative jacobian exists within the domain of the shell, not just at integration points.</p>
--------	---

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs.</p> <p>EQ.1: print message and delete element. EQ.2: print message, write D3DUMP file, and terminate GT.2: print message and delete element. When NFAIL4 elements are deleted then write D3DUMP file and terminate. These NFAIL4 failed elements also include all shell elements that failed for other reasons than distortion. Before the D3DUMP file is written, NFAIL4 is doubled, so the run can immediately be continued if desired.</p>

***CONTROL_TERMINATION**

***RESTART**

***CONTROL_TERMINATION**

Purpose: Stop the job.

Card 1 2 3 4 5 6 7 8

Variable	ENDTIM	ENDCYC						
Type	F	I						

VARIABLE

DESCRIPTION

ENDTIM

Termination time:
EQ.0.0: Termination time remains unchanged.

ENDCYC

Termination cycle. The termination cycle is optional and will be used if the specified cycle is reached before the termination time.
EQ.0.0: Termination cycle remains unchanged.

This is a reduced version of the *CONTROL_TERMINATION card used in the initial input deck.

*RESTART

*CONTROL_TIMESTEP

*CONTROL_TIMESTEP

Purpose: Set time step size control using different options.

Card	1	2	3	4	5	6	7	8
Variable	DUMMY	TSSFAC	ISDO	DUMMY	DT2MS	LCTM		
Type	F	F	I	F	F	I		

<u>VARIABLE</u>	<u>DESCRIPTION</u>
DUMMY	Dummy field, see remark 1 below.
TSSFAC	Scale factor for computed time step. EQ.0.0: TSSFAC remains unchanged.
ISDO	Basis of time size calculation for 4-node shell elements, ISDO 3-node shells use the shortest altitude for options 0,1 and the shortest side for option 2. This option has no relevance to solid elements, which use a length based on the element volume divided by the largest surface area: EQ.0: characteristic length=area/(longest side), EQ.1: characteristic length=area/(longest diagonal), EQ.2: based on bar wave speed and MAX [shortest side, area/longest side]. THIS LAST OPTION CAN GIVE A MUCH LARGER TIME STEP SIZE THAT CAN LEAD TO INSTABILITIES IN SOME APPLICATIONS, ESPECIALLY WHEN TRIANGULAR ELEMENTS ARE USED.
DUMMY	Dummy field, see remark 1 below.
DT2MS	New time step for mass scaled calculations. Mass scaling must be active in the time zero analysis. EQ.0.0: DT2MS remains unchanged.
LCTM	Load curve ID that limits maximum time step size: EQ.0: LCTM remains unchanged.

Remarks:

1. This a reduced version of the *CONTROL_TIMESTEP used in the initial analysis. The dummy fields are included to maintain compatibility. If using free format input then a 0.0 should be entered for the dummy values.

***DAMPING_GLOBAL**

Purpose: Define mass weighted nodal damping that applies globally to the deformable nodes.

Card 1 2 3 4 5 6 7 8

Variable	LCID	VALDMP						
Type	I	F						
Default	0	0.0						

VARIABLE	DESCRIPTION
LCID	Load curve ID which specifies node system damping: EQ.n: system damping is given by load curve n. The damping force applied to each node is $f = -d(t)mv$, where $d(t)$ is defined by load curve n.
VALDMP	System damping constant, d (this option is bypassed if the load curve number defined above is nonzero).

***DATABASE_OPTION**

Options for ASCII files include. If a file is not specified in the restart deck then the output interval for the file will remain unchanged.

SECFORC	Cross section forces.
RWFORC	Wall forces.
NODOUT	Nodal point data.
ELOUT	Element data.
GLSTAT	Global data.
DEFORC	Discrete elements.
MATSUM	Material energies.
NCFORC	Nodal interface forces.
RCFORC	Resultant interface forces.
DEFGEO	Deformed geometry file
SPCFORC	Set dt for spc reaction forces.
SWFORC	Nodal constraint reaction forces (spot welds and rivets).
ABSTAT	Set dt for airbag statistics.
NODFOR	Set dt for nodal force groups.
BNDOUT	Boundary condition forces and energy
RBDOUT	Set dt for rigid body data.
GCEOUT	Set dt for geometric contact entities.
SLEOUT	Set dt for sliding interface energy.
JNTFORC	Set dt for joint force file.
SBTOUT	Set dt for seat belt output file.
AVSFLT	Set dt for AVS database.
MOVIE	Set dt for MOVIE.
MPGS	Set dt for MPGS.
TPRINT	Set dt for thermal file.

Card 1 2 3 4 5 6 7 8

Variable	DT							
Type	F							

VARIABLE

DESCRIPTION

DT Time interval between outputs:
EQ.0.0: output interval is unchanged.

To terminate output to a particular file set DT to a high value.

If IACCOP=2 was specified in *CONTROL_OUTPUT, the best results are obtained in the NODOUT file by keeping the same DT on restart. When DT is changed for NODOUT, oscillations may occur around the restart time. If DT is larger than initially specified in the original input file, more memory is required to store the time states for the averaging than was originally allocated. A warning message is printed, and the filtering is applied using the available memory. When DT is smaller than initially specified, more oscillations may appear in the output than earlier in the calculation because the frequency content of the averaged output increases as DT decreases.

*RESTART

*DATABASE_BINARY

*DATABASE_BINARY_OPTION

Options for binary output files with the default names given include:

- D3PLOT** Dt for complete output states.
- D3THDT** Dt for time history data for element subsets.
- D3DUMP** Binary output restart files. Define output frequency in cycles
- RUNRSF** Binary output restart file. Define output frequency in cycles.
- INTFOR** Dt for contact surface Interface database.

Card	1	2	3	4	5	6	7	8
Variable	DT/CYCL							
Type	F							

VARIABLE

DESCRIPTION

- DT Time interval between outputs.
 EQ.0.0: Time interval remains unchanged.
- CYCL Output interval in time steps.
 EQ.0.0: output interval remains unchanged.

***DELETE_OPTION**

Available options are:

CONTACT

CONTACT_2DAUTO

ENTITY

PART

ELEMENT_BEAM

ELEMENT_SHELL

ELEMENT_SOLID

ELEMENT_TSHELL

FSI

Purpose: Delete contact surfaces, ALE FSI couplings, parts, or elements by a list of IDs. There are two contact algorithms for two-dimensional problems: the line-to-line contact and the automatic contact defined by part ID's. Each uses their own numbering.

For **CONTACT**, **CONTACT_2DAUTO**, **ENTITY**, **FSI**, or **PART** option.

Card 1 2 3 4 5 6 7 8

Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I

VARIABLE

DESCRIPTION

IDI

Contact ID/Coupling ID/Part ID

For ***DELETE_CONTACT**/***DELETE_FSI** a negative ID implies that the absolute value gives the contact surface/FSI coupling which is to be activated.

***RESTART**

***DELETE**

For the four **ELEMENT** options. Termination of input is when the next “*” card is read.

Card	1	2	3	4	5	6	7	8
Variable	ESID							
Type	I							

VARIABLE

DESCRIPTION

ESID

Element set ID, see *SET_SOLID, *SET_BEAM, *SET_SHELL,
*SET_TSHELL.

***INTERFACE_SPRINGBACK**

Purpose: Define a material subset for an implicit springback calculation in LS-NIKE3D and any nodal constraints to eliminate rigid body degrees-of-freedom. Generally, only the materials that make up the original blank are included in the springback calculation. After termination of the LS-DYNA3D computation, an input deck for LS-NIKE3D and a stress initialization file for LS-NIKE3D are written.

Card 1 2 3 4 5 6 7 8

Variable	PSID							
Type	I							

VARIABLE

DESCRIPTION

PSID Part set ID for springback, see *SET_PART.

Define a list of nodal points that are constrained for the springback. This section is terminated by an "*" indicating the next input section.

Card 1 2 3 4 5 6 7 8

Variable	NID	TC	RC					
Type	I	F	F					
Default	none	0.	0.					

VARIABLE

DESCRIPTION

NID Node ID

TC Translational constraint:
EQ.0: no constraints,
EQ.1: constrained x displacement,
EQ.2: constrained y displacement,
EQ.3: constrained z displacement,
EQ.4: constrained x and y displacements,

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.
RC	Rotational constraint: EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

***RIGID_DEFORMABLE_OPTION**

Available options include:

CONTROL

D2R (Deformable to rigid part switch)

R2D (Rigid to deformable part switch)

Purpose: Define parts to be switched from rigid to deformable and deformable to rigid in a restart. It is only possible to switch parts on a restart if part switching was activated in the time zero analysis. See *DEFORMABLE_TO_RIGID for details of part switching.

*RESTART

*RIGID_DEFORMABLE_CONTROL

*RIGID_DEFORMABLE_CONTROL

Card 1 2 3 4 5 6 7 8

Variable	NRBF	NCSF	RWF	DTMAX				
Type	I	I	I	F				
Default	0	0	0	none				

VARIABLE

DESCRIPTION

NRBF	Flag to delete or activate nodal rigid bodies. If nodal rigid bodies or generalized, weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities: EQ.0: no change, EQ.1: delete, EQ.2: activate.
NCSF	Flag to delete or activate nodal constraint set. If nodal constraint/spot weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities: EQ.0: no change, EQ.1: delete, EQ.2: activate.
RWF	Flag to delete or activate rigid walls: EQ.0: no change, EQ.1: delete, EQ.2: activate.
DTMAX	Maximum permitted time step size after restart.

***RIGID_DEFORMABLE_D2R**

Termination of this input is when the next “*” card is read.

Card 1 2 3 4 5 6 7 8

Variable	PID	MRB						
Type	I	I						
Default	none	0						

VARIABLE

DESCRIPTION

PID

Part ID of the part which is switched to a rigid material.

MRB

Part ID of the master rigid body to which the part is merged. If zero, the part becomes either an independent or master rigid body.

***RESTART**

***RIGID_DEFORMABLE_R2D**

***RIGID_DEFORMABLE_R2D**

Termination of this input is when the next “*” card is read.

Card 1 2 3 4 5 6 7 8

Variable	PID							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

PID

Part ID of the part which is switched to a deformable material.

***STRESS_INITIALIZATION_{OPTION}**

This keyword allows a full deck restart to be performed in LS-DYNA. For a full deck restart a complete input deck has to be included in the restart deck. The stress initialization feature allows all or selected parts to be initialized on restart, using data from the d3dump or runrsf database.

The options that are available with this keyword are:

<BLANK>

DISCRETE

SEATBELT

***STRESS_INITIALIZATION**

If this card is specified without further input as described below then all parts in the new input deck that existed in the previous input deck (with or without the same part IDs) are initialized from the d3dump or runrsf database. Further all seatbelt and discrete parts are initialized.

If only a subset of parts is to be initialized in the new analysis then define as many of the following cards as necessary. Termination of this input is when the next “*” card is read.

Card 1... 1 2 3 4 5 6 7 8

Variable	PIDO	PIDN						
Type	I	I						
Default	none	PIDO						

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PIDO	Old part ID, see *PART.
PIDN	New part ID, see *PART: EQ.0: New part ID is the same as the old part ID.

Remarks:

If one or more of the above cards are defined then discrete and seatbelt elements will not be initialized unless the additional option cards *STRESS_INITIALIZATION_DISCRETE and *STRESS_INITIALIZATION_SEATBELT are defined.

***RESTART**

***STRESS_INITIALIZATION**

***STRESS_INITIALIZATION_DISCRETE**

Initialize all discrete parts from the old parts. No further input is required with this card. This card is not required if *STRESS_INITIALIZATION is specified without further input.

***STRESS_INITIALIZATION_SEATBELT**

Initialize all seatbelt parts from the old parts. No further input is required with this card. This card is not required if *STRESS_INITIALIZATION is specified without further input.

***TERMINATION_OPTION**

Purpose: Stops the job depending on some displacement conditions.

Available options include:

NODE

BODY

Caution: The inputs are different for the nodal and rigid body stop conditions. The nodal stop condition works on the global coordinate position, while the body stop condition works on the relative global translation. The number of termination conditions cannot exceed the maximum of 10 or the number specified in the original analysis.

The analysis terminates for *TERMINATION_NODE when the current position of the node specified reaches either the maximum or minimum value (stops 1, 2 or 3), or picks up force from any contact surface (stop 4). For *TERMINATION_BODY the analysis terminates when the center of mass displacement of the rigid body specified reaches either the maximum or minimum value (stops 1, 2 or 3) or the displacement magnitude of the center of mass is exceeded (stop 4). If more than one condition is input, the analysis stops when any of the conditions is satisfied. *This input completely overrides the existing termination conditions defined in the time zero run.*

Termination by other means is controlled by the *CONTROL_TERMINATION control card.

For both options, the input is identical:

Card 1 2 3 4 5 6 7 8

Variable	NID/PID	STOP	MAXC	MINC				
Type	I	I	F	F				
Default	none	none	-	-				

*RESTART

*TERMINATION

For the **NODE** option:

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NID	Node ID
STOP	Stop criterion: EQ.1: global x direction, EQ.2: global y direction, EQ.3: global z direction, EQ.4: stop if node touches contact surface.
MAXC	Maximum (most positive) coordinate, options 1, 2 and 3 above only.
MINC	Minimum (most negative) coordinate, options 1, 2 and 3 above only.

For the **BODY** option:

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PID	Part ID of rigid body
STOP	Stop criterion: EQ.1: global x direction, EQ.2: global y direction, EQ.3: global z direction, EQ.4: stop if displacement magnitude is exceeded.
MAXC	Maximum (most positive) displacement, options 1, 2, 3 and 4: EQ.0.0: MAXC set to 1.0e21
MINC	Minimum (most negative) displacement, options 1, 2 and 3 above only: EQ.0.0: MINC set to -1.0e21

***TITLE**

***RESTART**

***TITLE**

Purpose: Define job title.

Card 1 2 3 4 5 6 7 8

Variable	TITLE							
Type	C							
Default	LS-DYNA USER INPUT							

VARIABLE

DESCRIPTION

TITLE

Heading to appear on output.

***RESTART**

***TITLE**

REFERENCES

- Abbo, A.J., and S.W. Sloan, "A Smooth Hyperbolic Approximation to the Mohr-Coulomb Yield Criterion," *Computers and Structures*, Vol. 54, No. 1, (1995).
- Allen, D.J., Rule, W.K., Jones, S.E., "Optimizing Material Strength Constants Numerically Extracted from Taylor Impact Data", *Experimental Mechanics*, Vol. 37, No 3, September (1997).
- Allman, D.J., "A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Analysis," *Computers and Structures*, 19, 1-8, (1984).
- Anagnone, A.U. and J.T. Wang, "A Semi-Empirical Method for Estimating the Effective Leak and Vent Areas of an Airbag", AMD-Vol. 237/BED-Vol. 45, pp. 195-217, (1999).
- Aretz, H. "Applications of a New Plane Stress Yield Function to Orthotropic Steel and Aluminum Sheet Metals," *Modeling and Simulation in Materials Science and Engineering*, 12, 491-509 (2004).
- Argon, AS., "A theory for the low-temperature plastic deformation of glassy polymers", *Philosophical Magazine*, 28, 839-865 (1973).
- Armstrong, P.J., and Frederick, C.O., "A Mathematical Representation of the Multiaxial Bauschinger Effect," CEGB Report, RD/B/N731, Berkeley Nuclear Laboratories (1966).
- Arruda, E. and M. Boyce, "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials," *Journal of the Mechanics and Physics of Solids*, Vol. 41, No. 2, pp. 389-412, (1993).
- Auricchio, F., R.L. Taylor and J. Lubliner, "Shape-memory alloys: macromodeling and numerical simulations of the superelastic behavior", *Computer Methods in Applied Mechanics and Engineering*, vol. 146, pp. 281-312, (1997).
- Auricchio, F. and R.L. Taylor, "Shape-memory alloys: modeling and numerical simulations of the finite-strain superelastic behavior", *Computer Methods in Applied Mechanics and Engineering*, vol. 143, pp. 175-194, (1997).
- Bahler AS: The series elastic element of mammalian skeletal muscle. *Am J Physiol* 213:1560-1564, (1967).
- Baker, E.L., "An Explosives Products Thermodynamic Equation of State Appropriate for Material Acceleration and Overdriven Detonation: Theoretical Background and Formulation," Technical Report ARAED-TR-911013, U.S. Army Armament Research, Development and Engineering Center, Picatinney Arsenal, New Jersey, 1991).

REFERENCES

- Baker, E.L. and J. Orosz, J., "Advanced Warheads Concepts: An Advanced Equation of State for Overdriven Detonation," Technical Report ARAED-TR-911007, U.S. Army Armament Research, Development and Engineering Center, Picatinney Arsenal, New Jersey, (1991).
- Baker, E.L. and L.I. Stiel, "Improved Quantitative Explosive Performance Prediction Using Jaguar," 1997 Insensitive Munitions and Energetic Materials Technology Symposium, Tampa, FL, (1997).
- Bammann, D.J. and E.C. Aifantis, "A Model for Finite-Deformation Plasticity," *Acta Mechanica*, 70, 1-13 (1987).
- Bammann, D.J. and G. Johnson, "On the Kinematics of Finite-Deformation Plasticity," *Acta Mechanica*, 69, 97-117 (1987).
- Bammann, D.J., "Modeling the Temperature and Strain Rate Dependent Large Deformation of Metals," Proceedings of the 11th US National Congress of Applied Mechanics, Tuscon, AZ, (1989).
- Bammann, D.J., M.L. Chiesa, A. McDonald, W.A. Kawahara, J.J. Dike, and V.D. Revelli, "Predictions of Ductile Failure in Metal Structures," in AMD-Vol. 107, Failure Criteria and Analysis in Dynamic Response, Edited by. H.E. Lindberg, 7-12, (1990).
- Bandak, F.A., private communications, U.S. Dept. of Trans., Division of Biomechanics Research, 400 7th St., S.W. Washington, D.C. 20590 (1991).
- Barlat, F. and J. Lian, "Plastic Behavior and Stretchability of Sheet Metals. Part I: A Yield Function for Orthotropic Sheets Under Plane Stress Conditions," *Int. J. of Plasticity*, Vol. 5, pp. 51-66 (1989).
- Barlat, F., D.J. Lege, and J.C. Brem, "A Six-Component Yield Function for Anisotropic Materials," *Int. J. of Plasticity*, 7, 693-712, (1991).
- Barlat, F., Y. Maeda, K. Chung, M. Yanagawa, J.C. Brem, Y. Hayashida, D.J. Lege, K. Matsui, S.J. Murtha, S. Hattori, R.C. Becker, and S. Makosey, "Yield Function Development for Aluminum Alloy Sheets", *J. Mech. Phys. Solids*, Vol. 45, No. 11-12, 1727-1763, (1997).
- Barlat, F., Brem, J.C., Yoon, J.W., Chung, K., Dick, R.E., Lege, D.J., Pourboghrat, F., Choi, S.H., Chu, E., "Plane Stress Yield Function for Aluminum Alloy Sheets – Part 1: Theory, *Int. J. Plast.* 19, 1-23, (2003).
- Bathe, K.-J. and Dvorkin, E.N. A four node plate bending element based on Mindlin-Reissner plate theory and a mixed interpolation, *Int. J. Num. Meth. Eng.*, 21, 367-383 (1985).
- Batoz, J.L. and Ben Tahar, M. Evaluation of a new quadrilateral thin plate bending element, *Int. J. Num. Meth. Eng.*, 18, 1644-1677 (1982).
- Batoz, J.-L. and M. Ben Tahar, Evaluation of a new quadrilateral thin plate bending element, *International Journal for Numerical Methods in Engineering*, 18, (1982), 1655-1677.

REFERENCES

- Bazeley, G.P., W.K. Cheung, R.M. Irons, and O.C. Zienkiewicz, "Triangular Elements in Plate Bending-Confirming and Nonconforming Solutions in Matrix Methods and Structural Mechanics," Proc. Conf. on Matrix Methods in Structural Analysis, Rept. AFFDL-R-66-80, Wright Patterson AFB, 547-576 (1965).
- Belytschko, T. and Bindeman, L. P. "Assumed Strain Stabilization of the Eight Node Hexahedral Element," *Comp. Meth. Appl. Mech. Eng.* **105**, 225-260 (1993).
- Belytschko, T.B. and A.H. Marchertas, "Nonlinear Finite Element Method for Plates and its Application to the Dynamic Response of Reactor Fuel Subassemblies," *Trans, ASME J. Pressure Vessel Tech.*, 251-257 (1974).
- Belytschko, T.B. and C.S. Tsay, "Explicit Algorithms for Nonlinear Dynamics of Shells," AMD-Vol.48, ASME, 209-231 (1981).
- Belytschko, T.B. and C.S. Tsay, "Explicit Algorithms for Nonlinear Dynamics of Shells," *Comp. Meth. Appl. Mech. Eng.*, 43, 251-276, (1984).
- Belytschko, T.B. and C.S. Tsay, "A Stabilization Procedure for the Quadrilateral Plate Element with One-Point Quadrature," *Int. J. Num. Method. Eng.*, 19, 405-419 (1983).
- Belytschko, T.B., H. Stolarski, and N. Carpenter, "A C⁰ Triangular Plate Element with One-Point Quadrature," *Int. J. Num. Meth. Eng.*, 20, 787-802 (1984).
- Belytschko, T.B., L. Schwer, and M.J. Klein, "Large Displacement Transient Analysis of Space Frames," *Int. J. Num. Eng.*, 11, 65-84 (1977).
- Benson, D.J. and J.O. Hallquist, "A Simple Rigid Body Algorithm for Structural Dynamics Programs," *Int. J. Numer. Meth. Eng.*, 22, (1986).
- Benson, D.J. and J.O. Hallquist, "A Single Surface Contact Algorithm for the Postbuckling Analysis of Shell Structures," *Comp. Meths. Appl. Mech. Eng.*, 78, 141-163 (1990).
- Benzeggagh, M.L. and Kenane, M., "Measurement of Mixed-mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-mode Bending Apparatus," *Composites Science and Technology*, 56, 439-449 (1996).
- Berstad, T., "Material Modeling of Aluminium for Crashworthiness Analysis", Dr.Ing. Dissertation, Department of Structural Engineering, Norwegian University of Science and Technology, Trondheim, Norway, (1996).
- Berstad, T., Hopperstad, O.S., Lademo, O.-G. and Malo, K.A., "Computational Model of Ductile Damage and Fracture in Shell Analysis", Second European LS-DYNA Conference, Gothenburg, Sweden, (1999).
- Berstad, T., Lademo, O.-G., Pedersen, K.O. and Hopperstad, O.S., "Formability modeling with LS-DYNA", 8th International LS-DYNA User's Conference, Detroit, May 3-5, 2004.

REFERENCES

- Berstad, T., Langseth, M. and Hopperstad, O.S., "Elasto-viscoplastic Constitutive Models in the Explicit Finite Element Code LS-DYNA3D," Second International LS-DYNA3D conference, San Francisco, (1994).
- Bilkhu, S.S., M. Founas, and G.S. Nasholtz, "Material Modeling of Structural Foams in Finite Element Analysis Using Compressive Uniaxial and Triaxial Data," SAE (Nat. Conf.) Detroit 1993, pp. 4-34.
- Blatz, P.J., and Ko, W.L., "Application of Finite Element Theory to the Deformation of Rubbery Materials," *Trans. Soc. of Rheology*, 6, 223-251 (1962).
- Boyce, M.C., Parks, D.M., and Argon, A.S., "Large inelastic deformation of glassy polymers. Part I: Rate dependent constitutive model". *Mechanics of Materials*, 7, 15-33 (1988).
- Boyce, M.C., Socrate, C. and Llana, P.G., "Constitutive model for the finite deformation stress-strain behavior of poly(ethylene terephthalate) above the glass transition". *Polymer*, 41, 2183-2201 (2000).
- Brekelmans, W.A.M., Scheurs, P.J.G., and de Vree, J.H.P., 1991, "Continuum damage mechanics for softening of brittle materials", *Acta Mechanica*, vol 93, pp 133-143
- Broadhouse, B.J., "The Winfrith Concrete Model in LS-DYNA3D," Report: SPD/D(95)363, Structural Performance Department, AEA Technology, Winfrith Technology Centre, U.K. (1995).
- Broadhouse, B.J. and Neilson, A.J., "Modelling Reinforced Concrete Structures in DYNA3D", Safety and Engineering Division, United Kingdom Atomic Energy Authority, Winfrith, AEEW-M 2465, 1987.
- Brode, H.L., "Height of Burst Effects at High Overpressure," RAND, RM-6301-DASA, DASA 2506, (1970).
- Brown, B.E. and J.O. Hallquist, "TAURUS: An Interactive Post-Processor for the Analysis Codes NIKE3D, DYNA3D, TACO3D, and GEMINI," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19392 (1982) Rev. 1 (1984).
- Bruneau, M., Uang, C.M., Whittaker, A., Ductile Design of Steel Structures, McGraw Hill, (1998).
- Burton, D.E. et al. "Physics and Numerics of the TENSOR Code," Lawrence Livermore National Laboratory, Internal Document UCID-19428, (July 1982).
- CEB Code 1993, Comite euro-international du beton, *CEB-FIP Model Code 1990*, Thomas Telford, London, (1993).
- Chang, F.K. and K.Y. Chang, "A Progressive Damage Model for Laminated Composites Containing Stress Concentration," *J. of Composite Materials*, 21, 834-855 (1987a).

REFERENCES

- Chang, F.K. and K.Y. Chang, "Post-Failure Analysis of Bolted Composite Joints in Tension or Shear-Out Mode Failure," *J. of Composite Materials*, 21 809-823 (1987b).
- Chang, F.S., "Constitutive Equation Development of Foam Materials," Ph.D. Dissertation, submitted to the Graduate School, Wayne State University, Detroit, Michigan (1995).
- Chen, W.F., and Baladi, G.Y., Soil Plasticity: Theory and Implementation, Elsevier, New York, (1985).
- Cheng, H., Obergefell, L.A., and Rizer, A., March 1994, "Generator of Body (GEBOD) Manual," Report No. AL/CF-TR-1994-0051.
- Chowdhury, S.R. and Narasimhan R., "A Cohesive Finite Element Formulation for Modeling Fracture and Delamination in Solids," *Sadhana*, 25(6), 561-587, (2000).
- Christensen, R.M. "A Nonlinear Theory of Viscoelasticity for Application to Elastomers," *Journal of Applied Mechanics*, Volume 47, American Society of Mechanical Engineers, pages 762-768, December 1980.
- Chu, C.C. and A. Needleman, "Void Nucleation Effects in Biaxially Stretched Sheets", ASME Journal of Engineering Materials and Technology, 102, 249-256 (1980).
- Chung, K. and K. Shah, "Finite Element Simulation of Sheet Metal Forming for Planar Anisotropic Metals," *Int. J. of Plasticity*, 8, 453-476, (1992).
- Cochran, S.G. and J. Chan, "Shock Initiation and Detonation Models in One and Two Dimensions," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18024 (1979).
- Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley and Sons, Inc. (1974).
- Couch, R., E. Albright, and N. Alexander, The Joy Computer Code, Lawrence Livermore National Laboratory, Internal Document Rept. UCID-19688, (January, 1983).
- Cowper, G.R. and P.S. Symonds, Strain Hardening and Strain Rate Effects in the Impact Loading of Cantilever Beams, Brown University, Applied Mathematics Report, 1958.
- CRAY-1 Computer System CFT Reference Manual, Cray Research Incorporated, Bloomington, NM., Publication No. 2240009 (1978).
- DeRuntz, J.A. Jr., "Reference Material for USA, The Underwater Shock Analysis Code, USA-STAGS, and USA-STAGS-CFA," Report LMSC-P032568, Computational Mechanics Laboratory, Lockheed Palo Alto Research Laboratory, Palo Alto, CA. (1993).
- Desai, C.S., and H.J. Siriwardane, Constitutive Laws for Engineering Materials with Emphasis On Geologic Materials, Prentice-Hall, Chapter 10, (1984).
- Deshpande, V.S. and N.A. Fleck, "Isotropic Models for Metallic Foams," *Journal of the Mechanics and Physics of Solids*, 48, 1253-1283, (2000).

REFERENCES

- Dick, R.E., and W.H. Harris, "Full Automated Rezoning of Evolving Geometry Problems," Numerical Methods in Industrial Forming Processes, Chenot, Wood, and Zienkiewicz, Editors, Bulkema, Rotterdam, 243-248, (1992).
- Dilger, W.H., R. Koch, and R. Kowalczyk, "Ductility of Plain and Confined Concrete Under Different Strain Rates," *ACI Journal*, January-February, (1984).
- Dobratz, B.M., "LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52997 (1981).
- Du Bois, P.A., "Numerical Simulation of Strandfoam" Daimler-Chrysler AG Abt. EP/CSV, Report (2001).
- Dufailly, J., and Lemaitre, J., "Modeling very low cycle fatigue", *International Journal of damage mechanics*, 4, pp. 153-170 (1995).
- Englemann, B. E., R.G. Whirley, and G.L. Goudreau, "A Simple Shell Element Formulation for Large-Scale Elastoplastic Analysis," CED-Vol. 3. Analytical and Computational Models of Shells, A.K. Noor, T. Belytschko, and J.C. Simo, Editors, 1989, pp. 399-416.
- Faßnacht, W., "Simulation der Rißbildung in Aluminiumgußbauteilen," Dissertation, Technische Universität Darmstadt, (1999).
- Feng, W.W. and Hallquist, J.O., "On Constitutive Equations for Elastomers and Elastomeric Foams", The 4th European LS-DYNA Conference, D-II-15, Ulm, Germany, May 2003.
- Feucht, M., "Ein gradientenabhängiges Gursonmodell zur Beschreibung duktiler Schädigung mit Entfestigung," Dissertation, Technische Universität Darmstadt, (1998).
- Fiolka, M. and Matzenmiller, A., "Delaminationsberechnung von Faserverbundstrukturen", *PAMM Proc. Appl. Math. Mech.* 5, S.393-394 (2005).
- Flanagan, D.P. and T. Belytschko, "A Uniform Strain Hexahedron and Quadrilateral and Orthogonal Hourglass Control," *Int. J. Numer. Meths. Eng.*, 17, 679-706 (1981).
- Fung, Y.C., Biomechanics, Springer, New York, 1993.
- Fung, Y.C., Foundations of Solid Mechanics, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1965.
- Gerlach, S., Fiolka M. and Matzenmiller, A., Modelling and analysis of adhesively bonded joints with interface elements for crash analysis, 4. LS-DYNA Forum, 20-21, (2005) Bamberg, DYNAmore GmbH, Stuttgart.
- Ginsberg, M. and J. Johnson, "Benchmarking the Performance of Physical Impact Simulation Software on Vector and Parallel Computers," Applications Track of Supercomputing, IEEE monograph, Computer Society Press, March, 1989.

REFERENCES

- Giroux, E.D. HEMP User's Manual, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).
- Goudreau, G.L. and J.O. Hallquist, "Recent Developments in Large Scale Finite Element Lagrangian Hydrocode Technology," *J. Comp. Meths. Appl. Mechs. Eng.*, 30 (1982).
- Goldak, J., Chakravarti, A., and Bibby, M., "A New Finite Element Model for Welding Heat Sources," *Metallurgical Transactions B*, vol. 15B, pp. 299-305, June, 1984.
- Govindjee, S., Kay, J.G., and Simo, J.C. [1994], Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete, Report No. UCB/SEMM-94/18, Department of Civil Engineering, University of California, Berkeley, CA 94720.
- Govindjee, S., Kay, J.G., and Simo, J.C. [1995], "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," *Int. J. Numer. Meth. Engng*, **38**, 3611-3633.
- Graefe, H., W. Krummheuer, and V. Siejak, "Computer Simulation of Static Deployment Tests for Airbags, Air Permeability of Uncoated Fabrics and Steady State Measurements of the Rate of Volume Flow Through Airbags," SAE Technical Paper Series, 901750, Passenger Car Meeting and Exposition, Dearborn, Michigan, September 17-20, 1990.
- Gran, J.K. and P.E. Senseny, "Compression Bending of Scale-Model Reinforced-Concrete Walls," *ASCE Journal of Engineering Mechanics*, Volume 122, Number 7, pages 660-668, July (1996).
- Guccione, J., A. McCulloch, and L. Waldman, "Passive Material Properties of Intact Ventricular Myocardium Determined from a Cylindrical Model", *ASME Journal of Biomechanical Engineering*, Vol. 113, pages 42-55, (1991).
- Gurson, A.L., Plastic Flow and Fracture Behavior of Ductile Materials Incorporating Void Nucleation, Growth, and Interaction, Ph.D. Thesis, Brown University, (1975).
- Gurson, A.L., "Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I - Yield Criteria and Flow Rules for Porous Ductile Media", *J. of Eng. Materials and Technology*, (1977).
- Hallquist, J.O., Preliminary User's Manuals for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimension), University of California, Lawrence Livermore National Laboratory, Rept. UCID-17268 (1976) and Rev. 1 (1979).[a]
- Hallquist, J.O., A Procedure for the Solution of Finite Deformation Contact-Impact Problems by the Finite Element Method, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52066 (1976).
- Hallquist, J.O., "A Numerical Procedure for Three-Dimensional Impact Problems," *American Society of Civil Engineering*, Preprint 2956 (1977).
- Hallquist, J.O., "A Numerical Treatment of Sliding Interfaces and Impact," in: K.C. Park and D.K. Gartling (eds.) *Computational Techniques for Interface Problems*, AMD Vol. 30, ASME, New York (1978).

REFERENCES

- Hallquist, J.O., NIKE2D: An Implicit, Finite-Element Code for Analyzing the Static and Dynamic Response of Two-Dimensional Solids, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52678 (1979).[b]
- Hallquist, J.O., User's Manual for DYNA2D – An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning, University of California, Lawrence Livermore National Laboratory, Rept. UCID-18756 (1980).
- Hallquist, J.O., User's Manual for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1981).[a]
- Hallquist, J. O., NIKE3D: An Implicit, Finite-Deformation, Finite-Element Code for Analyzing the Static and Dynamic Response of Three-Dimensional Solids, University of California, Lawrence Livermore National Laboratory, Rept. UCID-18822 (1981).[b]
- Hallquist, J.O., DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1982; Rev. 1: 1984; Rev. 2: 1986).
- Hallquist, J.O., Theoretical Manual for DYNA3D, University of California, Lawrence Livermore National Laboratory, Rept. UCID-19501 (March, 1983).
- Hallquist, J.O., DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1988, Rev. 4).
- Hallquist, J.O., LS-DYNA User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), Livermore Software Technology Corporation, Rept. 1007 (1990).
- Hallquist, J.O., D.J. Benson, and G.L. Goudreau, "Implementation of a Modified Hughes-Liu Shell into a Fully Vectorized Explicit Finite Element Code," Proceedings of the International Symposium on Finite Element Methods for Nonlinear Problems, University of Trondheim, Trondheim, Norway (1985).
- Hallquist, J.O. and D.J. Benson, "A Comparison of an Implicit and Explicit Implementation of the Hughes-Liu Shell," Finite Element Methods for Plate and Shell Structures, T.J.R. Hughes and E. Hinton, Editors, 394-431, Pineridge Press Int., Swansea, U.K. (1986).
- Hallquist, J.O. and D.J. Benson, DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (Rev. 2: 1986; Rev. 3: 1987).
- Hallquist, J.O., D.W. Stillman, T.J.R. Hughes, C. and Tarver,"Modeling of Airbags Using MVMA/DYNA3D," LSTC Report (1990).
- Hashin, Z, "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, 47, 329 (1980).

