*DATABASE

The database definitions are optional, but they are necessary to obtain output files containing results information. The ordering of the database definition cards in the input file is completely arbitrary. In this section the database keywords are defined in alphabetical order:

DATABASE_*OPTION1_{OPTION2}

*DATABASE_ACEOUT

*DATABASE_ADAMS

*DATABASE_ALE

*DATABASE_ALE_MAT

*DATABASE_ALE_OPERATION

DATABASE_BINARY_*OPTION1_{OPTION2}

*DATABASE_BINARY_D3PROP

*DATABASE_CPM_SENSOR

DATABASE_CROSS_SECTION_*OPTION1_{OPTION2}

*DATABASE_DEFRAGMENT

*DATABASE_D3FTG

*DATABASE_D3MAX

*DATABASE_EXTENT_AVS

*DATABASE_EXTENT_BINARY

*DATABASE_EXTENT_D3PART

*DATABASE_EXTENT_INTFOR

*DATABASE_EXTENT_MOVIE

*DATABASE_EXTENT_MPGS

*DATABASE_EXTENT_SSSTAT

*DATABASE_FATIGUE_STRESS_CYCLE

*DATABASE_FATXML

*DATABASE_FORMAT

DATABASE_FREQUENCY_ASCII_*OPTION

DATABASE_FREQUENCY_BINARY_*OPTION

*DATABASE_FSI

*DATABASE_FSI_SENSOR

DATABASE_HISTORY_*OPTION

*DATABASE_HISTORY_ACOUSTIC

*DATABASE_ISPHHTC

*DATABASE_MASSOUT

DATABASE_MAX_*OPTION

*DATABASE_NODAL_FORCE_GROUP

*DATABASE_PAP_OUTPUT

*DATABASE_PBLAST_SENSOR

*DATABASE_PROFILE

*DATABASE_PWP_FLOW

*DATABASE_PWP_OUTPUT

*DATABASE_RCFORC_MOMENT

*DATABASE_RECOVER_NODE

*DATABASE_RVE

*DATABASE_SALE

*DATABASE_SPRING_FORWARD

*DATABASE_SUPERPLASTIC_FORMING

*DATABASE_TRACER

*DATABASE_TRACER_ALE

*DATABASE_TRACER_GENERAL

*DATABASE_TRACER_GENERATE

***DATABASE**_OPTION1_{OPTION2}

OPTION1 specifies the type of database. LS-DYNA will *not* create an ASCII database unless the corresponding *DATABASE_*OPTION1* card is included in the input deck. *OPTION1* may be any of the items in the following list:

ABSTAT Airbag statistics. See *AIRBAG_*OPTION*.

ATDOUT Automatic tiebreak damage statistics for *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK with OPTION set to 7, 9, 10, or 11 (only SMP at the moment).

AVSFLT AVS database. See *DATABASE_EXTENT_AVS.

BEARING *ELEMENT_BEARING force output.

BNDOUT Boundary condition forces and energy.

CABLE Cross-sectional output from *DEFINE_CABLE

CURVOUT Output from *DEFINE_CURVE_FUNCTION.

DEFGEO Deformed geometry file (Note that to output this file in Chrysler format insert the following line in your .cshrc file: "setenv LSTC_DEFGEO chrysler"). The nasbdf file (NASTRAN Bulk Data) is created whenever the DEFGEO file is requested.

DCFAIL Failure function data for *MAT_SPOTWELD_DAIMLERCHRYSLER

DEBOND Mode of bond breakage between discrete element spheres. See *DATABASE_HISTORY_DES_*OPTION*.

DEFORC Discrete spring and discrete damper (*ELEMENT_DISCRETE) data. If the user wishes to be selective about which discrete elements are output in deforc, use *DATABASE_HISTORY_DISCRETE_*OPTION* to select elements for output (but only if BEAM = 0 in *DATABASE_BINARY_D3PLOT) or set PF = 1 in *ELEMENT_DISCRETE to turn off output for particular elements; otherwise all discrete elements are output.

DEMASSFLOW Mass flow rate across a plane as specified by *DEFINE_DE_MASSFLOW_PLANE.

DESTAT Frequency distribution and mean value of stress energies ratio for DES, including translational normal, translational shear, translational rolling, rotational normal, rotational shear, and rotational rolling energy. The stress energies of both DES-DES contact and DES-Structure contact (if *DEFINE_DE_TO_SURFACE_COUPLING is used) are included. The distribution is based on translational normal stress energy.

DISBOUT Discrete beam element, type 6, relative displacements, rotations, and force resultants, all in the local coordinate system, which is also output. Use with *DATABASE_HISTORY_BEAM.

ELOUT Element data. See *DATABASE_HISTORY_OPTION. Also, see the fields INTOUT and NODOUT on Card 3 of the *DATABASE_EXTENT_BINARY keyword. This latter option will output all integration point data or extrapolated data to the connectivity nodes in a file call eloutdet.

GCEOUT Geometric contact entity forces. See *CONTACT_ENTITY.

GLSTAT Global statistics and energies (recommended). See *CONTROL_ENERGY.

H3OUT Joint forces and moments for Hybrid III rigid body dummies. See *COMPONENT_HYBRIDIII.

ICVOUT Incompressible control volume output file. See *DEFINE_CONTROL_VOLUME.

JNTFORC Joint forces. See *CONSTRAINED_JOINT_*OPTION*.

MATSUM Part energies. See Remarks 1 and 2 below.

MOVIE See *DATABASE_EXTENT_MOVIE.

MPGS See *DATABASE_EXTENT_MPGS.

NCFORC Nodal contact forces. See *CONTACT, Card 1 (SAPR and SBPR).

NODFOR Nodal force groups. See *DATABASE_NODAL_FORCE_GROUP.

NODOUT Nodal translational motion and, where applicable, nodal rotational motion. See *DATABASE_HISTORY_NODE_*OPTION*.

PBSTAT Particle blast data. This data is written only to the binout binary database, under the branch name abstat_pbm. See *DEFINE_PARTICLE_BLAST.

PLLYOUT Pulley element data for *ELEMENT_BEAM_PULLEY.

PRTUBE Pressure tube data for *DEFINE_PRESSURE_TUBE.

PYRO Pyro actuator data for *LOAD_PYRO_ACTUATOR.

RBDOUT Motion of rigid bodies in global and local coordinate systems. See Remark 2.

RCFORC Resultant contact interface forces. To output in a local coordinate system, see *CONTACT, Optional Card C.

RCFORC_DR Resultant contact interface forces output during the dynamic relaxation phase. To output in a local coordinate system, see *CONTACT, Optional Card C.

RWFORC Rigidwall forces. See *RIGIDWALL_*OPTION*.

SBTOUT Seatbelt output file. See *ELEMENT_SEATBELT_*OPTION*. See Remark 1 of *SENSOR_DEFINE_MISC.

SECFORC Cross-sectional forces. See *DATABASE_CROSS_SECTION_*OPTION*.

SLEOUT Contact interface energies. See *CONTACT_*OPTION*.

SNSROUT Sensed value or status of general sensors. See *SENSOR_*OPTION*.

SPCFORC SPC reaction forces. See *BOUNDARY_SPC_*OPTION*.

SPGCPL SPG to surface coupling forces. The ASCII data file is named spgcplforc. See *DEFINE_SPG_TO_SURFACE_COUPLING.

SPHOUT SPH data. See *DATABASE_HISTORY_*OPTION*.

SPHMASSFLOW Mass flow rate across a plane as specified by *DEFINE_SPH_MASSFLOW_PLANEdefine_sph_massflow_plane_kword.

SPHVICINITY SPH vicinity sensor. See *DEFINE_SPH_VICINITY_SENSOR.

SSSTAT Subsystem data. See *DATABASE_EXTENT_SSSTAT.

SWFORC Spot weld forces (spot welds and rivets).

TPRINT Thermal output from a coupled structural/thermal or thermal-only analysis. Includes all nodes unless *DATABASE_HISTORY_NODE_*OPTION* is also provided in the keyword input.

TRHIST Tracer particle history information. See *DATABASE_TRACER.

Available options for *OPTION2* are:

<BLANK>

FILTER

MASS_PROPERTIES

FILTER can only be used when *OPTION1* is NCFORC or SECFORC. With the FILTER option, this keyword requires an additional data card; see Card 2 below. See Remark 12.

To include global and subsystem mass and inertial properties in the glstat and ssstat files, add the keyword option MASS_PROPERTIES, as shown below. Activating this option leads to the output of the current mass and inertia properties, including the principle inertia and their axes. The calculated properties do not include the mass of deleted nodes and rigid bodies.

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	1	2	3	4	5	6	7	8
Variable	DT	BINARY	LCUR	IOOPT	OPTION1	OPTION2	OPTION3	OPTION4
Type	F	I	I	I	F/I	I	I	I
Default	0.	1 or 2	none	0.	0	0	0	0

VARIABLE	DESCRIPTION
DT	Time interval between outputs. If DT is zero, no output is printed. If DT < 0.0, the result will be output every -DT time steps.
BINARY	Flag for binary output. See remarks under "Output Files and Post-Processing" in Appendix O, "LS-DYNA MPP User Guide." EQ.1: ASCII file is written. This is the default for shared memory parallel (SMP) LS-DYNA executables. EQ.2: Data written to a binary database binout, 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 for MPP LS-DYNA executables. EQ.3: ASCII file is written, and the data is also written to the binary database (NOTE: MPP LS-DYNA executables will only produce the binary database).
LCUR	Optional curve ID specifying time interval between outputs. Use *DEFINE_CURVE to define the curve. The abscissa is time, and the ordinate is the time interval between dumps.
IOOPT	Flag to govern behavior of the output frequency load curve defined by LCUR: EQ.1: When output is generated at time t_n, the next output time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCUR}(t_n)$. This is the default behavior. EQ.2: When output is generated at time t_n, the next output time t_{n+1} is computed as

VARIABLE	DESCRIPTION

$$t_{n+1} = t_n + \mathrm{LCUR}(t_{n+1}) \ .$$

EQ.3: Output is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

OPTION1

OPTION1 applies to the bndout, nodout, and elout files. For the nodout file OPTION1 is a *real* variable that defines the time interval between outputs for the high frequency file, nodouthf. If OPTION1 is zero, no output is printed. Nodal points that are to be output at a higher frequency are flagged using HFO in the input for *DATABASE_HISTORY_NODE_LOCAL.

For the elout file OPTION1 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the solid elements. See Remark 7 below for the elout file.

See Remark 9 for the bndout file.

OPTION2

OPTION2 applies to the bndout, nodouthf and elout files. For the nodouthf OPTION2 defines the binary file flag for the high frequency nodouthf file. See the field BINARY above.

For the elout file OPTION2 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the shell elements. See Remark 7 below for the elout file.

See Remark 9 for the bndout file.

OPTION3

OPTION3 applies to the bndout and elout files only. For the elout file OPTION3 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the thick shell elements. See Remark 7 below for the elout file and Remark 9 for the bndout file.

OPTION4

OPTION4 applies to the bndout and elout files only. For the elout file OPTION4 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the beam elements. See Remark 7 below for the elout file and Remark 9 for the bndout file.

The following Card 2 applies only to *DATABASE_NCFORC_FILTER and *DATABASE_SECFORC_FILTER.

Card 2	1	2	3	4	5	6	7	8
Variable	RATE	CUTOFF	WINDOW	TYPE				
Type	F	F	F	I				
Default	none	none	none	0	0	0	0	0

VARIABLE	DESCRIPTION
RATE	Time interval *T* between filter sampling.
CUTOFF	Frequency cut-off *C* in Hz.
WINDOW	The width of the window *W* in units of time for storing the single, forward filtering required for the TYPE = 2 filter option. Increasing the width of the window will increase the memory required for the analysis. A window that is too narrow will reduce the amplitude of the filtered result significantly, and values below 15 are not recommended for that reason. In general, the results for the TYPE = 2 option are sensitive to the width of the window and experimentation is required.
TYPE	Flag for filtering options: EQ.0: No filtering (default) EQ.1: Single pass, forward Butterworth filtering EQ.2: Two-pass filtering over the specified time window. Backward Butterworth filtering is applied to the forward Butterworth results that have been stored. This option improves the phase accuracy significantly at the expense of memory.