REFERENCES

- Hänsel, C., P. Hora, and J. Reissner, "Model for the Kinetics of Strain-Induced Martensitic Phase Transformation at Isothermal Conditions for the Simulation of Sheet Metal Forming Processes with Metastable Austenitic Steels," *Simulation of Materials Processing: Theory, Methods, and Applications*, Huétink and Baaijens (eds), Balkema, Rotterdam, (1998).
- Haward, R.N., and Thackray, G., "The use of a mathematical model to describe isothermal stress-strain curves in glassy thermoplastics". *Proc Roy Soc A*, 302, 453-472 (1968).
- Herrmann, L.R. and F.E. Peterson, "A Numerical Procedure for Viscoelastic Stress Analysis," Seventh Meeting of ICRPG Mechanical Behavior Working Group, Orlando, FL, CPIA Publication No. 177, 1968.
- Hill A.V., "The heat of shortening and the dynamic constants of muscle," *Proc Roy Soc B*126:136-195, (1938).
- Hill, R., "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," *Proceedings of the Royal Society of London, Series A.*, Vol. 193, pp. 281-197 (1948).
- Hill, R., "Aspects of Invariance in Solid Mechanics," *Advances in Applied Mechanics*, Vol. 18, pp. 1-75 (1978).
- Hill, R., "Constitutive Modeling of Orthotropic Plasticity in Sheet Metals," *J. Mech. Phys. Solids*, Vol. 38, No. 3, 1989, pp. 405-417.
- Hirth, A., P. Du Bois, and K. Weimar, "Improvement of LS-DYNA Material Law 83 (Fu Chang) for the Industrial Simulation of Reversible Energy-Absorbing Foams," CAD-FEM User's Meeting, Bad Neuenahr - Ahrweiler, Germany, October 7-9, Paper 2-40, (1998).
- Holmquist, T.J., G.R. Johnson, and W.H. Cook, "A Computational Constitutive Model for Concrete Subjected to Large Strains, High Strain Rates, and High Pressures", *Proceedings 14th International Symposium on Ballistics*, Quebec, Canada, pp. 591-600, (1993).
- Hopperstad, O.S. and Remseth, S., "A return Mapping Algorithm for a Class of Cyclic Plasticity Models", *International Journal for Numerical Methods in Engineering*, Vol. 38, pp. 549-564, (1995).
- Huang, Yuli, private communication, Livermore, (2006).
- Hughes, T.J.R. and E. Carnoy, "Nonlinear Finite Element Shell Formulation Accounting for Large Membrane strains," *AMD-Vol.48*, ASME, 193-208 (1981).
- Hughes, T.J.R. and W.K. Liu, "Nonlinear Finite Element Analysis of Shells: Part I. Three-Dimensional Shells." *Comp. Meths. Appl. Mechs.*, 27, 331-362 (1981a).
- Hughes, T.J.R. and W.K. Liu, "Nonlinear Finite Element Analysis of Shells: Part II. Two-Dimensional Shells." *Comp. Meths. Appl. Mechs.*, 27, 167-181 (1981b).

REFERENCES

- Hughes, T.J.R., W.K. Liu, and I. Levit, "Nonlinear Dynamics Finite Element Analysis of Shells." Nonlinear Finite Element Analysis in Struct. Mech., Eds. W. Wunderlich, E. Stein, and K.J. Bathe, Springer-Verlag, Berlin, 151- 168 (1981c).
- Huh, H. and Kang, W.J., "Crash-Worthiness Assessment of Thin-Walled Structures with the High-Strength Steel Sheet", *Int. Journal of Vehicle Design*, Vol. 30, Nos. 1/2 (2002).
- Ibrahimbegovic, A. and Wilson, E.L. "A unified formulation for triangular and quadrilateral flat shell finite elements with six nodal degrees of freedom", *Comm. Applied Num. Meth*, 7, 1-9 (1991).
- Isenberg, J., Vaughan, D.K., Sandler, I.S., Nonlinear Soil-Structure Interaction, Electric Power Research Institute report EPRI NP-945, Weidlinger Associates (1978).
- Johnson, G.C. and D.J. Bammann, "A discussion of stress rates in finite deformation problems," *Int. J. Solids Struct*, 20, 725-737 (1984).
- Johnson, G.R. and W.H. Cook, "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures." Presented at the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April 1983.
- Johnson, G.R. and T.J. Holmquist, "An Improved Computational Model for Brittle Materials" in High-Pressure Science and Technology - 1993 American Institute of Physics Conference Proceedings 309 (c 1994) pp.981-984 ISBN 1-56396-219-5.
- Jones, R.M., Mechanics of Composite Materials, Hemisphere Publishing Corporation, New York, (1975).
- Kenchington, G.J., "A Non-Linear Elastic Material Model for DYNA3D," Proceedings of the DYNA3D Users Group Conference, published by Boeing Computer Services (Europe) Limited (1988).
- Key, S.W. HONDO – A Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids, Sandia National Laboratories, Albuquerque, N.M., Rept. 74-0039 (1974).
- Kolling, S., Haufe, A., Feucht, M., DuBois, P. A. "SAMP-1: A Semi-Analytical Model for the Simulation of Polymers", 4. LS-DYNA Anwenderforum, October 20-21, Bamberg, Germany, (2005).
- Kolling, S., Hirth, A., Erhart, and Du Bois P.A., Private Communication, Livermore, California (2006).
- Krieg, R.D., A Simple Constitutive Description for Cellular Concrete, Sandia National Laboratories, Albuquerque, NM, Rept. SC-DR-72-0883 (1972).
- Krieg, R.D. and S.W. Key, "Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs," Vol. 20 of Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects (American Society of Mechanical Engineers, New York, N.Y., pp. 125-137 (1976).

REFERENCES

- Lademo, O.G., Berstad, T., Tryland, T., Furu, T., Hopperstad, O.S. and Langseth, M., “A model for process-based crash simulation”, 8th International LS-DYNA User’s Conference, Detroit, May 3-5, 2004.
- Lademo, O.G., Hopperstad, O.S., Berstad, T. and Langseth M., “Prediction of Plastic Instability in Extruded Aluminum Alloys Using Shell Analysis and a Coupled Model of Elasto-plasticity and Damage,” *Journal of Materials Processing Technology*, 2002 (Article in Press).
- Lademo, O.G., Hopperstad, O.S., Malo, K.A. and Pedersen, K.O., “Modelling of Plastic Anisotropy in Heat-Treated Aluminum Extrusions”, *Journal of Materials Processing Technology* **125-126**, pp. 84-88 (2002).
- Lee, E.L. and C.M. Tarver, “Phenomenological Model of Shock Initiation in Heterogenous Explosives,” *PHYS. Fluids*, Vol. 23, p. 2362 (1980).
- Lemaitre, J., A Course on Damage Mechanics, Springer-Verlag, (1992).
- Lemaitre, J., and Chaboche, J.L., Mechanics of Solid Materials, Cambridge University Press, (1990).
- Lemmen, P. P. M. and Meijer, G. J., “Failure Prediction Tool Theory and User Manual,” TNO Report 2000-CMC-R0018, (2001).
- Lewis, B.A., “Developing and Implementing a Road Side Safety Soil Model into LS-DYNA,” FHWA Research and Development Turner-Fairbank Highway Research Center, (1999).
- Li, Y.H. and Sellars, C.M., “Modeling Deformation Behavior of Oxide Scales and their Effects on Interfacial Heat Transfer and Friction during Hot Steel Rolling”, *Proc. Of the 2nd Int. Conf. Modeling of Metals Rolling Processes*, The Insitute of Materials, Londong, UK, 192-201 (1996).
- Lian, W., Memo entitled: “LS-DYNA Airbag Module Improvement Request”, General Motors Corporation (2000).
- MADYMO3D USER’S MANUAL, Version 4.3, TNO Road-Vehicles Research Institute, Department of Injury Prevention, The Hague, The Netherlands, (1990).
- Maker, B.N., Private communication Lawrence Livermore National Laboratory, Dr. Maker programmed and implemented the compressible Mooney Rivlin rubber model (1987).
- Malvar, L.J., Crawford, J.E., Morrill, K.B., K&C Concrete Material Model Release III — Automated Generation of Material Model Input, K&C Technical Report TR-99-24-B1, 18 August 2000 (*Limited Distribution*).
- Malvar, L.J., Crawford, J.E., Wesevich, J.W., Simons, D., “A Plasticity Concrete Material Model for DYNA3D,” *International Journal of Impact Engineering*, Volume 19, Numbers 9/10, pages 847-873, December 1997.

REFERENCES

- Malvar, L.J., and Ross, C.A., "Review of Static and Dynamic Properties of Concrete in Tension," *ACI Materials Journal*, Volume 95, Number 6, pages 735-739, November-December 1998.
- Malvar, L.J., and Simons, D., "Concrete Material Modeling in Explicit Computations," Proceedings, Workshop on Recent Advances in Computational Structural Dynamics and High Performance Computing, USAE Waterways Experiment Station, Vicksburg, MS, pages 165-194, April 1996. (LSTC may provide this reference upon request.)
- Malvar, H.S., Sullivan, G.S., and Wornell, G.W., "Lapped Orthogonal Vector Quantization", in Proc. Data Compression Conference, Snowbird, Utah, 1996.
- Marin, E.B., unpublished paper, Sandia National Laboratory, CA (2005).
- Matzenmiller, A., Lubliner, J., and Taylor, R.L., "A Constitutive Model for Anisotropic Damage in Fiber-Composites," *Mechanics of Materials*, Vol. 20, pp. 125-152 (1995).
- Matzenmiller, A. and J. K. Schweizerhof, "Crashworthiness Considerations of Composite Structures – A First Step with Explicit Time Integration in Nonlinear Computational Mechanics–State-of-the-Art," Ed. P. Wriggers, W. Wagner, Springer Verlag, (1991).
- Mauldin, P.J., R.F. Davidson, and R.J. Henninger, "Implementation and Assessment of the Mechanical-Threshold-Stress Model Using the EPIC2 and PINON Computer Codes," Report LA-11895-MS, Los Alamos National Laboratory (1990).
- McCormick, P.G., "Theory of flow localization due to dynamic strain ageing," *Acta Metallurgica*, 36, 3061-3067 (1988).
- Mi Y., Crisfield, M.A., Davies, A.O. Progressive delamination using interface elements. *J Compos Mater*, 32(14)1246-72 (1998).
- Moran, B., Ortiz, M. and Shih, C.F., "Formulation of implicit finite element methods for multiplicative finite deformation plasticity". *Int J for Num Methods in Engineering*, 29, 483-514 (1990).
- de Moura MFSF, Gonçalves, J.P., Marques, A.T., and de Castro, P.T., Elemento finito isoparamétrico de interface para problemas tridimensionais. *Revista Internacional de Métodos Numéricos Para Cálculo e Diseño en Ingeniería*, 14:447-66 (1996).
- Murray, Y.D., Users Manual for Transversely Isotropic Wood Model APTEK, Inc., Technical Report to the FHWA (to be published) (2002).
- Murray, Y.D. and Lewis, B.A., Numerical Simulation of Damage in Concrete APTEK, Inc., Technical Report DNA-TR-94-190, Contract DNA 001-91-C-0075, Defense Nuclear Agency, Alexandria VA 22310.
- Murray, Y.D., Users Manual for LS-DYNA Concrete Material Model 159, Report No. FHWA-HRT-05-062, Federal Highway Administration, (2007).

REFERENCES

- Murray, Y.D., A. Abu-Odeh, and R. Bligh, Evaluation of Concrete Material Model 159, Report No. FHWA-HRT-05-063, Federal Highway Administration, (2007).
- Nagararajah, Reinhorn, & Constantinou, "Nonlinear Dynamic Analysis of 3-D Base-Isolated Structures", *Journal of Structural Engineering* Vol 117, No 7, (1991).
- Neal, M.O., C-H Lin, and J. T. Wang, "Aliasing Effects on Nodal Acceleration Output from Nonlinear Finite Element Simulations," ASME 2000 International Mechanical Engineering Congress and Exposition, Orlando, Florida, November 5-10, (2000).
- Neilsen, M.K., H.S. Morgan, and R.D. Krieg, "A Phenomenological Constitutive Model for Low Density Polyurethane Foams," Rept. SAND86-2927, Sandia National Laboratories, Albuquerque, N.M., (1987).
- Nusholtz, G., W. Fong, and J. Wu, "Air Bag Wind Blast Phenomena Evaluation," *Experimental Techniques*, Nov.-Dec. (1991).
- Nusholtz, G., D. Wang, and E.B. Wylie, "Air Bag Momentum Force Including Aspiration," Preprint, Chrysler Corporation, (1996).
- Nusholz, private communication, (1996).
- Ogden, R.W., Non-Linear Elastic Deformations, Ellis Horwood Ltd., Chichester, Great Britain (1984).
- Oliver, J., "A Consistent Characteristic Length of Smeared Cracking Models," *International Journal for Numerical Methods in Engineering*, **28**, 461-474 (1989).
- Papadrakakis, M., "A Method for the Automatic Evaluation of the Dynamic Relaxation Parameters," *Comp. Meth. Appl. Mech. Eng.*, Vol. 25, pp. 35-48 (1981).
- Park, R. and Paulay, T., (1975) Reinforced Concrete Structures, J. Wiley and Sons, New York.
- Park, Y.J., Wen, Y.K, and Ang, A.H-S, "Random Vibration of Hysteretic Systems Under Bi-directional Ground Motions", *Earthquake Engineering and Structural Dynamics*, Vol. 14, pp. 543-557 (1986).
- Penelis, G.G. and Kappos, A.J., Earthquake-Resistant Concrete Structures, E&FN Spon., (1997).
- Pijaudier-Cabot, G., and Bazant, Z.P., "Nonlocal Damage Theory," *Journal of Engineering Mechanics*, ASCE, Vol. 113, No. 10, 1512-1533 (1987).
- Puso, M.A., "A Highly Efficient Enhanced Assumed Strain Physically Stabilized Hexahedral Element", *Int. J. Numer. Meth. Eng.*, Vol. 49, 1029-1064 (2000).
- Puso, M.A. and Weiss, J.A., "Finite Element Implementation of Anisotropic Quasilinear Viscoelasticity Using a Discrete Spectrum Approximation", *ASME J. Biomech. Engng.*, 120, 62-70 (1998).

REFERENCES

- Pelessone, D., Private communication, GA Technologies, P.O. Box 85608, San Diego, CA., Telephone No. 619-455-2501 (1986).
- Quapp, K.M. and Weiss, J.A., "Material Characterization of Human Medial Collateral Ligament", *ASME J. Biomech Engng.*, 120, 757-763 (1998).
- Reyes, A., O.S. Hopperstad, T. Berstad, and M. Langseth, Implementation of a Material Model for Aluminium Foam in LS-DYNA, Report R-01-02, Restricted, Department of Structural Engineering, Norwegian University of Science and Technology, (2002).
- Randers-Pehrson, G. and K. A. Bannister, Airblast Loading Model for DYNA2D and DYNA3D, Army Research Laboratory, Rept. ARL-TR-1310, publicly released with unlimited distribution, (1997).
- Roussis, P.C., and Constantinou, M.C., "Uplift-restraining Friction Pendulum seismic isolation system", *Earthquake Engineering and Structural Dynamics*, 35 (5), 577-593, (2006).
- Rupp, A., Grubisic, V., and Buxbaum, O., Ermittlung ertragbarer Beanspruchungen am Schweisspunkt auf Basis der ubertragbaren Schnittgrossen, FAT Schriftenreihe 111, Frankfurt (1994).
- Sala, M.O. Neal, and J.T. Wang, Private Communication, General Motors, May, 2004.
- Sackett, S.J., "Geological/Concrete Model Development," Private Communication (1987).
- Sandler, I.S. and D. Rubin, "An Algorithm and a Modular Subroutine for the Cap Model," *Int. J. Numer. Analy. Meth. Geomech.*, 3, pp. 173-186 (1979).
- Schedin, E., Prentzas, L. and Hilding D., "Finite Element Simulation of the TRIP-effect in Austenitic Stainless Steel," presented at SAE 2004, SAE Technical paper 2004-01-0885, (2004).
- Schwer, L.E., "A Viscoplastic Augmentation of the Smooth Cap Model," *Nuclear Engineering and Design*, Vol. 150, pp. 215-223, (1994).
- Schwer, L.E., "Demonstration of the Continuous Surface Cap Model with Damage: Concrete Unconfined Compression Test Calibration," LS-DYNA Geomaterial Modeling Short Course Notes, July (2001).
- Schwer, L.E., W. Cheva, and J.O. Hallquist, "A Simple Viscoelastic Model for Energy Absorbers Used in Vehicle-Barrier Impact," in *Computation Aspects of Contact, Impact, and Penetration*, Edited by R.F. Kulak and L.E. Schwer, Elmepress International, Lausanne, Switzerland, pp. 99-117 (1991).
- Schwer, L.E. and Y.D. Murray, "A Three-Invariant Smooth Cap Model with Mixed Hardening," *International Journal for Numerical and Analytical Methods in Geomechanics*, Volume 18, pp. 657-688, (1994).

REFERENCES

- Seeger, F., M. Feucht, T. Frank (DaimlerChrysler AG), and A. Haufe, B. Keding (DYNAmore GmbH), "An Investigation on Spotweld Modeling for Crash Simulation with LS-DYNA", 4th LS-DYNA-Forum, Bamberg, Germany, October (2005), ISBN 3-9809901-1-7.
- Sheppard, S.D., Estimations of Fatigue Propagation Life in Resistance Spot Welds, ASTM STP 1211, pp. 169-185, (1993).
- Sheppard, T. and Wright, D.S., "Determination of flow stress: Part 1 constitutive equation for aluminum alloys at elevated temperatures", *Metals Technology*, p. 215, June 1979.
- Shvets, I.T. and Dyban, E., P., "Contact Heat Transfer between Plane Metal Surfaces", *Int. Chem. Eng.*, Vol. 4, No. 4, 621 (1964).
- Simo, J.C., J.W. Ju, K.S. Pister, and R.L. Taylor, "An Assessment of the Cap Model: Consistent Return Algorithms and Rate-Dependent Extension," *J. Eng. Mech.*, Vol. 114, No. 2, 191-218 (1988a).
- Simo, J.C., J.W. Ju, K.S. Pister, and R.L. Taylor, "Softening Response, Completeness Condition, and Numerical Algorithms for the Cap Model," *Int. J. Numer. Analy. Meth. Eng.*, (in press) (1988b).
- Simo, J. C., J.W. Ju, K.S. Pister, and R.L. Taylor, "Softening Response, Completeness Condition, and Numerical Algorithms for the Cap Model," *Int. J. Numer. Analy. Meth. Eng.* (1990).
- Solberg, J.M., and C.M. Noble, "Contact Algorithm for Small-Scale Surface Features with Application to Finite Element Analysis of Concrete Arch Dams with Beveled Contraction Joints", Lawrence Livermore National Laboratory (2002).
- Steinberg, D.J. and M.W. Guinan, A High-Strain-Rate Constitutive Model for Metals, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-80465 (1978).
- Steinberg, D.J. and C.M. Lund, "A Constitutive Model for Strain Rates from 10^{-4} to 10^6 S⁻¹," *J. Appl. Phys.*, 65, p. 1528 (1989).
- Stillman, D.W. and J.O. Hallquist, INGRID: A Three-Dimensional Mesh Generator for Modeling Nonlinear Systems, University of California, Lawrence Livermore National Laboratory, Rept. UCID-20506. (1985).
- Stojko, S., privated communication, NNC Limited, Engineering Development Center (1990).
- Storakers, B., "On material representation and constitutive branching in finite compressible elasticity", *J. Mech. Phy. Solids*, 34 No. 2, 125-145 (1986).
- Stouffer and Dame, *Inelastic Deformation of Metals*, Wiley, (1996).
- Stout, M.G., D.E. Helling, T.L. Martin, and G.R. Canova, *Int. J. Plasticity*, Vol. 1, pp. 163-174, (1985).

REFERENCES

- Structural Engineers Association of California, Tentative Lateral Force Requirements, Seismology Committee, SEAOC, 1974, 1990, 1996.
- Sussman, T. and Bathe, K.J., "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *Computers & Structures*, **26**, Number 1/2, 357-409 (1987).
- Tahoe User Guide, Sandia National Laboratory, can be downloaded from: www.sandia.gov, Input version 3.4.1, (2003).
- Taylor, L.M. and D.P. Flanagan, PRONTO3D A Three-Dimensional Transient Solid Dynamics Program, Sandia Report: SAND87-1912, UC-32, (1989).
- Taylor, R.L. Finite element analysis of linear shell problems, in *Whiteman, J.R. (ed.), Proceedings of the Mathematics in Finite Elements and Applications*, Academic Press, New York, 191-203, (1987).
- Taylor, R.L. and Simo, J.C. Bending and membrane elements for the analysis of thick and thin shells, *Proc. of NUMETA Conference*, Swansea (1985).
- Tsai, S.W. and E.M. Wu, "A General Theory of Strength for Anisotropic Materials," J. Composite Materials, 5, pp. 73-96 (1971).
- Tuler, F.R. and B.M. Butcher, "A Criterion for the Time Dependence of Dynamic Fracture," *The International Journal of Fracture Mechanics*, Vol. 4, No. 4, (1968).
- Tvergaard, V. and J.W. Hutchinson, "The relation between crack growth resistance and fracture process parameters in elastic-plastic solids," *J. of the Mech. And Phy. of Solids*, 40, pp1377-1397, (1992)
- Tvergaard, V. and Needleman, A., "Analysis of the cup-cone fracture in a round tensile bar", *Acta Metallurgica*, 32, 157-169 (1984).
- Vawter, D., "A Finite Element Model for Macroscopic Deformation of the Lung," published in the *Journal of Biomechanical Engineering*, Vol.102, pp. 1-7 (1980).
- VDA Richtlinier (Surface Interfaces), Version 20, Verband der Automobilindustrie e.v., Frankfurt, Main, Germany, (1987).
- Vegter, H., and Boogaard, A.H. van den, "A plane stress yield function for anisotropic sheet material by interpolation of biaxial stress states", *International Journal of Plasticity* 22, 557-580 (2006).
- Wang, J.T. and O.J. Nefske, "A New CAL3D Airbag Inflation Model," SAE paper 880654, 1988.
- Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", Publication R&D 8332, General Motors Development Center, Warren, Mi. (1995).

REFERENCES

- Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", *AMD-Vol. 210, BED-Vol. 30, ASME*, pp 467-497, (1995).
- Weiss, J.A., Maker, B.N. and Govindjee, S., "Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity", *Comp. Meth. Appl. Mech. Eng.*, 135, 107-128 (1996).
- Wen, T.K. "Method for Random Vibration of Hysteretic Systems", *J. Engrg. Mech., ASCE*, Vol. 102, No. EM2, Proc. Paper 12073, pp.249-263 (1976).
- Whirley, R. G., and J. O. Hallquist, DYNA3D, A Nonlinear, Explicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics-Users Manual, Report No.UCRL-MA-107254 , Lawrence Livermore National Laboratory, (1991).
- Whirley, R. G., and G.A. Henshall, "Creep Deformation Structural Analysis Using An Efficient Numerical Algorithm," *IJNME*, Vol. 35, pp. 1427-1442, (1992).
- Wilkins, M.L., "Calculations of Elastic Plastic Flow," *Meth. Comp. Phys.*, **3**, (Academic Press), 211-263 (1964).
- Wilkins, M.L., Calculation of Elastic-Plastic Flow, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-7322, Rev. I (1969).
- Wilkins, M.L., R.E. Blum, E. Cronshagen, and P. Grantham, A Method for Computer Simulation of Problems in Solid Mechanics and Gas Dynamics in Three Dimensions and Time, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51574 (1974).
- Wilkins, M.L., J.E. Reaugh, B. Moran, J.K. Scudder, D.F. Quinones, M.E. Prado, Fundamental Study of Crack Initiation and Propagation Annual Progress Report, Report UCRL-52296, Lawrence Livermore National Laboratory, Livermore, CA. (1977).
- Wilson, E.L. Three Dimensional Static and Dynamic Analysis of Structures, Computers and Structures, Inc., Berkeley CA, (2000).
- Winters, J.M., "Hill-based muscle models: A systems engineering perspective," In *Multiple Muscle Systems: Biomechanics and Movement Organization*, JM Winters and SL-Y Woo eds, Springer-Verlag (1990).
- Winters J.M. and Stark L., "Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints," *J Biomechanics* 21:1027-1042, (1988).
- Woodruff, J.P., KOVEC User's Manual, University of California, Lawrence Livermore National Laboratory, Report UCRL-51079, (1973).
- Worswick, M.J., and Xavier Lalbin, Private communication, Livermore, California, (1999).
- Yamasaki, H., M. Ogura, R. Nishimura, and K. Nakamura, "Development of Material Model for Crack Propagation of Casting Aluminum", Presented at the 2006 JSAE Annual Congress, Paper Number 20065077, (2006).

REFERENCES

- Yen, C.F., "Ballistic Impact Modeling of Composite Materials," Proceedings of the 7th International LS-DYNA Users Conference, Dearborn, MI, May 19-21, 2002, 6.15-6.25.
- Yoshida, F. and Uemori, T., "A Model of Large-Strain Cyclic Plasticity and its Application to Springback Simulation," *International Journal of Mechanical Sciences*, Vol. 45, 1687-1702, (2003).
- Zajac F.E., "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control", *CRC Critical Reviews in Biomedical Engineering* 17(4):359-411, (1989).
- Zayas, V.A., Low, S.S. and Mahin, S.A., "A Simple Pendulum Technique for Achieving Seismic Isolation", *J. Earthquake Spectra*, Vol. 6, No. 2, pp. 317-334 (1990).
- Zhang, S., Approximate Stress Intensity Factors and Notch Stresses for Common Spot-Welded Specimens, *Welding Research Supplement*, pp. 173s-179s, (1999).
- Zhang, S., McCormick, P.G., Estrin, Y., "The morphology of Portevin-Le Chatelier bands: Finite element simulation for Al-Mg-Si", *Acta Materialia* 49, 1087-1094, (2001).

APPENDIX A: User Defined Materials

The user can supply his/her own subroutines defining material models in LS-DYNA. To invoke a user-defined material, one must

1. Write a user material subroutine that is called by the LS-DYNA user material interface.
2. Create a custom executable which includes the material subroutine.
3. Invoke that subroutine by defining a part in the keyword input deck that uses `*MAT_USER_DEFINED_MATERIAL_MODELS` with appropriate input parameters.

All subroutines, including interface, for the user-defined materials are collected in the file `dyn21.F` (Unix/Linux) or `lsdyna.f` (Windows). Up to ten user subroutines can currently be implemented simultaneously to update the stresses in solids, shells, beams, discrete beams and truss beams. This text serves as an introductory guide to implement such a model. Note that names of variables and subroutines below may differ from the actual ones depending on platform and current version of LS-DYNA.

General overview

When the keyword `*MAT_USER_DEFINED_MATERIAL_MODELS` is defined for a part in the keyword deck, LS-DYNA calls the subroutine `usrmat` with appropriate input data for the constitutive update. This routine in turn calls `urmathn` for 2D and 3D solid elements, `urmats` for 2D plane stress and 3D shell elements, `urmatb` for beam elements, `urmatd` for discrete beam elements and `urmatt` for truss beam elements. In these routines, which may be modified by the user if necessary, the following data structures are initialized for the purpose of being supplied to a specific *scalar* material subroutine.

`sig(6)` - stresses in previous time step
`eps(6)` - strain increments
`epsp` - effective plastic strain in previous time step
`hsv(*)` - history variables in previous time step excluding plastic strain
`dt1` - current time step size
`temper` - current temperature
`faile1` - flag indicating failure of element

If the *vectorization* flag is active (`IVECT=1`) on the material card, variables are in general stored in vector blocks of length `nlq`, with vector indexes ranging from `1ft` to `1lt`, which allows for a more efficient execution of the material routine. As an example, the data structures mentioned above are for the vectorized case exchanged for

`sigX(nlq)` - stresses in previous time step
`dX(nlq)` - strain increments
`epsps(nlq)` - effective plastic strains in previous time step
`hsvs(nlq,*)` - history variables in previous time step
`dtlsiz(nlq)` - current time step sizes
`temps(nlq)` - current temperatures
`failels(nlq)` - flags indicating failure of elements

where x ranges from 1 to 6 for the different components. Each entry in a vector block is associated with an integration point in the finite element mesh.

The number of entries in the history variables array (indicated by * in the above) matches the number of history variables requested on the material card (NHV). Hence the number NHV should equal to the number of history variables excluding the effective plastic strain since this variable is given a special treatment. All history variables, including the effective plastic strain, are initially zero. Furthermore, all user-defined material models require a bulk modulus and shear modulus for transmitting boundaries, contact interfaces, rigid body constraints, and time step calculations. This generally means that the length of material constants array LMC must be increased by 2 for the storage of these parameters. In addition to the variables mentioned above, the following data can be supplied to the user material routines, regardless of whether vectorization is used or not.

- cm(*) - material constants array
- capa - transverse shear correction factor for shell elements
- tt - current time
- crv(101,2,*) - array representation of curves defined in the keyword deck

A specific material routine, `umatXX` in the scalar case or `umatXXv` in the vector case, is now called with any necessary parameters of the ones above, and possibly others as well. The letters `XX` stands for a number between 41 and 50 and matches the number `MT` on the material card. This subroutine is written by the user, and should update the stresses and history variables to the current time. For shells and beams it is also necessary to determine the strain increments in the directions of constrained zero stress. To be able to write different stress updates for different elements, the following character string is passed to the user-defined subroutine

- etype - character string that equals `solid`, `shell`, `beam`, `dbeam` OR `tbeam`

A sample user subroutine of a hypo-elastic material in the scalar case is provided below.

Sample user subroutine 41

```

      subroutine umat41 (cm,eps,sig,eps,hs,dtl,capa,etype,
      .      tt,temper,failel,crv)
c*****
c | livermore software technology corporation (lstc) |
c | ----- |
c | copyright 1987-2003 |
c | all rights reserved |
c*****
c
c   Isotropic elastic material (sample user subroutine)
c
c   Variables
c
c   cm(1)=first material constant, here young's modulus
c   cm(2)=second material constant, here poisson's ratio
c   .
c   .
c   .
c   cm(n)=nth material constant
c
c   eps(1)=local x strain increment
c   eps(2)=local y strain increment
c   eps(3)=local z strain increment

```

```

c     eps(4)=local xy strain increment
c     eps(5)=local yz strain increment
c     eps(6)=local zx strain increment
c
c     sig(1)=local x  stress
c     sig(2)=local y  stress
c     sig(3)=local z  stress
c     sig(4)=local xy stress
c     sig(5)=local yz stress
c     sig(6)=local zx stress
c
c     epsp=effective plastic strain
c
c     hsv(1)=1st history variable
c     hsv(2)=2nd history variable
c     .
c     .
c     .
c     hsv(n)=nth history variable
c
c     capa=reduction factor for transverse shear
c
c     etype:
c     eq."brick" for solid elements
c     eq."shell" for all shell elements
c     eq."beam"  for all beam elements
c     eq."dbeam" for all discrete beam elements
c
c     temper=current temperature
c
c     dt1=current time step size
c
c     tt=current problem time.
c
c     crv=array representation of curves defined in keyword deck
c
c     failfl=flag for failure, set to .true. to fail an element
c
c     All transformations into the element local system are
c     performed prior to entering this subroutine. Transformations
c     back to the global system are performed after exiting this
c     routine.
c
c     All history variables are initialized to zero in the input
c     phase. Initialization of history variables to nonzero values
c     may be done during the first call to this subroutine for each
c     element.
c
c     Energy calculations for the energy balance are done
c     outside this subroutine.
c
c
c     character*(*) etype
c     dimension  cm(*),eps(*),sig(*),hsv(*),crv(101,2,*)
c     logical failfl
c
c     compute shear modulus, g
c     g2 =cm(1)/(1.+cm(2))
c     g  =.5*g2
c
c     if (etype.eq.'brick') then
c

```

```

davg=(-eps(1)-eps(2)-eps(3))/3.
p=-davg*cm(1)/((1.-2.*cm(2)))
sig(1)=sig(1)+p+g2*(eps(1)+davg)
sig(2)=sig(2)+p+g2*(eps(2)+davg)
sig(3)=sig(3)+p+g2*(eps(3)+davg)
sig(4)=sig(4)+g*eps(4)
sig(5)=sig(5)+g*eps(5)
sig(6)=sig(6)+g*eps(6)
c
elseif (etype.eq.'shell') then
c
gc      =capa*g
q1      =cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
q3      =1./(q1+g2)
eps(3)=-q1*(eps(1)+eps(2))*q3
davg    =(-eps(1)-eps(2)-eps(3))/3.
p       =-davg*cm(1)/((1.-2.*cm(2)))
sig(1)=sig(1)+p+g2*(eps(1)+davg)
sig(2)=sig(2)+p+g2*(eps(2)+davg)
sig(3)=0.0
sig(4)=sig(4)+g *eps(4)
sig(5)=sig(5)+gc*eps(5)
sig(6)=sig(6)+gc*eps(6)
c
elseif (etype.eq.'beam' ) then
c
q1      =cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
q3      =q1+2.0*g
gc      =capa*g
deti    =1./(q3*q3-q1*q1)
c22i    = q3*deti
c23i    =-q1*deti
fac     =(c22i+c23i)*q1
eps(2)=-eps(1)*fac-sig(2)*c22i-sig(3)*c23i
eps(3)=-eps(1)*fac-sig(2)*c23i-sig(3)*c22i
davg    =(-eps(1)-eps(2)-eps(3))/3.
p       =-davg*cm(1)/((1.-2.*cm(2)))
sig(1)=sig(1)+p+g2*(eps(1)+davg)
sig(2)=0.0
sig(3)=0.0
sig(4)=sig(4)+gc*eps(4)
sig(5)=0.0
sig(6)=sig(6)+gc*eps(6)
elseif (etype.eq.'tbeam' ) then
c
q1      =cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
q3      =q1+2.0*g
deti    =1./(q3*q3-q1*q1)
c22i    = q3*deti
c23i    =-q1*deti
fac     =(c22i+c23i)*q1
eps(2)=-eps(1)*fac
eps(3)=-eps(1)*fac
davg    =(-eps(1)-eps(2)-eps(3))/3.
p       =-davg*cm(1)/((1.-2.*cm(2)))
sig(1)=sig(1)+p+g2*(eps(1)+davg)
sig(2)=0.0
sig(3)=0.0
endif
c
return
end

```

Additional features

Load curves and tables

If the material of interest should require load curves, for instance a curve defining yield stress as a function of effective plastic strain, the variable `crv` should be used. Each curve defined in the keyword deck is represented by points (x_i, y_i) , $i = 1, \dots, 100$, stored in the array `crv` together with a number defining the increments Δx stored in position 101. To be more precise, the first x value is stored in `crv(1,1,*)`, the first y value in `crv(1,2,*)`, the second x value in `crv(2,1,*)`, the second y value in `crv(2,2,*)`, and so on. The increment Δx is stored in `crv(101,1,*)`. The third index in the `crv` array represents the internal load curve id. There are two ways to extract the values from a load curve from a user defined materials routine.

First, there are two subroutines that can be called from within the user defined routine, these are

```
subroutine crvval(crv,eid,xval,yval,slope)
```

and

```
subroutine crvval_v(crv,eid,xval,yval,slope,lft,llt)
```

where the former routine is used in the scalar context and the latter for vectorized `umat`. The arguments are the following

<code>crv</code>	-	the load curve array
<code>eid</code>	-	external load curve ID, i.e., the load curve ID taken from the keyword deck
<code>xval</code>	-	abscissa value
<code>yval</code>	-	ordinate value (output from routine)
<code>slope</code>	-	slope of curve (output from routine)
<code>lft</code>	-	first index of vector
<code>llt</code>	-	final index of vector

where `xval`, `yval` and `slope` are scalars in the scalar routine and vectors of length `n1q` in the vectorized routine. Note that `eid` should be passed as float.

Second, for efficiency considerations the user may extract values on his/her own. The following few lines of code shows how to extract the ordinate value y at the abscissa x for a curve with external curve id (in the keyword deck) given by `crvid_ext`.

```
integer crvid_int
c
c obtain internal curve id
c
c crvid_int=lcids(nint(crvid_ext))
c
c proceed if curve id is valid
c
c if (crvid_int.gt.0) then
c
c obtain increment in x and first x value
c
c xinc=crv(101,1,crvid_int)
```

```

        xbgncrv(1,1,crvid_int)
c
c   find interval in which x is situated
c
        ind=aint((x-xbgncrv)/xinc)+1
        ind=min(ind,99)
        ind=max(ind,1)
c
c   find slope of that particular segment
c
        slope=(crv(ind+1,2,crvid_int)-crv(ind,2,crvid_int))/
1         (crv(ind+1,1,crvid_int)-crv(ind,1,crvid_int))
c
c   evaluate ordinate value y
c
        y=crv(ind,2,crvid_int)+slope*(x-crv(ind,1,crvid_int))
c
endif

```

For tables, two subroutines are available for extracting values. A scalar version is

```
subroutine tabval(crv,eid,dxval,yval,dslope,xval,slope)
```

and a vector version is

```
subroutine tabval_v(crv,eid,dxval,yval,dslope,lft,llt,xval,slope)
```

where

```

crv - curve array
eid - external curve id, i.e., curve id taken from keyword deck
dxval - abscissa value (x2-axis)
yval - ordinate value (y-axis, output from routine)
dslope- slope of curve (dy/dx2, output from routine)
xval - abscissa value (x1-axis)
slope - slope of curve (dy/dx1, output from routine)
lft - vector index
llt - vector index

```

In the scalar routine, dxval, yval, dslope, xval and slope are all scalars whereas in the vector routine they are vectors of length nlq.

Local coordinate system

If the material model has directional properties, such as composites and anisotropic plasticity models, the local coordinate system option can be invoked. This is done by putting `IORTHO` equal to 1 on the material card. This also requires two additional cards with values for how the coordinate system is formed and updated. When this option is used, all data passed to the constitutive routine `umatXX` or `umatXXv` is in the local system and the transformation back to the global system is done outside this user-defined routine. There is one exception however, see the section on the deformation gradient.

Temperature

For a material with thermal properties, temperatures are made available by putting the flag `ITHERMAL` equal to 1 on the material card. The temperatures in the elements are then available in the `temper` variable for a scalar and `temps` array for the vectorized implementation. For a coupled thermal structural analysis, the thermal problem is solved first and temperatures at the current time are available in the user-defined subroutine. Calculation of dissipated heat in the presence of plastic deformation is taken care of by LS-DYNA and needs not be considered by the user. If the time derivative of the temperature is needed for the stress update, a history variable that contains the temperature in the previous time step should be requested. The time derivative can then be obtained by a backward finite difference estimate.

Failure

It is possible to include failure in the material model, resulting in the deletion of elements that fulfill a certain failure criterion. To accomplish this, the flag `IFAIL` must be set to 1 on the material card. For a scalar implementation, the variable `faile1` is set to `.true.` when a failure criterion is met. For a vectorized implementation, the corresponding entry in the `failels` array is set to `.true.`

Deformation gradient

For some materials, the stresses are not obtained from incremental strains, but are expressed in terms of the deformation gradient \mathbf{F} . This is the case for hyper-elastic(-plastic) materials. To make the deformation gradient available for bricks and shells in the user-defined material subroutines, the variable `IHYPER` on the material card should be set to 1. The deformation gradient components F_{11} , F_{21} , F_{31} , F_{12} , F_{22} , F_{32} , F_{13} , F_{23} and F_{33} can then be found in the history variables array in positions `NHV+1` to `NHV+9`, i.e., the positions coming right after the requested number of history variables.

For shell elements, the components of the deformation gradient are with respect to the co-rotational system for the element currently used. In this case the third row of the deformation gradient, i.e., the components F_{31} , F_{32} and F_{33} , will not be properly updated when entering the user-defined material routine. These components depend on the thickness strain increment which in turn must be determined so that the normal stress in the shell vanishes. For a given thickness strain increment `d3`, these three components, `f31`, `f32` and `f33`, can be determined by calling the subroutine

```
subroutine compute_f3s(f31,f32,f33,d3)
```

for a scalar implementation and

```
subroutine compute_f3(f31,f32,f33,d3,lft,llt)
```

for a vector implementation. The first four arguments are arrays of length `n1q` for the vector routine and scalars for the scalar routine.

For hyper-elastic materials there are push forward operations that can be called from within the user defined subroutines. These are

```
subroutine push_forward_2(sig1,sig2,sig3,sig4,sig5,sig6,
    f11,f21,f31,f12,f22,f32,f13,f23,f33,lft,llt)
```

which performs a push forward operation on the stress tensor, and the corresponding scalar routine

```
subroutine push_forward_2s(sig1,sig2,sig3,sig4,sig5,sig6,
    f11,f21,f31,f12,f22,f32,f13,f23,f33)
```

In the latter subroutine all arguments are scalars whereas the corresponding entries in the vectorized routine are vectors of length `nlq`. The `sig1` to `sig6` are components of the stress tensor and `f11` to `f33` are components of the deformation gradient.

If the local coordinate system option is invoked (`IORTHO=1`), then the deformation gradient is transformed to this local system prior to entering the user-defined material routine according to

$$\bar{F}_{ij} = Q_{ki}^s F_{kj}$$

where Q_{ij}^s refers to a transformation between the current global and material frames. For `IORTHO` equal to 1 one can choose to put `IHYPER` equal to -1 which results in that the deformation gradient is transformed according to

$$\bar{F}_{ij} = F_{ik} Q_{kj}^r$$

where Q_{ij}^r is the transformation between the reference global and material and frames. For this latter option the spatial frame remains the global one so the stresses should be expressed in this frame of reference upon exiting the user defined routines. The suitable choice of `IHYPER` depends on the formulation of the material model.

In the following, a Neo-Hookean material is used as an example of the usage of the deformation gradient in user-defined materials. With λ and μ being the Lamé parameters in the linearized theory, the strain energy density for this material is given by

$$\psi = \frac{1}{2} \lambda (\ln(\det \mathbf{F}))^2 - \mu \ln(\det \mathbf{F}) + \frac{1}{2} \mu (\text{tr}(\mathbf{F}^T \mathbf{F}) - 3)$$

meaning that the Cauchy stress can be expressed as

$$\boldsymbol{\sigma} = \frac{1}{\det \mathbf{F}} (\lambda \ln(\det \mathbf{F}) \mathbf{I} + \mu (\mathbf{F} \mathbf{F}^T - \mathbf{I})).$$

Sample user subroutine 42

```

      subroutine umat42 (cm,eps,sig,epsp,hsv,dt1,capa,
        .           etype,tt,temper,failel,crv)
c*****
c|   livermore software technology corporation   (lstc)   |
c|   -----|
c|   copyright 1987-2003|
c|   all rights reserved|
c*****
c
c   Neo-Hookean material (sample user subroutine)
c
c   Variables
c
c   cm(1)=first material constant, here young's modulus
c   cm(2)=second material constant, here poisson's ratio
c   .
c   .
c   .
c   cm(n)=nth material constant
c
c   eps(1)=local x   strain increment
c   eps(2)=local y   strain increment
c   eps(3)=local z   strain increment
c   eps(4)=local xy  strain increment
c   eps(5)=local yz  strain increment
c   eps(6)=local zx  strain increment
c
c   sig(1)=local x   stress
c   sig(2)=local y   stress
c   sig(3)=local z   stress
c   sig(4)=local xy  stress
c   sig(5)=local yz  stress
c   sig(6)=local zx  stress
c
c   epsp=effective plastic strain
c
c   hsv(1)=1st history variable
c   hsv(2)=2nd history variable
c   .
c   .
c   .
c   .
c   hsv(n)=nth history variable
c
c   dt1=current time step size
c   capa=reduction factor for transverse shear
c   etype:
c     eq."brick" for solid elements
c     eq."shell" for all shell elements
c     eq."beam"  for all beam elements
c     eq."dbeam" for all discrete beam elements
c
c   tt=current problem time.
c   temper=current temperature
c
c   crv=array representation of curves defined in keyword deck
c
c   failel=flag for failure, set to .true. to fail an element
c
c   All transformations into the element local system are

```

```

c     performed prior to entering this subroutine.  Transformations
c     back to the global system are performed after exiting this
c     routine.
c
c     All history variables are initialized to zero in the input
c     phase.  Initialization of history variables to nonzero values
c     may be done during the first call to this subroutine for each
c     element.
c
c     Energy calculations for the energy balance are done
c     outside this subroutine.
c
c     character*(*) etype
c     dimension cm(*),eps(*),sig(*),hsv(*),crv(101,2,*)
c     logical failed
c
c     compute lame parameters
c
c     xlambda=cm(1)*cm(2)/((1.+cm(2))*(1.-2.*cm(2)))
c     xmu=.5*cm(1)/(1.+cm(2))
c
c     if (etype.eq.'brick') then
c
c     no history variables, NHV=0
c     deformation gradient stored in hsv(1),...,hsv(9)
c
c     compute jacobian
c
c     1     detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
c     2     -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
c         +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c
c     compute left cauchy-green tensor
c
c     b1=hsv(1)*hsv(1)+hsv(4)*hsv(4)+hsv(7)*hsv(7)
c     b2=hsv(2)*hsv(2)+hsv(5)*hsv(5)+hsv(8)*hsv(8)
c     b3=hsv(3)*hsv(3)+hsv(6)*hsv(6)+hsv(9)*hsv(9)
c     b4=hsv(1)*hsv(2)+hsv(4)*hsv(5)+hsv(7)*hsv(8)
c     b5=hsv(2)*hsv(3)+hsv(5)*hsv(6)+hsv(8)*hsv(9)
c     b6=hsv(1)*hsv(3)+hsv(4)*hsv(6)+hsv(7)*hsv(9)
c
c     compute cauchy stress
c
c     detfinv=1./detf
c     dmux=xmu-xlambda*log(detf)
c     sig(1)=detfinv*(xmu*b1-dmux)
c     sig(2)=detfinv*(xmu*b2-dmux)
c     sig(3)=detfinv*(xmu*b3-dmux)
c     sig(4)=detfinv*xmu*b4
c     sig(5)=detfinv*xmu*b5
c     sig(6)=detfinv*xmu*b6
c
c     else if (etype.eq.'shell') then
c
c     no history variables, NHV=0
c     deformation gradient stored in hsv(1),...,hsv(9)
c
c     compute part of left cauchy-green tensor
c     independent of thickness strain increment
c
c     b1=hsv(1)*hsv(1)+hsv(4)*hsv(4)+hsv(7)*hsv(7)
c     b2=hsv(2)*hsv(2)+hsv(5)*hsv(5)+hsv(8)*hsv(8)
c     b4=hsv(1)*hsv(2)+hsv(4)*hsv(5)+hsv(7)*hsv(8)

```

```

c
c   secant iterations for zero normal stress
c
c       do iter=1,5
c
c           if (iter.eq.1) then
c
c               first thickness strain increment initial guess
c               assuming Poisson's ratio different from zero
c
c                   eps(3)=-xlambda*(eps(1)+eps(2))/(xlambda+2.*xmu)
c
c           else if (iter.eq.2) then
c
c               second thickness strain increment initial guess
c
c                   sigold=sig(3)
c                   epsold=eps(3)
c                   eps(3)=0.
c
c           else if (abs(sig(3)-sigold).gt.0.0) then
c
c               secant update of thickness strain increment
c
c                   deps=- (eps(3)-epsold)/(sig(3)-sigold)*sig(3)
c                   sigold=sig(3)
c                   epsold=eps(3)
c                   eps(3)=eps(3)+deps
c
c           endif
c
c       compute last row of deformation gradient
c
c           call compute_f3s(hsv(3),hsv(6),hsv(9),eps(3))
c
c       compute jacobian
c
c           detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
1           -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
2           +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c
c       compute normal component of left cauchy-green tensor
c
c           b3=hsv(3)*hsv(3)+hsv(6)*hsv(6)+hsv(9)*hsv(9)
c
c       compute normal stress
c
c           detfinv=1./detf
c           dmu=xmu-xlambda*log(detf)
c           sig(1)=detfinv*(xmu*b1-dmu)
c           sig(2)=detfinv*(xmu*b2-dmu)
c           sig(3)=detfinv*(xmu*b3-dmu)
c           sig(4)=detfinv*xmu*b4
c
c       exit loop if normal stress is sufficiently small
c
c           if (abs(sig(3)).le.1.e-4*
1           (abs(sig(1))+abs(sig(2))+abs(sig(4)))) goto 10
c
c       enddo
c
c       compute remaining components of left cauchy-green tensor
c

```

```

10      b5=hsv(2)*hsv(3)+hsv(5)*hsv(6)+hsv(8)*hsv(9)
        b6=hsv(1)*hsv(3)+hsv(4)*hsv(6)+hsv(7)*hsv(9)
c
c      compute remaining stress components
c
        sig(5)=detfinv*xmu*b5
        sig(6)=detfinv*xmu*b6
c
        else
c
c      material model only available for bricks and shells
c
        write (*,20) etype
        write (13,20) etype
        write (59,20) etype
        call adios(2)
c
        endif
c
20      format(/
1         '*** error element type ',a,' can not be'
2         'run with the current material model. ***')
c
        return
        end

```

Implicit analysis

For brick and shell elements, a user-defined material model can also be run with implicit analysis. When an implicit analysis is requested in the input keyword deck, LS-DYNA calls the subroutine `urtanh` for bricks and `urtans` for shells with appropriate input data for the calculation of the material tangent modulus. For a scalar implementation, this routine in turn calls `utanXX` with all necessary input parameters including

`es(6,6)` – material tangent modulus

Again, `XX` is the number that matches `MT` on the material card. For a vectorized implementation, the routine `utanXXv` is called, this time with the corresponding vector block

`dsave(nlq,6,6)` – material tangent modulus

This subroutine builds the tangent modulus to be used for assembling the tangent stiffness matrix and must be provided by the user. This matrix is equal to the zero matrix when entering the user-defined routine, it must be symmetric and if the local coordinate system option is invoked for bricks, then it should be expressed in this local system. For shell elements, it should be expressed in the co-rotational system defined for the current shell element. All transformations back to the global system are made after exiting the user-defined routine.

If the material is hyper-elastic, there are push forward operations of tangent modulus tensor available in

```

subroutine push_forward_4(dsave,
.      f11, f21, f31, f12, f22, f32, f13, f23, f33, lft, llt)

```

which performs a push forward operation on the tangent modulus tensor, and the corresponding scalar routine

```
subroutine push_forward_4s(es,
.      f11,f21,f31,f12,f22,f32,f13,f23,f33)
```

In the latter subroutine all arguments are scalars whereas the corresponding entries in the vectorized routine are vectors of length $n1q$. The $f11$ to $f33$ are components of the deformation gradient.

The following sample user subroutine illustrates how to implement the tangent stiffness modulus for the Neo-Hookean material above. The material tangent modulus is for this material given by

$$\mathbf{C} = \frac{1}{\det \mathbf{F}} (\lambda \mathbf{I} \otimes \mathbf{I} + 2(\mu - \lambda \ln(\det \mathbf{F})) \mathbf{I}).$$

Sample user subroutine 42, tangent modulus

```
subroutine utan42(cm,eps,sig,epsp,hsv,dt1,capa,
.      etype,tt,temper,es,crv)
c*****
c|   livermore software technology corporation   (lstc)   |
c|   -----
c|   copyright 1987-1999
c|   all rights reserved
c*****
c
c   Neo-Hookean material tangent modulus (sample user subroutine)
c
c   Variables
c
c   cm(1)=first material constant, here young's modulus
c   cm(2)=second material constant, here poisson's ratio
c   .
c   .
c   .
c   cm(n)=nth material constant
c
c   eps(1)=local x   strain increment
c   eps(2)=local y   strain increment
c   eps(3)=local z   strain increment
c   eps(4)=local xy  strain increment
c   eps(5)=local yz  strain increment
c   eps(6)=local zx  strain increment
c
c   sig(1)=local x   stress
c   sig(2)=local y   stress
c   sig(3)=local z   stress
c   sig(4)=local xy  stress
c   sig(5)=local yz  stress
c   sig(6)=local zx  stress
c
c   epsp=effective plastic strain
c
c   hsv(1)=1st history variable
c   hsv(2)=2nd history variable
c   .
c   .
c   .
```

```

c
c      .
c      hsv(n)=nth history variable
c
c      dt1=current time step size
c      capa=reduction factor for transverse shear
c      etype:
c          eq."brick" for solid elements
c          eq."shell" for all shell elements
c          eq."beam"  for all beam elements
c          eq."dbeam" for all discrete beam elements
c
c      tt=current problem time.
c
c      temper=current temperature
c
c      es=material tangent modulus
c
c      crv=array representation of curves in keyword deck
c
c      The material tangent modulus is set to 0 prior to entering
c      this routine. It should be expressed in the local system
c      upon exiting this routine. All transformations back to the
c      global system is made outside this routine.
c
c      character*(*) etype
c      dimension cm(*),eps(*),sig(*),hsv(*),crv(101,2,*)
c      dimension es(6,*)
c
c      no history variables, NHV=0
c      deformation gradient stored in hsv(1),...,hsv(9)
c
c      compute jacobian
c
c      detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
1      -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
2      +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c
c      compute lame parameters
c
c      xlambda=cm(1)*cm(2)/((1.+cm(2))*(1.-2.*cm(2)))
c      xmu=.5*cm(1)/(1.+cm(2))
c
c      compute tangent stiffness
c      same for both shells and bricks
c
c      detfinv=1./detf
c      dmu=xmu-xlambda*log(detf)
c      es(1,1)=detfinv*(xlambda+2.*dmu)
c      es(2,2)=detfinv*(xlambda+2.*dmu)
c      es(3,3)=detfinv*(xlambda+2.*dmu)
c      es(4,4)=detfinv*dmu
c      es(5,5)=detfinv*dmu
c      es(6,6)=detfinv*dmu
c      es(2,1)=detfinv*xlambda
c      es(3,2)=detfinv*xlambda
c      es(3,1)=detfinv*xlambda
c      es(1,2)=es(2,1)
c      es(2,3)=es(3,2)
c      es(1,3)=es(3,1)
c
c      return
c      end

```

Post-processing a user-defined material

Post-processing a user-defined material is very similar to post-processing a regular LS-DYNA material. There are however some things that are worth being stressed, all dealing with how to post-process history variables.

First, the effective plastic strain is always written to the d3plot database and thus need not be requested by the user. It is in LS-PRE/POST treated just as it is for any other LS-DYNA material.

The number of additional history variables written to the d3plot database must be requested as the parameter `NEIPH` (for bricks) or `NEIPS` (for shells) on `*DATABASE_EXTENT_BINARY`. For instance, if `NEIPH` (`NEIPS`) equals 2 the first two history variables in the history variables array are obtained as history var#1 and history var#2 in the d3plot database. By putting `NEIPH` (`NEIPS`) equal to `NHV`, all history variables are written to the d3plot database. Furthermore, if the material uses the deformation gradient (`IHYPER=1`) an additional 9 variables must be requested to make this available for post-processing, i.e., put `NEIPH` (`NEIPS`) equal to `NHV+9`. This makes the deformation gradient available in the d3plot database as history variables `NHV+1` to `NHV+9`, note however that for shells it is expressed in the co-rotational system. If the local coordinate system option (`IORTHO=1`) is used, then the deformation gradient is expressed in this local system. To make the deformation gradient in the global system for bricks and co-rotational system for shells available and stored as history variables `NHV+10` to `NHV+18`, `NEIPH` (`NEIPS`) is put equal to `NHV+9+9 (=NHV+18)`.

APPENDIX B: User Defined Equation of State

The user can supply his/her own subroutines defining equation of state (EOS) models in LS-DYNA. To invoke a user-defined EOS, one must

1. Write a user EOS subroutine that is called by the LS-DYNA user EOS interface.
2. Create a custom executable which includes the EOS subroutine.
3. Invoke that subroutine by defining a part in the keyword input deck that uses `*EOS_USER_DEFINED` with the appropriate input parameters.

All subroutines, including the interface, for the user-defined materials are collected in the file `dyn21.F` (Unix/Linux) or `lsdyna.f` (Windows). Up to ten user subroutines can currently be implemented simultaneously to update the pressure in solids and shells. This text serves as an introductory guide to implementing such a model. Note that names of variables and subroutines below may differ from the actual ones depending on platform and current version of LS-DYNA.

General overview

When the keyword `*EOS_USER_DEFINED` is defined for a part in the keyword deck, LS-DYNA calls the subroutine `ueoslib` with the appropriate input data for the EOS update. This subroutine is called twice for each integration point in each element. The first call requires the EOS to calculate the bulk modulus, and the second updates the pressure and internal energy. In these routines, which may be modified by the user if necessary, the following data structures are initialized for the purpose of being supplied to a specific *scalar* material subroutine.

`iflag` - =0 for calculating the bulk modulus, =1 for the pressure and energy update
`cb` - bulk modulus
`pnew` - the new pressure
`rho0` - reference density
`hist` - array of user-defined history variables `NHV` in length
`specen` - internal energy per unit reference volume
`df` - volume ratio, V/V_0
`v0` - the initial volume.
`dvol` - volume increment
`pc` - pressure cut-off

If the *vectorization* flag is active (`IVECT=1`) on the EOS card, variables are, in general, stored in vector blocks of length `nlq`, with vector indices ranging from `1ft` to `1lt`, which allows for a more efficient execution of the EOS routine. As an example, the data structures mentioned above for the vectorized case are

`cb(nlq)` - bulk modulus
`pnew(nlq)` - the new pressure
`hist(nlq,*)` - array of user-defined history variables with `NHV` columns
`specen(nlq)` - internal energy per unit reference volume
`df(nlq)` - volume ratio, V/V_0
`v0(nlq)` - the initial volume
`dvol(nlq)` - volume increment
`pc(nlq)` - pressure cut-off

The value of `nlq` is set as a parameter in the include file `nlqparm`, included at the top of the subroutine, and varies between machines and operating systems. Each entry in a vector block is associated with an integration point in the finite element mesh. The number of entries in the history variables array (indicated by * in the above) matches the number of history variables requested on the material card (`NHV`). All history variables are initially zero and are initialized within the EOS on the first time step, when the logical variable `first`, passed through the argument list, is `.TRUE`. Furthermore, all user-defined EOS models require a bulk modulus, `cb`, for transmitting boundaries, contact interfaces, rigid body constraints, and time step calculations. In addition to the variables mentioned above, the following data can be supplied to the user material routines, regardless of whether vectorization is used or not.

```
eosp(*) - array of material constants from the input file
tt - current time
crv(101,2,*) - array representation of curves defined in the keyword deck.
```

A specific material routine, `ueosXXs` in the scalar case or `ueosXXv` in the vector case, is now called with any necessary parameters of the ones above, and possibly others as well. The use of curves is discussed in Appendix A. The letters `xx` stands for a number between 21 and 30 and matches the number `EOST` on the EOS card. This subroutine is written by the user, and should calculate the bulk modulus when `iflag=0`, and update the pressure, internal energy and history variables to the current time when `iflag=1`. During the input phase, the EOS is called with `iflag=-1` to permit the initialization of any desired constants in `eosp`. Although fewer than 48 constants may be read into `eosp` during the input, the user may use all 48 within the EOS subroutines.

A sample scalar user subroutine for a Gruneisen EOS is provided below and it is immediately followed by its vector counterpart.

Sample user subroutine 21

```
subroutine ueos21s(iflag,cb,pnew,hist,rho0,eosp,specen,
&                df,dvol,v0,pc,dt,tt,crv,first)
include 'nlqparm'
c
c*** example scalar user implementation of the Gruneisen EOS
c
c*** variables
c     iflag ----- =0 calculate bulk modulus
c                   =1 update pressure and energy
c     cb ----- bulk modulus
c     pnew ----- new pressure
c     hist ----- history variables
c     rho0 ----- reference density
c     eosp ----- EOS constants
c     specen ---- energy/reference volume
c     df ----- volume ratio, v/v0 = rho0/rho
c     dvol ----- change in volume over time step
c     v0 ----- reference volume
c     pc ----- pressure cut-off
c     dt ----- time step size
c     tt ----- current time
c     crv ----- curve array
c     first ----- logical .true. for tt,crv,first time step
```

```

c                                     (for initialization of the history variables)
c
c   logical first
c
c   dimension hist(*),eosp(*),crv(101,2,*)
c
c   c =eosp(1)
c   s1 =eosp(2)
c   s2 =eosp(3)
c   s3 =eosp(4)
c   g0 =eosp(5)
c   sa =eosp(6)
c   s11=s1-1.
c   s22=2.*s2
c   s33=3.*s3
c   s32=2.*s3
c   sad2=.5*sa
c   g0d2=1.-.5*g0
c   roc2=rho0*c**2
c
c*** calculate the bulk modulus for the EOS contribution to the sound speed
c   if (iflag.eq.0) then
c     xmu=1.0/df-1.
c     dfmu=df*xmu
c     facp=.5*(1.+sign(1.,xmu))
c     facn=1.-facp
c     xnum=1.+xmu*(+g0d2-sad2*xmu)
c     xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
c     tmp=facp/(xdem*xdem)
c     a=roc2*xmu*(facn+tmp*xnum)
c     b=g0+sa*xmu
c     pnum=roc2*(facn+facp*(xnum+xmu*(g0d2-sa*xmu)))
c     pden=2.*xdem*(-s11+dfmu*(-s22+dfmu*(s2-s33+s32*dfmu)))
c     cb=pnum*(facn+tmp)-tmp*a*pden+sa*specen+
c     &      b*df**2*max(pc,(a+b*specen))
c
c*** update the pressure and internal energy
c   else
c     xmu=1.0/df-1.
c     dfmu=df*xmu
c     facp=.5*(1.+sign(1.,xmu))
c     facn=1.-facp
c     xnum=1.+xmu*(+g0d2-sad2*xmu)
c     xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
c     tmp=facp/(xdem*xdem)
c     a=roc2*xmu*(facn+tmp*xnum)
c     b=g0+sa*xmu
c     dvov0=0.5*dvol/v0
c     denom=1.+ b*dvov0
c     pnew=(a+specen*b)/max(1.e-6,denom)
c     pnew=max(pnew,pc)
c     specen=specen-pnew*dvov0
c   endif
c
c   return
c   end
c   subroutine ueos21v(lft,llt,iflag,cb,pnew,hist,rho0,eosp,specen,
c   &      df,dvol,v0,pc,dt,tt,crv,first)
c   include 'nlqparm'
c
c*** example vectorized user implementation of the Gruneisen EOS
c
c*** variables

```

```

c      lft,llt --- tt,crv,first and last indices into arrays
c      iflag ----- =0 calculate bulk modulus
c                  =1 update pressure and energy
c      cb ----- bulk modulus
c      pnw ----- new pressure
c      hist ----- history variables
c      rho0 ----- reference density
c      eosp ----- EOS constants
c      specen ---- energy/reference volume
c      df ----- volume ratio, v/v0 = rho0/rho
c      dvol ----- change in volume over time step
c      v0 ----- reference volume
c      pc ----- pressure cut-off
c      dt ----- time step size
c      tt ----- current time
c      crv ----- curve array
c      first ----- logical .true. for tt,crv,first time step
c                  (for initialization of the history variables)
c
c      logical first
c
c      dimension cb(*),pnw(*),hist(nlq,*),eosp(*),
&              specen(*),df(*),dvol(*),pc(*),v0(*)
c
c      =eosp(1)
s1 =eosp(2)
s2 =eosp(3)
s3 =eosp(4)
g0 =eosp(5)
sa =eosp(6)
s11=s1-1.
s22=2.*s2
s33=3.*s3
s32=2.*s3
sad2=.5*sa
g0d2=1.-.5*g0
roc2=rho0*c**2
c
c*** calculate the bulk modulus for the EOS contribution to the sound speed
if (iflag.eq.0) then
  do i=lft,llt
    xmu=1.0/df(i)-1.
    dfmu=df(i)*xmu
    facp=.5*(1.+sign(1.,xmu))
    facn=1.-facp
    xnum=1.+xmu*(+g0d2-sad2*xmu)
    xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
    tmp=facp/(xdem*xdem)
    a=roc2*xmu*(facn+tmp*xnum)
    b=g0+sa*xmu
    pnum=roc2*(facn+facp*(xnum+xmu*(g0d2-sa*xmu)))
    pden=2.*xdem*(-s11 +dfmu*(-s22+dfmu*(s2-s33+s32*dfmu)))
    cb(i)=pnum*(facn+tmp)-tmp*a*pden+sa*specen(i)+
&          b*df(i)**2*max(pc(i),(a+b*specen(i)))
  enddo
c
c*** update the pressure and internal energy
else
  do i=lft,llt
    xmu=1.0/df(i)-1.
    dfmu=df(i)*xmu
    facp=.5*(1.+sign(1.,xmu))
    facn=1.-facp

```

```

xnum=1.+xmu*(+g0d2-sad2*xmu)
xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
tmp=facp/(xdem*xdem)
a=roc2*xmu*(facn+tmp*xnum)
b=g0+sa*xmu
dvov0=0.5*dvol(i)/v0(i)
denom=1.+b*dvov0
pnew(i)=(a+specen(i)*b)/max(1.e-6,denom)
pnew(i)=max(pnew(i),pc(i))
specen(i)=specen(i)-pnew(i)*dvov0
enddo
endif
c
return
end

```

The Gruneisen EOS implemented in the example subroutines has the same form as *EOS_GRUNEISEN, EOS Form 4. Its update of the pressure and the internal energy are typical for an EOS that is linear in the internal energy,

$$P = A(\rho) + B(\rho)E$$

where A and B correspond to the variables a and b in the example subroutines, and E is `specen`. Integrating the energy equation with the trapezoidal rule gives

$$E^{n+1} = E^n + \frac{1}{2}(\sigma'^n + \sigma'^{n+1})\Delta\varepsilon - \frac{1}{2}(P^n + q^n + P^{n+1} + q^{n+1})\frac{\Delta V}{V_0}$$

where the superscripts refer to the time step, ΔV is the change in the volume associated with the Gauss point and V_0 is the reference volume. Collecting all the energy contributions on the right hand side except for the contribution from the new pressure gives a simple linear relationship between the new internal energy and pressure,

$$E^{n+1} = \tilde{E} - \frac{P^{n+1}\Delta V}{2V_0}.$$

The value of `specen` passed to `ueosXX` for the pressure and energy update corresponds to \tilde{E} . Substituting this relation into the EOS and solving for the new pressure gives

$$P^{n+1} = \frac{A(\rho^{n+1}) + B(\rho^{n+1})\tilde{E}}{1 + \frac{B\Delta V}{2V_0}}.$$

The final update of the new energy is calculated using the new pressure. For a more general EOS, the nonlinear equation in the new pressure,

$$P^{n+1} = P(\rho^{n+1}, \tilde{E} - \frac{P^{n+1}\Delta V}{2V_0})$$

is solved iteratively using Newton iteration or successive substitution.

The pressure cut-off, p_c , is used to limited the amount of pressure that can be generated by tensile loading, $p_{new} = \max(p_{new}, p_c)$. Its value is usually specified in the *MAT input, e.g., *MAT_JOHNSON_COOK. It is not enforced outside of the EOS subroutines, and it is up to the user to determine whether or not to enforce the pressure cut-off in ueosXX. If the user does enforce it, the pressure cut-off should be applied before the final update to the internal energy otherwise the energy will be incorrect.

Many of the calculations performed to calculate the bulk modulus are the same as those for updating the pressure and energy. Since the bulk modulus calculation always precedes the pressure update, the values may be saved in a common block during the bulk modulus calculation to reduce the cost of the pressure update. The arrays used to store the values in the vectorized subroutines should be dimensioned by n1q.

One of the most common errors in implementing an EOS from a paper or book is the use of the wrong internal energy. There are three internal energies in common use: the energy per unit mass, e_M , the energy per unit current volume, e_v , and the energy per unit reference volume, E . LS-DYNA always uses the energy per unit reference volume. Some useful relations for converting between EOS in the literature and the variables in LS-DYNA are

$$e_v = E \frac{V_0}{V} = \text{specen} / \text{df}$$

$$e_M = E \frac{V_0}{M} = \text{specen} / \text{rho0}$$

$$\rho = \rho_0 \frac{V_0}{V} = \text{rho0} / \text{df}$$

APPENDIX C: User Defined Element Interface for Solids and Shells

In this appendix the user-defined element interface for solids and shells is described. The interface can accommodate either an integrated or a resultant element. For the integrated element, the user needs to supply two matrices defining the kinematical properties of the element, and choose between using standard LS-DYNA hourglass stabilization, a user-defined stabilization, or no stabilization when zero energy modes are not present. The number and location of the integration points is arbitrary, i.e., user-defined. For the resultant/discrete element formulations, the force and stiffness assembly must also be implemented. History variables can be associated with the user defined elements. If desired, the element may utilize more than the conventional 3 (for bricks) and 6 (for shells) degrees-of-freedom per node.

USER DEFINED ELEMENTS

The user element is implemented according to how standard elements are implemented in LS-DYNA with the exception that two user routines are called for setting up the matrices of interest. In the end, the gradient-displacement matrix B_{ijkK} is constructed with the property that

$$B_{ijkK} u_{kK} = \frac{\partial v_i}{\partial x_j}$$

where u_{kK} is the vector of velocity nodal degrees of freedom and the right hand side is the velocity gradient. Moreover, the determinant J of the jacobian matrix determining the mapping from the isoparametric to physical domain is needed for numerical integration. From these expressions, the strains are determined as the symmetric part of the velocity gradient and the spin as the corresponding antisymmetric part. The stresses are evaluated using the constitutive models in LS-DYNA and the internal forces are obtained from

$$f_{kK} = \int \sigma_{ij} B_{ijkK} dV$$

where σ_{ij} are the stresses. Furthermore, the geometric and material tangent stiffnesses are obtained through

$$K_{iljJ}^{mat} = \int C_{klmn} B_{klij} B_{mnjJ} dV$$

and

$$K_{iljJ}^{geo} = \int \sigma_{mn} B_{kmil} B_{knjJ} dV$$

where C_{klmn} is the tangent modulus for the material. The integrals are evaluated using user-defined quadrature using the determinant J .

For user-defined hourglass control, the user must provide the corresponding internal force and stiffness contribution in a separate user routine. There is also the option to provide the force and stiffness matrix directly for the entire element.

To invoke a user-defined element one must do the following:

1. Write user element subroutine that defines the kinematics or kinetics of the element.
2. Create a custom executable which includes these subroutines.
3. Invoke the element by specifying this on the corresponding *SECTION card.

The dummy subroutines for the user defined elements are provided to the user in a FORTRAN source file for you to modify along with the necessary object files to compile a new executable. Contact LSTC or your local distributor for information about how to obtain these files as well as what compiler/version to use for your specific platform. Up to five user elements can simultaneously be used for bricks and shells (i.e. a total of ten). This text serves as an introductory guide on how to implement such an element.

General overview

To activate a user-defined element, it is necessary to set `ELFORM` to a number between 101 and 105 on the *SECTION definition. By doing so, the kinematics of the elements in the corresponding part will be determined from calling the subroutine

```
subroutine uXXX_bYYY(bmtrx,gmtrx,gjac,...
dimension bmtrx(nlq,3,3,*),gmtrx(nlq,3,3),gjac(*)
```

where `XXX` is substituted for `shl` for a shell-section and `slid` for a solid-section and `YYY` is the number specified in position `ELFORM`. Depending on the choice of `ITAJ` in the input, the user should set the matrices as follows.

If `ITAJ=0`, then set the isoparametric gradient-displacement matrix, represented by the array `bmtrx`, and jacobian matrix, represented by the array `gmtrx`. Here, the first index corresponds to the LS-DYNA block loop index where `nlq` is the block size. For a more convenient notation in the following, we assign a correspondence between the arrays `gmtrx` and `bmtrx` in the subroutines to matrices/tensors as follows

$$\begin{aligned} \text{gmtrx}(*,i,j) & \quad - \quad g_{ij} \\ \text{bmtrx}(*,i,j,k) & \quad - \quad b_{ijk} \end{aligned}$$

These matrices should be determined so that at the current integration point:

$$\begin{aligned} g_{ij} &= \frac{\partial x_i}{\partial \xi_j} \\ b_{ijk} u_k &= \frac{\partial v_i}{\partial \xi_j} \Delta t \end{aligned}$$

In the above, summation over repeated indices is assumed. We use the following notation

$$x_i(\xi_1, \xi_2, \xi_3, t) = \textit{i} \textit{th} \textit{ component of the current position vector} \\ \textit{at isoparametric coordinate } (\xi_1, \xi_2, \xi_3) \textit{ and time } t.$$

$v_i(\xi_1, \xi_2, \xi_3, t) =$ i th component of the velocity vector
at isoparametric coordinate (ξ_1, ξ_2, ξ_3) and time t .