The file names and corresponding unit numbers are:

Description	I/O Unit #	File Name
Airbag statistics	43	abstat
Automatic tiebreak damage	92	atdout
ASCII database	44	avsflt

Description	I/O Unit #	File Name
Boundary conditions	46	bndout (nodal forces and energies)
Cable output	422	cableout
Discrete element sphere bonds	498	debond
Smug animator database	40	defgeo
Discrete elements	36	deforc
Discrete elements mass flow	219	demflow
Discrete beam elements	215	disbout
Element data	34	elout
Contact entities	48	gceout
Global data	35	glstat
Joint forces	53	jntforc
Material energies	37	matsum
MOVIE file family	50	movie*nnn*.xxx where.*nnn*=001-999
MPGS file family	50	mpgs*nnn*.xxx where *nnn* = 001-999
Nastran/BDF file	49	nasbdf (see comment below)
Nodal interface forces	38	ncforc
Nodal force group	45	nodfor
Nodal point data	33	nodout
Pulley element data	216	pllyout
Pressure tube data	421	prtube
Pyro actuator data	442	pyro
Rigid body data	47	rbdout
Resultant interface forces	39	rcforc
Resultant interface forces	39	rcforc_dr (for dynamic relaxation)
Rigidwall forces	32	rwforc
Seat belts	52	sbtout
Cross-section forces	31	secforc
Interface energies	51	sleout
SPC reaction forces	41	spcforc
SPG surface coupling forces	124	spgcplforc

Description	I/O Unit #	File Name
SPH element data	68	sphout
Subsystems statistics	58	ssstat
Nodal constraint resultants	42	swforc (spot welds/rivets)
Thermal output	73	tprint
Tracer particles	70	trhist

Output Components for ASCII Files.

ABSTAT
volume
pressure
internal energy
input mass flow rate
output mass flow rate (total)
mass
temperature
density
surface area of airbag
reaction
output mass flow rate (poros-ity)
output mass flow rate (venting)

ABSTAT_PBM
internal energy
translational energy
part pressure
part area
part x, y, z force

BNDOUT
x, y, z force
x, y, z, moment
energies

CABLEOUT
cross-sectional area
contact force
stress

DCFAIL
failure function
normal term
bending term
shear term
weld area
effective strain rate
axial force
shear force
torsional moment
bending moment

DEBOND
normal break
shear break
total break

DEFORC
x, y, z force / moment
relative rotation / displacement

ELOUT		
Beams	**(t)Shells**	**Solids**
axial force resultant	xx, yy, zz stress	xx, yy, zz stress
s shear resultant	xy, yz, zx stress	xy, yz, zx stress
t shear resultant	plastic strain	effective stress
s moment resultant	xx, yy, zz strain†	yield function††
t moment resultant	xy, yz, zx strain†	xx, yy, zz strain†
torsional resultant		xy, yz, zx strain†
for ELFORM = 1,7,8,9,11,14		
sig11, sig12, sig31		
plastic strain		

† Strains written for solids and for lower and upper integration points of shells and tshells if STRFLG = 1 in *DATABASE_EXTENT_BINARY.

†† "yield function" is whatever is written as the 7th history variable. For example, this would be effective plastic strain for *MAT_024.

GCEOUT	
x, y, z force	x, y, z moment

GLSTAT	
time step	total energy
kinetic energy	external work
internal energy	total and initial energy
sprint and damper energy	energy ratio without eroded energy
hourglass energy	element & part 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
added mass	percentage [mass] increase
drilling energy	dissipation energy†

† For implicit, integration errors due to large time steps will result in inaccurate estimates of kinetic and internal energies, regardless of how tight the convergence tolerances are. The "lost" energy arising from this discretization error is accumulated in the dissipated kinetic and internal energies, respectively, to render energy balance in the sense described in the introduction. Energy balance in implicit is an implication of having solved the implicit problem to sufficient degree of accuracy, and even though the opposite may not be true it can still be used as an indicator; energy balance *likely* implies a good solution while poor energy balance *definitely* implies a less accurate one.

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
x force
y force
z force

NODOUT
x, y, z displacement
x, y, z velocity
x, y, z acceleration
x, y, z rotation
x, y, z rotational velocity
x, y, z rotation acceleration

NODFOR
x, y. z force

PRTUBE
cross section area
pressure
velocity
density

PYRO
pressure
volume
massflow
mass
temperature
energy
density

PLLYOUT
adjacent beam IDs
slip
slip rate
resultant force
wrap angle

RBDOUT
x, y, z displacement
x, y, z velocity
x, y, z acceleration

RCFORC/RCFORC_DR
x, y, z force
mass of nodes in contact

RWFORC
normal
x, y, z force

SECFORC
x, y, z force
x, y, z moment
x, y, z center
area
resultant force

SLEOUT
surfa energy
surfb energy
frictional energy

SPCFORC	SWFORC	SPHOUT
x, y, z force	axial force	xx, yy, zz stress
x, y, z moment	shear force	xy, yz, zx stress
	failure function	density
	weld length	number of neighbors
	resultant moment	xx, yy, zz strain
	torsion	xy, yz, zx strain
		half of smoothing length
		plastic strain
		particle active state
		effective stress
		temperature
		xx, yy, zz strain rate
		xy, yz, zx strain rate
		SPH to SPH coupling forces

Remarks:

1. **Discrepancies between "matsum" and "glstat" Output.** 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. **PRINT Keyword Option on *PART.** 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. **The Restart Feature.** This keyword is also used in the restart phase; see Restart Input Data. Thus, the output interval can be changed when restarting.

4. **LS-PrePost.** All information in the files except in AVSFLT, MOVIE, and MPGS can also be plotted using LS-PrePost. Arbitrary cross-plotting of results between ASCII files is easily handled.

5. **The "rcforc" File.** Resultant contact forces reported in rcforc are averaged over the preceding output interval.

6. **Spring and damper energy.** "Spring and damper energy" reported in glstat is a subset of "internal energy." The "spring and damper energy" includes the internal energy of discrete elements, seatbelt elements, and that associated with joint stiffness (see *CONSTRAINED_JOINT_STIFFNESS_…).

7. **OPTION*n* field for "elout."** OPTION1, OPTION2, OPTION3, and OPTION4 give the number of additional history variables output for the integrated solids, shells, thick shells, and beams, respectively. Within this special option, each integration point is printed with its corresponding history data. No integration points are averaged. This is different than the default output where the stress data within a shell ply of a fully integrated shell, for example, are averaged and then written as output. The primary purpose of this database extension is to allow the actual integration point stress data and history variable data to be checked. There are no transformations applied to either the output stresses or history data; for example, EOCS in *CONTROL_OUTPUT and CMPFLG in *DATABASE_EXTENT_BINARY do not apply to elout when OPTION2 is nonzero.

8. **The failure function.** The failure function reported to the DCFAIL database is set to zero when the weld fails. If damage is active, then it is set to the negative of the damage scale factor which goes from 1 to 0 as damage grows.

9. **OPTION*n* field for "bndout."** For the bndout file, OPTION1 controls the nodal force group output, OPTION2 controls the concentrated force output, OPTION3 controls the pressure boundary condition output, and OPTION4 controls the velocity/displacement/acceleration nodal boundary conditions. If the value is 0 or left blank, the category is included (the default), and if it is 1, the category is not included in the bndout file.

10. **Contents of "glstat."** The glstat table above includes all items that *may* appear in the glstat data. The items that are actually written depend on the contents of the input deck. For example, hourglass energy appears only if HGEN = 2 in *CONTROL_ENERGY and added mass only appears if DT2MS < 0 in *CONTROL_TIMESTEP.

11. **Element ID Controlling the Time Step.** The element ID controlling the time step is included in the glstat data but is not read by LS-PrePost. If the element

ID is of interest to the user, the ASCII version of the glstat file can be opened with a text editor.

12. **The FILTER Option.** The FILTER option uses a single-pass or double-pass Butterworth filter. For the single-pass filter, the data is forward filtered, while for the double-pass filter, forward filtered results undergo an additional backward filter. The forward filtered output $Y(n)$ at sampling interval n is obtained from the solution value $X(n)$ using the formula

$$Y(n) = a_0 X(n) + a_1 X(n-1) + a_2 X(n-2) + b_1 Y(n-1) + b_2 Y(n-2) \ ,$$

where the coefficients are

$$\begin{aligned}
\omega_d &= 2\pi \left(\frac{C}{0.6}\right) 1.25 \\
\omega_a &= \tan(\omega_a T/2) \\
a_0 &= \omega_a^2/(1+\sqrt{2}\omega_a+\omega_a^2) \\
a_1 &= 2a_0 \\
a_2 &= a_0 \\
b_1 &= 2\left(1-\omega_a^2\right)/\left(1+\sqrt{2}\omega_a+\omega_a^2\right) \\
b_2 &= \left(-1+\sqrt{2}\omega_a-\omega_a^2\right)/\left(1+\sqrt{2}\omega_a+\omega_a^2\right)
\end{aligned}$$

The two previous solution values and filtered values at $n-1$ and $n-2$ are stored.

Backward filtering improves the phase response of the filtered output. It is performed according to the formula

$$Z(n) = a_0 Y(n) + a_1 Y(n+1) + a_2 Y(n+2) + b_1 Z(n+1) + b_2 Z(n+2) \ ,$$

where $Z(n)$ is the backward filtered value at sample time n. This implies that all the forward filtered values $Y(n)$ are stored during the analysis which requires a prohibitive amount of memory. To limit the amount of memory required, the forward filtered values are stored for the time interval W, where the number of stored states is W/T, and the backward filtering is applied starting at the last saved value of the forward filtered values. As the window width increases, the filtered values approach the values that would be obtained from storing all of the forward filtered values.

The results of the backward filtering are sensitive to the window width, and experimentation with the width is necessary to obtain good results with the minimum window width. A window width of at least 10 to 15 times the sample rate T should be used as a starting point. Some applications may require a window width that is much larger. The required window width decreases as the cut-off frequency increases. Or, to put it another way, the window width must be increased to make the filtered output smoother.

As an example, a random series of numbers between 0 and 1 was generated and filtered at intervals of 0.1 milliseconds with cut-off frequencies from 60 Hz to 420 Hz. The reverse filtering was applied with various window widths to determine how many forward-filtered states must be saved to achieve fixed levels of accuracy compared to complete reverse filtering from the last state to the first state. The results are shown in the table below. Note that the error is calculated *only* for the first state and the numbers being filtered are random. *This example should only be used as a very rough guide that indicates the overall trends and not as a recommendation for specific problems*.

Cut-off Frequency	No. of States 50% Error	No. of States 25% Error	No. of States 10% Error	No. of States 5% Error	No. of States 1% Error
60 Hz	26	33	55	68	87
120 Hz	13	16	30	37	44
180 Hz	8	10	22	26	30
240 Hz	6	8	17	19	23
300 Hz	5	6	12	15	18
360 Hz	5	6	10	12	16
420 Hz	4	5	9	10	15

*DATABASE_ACEOUT

Purpose: Time interval for nodal output of acoustic solution results of SSD and spectral analyses invoked with *CONTROL_IMPLICIT_SSD_DIRECT and *CONTROL_ACOUSTIC_SPECTRAL.

Card 1	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
DTOUT	Time interval between the outputs

*DATABASE_ADAMS

Purpose: Request output of an MDI Modal Neutral File, d3mnf, for later use in Hexagon's Adams software. It is available for MPP only.

Card 1	1	2	3	4	5	6	7	8
Variable	IFLAG	M_UNITS	L_UNITS	T_UNITS				
Type	I	I	I	I				
Default	0	none	none	none				

VARIABLE	DESCRIPTION
IFLAG	Flag controlling write of d3mnf after eigenvalue analysis: EQ.0: Do not write (default). EQ.1: Write to d3mnf.
M_UNITS	Mass units of measure used in the model: EQ.-1: Kilogram EQ.-2: Gram EQ.-3: Megagram (metric ton) EQ.-4: lbf × s2/in (psi-compatible) EQ.-5: Slug EQ.-6: Pound-mass
L_UNITS	Length units of measure used in the model: EQ.-1: Meter EQ.-2: Centimeter EQ.-3: Millimeter EQ.-4: Inch EQ.-5: Foot
T_UNITS	Time units of measure used in the model: EQ.-1: Second

VARIABLE	DESCRIPTION
	EQ.-2: Millisecond
	EQ.-3: Minute
	EQ.-4: Hour

Remarks:

1. **Units.** Models must be created using a combination of the above units.

2. **Restrict to node set.** To restrict d3mnf to a node set, set IPARM3 on *CONTROL_IMPLICIT_EIGENVALUE. This ability is available for EIGMTH = 2, 3, 5, 6, 102, and 103.

*DATABASE_ALE

Purpose: For each ALE group (or material), this card controls the output for element time-history variables (in a tabular format that can be plotted in LS-PrePost by using the XYPlot button). It can also control d3plot output for ALE elements (see Remark 3).