$\Delta t =$ current time step

$u_k =$ k th component of the generalized local displacements

$\xi_i =$ i th component of the isoparametric coordinate ranging from -1 to 1,

For shells, there is an option to get all variables in either the LS-DYNA local coordinate system ($ILOC=0$) or in the global coordinate system ($ILOC=1$). The matrix for the coordinate system transformation is also passed to the user routines where the columns represent the local unit base vectors. The resulting strains must always be in the local coordinate system for the constitutive evaluations. For no extra degrees of freedom (see below), the index k in the displacement expression is determined from the formula

$$k = n(m-1) + d$$

where $n = 3$ if only translational degrees of freedom are present (typical for solids) and $n = 6$ if rotational degrees of freedom are present (typical for shells), m is the local node number ($m = 1, 2, \dots$) and d is the degree of freedom. The translational degrees of freedom correspond to $d \leq 3$ and the rotational degrees of freedom to $4 \leq d \leq 6$.

If $ITAJ=1$, the user should set up the physical gradient-displacement matrix, represented by the array `bmtx`, and jacobian determinant, represented by the array `gjac`. Again, we assign a correspondence between the arrays `gjac` and `bmtx` in the subroutines to matrices/tensors as follows

$$\begin{array}{ll} \text{gjac} (*) & - \quad J \\ \text{bmtx} (*, i, j, k) & - \quad b_{ijk} \end{array}$$

These matrices should be determined so that at the current integration point:

$$\begin{aligned} J &= \det \frac{\partial x_i}{\partial \xi_j} \\ b_{ijk} u_k &= \frac{\partial v_i}{\partial x_j} \Delta t \end{aligned}$$

To be able to set up these matrices, a set of additional auxiliary variables are passed to the user element subroutines. These include the isoparametric coordinate, the element thickness, and the shape function values, and derivatives. Again, for shells these are expressed in either the local or global coordinate system depending on the user's choice. For more information on these variables, the user is referred to the comments in the subroutines.

The integrated elements can use up to a total of 100 integration points (in the plane for shells) at arbitrary locations. These must be specified in terms of isoparametric coordinates and weights following the first of the user-defined cards in the *SECTION_... input. The isoparametric coordinates should range from -1 to 1 and the weights should sum up to 4 for shells and 8 for solids.

It may be necessary to incorporate hourglass stabilization to suppress zero energy modes, this is done by putting IHGF.GT.0 in the input. For IHGF.EQ.1, the LS-DYNA hourglass routines are used automatically and for IHGF.EQ.2 or IHGF.EQ.3 the user must provide hourglass force and stiffness in a specific user-defined routine. If IHGF.EQ.3, physical stabilization becomes available since the resultant material tangent moduli are passed to the hourglass routine to provide the current membrane, bending and coupled membrane-bending stiffness of the material. With C_{ij} denoting the material tangent modulus in matrix form, the resultant tangent moduli are expressed as

$$\begin{aligned}\bar{C}_{ij}^0 &= \int C_{ij} dV && \text{(membrane)} \\ \bar{C}_{ij}^1 &= \int z^1 C_{ij} dV && \text{(membrane-bending)} \\ \bar{C}_{ij}^2 &= \int z^2 C_{ij} dV && \text{(bending)}\end{aligned}$$

where z is the thickness coordinate for shells. For solids, only the first resultant modulus is passed. In this case the array has 21 entries that correspond to the subdiagonal terms of the 6 by 6 resultant matrix. For the matrix index (i, j) in the material tangent modulus matrix, where $i \geq j$, the index I of the array passed to the routine is given by

$$I = i(i-1)/2 + j$$

i.e., the subdiagonal terms are stored row-wise in the array. For shells, all three moduli are passed in the local coordinate system where each array has 15 entries corresponding to the subdiagonal terms of the 5 by 5 resultant matrices. The through thickness direction is here eliminated from the plane stress assumption. The formula for the array indices transformation above holds. This subroutine is called

```
subroutine uXXX_eYYY(force,stiff,ndtot,...
dimension force(nlq,*),stiff(nlq,ndtot,*))
```

where again XXX and YYY should be substituted as described for the other subroutines in the above. The variables in the subroutine corresponds to the force and stiffness as

```
force(*,i)      -    $f_i$ 
stiff(*,i,j)    -    $K_{ij}$ 
```

where the indices corresponds to node and degree of freedom numbers exactly as for the displacements. For shells the force and stiffness is set up in the local element system (ILOCF=0) or global system (ILOCF=1). The variable $ndtot$ is the total number of degrees of freedom for the element. Passed to this subroutine are also the property parameters and history variables

associated with the element. The values of the property parameters are defined in the input of a user-defined element. No more than 40 property parameters and 100 history variables can be used for each user-defined element. The history variables must be updated in this routine by the user.

Resultant/discrete elements

By putting $NIP(P)$ equal to 0 in the input, a resultant/discrete element is assumed understood. For this option (which is incompatible with $IHGF.GT.0$) the user must provide force and stiffness in the same user-defined routine as for the user-defined hourglass control. This means that no material routine is called to update stresses and history variables, but all that must be accounted for in the user element routine. Nevertheless, the user should define $*MAT_ELASTIC$ as the material for the corresponding part with suitable values of the Young's modulus and Poisson's ratio. These material properties are used for time step calculations and for contact stiffnesses. Again, property parameters and history variables are passed to the routine, and for shells also the thicknesses of the elements. For the shell thickness update option ($ISTUPD.GT.0$ on $*CONTROL_SHELL$) it is up to the user to update the thicknesses in this routine.

In what follows, a short description of the additional features associated with the user elements is given.

Nodal fiber vectors

If a user-defined shell element formulation uses the nodal fiber vectors, this must be specified by putting $IUNF=1$ on the $*SECTION_SHELL$ card. With this option the nodal fiber vectors are processed in the element routines and can be used as input for determining the b_{ijk} , g_{ij}/J , f_i and K_{ij} tensors/matrices in the user routines. If not, it is assumed that the fiber direction is normal to the plane of the shell at all times. These are expressed in either the local or global system depending on the user's choice. See comments in the subroutines for more information.

Extra degrees of freedom

Exotic element formulations may require extra degrees-of-freedom per node besides the translational (and rotational) degrees-of-freedom. Currently, up to 3 extra degrees of freedom per node can be used for user-defined elements. To use extra degrees of freedom, a scalar node must be defined for each node that makes up the connectivity of the user element. A scalar node is defined using the keyword $*NODE_SCALAR_VALUE$, in which the user also prescribe initial and boundary conditions associated with the extra variables. The connectivity of the user elements must then be specified with the option $*ELEMENT_SOLID_DOF$ or $*ELEMENT_SHELL_DOF$, where an extra line is used to connect the scalar nodes to the element. As an example:

```

*NODE_SCALAR_VALUE
$   NID          V1          V2          V3      NDF
    11          1.0
    12          1.0
    13          1.0
    14          1.0
                                     1
                                     1
                                     1
                                     1

*ELEMENT_SHELL_DOF
$   EID      PID      N1      N2      N3      N4
    1        1        1       2       3       4
$
                                     NS1     NS2     NS3     NS4
                                     11      12      13      14
    
```

defines an element with one extra degree of freedom. The initial value of the corresponding variable is 1.0 and it is unconstrained. Finally, the user sets the parameter NXDOF on the *SECTION_... card to 1, 2 or 3 depending on how many extra degrees of freedom that should be used in the user-defined element. An array `xdof` containing the current values of these extra variables are passed to the user routines for setting up the correct kinematical properties, see comments in the routines for more information. The formula for the displacement index changes to

$$k = (n + n_{xdof})(m - 1) + d$$

where n_{xdof} is the number of extra degrees of freedom. The extra degrees of freedom for each node corresponds to $n + 1 \leq d \leq n + n_{xdof}$. For dynamic simulations, the mass corresponding to these extra nodes are defined using *ELEMENT_INERTIA or *ELEMENT_MASS.

Related keywords

The following is a list of keywords that apply to the user defined elements

The *SECTION_SHELL card

A third card with accompanying optional cards of the *SECTION_SHELL keyword must be added if the user defined element option is invoked

Card 3 and accompanying cards

Define if and only if ELFORM=101,102,103,104 or 105

Variable	NIPP	NXDOF	IUNF	IHGF	ITAJ	LMC	NHSV	ILOC
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Define NIPP cards according to the following format.

Variable	XI	ETA	WGT					
Type	F	F	F					

Define LMC property parameters using 8 parameters per card.

Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

<u>VARIABLE</u>	<u>DESCRIPTION</u>
ELFORM	... GT.100.AND.LT.106: User-defined shell
NIPP	Number of in-plane integration points for user-defined shell (0 if resultant element)
NXDOP	Number of extra degrees of freedom per node for user-defined shell
IUNF	Flag for using nodal fiber vectors in user-defined shell EQ.0: Nodal fiber vectors are not used. EQ.1: Nodal fiber vectors are used
IHFG	Flag for using hourglass stabilization (NIPP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used
ITAJ	Flag for setting up finite element matrices (NIPP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
ILOC	Local coordinate system option EQ.0: All variables are passed in the local element system EQ.1: All variables are passed in the global system
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

For more information on the variables the user may consult the previous sections in this appendix.

The *SECTION_SOLID card

A second card with accompanying optional cards of the *SECTION_SOLID keyword must be added if the user defined elements option is invoked.

Card 2

Define if and only if ELFORM=101,102,103,104 or 105

Variable	NIP	NXDOP	IHGF	ITAJ	LMC	NHSV		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Define NIP cards according to the following format.

Variable	XI	ETA	ZETA	WGT				
Type	F	F	F	F				

Define LMC property parameters using 8 parameters per card.

Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
ELFORM	... GT.100.AND.LT.106: User-defined solid
NIP	Number of integration points for user-defined solid (0 if resultant element)
NXDOF	Number of extra degrees of freedom per node for user-defined solid
IHFG	Flag for using hourglass stabilization (NIP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used
ITAJ	Flag for setting up finite element matrices (NIP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
ZETA	Third isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

For more information on the variables the user may consult the previous sections in this appendix.

Sample User Shell Element 101 (Belytschko-Tsay shell)

The geometry of the Belytschko-Tsay element in local coordinates can be written

$$x_i = (x_{il} + \frac{t}{2} \xi_3 \delta_{i3}) N_I(\xi_1, \xi_2)$$

$$v_i = (v_{il} + \frac{t}{2} \xi_3 e_{ij3} \omega_{jl}) N_I(\xi_1, \xi_2)$$

where

$x_{il} = i$: th component of coordinate of node I

$v_{il} = i$: th component of translational velocity of node I

$\omega_{jl} = j$: th component of rotational velocity of node I

t = thickness of element

e_{ijk} = permutation tensor

N_I = shape function localized at node I

δ_{i3} = Kronecker delta

Taking the derivative of these expressions with respect to the isoparametric coordinate yields

$$\frac{\partial x_i}{\partial \xi_1} = (x_{i1} + \frac{t}{2} \xi_3 \delta_{i3}) \frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial x_i}{\partial \xi_2} = (x_{i2} + \frac{t}{2} \xi_3 \delta_{i3}) \frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial x_i}{\partial \xi_3} = \frac{t}{2} \delta_{i3}$$

and

$$\frac{\partial v_i}{\partial \xi_1} = (v_{i1} + \frac{t}{2} \xi_3 e_{ij3} \omega_{jl}) \frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial v_i}{\partial \xi_2} = (v_{i2} + \frac{t}{2} \xi_3 e_{ij3} \omega_{jl}) \frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial v_i}{\partial \xi_3} = \frac{t}{2} e_{ij3} \omega_{jl} N_I$$

respectively. Using these expressions the element is implemented as a user-defined shell as follows.

```

      subroutine ushl_b101 (bmtrx, gmtrx, gjac,
1         xi, eta, zeta,
2         n1, n2, n3, n4,
3         dn1dxi, dn2dxi, dn3dxi, dn4dxi,
4         dn1deta, dn2deta, dn3deta, dn4deta,
5         x1, x2, x3, x4, y1, y2, y3, y4, z1, z2, z3, z4,
6         xdof,
7         thick, thck1, thck2, thck3, thck4,
8         fx1, fx2, fx3, fx4,
9         fy1, fy2, fy3, fy4,
.         fz1, fz2, fz3, fz4,
.         gl11, gl21, gl31, gl12, gl22, gl32, gl13, gl23, gl33,
.         lft, llt)
      include 'nlqpam'
c
c      Compute b and g matrix for user-defined shell 101
c
      dimension bmtrx(nlq, 3, 3, *), gmtrx(nlq, 3, 3), gjac(nlq)
      REAL n1, n2, n3, n4
      dimension x1(nlq), x2(nlq), x3(nlq), x4(nlq)
      dimension y1(nlq), y2(nlq), y3(nlq), y4(nlq)
      dimension z1(nlq), z2(nlq), z3(nlq), z4(nlq)
      dimension thick(nlq)
      dimension thck1(nlq), thck2(nlq), thck3(nlq), thck4(nlq)
      dimension xdof(nlq, 8, 3)
      dimension fx1(nlq), fx2(nlq), fx3(nlq), fx4(nlq)
      dimension fy1(nlq), fy2(nlq), fy3(nlq), fy4(nlq)
      dimension fz1(nlq), fz2(nlq), fz3(nlq), fz4(nlq)
      dimension gl11(nlq), gl21(nlq), gl31(nlq),
.         gl12(nlq), gl22(nlq), gl32(nlq),
.         gl13(nlq), gl23(nlq), gl33(nlq)
c
      do i=lft, llt
c

```

```

gmtrx(i,1,1)=
1      x1(i)*dn1dxi+x2(i)*dn2dxi+
2      x3(i)*dn3dxi+x4(i)*dn4dxi
gmtrx(i,2,1)=
1      y1(i)*dn1dxi+y2(i)*dn2dxi+
2      y3(i)*dn3dxi+y4(i)*dn4dxi
gmtrx(i,3,1)=
1      0.
gmtrx(i,1,2)=
1      x1(i)*dn1deta+x2(i)*dn2deta+
2      x3(i)*dn3deta+x4(i)*dn4deta
gmtrx(i,2,2)=
1      y1(i)*dn1deta+y2(i)*dn2deta+
2      y3(i)*dn3deta+y4(i)*dn4deta
gmtrx(i,3,2)=
1      0.
gmtrx(i,1,3)=
1      0.
gmtrx(i,2,3)=
1      0.
gmtrx(i,3,3)=
1      .5*thick(i)
c
coef=.5*thick(i)*zeta
c
bmtrx(i,1,1,1) =dn1dxi
bmtrx(i,1,1,7) =dn2dxi
bmtrx(i,1,1,13)=dn3dxi
bmtrx(i,1,1,19)=dn4dxi
c
bmtrx(i,1,1,5) =coef*dn1dxi
bmtrx(i,1,1,11)=coef*dn2dxi
bmtrx(i,1,1,17)=coef*dn3dxi
bmtrx(i,1,1,23)=coef*dn4dxi
c
bmtrx(i,1,2,1) =dn1deta
bmtrx(i,1,2,7) =dn2deta
bmtrx(i,1,2,13)=dn3deta
bmtrx(i,1,2,19)=dn4deta
c
bmtrx(i,1,2,5) =coef*dn1deta
bmtrx(i,1,2,11)=coef*dn2deta
bmtrx(i,1,2,17)=coef*dn3deta
bmtrx(i,1,2,23)=coef*dn4deta
c
bmtrx(i,2,1,2) =dn1dxi
bmtrx(i,2,1,8) =dn2dxi
bmtrx(i,2,1,14)=dn3dxi
bmtrx(i,2,1,20)=dn4dxi
c
bmtrx(i,2,1,4) =-coef*dn1dxi
bmtrx(i,2,1,10)=-coef*dn2dxi
bmtrx(i,2,1,16)=-coef*dn3dxi
bmtrx(i,2,1,22)=-coef*dn4dxi
c
bmtrx(i,1,3,5) =.5*thick(i)*n1
bmtrx(i,1,3,11)=.5*thick(i)*n2
bmtrx(i,1,3,17)=.5*thick(i)*n3
bmtrx(i,1,3,23)=.5*thick(i)*n4
c
bmtrx(i,3,1,3) =dn1dxi
bmtrx(i,3,1,9) =dn2dxi
bmtrx(i,3,1,15)=dn3dxi

```

```

      bmtrx(i,3,1,21)=dn4dxi
c
      bmtrx(i,2,2,2) =dn1deta
      bmtrx(i,2,2,8) =dn2deta
      bmtrx(i,2,2,14)=dn3deta
      bmtrx(i,2,2,20)=dn4deta
c
      bmtrx(i,2,2,4) =-coef*dn1deta
      bmtrx(i,2,2,10)=-coef*dn2deta
      bmtrx(i,2,2,16)=-coef*dn3deta
      bmtrx(i,2,2,22)=-coef*dn4deta
c
      bmtrx(i,2,3,4) =-.5*thick(i)*n1
      bmtrx(i,2,3,10)=-.5*thick(i)*n2
      bmtrx(i,2,3,16)=-.5*thick(i)*n3
      bmtrx(i,2,3,22)=-.5*thick(i)*n4
c
      bmtrx(i,3,2,3) =dn1deta
      bmtrx(i,3,2,9) =dn2deta
      bmtrx(i,3,2,15)=dn3deta
      bmtrx(i,3,2,21)=dn4deta
c
      enddo
c
      return
      end

```

To use the element for a part the section card can be written as

```

*SECTION SHELL
$   SECID      ELFORM
      1         101
$   T1         T2         T3         T4
$   NIPP      NXDOF      IUNF      IHGF
      1         0         0         1
$   XI        ETA        WGT
      0.        0.        4.

```

Sample User Solid Element 101 (constant stress solid)

The geometry for the constant stress solid is given as

$$x_i = x_{il} N_I(\xi_1, \xi_2)$$

$$v_i = v_{il} N_I(\xi_1, \xi_2)$$

where

$x_{il} = i$: th component of coordinate of node I

$v_{il} = i$: th component of translational velocity of node I

N_I = shape function localized at node I

The matrices necessary for implementing this element as a user-defined solid are derived from the expressions given by

$$\frac{\partial x_i}{\partial \xi_1} = x_{i1} \frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial x_i}{\partial \xi_2} = x_{i2} \frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial x_i}{\partial \xi_3} = x_{i3} \frac{\partial N_I}{\partial \xi_3}$$

and

$$\frac{\partial v_i}{\partial \xi_1} = v_{i1} \frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial v_i}{\partial \xi_2} = v_{i2} \frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial v_i}{\partial \xi_3} = v_{i3} \frac{\partial N_I}{\partial \xi_3}$$

The user element implementation is given by

```

      subroutine usld_b101 (bmtrx, gmtrx, gjac,
1         xi, eta, zeta,
2         n1, n2, n3, n4, n5, n6, n7, n8,
3         dn1dxi, dn2dxi, dn3dxi, dn4dxi,
4         dn5dxi, dn6dxi, dn7dxi, dn8dxi,
5         dn1deta, dn2deta, dn3deta, dn4deta,
6         dn5deta, dn6deta, dn7deta, dn8deta,
7         dn1dzeta, dn2dzeta, dn3dzeta, dn4dzeta,
8         dn5dzeta, dn6dzeta, dn7dzeta, dn8dzeta,
9         x1, x2, x3, x4, x5, x6, x7, x8,
.         y1, y2, y3, y4, y5, y6, y7, y8,
.         z1, z2, z3, z4, z5, z6, z7, z8,
.         xdof,
.         lft, llt)
      include 'nlqparm'
c
c      Compute b and g matrix for user-defined solid 101
c
      dimension bmtrx (nlq, 3, 3, *), gmtrx (nlq, 3, 3), gjac (nlq)
      REAL n1, n2, n3, n4, n5, n6, n7, n8
      dimension x1 (nlq), x2 (nlq), x3 (nlq), x4 (nlq)
      dimension x5 (nlq), x6 (nlq), x7 (nlq), x8 (nlq)
      dimension y1 (nlq), y2 (nlq), y3 (nlq), y4 (nlq)
      dimension y5 (nlq), y6 (nlq), y7 (nlq), y8 (nlq)
      dimension z1 (nlq), z2 (nlq), z3 (nlq), z4 (nlq)
      dimension z5 (nlq), z6 (nlq), z7 (nlq), z8 (nlq)
      dimension xdof (nlq, 8, 3)
c
      do i=lft, llt
c
          gmtrx (i, 1, 1) = x1 (i) * dn1dxi + x2 (i) * dn2dxi +
1             x3 (i) * dn3dxi + x4 (i) * dn4dxi +
2             x5 (i) * dn5dxi + x6 (i) * dn6dxi +
3             x7 (i) * dn7dxi + x8 (i) * dn8dxi

```

```

gmtrx(i,2,1)=y1(i)*dn1dxi+y2(i)*dn2dxi+
1   y3(i)*dn3dxi+y4(i)*dn4dxi+
2   y5(i)*dn5dxi+y6(i)*dn6dxi+
3   y7(i)*dn7dxi+y8(i)*dn8dxi
gmtrx(i,3,1)=z1(i)*dn1dxi+z2(i)*dn2dxi+
1   z3(i)*dn3dxi+z4(i)*dn4dxi+
2   z5(i)*dn5dxi+z6(i)*dn6dxi+
3   z7(i)*dn7dxi+z8(i)*dn8dxi
gmtrx(i,1,2)=x1(i)*dn1deta+x2(i)*dn2deta+
1   x3(i)*dn3deta+x4(i)*dn4deta+
2   x5(i)*dn5deta+x6(i)*dn6deta+
3   x7(i)*dn7deta+x8(i)*dn8deta
gmtrx(i,2,2)=y1(i)*dn1deta+y2(i)*dn2deta+
1   y3(i)*dn3deta+y4(i)*dn4deta+
2   y5(i)*dn5deta+y6(i)*dn6deta+
3   y7(i)*dn7deta+y8(i)*dn8deta
gmtrx(i,3,2)=z1(i)*dn1deta+z2(i)*dn2deta+
1   z3(i)*dn3deta+z4(i)*dn4deta+
2   z5(i)*dn5deta+z6(i)*dn6deta+
3   z7(i)*dn7deta+z8(i)*dn8deta
gmtrx(i,1,3)=x1(i)*dn1dzeta+x2(i)*dn2dzeta+
1   x3(i)*dn3dzeta+x4(i)*dn4dzeta+
2   x5(i)*dn5dzeta+x6(i)*dn6dzeta+
3   x7(i)*dn7dzeta+x8(i)*dn8dzeta
gmtrx(i,2,3)=y1(i)*dn1dzeta+y2(i)*dn2dzeta+
1   y3(i)*dn3dzeta+y4(i)*dn4dzeta+
2   y5(i)*dn5dzeta+y6(i)*dn6dzeta+
3   y7(i)*dn7dzeta+y8(i)*dn8dzeta
gmtrx(i,3,3)=z1(i)*dn1dzeta+z2(i)*dn2dzeta+
1   z3(i)*dn3dzeta+z4(i)*dn4dzeta+
2   z5(i)*dn5dzeta+z6(i)*dn6dzeta+
3   z7(i)*dn7dzeta+z8(i)*dn8dzeta
c
bmtrx(i,1,1,1) =dn1dxi
bmtrx(i,1,1,4) =dn2dxi
bmtrx(i,1,1,7) =dn3dxi
bmtrx(i,1,1,10)=dn4dxi
bmtrx(i,1,1,13)=dn5dxi
bmtrx(i,1,1,16)=dn6dxi
bmtrx(i,1,1,19)=dn7dxi
bmtrx(i,1,1,22)=dn8dxi
c
bmtrx(i,2,1,2) =dn1dxi
bmtrx(i,2,1,5) =dn2dxi
bmtrx(i,2,1,8) =dn3dxi
bmtrx(i,2,1,11)=dn4dxi
bmtrx(i,2,1,14)=dn5dxi
bmtrx(i,2,1,17)=dn6dxi
bmtrx(i,2,1,20)=dn7dxi
bmtrx(i,2,1,23)=dn8dxi
c
bmtrx(i,3,1,3) =dn1dxi
bmtrx(i,3,1,6) =dn2dxi
bmtrx(i,3,1,9) =dn3dxi
bmtrx(i,3,1,12)=dn4dxi
bmtrx(i,3,1,15)=dn5dxi
bmtrx(i,3,1,18)=dn6dxi
bmtrx(i,3,1,21)=dn7dxi
bmtrx(i,3,1,24)=dn8dxi
c
bmtrx(i,1,2,1) =dn1deta
bmtrx(i,1,2,4) =dn2deta
bmtrx(i,1,2,7) =dn3deta

```

```
      bmtrx(i,1,2,10)=dn4deta
      bmtrx(i,1,2,13)=dn5deta
      bmtrx(i,1,2,16)=dn6deta
      bmtrx(i,1,2,19)=dn7deta
      bmtrx(i,1,2,22)=dn8deta
c
      bmtrx(i,2,2,2) =dn1deta
      bmtrx(i,2,2,5) =dn2deta
      bmtrx(i,2,2,8) =dn3deta
      bmtrx(i,2,2,11)=dn4deta
      bmtrx(i,2,2,14)=dn5deta
      bmtrx(i,2,2,17)=dn6deta
      bmtrx(i,2,2,20)=dn7deta
      bmtrx(i,2,2,23)=dn8deta
c
      bmtrx(i,3,2,3) =dn1deta
      bmtrx(i,3,2,6) =dn2deta
      bmtrx(i,3,2,9) =dn3deta
      bmtrx(i,3,2,12)=dn4deta
      bmtrx(i,3,2,15)=dn5deta
      bmtrx(i,3,2,18)=dn6deta
      bmtrx(i,3,2,21)=dn7deta
      bmtrx(i,3,2,24)=dn8deta
c
      bmtrx(i,1,3,1) =dn1dzeta
      bmtrx(i,1,3,4) =dn2dzeta
      bmtrx(i,1,3,7) =dn3dzeta
      bmtrx(i,1,3,10)=dn4dzeta
      bmtrx(i,1,3,13)=dn5dzeta
      bmtrx(i,1,3,16)=dn6dzeta
      bmtrx(i,1,3,19)=dn7dzeta
      bmtrx(i,1,3,22)=dn8dzeta
c
      bmtrx(i,2,3,2) =dn1dzeta
      bmtrx(i,2,3,5) =dn2dzeta
      bmtrx(i,2,3,8) =dn3dzeta
      bmtrx(i,2,3,11)=dn4dzeta
      bmtrx(i,2,3,14)=dn5dzeta
      bmtrx(i,2,3,17)=dn6dzeta
      bmtrx(i,2,3,20)=dn7dzeta
      bmtrx(i,2,3,23)=dn8dzeta
c
      bmtrx(i,3,3,3) =dn1dzeta
      bmtrx(i,3,3,6) =dn2dzeta
      bmtrx(i,3,3,9) =dn3dzeta
      bmtrx(i,3,3,12)=dn4dzeta
      bmtrx(i,3,3,15)=dn5dzeta
      bmtrx(i,3,3,18)=dn6dzeta
      bmtrx(i,3,3,21)=dn7dzeta
      bmtrx(i,3,3,24)=dn8dzeta
c
      enddo
c
      return
      end
```

To use the element for a part the section card can be written as

```
*SECTION_SOLID
$   SECID   ELFORM
      1       101
$   NIP     NXDOF   IHGF
      1         0       1
$   XI      ETA     ZETA   WGT
      0.       0.     0.     8.0
```

Examples

Tension test (3D solid)

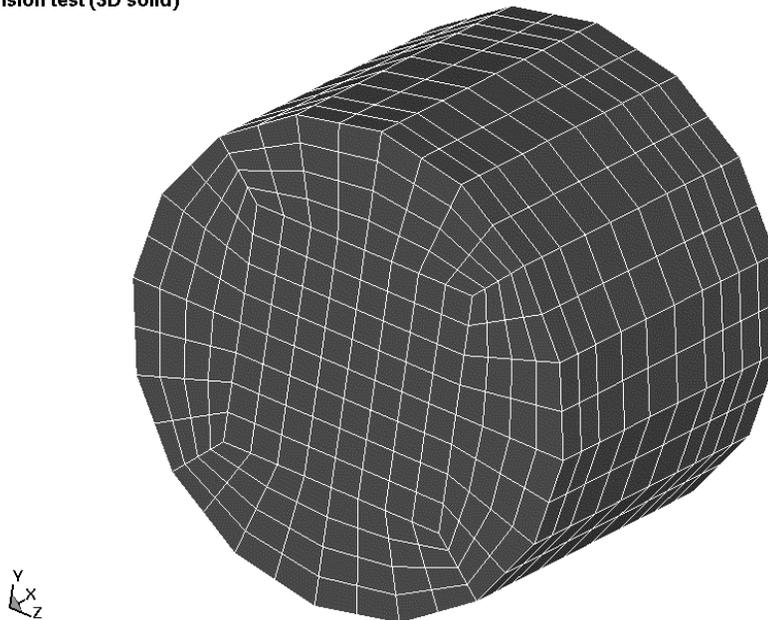


Figure C.1 Solid mesh for user element test.

We present three test examples.

One example was a simple tension-compression test of a solid cylinder. The geometry is shown in Figure C.1. The problem is using the sample implementations of user elements and compared the results and performance with standard LS-DYNA elements. As for the computational efficiency, we note that the performance is worse but this is expected since there is little room for optimization of the code while retaining a user friendly interface. The implicit performance compares well with the other elements in LS-DYNA.

The second example was a combined bending and stretching example with the geometry shown in Figure C.2. Again we ran the problem with the user element implementations and compared the results and performance with standard LS-DYNA elements. We could see the same tendencies as for the solid elements.



Figure C.2 Shell mesh for the user element test.

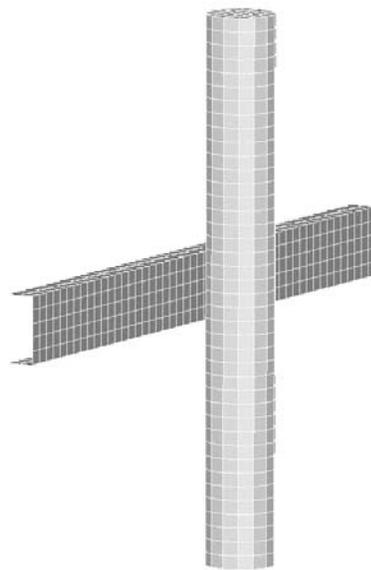


Figure C.3 Impact between a user-defined shell and user-defined solid part.

The third and final example is an impact between a solid bar and shell beam. Both parts are modeled with user-defined elements. The results were very similar to the ones obtained by substituting the sections for standard LS-DYNA sections, but the simulation time was about 3-4 times longer.

APPENDIX D: User Defined Airbag Sensor

The addition of a user sensor subroutine into LS-DYNA is relatively simple. The sensor is mounted on a rigid body which is attached to the structure. The motion of the sensor is provided in the local coordinate system defined for the rigid body in the definition of material model 20—the rigid material. When the user defined criterion is met for the deployment of the airbag, a flag is set and the deployment begins. All load curves relating to the mass flow rate versus time are then shifted by the initiation time. The user subroutine is given below with all the necessary information contained in the comment cards.

```

SUBROUTINE AIRUSR (RBU, RBV, RBA, TIME, DT1, DT2, PARAM, HIST, ITRNON,
.   RBUG, RBVG, RBAG)
C*****
C | LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC) |
C | ----- |
C | COPYRIGHT 1987, 1988, 1989 JOHN O. HALLQUIST, LSTC |
C | ALL RIGHTS RESERVED |
C*****
C
C   USER SUBROUTINE TO INITIATE THE INFLATION OF THE AIRBAG
C
C   VARIABLES
C
C   DISPLACEMENTS ARE DEFINED AT TIME N+1 IN LOCAL SYSTEM
C   VELOCITIES ARE DEFINED AT TIME N+1/2 IN LOCAL SYSTEM
C   ACCELERATIONS ARE DEFINED AT TIME N IN LOCAL SYSTEM
C
C       RBU(1-3) TOTAL DISPLACEMENTS IN THE LOCAL XYZ DIRECTIONS
C       RBU(3-6) TOTAL ROTATIONS ABOUT THE LOCAL XYZ AXES
C       RBV(1-3) VELOCITIES IN THE LOCAL XYZ DIRECTIONS
C       RBV(3-6) ROTATIONAL VELOCITIES ABOUT THE LOCAL XYZ AXES
C       RBA(1-3) ACCELERATIONS IN THE LOCAL XYZ DIRECTIONS
C       RBA(3-6) ROTATIONAL ACCELERATIONS ABOUT THE LOCAL XYZ AXES
C       TIME IS THE CURRENT TIME
C       DT1 IS TIME STEP SIZE AT N-1/2
C       DT2 IS TIME STEP SIZE AT N+1/2
C       PARAM IS USER DEFINED INPUT PARAMETERS (MAX 25)
C       HIST IS USER DEFINED HISTORY VARIABLES (MAX 25)
C       ITRNON IS FLAG TO TURN ON THE AIRBAG INFLATION
C       RBUG, RBVG, RBAG, ARE SIMILAR TO RBU, RBV, RBA BUT ARE DEFINED
C       GLOBALLY.
C
C   THE USER SUBROUTINE SETS THE VARIABLE ITRNON TO:
C
C       ITRNON=0 BAG IS NOT INFLATED
C       ITRNON=1 BAG INFLATION BEGINS AND THIS SUBROUTINE IN NOT
C           CALLED AGAIN
C
C   DIMENSION RBU(6), RBV(6), PARAM(25), HIST(25),
C   .   RBUG(6), RBVG(6), RBAG(6)
RETURN
END

```


APPENDIX E: User Defined Solution Control

This subroutine may be provided by the user to control the I/O, monitor the energies and other solution norms of interest, and to shut down the problem whenever he pleases. The arguments are defined in the listing provided below. This subroutine is called each time step and does not need any control card to operate.

```

SUBROUTINE UCTRL1 (NUMNP, NDOF, TIME, DT1, DT2, PRTC, PLTC, FRCI, PRTO,
. PLTO, FRCO, VT, VR, AT, AR, UT, UR, XMST, XMSR, IRBODY, RBDYN, USRHV,
. MESSAG, TOTALM, CYCL, IDRINT)
C*****
C | LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC) |
C | ----- |
C | COPYRIGHT 1987, 1988, 1989 JOHN O. HALLQUIST, LSTC |
C | ALL RIGHTS RESERVED |
C*****
C
C CHARACTER*(*) MESSAG
C INTEGER CYCLE
C
C
C USER SUBROUTINE FOR SOLUTION CONTROL
C
C NOTE: LS-DYNA USED AN INTERNAL NUMBERING SYSTEM TO
C ACCOMMODATE ARBITRARY NODE NUMBERING. TO ACCESS
C INFORMATION FOR USER NODE N, ADDRESS ARRAY LOCATION M,
C M=LQF(N,1). TO OBTAIN USER NODE NUMBER, N,
C CORRESPONDING TO ARRAY ADDRESS M, SET N=LQFINV(M,1)
C
C ARGUMENTS:
C NUMNP=NUMBER OF NODAL POINTS
C NDOF=NUMBER OF DEGREES IF FREEDOM PER NODE
C TIME=CURRENT SOLUTION TIME
C PRTC=OUTPUT INTERVAL FOR LS-DYNA TIME HISTORY DATA
C PLTC=OUTPUT INTERVAL FOR LS-DYNA STATE DATA
C FRCI=OUTPUT INTERVAL FOR LS-DYNA INTERFACE FORCE DATA
C PRTO=OUTPUT TIME FOR TIME HISTORY FILE
C PLTO=OUTPUT TIME FOR STATE DATA
C FRCO=OUTPUT TIME FOR FORCE DATA
C VT(3,NUMNP) =NODAL TRANSLATIONAL VELOCITY VECTOR
C VR(3,NUMNP) =NODAL ROTATIONAL VELOCITY VECTOR. THIS ARRAY
C IS DEFINED IF AND ONLY IF NDOF=6
C AT(3,NUMNP) =NODAL TRANSLATIONAL ACCELERATION VECTOR
C AR(3,NUMNP) =NODAL ROTATIONAL ACCELERATION VECTOR. THIS
C ARRAY IS DEFINED IF AND ONLY IF NDOF=6
C UT(3,NUMNP) =NODAL TRANSLATIONAL DISPLACEMENT VECTOR
C UR(3,NUMNP) =NODAL ROTATIONAL DISPLACEMENT VECTOR. THIS ARRAY
C IS DEFINED IF AND ONLY IF NDOF=6
C XMST(NUMNP) =RECIPROCAL OF NODAL TRANSLATIONAL MASSES
C XMSR(NUMNP) =RECIPROCAL OF NODAL ROTATIONAL MASSES. THIS
C ARRAY IS DEFINED IF AND ONLY IF NDOF=6
C IRBODY =FLAG FOR RIGID BODY NODAL POINTS
C IF DEFORMABLE NODE THEN SET TO 1.0
C IF RIGID BODY NODE THEN SET TO 0.0
C DEFINED IF AN ONLY IF RIGID BODY ARE PRESENT
C I.E., IRBODY.NE.0 IF NO RIGID BODY ARE PRESENT
C USRHV (LENHV) =USER DEFINED HISTORY VARIABLES THAT ARE STORED

```

```

C          IN THE RESTART FILE.  LENHV=100+U*NUMMAT WHERE
C          NUMMAT IS THE # OF MATERIALS IN THE PROBLEM.
C          ARRAY USRHV IS UPDATED ONLY IN THIS SUBROUTINE.
C          MESSAG      =FLAG FOR DYNA3D WHICH MAY BE SET TO:
C                      'SW1.' LS-DYNA TERMINATES WITH RESTART FILE
C                      'SW3.' LS-DYNA WRITES A RESTART FILE
C                      'SW4.' LS-DYNA WRITES A PLOT STATE
C          TOTALM      =TOTAL MASS IN PROBLEM
C          CYCLE        =CYCLE NUMBER
C          IDRINT        =FLAG FOR DYNAMIC RELAXATION PHASE
C                      .NE.0:  DYNAMIC RELAXATION IN PROGRESS
C                      .EQ.0:  SOLUTION PHASE
C
C          COMMON/PTIMES/  PRTIMS(32) , PRTLST(32) , IGMPT
C
C          PRTIMS(32)=OUTPUT INTERVALS FOR ASCII FILES
C
C          ASCII FILES
C          ( 1)=CROSS SECTION FORCES
C          ( 2)=RIGID WALL FORCES
C          ( 3)=NODAL DATA
C          ( 4)=ELEMENT DATA
C          ( 5)=GLOBAL DATA
C          ( 6)=DISCRETE ELEMENTS
C          ( 7)=MATERIAL ENERGIES
C          ( 8)=NODAL INTERFACE FORCES
C          ( 9)=RESULTANT INTERFACE FORCES
C          (10)=SMUG ANIMATOR
C          (11)=SPC REACTION FORCES
C          (12)=NODAL CONSTRAIN RESULTANT FORCES
C          (13)=AIRBAG STATISTICS
C          (14)=AVS DATABASE
C          (15)=NODAL FORCE GROUPS
C          (16)=OUTPUT INTERVALS FOR NODAL BOUNDARY CONDITIONS
C          (17)-(32)=UNUSED AT THIS TIME
C
C          PRTLST(32)=OUTPUT TIMES FOR ASCII FILES ABOVE.  WHEN SOLUTION TIME
C          EXCEEDS THE OUTPUT TIME A PRINT STATE IS DUMPED.
C
C          COMMON/RBKENG/ENRBDY ,RBDYX ,RBDYY ,RBDYZ
C
C          TOTAL RIGID BODY ENERGIES AND MOMENTUMS:
C          ENRBDY=RIGID BODY KINETIC ENERGY
C          RBDYX =RIGID BODY X-MOMENTUM
C          RBDYY =RIGID BODY Y-MOMENTUM
C          RBDYZ =RIGID BODY Z-MOMENTUM
C
C          COMMON/RBKENG/ENRBDY ,RBDYX ,RBDYY ,RBDYZ
C
C          TOTAL RIGID BODY ENERGIES AND MOMENTUMS:
C          SWXMOM=STONEWALL X-MOMENTUM
C          SWYMOM=STONEWALL Y-MOMENTUM
C          SWZMOM=STONEWALL Z-MOMENTUM
C          ENRBDY=STONEWALL KINETIC ENERGY
C

```

```

COMMON/DEENG/DEENG
C
C DEENG=TOTAL DISCRETE ELEMENT ENERGY
C
COMMON/ENERGY/XPE
C
C XPE =TOTAL INTERNAL ENERGY IN THE FINITE ELEMENTS
C
DIMENSION VT(3,*),VR(3,*),AT(3,*),AR(3,*),UT(3,*),UR(3,*),
XNST(*),XMSR(*),RBDYN(*),USRHV(*)
C
C SAMPLE MOMENTUM AND KINETIC ENERGY CALCULATIONS
C
C REMOVE ALL COMMENTS IN COLUMN 1 BELOW TO ACTIVATE
CC
CC INITIALIZE KINETIC ENERGY, XKE, AND X,Y,Z MOMENTUMS.
CC
C XKE=2.*SWKENG+2.*ENRBDY
C XM=SWXMOM+RBDYX
C YM=SWYMOM+RBDYY
C ZM=SWZMOM+RBDYZ
CC
C NUMNP2=NUMNP
C IF (NDOF.EQ.6) THEN
C NUMNP2=NUMNP+NUMNP
C ENDIF
C PRINT *,NDOF
C IF (IRBODY.EQ.0) THEN
CC
CC NO RIGID BODIES PRESENT
CC
CC NOTE IN BLANK COMMENT VR FOLLOWS VT. THIS FACT IS USED BELOW.
C DO 10 N=1,NUMNP2
C XMSN=1./XNST(N)
C VN1=VT(1,N)
C VN2=VT(2,N)
C VN3=VT(3,N)
C XM=XM+XMSN*VN1
C YM=YM+XMSN*VN2
C ZM=ZM+XMSN*VN3
C XKE=XKE+XMSN*(VN1*VN1+VN2*VN2+VN3*VN3)
C 10 CONTINUE
CC
C ELSE
CC
CC RIGID BODIES PRESENT
CC
C DO 20 N=1,NUMNP
C XMSN=1./XNST(N)
C VN1=RBDYN(N)*VT(1,N)
C VN2=RBDYN(N)*VT(2,N)
C VN3=RBDYN(N)*VT(3,N)
C XM=XM+XMSN*VN1
C YM=YM+XMSN*VN2
C ZM=ZM+XMSN*VN3
C XKE=XKE+XMSN*(VN1*VN1+VN2*VN2+VN3*VN3)
C 20 CONTINUE
C IF (NDOF.EQ.6) THEN
C DO 30 N=1,NUMNP

```

```
C      XMSN=1./XMSR(N)
C      VN1=RBDYN(N)*VR(1,N)
C      VN2=RBDYN(N)*VR(2,N)
C      VN3=RBDYN(N)*VR(3,N)
C      XM=XM+XMSN*VN1
C      YM=YM+XMSN*VN2
C      ZM=ZM+XMSN*VN3
C      XKE=XKE+XMSN*(VN1*VN1+VN2*VN2+VN3*VN3)
C  30  CONTINUE
C      ENDIF
CC
C      ENDIF
C      RETURN
C      END

CC
CC.....TOTAL KINETIC ENERGY
C      XKE=.5*XKE
CC.....TOTAL INTERNAL ENERGY
C      XIE=.XPE+DEENG
CC.....TOTAL ENERGY
C      XTE=XKE+XPE+DEENG
CC.....TOTAL X-RIGID BODY VELOCITY
C      XRBV=XM/TOTALM
CC.....TOTAL Y-RIGID BODY VELOCITY
C      YRBV=YM/TOTALM
CC.....TOTAL Z-RIGID BODY VELOCITY
C      ZRBV=ZM/TOTALM
C
C      RETURN
C      END
```

APPENDIX F: User Defined Interface Control

This subroutine may be provided by the user to turn the interfaces on and off. This option is activated by the *USER_INTERFACE_CONTROL keyword. The arguments are defined in the listing provided below.

```

SUBROUTINE UCTRL2 (NSI,NTY,TIME,CYCLE,MSR,NMN,NSV,NSN,
1 THMR,THSV,VT,XI,UT,ISKIP,IDRINT,NUMNP,DT2,NINPUT,UA)
C*****
C LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC)
C -----
C COPYRIGHT 1987, 1988, 1989 JOHN O. HALLQUIST, LSTC
C ALL RIGHTS RESERVED
C*****
C
C INTEGER CYCLE
C
C
C USER SUBROUTINE FOR INTERFACE CONTROL
C
C NOTE: LS-DYNA USED AN INTERNAL NUMBERING SYSTEM TO
C ACCOMMODATE ARBITRARY NODE NUMBERING. TO ACCESS
C INFORMATION FOR USER NODE N, ADDRESS ARRAY LOCATION M,
C M=LQF(N,1). TO OBTAIN USER NODE NUMBER, N,
C CORRESPONDING TO ARRAY ADDRESS M, SET N=LQFINV(M,1)
C
C ARGUMENTS:
C NSI =NUMBER OF SLIDING INTERFACE
C NTY =INTERFACE TYPE.
C .EQ.4:SINGLE SURFACE
C .NE.4:SURFACE TO SURFACE
C TIME =CURRENT SOLUTION TIME
C CYCLE =CYCLE NUMBER
C MSR(NMN) =LIST OF MASTER NODES NUMBERS IN INTERNAL
C NUMBERING SCHEME
C NMN =NUMBER OF MASTER NODES
C NSV(NSN) =LIST OF SLAVE NODES NUMBERS IN INTERNAL
C NUMBERING SCHEME
C NSN =NUMBER OF SLAVE NODES
C THMR(NMN) =MASTER NODE THICKNESS
C THSV(NSN) =SLAVE NODE THICKNESS
C VT(3,NUMNP) =NODAL TRANSLATIONAL VELOCITY VECTOR
C XI(3,NUMNP) =INITIAL COORDINATES AT TIME=0
C UT(3,NUMNP) =NODAL TRANSLATIONAL DISPLACEMENT VECTOR
C IDRINT =FLAG FOR DYNAMIC RELAXATION PHASE
C .NE.0:DYNAMIC RELAXATION IN PROGRESS
C .EQ.0:SOLUTION PHASE
C NUMNP =NUMBER OF NODAL POINTS
C DT2 =TIME STEP SIZE AT N+1/2
C NINPUT =NUMBER OF VARIABLES INPUT INTO UA
C UA(*) =USER'S ARRAY, FIRST NINPUT LOCATIONS
C DEFINED BY USER. THE LENGTH OF THIS
C ARRAY IS DEFINED ON CONTROL CARD 10.
C THIS ARRAY IS UNIQUE TO INTERFACE NSI.
C
C SET FLAG FOR ACTIVE CONTACT
C ISKIP=0 ACTIVE
C ISKIP=1 INACTIVE
C*****

```

```

DIMENSION MSR(*),NSV(*),THMR(*),THSV(*),VT(3,*),XI(3,*),
          UT(3,*)UA(*)
C
C THE FOLLOWING SAMPLE OF CODEING IS PROVIDED TO ILLUSTRATE HOW
C THIS SUBROUTINE MIGHT BE USED. HERE WE CHECK TO SEE IF THE
C SURFACES IN THE SURFACE TO SURFACE CONTACT ARE SEPARATED. IF
C SO THE ISKIP=1 AND THE CONTACT TREATMENT IS SKIPPED.
C
      IF (NTY.EQ.4) RETURN
      DT2HLF=DT2/2.
      XMIN= 1.E20
      XMAX=-XMIN
      YMIN= 1.E20
      YMAX=-YMIN
      ZMIN= 1.E20
      ZMAX=-ZMIN
      XMINM= 1.E20
      XMAXM=-XMINM
      YMINM= 1.E20
      YMAXM=-YMINM
      ZMINM= 1.E20
      ZMAXM=-ZMINM
      THKS=0.0
      THKM=0.0
      DO 10 I=1,NSN
      DSP1=UT(1,NSV(I))+DT2HLF*VT(1,NSV(I))
      DSP2=UT(2,NSV(I))+DT2HLF*VT(2,NSV(I))
      DSP3=UT(3,NSV(I))+DT2HLF*VT(3,NSV(I))
      X1=XI(1,NSV(I))+DSP1
      X2=XI(2,NSV(I))+DSP2
      X3=XI(3,NSV(I))+DSP3
      THKS =MAX(THSV(I),THKS)
      XMIN=MIN(XMIN,X1)
      XMAX=MAX(XMAX,X1)
      YMIN=MIN(YMIN,X2)
      YMAX=MAX(YMAX,X2)
      ZMIN=MIN(ZMIN,X3)
      ZMAX=MAX(ZMAX,X3)
10 CONTINUE
      DO 20 I=1,NMN
      DSP1=UT(1,MSR(I))+DT2HLF*VT(1,MSR(I))
      DSP2=UT(2,MSR(I))+DT2HLF*VT(2,MSR(I))
      DSP3=UT(3,MSR(I))+DT2HLF*VT(3,MSR(I))
      X1=XI(1,MSR(I))+DSP1
      X2=XI(2,MSR(I))+DSP2
      X3=XI(3,MSR(I))+DSP3
      THKM =MAX(THMR(I),THKS)
      XMIN=MIN(XMIN,X1)
      XMAX=MAX(XMAX,X1)
      YMIN=MIN(YMIN,X2)
      YMAX=MAX(YMAX,X2)
      ZMIN=MIN(ZMIN,X3)
      ZMAX=MAX(ZMAX,X3)
20 CONTINUE
      IF (XMAX+THKS.LT.XMIN-THKM) GO TO 40
      IF (YMAX+THKS.LT.YMIN-THKM) GO TO 40
      IF (ZMAX+THKS.LT.ZMIN-THKM) GO TO 40
      IF (XMAX+THKM.LT.XMIN-THKS) GO TO 40
      IF (YMAX+THKM.LT.YMIN-THKS) GO TO 40
      IF (ZMAX+THKM.LT.ZMIN-THKS) GO TO 40
      ISKIP=0
      RETURN
40 ISKIP=1
      RETURN
      END

```

APPENDIX G: User Defined Interface Friction

This subroutine may be provided by the user to set the Coulomb friction coefficients. This option is activated by the *USER_INTERFACE_FRICTION keyword. The arguments are defined in the listing provided below.

```

subroutine usrfrc(nosl,time,ncycle,dt2,insv,areas,xs,ys,zs,
. lsv,ix1,ix2,ix3,ix4,aream,xx1,xx2,xx3,stfn,stf,fni,
. dx,dy,dz,fdt2,ninput,ua,side,iisv5,niisv5,n1,n2,n3,fric1,
. fric2,fric3,fric4,bignum,fdat,iseg,fxis,fyis,fzis,ss,tt,
. ilbsv,stfk,frc,numnp,np,pld,lcfst,lcfdt,temp,temp_bot,
. temp_top,isurface)
c
c*****
c | LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC) |
c | ----- |
c | COPYRIGHT © 1987-2007 JOHN O. HALLQUIST, LSTC |
c | ALL RIGHTS RESERVED |
c*****
c
c user subroutine for interface friction control
c
c note: LS-DYNA uses an internal numbering system to
c accomodate arbitrary node numbering. to access
c information for user node n, address array location m,
c m=lqf(n,1). to obtain user node number, n,
c corresponding to array address m, set n=lqfinv(m,1)
c
c arguments:
c
c nosl =number of sliding interface
c time =current solution time
c ncycle =ncycle number
c dt2 =time step size at n+1/2
c insv =slave node array where the nodes are stored
c in ls-dyna3d internal numbering. User numbers
c are given by function: lqfinv(insv(ii),1)
c for slave node ii.
c areas(ii) =slave node area (interface types 5&10 only) for
c slave node ii
c xs(ii) =x-coordinate slave node ii (projected)
c ys(ii) =y-coordinate slave node ii (projected)
c zs(ii) =z-coordinate slave node ii (projected)
c lsv(ii) =master segment number for slave node ii
c ix1(ii), ix2(ii), ix3(ii), ix4(ii)
c =master segment nodes in ls-dyna3d internal
c numbering for slave node ii
c aream(ii) =master segment area for slave node ii.
c xx1(ii,4) =x-coordinates master surface (projected) for
c slave node ii
c xx2(ii,4) =y-coordinates master surface (projected) for
c slave node ii
c xx3(ii,4) =z-coordinates master surface (projected) for
c slave node ii
c stfn =slave node penalty stiffness
c stf =master segment penalty stiffness
c fni =normal force
c dx,dy,dz =relative x,y,z-displacement between slave node and
c master surface. Multipling by fdt2 defines the
c relative velocity.
c n1,n2,n3 =x,y, and z components of master segments normal
c vector
c

```

```

c*****
c      frictional coefficients defined for the contact interface
c
c      fric1      =static friction coefficient
c      fric2      =dynamic friction coefficient
c      fric3      =decay constant
c      fric4      =viscous friction coefficient (setting fric4=0
c                  turns this option off)
c
c*****
c
c      bignum      =0.0 for one way surface to surface and
c                  for surface to surface, and 1.e+10 for nodes
c                  to surface contact
c      ninput      =number of variables input into ua
c      ua(*)       =users' array, first ninput locations
c                  defined by user.  the length of this
c                  array is defined on control card 10.
c                  this array is unique to interface nosl.
c
c      side        ='master' for first pass.  the master
c                  surface is the surface designated in the
c                  input
c                  ='slave' for second pass after slave and
c                  master surfaces have be switched for
c                  the type 3 symmetric interface treatment.
c
c      iisv5       =an array giving the pointers to the active nodes
c                  in the arrays.
c
c      niisv5      =number of active nodes
c
c      fdat        =contact history data array
c      iseg        =contact master segment from previous step.
c      fxis        =slave node force component in global x dir.
c                  to be updated to include friction
c      fyis        =slave node force component in global y dir.
c                  to be updated to include friction
c      fzis        =slave node force component in global z dir.
c                  to be updated to include friction
c      ss(ii)      =s contact point (-1 to 1) in parametric coordinates
c                  for slave node ii.
c      tt(ii)      =t contact point (-1 to 1) in parametric coordinates
c                  for slave node ii.
c      ilbsv(ii)   =pointer for node ii into global arrays.
c      stfk(ii)    =penalty stiffness for slave node ii which was used
c                  to compute normal interface force.
c      frc(1,lsv(ii))
c                  =Coulomb friction scale factor for segment lsv(ii)
c      frc(2,lsv(ii))
c                  =viscous friction scale factor for segment lsv(ii)
c
c*****
c      parameters for a coupled thermal-mechanical contact
c
c      numnp       = number of nodal points in the model
c      npc         = load curve pointer
c      pld         = load curve (x,y) data
c      lcfst(nosl)= load curve number for static coefficient of
c                  friction versus temperture for contact
c                  surface nosl
c      lcfdt(nosl)= load curve number for dynamic coefficient of
c                  friction versus temperture for contact
c                  surface nosl
c      temp(j)     = temperature for node point j
c      temp_bot(j)= temperature for thick thermal shell bottom

```

```
c          surface
c      temp_top(j)= temperature for thick thermal shell top
c          surface
c      numsh12   = number of thick thermal shells
c      itopaz(1) = 999 ==> thermal-mechanical analysis
c      isurface  = thick thermal shell surface pointer
c
c*****
```


APPENDIX H: User Defined Thermal Material Model

The addition of a thermal user material routine into LS-DYNA is fairly straightforward. The thermal user material is controlled using the keyword `*MAT_THERMAL_USER_DEFINED`, which is described at the appropriate place in the manual.

The thermal user material can be used alone or in conjunction with any given mechanical material model in a coupled thermal-mechanical solution. A heat-source can be included and the specific heat updated so that it possible to model e.g. phase transformations including melt energy.

If for the same part (shell or solid elements) both a thermal and mechanical user material model is defined then the two user material models have (optionally) read access to each other's history variables. If the integration points of the thermal and mechanical elements not are coincident then interpolation or extrapolation is used when reading history variables. Linear interpolation or extrapolation using history data from the two closest integration points is used in all cases except when reading history variables from the thick thermal shell (`TSHELL=1` on `*CONTROL_SHELL`). For the latter thermal shell, the shape functions of the element are used for the interpolation or extrapolation.

The thermal user materials are thermal material types 11-15. These thermal user material subroutines are defined in file `dyn21.f` as subroutines `thumat11`, ... , `thumat15`. The latter subroutines are called from the subroutine `thusrmat`. The source code of subroutine `thusrmat` is also in file `dyn21.f`. Additional useful information is available in the comments of subroutines `thusrmat`, `thumat12`, and `umat46` that all reside in the source file `dyn21.f`

Thermal history variables

Thermal history variables can be used by setting `NVH` greater than 0. Thermal history variables are output to the `tprint` file, see `*DATABASE_TPRINT`.

Interchange of history variables with mechanical user material

In a coupled thermo-mechanical solution there is for each mechanical shell, thick shell, or solid element a corresponding thermal element. A pair consisting of a mechanical and a corresponding thermal element both have integration points and possibly history variables. The mechanical and thermal elements do not necessarily have the same number of integration points.

By setting `IHVE` to 1, a thermal user material model can read, but not write, the history variables from a mechanical user material model and vice versa.

If the locations of the points where the history variables are located differ between the mechanical and thermal element interpolation or extrapolation is used to calculate the history value. More information is available in the comments to the subroutines `thusrmat` and `thumat11`.

Limitations

Currently there are a few limitations of the thermal user material implementation. LS-DYNA will in most cases give an appropriate warning or error message when such a limit is violated. The limitations include:

1. Option `IHVE.EQ.1` is only supported for a limited range of mechanical elements:
 - Solid elements: `ELFORM=1, 2, 10, 13`.

- Shell elements: ELFORM=2, 3, 4, 16. Note that user-defined integration rules are not supported.
2. Thermal history variables limitations:
 - Restart including thermal history variables is not supported in MPP-DYNA.
 - Thermal history variables cannot be output to a DYNAIN file.
 - Thermal history variables are not output to d3plot.
 - 3D-shell mesh adaptivity is not supported when using thermal history variables.
 3. The thermal solver includes not only the plastically dissipated energy as a heat source but also wrongly the elastic energy. The latter however is in most cases not of practical importance.

Example source code

Example source code for thermal user material models is available in thumat11 and thumat12 as well as in umat46. Note that there is space for up to 64 material parameters in r_matp (material parameter array) but only 32 can be read in from the *MAT_THERMAL_USER_DEFINED card. The material parameters in r_matp(i), i=41-64, which are initially set to 0.0, may be used by the user to store additional data.

Subroutine crvval evaluates load curves. Note that when using crvval the load curves are first re-interpolated to 100 equidistant points. See Appendix A for more information on subroutine crvval.

Following is a short thermal user material model. The card format is in this case, if enabling orthotropic conduction, and with sample input in SI-units:

*MAT_THERMAL_USER_DEFINED

Card 1	1	2	3	4	5	6	7	8
Variable	MID	RO	MT	LMC	NVH	AOPT	IORTHO	IHVE
Type	21	7800.0	12.0	6.0	3.0	0.0	1.0	0.0

Card 2

Variable	XP	YP	ZP	A1	A2	A3		
Type	0.0	0.0	0.0	0.0	0.0	0.0		

Card 3 1 2 3 4 5 6 7 8

Variable	D1	D2	D3					
Type	0.0	0.0	0.0					

Card 4

Variable	C1	C2	C3	HC	HSRC	HCFAC		
Type	25.0	25.0	20.0	470.0	11.0	12.0		

VARIABLE	DESCRIPTION
----------	-------------

C1-C3	Heat conduction in 11, 22, and 33 direction of material coordinate system.
HC	Heat capacity
HSRC	Load curve ID of load curve giving a heat source output (W/m3) as a function of time.
HCFAC	Load curve ID of load curve giving a scaling of the heat capacity as function of time.

The source code is:

```

subroutine thumat12(c1,c2,c3,cvl,dcvdt1,hsrcl,dhsrctl,
1      hsv,hsvm,nmecon,r_matp,crv,
2      nel,nep,iop,eltype,dt,atime,ihsrcl)
character*(*) eltype
dimension hsv(*),hsvm(*),r_matp(*),crv(101,2,*)
include 'iounits.inc'
c
c      Thermal user-material number 12.
c
c      See comments at the beginning of subroutine thusrmat
c      for instructions.
c
c      Example: isotropic/orthotropic material with k1=P1 and
c      cvl=P2 for solid and shell elements including optional
c      change of heat capacity and a heat source, both functions
c      of time input as load curves.
c
c      Print out some info on start-up, use material parameter 64
c      as a flag.
if(nint(r_matp(64)).eq.0) then
  r_matp(64)=1.
  write(*,1200) (r_matp(8+i),i=1,6)
  write(iohsp,1200) (r_matp(8+i),i=1,6)
  write(59,1200) (r_matp(8+i),i=1,6)

```

```
endif
c
c Calculate response
c1=r_matp(8+1)
c2=r_matp(8+2)
c3=r_matp(8+3)
cvl=r_matp(8+4)
dcvdtl=0.0
eid=nint(r_matp(8+6))
if(nint(eid).gt.0) then
  call crvval(crv,eid,atime,cvlfac,tmp1)
  cvl=cvl*cvlfac
  dcvdtl=0.0
endif
c
c If flux or time step calculation then we are done.
if(eltype.eq.'soliddt'.or.eltype.eq.'flux'.or.
.  eltype.eq.'shelldt') return
eid=nint(r_matp(8+5))
if(nint(eid).gt.0) then
  ihsrcl=1
  call crvval(crv,eid,atime,hsrcl,tmp1)
  dhsrcl=0.0
endif
c
c Update history variables
hsv(1)=cvl
hsv(2)=atime
hsv(3)=hsv(3)+1.0
c
c Done
return
1200 format(/'This is thermal user defined material #12. '/
1      /' Material parameter c1-c3      : ',3E10.3
2      /' Material parameter hc        : ',E10.3
3      /' Heat source load curve       : ',F10.0
4      /' hc scale factor load curve   : ',F10.0
5      /' Thermal History variable 1   : cv'
6      /' Thermal History variable 2-3 : Dummy'/)
return
end
```

APPENDIX I: Occupant Simulation Including the Coupling to Programs CAL3D and MADYMO

INTRODUCTION

LS-DYNA is coupled to occupant simulation codes to generate solutions in automotive crashworthiness that include occupants interacting with the automotive structure. In such applications LS-DYNA provides the simulation of the structural and deformable aspects of the model and the OSP (Occupant Simulation Program) simulates the motion of the occupant. There is some overlap between the two programs which provides flexibility in the modeling approach. For example, both the OSP and LS-DYNA have the capability of modeling seat belts and other deformable restraints. The advantage of using the OSP is related to the considerable databases and expertise that have been developed in the past for simulating dummy behavior using these programs.

The development of the interface provided LSTC a number of possible approaches. The approach selected is consistent with the LSTC philosophy of providing the most flexible and useful interface possible. This is important because the field of non-linear mechanics is evolving rapidly and techniques which are used today are frequently rendered obsolete by improved methodologies and lower cost computing which allows more rigorous techniques to be used. This does make the learning somewhat more difficult as there is not any single procedure for performing a coupling.

One characteristic of LS-DYNA is the large number of capabilities, particularly those associated with rigid bodies. This creates both an opportunity and a difficulty: LSDYNA3D has many ways approximating different aspects of problems, but they are frequently not obvious to users without considerable experience. Therefore, in this Appendix we emphasize modeling methods rather than simply listing capabilities.

THE LS-DYNA/OCCUPANT SIMULATION PROGRAM LINK

Coupling between the OSP and LS-DYNA is performed by combining the programs into a single executable. In the case of CAL3D, LS-DYNA calls CAL3D as a subroutine, but in the case of MADYMO, LS-DYNA is called as a subroutine. The two programs are then integrated in parallel with the results being passed between the two until a user defined termination time is reached.

The OSP and LS-DYNA have different approaches to the time integration schemes. The OSP time integrators are based on accurate implicit integrators which are valid for large time steps which are on the order of a millisecond for the particular applications of interest here. An iterative solution is used to insure that the problem remains in equilibrium. The implicit integrators are extremely good for smoothly varying loads, however, sharp nonlinear pulses can introduce considerable error. An automatic time step size control which decreases the time step size quickly restores the accuracy for such events. The LS-DYNA time integrator is based on an explicit central difference scheme. Stability requires that the time step size be less than the highest frequency in the system. For a coarse airbag mesh, this number is on the order of 100

microseconds while an actual car crash simulation is on the order of 1 microsecond. The smallest LS-DYNA models have at least 1,000 elements. Experience indicates that the cost of a single LS-DYNA time step for a small model is at least as great as the cost of a time step in the OSP. Therefore, in the coupling, the LS-DYNA time step is used to control the entire simulation including the OSP part. This approach has negligible cost penalties and avoids questions of stability and accuracy that would result by using a subcycling scheme between the two programs. Optionally, a subcycling scheme can be used, however, the results of the analysis have to be checked with care.

LS-DYNA has a highly developed rigid body capability which is used in different parts of automobile crash simulation. In particular, components such as the engine are routinely modeled with rigid bodies. These rigid bodies have been modified so that they form the basis of the coupling procedure in LS-DYNA to the OSP.

In LS-DYNA, the geometry of a model is broken down into nodal points which identify positions in space. These nodes are then connected by elements so that the volume of a structure is identified. Each element has a “material” associated with it. If the element is deformable, then the material will specify its characteristics such as density and Young’s Modulus. A crash model can consist of 100 or more separate materials which are each assigned a “material number,” and each material number has an associated “material type” which determines if it is elastic, plastic, viscoelastic, orthotropic, etc.

The material type may also specify that it is a rigid body. In this case, all elements of the same material number are treated as a single rigid body. These elements are integrated to determine the mass, centroid and moments of inertia for the group. This group is then treated as a rigid body with six degrees-of-freedom including three translations and three rotations. The positions of the rigid bodies are updated in LS-DYNA by a time integrator which works together with the central difference time integration.

There is an additional flag which specifies that the LS-DYNA rigid body is coupled to an OSP rigid body. This flag can be found in the description of the rigid body material *MAT_RIGID (formerly material type 20). In coupled updates, the OSP rigid body time integrator takes over control of the LS-DYNA rigid body and the normal LS-DYNA updates are bypassed. The time integration procedure is then as follows:

1. At the beginning of a step, LS-DYNA determines the locations and updates the positions of all of the rigid bodies which are coupled to the OSP. This information is obtained from common block information in the OSP.
2. Using the information on rigid body locations, LS-DYNA proceeds to update the stresses and history variables of all of the deformable structures and computes the resultant forces acting on all rigid bodies.
3. The resultant forces are stored into an OSP common block along with the current time step. Control is then returned to the OSP so that the step can be completed by the OSP determining the new positions of the rigid bodies based on the applied forces.

At the end of the calculation LS-DYNA terminates normally, closing its files, and then control is returned to OSP which will also terminate normally. The termination time for the coupled run is

taken as the minimum of the termination time provided to LS-DYNA and the termination time provided to the OSP.

The executable for the coupling with MADYMO currently needs to be specially created at each site. TNO provides all of the appropriate load modules with their libraries, and the appropriate load modules for LS-DYNA may be obtained by the corporate contact point at the LS-DYNA distributor. A complete executable must then be made by linking the two libraries. A revised password file must be obtained from TNO prior to running the coupled code. Coupling with CAL3D requires special on-site modification of the client's CAL3D version to eliminate conflicting I/O unit numbers and to ensure that the common block lengths between the codes are consistent. LSTC does not distribute or support CAL3D.

To make the coupled program run, an input deck must be provided to both the OSP and LS-DYNA. The two input decks must be provided in the same set of consistent units. This can potentially require a major conversion to either the OSP input or the LS-DYNA input. With two legitimate and consistent input decks, the coupled program should run to completion with no problems. Additional inputs are required to make the models interact between the OSP and LS-DYNA portions of the run.

The simplest form of a coupled simulation is simply to include a single body in an OSP run. No special modifications are needed to the OSP input deck for use in the coupled simulation. Ellipsoids and planes in the OSP are usually attached to "segments" which correspond to LS-DYNA "rigid bodies." Because the coupling procedure works on the basis of shared information on LS-DYNA rigid bodies with the OSP segments, the ellipsoids/planes listed in the OSP section must correspond to the segments which are to be coupled. These ellipsoids and planes may be actual geometry which is used for contact, or they may be simply artificial shapes to permit the data transfer between the OSP and LS-DYNA.

DUMMY MODELING

The dummy is typically modeled entirely within the OSP. The coupling of the dummy into LS-DYNA requires the creation of a separate LS-DYNA rigid body material for each segment of the OSP. The easiest way to create a mesh for the model is to set the LS-DYNA rigid body coupling option to 2.0. This causes LS-DYNA to search all of the ellipsoids connected to the appropriate segment and generate meshes which are then slaved to the OSP dummy. Thus, with minimal input, a complete dummy may be generated and the kinematics may be traced in LS-DYNA and displayed in the LS-DYNA post-processor, LS-PREPOST.

Once the basic dummy coupling has been accomplished, the deformable finite element structure can be added. Assuming that an ellipsoid is available for the steering wheel, a flat airbag can be added in the proper location. One or more nodes must be attached to the steering wheel. This is done by identifying the attached nodes as "Extra Nodes for Rigid Body" which is input in LS-DYNA by *CONSTRAINED_EXTRA_NODES_Option. The nodes are slaved to the LS-DYNA material which has been coupled to the MADYMO steering wheel model. Contact must now be identified between the airbag and the steering wheel, the windshield, and the various body parts which may be affected. This requires the use of one geometric contact entity (see *CONTACT_ENTITY) for each plane or ellipsoid which may interact with the airbag. A control volume specifying inflation properties for the airbag must be specified (see *AIRBAG_OPTION) to complete the model.

AIRBAG MODELING

Modeling of airbags is accomplished by use of shell or membrane elements in conjunction with a control volume (see *AIRBAG_OPTION) and possibly a single surface contact algorithm to eliminate interpenetrations during the inflation phase (see *CONTACT_OPTION). The contact types showing an “a” in front are most suited for airbag analysis. Current recommended material types for the airbags are:

*MAT_ELASTIC = Type 1. Elastic

*MAT_COMPOSITE_DAMAGE = Type 22. Layered orthotropic elastic for composites

*MAT_FABRIC = Type 34. Fabric model for folded airbags

Model 34 is a “fabric” model which can be used for flat bags. As a user option this model may or may not support compression.

The elements which can be used are as follows:

Belytschko-Tsay quadrilateral with 1 point quadrature. This element behaves rather well for folded and unfolded cases with only a small tendency to hourglass. The element tends to be a little stiff. Stiffness form hourglass control is recommended.

Belytschko-Tsay membrane. This model is softer than the normal Belytschko-Tsay element and can hourglass quite badly. Stiffness form hourglass is recommended. As a better option, the fully integrated Belytschko-Tsay membrane element can be chosen.

C0 Triangular element. The C0 triangle is very good for flat bag inflation and has no tendency to hourglass.

The best choice is a specially developed airbag membrane element with quadrilateral shape. This is an automatic choice when the fabric material is used.

As an airbag inflates, a considerable amount of energy is transferred to the surrounding air. This energy transfer decreases the kinetic energy of the bag as it inflates. In the control volume logic, this is simulated either by using either a mass weighted damping option or a back pressure on the bag based on a stagnation pressure. In both cases, the energy that is absorbed is a function of the fabric velocity relative to a rigid body velocity for the bag. For the mass weighted case, the damping force on a node is proportional to the mass times the damping factor times the velocity vector. This is quite effective in maintaining a stable system, but has little physical justification. The latter approach using the stagnation pressure method estimates the pressure needed to accelerate the surrounding air to the speed of the fabric. The formula for this is:

$$P = Area \times \alpha \times \left((\vec{V}_i - \vec{V}_{cg}) \cdot \hat{n} \right)^2$$

This formula accomplishes a similar function and has a physical justification. Values of the damping factor, α , are limited to the range of 0 to 1, but a value of 0.1 or less is more likely to be a good value.