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT	SETID						
Type	F	I						
Default	none	none						

Variable Cards. Optional cards that can be used to add more variables with the volume fractions in the database (the volume fractions are always output). Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7	VAR8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
DTOUT	Time interval between the outputs (see Remark 3)
SETID	ALE element set ID (see Remark 3). If the model is 1D (*SECTION_ALE1D), the set should be *SET_-BEAM. If the model is 2D (*SECTION_ALE2D), the set should be *SET_-SHELL. If the model is 3D (*SECTION_SOLID), the set should be *SET_-SOLID.
VAR*i*	Variable rank in the following list: LT.0: |VAR*i*| is the rank of the auxiliary variable to be replaced in d3plot (see Remark 3)

VARIABLE	DESCRIPTION
	EQ.1: *xx*-stress
	EQ.2: *yy*-stress
	EQ.3: *zz*-stress
	EQ.4: *xy*-stress
	EQ.5: *yz*-stress
	EQ.6: *zx*-stress
	EQ.7: plastic strain
	EQ.8: internal energy
	EQ.9: bulk viscosity
	EQ.10: previous volume
	EQ.11: pressure
	EQ.12: mass
	EQ.13: volume
	EQ.14: density
	EQ.15: kinetic energy
	EQ.16: internal energy density
	EQ.17: impulse (pressure integrated over time)
	If there is a blank column between 2 VAR*i* values, the list between these 2 values is selected. For example, if the card is input as follows:
	`1, ,6`
	then the 6 stresses are added to the database.

Remarks:

1. **Creating .xy Files.** The .xy files are created when the termination time is reached or if one of the following switches (after pressing the keys Ctrl - C) stops the job: sw1, stop, quit. During the run, they can be created with the switch sw2.

2. **Curve Output.** The .xy files are created by element. There is a curve for each ALE group (or material). A curve can be added for volume averaged variables.

3. **Modification of Auxiliary Variables in d3plot.** If VAR1 and VAR2 are input with VAR1 > 0 and VAR2 < 0, |VAR2| is a location in the auxiliary array (array listing for each element and ALE group: 6 stresses, plastic strain, internal energy,

bulk viscosity, previous volume, history variable #1, history variable #2,…). Before outputting data to d3plot, the variable at location |VAR2| is replaced by the variable at location VAR1 in the variable list. For instance, if VAR1 = 10 and VAR2 = -1, then the previous volume will be output in the *xx*-stress spot in d3plot. More than one variable can be output using this method by setting additional VAR*i*. This is a way to control output to d3plot. If this method is used, the .xy file is not output. Therefore, DTOUT is not used (see *DATABASE_BINARY_D3PLOT). If SETID is not provided, all the ALE elements are considered.

*DATABASE_ALE_MAT

Purpose: For each ALE group (or material), this card activates extra output for:

1. material volume: alematvol.xy,
2. material mass: alematmas.xy,
3. internal energy: alematEint.xy,
4. kinetic energy: alematEkin.xy,
5. and kinetic energy loss during the advection: alematEkinlos.xy.

These files are written in the ".xy" format, which LS-PrePost can plot with its "XYPlot" button.

Card	1	2	3	4	5	6	7	8
Variable	DTOUT	BOXLOW	BOXUP	DTXY				
Type	F	I	I	F				
Default	none	0	0	0				

VARIABLE	DESCRIPTION
DTOUT	Time interval between the outputs
BOXLOW, BOXUP	Range of *DEFINE_BOX ids. BOXLOW is the lower bound for the range while BOXUP is the upper bound. The series of volumes covered by the specified range of *DEFINE_BOX determines the mesh regions for which ALE material data are to be output.
DTXY	Time interval between the extractions of ".xy" files from datalemat.tmp.

Remarks:

The ".xy" files are created at termination or if one of the following switches (Ctrl-C) is encountered: sw2, sw1, stop, quit.

*DATABASE_ALE_OPERATION

Purpose: Output values from a function defined by *DEFINE_FUNCTION to certain databases, namely d3plot and .xy history files. The function arguments are defined by the variable cards in this keyword as well as predefined variables (see Remark 1). The value computed by the function for each ALE element can replace a history variable in d3plot at a frequency defined in *DATABASE_BINARY_D3PLOT. Also, the function can output values for a set of elements to .xy history files (in a tabular format that can be plotted in LS-PrePost by using the XYPlot button) at a frequency defined by a time interval set in the input for this keyword. This keyword is only supported for ALE and ALE 2D.

Card Summary:

Card 1. This card is required.

FCT	HISVN	WRT					

Card 2. This card is included if WRT ≥ 10.

DT	SETID						

Card 3. This card is optional. Include as many of this card as needed. The next keyword ("*") card terminates this input.

VAR	VAR	VAR	VAR	VAR	VAR	VAR	VAR

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	FCT	HISVN	WRT					
Type	I	I	I					
Default	none	none	1					

VARIABLE	DESCRIPTION
FCT	*DEFINE_FUNCTION ID; see Remark 1.
HISVN	Number of the history variable replaced in d3plot (see Remark 2.)

VARIABLE	DESCRIPTION
WRT	File output flag. WRT must be a two digit number: $$WRT = L + M \times 10$$ The 1's digit controls the replacement of the history variable number HISVN in d3plot: L.EQ.1: For each ALE element in the mesh, replace the values of the history variable with values computed by the function FCT. (See Remark 2.) L.EQ.0: Do not modify d3plot. The 10's digit controls the history output of values computed by the function FCT: M.EQ.1: For each ALE element in the set SETID, write .xy file that stores values computed by FCT at a frequency DT. (See Remarks 3 and 4.) M.EQ.0: Do not output this history file.

History Card. This card is included if WRT $\geq$ 10.

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

VARIABLE	DESCRIPTION
DT	Time interval between computed function values included in the .xy file. (See Remarks 3 and 4.)
SETID	ALE element set ID. See Remarks 3 and 4. If the model is 2D (*SECTION_ALE2D), the set should be a shell set (see *SET_SHELL). If the model is 3D (*SECTION_SOLID), the set should be a solid set (see *SET_SOLID).

Function Argument Cards. Optional cards used to add more arguments to the user-defined function (time, time step, cycle number and volume fractions are always included as the first arguments to the function). Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	VAR	VAR	VAR	VAR	VAR	VAR	VAR	VAR
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
VAR	Arguments that can be included in function FCT (see Remark 1): EQ.1: *xx*-stress EQ.2: *yy*-stress EQ.3: *zz*-stress EQ.4: *xy*-stress EQ.5: *yz*-stress EQ.6: *zx*-stress EQ.7: Plastic strain EQ.8: Internal energy EQ.9: Bulk viscosity EQ.10: Previous volume EQ.11: Mass EQ.12: Volume EQ.13: Nodal *x*-positions EQ.14: Nodal *y*-positions EQ.15: Nodal *z*-positions EQ.16: Nodal *x*-velocities EQ.17: Nodal *y*-velocities EQ.18: Nodal *z*-velocities EQ.19: Nodal *x*-accelerations

VARIABLE	DESCRIPTION
	EQ.20: Nodal y-accelerations
	EQ.21: Nodal z-accelerations

Remarks:

1. **Function Argument Specification.** The argument values are presented to the function FCT as floating point data (see the example below). The time, time step, cycle number and volume fractions are the first arguments of the function by default. Then the order of the following arguments appearing in *DEFINE_-FUNCTION should match the order of the input in Card 3. For example, if the model has 2 ALE groups (so 2 volume fractions) and Card 3 has input "7, 8, 11", the arguments should be as follows:

   - Time,
   - Time step,
   - Cycle number,
   - Volume fraction of the 1st group,
   - Volume fraction of the 2nd group,
   - Plastic strain of the 1st group,
   - Plastic strain of the 2nd group,
   - Internal energy of the 1st group,
   - Internal energy of the 2nd group,
   - Mass of the 1st group,
   - Mass of the 2nd group.

   If there is a blank column between 2 inputs, all the variables between the two values are also included. For example, if the card contains "1, ,6", then the 6 stresses (1 through 6) are selected as arguments.

   For the nodal variables (VAR $\geq$ 13), the number of arguments in the function depends on the element type and its number of nodes. For a solid, 8 nodal variables are expected as arguments each time VAR is greater than or equal to 13., one for each node. For a shell, 4 nodal variables are expected as arguments. For example, VAR = 17 in a 3D ALE model would mean 8 y-velocities are arguments of the function FCT.

2. **Modification of History Variables in d3plot.** The history variables appear in d3plot after the first 7 volume averaged auxiliary variables: 6 stresses and plastic strain. The number of additional history variables output to d3plot is controlled by NEIPH (for solids) or NEIPS (for shells) in *DATABASE_EXTENT_BINARY. The history variables are listed in the MISC menu of LSPREPOST as history variable #1, history variable #2, … (the exact number in the code depends on *MAT

and *EOS). Before outputting data in d3plot, the history variable at history variable number HISVN is replaced by the value computed using FCT.

3. **Creating .xy Files.** The .xy files are created when the termination time is reached or if one of the following switches (after pressing the keys Ctrl - C) stops the job: sw1, stop, quit. During the run, they can be created with switch sw2. The file names start with "dataleop" followed by the Element ID and end with the extension .xy

4. **Curve Output.** Values computed by the function FCT are included in the .xy file for each element listed in the shell (2D) or solid (3D) set SETID at a frequency defined by the time interval DT.

Example:

In the example below, LS-DYNA computes the volume averaged pressure:

- for each ALE element so that the pressure can be fringed in d3plot in the place of history variable #2
- for each ALE element included in SETID so that the pressure history for these elements can be output to a .xy file

```
*DATABASE_ALE_OPERATION
$#     fct     hisvn       wrt
         1         2        11
$#      dt     setid
      1e-9         1
$#     var       var       var
         1         2         3
*DEFINE_FUNCTION
$#     fct
         1
float pressure(float     time, float timestep, float cycle
              ,float volfrac1, float volfrac2
              ,float xstress1, float xstress2
              ,float ystress1, float ystress2
              ,float zstress1, float zstress2)
{
  float pres,pres1,pres2;
  pres1 = -(xstress1+ystress1+zstress1)/3.0;
  pres2 = -(xstress2+ystress2+zstress2)/3.0;
  pres = volfrac1*pres1+volfrac2*pres2;
  return pres;
}
```

DATABASE_BINARY_*OPTION1_{OPTION2}

Purpose: Request binary output. See also *DATABASE_EXTENT_BINARY.

Choices for *OPTION1* are:

BLSTFOR Blast pressure database. See also *LOAD_BLAST_ENHANCED and Remark 3.

CPMFOR Corpuscular Particle Method interface force database. See Remark 2.

D3CRACK Option to control output interval for ASCII aea_crack file for the Winfrith concrete model (*MAT_084/085). Oddly, this command does not control the output of the binary crack database for the Winfrith concrete model. The binary crack database is written when "q = " appears on the execution line, and its output interval is taken from *DATABASE_BINARY_D3PLOT. LS-PrePost displays cracks in deformed Winfrith concrete materials with the combination of this file and d3plot.

D3DAT Database for CPG model data.

D3DRLF Dynamic relaxation database.

D3DUMP Database for restarts. Define output frequency in cycles.

D3PART Database for subset of parts. See also *DATABASE_EXTENT_BINARY and *DATABASE_EXTENT_D3PART.

D3PLOT Database for entire model. See also *DATABASE_EXTENT_BINARY.

D3PROP Database containing property data. See *DATABASE_BINARY_D3PROP.

D3THDT Database containing time histories for subsets of elements and nodes. See *DATABASE_HISTORY. This database does not include geometry.

DEMFOR DEM interface force database. See Remark 5.

FSIFOR ALE interface force database. The fsifor database does *not* have a default filename, so it *must* be given a filename using "h=" on the execution line. See Remark 1.

FSILNK ALE interface linking database. See Remark 4.

INTFOR Contact interface database. The intfor database does *not* have a default filename, so it *must* either be given a filename using the FILE option or using "S=" on the execution line. Also see *CONTACT fields SAPR and SBPR.

ISPHFOR Incompressible SPH interface force database. See Remark 8.

PBMFOR	Particle Blast Method interface force database. See Remark 6.
RUNRSF	Database for restarts. Define output frequency in cycles.

The only choice for *OPTION2* is FILE. When *OPTION2* is set to FILE, the LS-DYNA keyword reader requires one extra data card indicating the file name for the database selected in *OPTION1*. *Presently, this is only implemented for *DATABASE_BINARY_INTFOR and *DATABASE_BINARY_ISPHFOR.*

The D3DUMP and the RUNRSF options create complete databases that are necessary for restarts. See Restart Input Data. When RUNRSF is specified, the same file is overwritten after each interval, unless the field NR is set, which causes a series of restart files to be overwritten in a cyclic order. When D3DUMP is specified, a new d3dump file is created after each interval. After the first interval, LS-DYNA writes d3dump01, after the second d3dump02, and so on. The default file names are runrsf and d3dump unless other names are specified on the execution line, see EXECUTION SYNTAX in the GETTING STARTED section. Since all data held in memory is written to the restart files, these files can be quite large. 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 the output interval for d3plot is automatically set to one-twentieth of the termination time.