KNEE BOLSTER

The knee-to-knee bolster interactions are characterized by the stiffness of the knee being comparable to that of the knee bolster. Therefore, modeling the knee as a rigid body may produce large errors in the interaction forces. Calibrated force-deflection curves could be determined, but they would have no predictive value for slight changes to knee bolster designs. For this reason, a more accurate modeling of the compliance of the knee bolster and the knee is required.

The knee can be modeled as a combined rigid/deformable body. The rigid body is coupled to the OSP. Overlaying the rigid body are brick elements which model the “skin” that exists over the knees of the dummy. These brick elements use material type 6 (*MAT_VISCOELASTIC) which is a viscoelastic model that does a reasonable job of approximating the hysteretic behavior of rubbers. The inner layer of the brick elements is attached to the rigid body through the *CONSTRAINED_EXTRA_NODES Option. Between the knee bolster is a SURFACE-TO-SURFACE contact definition.

COMMON ERRORS

1. **Improper airbag inflation or no inflation.**

The most common problem is inconsistency in the units used for the input constants. An inflation load curve must also be specified. The normals for the airbag segments must all be consistent and facing outwards. If a negative volume results, this can sometimes be quickly cured by using the “flip” flag on the control volume definition to force inward facing normals to face outwards.

2. **Excessive airbag distortions.**

Check the material constants. Triangular elements should have less distortion problems than quadrilaterals. Overlapped elements at time zero can cause locking to occur in the contact leading to excessive distortions. The considerable energy input to the bag will create numerical noise and some damping is recommended to avoid problems.

3. **The dummy passes through the airbag.**

A most likely problem is that the contacts are improperly defined. Another possibility is that the models were developed in an incompatible unit system. The extra check for penetration flag if set to 1 on the contact control cards variable PENCHK in the *CONTACT_... definitions may sometimes cause nodes to be prematurely released due to the softness of the penalties. In this case the flag should be turned off.

4. **The OSP fails to converge.**

This may occur when excessively large forces are passed to the OSP. First, check that unit systems are consistent and then look for improperly defined contacts in the LS-DYNA input.

5. Time step approaches zero.

This is almost always in the airbag. If elastic or orthotropic (*MAT_ELASTIC or *MAT_COMPOSITE material 1 or 22) is being used, then switch to fabric material *MAT_FABRIC which is less time step size sensitive and use the fully integrated membrane element. Increasing the damping in the control volume usually helps considerably. Also, check for “cuts” in the airbag where nodes are not merged. These can allow elements to deform freely and cut the time step to zero.

APPENDIX J: Interactive Graphics Commands

Only the first four or less characters of command are significant. These commands are available in the interactive phase of LS-DYNA. The interactive graphics are available by using the "SW5." command after invoking the Ctrl-C interrupt. The MENU command brings up a push button menu. Only available in Unix and Linux.

ANIMATE	Animate saved sequence, stop with switch 1.
BACK	Return to previous display size after zoom, then list display attributes.
BGC	Change display background color RGB proportions BGC <red> <green> <blue>.
BIP	Select beam integration point for contour; BIP <#>.
CENTER	Center model, center on node, or center with mouse, i.e., center cent <value> or cent gin.
CL	Classification labels on display; class commercial_in_confidence.
CMA	Color materials on limited color displays.
COLOR	Set or unset shaded coloring of materials.
CONTOUR	View with colored contour lines; contour <component #> <list mat #>; see TAURUS manual.
COOR	Get node information with mouse.
COP	Hardcopy of display on the PC copy <laserj paintj tekcol coljet or epson>.
CR	Restores cutting plane to default position.
CUT	Cut away model outside of zoom window; use mouse to set zoom window size.
CX	Rotate slice plane at zmin about x axis.
CY	Rotate slice plane at zmin about y axis.
CZ	Rotate slice plane at zmin about z axis.
DIF	Change diffused light level for material; DIF <mat #, -1 for all> <value>.

DISTANCE	Set distance of model from viewer; DIST <value in normalized model dimensions>.
DMATERIALS	Delete display of material in subsequent views; DMAT <ALL or list of numbers>.
DRAW	Display outside edges of model.
DSCALE	Scale current displacement from initial shape.
DYN	After using TAURUS command will reset display to read current DYNA3D state data.
ELPLT	Set or unset element numbering in subsequent views.
END	Delete display and return to execution.
ESCAPE	Escapes from menu pad mode.
EXECUTE	Return to execution and keep display active.
FCL	Fix or unfix current contour levels.
FOV	Set display field of view angle; FOV <value in degrees>.
FRINGE	View with colored contour fringes; fringe <component #> <list mat #>; see TAURUS manual.
GETFRAME	Display a saved frame; GETF <frame #>.
HARDWARE	Hardware mode; workstation hardware calls are used to draw, move and color model; repeat command to reset to normal mode.
HELP	
HZB	Switch on or off hardware zbuffer for a subsequent view, draw or contour command; rotations and translations will be in hardware.
LIMIT	Set range of node numbers subsequent views; limit <first node #> <last node #>.
MAT	Re-enable display of deleted materials mat <all or list of numbers>.
MENU	Button menu pad mode.

MOTION	Motion of model through mouse movement or use of a dial box. The left button down enables translation in the plane, middle button rotation about axes in the plane; and with right button down in the out of plane axis; left and middle button down quit this mode.
MOV	Drag picked part to new position set with mouse.
NDPLT	Set or unset node numbering in subsequent views.
NOFRAME	Set and unset drawing of a frame around the picture.
PAUSE	Animation display pause in seconds
PHS2 or THISTORY	Time history plotting phase. Similar to LS-TAURUS.
PICK	Get element information with mouse.
POST	Enable or disable postscript mode on the PC and eps file is written as picture is drawn; remove eofs and initgraphics for eps use.
QUIT	Same as execute.
RANGE	Set fix range for contour levels; range <minvalue> <maxvalue>.
RAX	Reflect model about xy plane; restore command will switch-off reflections.
RAY	Reflect model about yz plane; restore command will switch-off reflections.
RAZ	Reflect model about zx plane, restore command will switch-off reflections.
RESTORE	Restores model to original position, also switches off element and node numbers, slice capper, reflections and cut model.
RETURN	Exit.
RGB	Change color red green blue element <mat #> <red> <green> <blue>.
RX	Rotate model about x axis.
RY	Rotate model about y axis.
RZ	Rotate model about z axis.

SAVE	Set or unset saving of display for animation.
SEQUENCE	Periodic plot during execution; SEQ <# of cycles> <commands> EXE.
SHR	Shrink element facets towards centroids in subsequent views, shrink <value>.
SIP	Select shell integration point for contour; SIP <#>.
SLICE	Slice model a z-minimum plane; slice <value in normalized model dimension> this feature is removed after using restore. Slice enables internal details for brick elements to be used to generate new polygons on the slice plane.
SNORMAL	Set or unset display of shell direction normals to indicate topology order.
SPOT	Draw node numbers on model spot <first #> <last # for range>.
TAURUS	LS-DYNA database, TAU <state #>, or state <state #>, reads LS-TAURUS file to extract previous state data.
TRIAD	Set or unset display of axis triad.
TSHELL	Set or unset shell element thickness simulation in subsequent views.
TV	Change display type.
TX	Translates model along x axis.
TY	Translates model along y axis.
TZ	Translates model along z axis.
V	Display model using painters algorithm.
VECTOR v or d	View with vector arrows of velocity or displacement; <v> or <d>.
ZB	Switch on or off zbuffer algorithm for subsequent view; or draw commands.
ZIN	Zoom in using mouse to set display size and position.
ZMA	Set position of zmax plane; ZMAX <value in normalized model dimensions>.
ZMI	Set position of zmin plane; ZMIN <value in normalized model dimensions>.
ZOUT	Zoom out using mouse to set displays size expansion and position.

APPENDIX K: Interactive Material Model Driver

INTRODUCTION

The interactive material model driver in LS-DYNA allows calculation of the material constitutive response to a specified strain path. Since the constitutive model subroutines in LS-DYNA are directly called by this driver, the behavior of the constitutive model is precisely that which can be expected in actual applications. In the current implementation the constitutive subroutines for both shell elements and solid elements can be examined.

INPUT DEFINITION

The material model driver is invoked by setting the total number of beam, shell, and solid elements to zero in a standard LS-DYNA input file. The number of material model definitions should be set to one, the number of load curves should be nine, and the termination time to the desired length of the driver run. The complete state dump interval is interpreted as the time step to be used in the material model driver run. Plotting information is saved for every step of a driver run and sufficient memory is allocated to save this information in core for the interactive plotting phase.

The input deck consists only of the TITLE card, the CONTROL cards, one MATERIAL DEFINITION, and NINE LOAD CURVES describing the strain path should be defined. These nine curves define the time history of the displacement gradient components shown in Table K.1.

The velocity gradient matrix, L_{ij} , is approximated by taking the time derivative of the components in Table K.1. If these components are considered to form a tensor S_{ij} , then

$$L_{ij}(t) = \frac{S_{ij}(t) - S_{ij}(t_{k-1})}{(t - t_k)}$$

and the strain rate tensor is defined as

$$d_{ij} = \frac{L_{ij} + L_{ij}^t}{2}$$

and the spin tensor as

$$\omega_{ij} = \frac{L_{ij} - L_{ij}^t}{2}$$

Table K.1 Load Curve Definitions versus Time

Load Curve Number	Component Definition
1	$\frac{\partial u}{\partial x}$
2	$\frac{\partial v}{\partial y}$
3	$\frac{\partial w}{\partial z}$
4	$\frac{\partial u}{\partial y}$
5	$\frac{\partial v}{\partial x}$
6	$\frac{\partial u}{\partial z}$
7	$\frac{\partial w}{\partial x}$
8	$\frac{\partial v}{\partial z}$
9	$\frac{\partial w}{\partial y}$

INTERACTIVE DRIVER COMMANDS

After reading the input file and completing the calculations, LS-DYNA gives a command prompt to the terminal. A summary of the available interactive commands is given below. An on-line help package is available by typing HELP. Only available in Unix and Linux.

ACCL	Scale all abscissa data by f. Default is f=1.
ASET amin omax	Set min and max values on abscissa to amin and amax, respectively. If amin=amax=0, scaling is automatic.
CHGL n	Change label for component n. LS-DYNA prompts for new label.
CONTINUE	Re-analyze material model.
CROSS c1 c2	Plot component c1 versus c2.
ECOMP	Display component numbers on the graphics display: 1 x-stress, 2 y-stress, 3 z-stress, 4 xy-stress, 5 yz-stress, 6 zx-stress, 7 effective plastic strain, 8 pressure, 9 von Mises (effective) stress, 10 1st principal deviatoric stress, 11 2nd principal deviatoric stress, 12 3rd principal deviatoric stress, 13 maximum shear stress, 14 1st principal stress, 15 2nd principal stress, 16 3rd principal stress, 17 $\ln(v/v_0)$, 18 relative volume, 19 $v_0/v - 1.0$, 20 1st history variable, 21 2nd history variable. Adding 100 or 400 to component numbers 1-16 yields strains and strain rates, respectively.
FILE name	Change pampers filename to name for printing.
GRID	Graphics displays will be overlaid by a grid of orthogonal lines.

NOGRID	Graphics displays will not be overlaid by a grid of orthogonal lines.
OSCL	Scale all ordinate data by f. Default is f=1.
OSET omin omax	Set min and max values on ordinate to omin and omax, respectively. If omin=omax=0, scaling is automatic.
PRINT	Print plotted time history data into file "pampers." Only data plotted after this command is printed. File name can be changed with the "file" command.
QUIT, END, T	Exit the material model driver program.
RDLC m n r ₁ z ₁ ... r _n z _n	Redefine load curve m using n coordinate pairs (r ₁ ,z ₁), (r ₂ ,z ₂),...,(r _n ,z _n).
TIME c	Plot component c versus time.
TV n	Use terminal output device type n. LS-DYNA provides a list of available devices.

Presently, the material model drive is implemented for solid and shell element material models. The driver does not yet support material models for beam elements.

APPENDIX L: VDA Database

VDA surfaces describe the surface of geometric entities and are useful for the simulation of sheet forming problems. The German automobile and automotive supplier industry (VDA) has defined the VDA guidelines [VDA 1987] for a proper surface definition used for the exchange of surface data information. In LS-DYNA, this format can be read and used directly. Some files have to be provided for proper linkage to the motion of the correlation parts/materials in LS-DYNA.

Linking is performed via names. To these names surfaces are attached, which in turn can be linked together from many files externally to LS-DYNA. Thus, arbitrary surfaces can be provided by a preprocessor and then can be written to various files. The so-called VDA file given on the LS-DYNA execution line via `V=vda` contains references to all other files. It also contains several other parameters affecting the treatment in the contact subroutines; see below.

The procedure is as follows. If VDA surfaces are to be used, the file specified by `vda` must have the following form. The file is free formatted with blanks as delimiters. Note that the characters “}” and “{” must be separated from the other input by spaces or new lines. The `vda` file may contain any number of input file specifications of the form:

```
file afile bfile {
```

```
    alias definitions
```

```
    }
```

```
    alias definitions
```

```
    followed by optional runtime parameters and a final end statement.
```

The file, `afile`, is optional, and if given must be the name of an ASCII input file formatted in accordance with the VDA Surface Interface Definitions as defined by the German automobile and automotive supply industry. `bfile` is required, and is the name of a binary VDA file. In a first run `afile` is given and `bfile` is created. In any further run, if the definitions have not changed, `afile` can be dropped and only `bfile` is needed. The purpose of `bfile` is that it allows for much faster initialization if the same VDA surfaces are to be used in a future LS-DYNA run.

If `afile` is given, `bfile` will always be created or overwritten. The alias definitions are used for linking to LS-DYNA and between the various surface definitions in the files defined by `afile` and `bfile`.

The alias definitions are of the form

```
alias name { e11 e12 ... e1n }
```

where `name` is any string of up to 12 characters, and `e11,...,e1n` are the names of VDA elements as specified in `afile`. The list of elements can be empty, in which case all the SURF and FACE VDA elements in `afile` will be used. Care should be taken to ensure that the alias `name` is unique, not only among the other aliases, but among the VDA element names in `afile`. This

collection of VDA elements can later be indicated by the alias **name**. In particular, **name** may appear in later alias definitions.

Often it is required that a punch or die be created by a simple offset. This can be achieved in the **vda** files in two ways, either on VDA elements directly, or on parts defined by aliases. This feature offers great capability in generating and using surface data information.

Offset version 1:

As an option, the keyword **offset** may appear in the alias list which allows a new surface to be created as a normal offset (plus translation) of a VDA element in the file. The keyword **offset** may be applied to VDA elements only, not aliases. The usage of **offset** follows the form

```
offset elem normal x y z
```

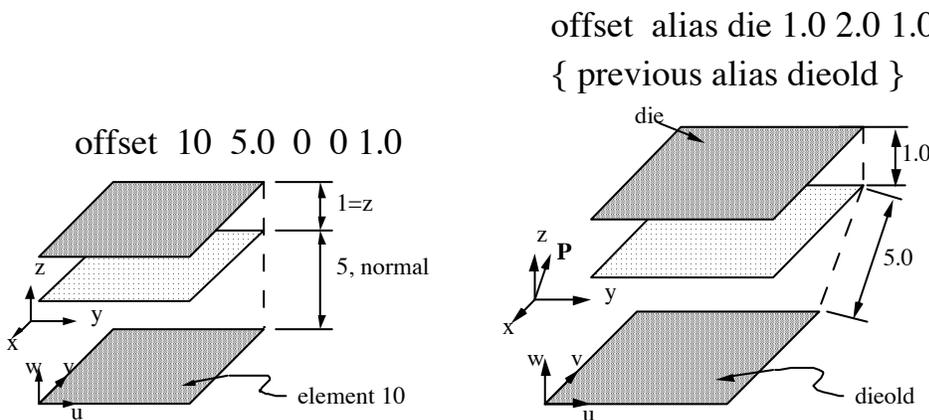
where **normal** is the amount to offset the surface along the normal direction, and **x,y,z** are the translations to be applied. The default normal direction is given by the cross product of the local u and v directions on the VDA surface, taken in that order. **normal** can be negative.

Offset version 2:

Frequently, it is convenient to create a new alias **name** by offsetting and translating an existing **name**. The keyword **goffset** provides this function:

```
goffset alias_name x_c y_c z_c normal x y z { previous alias_name }
```

where **normal**, **x**, **y**, and **z** are defined as in the offset keyword. A reference point **x_c**, **y_c**, and **z_c** defines a point in space which determines the normal direction to the VDA surface, which is a vector from the origin to P(x_c,y_c,z_c). See example below.



Finally, several parameters affecting the VDA surface iteration routines can be reset in the file **vda**. These parameters, and their default values in square brackets [], are:

- gap** [5.0] The maximum allowable surface gap to be filled in during the iterations. Points following the surface will effectively extend the edges of surfaces if necessary to keep them from falling through cracks in the surface smaller than this. This number should be set as small as possible while still allowing correct results. In particular, if your VDA surfaces are well formed (having no gaps), this parameter can be set to 0.0. The default value is 5.0.
- track** [2.0] A point must be within this distance of contact to be continually tracked. When a point not being tracked comes close to a surface, a global search is performed to find the near surface point. While a point is being tracked, iterations are performed every cycle. These iterations are much faster, but if the point is far away it is faster to occasionally do the global search. The default value is 2.0.
- track2** [5.0] Every VDA surface is surrounded by a bounding box. When a global search needs to be performed but the distance from a point to this box is $>$ **track2**, the actual global search is not performed. This will require another global search to be performed sooner than if the actual distance to the surface were known, but also allows many global searches to be skipped. The default value is 5.0.
- ntrack** [4] The number of VDA surfaces for which each point maintains actual distance information. A global lower bound on distance is maintained for all remaining surfaces. Whenever the point moves far enough to violate this global lower bound, all VDA surfaces must have the global search performed for them. Hence, this parameter should be set to the maximum number of surfaces that any point can be expected to be near at one time (the largest number of surfaces that come together at one point). Setting **ntrack** higher will require more memory but result in faster execution. If **ntrack** is too low, performance may be unacceptably slow. The default value is 4.0.
- toroid** [.01] Any surface with opposing edges which are within distance [t] of each other is assumed to be cylindrical. Contacts occurring on one edge can pass to the adjacent edge. The default value is 0.01.
- converge** [.01] When surface iterations are performed to locate the near point, iteration is continued until convergence is detected to within this distance (all VDA coordinates are in mm). The default value is 0.01.
- iterate** [8] Maximum number of surface iterations allowed. Since points being tracked are checked every cycle, if convergence fails it will be tried again next cycle, so setting this parameter high does not necessarily help much. On the other hand, a point converging to a crease in the VDA surface (a crease between patches with discontinuous derivative, for example) may bounce back and forth between patches up to this many times, without actually moving. Hence, this value should not be too large. The default value is 8.
- el_size** [t mx mn]

Controls the generation of elements where:

t =surface tolerance for mesh generation,
 mx=maximum element size to generate,
 mn=minimum element size to generate.

The default values are [0.25 100. 1.0]

aspect [s1 s2] Controls the generation of elements where:

s1=maximum difference in aspect ratio between elements generated in
 neighboring VDA patches,
 s2=maximum aspect ratio for any generated element.

The default values are [1.5 4.0]

cp_space [10] Determines the spacing around the boundaries of parts at which the size of elements is controlled. In the interior of the part, the element size is a weighted function of these control points as well as additional control points in the interior of the region. If there are too few control points around the boundary, elements generated along or near straight boundaries, but between control points, may be too small. The default value is 10.

meshonly The existence of this keyword causes LS-DYNA to generate a file containing the mesh for the VDA surfaces and then terminate.

onepatch The existence of this keyword causes LS-DYNA to generate a single element on each VDA patch.

somepatch [n] Like onepatch, but generates an element for 1 out of every [n] patches.

Example for file V=**vda**. It contains the following data:

```

file vda1 vda1.bin {
  alias die {
    sur0001
    sur0003
    offset fce0006 1.5 0 0 120
  }
  alias holder1 { sur008 }
}
file vda2 vda2.bin {
  alias holder2 { sur003 }
}
alias holder { holder1 holder2 }
ntrack 6
gap 0.5

end

```

Explanation:

vda1	This file contains the surfaces/face elements sur0001,sur0003, fce0006, and sur0008.
alias die face	Combines the surface/face elements sur0001, sur0003, and the offsetted fce0006 to a global surface.
alias holder1	Defines the surface/face element sur0008 as holder1.
vda2	This file contains the surface/face element sur0003.
alias holder2	Defines the surface/face element sur0003 as holder2.
alias holder	Combines the surfaces holder1 and holder2 into a combined surface holder.
ntrack 6	For each point the actual distances to 6 VDA surfaces are maintained.
gap 0.5	Surface gaps of 0.5mm or less are filled.
end	Closes reading of this file.

APPENDIX M: Commands for Two-Dimensional Rezoning

The rezoner in LS-DYNA contains many commands that can be broken down into the following categories:

- general,
- termination of interactive rezoning,
- redefinition of output intervals for data,
- graphics window controls,
- graphics window controls for x versus y plots,
- mesh display options,
- mesh modifications,
- boundary modifications,
- MAZE line definitions,
- calculation graphics display control parameters,
- calculation graphics display,
- cursor commands.

The use of the rezoner is quite simple. Commands for rezoning material number *n* can be invoked after the material is specified by the “M *n*” command. To view material *n*, the command “V” is available. The interior mesh can be smoothed with the “S” command and the boundary nodes can be adjusted after the “B” command is used to display the part side and boundary node numbers. Commands that are available for adjusting boundary nodes following the “B” command include:

ER, EZ, ES, VS, BD, ERS, EZS, ESS, VSS, BDS, SLN, SLNS

Rezoning is performed material by material. An example is shown.

Do not include the graphics display type number (see the “TV” command below) when setting up a command file for periodic noninteractive rezoning. No plotting is done when the rezoner is used in this mode.

REZONING COMMANDS BY FUNCTION

Interactive Real Time Graphics

SEQ n commands EXE Every n time steps execute the graphics commands which follow. For example the line seq 100 g exe would cause the grid to be updated on the graphics display device every 100 cycles. The real time graphics can be terminated by using ctrl-c and typing "sw7."

General

C Comment - proceed to next line.

FRAME Frame plots with a reference grid (default).

HELP Enter HELP package and display all available commands. Description of each command is available in the HELP package.

HELP/commandname Do not enter HELP package but print out the description on the terminal of the command following the slash.

LOGO Put LLNL logo on all plots (default). Retyping this command removes the logo.

NOFRAME Do not plot a reference grid.

PHP ans Print help package - If answer equals 'y' the package is printed in the high speed printer file.

RESO n_x n_y Set the x and y resolutions of plots to n_x and n_y , respectively. We default both n_x and n_y to 1024.

TV n Use graphics output device type n. The types are installation dependent and a list will be provided after this command is invoked.

TR t At time t, LS-DYNA will stop and enter interactive rezoning phase.

Termination of Interactive Rezoning

F Terminate interactive phase, remap, continue in execution phase.

FR	Terminate interactive phase, remap, write restart dump, and call exit.
T or END	Terminate.

Redefinition of Output Intervals for Data

PLTI Δt	Reset the node and element data dump interval Δt .
PRTI Δt	Reset the node and element printout interval Δt .
TERM t	Reset the termination to t.

Graphics Window Controls

ESET n	Center picture at element n with a Δr by Δz window. This window is set until it is released by the unfix command or reset with another window.
FF	Encircle picture with reference grid with tickmarks. Default grid is plotted along bottom and left side of picture.
FIX	Set the display to its current window. This window is set until it is reset by the “GSET”, “FSET”, or “SETF” commands or released by the “UNFIX” command.
FSET n Δr Δz	Center display at node n with a rectangular $\Delta r \times \Delta z$ window. This window is set until it is reset with or the “UNFIX” command is typed.
GSET r z Δl	Center display picture at point (r,z) with square window of width Δl . This window is set until it is reset or the “UNFIX” command is typed.
GRID	Overlay graphics displays with a grid of orthogonal lines.
NOGRID	Do not overlay graphics displays with a grid of orthogonal lines (default).
SETF r z Δr Δz	Center display at point (r,z) with a rectangular $\Delta r \times \Delta z$ window. This window is set until it is reset or the “UNFIX” command is typed.
UNFIX	Release current display window set by the “FIX”, “GSET”, “FSET” or “SETF” commands.

UZ a b Δl	Zoom in at point (a,b) with window Δl where a, b, and Δl are numbers between 0 and 1. The picture is assumed to lie in a unit square.
UZG	Cover currently displayed picture with a 10 by 10 square grid to aid in zooming with the unity zoom, “UZ”, command.
UZOU a b Δl	Zoom out at point (a,b) with window Δl where a, b, and Δl are numbers between 0 and 1. The current window is scaled by the factor $1/\Delta l$. The picture is assumed to lie in a unit square.
Z r z Δl	Zoom in at point (r,z) with window Δl .
ZOUT r z Δl	Zoom out at point (r,z) with window Δl . The window is enlarged by the ratio of the current window and Δl . The cursor may be used to zoom out via the cursor command DZOU and entering two points with the cursor to define the window. The ratio of the current window with the specified window determines the picture size reduction. An alternative cursor command, DZZO, may be used and only needs one point to be entered at the location where the reduction (2 \times) is expected.

Graphics Window Controls for x versus y plots

The following commands apply to line plots, interface plots, etc.

ASCL f_a	Scale all abscissa data by f_a . The default is $f_a = 1$.
ASET amin amax	Set minimum and maximum values on abscissa to amin and amax, respectively. If amin=amax=0.0 (default) LS-DYNA determines the minimum and maximum values.
OSCL f_o	Scale all ordinate data by f_o . The default is $f_o = 1$.
OSET omin omx	Set minimum and maximum values on ordinate to omin and omx, respectively. If omin=omx=0.0 (default) LS-DYNA determines the minimum and maximum values.
SMOOTH n	Smooth a data curve by replacing each data point by the average of the 2n adjacent points. The default is n=0.

Mesh Display Options

ELPLT	Plot element numbers on mesh of material n.
FSOFF	Turn off the “FSON” command.

FSON	Plot only free surfaces and slideline interfaces with “O” command. (Must be used before “O” command.)
G	View mesh.
GO	View mesh right of centerline and outline left of centerline.
GS	View mesh and solid fill elements to identify materials by color.
M n	Material n is to be rezoned.
MNOFF	Do not plot material numbers with the “O”, “G”, and “GO” commands (default).
MNON	Plot material numbers with “O”, “G”, and “GO” commands.
NDPLT	Plot node numbers on mesh of material n.
O	Plot outlines of all material.
RPHA	Reflect mesh, contour, fringe, etc., plots about horizontal axis. Retyping “RPHA” turns this option off.
RPVA	Reflect mesh, contour, fringe, etc., plots about vertical axis. Retyping “RPVA” turns this option off.
TN r z Δl	Type node numbers and coordinates of all nodes within window ($r \pm \Delta l/2$, $z \pm \Delta l/2$).
UG	Display undeformed mesh.
V	Display material n on graphics display. See command M.
VSF	Display material n on graphics display and solid fill elements.

Mesh Modifications

BACKUP	Restore mesh to its previous state. This command undoes the result of the last command.
BLEN s	Smooth option where $s=0$ and $s=1$ correspond to equipotential and isoparametric smoothing, respectively. By letting $0 \leq s \leq 1$ a combined blending is obtained.
CN m r z	Node m has new coordinate (r,z).

DEB n f ₁ l ₁ ... f _n l _n	Delete n element blocks consisting of element numbers f ₁ to l ₁ , f ₂ to l ₂ ... , and f _n to l _n inclusive. These elements will be inactive when the calculation resume.
DE e ₁ e ₂	Delete elements e ₁ to e ₂ .
DMB n m ₁ m ₂ ... m _n	Delete n material blocks consisting of all elements with material numbers m ₁ , m ₂ ,..., and m _n . These materials will be inactive when the calculations resume.
DM n m ₁ m ₂ ... m _n	Delete n materials including m ₁ , m ₂ ,..., and m _n .
DZER k d incr nrow	Delete element row where k is the kept element, d is the deleted element, incr is the increment, and nrow is the number of elements in the row.
DZLN number n ₁ n ₂ n ₃ ...n _{last}	Delete nodal row where number is the number of nodes in the row and n ₁ , n ₂ , ... n _{last} are the ordered list of deleted nodes.
DZNR l j incr	Delete nodal row where l is the first node in the row, j is the last node in the row, and incr is the increment.
R	Restore original mesh.
S	Smooth mesh of material n. To smooth a subset of elements, a window can be set via the "GSET", "FSET", OR "SETF" commands. Only the elements lying within the window are smoothed.

Boundary Modifications

A	Display all slidelines. Slave sides are plotted as dashed lines.
B	Determine boundary nodes and sides of material n and display boundary with nodes and side numbers.
BD m n	Dekink boundary from boundary node m to boundary node n (counterclockwise).
BDS s	Dekink side s.
DSL n l ₁ l ₂ ...l _n	Delete n slidelines including slideline numbers l ₁ l ₂ ..., and l _n .
ER m n	Equal space in r-direction boundary nodes m to n (counterclockwise).
ERS s	Equal space in the r-direction boundary nodes on side s.

ES m n	Equal space along boundary, boundary nodes m to n (counterclockwise).
ESS s	Equal space along boundary, boundary nodes on side s.
EZ m n	Equal space in z-direction boundary nodes m to n (counterclockwise).
EZS s	Equal space in the z-direction boundary nodes on side s.
MC n	Check master nodes of slideline n and put any nodes that have penetrated through the slave surface back on the slave surface.
MD n	Dekink master side of slideline n. After using this command, the SC or MC command is sometimes advisable.
MN n	Display slideline n with master node numbers.
SC n	Check slave nodes of slideline n and put any nodes that have penetrated through the master surface back on the master surface.
SD n	Dekink slave side of slideline n; after using this command, the SC or MC command is sometimes advisable.
SLN m n	Equal space boundary nodes between nodes m to n on a straight line connecting node m to n.
SLNS n	Equal space boundary nodes along side n on a straight line connecting the corner nodes.
SN n	Display slideline n with slave node numbers.
VS m n r	Vary the spacing of boundary nodes m to n such that r is the ratio of the first segment length to the last segment length.
VSS s r	Vary the spacing of boundary nodes on side s such that r is the ratio of the first segment length to the last segment length.

MAZE Line Definitions

B	Determine boundary nodes and sides of material n and display boundary with nodes and side numbers. See command "M".
LD n k l	Line definition n for MAZE includes boundary nodes k to l
LDS n l	Line definition n for MAZE consists of side number l.
M n	Material n is active for the boundary command B.

Calculation Graphics Display Control Parameters

MOLP	Overlay the mesh on the contour, fringe, principal stress, and principal strain plots. Retyping “MOLP” turns this option off.
NLOC	Do not plot letters on contour lines.
NUMCON n	Plot n contour levels. The default is 9.
PLOC	Plot letters on contour lines to identify their levels (default).
RANGE r ₁ r ₂	Set the range of levels to be between r ₁ and r ₂ instead of in the range chosen automatically by LS-DYNA. To deactivate this command, type RANGE 0 0.

Calculation Graphics Display

CONTOUR c n m ₁ m ₂ ...m _n	Contour component number c on n materials including materials m ₁ , m ₂ , ..., m _n . If n is zero, only the outline of material m ₁ with contours is plotted. Component numbers are given in Table M.1.
FRINGE c n m ₁ m ₂ ...m _n	Fringe component number c on n materials including m ₁ , m ₂ ,...,m _n . If n is zero, only the outline of material m ₁ with contours is plotted. Component numbers are given in Table M.1.
IFD n	Begin definition of interface n. If interface n has been previously defined, this command has the effect of destroying the old definition.
IFN l m	Include boundary nodes l to m (counterclockwise) in the interface definition. This command must follow the “B” command.
IFP c m	Plot component c of interface m. Component numbers are given in Table M.2.
IFS m	Include side m in the interface definition. Side m is defined for material n by the “B” command.
IFVA r _c z _c	Plot the angular location of the interface based on the center point (r _c ,z _c) along the abscissa. Positive angles are measured counterclockwise from the y-axis.
IFVS	Plot the distance along the interface from the first interface node along the abscissa (default).

LINE c n m ₁ m ₂ ...m _n	Plot variation of component c along line defined with the “NLDF”, “PLDF”, “NSDF”, or the “NSSDF” commands given below. In determining variation, consider n materials including material number m ₁ , m ₂ ,...m _n .
NCOL n	Number of colors in fringe plots is n. The default value for n is 6 which includes colors magenta, blue, cyan, green, yellow, and red. An alternative value for n is 5 which eliminates the minimum value magenta.
NLDF n n ₁ n ₂ ...n ₃	Define line for “LINE” command using n nodes including node numbers n ₁ , n ₂ ,...n _n . This line moves with the nodes.
NSDF m	Define line for “LINE” command as side m. Side m is defined for material n by the “B” command.
NSSDF l m	Define line for “LINE” command and that includes boundary nodes l to m (counterclockwise) in the interface definitions. This command must follow the “B” command.
PLDF n r ₁ z ₁ ...r _n z _n	Define line for “LINE” command using n coordinate pairs (r ₁ ,z ₁), (r ₂ ,z ₂), ...(r _n ,z _n). This line is fixed in space.
PRIN c n m ₁ m ₂ ...m _n	Plot lines of principal stress and strain in the yz plane on n materials including materials m ₁ , m ₂ ,...,m _n . If n is zero, only the outline of material m ₁ is plotted. The lines are plotted in the principal stress and strain directions. Permissible component numbers in Table M.1 include 0, 5, 6, 100, 105, 106,...,etc. Orthogonal lines of both maximum and minimum stress are plotted if components 0, 100, 200, etc. are specified.
PROFILE c n m ₁ m ₂ ...m _n	Plot component c versus element number for n materials including materials m ₁ , m ₂ ,...,m _n . If n is 0 then component c is plotted for all elements. Component numbers are given in Table M.1.
VECTOR c n m ₁ m ₂ ...m _n	Make a vector plot of component c on n materials including materials m ₁ , m ₂ ,...,m _n . If n is zero, only the outline of material m ₁ with vectors is plotted. Component c may be set to “D” and “V” for vector plots of displacement and velocity, respectively.

No.	Component	No.	Component
1	y	21*	ln (V/Vo) (volumetric strain)
2	z	22*	y-displacement
3	hoop	23*	z-displacement
4	yz	24*	maximum displacement
5	maximum principal	25*	y-velocity, y-heat flux
6	minimum principal	26*	z-velocity, y-heat flux
7	von Mises (Appendix A)	27*	maximum velocity, maximum heat flux
8	pressure or average strain	28	ij normal
9	maximum principal-minimum principal	29	jk normal
10	y minus hoop	30	kl normal
11	maximum shear	31	li normal
12	ij and kl normal (Appendix D)	32	ij shear
13	jk and li normal	33	jk shear
14	ij and kl shear	34	kl shear
15	jk and li shear	35	li shear
16	y-deviatoric	36*	relative volume V/Vo
17	z-deviatoric	37*	VoV-1
18	hoop-deviatoric	38*	bulk viscosity, Q
19*	effective plastic strain	39*	P + Q
20*	temperature/internal energy density	40*	density
41*-70*	element history variables		
71*	r-peak acceleration	76*	peak value of min in plane prin. stress
72*	z-peak acceleration	77*	peak value of maximum hoop stress
73*	r-peak velocity	78*	peak value of minimum hoopstress
74*	z-peak velocity	79*	peak value of pressure
75*	peak value of max. in plane prin. stress		

Table M.1. Component numbers for element variables. By adding 100, 200 300, 400, 500 and 600 to the component numbers not followed by an asterisk, component numbers for infinitesimal strains, lagrange strains, almanshi strains, strain rates, extensions, and residual strain are obtained. Maximum and minimum principal stresses and strains are in the rz plane. The corresponding hoop quantities must be examined to determine the overall extremum. ij, jk, etc. normal components are normal to the ij, jk, etc side. The peak value database must be flagged on Control Card 4 in columns 6-10 or components 71-79 will not be available for plotting.

No.	Component
1	pressure
2	shear stress
3	normal force
4	tangential force
5	y-force
6	z-force

Table M.2. Component numbers for interface variables. In axisymmetric geometries the force is per radian.

Cursor Commands

DBD a b	Use cursor to define points a and b on boundary. Dekink boundary starting at a, moving counterclockwise, and ending at b.
DCN a b	Use cursor to define points a and b. The node closest to point a will be moved to point b.
DCSN n a	Move nodal point n to point a defined by the cursor.
DCNM a b	Use cursor to define points a and b. The node at point a is given the coordinate at point b.
DER a b	Use cursor to define points a and b on boundary. Equal space nodes in r-direction along boundary starting at a, moving counterclockwise, and ending at b.
DES a b	Use cursor to define points a and b on boundary. Equal space nodes along boundary starting at a, moving counterclockwise, and ending at b.
DEZ a b	Use cursor to define points a and b on boundary. Equal space nodes in z-direction along boundary starting at a, moving counterclockwise, and ending at b.
DTE a b	Use cursor to define points a and b on the diagonal of a window. The element numbers and coordinates of elements lying within the window are typed on the terminal.
DTN a b	Use cursor to define points a and b on the diagonal of a window. The node numbers and coordinates of nodal points lying within the window are typed on the terminal.

DTNC a	Use cursor to define point a. The nodal point number and nodal coordinates of the node lying closest to point a will be printed.
DVS a b r	Use cursor to define points a and b on boundary. Variable space nodes along boundary starting at a, moving counterwise, and ending at b. The ratio of the first segment length to the last segment length is give by r (via terminal).
DZ a b	Use cursor to define points a and b on the diagonal of a window for zooming.
DZOUT a b	Enter two points with the cursor to define the window. The ratio of the current window with the specified window determines the picture size reduction.
DZZ a	Use cursor to define point a and zoom in at this point. The new window is .15 as large as the previous window. The zoom factor can be reset by the crzf command for the .15 default.
DZZO a	Zoom out at point a by enlarging the picture two times.

APPENDIX N: Rigid Body Dummies

The two varieties of rigid body dummies available in LS-DYNA are described in this appendix. These are generated internally by including the appropriate *COMPONENT keyword. A description of the GEBOD dummies begins on this page and the HYBRID III family on page N.7.

GEBOD Dummies

Rigid body dummies can be generated and simulated within LS-DYNA using the keyword *COMPONENT_GEBOD. Physical properties of these dummies draw upon the GEBOD database [Cheng et al. 1994] which represents an extensive measurement program conducted by Wright-Patterson AFB and other agencies. The differential equations governing motion of the dummy are integrated within LS-DYNA separate from the finite element model. Interaction between the dummy and finite element structure is achieved using contact interfaces (see *CONTACT_GEBOD).

The dynamical system representing a dummy is comprised of fifteen rigid bodies (segments) and include: lower torso, middle torso, upper torso, neck, head, upper arms, forearms/hands, upper legs, lower legs, and feet. Ellipsoids are used for visualization and contact purposes. Shown in Figure N.1 is a 50th percentile male dummy generated using the keyword command *COMPONENT_GEBOD_MALE. Note that the ellipsoids representing the shoulders are considered to be part of the upper torso segment and the hands are rigidly attached to the forearms.

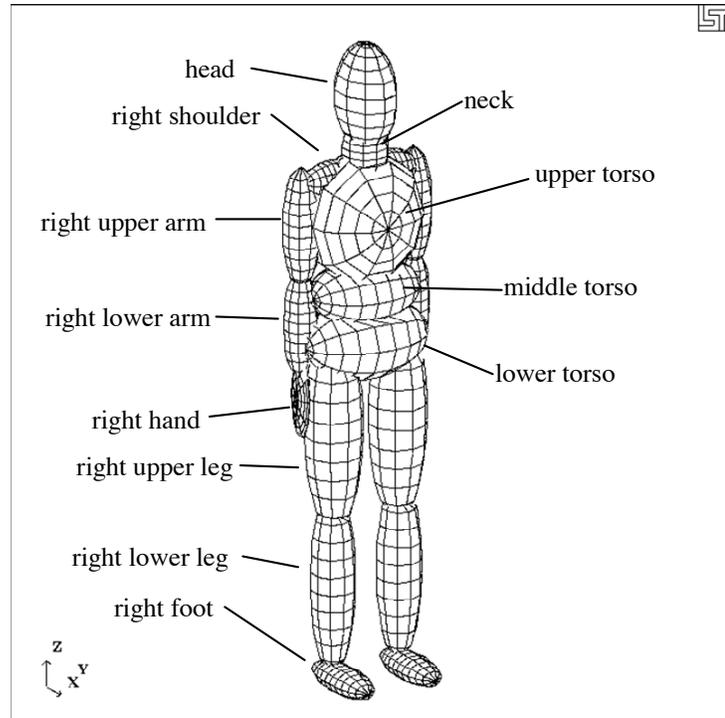


Figure N.1 50th percentile male dummy in the nominal position.

Each of the rigid segments which make up the dummy is connected to its neighbor with a joint which permits various relative motions of the segments. Listed in the Table N.1 are the joints and their applicable degrees of freedom.

Table N.1 Joints and associated degrees of freedom. Local axes are in parentheses.

Joint Name	Degree(s) of Freedom		
	1 st	2nd	3rd
pelvis	lateral flexion (x)	forward flexion (y)	torsion (z)
waist	lateral flexion (x)	forward flexion (y)	torsion (z)
lower neck	lateral flexion (x)	forward flexion (y)	torsion (z)
upper neck	lateral flexion (x)	forward flexion (y)	torsion (z)
shoulders	abduction-adduction (x)	internal-external rotation (z)	flexion-extension (y)
elbows	flexion-extension (y)	n/a	n/a
hips	abduction-adduction (x)	medial-lateral rotation (z)	flexion-extension (y)
knees	flexion-extension (y)	n/a	n/a
ankles	inversion-eversion (x)	dorsi-plantar flexion (y)	medial-lateral rotation (z)

Orientation of a segment is effected by performing successive right-handed rotations of that segment relative to its parent segment - each rotation corresponds to a joint degree of freedom. These rotations are performed about the local segment axes and the sequence is given in Table N.1. For example, the left upper leg is connected to the lower torso by the left hip joint; the limb is first abducted relative to lower torso, it then undergoes lateral rotation, followed by extension. The remainder of the lower extremity (lower leg and foot) moves with the upper leg during this orientation process.

By default all joints are assigned stiffnesses, viscous characteristics, and stop angles which should give reasonable results in a crash simulation. One or all default values of a joint may be altered by applying the **COMPONENT_GEBOD_JOINT_OPTION* command to the joint of interest. The default shape of the resistive torque load curve used by all joints is shown in Figure N.2. A scale factor is applied to this curve to obtain the proper stiffness relationship. Listed in Table N.2 are the default values of joint characteristics for dummies of all types and sizes. These values are given in the English system of units; the appropriate units are used if a different system is specified in card 1 of **COMPONENT_GEBOD_OPTION*.

Table N.2 Default joint characteristics for all dummies.

joint degrees of freedom	load curve scale factor (in-lbf)	damping coef. (in-lbf-s/rad)	low stop angle (degrees)	high stop angle (degrees)	neutral angle (degrees)
pelvis - 1	65000	5.77	-20	20	0
pelvis - 2	65000	5.77	-20	20	0
pelvis - 3	65000	5.77	-5	5	0
waist - 1	65000	5.77	-20	20	0
waist - 2	65000	5.77	-20	20	0
waist - 3	65000	5.77	-35	35	0
lower neck - 1	10000	5.77	-25	25	0
lower neck - 2	10000	5.77	-25	25	0
lower neck - 3	10000	5.77	-35	35	0
upper neck - 1	10000	5.77	-25	25	0
upper neck - 2	10000	5.77	-25	25	0
upper neck - 3	10000	5.77	-35	35	0
l. shoulder - 1	100	5.77	-30	175	0
r. shoulder - 1	100	5.77	-175	30	0
shoulder - 2	100	5.77	-65	65	0
shoulder - 3	100	5.77	-175	60	0
elbow - 1	100	5.77	1	-140	0
l. hip - 1	10000	5.77	-25	70	0
r. hip - 1	10000	5.77	-70	25	0
hip - 2	10000	5.77	-70	70	0
hip - 3	10000	5.77	-140	40	0
knee - 1	100	5.77	-1	120	0
l. ankle - 1	100	5.77	-30	20	0
l. ankle - 1	100	5.77	-20	30	0
ankle - 2	100	5.77	-20	45	0
ankle - 3	100	5.77	-30	30	0

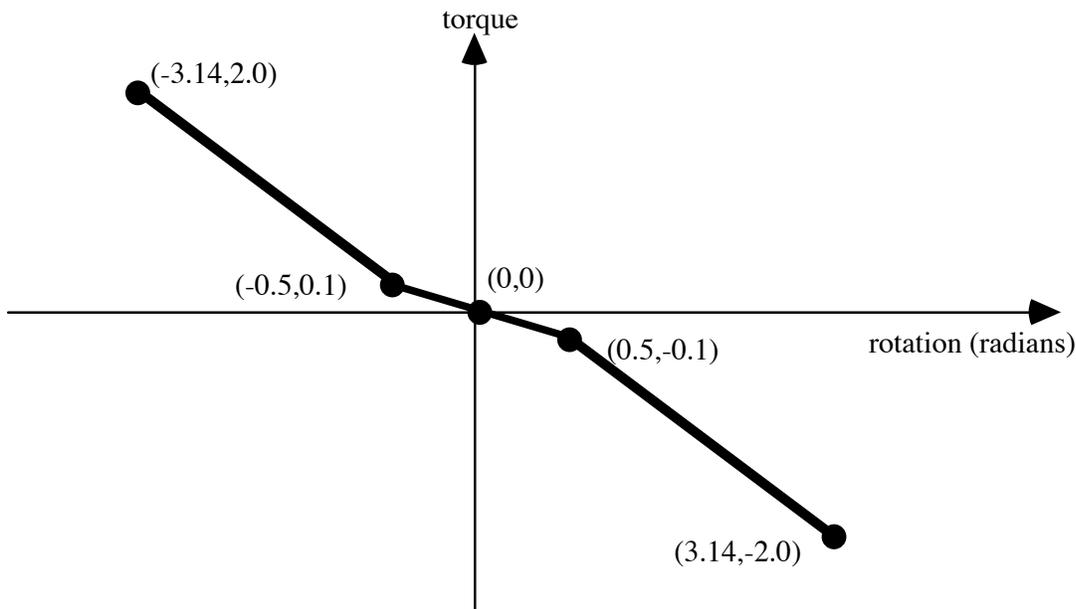


Figure N.2 Characteristic torque curve shape used by all joints.

The dummy depicted in Figure N.1 appears in what is referred to as its "nominal" position. In this position the dummy is standing upright facing in the positive x direction and the toe-to-head direction points in positive z. Additionally, the dummy's hands are at the sides with palms facing inward and the centroid of the lower torso is positioned at the origin of the global coordinate system. Each of the dummy's segments has a local coordinate system attached to it and in the nominal position all of the local axes are aligned with the global axes.

When performing a simulation involving a *COMPONENT_GEBOD dummy, a positioning file named "gebod.did" must reside in the directory with the LS-DYNA input file; here the extension *did* is the dummy ID number, see card 1 of *COMPONENT_GEBOD_OPTION. The contents of a typical positioning file is shown in Table N.3; it consists of 40 lines formatted as (59a1,e30.0). All of the angular measures are input as degrees, while the lower torso global positions depend on the choice of units in card 1 of *COMPONENT_GEBOD_OPTION. Setting all of the values in this file to zero yields the so-called "nominal" position.

Table N.3 Typical contents of a dummy positioning file.

lower torso	centroid global x position		0.0
lower torso	centroid global y position		0.0
lower torso	centroid global z position		0.0
total body	global x rotation		0.0
total body	global y rotation		-20.0
total body	global z rotation		180.0
pelvis	lateral flexion	+ = tilt right	0.0
pelvis	forward flexion	+ = lean fwd	0.0
pelvis	torsion	+ = twist left	0.0
waist	lateral flexion	+ = tilt right	0.0
waist	forward flexion	+ = lean fwd	0.0
waist	torsion	+ = twist left	0.0
lower neck	lateral flexion	+ = tilt right	0.0
lower neck	forward flexion	+ = nod fwd	0.0
lower neck	torsion	+ = twist left	0.0
upper neck	lateral flexion	+ = tilt right	0.0
upper neck	forward flexion	+ = nod fwd	0.0
upper neck	torsion	+ = twist left	0.0
left shoulder	abduction-adduction	+ = abduction	30.0
left shoulder	internal-external rotation	+ = external	-10.0
left shoulder	flexion-extension	- = fwd raise	-40.0
right shoulder	abduction-adduction	- = abduction	-30.0
right shoulder	internal-external rotation	- = external	10.0
right shoulder	flexion-extension	- = fwd raise	-40.0
left elbow	flexion-extension	+ = extension	-60.0
right elbow	flexion-extension	+ = extension	-60.0
left hip	abduction-adduction	+ = abduction	0.0
left hip	medial-lateral rotation	+ = lateral	0.0
left hip	flexion-extension	+ = extension	-80.0
right hip	abduction-adduction	- = abduction	0.0
right hip	medial-lateral rotation	- = lateral	0.0
right hip	flexion-extension	+ = extension	-80.0
left knee	flexion-extension	+ = flexion	50.0
right knee	flexion-extension	+ = flexion	50.0
left ankle	inversion-eversion	+ = eversion	0.0
left ankle	dorsi-plantar flexion	+ = plantar	0.0
left ankle	medial-lateral rotation	+ = lateral	0.0
right ankle	inversion-eversion	- = eversion	0.0
right ankle	dorsi-plantar flexion	+ = plantar	0.0
right ankle	medial-lateral rotation	- = lateral	0.0

In Figure N.3 the 50th percentile male dummy is shown in a seated position and some of its joints are labeled. The file listed in Table N.3 was used to put the dummy into the position shown. Note that the dummy was first brought into general orientation by setting nonzero values for two of the lower torso local rotations. This is accomplished by performing right-handed rotations successively about local axes fixed in the lower torso, the sequence of which follows: the first about local x, next about local y, and the last about local z. The dummy in Figure N.3 was made to pitch backward by setting "total body global y rotation" equal to -20. Setting the "total body global z rotation" equal to 180 caused the dummy to rotate about the global z-axis and face in the -x direction.

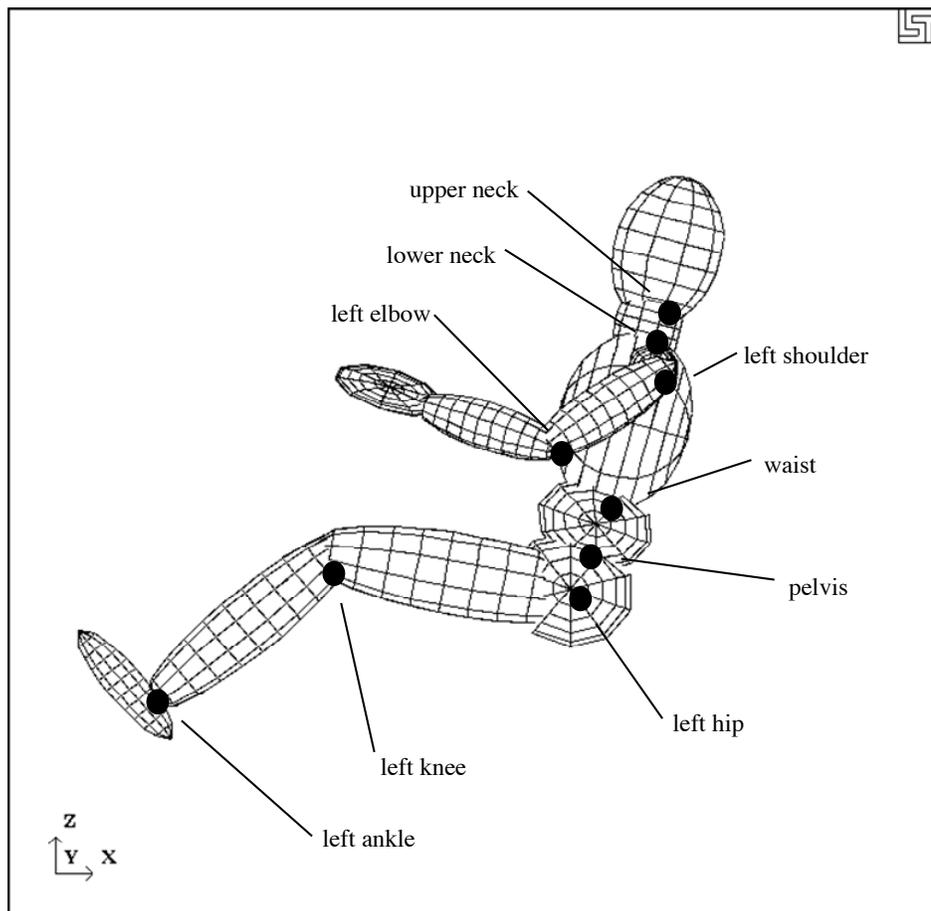


Figure N.3 Dummy seated using the file listed in Table N.3.

HYBRID III Dummies

A listing of applicable joint degrees of freedom of the Hybrid III dummy is given below.

Table N.4 Joints and associated degrees of freedom. Local axes are in parentheses.

Joint Name	Degree(s) of Freedom		
	1 st	2nd	3rd
lumbar	flexion (y)	torsion (z)	
lower neck	flexion (y)	torsion (z)	
upper neck	flexion (y)	torsion (z)	
shoulders	flexion-extension (y)	abduction-adduction (x)	n/a
elbows	flexion-extension (y)	n/a	n/a
wrists	flexion-extension (x)	n/a	n/a
hips	abduction-adduction (x)	medial-lateral rotation (z)	flexion-extension (y)
knees	flexion-extension (y)	n/a	n/a
ankles	inversion-eversion (x)	medial-lateral rotation (z)	dorsi-plantar flexion (y)
sternum	translation (x)	rotation (y)	rotation (z)
knee sliders	translation (z)		

Joint springs of the *COMPONENT_HYBRIDIII dummies are formulated in the following manner.

$$T = a_{lo}(q - q_{lo}) + b_{lo}(q - q_{lo})^3 \quad q \leq q_{lo}$$

$$T = a_{hi}(q - q_{hi}) + b_{hi}(q - q_{hi})^3 \quad q \geq q_{hi}$$

$$T = 0 \quad q_{lo} < q < q_{hi}$$

where

T is the joint torque

q is the joint generalized coordinate

a_{lo} and b_{lo} are the linear and cubic coefficients, respectively, for the low regime

a_{hi} and b_{hi} are the linear and cubic coefficients, respectively, for the high regime

q_{lo} and q_{hi} are the activation values for the low and high regimes, respectively

APPENDIX O: LS-DYNA MPP User Guide

This is a short user's guide for the MPP version of LS-DYNA. For a general guide to the capabilities of LS-DYNA and a detailed description of the input, consult the LS-DYNA User's Manual. If you have questions about this guide, find errors or omissions in it, please email manual@lstc.com.

Supported Features

The only input formats currently supported are 920 and later, including keyword. Models in any of the older formats will need to be converted to one of these input formats before they can be run with the current version of LS-DYNA for massively parallel processors, mpp.

The large majority of LS-DYNA options are available on MPP computers. Those that are not supported are being systematically added. Unless otherwise noted here, all the options of LS-DYNA version 93x are supported by MPP/LS-DYNA.

Here is the list of **unsupported** features:

- *BOUNDARY_THERMAL_WELD
- *BOUNDARY_USA_SURFACE
- *CONTACT_1D
- *DATABASE_AVS
- *DATABASE_MOVIE
- *DATABASE_MPGS
- *DATABASE_TRACER
- *DATABASE_BINARY_XTFIELD
- *INTERFACE_JOY
- *LOAD_SUPERELASTIC_OPTION
- *USER
- *TERMINATION_NODE

Contact Interfaces

MPP/LS-DYNA uses a completely redesigned, highly parallel contact algorithm. The contact options currently **unsupported** include:

- *CONTACT_TIED with birth time
- *CONTACT_FORCE_TRANSDUCER_CONSTRAINT

Because these options are all supported via the new, parallel contact algorithms, slight differences in results may be observed as compared to the serial and SMP versions of LS-DYNA. Work has been done to minimize these differences, but they may still be evident in some models.

For each of the supported CONTACT_ control cards, there is an optional string _MPP which can be appended to the end. Adding these characters triggers the reading of a new control card immediately following (but after the TITLE card, if any). This card contains 5 integer parameters in I10 format. The parameters are:

trackpen

If 1, any initial penetrations for slave nodes are compensated for in the contact algorithm. No nodes are moved to eliminate penetrations, and no initial penetration checking is performed. The algorithm detects these penetrations and allows for them in computing forces, so excessively large forces are avoided. As the slave node moves in such a way as to reduce or eliminate the penetration, the full contact distance/material thickness is imposed. Use of this option is encouraged as it can greatly help stability, particularly in models with many initial penetrations. By default this option is disabled.

bucket

Bucket sorting frequency for this contact interface

lcbucket

Load curve giving bucket sort frequency as a function of simulation time. Currently this option is not supported by any of the contact algorithms.

nseg2track

Number of contact segments to track for each slave node

inititer

Number of iterations for initial penetration checking

The defaults for each are taken from the corresponding options in the pfile (described below). For example, if you had the control card:

```
*CONTACT_SINGLE_SURFACE_TITLE  
This is my title card
```

you could change this to

```
*CONTACT_SINGLE_SURFACE_TITLE_MPP  
This is my title card
```

1

to turn on the initial penetration tracking option. The serial/SMP code will ignore these options.

Output Files and Post-Processing

For performance reasons, many of the ASCII output files normally created by LS-DYNA have been combined into a new binary format used by MPP/LS-DYNA. There is a post-processing program l2a, which reads this binary database of files and produces as output the corresponding ASCII files. The new binary files will be created in the directory specified as the global directory in the pfile (See section pfile). The files (up to one per processor) are named binout.nnnn, where nnnn is replaced by the four-digit processor number. To convert these files to ASCII simply feed them to the l2a program like this:

l2a binout*

LS-PREPOST is able to read the binout files directly, so conversion is not required, it is provided for backward compatibility.

The **supported** ASCII files are:

- *DATABASE_SECFORC
- *DATABASE_RWFORC
- *DATABASE_NODOUT
- *DATABASE_NODOUTHF
- *DATABASE_ELOUT
- *DATABASE_GLSTAT
- *DATABASE_DEFORC
- *DATABASE_MATSUM
- *DATABASE_NCFORC
- *DATABASE_RCFORC
- *DATABASE_SPCFORC
- *DATABASE_SWFORC
- *DATABASE_DEFGEO
- *DATABASE_ABSTAT
- *DATABASE_NODOFR
- *DATABASE_BNDOUT
- *DATABASE_GCEOUT
- *DATABASE_RBDOUT
- *DATABASE_SLEOUT
- *DATAGASE_JNTFORC
- *DATABASE_SBTOUT
- *DATABASE_SPHOUT
- *DATABASE_TPRINT

Some of the normal LS-DYNA files will have corresponding collections of files produced by MPP/LS-DYNA, with one per processor. These include the d3dump files (new names = d3dump.nnnn), the messag files (now mesnnnn) and others. Most of these will be found in the local directory specified in the pfile.

The format of the d3plot file has not been changed. It will be created in the global directory, and can be directly handled with your current graphics post-processor.

Parallel Specific Options

There are a few new command line options that are specific to the MPP version of LS-DYNA.

In the serial and SMP versions of LS-DYNA, the amount of memory required to run the problem can be specified on the command line using the keyword *memory=XXX*, where XXX is the number of words of memory to be allocated. For the MPP code, this will result in each processor allocating XXX words of memory. If pre-decomposition has not been performed, one processor must perform the decomposition of the problem. This can require substantially more memory than will be required once execution has started. For this reason, there is a second memory command line option, *memory2=YYY*. If used together with the memory keyword, the decomposing processor will allocate XXX words of memory, and all other processors will allocate YYY words of memory.

For example, in order to run a 250,000 element crash problem on 4 processors, you might need *memory=80m* and *memory2=20m*. To run the same problem on 16 processors, you still need *memory=80m*, but can set *memory2=6m*. The value for *memory2* drops nearly linearly with the number of processors used to run the program, which works well for shared-memory systems.

The full deck restart capability is supported by the MPP version of LS-DYNA, but in a manner slightly different than the SMP code. Each time a restart dump file is written, a separate restart file is also written with the base name D3FULL. For example, when the third restart file d3dump03 is written (one for each processor, d3dump03.0000, d3dump03.0001, etc), there is also a single file written named d3full03. This file is required in order to do a full deck restart and the d3dump files are not used in this case by the MPP code. In order to perform a full deck restart with the MPP code, you first must prepare a full deck restart input file as for the serial/SMP version. Then, instead of giving the command line option *r=d3dump03* you would use the special option *n=d3full03*. The presence of this command line option tells the MPP code that this is a restart, not a new problem, and that the file d3full03 contains the geometry and stress data carried over from the previous run.

PFILE

There is a new command line option: *p=pfile*. pfile contains MPP specific parameters that affect the execution of the program. The file is split into sections, with several options in each section. Currently, these sections: **directory**, **decomposition**, **contact**, and **general** are available. First, here is a sample pfile:

```
directory {  
  global rundir  
  local /tmp/rundir  
}  
contact {  
  inititer 3  
}
```

The file is case insensitive and free format input. The sections and options currently supported are:

- **directory.** Holds directory specific options

transfer_files

If this keyword is given, then processor 0 will write all output and restart files to the **global** directory (see “global” below), and scratch files to the **local** directory. All other processors will write all data to their **local** directory. At normal termination, all restart and output files will be copied from the processor specific **local** directories to the **global** directory. Also, if this is a restart from a dump file, the dump files will be distributed to the processors from the **global** directory. With this option enabled, there is no need for the processors to have shared access to a single disk for output – all files will be transferred as needed to and from the **global** directory.

Default = disabled.

global path

Path to a directory where program output should be written. If **transfer_files** is not given, this directory needs to be accessible to all processors – otherwise it is only accessed by processor 0. This directory will be created if necessary.

Default = current working directory

local path

Path to a processor specific local directory for scratch files. This directory will be created if necessary. This should be a local disk on each processor, for performance reasons.

Default = global path

rmlocal

If this keyword is given and **transfer_files** is active, the program attempts to clean up the **local** directories on each processor. In particular, it deletes files that are successfully transferred back to the **global** directory, and removes the **local** directory if it was created. It will not delete any files if there is a failure during file copying, nor will it delete directories it did not create.

Default = disabled

repository path

Path to a safe directory accessible from processor 0. This directory will be created if necessary. This is intended to be used as a safekeeping/backup of files during execution

and should only be used if **transfer_files** is also given. If this directory is specified then the following actions occur:

- At program start up, any required files (d3dump, binout, etc) that cannot be located in the **global** directory are looked for in the **repository** for copying to the **local** processor directories.
- Important output files (d3dump, runrsf, d3plot, binout and others) are synchronized to the repository regularly. That is, every time one of these files is updated on the node local or the global directories, a synchronized copy is updated in the repository.

The intension is that the repository be on a redundant disk, such as NAS, to allow restarting the problem if a hardware failure should occur on the machine running the problem. It must be noted that some performance penalty must be paid for the extra communication and I/O. Effort has been made to minimize this overhead, but this option is not recommended for general use.

Default = unspecified

- **decomposition** Holds decomposition specific options

file *filename*

The name of the file that holds the decomposition information. This file will be created in the current working directory if it does not exist. If the filename does not end with the extension *.pre* then this extension is added. If this option is not specified, MPP/LS-DYNA will perform the decomposition.

Default = None

numproc *n*

The problem will be decomposed for *n* processors. If $n > 1$ and you are running on 1 processor, or if the number of processors you are running on does not evenly divide *n*, then execution terminates immediately after decomposition. Otherwise, the decomposition file is written and execution continues. For a decomposition only run, both numproc and file should be specified.

Default = the number of processors you are running on.

method *name*

Currently, there are two decomposition methods supported, namely *rcb* and *greedy*. Method *rcb* is Recursive Coordinate Bisection. Method *greedy* is a simple neighborhood expansion algorithm. The impact on overall runtime is problem dependent, but *rcb* generally gives the best performance.

Default = rcb

region *rx ry rz sx sy sz c2r s2r 3vec mat*

See the section below on Special Decompositions for details about these decomposition options.

show

If this keyword appears in the decomposition section, the d3plot file is doctored so that the decomposition can be viewed with the post processor. Displaying material 1 will show that portion of the problem assigned to processor 0, and so on. The problem will

not actually be run, but the code will terminate once the initial d3plot state has been written.

rcblog filename

This option is ignored unless the decomposition method is RCB. A record is written to the indicated file recording the steps taken during decomposition. This is an ascii file giving each decomposition **region** (see the section on Special Decompositions) and the location of each subdivision for that **region**. This information can be placed in the **decomposition** section of the pfile for a subsequent problem, which will result in a decomposition as similar as possible between the two runs. For example, suppose a simulation is run twice, but the second time with a slightly different mesh. Because of the different meshes the problems will be distributed differently between the processors, resulting in slightly different answers due to roundoff errors. If an rcblog is used, then the resulting decompositions would be as similar as possible.

vspeed

If this option is specified a brief measurement is taken of the performance of each processor by timing a short floating point calculation. The resulting information is used during the decomposition to distribute the problem according to the relative speed of the processors. This might be of some use if the cluster has machines of significantly different speed.

automatic

If this option is given, an attempt is made to automatically determine a reasonable decomposition, primarily based on the initial velocity of nodes in the model. Use of the **show** option is recommended to verify a reasonable decomposition.

aledist

Distribute ALE elements to all processors.

dcmem n

It may be in some cases that the memory requirements during the first phase of decomposition are too high. If that is found to be the case (if you get out of memory errors during decomposition phase 1), then this may provide a work around. Specifying a value **n** here will cause some routines to process the model in blocks of **n** items, when normal processing would read the whole set (of nodes, elements, whatever) all at once. This will reduce memory requirements at the cost of greater communication overhead. Most users will not need this option. Values in the range of 10,000 to 50,000 would be reasonable.

- **contact**

This section has been largely replaced by the `_MPP` option on the normal contact card. The only remaining useful option here is:

alebkt n

Sets the bucket sort frequency for FSI (fluid structure interaction) to once every **n** cycles.
default = 50

- **general** Holds general options

nodump

If this keyword appears, all restart dump file writing will be suppressed: d3dump, runrsf, and d3full files will not be written.

nofull

If this keyword appears, writing of d3full (full deck restart) files will be suppressed.

nod3dump

If this keyword appears, writing of d3dump and runrsf files will be suppressed.

runrsfonly

If this keyword appears, writing of d3dump files will not occur – runrsf files will be written instead. Any time a d3dump OR runrsf file would normally be written, a runrsf file will be written.

nofail

If this keyword appears, the check for failed elements in the contact routines will be skipped. This can improve efficiency if you do not have element failure in the model.

swapbytes

If this keyword appears, the d3plot and interface component analysis files are written in swapped byte ordering.

nobeamout

Generally, whenever a beam, shell, or solid element fails, and element failure report is written to the d3hsp and message files. This can generate a lot of output. If this keyword appears, the element failure report is suppressed.

Special Decompositions

These options appear in the "decomposition" section of the pfile and are only valid if the decomposition method is **rcb**. The rcb decomposition method works by recursively dividing the model in half, each time slicing the current piece of the model perpendicularly to one of the three coordinate axes. It chooses the axis along which the current piece of the model is longest. The result is that it tends to generate cube shaped domains aligned along the coordinate axes. This is inherent in the algorithm, but is often not the behavior desired.

This situation is addressed by providing a set of coordinate transformation functions which are applied to the model before it is decomposed. The resulting deformed geometry is then passed to the decomposition algorithm, and the resulting domains are mapped back to the undeformed model. As a simple example, suppose you wanted rectangular domains aligned along a line in the xy plane, 30 degrees from the x axis, and twice as long along this line as in the other two dimensions. If you applied these transformations:

```
sx 0.5  
rz -30
```

then you would achieve the desired effect.

Furthermore, it may be desirable for different portions of the model to be decomposed differently. It is now possible to specify different regions of the model to be decomposed with different transformations. The general form for a special decomposition would look like this:

```
decomposition {
  region { <region specifiers> <transformation> <grouping> }
  region { <region specifiers> <transformation> <grouping> }
  <transformation>
}
```

Where the region specifiers are logical combinations of **box**, **sphere**, **cylinder**, **parts**, and **silist**. The transformation is a series of **sx**, **sy**, **sz**, **rx**, **ry**, **rz**, **c2r**, **s2r**, **3vec**, and **mat**. The grouping is either **lumped** or empty. The portion of the model falling in the first region will be decomposed according to the given transformation. Any remaining part of the model in the second region will then be treated, and finally anything left over will be decomposed according to the final transformation. Any number of regions may be given, including 0. Any number of transformations may be specified. They are applied to the region in the order given.

The region specifiers are:

box xmin xmax ymin ymax zmin zmax

A box with the given extents.

sphere xc yc zc r

The sphere centered at **(xc,yc,zc)** and having radius **r**. If **r** is negative it is treated as infinite.

cylinder xc yc zc ax ay az r d

A cylinder with center at **(xc,yc,zc)** and radius **r**, extending out in the direction of **(ax,ay,az)** for a distance of **d**. If **d** is 0, the cylinder is infinite in both directions.

parts n1 n2 n3 n4....

All parts whose user id matches one of the given values are included in the region. Any number of values may be given.

silist n1 n2 n3 n4....

All elements involved in a contact interface whose user id matches one of the given values are included in the region.

The transformations available are:

sx t

scale the current *x* coordinates by *t*.

sy t

scale the current *y* coordinates by *t*.

sz t

scale the current z coordinates by t .

rx t

rotate around the current x axis by t degrees.

ry t

rotate around the current y axis by t degrees.

rz t

rotate around the current z axis by t degrees.

mat m11 m12 m13 m21 m22 m23 m31 m32 m33

transform the coordinates by matrix multiplication:

transformed		original
x	$=$	$m11\ m12\ m13\ x$
y	$=$	$m21\ m22\ m23\ y$
z	$=$	$m31\ m32\ m33\ z$

3vec v11 v12 v13 v21 v22 v23 v31 v32 v33

Transform the coordinates by the inverse of the transpose matrix:

original		transformed
x	$=$	$v11\ v21\ v31\ x$
y	$=$	$v12\ v22\ v32\ y$
z	$=$	$v13\ v23\ v33\ z$

This appears complicated, but in practice is very intuitive: instead of decomposing into cubes aligned along the coordinate axes, rcb will decompose into parallelipeds whose edges are aligned with the three vectors $(v11, v12, v13)$, $(v21, v22, v23)$, and $(v31, v32, v33)$. Furthermore, the relative lengths of the edges of the decomposition domains will correspond to the relative lengths of these vectors.

C2R x0 y0 z0 vx1 vy1 vz1 vx2 vy2 vz2

The part is converted into a cylindrical coordinate system with origin at $(x0, y0, z0)$, cylinder axis $(vx1, vy1, vz1)$ and $\theta=0$ along the vector $(vx2, vy2, vz2)$. You can think of this as tearing the model along the $(vx2, vy2, vz2)$ vector and unwrapping it around the $(vx1, vy1, vz1)$ axis. The effect is to create decomposition domains that are "cubes" in cylindrical coordinates: they are portions of cylindrical shells. The actual transformation is:

$$\text{new } (x,y,z) = \text{cylindrical coordinates } (r,\theta,z)$$

Knowing the order of the coordinates is important if combining transformations, as in the example below.

S2R x0 y0 z0 vx1 vy1 vz1 vx2 vy2 vz2

Just like the above, but for spherical coordinates. The (vx1,vy1,vz1) vector is the phi=0 axis.

New (x,y,z) = spherical coordinates (rho, theta, phi)

The grouping qualifier is:

lumped

Group all elements in the region on a single processor. If this qualifier is not given, the elements in the region are distributed across all processors.

Examples:

rz 45

will generate domains rotated -45 degrees around the z axis.