The d3plot, d3part, d3drlf, d3dat, and intfor databases contain histories of geometry and of state variables. With the LS-PrePost program, these databases can be used to, for example, animate deformed geometry and plot time histories of element stresses and nodal displacements. The d3thdt database contains time history data for element subsets and global information, but it does not include information about the geometry. See *DATABASE_HISTORY. This data can be plotted with the LS-PrePost program. For the contents of the d3plot, d3part, and d3thdt databases, see the *DATABASE_EXTENT_BINARY keyword. The size of the databases can be reduced by restricting the information that is dumped. The contents of the d3thdt database are also specified with the *DATABASE_HISTORY definition. Note that, in particular, the databases can be considerably reduced for models with rigid bodies containing many elements.

The fsifor database, like the intfor database, does not have a default filename. For this database a unique filename must be specified on the execution line with h=filename; see EXECUTION SYNTAX in the GETTING STARTED section. 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.

Card Summary:

Card 0. Additional card for the FILE keyword option (*OPTION2*). Presently, this is only implemented for *DATABASE_BINARY_INTFOR and *DATABASE_BINARY_ISPHFOR.

FNAME

Card 1. This card is required.

DT/CYCL	LCDT/NR	BEAM	NPLTC	PSETID	CID		

Card 2a. This card is read when *OPTION1* is D3PLOT. It is optional.

IOOPT	RATE	CUTOFF	WINDOW	TYPE	PSET		

Card 2b. This card is read when *OPTION1* is D3PART. It is optional.

HSETID	BSETID	SSETID	TSETID				

Card 2c. This card is read when *OPTION1* is INTFOR, D3DUMP, or D3DAT. It is optional.

IOOPT							

Data Card Definitions:

FILE Card. When *OPTION2* is set to FILE, include this card. Presently, this is only implemented for *DATABASE_BINARY_INTFOR and *DATABASE_BINARY_ISPHFOR.

Card 0	1	2	3	4	5	6	7	8
Variable	FNAME							
Type	A80							

VARIABLE	DESCRIPTION
FNAME	Name of the database. S=filename on the execution line overrides FNAME for *DATABASE_BINARY_INTFOR, while isph=filename on the execution line overrides FNAME for *DATABASE_BINARY_ISPHFOR

Card 1	1	2	3	4	5	6	7	8
Variable	DT/CYCL	LCDT/NR	BEAM	NPLTC	PSETID	CID	NSKIP	
Type	F/I	I	I	I	I	I	I	
Default	none	0/1	0	optional	optional	0	1	

VARIABLE	DESCRIPTION
DT	This field defines the time interval between output states, DT, for all options except D3DUMP, RUNRSF, and D3DRLF.
CYCL	For D3DUMP and RUNRSF options, this field is the number of time steps between output states. For the D3DLF option, the value, n, inputted in this field causes an output state to be written every n^{th} convergence check during the explicit dynamic relaxation phase.
LCDT	Optional load curve ID specifying the output time interval as a function of time. This variable is only available for options D3DAT, D3DUMP, D3PART, D3PLOT, D3THDT, INTFOR and BLSTFOR.
NR	Number of RUNning ReStart Files, runrsf, written in a cyclical fashion. The default is 1, that is, only one runrsf file is created, and the data therein is overwritten each time data is output.
BEAM	Discrete element option flag (*DATABASE_BINARY_D3PLOT only): EQ.0: Discrete spring and damper elements are added to the d3plot database where they are displayed as beam elements. The discrete elements' global x, global y, global z and resultant forces (moments) and change in length (rotation) are written to the database where LS-PrePost (incorrectly) labels them as though they were beam quantities, such as axial force, S-shear resultant, T-shear resultant, etc. EQ.1: No discrete spring, damper. and seatbelt elements are added to the d3plot 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 IDs, and beam elements may be created for the spring and dampers with identical IDs to existing beam elements, causing a fatal error. However, this option comes with some limitations and, therefore, should be used

VARIABLE	DESCRIPTION

with caution.

1. Contact interfaces which are based on part IDs of seatbelt elements will not be properly generated if this option is used.

2. DEFORMABLE_TO_RIGID will not work if PID refers to discrete, damper, or seatbelt elements.

EQ.2: Discrete spring and damper elements are added to the d3plot 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.

This flag, set in *DATABASE_BINARY_D3PLOT, also affects the display of discrete elements in several other databases, such as d3drlf and d3part.

NPLTC: DT = ENDTIM/NPLTC. Applies to D3PLOT, D3PART, D3DUMP, D3DAT, DEMFOR, and INTFOR options only. This overrides the DT specified in the first field. ENDTIM is specified in *CONTROL_TERMINATION.

PSETID: Part set ID for D3PART and D3PLOT options only. See *SET_PART. Parts in PSETID are excluded from the d3plot database. The d3dat database also honors excluding the parts excluded from d3plot, but this option may not be provided to d3dat directly. Only parts in PSETID are included in the d3part database.

CID: Coordinate system ID for FSIFOR and FSILNK. See *DEFINE_COORDINATE_SYSTEM.

NSKIP: For options D3DUMP and RUNRSF, MPP/HYBRID executables output both local and global restart files. Each processor creates its own local file for a small or simple restart. Then, processor 0 collects data from all processors and outputs the full deck restart file, but this output requires lots of data communication and disk space. This option provides a scale factor for CYCL (number of time steps between output states) to reduce the number of full deck restart output states to save CPU time. Thus, a full deck restart file is output every NSKIP × NCYCL time steps. The last full deck restart is always output with the regular restart file at the end of the job. The default value of

VARIABLE	DESCRIPTION
	NSKIP is 1.

D3PLOT Card. Additional card for the D3PLOT option. It is optional.

Card 2a	1	2	3	4	5	6	7	8
Variable	IOOPT	RATE	CUTOFF	WINDOW	TYPE	PSET		
Type	I	F	F	F	I	I		
Default	0	none	none	none	0	0		

VARIABLE	DESCRIPTION
IOOPT	This input field governs how the plot state frequency is determined from curve LCDT: EQ.1: When a plot is generated at time t_n, the next plot time t_{n+1} is computed as $$t_{n+1} = t_n + \mathrm{LCDT}(t_n)\ .$$ This is the default behavior. EQ.2: When a plot is generated at time t_n, the next plot time t_{n+1} is computed as $$t_{n+1} = t_n + \mathrm{LCDT}(t_{n+1})\ .$$ EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.
RATE	Time interval T between filter sampling. See Remark 7.
CUTOFF	Frequency cut-off C in Hz. See Remark 7.
WINDOW	The width of the window W in units of time for storing the single, forward filtering required for the TYPE = 2 filter option. Increasing the width of the window will increase the memory required for the analysis. A window that is too narrow will reduce the amplitude of the filtered result significantly, and values below 15 are not recommended for that reason. In general, the results for the TYPE = 2 option are sensitive to the width of the window and experimentation is required. See Remark 7.

VARIABLE	DESCRIPTION
TYPE	Flag for filtering options. See Remark 7. EQ.0: No filtering (default). EQ.1: Single pass, forward Butterworth filtering. EQ.2: Two pass filtering over the specified time window. Backward Butterworth filtering is applied to the forward Butterworth results that have been stored. This option improves the phase accuracy significantly at the expense of memory.
PSET	Part set ID for filtering. If no set is specified, all parts are included. For each element integration point in the d3plot file, 24 words of memory are required in LS-DYNA for the single-pass filtering and more for the two-pass filtering. Specifying PSET is recommended to minimize the memory requirements. See Remark 7.

Optional D3PART Card. This card is read for the D3PART option. It is meant to specify element sets for which data is to be output and works in addition to PSETID. It is optional.

Card 2b	1	2	3	4	5	6	7	8
Variable	HSETID	BSETID	SSETID	TSETID				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
HSETID	Set ID of a *SET_SOLID for which data will be output to d3part
BSETID	Set ID of a *SET_BEAM for which data will be output to d3part
SSETID	Set ID of *SET_SHELL for which data will be output to d3part
TSETID	Set ID of *SET_TSHELL for which data will be output to d3part

Optional D3DAT, D3DUMP, and INTFOR Card. Additional card for the D3DAT, D3-DUMP, and INTFOR options. This card is optional.

Card 2c	1	2	3	4	5	6	7	8
Variable	IOOPT							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
IOOPT	This input field governs how the plot state frequency is determined from curve LCDT:

EQ.1: When a plot is generated at time t_n, then next plot time t_{n+1} is computed as

$$t_{n+1} = t_n + \text{LCDT}(t_n) \ .$$

This is the default behavior.

EQ.2: When a plot is generated at time t_n, then next plot time t_{n+1} is computed as

$$t_{n+1} = t_n + \text{LCDT}(t_{n+1}) \ .$$

EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

Remarks:

1. **FSIFOR.** *DATABASE_BINARY_FSIFOR only applies to models having penalty-based coupling between Lagrangian and ALE materials CTYPE = 4 or 5 in the coupling card, *CONSTRAINED_LAGRANGE_IN_SOLID, or in case of S-ALE, *ALE_STRUCTURED_FSI. 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. However, this coupling pressure is averaged over the whole surface entity being monitored. To obtain the coupling pressure contour plot as a function of time over the coupled surface, define *DATABASE_BINARY_FSIFOR. To use it, three things must be done:

a) The INTFORC parameter (*CONSTRAINED_LAGRANGE_IN_SOLID, 4th row, 3rd column) must be turned ON (INTFORC = 1). This is not required for *ALE_STRUCTURED_FSI. In that case, it is always ON.

b) A *DATABASE_BINARY_FSIFOR card is defined, controlling the output interval. The parameter DT in this card defines the time interval between outputs.

c) Executing LS-DYNA as follows activates writing this interface force file:

lsdyna i=**inputfilename.k** ... h=**interfaceforcefilename**

This binary interface force file contains the segment coupling pressure and forces over the whole simulation interval.

d) Forces from a 2D axisymmetric simulation are reported per radian, regardless of the element formulation used. Multiply the 2D force by 2π to get the 3D force.

2. **CPMFOR.** *DATABASE_BINARY_CPMFOR applies to models using *AIRBAG_PARTICLE feature. It controls the output interval of the CPM interface force file. This file does not have a default name. Thus, activating the writing of this file requires the cpm command line option (cpm=):

lsdyna i=**inputfilename.k** … cpm=**interfaceforce_filename**

The CPM interface force file stores each segment's coupling pressure and forces. The coupling pressure is averaged over each segment without considering the effect of ambient pressure, P_{atm}.

3. **BLSTFOR.** The BLSTFOR database is available for two-dimensional axisymmetric analysis. While line plotting is possible, fringing is not, at this time.

4. **FSILNK.** The *DATABASE_BINARY_FSILNK variant writes the selected *CONSTRAINED_LAGRANGE_IN_SOLID interface's segment pressure to the fsilink file for the next analysis without ALE meshes. To select the interface, set INTFORC = 1 on *CONSTRAINED_LAGRANGE_IN_SOLID. This file does not have a default name. Thus, activating the writing of this file requires the fsilnk command line option (fsilnk=):

lsdyna i=**inputfilename.k** … fsilnk=**filename**

The output binary file can be converted into a keyword file in which the data is converted into *LOAD_SEGMENTs and *DEFINE_CURVEs. To create this keyword file, run LS-DYNA with only the fsilnk command option:

lsdyna fsilnk=**filename**

5. **DEMFOR.** *DATABASE_BINARY_DEMFOR applies to models using DEM coupling option *DEFINE_DE_TO_SURFACE_COUPLING. This keyword controls the output interval of DEM interface force file. This file does not have a default name. Thus, activating the writing of this file requires the dem command line option (dem=):

lsdyna i=**inputfilename.k** … dem=**interfaceforce_filename**

The DEM interface force file stores the coupling pressure and forces at each segment.

6. **PBMFOR.** *DATABASE_BINARY_PBMFOR applies to models using *DEFINE_PARTICLE_BLAST. It controls the output interval of PBM interface force file. This file does not have a default name. Thus, activating the writing of this file requires the pbm command line option (pbm=):

lsdyna i=**inputfilename.k** … pbm=**interfaceforce_filename**

The PBM interface force file stores the pressure and forces applied to the segments of the structure.