C2R 0 0 0 0 0 1 1 0 0

will generate cylindrical shells of domains. They will have their axis along the vector (0,0,1), and will start at the vector (1,0,0) Note that the part will be cut at (1,0,0), so no domains will cross this boundary. If there is a natural boundary or opening in your part, the "theta=0" vector should point through this opening. Note also that if the part is, say, a cylinder 100 units tall and 50 units in radius, after the C2R transformation the part will fit inside the box $x=[0,50]$, $y=[0, 2\text{PI}]$, $z=[0,100]$. In particular, the new y coordinates (theta) will be very small compared to the other coordinate directions. It is therefore likely that every decomposition domain will extend through the complete transformed y direction. This means that each domain will be a shell completely around the original cylinder. If you want to split the domains along radial lines, try this pair of transformations:

C2R 0 0 0 0 0 1 1 0 0

SY 5000

This will do the above C2R, but then scale y by 5000. This will result in the part appearing to be about 30,000 long in the y direction -- long enough that every decomposition domain will divide the part in this (transformed) y direction. The result will be decomposition domains that are radial "wedges" in the original part.

General combinations of transformations can be specified, and they are applied in order:

SX 5 SY .2 RZ 30

will scale x, then y, then rotate.

A more general decomposition might look like:

decomposition { rx 45 sz 10

region { parts 1 2 3 4 5 and sphere 0 0 0 200 lumped }

region { box 0 100 -1.e+8 1.e+8 0 500 or sphere 100 0 200 200 rx 20 }

}

This would take elements that have user ID 1, 2, 3, 4, or 5 for their part, AND that lie in the sphere of radius 200 centered at (0,0,0), and place them all on one processor.

Then, any remaining elements that lie in the given box OR the sphere of radius 200 centered at (100,0,200) would be rotated 20 degrees in x then decomposed across all processors. Finally, anything remaining would be rotated 45 degrees in x, scaled 10 in z, and distributed to all processors. In general, region qualifiers can be combined using the logical operations **and**, **or**, and **not**. Grouping using parentheses is also supported.

Execution of MPP/LS-DYNA

MPP/LS-DYNA runs under a parallel environment which provided by the hardware vendor. The execution of the program therefore varies from machine to machine. On some platforms, command line parameters can be passed directly on the command line. For others, the use of the names file is required. The names file is supported on all systems.

The serial/SMP code supports the use of the SIGINT signal (usually Ctrl-C) to interrupt the execution and prompt for user input, generally referred to as "sense switches." The MPP code also supports this capability. However, on many systems a shell script or front end program (generally "mpirun") is required to start MPI applications. Pressing Ctrl-C on some systems will kill this process, and thus kill the running MPP-DYNA executable. As a workaround, when the MPP code begins execution it creates a file "bg_switch" in the current working directory. This file contains the following single line:

```
rsh <machine name> kill -INT <PID>
```

where <machine name> is the hostname of the machine on which the root MPP-DYNA process is running, and <PID> is its process id. (on HP systems, "rsh" is replaced by "remsh"). Thus, simply executing this file will send the appropriate signal.

Here is a simple table to show how to run the program on various platforms. Of course, scripts are often written to mask these differences.

Platform	Execution Command
DEC Alpha	<code>dmpirun -np <i>n</i> <i>mpp-dyna</i></code>
Fujitsu	<code>jobexec -vp <i>n</i> -mem <i>m</i> <i>mpp-dyna</i></code>
Hitachi	<code>mpirun -np <i>n</i> <i>mpp-dyna</i></code>
HP	<code><i>mpp-dyna</i> -np <i>n</i></code>
IBM	<code>#!/bin./ksh export MP_PROC=<i>n</i> export MP_LABELIO=no export MP_EUILIB=us export MPI_EUIDEVICE=cross0 poe <i>mpp-dyna</i></code>
NEC	<code>mpirun -np <i>n</i> <i>mpp-dyna</i></code>
SGI	<code>mpirun -np <i>n</i> <i>mpp-dyna</i></code>
Sun	<code>tmrn -np <i>n</i> <i>mpp-dyna</i></code>

Where *n* is the number of processors, *mpp-dyna* is the name of the MPP/LS-DYNA executable, and *m* is the MB of real memory.

APPENDIX P: Implicit Solver

INTRODUCTION

The terms implicit and explicit refer to time integration algorithms. In the explicit approach, internal and external forces are summed at each node point, and a nodal acceleration is computed by dividing by nodal mass. The solution is advanced by integrating this acceleration in time. The maximum time step size is limited by the Courant condition, producing an algorithm which typically requires many relatively inexpensive time steps.

While explicit analysis is well suited to dynamic simulations such as impact and crash, it can become prohibitively expensive to conduct long duration or static analyses. Static problems such as sheet metal springback after forming are one application area for implicit methods.

In the implicit method, a global stiffness matrix is computed, inverted, and applied to the nodal out-of-balance force to obtain a displacement increment. The advantage of this approach is that time step size may be selected by the user. The disadvantage is the large numerical effort required to form, store, and factorize the stiffness matrix. Implicit simulations therefore typically involve a relatively small number of expensive time steps.

The implicit analysis capability was first released in Version 950. Initially targeted at metal forming springback simulation, this new capability allowed static stress analysis. Version 970 adds many additional implicit features, including new element formulations for linear and modal analysis.

For best implicit performance, it is important to provide enough memory to allow the stiffness matrix factorization to run in-core. In most cases, the default memory size must be increased. See the Linear Equation Solver section below.

SETTING UP AN IMPLICIT SIMULATION

The keyword `*CONTROL_IMPLICIT_GENERAL` is used to activate the implicit method. LS-DYNA can conduct either a linear or a nonlinear implicit analysis. The keyword `*CONTROL_IMPLICIT_SOLUTION` is used to select between these implicit analysis types. In addition, an implicit eigenvalue analysis can be performed to extract frequencies and mode shapes.

To perform a linear implicit analysis, use the `*CONTROL_IMPLICIT_GENERAL` keyword to activate the implicit method and to specify the time step size. Enter the termination time using the `*CONTROL_TERMINATION` keyword. For a single step analysis, select the step size to be equal to the termination time. Use the `*CONTROL_IMPLICIT_SOLUTION` keyword to request a linear analysis. Select linear element formulations using the `*SECTION_SOLID` and/or `*SECTION_SHELL` keywords. For best accuracy, a double precision version of LS-DYNA should be used for linear analysis.

To perform an eigenvalue analysis, use the `*CONTROL_IMPLICIT_GENERAL` keyword to activate the implicit method and to specify a time step size. Enter the termination time using the `*CONTROL_TERMINATION` keyword (the time step size and termination time must be nonzero, but will otherwise be ignored as LS-DYNA will presently just compute the eigenvalues and stop.) Use the `*CONTROL_IMPLICIT_EIGENVALUE` keyword to indicate the desired number of eigenvalues and frequency ranges of interest. For best accuracy, a double precision version of LS-DYNA should be used for eigenvalue analysis.

A nonlinear implicit simulation is typically divided into several steps. In a dynamic simulation, these are *time steps*. In a static simulation, these are *load steps*. Multiple steps may be used to divide the nonlinear behavior into manageable pieces, to obtain results at intermediate stages during the simulation, or perhaps to resolve a particular frequency of motion in dynamic simulations. In each step, an equilibrium geometry is sought which balances internal and external forces in the model. The *nonlinear equation solver* performs an iterative search using one of several Newton based methods. *Convergence* of this iterative process is obtained when norms of displacement and/or energy fall below user-prescribed tolerances.

Control parameters for the nonlinear equation solver are input using the keyword `*CONTROL_IMPLICIT_SOLUTION`. By default, the progress of the equilibrium search is not shown to the screen. This output can be activated either using the `NLPRINT` input parameter, or interactively toggled on and off by entering “<ctrl-c> nlprint”. The box below shows a typical iteration sequence, where the norms of displacement (du/u) and energy (E_i/E_0) are displayed. When these norms are reduced below user prescribed tolerances (default $1.0e-3$ and $1.0e-2$, respectively), equilibrium is reached within sufficient accuracy, the iteration process is said to have *converged*, and the solution proceeds to the next time step.

```
BEGIN time step      3
=====
                time =  1.50000E-01
    current step size =  5.00000E-02
Iteration:   1      *|du|/|u| =  3.4483847E-01      *Ei/E0 =  1.0000000E+00
Iteration:   2      *|du|/|u| =  1.7706435E-01      *Ei/E0 =  2.9395439E-01
Iteration:   3      *|du|/|u| =  1.6631174E-03      *Ei/E0 =  3.7030904E-02
Iteration:   4      *|du|/|u| =  9.7088516E-05      *Ei/E0 =  9.6749731E-08
```

A typical print-out showing the progress of the nonlinear equation solver. By default, the progress of the equilibrium search is not shown to the screen. This output can be activated either using the `NLPRINT` input parameter, or interactively toggled on and off by entering “<ctrl-c> nlprint”.

LINEAR EQUATION SOLVER

Within each equilibrium iteration, a linear system of equations of the form $\mathbf{K}\Delta\mathbf{u} = \mathbf{R}$ must be solved. To do this, the stiffness matrix \mathbf{K} is inverted and applied to the out-of-balance load or residual \mathbf{R} , yielding a displacement increment $\Delta\mathbf{u}$. Storing and solving this linear system represents a large portion of the memory and CPU costs of an implicit analysis.

Control parameters for solving the linear system $\mathbf{K}\Delta\mathbf{u}=\mathbf{R}$ are input using the keyword *CONTROL_IMPLICIT_SOLVER. Several different linear equation solvers are available, including direct (Gaussian elimination) and iterative (conjugate gradient, Lanczos) methods. A sparse storage scheme is used to minimize memory requirements, which are still often substantial. Two options are available for matrix reordering, allowing nodes and elements to be numbered arbitrarily by the user.

It is very important to allow enough memory for the stiffness matrix factorization to run incore. Although the direct solvers can run out-of-core, using disk files for scratch space, this can slow performance by 100x or more. To view memory requirements for a particular model, select LPRINT=1 on *CONTROL_IMPLICIT_SOLVER, or interactively type “<ctrl-c> lprint”. Summary information will be printed to the screen and message files. Use the command line option “memory=...” to increase memory until this summary reports that the TOTAL AVAILABLE memory is large enough that the solver runs “INCORE”. The memory size may also be specified on *KEYWORD.

NONLINEAR EQUATION SOLVER

Several different nonlinear equation solvers are available for finding equilibrium within each step. All are iterative in nature. In the *full Newton method*, a new stiffness matrix is formed and inverted each equilibrium iteration. This is the most costly method, but can require fewer iterations to reach equilibrium. In the *modified Newton method*, several iterations are performed using the same stiffness matrix. After each iteration, the geometry is updated using $\Delta\mathbf{u}$ and a new \mathbf{R} is computed. This approach reduces cost by avoiding some forming and factoring of the stiffness matrix \mathbf{K} , but usually requires more iterations to reach equilibrium.

The default nonlinear equation solver is the BFGS solver, which uses a *quasi-Newton method*. In this method, the inverted stiffness matrix \mathbf{K} is used for several iterations, but is improved after each iteration using an inexpensive rank two update. If convergence is not reached after 10 iterations, or if *divergence* (increasing \mathbf{R}) is detected, then a new stiffness matrix is automatically formed and inverted. This hybrid method combines the efficiency of the modified Newton method with the reliability of the full Newton method. The number of iterations between stiffness matrix reformations is a user input, defaulting to 10. If a value of one is chosen, then the full Newton method is recovered.

```

BEGIN time step      1
=====
                time = 1.00000E+00
            current step size = 1.00000E+00

Iteration:   1      *|du|/|u| = 2.5517753E+00      *Ei/E0 = 1.0000000E+00

DIVERGENCE (increasing residual norm) detected:
  |{Fe}-{Fi}| ( 7.5426269E+03) exceeds |{Fe}| ( 5.0000000E+00)
automatically REFORMING stiffness matrix...

Iteration:   2      *|du|/|u| = 6.0812935E-01      *Ei/E0 = 4.0526413E-01
Iteration:   4      *|du|/|u| = 1.0974191E-02      *Ei/E0 = 2.3907781E-04
Iteration:   5      *|du|/|u| = 1.0978787E-02      *Ei/E0 = 1.7910795E-04
Iteration:   6      *|du|/|u| = 4.2201181E-03      *Ei/E0 = 4.2557768E-05
Iteration:   7      *|du|/|u| = 4.1142219E-03      *Ei/E0 = 3.0658711E-05
Iteration:   8      *|du|/|u| = 1.9794757E-03      *Ei/E0 = 9.1215551E-06
Iteration:   9      *|du|/|u| = 1.7957653E-03      *Ei/E0 = 6.1669480E-06
Iteration:  10      *|du|/|u| = 1.2022830E-03      *Ei/E0 = 2.9031284E-06

ITERATION LIMIT reached, automatically REFORMING stiffness matrix...

Iteration:  11      *|du|/|u| = 5.4011414E-04      *Ei/E0 = 1.0553019E-06

```

The print-out above shows typical behavior of the default BFGS nonlinear equation solver. Two automatic stiffness reformations are performed, initially due to divergence, and later when the default limit of 10 iterations is exceeded. By default, the progress of the equilibrium search is not shown to the screen. This output can be activated either using the NLPRINT input parameter, or interactively toggled on and off by entering “<ctrl-c> nlprint”.

$$\mathbf{K}_{n+1}^{-1} = (\mathbf{I} + \mathbf{w}\mathbf{v}^T) \mathbf{K}_n^{-1} (\mathbf{I} + \mathbf{v}\mathbf{w}^T)$$

The BFGS update: A new stiffness matrix inverse is approximated by the old stiffness matrix inverse, and the outer product of two carefully chosen vectors.

ELEMENT FORMULATIONS FOR IMPLICIT ANALYSIS

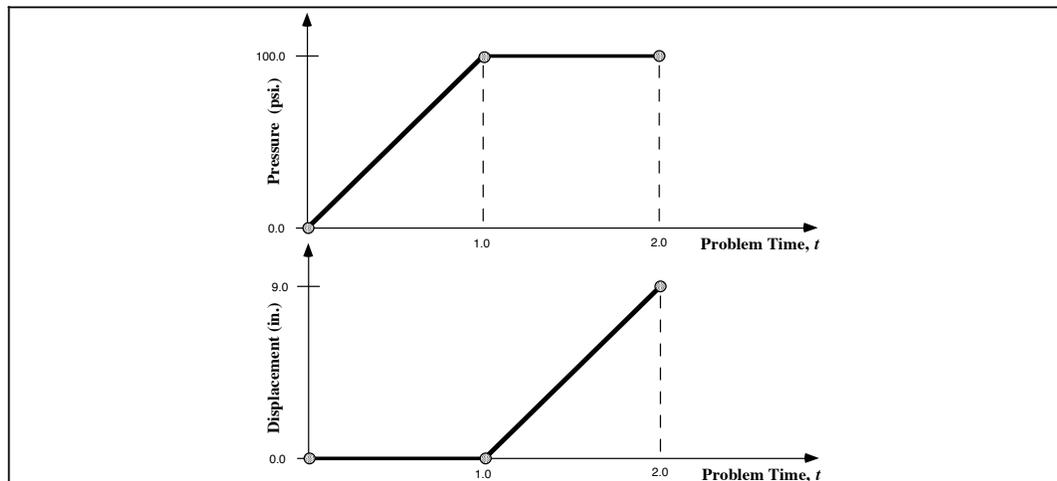
The default element formulations in LS-DYNA are highly efficient, using single point integration. For implicit analysis it is generally more effective to use more expensive element formulations which are less susceptible to hourglass instability. The Hughes-Liu brick element #2 and shell element #6, and the fast shell #16 are good choices for implicit analysis. Stiffness forms of hourglass control are recommended, with hourglass type #6 required for use with implicit solid elements.

APPLYING LOADS DURING IMPLICIT ANALYSIS

Loading is applied using the same keywords as in explicit analysis. Load curves are used to control the magnitude of each load as the simulation proceeds. Typically, the magnitude of each load begins at zero, and is increased to its full value at the end of the last step in the simulation. In this case, the load curve may be defined using only two points.

For example, consider a static analysis where a pressure of 100 psi. is to be applied in 4 steps. Since the analysis is static, the step size can be chosen arbitrarily. For convenience, choose a step size of 0.25, giving a termination time of 1.0. For this problem, the load curve has only two points: (0.0, 0.0) and (1.0, 100.0). LS-DYNA will automatically use linear interpolation to determine the load magnitude at each of the intermediate steps.

In a more complex example, consider a static problem with two types of loading. First, a static pressure of 100 psi. is to be applied, followed by a prescribed displacement of 9 inches. Two load curves are used for this problem, one to control the pressure, and one for the displacement, as shown below. Notice that the displacement is prescribed to be zero while the pressure is applied, then the pressure is held constant while the displacement is applied.



Load curves for a static simulation with two loading events. For convenience in this static simulation, the pressure loading is specified to begin at time 0.0 and end at time 1.0, and the displacement begins at time 1.0 and ends at time 2.0.

AUTOMATIC TIME STEP SIZE CONTROL

In the most simple multi-step nonlinear implicit analysis, the user specifies the *termination time* using the `*CONTROL_TERMINATION` keyword, and the *time step size* using the `*CONTROL_IMPLICIT_GENERAL` keyword, and each step is the same size. But for many simulations, the degree of nonlinearity varies through the course of the analysis. In this case the step size should ideally be varied such that solving for equilibrium in each step is equally difficult. This is accomplished by invoking automatic time step control, using the `*CONTROL_IMPLICIT_AUTO` keyword.

There are two advantages to using automatic time step control. First, the time step size is automatically increased and/or decreased in response to the nonlinearity of the analysis. Nonlinearity is measured simply by the number of iterations required to reach equilibrium. An additional advantage is that if the equilibrium search fails during a time step, LS-DYNA does not terminate. Instead, the step is automatically repeated using a different step size. This process of backing up and retrying difficult steps lends much persistence to the analysis, and is often the only procedure for solving highly nonlinear problems short of adjusting the step size manually.

The input parameters for automatic time step control allow specification of the *optimum number of equilibrium iterations per step*. This indicates how hard LS-DYNA should work in each time step. If equilibrium is reached in fewer than optimum iterations, the size of the next step is increased, and likewise if the equilibrium search requires more than the optimum number of iterations, then the next step size is decreased. Minimum and maximum limits for step size are also input.

IMPLICIT STRESS INITIALIZATION

A common application of the implicit method is to perform static stress initialization for an explicit dynamic calculation. This can be done using two individual calculations, or by switching methods during a calculation. In the first approach, the keyword `*INTERFACE_SPRINGBACK_LSDYNA` is used to generate a "dynain" output file at the end of the simulation. This file is written in keyword format at the end of the simulation, and contains `*NODE`, `*ELEMENT`, and `*INITIAL_STRESS` data. The dynain file can be included into a second input deck to initialize the explicit dynamic analysis.

LS-DYNA can switch "on-the-fly" between the implicit and explicit methods. To use this feature, define a curve which indicates which formulation to use as a function of simulation time. Formulation switching incurs no overhead, and may be performed several times during a simulation. See the `IMFLAG` parameter on the `*CONTROL_IMPLICIT_GENERAL` keyword for more information.

TROUBLESHOOTING CONVERGENCE PROBLEMS

Convergence of the nonlinear equilibrium iteration process presents one of the greatest challenges to using the implicit mode of LS-DYNA. Below are some useful troubleshooting approaches:

Eigenvalue Analysis

Many convergence problems in static implicit analysis are caused by unconstrained rigid body modes. These are created when an insufficient number of constraints are applied to the model, or when individual model parts are left disconnected. Eigenvalue analysis is an excellent diagnostic tool to check for these problems.

To perform an eigenvalue analysis, simply add the `*CONTROL_IMPLICIT_EIGENVALUE` keyword to an implicit input deck. Use the first parameter `NEIGV=20` to compute the lowest 20 modes. Then view the frequencies in the output text file "eigout" and animate the mode shapes

in the binary output file d3eigv using LS-PREPOST. Look for frequencies which are nearly zero. Add constraints as necessary to eliminate unconstrained motion.

D3ITER Plot Database

To diagnose convergence trouble which develops in the middle of a simulation, get a picture of the deformed mesh. Adjust the d3plot output interval to produce an output state after every step leading up to the problematic time.

An additional binary plot database named “d3iter” is available which shows the deformed mesh during each equilibrium iteration. This output is activated and de-activated interactively by entering "<ctrl-c> iteration". View this database using LS-PREPOST. Note that stress data is not included. Frequently the problem will become obvious, especially as deformation is magnified.

Prescribed Motion with Death Time

A common static analysis problem occurs when small contact gaps exist between parts at time=0. An example is a load-driven punch which deforms a panel, with a small initial contact gap. This creates instantaneous unconstrained rigid body modes until contact between parts is established. (These modes will be obvious in an eigenvalue analysis, as described above.) To overcome this problem, apply a prescribed motion boundary condition to move the parts into contact. Once contact is established, use the optional death time to “kill” the prescribed motion, and allow the applied force or pressure to provide further loading. Monitor reaction forces from the prescribed motion, and adjust the applied loads to match reasonably well at the death time.

APPENDIX Q: User Defined Weld Failure

The addition of a user weld failure subroutine into LS-DYNA is relatively simple. The UWELDFAIL subroutine is called every time step when OPT=2 is specified in MAT_SPOTWELD. As data, the identification number for the spotweld material, six constants specified in the input by the locations NRR through MTT, the radius of the cross section of the spotwelds, the current time, and the current values of the resultants for the spotwelds, which are stored in array STRR, are passed to the subroutine. The subroutine loops over the welds from LFT through LLT, and sets the values of the failure flag array FLAG.

```

      SUBROUTINE UWELDFAIL (IDWELD, STRR, FAIL, FIBL, CM, TT, LFT, LLT)
C*****
C | LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC) |
C | ----- |
C | COPYRIGHT 2002 JOHN O. HALLQUIST, LSTC |
C | ALL RIGHTS RESERVED |
C*****
C
C*** SPOTWELD FAILURE ROUTINE
C
C*** LOCAL COORDINATES: X IS TANGENT TO BEAM, Y & Z ARE NORMAL
C
C*** VARIABLES
C      IDWELD ---- WELD ID NUMBER
C      STRR ----- STRESS RESULTANTS
C                   (1) AXIAL (X DIRECTION) FORCE
C                   (2) Y SHEAR FORCE
C                   (3) Z SHEAR FORCE
C                   (4) MOMENT ABOUT Z
C                   (5) MOMENT ABOUT Y
C                   (6) TORSIONAL RESULTANT
C      FAIL ----- FAILURE FLAG
C                   =0 NOT FAILED
C                   =1 FAIL ELEMENT
C      FIBL ----- LOCATION (1,*) GIVES THE SPOTWELD DIAMETER
C      CM ----- 6 CONSTANTS SUPPLIED BY USER
C      TT ----- CURRENT SIMULATION TIME
C      LFT,LLT --- DO-LOOP RANGE FOR STRR
C
C      DIMENSION IDWELD(*), STRR(6,*), FAIL(*), CM(*), FIBL(5,*)
C
C
C      RETURN
C      END

```


APPENDIX R: User Defined Cohesive Model

The addition of a user cohesive material subroutine into LS-DYNA is relatively simple. The UMATiC subroutine is called every time step where *i* ranges from 41 to 50. Input for the material model follows the *MAT_USER_DEFINED_MATERIAL definition. The user has the option of providing either a scalar or vectorized subroutine. As discussed in the Remarks for the user-defined material, the first two material parameters are reserved to specify how the density is treated and the number of integration points required for the failure of the element.

The cohesive model calculates the tractions on the mid-surface of the element as a function of the differences of the displacements and velocities of the upper (defined by nodes 5-6-7-8) and lower surfaces (defined by nodes 1-2-3-4). The displacements, velocities, and the calculated tractions are in the local coordinate system of the element, where the first two components of the vectors are in the plane of the mid-surface and the third component is normal to the mid-surface.

A stiffness must also be calculated by the user for the explicit time step calculation in LS-DYNA. This stiffness must provide an upper bound on the stiffness in all three directions.

The material fails at an integration point when *ifail*=.true. For an element to be deleted from the calculation, the number of integration points specified by the second material parameter must fail. If the second parameter is zero, elements cannot fail regardless of the specification of IFAIL in the user-defined material input.

The following example is a vectorized model with two elastic constants and failure:

```

subroutine umat41c(idpart,cm,lft,llt,fc,dx,dxdt,aux,ek,
&                ifail,dtlsiz,crv)
include 'nlqparm'
c
c*** vector cohesive material user model example
c
c*** variables
c      idpart ---- Part ID
c      cm ----- material constants
c      lft,llt --- start and end of block
c      fc ----- components of the cohesive force
c      dx ----- components of the displacement
c      dxdt ----- components of the velocity
c      aux ----- history storage
c      ek ----- max. stiffness/area for time step calculation
c      ifail ---- =.false. not failed
c                =.true. failed
c      dtlsiz ---- time step size
c      crv ----- curve array
c
c*** dx, dxdt, and fc are in the local coordinate system:
c      components 1 and 2 are in the plane of the cohesive surface
c      component 3 is normal to the plane
c
c*** cm storage convention
c      (1) =0 density is per area
c          =1 density is per volume
c      (2) number of integration points for element deletion
c          =0 no deletion
c      (3:48) material model constants
c
      logical ifail
      dimension cm(*),fc(nlq,*),dx(nlq,*),dxdt(nlq,*),
&              aux(nlq,*),ek(*),ifail(*),dtlsiz(*),crv(101,2,*)

```

```

c
  et=cm(3)
  en=cm(4)
  eki=max(et,en)
  fcfail=cm(5)
c
  do i=lft,llt
    fc(i,1)=et*dx(i,1)
    fc(i,2)=et*dx(i,2)
    fc(i,3)=en*dx(i,3)
    ek(i)=eki
    ifail(i)=fc(i,3).gt.fcfail
  enddo
c
  return
end

```

The second example implements the Tveergard-Hutchinson cohesive model with failure in both the vectorized (UMAT42C) and scalar (UMAT43C) forms. Note the LFT and LLT are passed to the scalar version, however their value is zero.

```

      subroutine umat42c(idpart,params,lft,llt,fTraction,jump_u,dxdt,
&      aux,ek,ifail,dt1siz,crv)
      include 'nlqparm'
c
c***  vector cohesive material user model example
c
c***  variables
c      idpart ---- part ID
c      params ---- material constants
c      lft,llt --- start and end of block
c      fTraction - components of the cohesive force
c      jump_u ---- components of the displacement
c      dxdt ----- components of the velocity
c      aux ----- history storage
c      ek ----- max. stiffness/area for time step calculation
c      ifail ----- =.false. not failed
c                  =.true. failed
c      dt1siz ---- time step size
c      crv ----- curve array
c
c***  jump_u, dxdt, and fTraction are in the local coordinate system:
c      components 1 and 2 are in the plane of the cohesive surface
c      component 3 is normal to the plane
c
c***  cm storage convention
c      (1) =0 density is per area
c          =1 density is per volume
c      (2) number of integration points for element deletion
c          =0 no deletion
c      (3:48) material model constants
c
c      Tveergard-Hutchinson model based on:
c      tahoe/src/elements/cohesive_surface/cohesive_models/TvergHutch3DT.cpp
c
c      the declaration below is processed by the C preprocessor and
c      is real*4 or real*8 depending on whether LS-DYNA is single or double
c      precision
c      REAL L,jump_u
c
c      logical ifail

```

```

dimension params(*), fTraction(nlq,*), jump_u(nlq,*),
&          dxdt(nlq,*), aux(nlq,*), ek(*), ifail(*), dtlsiz(*),
&          crv(101,2,*)

fsigma_max=params(3)
fd_c_n=params(4)
fd_c_t=params(5)
fL_1=params(6)
fL_2=params(7)
fpenalty=params(8)

fK=fpenalty*fsigma_max/(fL_1*fd_c_n)

fac=min(fd_c_n/fd_c_t**2,1./fd_c_n)

do i=lft,llt
  u_t1 = jump_u(i,1)
  u_t2 = jump_u(i,2)
  u_n = jump_u(i,3)

  r_t1 = u_t1/fd_c_t
  r_t2 = u_t2/fd_c_t
  r_n = u_n/fd_c_n
  L = sqrt(r_t1*r_t1 + r_t2*r_t2 + r_n*r_n)

  if (L .lt. fL_1) then
    sigbyL=fsigma_max/fL_1
  else if (L .lt. fL_2) then
    sigbyL = fsigma_max/L
  else if (L .lt. 1.) then
    sigbyL = fsigma_max*(1. - L)/(1. - fL_2)/L
  else
    sigbyL = 0.0
    ifail(i)=.true.
  endif

  fTraction(i,1) = sigbyL*r_t1*(fd_c_n/fd_c_t)
  fTraction(i,2) = sigbyL*r_t2*(fd_c_n/fd_c_t)
  fTraction(i,3) = sigbyL*r_n

c   penetration
  if (u_n .lt. 0) fTraction(i,3)=fTraction(i,3)+fK*u_n

c   approximate stiffness for time step
  if (u_n .lt. 0) then
    ek(i)=fac*sigbyL+fK
  else
    ek(i)=fac*sigbyL
  endif

enddo

return
end
subroutine umat43c(idpart,params,lft,llt,fTraction,jump_u,dxdt,
&          aux,ek,ifail,dtlsiz,crv)
c
c*** scalar cohesive material user model example
c
c*** variables
c          idpart ---- part ID
c          params ---- material constants

```

```

c      lft,llt --- start and end of block
c      fTraction - components of the cohesive force
c      jump_u ---- components of the displacement
c      dxdt ----- components of the velocity
c      aux ----- history storage
c      ek ----- max. stiffness/area for time step calculation
c      ifail ----- =.false. not failed
c                =.true. failed
c      dt1siz ---- time step size
c      crv ----- curve array
c
c***  jump_u, dxdt, and fTraction are in the local coordinate system:
c      components 1 and 2 are in the plane of the cohesive surface
c      component 3 is normal to the plane
c
c***  cm storage convention
c      (1) =0 density is per area
c          =1 density is per volume
c      (2) number of integration points for element deletion
c          =0 no deletion
c      (3:48) material model constants
c
c      Tveergard-Hutchinson model based on:
c      tahoe/src/elements/cohesive_surface/cohesive_models/TvergHutch3DT.cpp
c
c      the declaration below is processed by the C preprocessor and
c      is real*4 or real*8 depending on whether LS-DYNA is single or double
c      precision
c      REAL L,jump_u

      logical ifail
      dimension params(*),fTraction(*),jump_u(*),
&          dxdt(*),aux(*),crv(101,2,*)

      fsigma_max=params(3)
      fd_c_n=params(4)
      fd_c_t=params(5)
      fL_1=params(6)
      fL_2=params(7)
      fpenalty=params(8)

      fK=fpenalty*fsigma_max/(fL_1*fd_c_n)

      fac=min(fd_c_n/fd_c_t**2,1./fd_c_n)

      u_t1 = jump_u(1)
      u_t2 = jump_u(2)
      u_n = jump_u(3)

      r_t1 = u_t1/fd_c_t
      r_t2 = u_t2/fd_c_t
      r_n = u_n/fd_c_n
      L = sqrt(r_t1*r_t1 + r_t2*r_t2 + r_n*r_n)

      if (L .lt. fL_1) then
          sigbyL=fsigma_max/fL_1
      else if (L .lt. fL_2) then
          sigbyL = fsigma_max/L
      else if (L .lt. 1.) then
          sigbyL = fsigma_max*(1. - L)/(1. - fL_2)/L
      else
          sigbyL = 0.0

```

```
        ifail=.true.
        endif

        fTraction(1) = sigbyL*r_t1*(fd_c_n/fd_c_t)
        fTraction(2) = sigbyL*r_t2*(fd_c_n/fd_c_t)
        fTraction(3) = sigbyL*r_n

c      penetration
        if (u_n .lt. 0) fTraction(3)=fTraction(3)+fK*u_n

c      approximate stiffness for time step
        if (u_n .lt. 0) then
            ek=fac*sigbyL+fK
        else
            ek=fac*sigbyL
        endif

        return
    end
```


APPENDIX S: User Defined Boundary Flux

A user defined boundary flux interface is provided in LS-DYNA where it is possible to define the thermal heat flux (power per surface area) in or out of a surface segment as an arbitrary function of temperature and history. The user may associate history variables with each individual flux interface and also use load curves.

The user flux interface is invoked using the keyword `*BOUNDARY_FLUX_OPTION`. This is accomplished with the parameter `NHISV`. When it is defined with a value greater than 0, the user subroutine

```
subroutine usrflux(fl,flp,...)
```

is called to compute the flux (`fl`) defined as heat (energy) per time and per surface area.

Other parameters that are passed to the user flux subroutine include the segment nodal temperatures at the previous (T_0) and current time (T_1), the segment nodal coordinates and the time integration parameter α . Also, the current thermal simulation time t , the time step Δt and average segment temperature (T_α) at time $t+\alpha\Delta t$ is provided together with the curve array for accessing defined load curves in the keyword input file. For computing load curve values, note that load curve IDs need to be transformed to internal numbers or the subroutine `crvval` should be used, see the appendix on user defined materials for details.

The segment coordinates available in the subroutine are such that the outward normal vector follows the well-known right-hand rule, thus segments corresponding to the lower surface of thick thermal shells are reversed before passed to the subroutine. For shells in general, the segment connectivity should follow the connectivity of the actual shell element to avoid problems.

Optionally, the user may define the derivative of the flux `fl` with respect to the average segment temperature (T_α) at time $t+\alpha\Delta t$, `flp`. This value is used in the nonlinear thermal solver for assembling the correct stiffness matrix and must be set by the user. If possible, it is recommended to use a value that reflects the nonlinearity of the flux model, otherwise the value 0 should be used.

An array of history variables, identical with the input parameters defined in the keyword input file, are passed to the subroutine that can be updated with time or kept constant throughout the simulation. An example of usage would be to integrate the flux with time to keep track of the dissipated energy per surface area in order to simulate the effects of spray cooling in hot-stamping.

```

subroutine usrflux(fl,flp,x,tnpl,tnl,nodes,
.      alpha,atime,atemp,dt,time,fhsv,nfhsv,crv)
C*****
C | LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC) |
C | ----- |
C | COPYRIGHT © 2007 JOHN O. HALLQUIST, LSTC |
C | ALL RIGHTS RESERVED |
C*****
c
c      User subroutine for boundary thermal flux

```

```

c
c Purpose: To define thermal flux parameter (heat per surface area and
c time)
c
c Variables:
c
c fl           = flux intensity (output)
c flp          = flux intensity derivative wrt atemp (output)
c x(3,nodes)   = global segment coordinates (input)
c tnpl(nodes)  = temperatures at time time (input)
c tnl(nodes)   = temperatures at time time-dt (input)
c nodes        = number of nodes in segment (3,4 or 6) (input)
c alpha        = time integration parameter (input)
c atime        = time+(alpha-1)*dt
c atemp        = average segment temperature at time atime
c dt           = time step size (input)
c time         = time at which the new temperature is sought (input)
c fhsv(nfhsv)  = flux history variables (input/output)
c nfhsv        = number of flux history variables for this segment
c              (input)
c crv          = curve array (input)
c
c include 'nlqparm'
c dimension x(3,*),tnpl(*),tnl(*)
c dimension fhsv(*),crv(lq1,2,*)
c
c Define flux by linear convection
c that optionally decays (in an ad-hoc way) as power
c dissipates from surface
c
c fhsv(1) = convection coefficient
c fhsv(2) = ambient temperature
c fhsv(3) = total amount of energy per surface area available
c fhsv(4) = dissipated energy per surface area at current
c
c hcon=fhsv(1)
c tinf=fhsv(2)
c flin=hcon*(tinf-atemp)
c if (nfhsv.gt.2) then
c   q=(1.-fhsv(4)/fhsv(3))/
c     (1.+0.5*dt*flin/fhsv(3))
c   flp=-q*hcon
c   if (q.gt.1.) then
c     q=1.
c     flp=-hcon
c   elseif (q.lt.0.) then
c     q=0.
c     flp=0.
c   endif
c   fl=q*flin
c   fhsv(4)=fhsv(4)+dt*.5*fl
c   fhsv(4)=min(fhsv(3),fhsv(4))
c else
c   fl=flin
c   flp=-hcon
c endif
c
c return
c end

```