7. **D3PLOT filtering.** Filtered stress data replaces the stress data in the d3plot file when filtering is enabled.

8. **ISPHFOR.** *DATABASE_BINARY_ISPHFOR applies to models using incompressible SPH (FORM = 13 in *CONROL_SPH) with *DEFINE_SPH_MESH_SURFACE. This keyword controls the output interval of the ISPH interface force file. This file does not have a default name. Thus, activating the writing of this file requires either giving the file name with the FILE keyword option or using the isph command line option (isph=):

lsdyna i=**inputfilename.k** … isph=**interfaceforce_filename**

If both are provided, the file name given on the command line is used. The incompressible SPH interface force file stores the pressures and forces applied to the segments of the mesh surface by the SPH fluid. It also reports whether a given segment is wet or not (instantaneous wetness flag) at each output time step as well as how long a segment has been wet (time-accumulated wetness).

*DATABASE_BINARY_D3PROP

Purpose: This card causes LS-DYNA to add the part, material, equation of state, section, and hourglass data to the first d3plot file or else write the data to a separate database d3prop. Rigidwall data can also be included.

Card 1	1	2	3	4	5	6	7	8
Variable	IFILE	IMATL	IWALL					
Type	I	I	I					
Default	1	0	0					

VARIABLE	DESCRIPTION
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

*DATABASE_CPM_SENSOR

Purpose: This card activates an ASCII file cpm_sensor. Its input defines the sensors' locations based on the positions of some Lagrangian segments. The output gives the history of the velocity, temperature, density and pressure averaged on the number of particles contained in the sensors. This card is activated only when the *AIRBAG_PARTICLE card is used.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY						
Type	F	I						

Sensor Definition Cards. Each card defines one sensor. This card may be repeated to define multiple sensors. Input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SEGID	OFFSET	LX	LY	LZ			
Type	I	F	F	F	F			

VARIABLE	DESCRIPTION
DT	Output interval
BINARY	Flag for the binary file EQ.1: ASCII file is written, EQ.2: Data is written to the binary file binout, EQ.3: ASCII file is written and the data is written to the binary file binout.
SEGID	Segment set ID. See *SET_SEGMENT.
OFFSET	Offset distance, d, between sensor and the segment center. See Remark 2.
LX	Radius(sphere and cylinder)/length in local x-direction(rectangular) of the sensor. See Remarks 1 and 2.

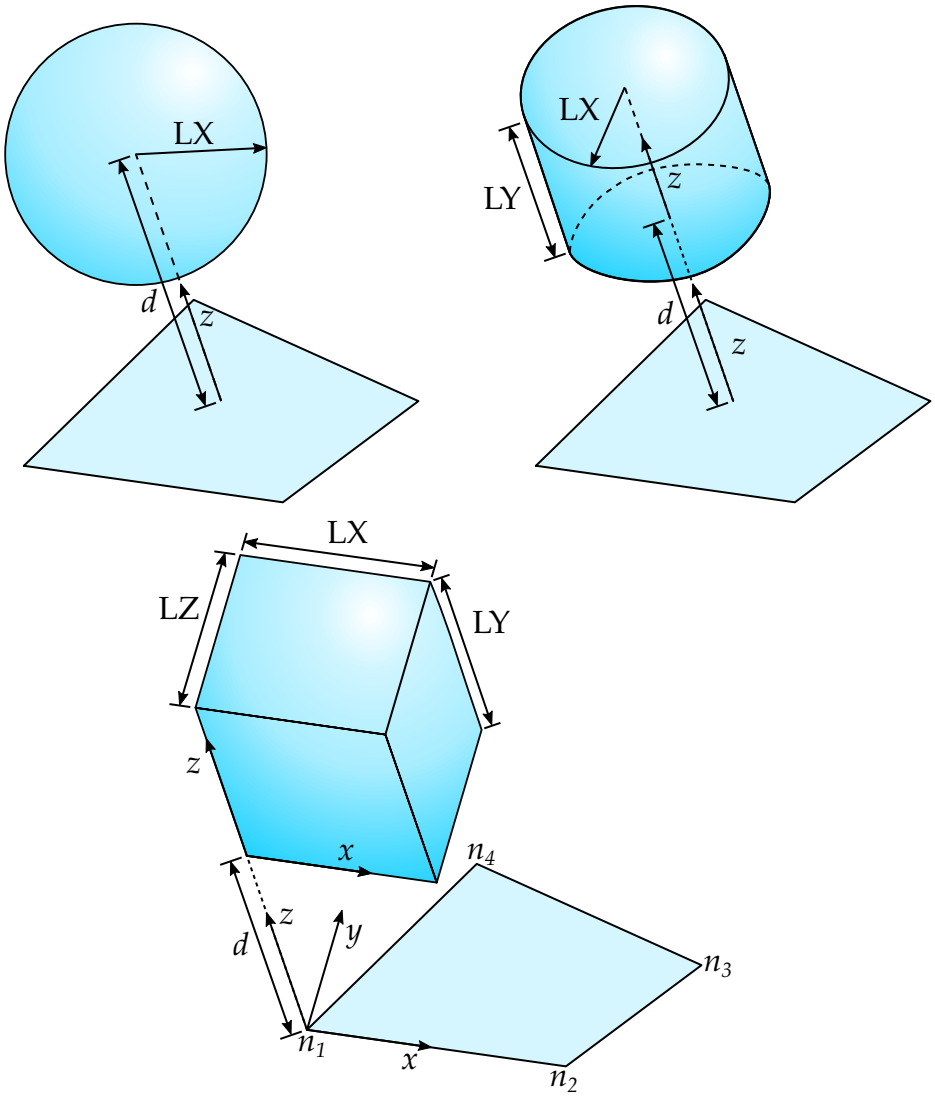

Figure 16-1. Sensor shapes

VARIABLE	DESCRIPTION
LY	Length(cylinder)/length in local *y*-direction(rectangular) of the sensor. See Remarks 1 and 2.
LZ	Length in local *z*-direction(rectangular) of the sensor. See Remarks 1 and 2.

Remarks:

1. **Sensor Shape.** The sensor can be three different shapes, a sphere, cylinder, or rectangle, depending on the length fields defined (LX, LY, and LZ). See Figure 16-1.

   a) If only LX is defined, then the sensor is sphere with radius LX.

b) If LX and LY are defined, then the sensor is a cylinder with radius LX and length LY.

c) If LX, LY, and LZ are defined, then the sensor is a rectangular prism with side lengths LX, LY, and LZ.

2. **OFFSET.** Each segment has a sensor. The distance between the segment and the sensor is defined by OFFSET (d in Figure 16-1) in the normal direction of the segment. This distance is constant, that is, the sensor moves along with the segment.

a) For a spherical sensor, OFFSET is the distance between the sphere center and the segment center. OFFSET should be larger than the radius of the sensor to prevent the segment from cutting the sphere.

b) For a cylindrical sensor, OFFSET is the distance from the segment center to the center of the base of the cylinder. The cylinder is extruded in the normal direction of the segment from the base.

c) For a rectangular sensor, OFFSET is the distance from the segment to the sensor. The sensor is defined using the segment's coordinates system. The base point for the coordinate system is n_1 from the segment definition (see *SET_SEGMENT). The local x-direction is along the vector $n_2 - n_1$. The local z-direction is the segment normal direction, and the local y-direction is constructed by the cross product of the local z- and x-directions.

3. **Output.** The output parameters in the cpm_sensor file are:

velx	=	x-velocity
vely	=	y-velocity
velz	=	z-velocity
velr	=	velocity
temp	=	temperature
dens	=	density
pres	=	pressure

These values are averaged on the number of particles in the sensor. The sensor should be large enough to contain a reasonable number of particles for the averages.

Example:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|.
$ INPUT:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|.
*DATABASE_CPM_SENSOR
      0.01
```

```
$   SEGSID    OFFSET        LX        LY
       123       5.0       5.0
       124      -0.2       0.1
       125       0.7       0.6       1.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|..
$ The segment set id: 123 has 1 segment.
$ The segment set id: 124 has 1 segment.
$ The segment set id: 125 has 11 segments.

$ Each segment has an ID defined in D3HSP
$ The D3HSP file looks like the following:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|..
Segments for sensor          1
  Sensor id        n1        n2        n3        n4
         1      3842      3843      3848      3847
 Segments for sensor          2
  Sensor id        n1        n2        n3        n4
         2      3947      3948      3953      3952

 Segments for sensor          3
  Sensor id        n1        n2        n3        n4
         3      3867      3868      2146      2145
         4      3862      3863      3868      3867
         5      3857      3858      3863      3862
         6      3852      3853      3858      3857
         7      3847      3848      3853      3852
         8      3837      3838      3843      3842
         9      3842      3843      3848      3847
        10      3832      3833      3838      3837
        11      3827      3828      3833      3832
        12      3822      3823      3828      3827
        13      1125      1126      3823      3822
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|..
```

DATABASE_CROSS_SECTION_*OPTION1_{OPTION2}

OPTION1 includes:

PLANE

SET

To define an ID and heading for the database cross section, set *OPTION2* to:

ID

Purpose: Define a cross section for resultant forces written to ASCII file secforc.

1. For the PLANE option, a set of two cards defines a cutting plane, see Figure 16-2. Based on this cutting plane, a node set and element set(s) which comprise the cross section are internally generated. Using the variable ICRFILE in *CONTROL_OUTPUT, those sets may be output for the purpose of displaying the nodes and elements of the cross section using LS-PrePost.

2. If the SET option is used, just one card is needed which identifies a node set and at least one element set. In this case the node set(s) defines the cross section, and the forces from the elements belonging to the element set(s) are summed up to calculate the section forces. Thus, the element set(s) should include elements on only one side (not both sides) of the cross section.

The cross-section should cut through deformable elements only, not rigid bodies. Cutting through reference segments for deformable solid element spot welds can lead to incorrect section forces since the constraint forces are not accounted for in the force and moment summations. Beam element modeling of welds do *not* require any special precautions.

Card Summary:

Card ID. This card is included if the ID keyword option is used.

CSID	HEADING

Card 1a.1. This card is included if the PLANE keyword option is used.

PSID	XCT	YCT	ZCT	XCH	YCH	ZCH	RADIUS

Card 1a.2. This card is included if the PLANE keyword option is used.

XHEV	YHEV	ZHEV	LENL	LENM	ID	ITYPE	

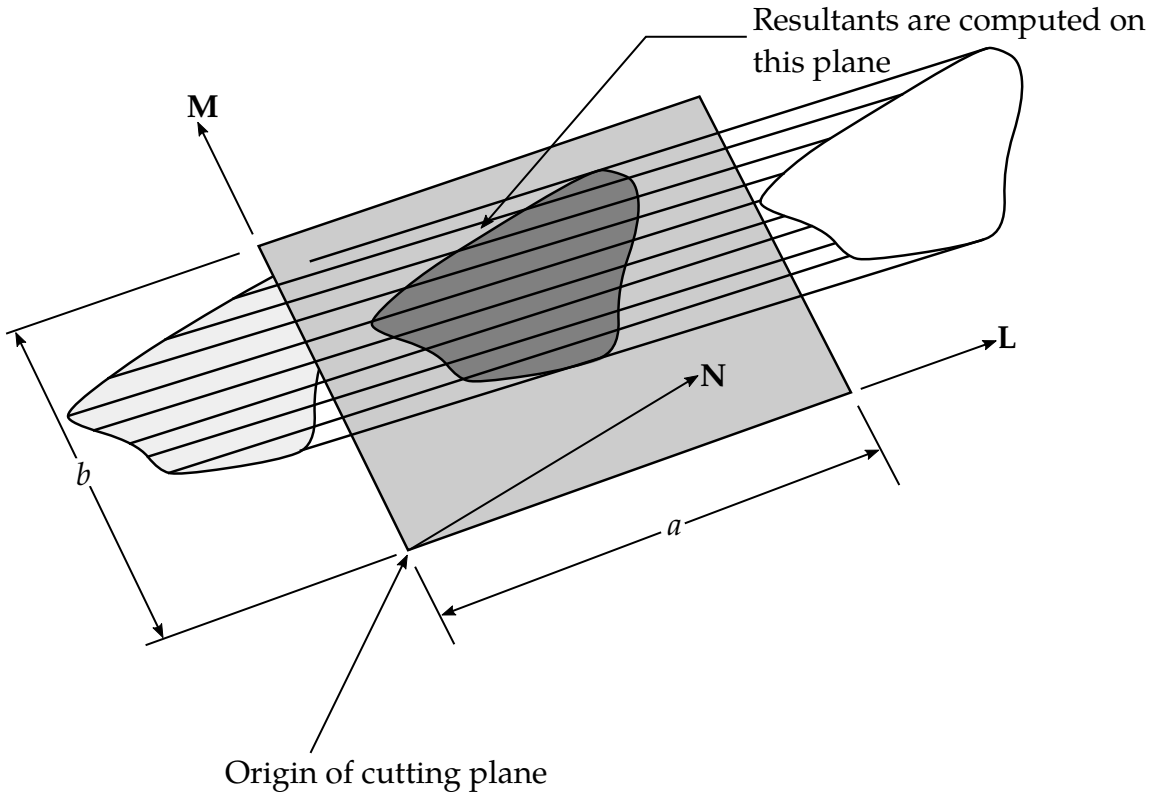

Figure 16-2. Definition of cutting plane for automatic definition of interface for cross-sectional forces. The automatic definition does not check for springs and dampers in the section. For best results the cutting plane should cleanly pass through the middle of the elements, distributing them equally on either side. Elements that intersect the edges of the cutting plane are deleted from the cross-section.

Card 1b. This card is included if the SET keyword option is used.

NSID	HSID	BSID	SSID	TSID	DSID	ID	ITYPE

Data Card Definitions:

ID Card. Additional card for ID keyword option. The heading is picked up by some of the peripheral LS-DYNA codes to aid in post-processing.

Card ID	1	2	3	4	5	6	7	8
Variable	CSID	HEADING						
Type	I	A70						

VARIABLE	DESCRIPTION
CSID	Cross section ID. This must be a unique number.
HEADING	Cross section descriptor. It is suggested that unique descriptions be used.

Plane Card 1. First additional card for PLANE keyword option.

Card 1a.1	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.

VARIABLE	DESCRIPTION
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 16-2.
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: EQ.0.0: Not used. GT.0.0: A circular cut plane will be created that is centered at (XCT,YCT,ZCT) with radius = RADIUS and has a normal vector originating at (XCT,YCT,ZCT) and pointing towards (XCH,YCH,ZCH). LT.0.0: The radius will be the absolute value of RADIUS and XCT and XCH will be nodes IDs. The node with ID XCT is the center of the circular cut plane. The normal vector

VARIABLE	DESCRIPTION
	of the plane is the vector pointing from the node with ID XCT to the node with ID XCH. YCT, ZCT, YCH, and ZCH are ignored.
	If RADIUS ≠ 0.0, the variables XHEV, YHEV, ZHEV, LENL, and LENM, which are specified on Card 1a.2, will be ignored.

Plane Card 2. Second additional card for PLANE keyword option.

Card 1a.2	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	

VARIABLE	DESCRIPTION
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**.
LENL	Length of edge a, in **L** direction.
LENM	Length of edge b, in **M** direction.
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 *updated* local system of the rigid body or accelerometer. For ITYPE = 2, the force resultants are output in the updated local coordinate system if FLAG = 1 in *DEFINE_COORDINATE_NODES or if NID is nonzero in *DEFINE_COORDINATE_VECTOR.
ITYPE	Flag that specifies whether ID above pertains to a rigid body, an accelerometer, or a coordinate system. EQ.0: Rigid body, EQ.1: Accelerometer,

VARIABLE	DESCRIPTION
	EQ.2: Coordinate system.

Set Card. Additional Card for the SET keyword option. The set option requires that the equivalent of the automatically generated input by the cutting plane capability 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 must be defined.

Card 1b	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
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 *updated* local system of the rigid body or accelerometer. For ITYPE = 2, the force resultants are output in the updated local coordinate system if FLAG = 1 in *DEFINE_COORDINATE_NODES or if NID is nonzero in *DEFINE_COORDINATE_VECTOR.
ITYPE	Flag that specifies whether ID above pertains to a rigid body, an accelerometer, or a coordinate system. EQ.0: Rigid body, EQ.1: Accelerometer,

VARIABLE	DESCRIPTION
	EQ.2: Coordinate system.

*DATABASE_DEFRAGMENT

Purpose: Output the fragment volume, radius, and maximum size at a specified time interval for the bonded DEM. Note that *DATABASE_DEFRAGMENT is applicable only to bonded DEM (*DEFINE_DE_BOND).

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY						
Type	F	I						
Default	none	0						

VARIABLE	DESCRIPTION
DTOUT	Time interval between outputs. If DT is zero, no output is printed.
BINARY	Flag for binary output: EQ.0: Do not write the file. EQ.1: Write data to an ASCII file. EQ.2: Write data to binary database binout. EQ.3: Write data to an ASCII file and to binary database binout.

*DATABASE_D3FTG

Purpose: This card activates writing binary plot database d3ftg, which saves fatigue analysis results.

Card 1	1	2	3	4	5	6	7	8
Variable	BINARY	DT						
Type	I	F						
Default	1	0.0						

VARIABLE	DESCRIPTION
BINARY	Flag for writing the binary plot file: EQ.0: off EQ.1: write the binary plot file d3ftg.
DT	Time interval between output states in time domain fatigue analysis (see *FATIGUE_*{OPTION}*) EQ.0.0: only fatigue results at the end of the analysis are output.

Remarks:

1. **Frequency Domain Fatigue.** To activate writing the d3ftg database for frequency domain fatigue analysis, including random vibration fatigue (*FREQUENCY_DOMAIN_RANDOM_VIBRATOIN_FATIGUE) and steady state vibration fatigue (*FREQUENCY_DOMAIN_SSD_FATIGUE), either this keyword or *DATABASE_FREQUENCY_BINARY_D3FTG can be used. More details about the d3ftg database can be found in *DATABASE_FREQUENCY_BINARY_*OPTION.*

2. **Time Domain Fatigue.** For time domain fatigue (keyword *FATIGUE_*OPTION*), if DT is blank or is defined as 0, only the fatigue results at the end of the analysis are saved to d3ftg, so d3ftg includes only one state. If DT is defined as a nonzero value, the fatigue results at every DT are saved in d3ftg, so d3ftg includes multiple states. With this option, the user can observe the evolution of the fatigue results (e.g. the cumulative damage ratio) in the model over time.

*DATABASE_D3MAX

Purpose: Activate writing binary plot database d3max (accessible with LS-PrePost) d3max contains the maximum or minimum (depending on OUTPUT in Card 1) values of stresses or strains for each element or certain elements during a transient analysis. The maximum (or minimum) stresses/strains can be dumped to d3part database if needed. *DATABASE_MAX_*OPTION* restricts which elements have their maximum (or minimum) values extracted (see Remark 1). Note that when this keyword is enabled, stresses are always output. The output of the strains, however, is controlled by STRFLG on *DATABASE_EXTENT_BINARY.

Card 1	1	2	3	4	5	6	7	8
Variable	DTCHECK	ME	PSTRS	PSTRN	IFILT	OUTPUT	FCUTOUT	
Type	F	I	I	I	I	I	F	
Default	none	1	0	0	0	0	0.	

VARIABLE	DESCRIPTION
DTCHECK	Time step for checking and updating maximum (or minimum) values. For instance, if DTCHECK = 10^{-6}, LS-DYNA will check and update the maximum values every 10^{-6} seconds (assuming for this example the time units are seconds). It will compare the current values (stress or strain) with the maximum values up to now. If the current values are larger, the maximum values will be replaced by the current values. Otherwise, the maximum values will remain unchanged. If OUTPUT > 2, minimum values are output instead of maximum. The same algorithm applies to the minimum values, except the minimum values are replaced if the current values are smaller.
ME	Method for extracting stresses / strains: EQ.1: Extract maximum (or minimum) stress / strain during transient analysis. EQ.2: Extract maximum (or minimum) stress / strain after transient analysis (not used).
PSTRS	Output maximum or minimum principal stress in place of maximum or minimum normal stresses in the global coordinate system (see Remark 2):

VARIABLE	DESCRIPTION
	EQ.0: No EQ.1: Yes
PSTRN	Output maximum or minimum principal strain in place of the maximum or minimum normal strains in the global coordinate system (see Remark 2): EQ.0: No EQ.1: Yes
IFILT	Use filter: EQ.0: No EQ.1: Use low pass 2nd order Butterworth filter
OUTPUT	Output format and flag to determine whether minimum or maximum values are output (see Remark 4): EQ.0: Write maximum stress / strain to d3max EQ.1: Append the maximum stress / strain results to d3part EQ.2: Write the maximum stress / strain results to d3part instead of the normal data that goes into d3part (negative time stamps are used in d3part to distinguish when this is done from the normal d3part output, which saves time history results for selected parts) EQ.3: Write minimum stress / strain to d3max EQ.4: Append the minimum stress / strain results to d3part EQ.5: Write the minimum stress / strain results to d3part instead of the normal data that goes into d3part (negative time stamps are used in d3part to distinguish when this is done from the normal d3part output, which saves time history results for selected parts EQ.6: Same as OUTPUT = 0, but map the results onto the original geometry instead of using the deformed geometry in d3max. EQ.7: Same as OUTPUT = 3, but map the results onto the original geometry instead of using the deformed geometry in d3max
FCUTOUT	Cutout frequency for Butterworth filter

Remarks:

1. **D3MAX.** This keyword activates saving the maximum stress and/or strain values of elements in a structure during a transient simulation to the d3max or d3part database. The time step, DTCHECK, determines how often the maximum values are updated. Only elements or sets of elements specified with *DATABASE_MAX_*OPTION* have nonzero values for the stress/strain in d3max. Stresses/strains for all elements not included in a *DATABASE_MAX_*OPTION* are also output to d3max but with zero values.

2. **Principal stress and strain.** By default, the maximum or minimum stress / strain components in the global coordinate system are output. When PSTRS = 1, the maximum or the minimum of the principal stress components are stored at the location of the normal stresses in the global coordinate system in the output file. Similarly, when PSTRN = 1, the maximum or the minimum of the principal strain components are stored at the location of normal strain components in the global coordinate system. The maximum or minimum shear stress / strain components in the global coordinate system are output regardless of the setting of PSTRS / PSTRN.

3. **Save d3max data to d3part.** With OUTPUT = 2, the maximum stress / strain data for each state is output to the d3part database. The normal data in d3part is not output in this case. The time stamps of the output data to d3part are changed to negative values. Only the elements in the parts specified with PSETID on *DATABASE_BINARY_D3PART are saved to d3part. Elements specified with *DATABASE_MAX_*OPTION* that are not in parts in PSETID are not in the output. However, if *DATABASE_MAX_*OPTION* is used and does not include some elements that are in the parts of PSETID, the elements that are in the parts but not in *DATABASE_MAX_*OPTION* will have values of zero in the output to d3part.

4. **OUTPUT.** OUTPUT sets whether the maximum or the minimum stress / strain is output to d3max or d3part. Maximum (OUTPUT = 0, 1, 2, or 6) should be used if the stress values are mostly positive (tension), and minimum (OUTPUT = 3, 4, 5, or 7) should be used if the stress values are mostly negative (compression). For OUTPUT = 0 to 5, deformed or updated geometry is used in d3max or d3part. For OUTPUT = 6 and 7, original geometry is used in d3max. Original geometry can be useful if the model experiences large deformation and it is difficult to review the stress plot on the deformed geometry.

*DATABASE_EXTENT

This family of keywords control to some extent the content of specific output databases. They are listed below in alphabetical order:

*DATABASE_EXTENT_AVS

*DATABASE_EXTENT_BINARY

*DATABASE_EXTENT_D3PART

*DATABASE_EXTENT_INTFOR

*DATABASE_EXTENT_MOVIE

*DATABASE_EXTENT_MPGS

*DATABASE_EXTENT_SSSTAT

The BINARY option of *DATABASE_EXTENT applies to the binary databases d3plot, d3thdt, and d3part. In the case of the d3part database, variables set using the D3PART option will override the corresponding variables of the BINARY option. See also *DATABASE_BINARY_*OPTION*.

The AVS, MOVIE, and MPGS databases will be familiar to users that have a use for those databases.

*DATABASE_EXTENT_AVS

Purpose: Control the content written to the avsflt database. See AVSFLT option to *DATABASE card.

Variable Cards. Define as many cards as needed. Input ends at next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	VTYPE	COMP						
Type	I	I						

VARIABLE	DESCRIPTION
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 16-1, VTYPE.EQ.1: Table 16-2, VTYPE.EQ.2: not supported, VTYPE.EQ.3: Table 16-3, 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.

Table 16-1. Nodal Quantities

Component ID	Quantity
1	x, y, z-displacements
2	x, y, z-velocities
3	x, y, z-accelerations

Table 16-2. 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 16-3. Shell and Thick Shell Element Quantities

Component ID	Quantity
1	mid-surface x-stress
2	mid-surface y-stress
3	mid-surface z-stress
4	mid-surface xy-stress
5	mid-surface yz-stress
6	mid-surface xz-stress
7	mid-surface 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

Component ID	Quantity
13	inner surface zx-stress
14	inner surface effective plastic strain
15	outer surface x-stress
16	outer surface y-stress
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 M_{xx} (4-node shell)
23	bending moment M_{yy} (4-node shell)
24	bending moment M_{xy} (4-node shell)
25	shear resultant Q_{xx} (4-node shell)
26	shear resultant Q_{yy} (4-node shell)
27	normal resultant N_{xx} (4-node shell)
28	normal resultant N_{yy} (4-node shell)
29	normal resultant N_{xy} (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

Component ID	Quantity
43	outer surface *yz*-strain
44	outer surface *zx*-strain
45	internal energy
46	mid-surface effective stress
47	inner surface effective stress
48	outer surface effective stress
49	mid-surface max. principal strain
50	through thickness strain
51	mid-surface 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 16-4. 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

*DATABASE_EXTENT_BINARY_{OPTION}

Purpose: Control to some extent the content of binary output databases d3plot, d3thdt, and d3part. See also *DATABASE_BINARY_*OPTION* and *DATABASE_EXTENT_D3PART. The content of the binary output database intfor may be modified using *DATABASE_EXTENT_INTFOR.

Available options include:

<BLANK>

COMP

The *DATABASE_EXTENT_BINARY_COMP reduces the content of binary output databases d3plot and d3eigv to a maximum of only seven categories (see the seven variables of Card 1a) and therefore overrides most of the settings in *DATABASE_EXTENT_BINARY. Furthermore, a maximum of three through-thickness shell or thick shell integration points are output to all binary databases (d3plot, d3eigv, d3part, d3drlf, etc.) when *DATABASE_EXTENT_BINARY_COMP is used, that is, if MAXINT in *DATABASE_EXTENT_BINARY is set greater than 3, that variable is automatically reset to 3.

Card Summary:

Card 1a. This card is included if the COMP keyword option is used

IGLB	IXYZ	IVEL	IACC	ISTRS	ISTRA	ISED	

Card 1b. This card is included if no keyword option is used.

NEIPH	NEIPS	MAXINT	STRFLG	SIGFLG	EPSFLG	RLTFLG	ENGFLG

Card 2. This card is included if no keyword option is used.

CMPFLG	IEVERP	BEAMIP	DCOMP	SHGE	STSSZ	N3THDT	IALEMAT

Card 3. This card may be included if no keyword option is used. This card is optional.

NINTSLD	PKP_SEN	SCLP	HYDRO	MSSCL	THERM	INTOUT	NODOUT

Card 4. This card may be included if no keyword option is used. This card is optional.

DTDT	RESPLT	NEIPB	QUADSLD	CUBSLD	DELERES		

Data Card Definitions:

COMP Card. For COMP option, use this card (no Cards 2-4).

Card 1a	1	2	3	4	5	6	7	8
Variable	IGLB	IXYZ	IVEL	IACC	ISTRS	ISTRA	ISED	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

VARIABLE	DESCRIPTION
IGLB	Output flag for global and part history variables such as internal energy, kinetic energy, rigid body velocity, etc., that is, the sort of data that can also be output to glstat and matsum. EQ.0: No EQ.1: Yes
IXYZ	Output flag for nodal coordinates: EQ.0: No EQ.1: Yes
IVEL	Output flag for nodal velocities: EQ.0: No EQ.1: Yes
IACC	Output flag for nodal accelerations: EQ.0: No EQ.1: Yes
ISTRS	Output flag for stress tensor and “plastic strain”: EQ.0: No EQ.1: Yes
ISTRA	Output flag for strain tensor: EQ.0: No

VARIABLE	DESCRIPTION
	EQ.1: Yes
ISED	Output flag for strain energy density: EQ.0: No EQ.1: Yes

Card 1b	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	10				

VARIABLE	DESCRIPTION
NEIPH	Number of additional integration point history variables written to the binary databases (d3plot, d3part, d3drlf) for solid elements and SPH particles. 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 the history data that is needed for plotting should be stored before the data which is not of interest. See also *DEFINE_MATERIAL_HISTORIES. For output of additional integration point history variables for solid elements to the elout database, see the field OPTION1 for *DATABASE_ELOUT.
NEIPS	Number of additional integration point history variables written to the binary databases (d3plot, d3part, d3drlf) for both shell and thick shell elements for each integration point; see NEIPH above and *DEFINE_MATERIAL_HISTORIES. For output of additional integration point history variables for shell and thick shell elements to the elout database, see the fields OPTION2 and OPTION3, respectively, in *DATABASE_ELOUT.

VARIABLE	DESCRIPTION

MAXINT

Number of shell and thick shell through-thickness integration points for which output is written to d3plot. This does not apply to the strain tensor output flagged by STRFLG.

MAXINT (def = 3)	Number of Integration Points	Description
3	> 3 (even & odd)	Results are output for the outermost (top) and innermost (bottom) integration points together with results for the neutral axis.
3	1	All three results are identical.
> 3	≤ MAXINT	Results for the first MAXINT integration points in the element will be output.
≠ 3	Even	See above. This will *exclude* mid-surface results, whereas when MAXINT = 3 mid-surface results are calculated and reported.
< 0	Any	MAXINT integration points are output for each in plane integration point location and no averaging is used. This can greatly increase the size of the binary databases d3plot, d3thdt, and d3part.

See Remark 1 for more information.

STRFLG

Flag for output of strain tensors. STRFLG is interpreted digit-wise STRFLG = [*NML*],

$$\text{STRFLG} = L + M \times 10 + N \times 100$$

L.EQ.1: Write strain tensor data to d3plot, elout, and dynain. 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.

M.EQ.1: Write plastic strain data to d3plot.

N.EQ.1: Write thermal strain data to d3plot.

Examples. For STRFLG = 11 (011) LS-DYNA will write both strain and plastic strain tensors, but no thermal strain tensors. Whereas

VARIABLE	DESCRIPTION
	for STRFLG = 110, LS-DYNA will write plastic and thermal strain tensors but no strain tensors. For more information and supported elements and materials, see Remark 10.
SIGFLG	Flag for including the stress tensor for shells and solids. EQ.1: Include (default), EQ.2: Exclude for shells, include for solids. EQ.3: Exclude for shells and solids.
EPSFLG	Flag for including the effective plastic strains for shells and solids: EQ.1: Include (default), EQ.2: Exclude for shells, include for solids. EQ.3: Exclude for shells and solids.
RLTFLG	Flag for including stress resultants in the shell LS-DYNA database: EQ.1: Include (default), EQ.2: Exclude.
ENGFLG	Flag for including shell, tshell, and beam internal energy density and shell thickness: EQ.1: Include (default), EQ.2: Exclude.

Card 2	1	2	3	4	5	6	7	8
Variable	CMPFLG	IEVERP	BEAMIP	DCOMP	SHGE	STSSZ	N3THDT	IALEMAT
Type	I	I	I	I	I	I	I	I
Default	0	0	0	1	1	1	2	1

VARIABLE	DESCRIPTION
CMPFLG	Flag to indicate the coordinate system for output of stress and strain of solids, shells, and thick shells comprised of orthotropic or

VARIABLE	DESCRIPTION
	anisotropic materials. CMPFLG affects d3plot, d3part, eloutdet, and elout, with exceptions as noted below. EQ.-1: Same as 1, but for *MAT_FABRIC (FORM = 14 or -14) and *MAT_FABRIC_MAP the stress and strain are in engineering quantities instead of Green-Lagrange strain and 2nd Piola-Kirchhoff stress. EQ.0: Global coordinate system with the exception of elout for shells (see EOCS in *CONTROL_OUTPUT). EQ.1: Local material coordinate system (as defined by AOPT and associated parameters in the *MAT input, and if applicable, by angles B1, B2, etc. in *SECTION_SHELL, *SECTION_TSHELL, or *PART_COMPOSITE, and by optional input in the *ELEMENT data). The effect of CMPFLG = 1 on shell output in elout is overridden by EOCS = 1 or 2 in *CONTROL_OUTPUT or by OPTION2 > 0 in *DATABASE_ELOUT. These overriding conditions do not apply to eloutdet.
IEVERP	Every output state for the d3plot database is written to a separate file. EQ.0: More than one state can be on each plot file, EQ.1: One state only on each plot file.
BEAMIP	Number of beam integration points for output. This option applies to integrated beams but also generates certain limited output for resultant beams. See Remark 2.
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 all nodal velocities and accelerations are eliminated from the database. EQ.4: On, rigid body data compression active and all nodal velocities and accelerations are eliminated from the database. EQ.5: On, rigid body data compression active and rigid nodal data are eliminated from the database. Only 6 DOF rigid body motion is written. EQ.6: On, rigid body data compression active, rigid nodal data, and all nodal velocities and accelerations are eliminated

VARIABLE	DESCRIPTION
	from the database. Only 6 DOF rigid body motion is written.
SHGE	Flag for including shell hourglass energy density: EQ.1: Off (default), no hourglass energy written, EQ.2: On.
STSSZ	Flag for including 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	Flag for including material energy in 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)

Card 3	1	2	3	4	5	6	7	8
Variable	NINTSLD	PKP_SEN	SCLP	HYDRO	MSSCL	THERM	INTOUT	NODOUT
Type	I	I	F	I	I	I	A	A
Default	1	0	1.0	0	0	1 or 2	none	none

VARIABLE	DESCRIPTION
NINTSLD	Number of solid element integration points written to the LS-DYNA database. When NINTSLD is set to 1 (default) or to any value other than 8, integration point values are averaged and only those averages are written output. To obtain values for individual integration points, set NINTSLD to 8, even if the multi-integration point solid has fewer than 8 integration points.

VARIABLE	DESCRIPTION
PKP_SEN	Flag to output the peak pressure and surface energy (energy per unit area) 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. EQ.0: No data is written. EQ.1: Output the peak pressures and surface energy by contact interface.
SCLP	Weighting or smoothing factor used to apply exponential smoothing to the calculation of peak pressure. It must be a value between 0.0 and 1.0. When set to a value less than 1.0, the calculation uses previous values of the pressure.
HYDRO	Either 3, 5 or 7 additional history variables useful to shock physics are output as the last history variables to d3plot (does not apply to elout). For HYDRO = 1, the internal energy per reference volume, the reference volume, and the pressure from bulk viscosity are added to the database; for HYDRO = 2, the relative volume and current density are also added; and for HYDRO = 4, volumetric strain (defined as relative volume – 1.0) and hourglass energy per unit initial volume are additionally added. These history variables are not valid for ALE elements.
MSSCL	Output nodal and part information related to mass scaling into the d3plot database. This option can be activated only if DT2MS < 0.0. See Remark 3. Also, see *CONTROL_TIMESTEP. MSSCL is interpreted digit-wise, MSSCL = [*ML*], $\text{MSSCL} = L + M \times 10$ *L* sets the nodal mass output: L.EQ.0: No incremental nodal mass data is written. L.EQ.1: Output incremental nodal mass. L.EQ.2: Output percentage increase in nodal mass. *M* determines what mass is output for each part into d3plot: M.EQ.0: Output original mass for each part.

VARIABLE	DESCRIPTION
	M.EQ.1: Output effective mass for each part. Effective mass is the original mass plus the incremental mass.
THERM	Output of thermal data (if applicable) to d3plot. The effect of the parameter is different in thermal-only analyses compared to coupled simulations. For coupled simulations: EQ.1: Output temperature (default). EQ.2: Output temperature and flux. EQ.3: Output temperature, flux, and the temperatures at the shell's bottom and top surfaces. For thermal-only simulations: EQ.1: Output temperature and flux. EQ.2: Output temperature and flux (default). EQ.3: Output temperature, flux, and the temperatures at the shell's bottom and top surfaces.
INTOUT	Output stress/strain at all integration points for detailed element output in the ASCII file eloutdet. DT and BINARY of *DATABASE_ELOUT apply to eloutdet. See Remark 4. EQ.STRESS: When stress output is required EQ.STRAIN: When strain output is required EQ.ALL: When both stress and strain output are required
NODOUT	Output extrapolated stress/strain at connectivity nodes for detailed element output in the ASCII file eloutdet. DT and BINARY of *DATABASE_ELOUT apply to eloutdet. See Remark 4. EQ.STRESS: When stress output is required EQ.STRAIN: When strain output is required EQ.ALL: When both stress and strain output are required EQ.STRESS_GL: When nodal averaged stress output along the global coordinate system is required EQ.STRAIN_GL: When nodal averaged strain output along the global coordinate system is required EQ.ALL_GL: For global nodal averaged stress and strain output

Card 4	1	2	3	4	5	6	7	8
Variable	DTDT	RESPLT	NEIPB	QUADSLD	CUBSLD	DELERES		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
DTDT	Output of node point Δtemperature/Δtime data to d3plot. EQ.0: No output (default) EQ.1: Output $\Delta T/\Delta t$
RESPLT	Output of translational and rotational residual forces to d3plot and d3iter. EQ.0: No output EQ.1: Output residual
NEIPB	Number of additional element or integration point history variables written to the binary databases (d3plot, d3part, d3drlf) for beam elements; see NEIPH above, BEAMIP and *DEFINE_MATERIAL_HISTORIES. See also Remark 12. For output of additional integration point history variables for beam elements to the elout database, see the variable OPTION4 in *DATABASE_ELOUT.
QUADSLD	(Under development) Output flag for quadratic solid types 24, 25, and 26: EQ.0: Average stress and strain and rendered with 8 nodes EQ.1: Average stress and strain and rendered with all edge and face nodes EQ.2: All integration points written and rendered with all edge and face nodes
CUBSLD	(Under development) Output flag for cubic solids types 27, 28, and 29: EQ.0: Average stress and strain and rendered with 8 nodes

VARIABLE	DESCRIPTION
	EQ.1: Average stress and strain and rendered with all edge and face nodes
	EQ.2: All integration points written and rendered with all edge and face nodes
DELERES	Output flag for results of deleted elements:
	EQ.0: No results output (all zero)
	EQ.1: Last available results, such as stresses and history variables, are written to d3plot and d3part.

Remarks:

1. **MAXINT field.** If MAXINT is set to 3, then mid-surface, inner-surface, and outer-surface stresses are output at the center of the element. 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. **BEAMIP field.** BEAMIP causes the output of additional data to d3plot and d3thdt for beam elements. For integrated beams, such as the Hughes-Liu beam (ELFORM = 1), the additional data consists of stresses and strains at the first BEAMIP integration points. The order of the output data matches the order of the integration points. 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 (ELFORM = 4 and 5), the transverse shear stress components are not used in the formulation.

   No data is output for Belytschko-Schwer resultant beams (ELFORM = 2), discrete beams (ELFORM = 6), or linear beams (ELFORM = 12), except under these conditions:

   a) For ELFORM = 2 and 13, beam elements have a linear distribution of moment along their length. The default moment data output to d3plot, d3dtdt,

and elout are the moments at the Node 1 end of the beams only. This restriction presents an incomplete picture of the moment distribution in the beam and may miss the maximum moment value. However, the moments at both ends of the beam can be obtained by setting BEAMIP greater than or equal to 3 (for ELFORM = 13, this is available only in releases starting from R16). In this case, the moments at both ends of the beam are written to the d3plot and d3thdt files in the positions where post-processors expect to find the following data components:

Integration point	Post-processing data component	Actual meaning
2	ZX shear stress	Moment about s-axis at Node 1 end
2	Plastic strain	Moment about s-axis at Node 2 end
2	Axial strain	Moment about t-axis at Node 1 end
3	Axial stress	Moment about t-axis at Node 2 end

b) Belytschko-Schwer beams (ELFORM = 2) with material types 191 and 209 automatically increase BEAMIP and write additional data specific to those material models as well as the moments at both ends of the beam. See Remarks under these material types.

c) Discrete beams (ELFORM = 6) with material types 205 and 208 automatically increase BEAMIP and write additional data specific to those material models. See Remarks under these material types.

3. **Mass scaling.** 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.

4. **Estimated stress and strain output to eloutdet for nodes.**

a) When NODOUT is set to STRESS, STRAIN, or ALL. Each node of the element nodal connectivity will be output. See Example 1.

b) When NODOUT is set to STRESS_GL, STRAIN_GL, or ALL_GL. Averaged nodal results are calculated by:

$$\frac{1}{n}\sum_{i=1}^{n} c_{\mathrm{ip}(i)}$$

where n is the number of elements sharing the node and $c_{\mathrm{ip}(i)}$ is the contribution from the closest integration of element i to the node. Averaged

nodal values are always output in the global coordinate system. See Example 2.

5. **Contents of eloutdet.** Available stress/strain components in eloutdet stress components include 6 stress components (sig-*xx*, sig-*yy*, sig-*zz*, sig-*xy*, sig-*yz*, sig-*zx*), yielding status, and effective plastic strain. Strain components include 6 strain components.

6. **Shell element output at integration points.** Stresses at all integration points can be output. The strain at the top and bottom integration layer can be output. At a connective node the extrapolated stress and strain at the top and bottom layer can be output.

7. **Thick shells.** Thick shell element output includes the six stress components at each integration point. Strain at the top and bottom layers can be output. At the element node, values at the bottom layer are extrapolated to yield the values of nodes 1 - 4, and values at the top layer are extrapolated to yield values of nodes 5 - 8.

8. **Integration point locations.** Stresses and strain at all integration points can be output. The integration point order is as follows:

   a) point #1 is the point closest to node #1 in the connectivity array

   b) point #2 is the closest point to node #2, etc

   c) For tetrahedral types 4, 16, and 17 with 5 integration points, point #5 is the midpoint.

   d) For the nodal points, values at the integration points are extrapolated.

9. **Reporting residual forces and moments.** The output of residual forces and moments is supported for implicit and double precision only. With this option, the forces and moments appear under the *Ndv* button in the fringe menu in LS-PrePost. The residual for rigid bodies is distributed to the constrained nodes for the body without scaling for the purpose of capturing the complete residual vector.

10. **Calculation of strains (STRFLG).** The strain tensors ε that are output to the d3plot database are calculated using proper time integration of the rate-of-deformation tensor $\mathbf{D}$. More specifically, to assert objectivity of the resulting strain, it is for solids using a Jaumann rate of strain, whereas for shells, it uses the co-rotational strain rate. In mathematical terms, the integration uses the following strain rates

$$\dot{\varepsilon} = \mathbf{D} - \varepsilon\mathbf{W} + \mathbf{W}\varepsilon \quad \text{(solids)}$$

$$\dot{\varepsilon} = \mathbf{D} - \varepsilon\boldsymbol{\Omega} + \boldsymbol{\Omega}\varepsilon \quad \text{(shells)}$$

where **W** is the spin tensor and $\mathbf{\Omega} = \dot{\mathbf{Q}}\mathbf{Q}^T$ is the rotational velocity of the co-rotational system **Q** used for the shell element in question, taking into account invariant node numbering and such. This is to say that the resulting strains would be equal to the Cauchy stress for a hypo-elastic material (*MAT_ELASTIC) with a Young's modulus of 1 and a Poisson's ratio of 0. This should be kept in mind when interpreting the results since they are not invariant to changes in element formulations and possibly nodal connectivities.

11. **Plastic and thermal strain (STRFLG).** The algorithm for writing plastic and thermal strains, which is also activated using STRFLG, is a modification of the algorithm used for mechanical strains (see Remark 10).

    a) For solids the element average strain in the global system having 6 components is written (local system if CMPFLG is set).

    b) For shells both plastic and thermal strains have 6 components. The thermal strain is written as a single tensor as in the solid case. The plastic strain output consists of 3 plane-averaged tensors: one for the bottom, one for the middle, and one for the top. For an even number of through-thickness integration points, the middle is taken to be the average of the two integration points closest to the mid surface. Currently, only the following element/material combinations are supported, but others *will* be added upon request.

Thermal strain tensors			*Plastic strain tensors*		
Shells	Solids	Materials	Shells	Solids	Materials
2, 16, 23	1, 2	Add thermal expansion, 255	2, 16, 23	1, 2	15, 24, 106, 123, 224, 251, 255

12. **History variables for beams (NEIPB).** In general, NEIPB follows the same conventions as NEIPH and NEIPS do for solid and shell elements and is supported in LS-PrePost v4.3 or later. Average, min, and max values for each element are output, including data for resultant elements. If BEAMIP is nonzero, then element data is complemented with BEAMIP integration point values that can be examined individually. Beam history data is post-processed similarly to that of solid and shell element history data.

Example 1:

Excerpt from eloutdet file for a shell element with two through-thickness integration points and four in-plane integration points, with INTOUT = STRESS and NODOUT = STRESS:

```
element  materl
     ipt  stress  sig-xx   sig-yy   sig-zz   sig-xy   sig0yz   sig-zx   yield         location
       1-      1
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  1
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  2
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  3
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  4
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       21
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       22
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       20
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       19
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  1
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  2
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  3
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  4
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       21
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       22
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       20
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       19
```

Example 2:

Excerpt from eloutdet file for averaged nodal strain:

```
nodal strain calculations for time step   24  (at time 9.89479E+01 )
 node (global)
          strain   eps-xx     eps-yy         eps-zz   eps-xy       eps-yz      eps-zx
          1-
 lower surface     2.0262E-01 -2.6058E-02 -7.5669E-02 -5.1945E-03  0.0000E+00  0.0000E+00
 upper surface     2.0262E-01 -2.6058E-02 -7.5669E-02 -5.1945E-03  0.0000E+00  0.0000E+00
          2-
 lower surface     1.9347E-01  2.3728E-04 -8.3019E-02 -1.4484E-02  0.0000E+00  0.0000E+00
 upper surface     1.9347E-01  2.3728E-04 -8.3019E-02 -1.4484E-02  0.0000E+00  0.0000E+00
          3-
 lower surface     2.0541E-01 -5.7521E-02 -6.3383E-02 -1.7668E-03  0.0000E+00  0.0000E+00
 upper surface     2.0541E-01 -5.7521E-02 -6.3383E-02 -1.7668E-03  0.0000E+00  0.0000E+00
          ⋮           ⋮          ⋮           ⋮          ⋮           ⋮           ⋮
```

*DATABASE_EXTENT_D3PART

The following cards control content to the d3part binary database (Card 3 is optional). The parameters given here will supercede the corresponding parameters in *DATABASE_EXTENT_BINARY when writing the d3part binary database. See also *DATABASE_BINARY_D3PART which defines the output interval for d3part and the set of parts included in d3part.

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

Card 2	1	2	3	4	5	6	7	8
Variable		IEVERP			SHGE	STSSZ		
Type		I			I	I		
Default		0			0	0		

Card 3	1	2	3	4	5	6	7	8
Variable	NINTSLD							
Type	I							
Default	1							

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

VARIABLE	DESCRIPTION
	defined materials the history data that is important for plotting should be stored before the data that 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. See Remark 1.
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 the stress tensor for shells: EQ.1: Include (default), EQ.2: Exclude.
EPSFLG	Flag for including the effective plastic strains for shells: EQ.1: Include (default), EQ.2: Exclude.
RLTFLG	Flag for including stress resultants for shells: EQ.1: Include (default), EQ.2: Exclude.
ENGFLG	Flag for including shell internal energy density and shell thickness: EQ.1: Include (default), EQ.2: Exclude.
IEVERP	Every plot state for the d3part 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 plot file, EQ.1: One state only on each plot file.
SHGE	Flag for including shell hourglass energy density: EQ.1: Off (default), no hourglass energy written,

VARIABLE	DESCRIPTION
	EQ.2: On.
STSSZ	Flag for including 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 2 below.
NINTSLD	Number of solid element integration points written. 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.

Remarks:

1. **MAXINT.** 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 that 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. If MAXINT is set to a negative number, MAXINT integration points are output for each in-plane integration point location and no averaging is used. This can greatly increase the size of the binary d3part database.

2. **Mass scaling.** 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_EXTENT_INTFOR

Purpose: The following card controls to some extent the content of the optional intfor binary database. See also *DATABASE_BINARY_INTFOR. The intfor database contains geometry and time history data pertaining to those contact surfaces which are flagged in *CONTACT with the variables SAPR and/or SBPR. The name of the intfor database may be given either on the execution line using "s=**filename**", or using the option FILE on *DATABASE_BINARY_INTFOR.

Card 1	1	2	3	4	5	6	7	8
Variable	NGLBV	NVELO	NPRESU	NSHEAR	NFORC	NGAPC	NFAIL	IEVERF
Type	I	I	I	I	I	I	I	I
Default	1	1	1	1	1	1	0	0

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	NWEAR	NWUSR	NHUF	NTIED	NENG	NPEN		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
NGLBV	Output global variables: EQ.-1: No, EQ.1: Yes (default).
NVELO	Output nodal velocity: EQ.-1: No, EQ.1: Yes (default).
NPRESU	Output pressures: EQ.-1: No,

VARIABLE	DESCRIPTION
	EQ.1: Normal interface pressure (default), EQ.2: Normal interface pressure and peak pressure, EQ.3: Normal interface pressure, peak pressure and time to peak.
NSHEAR	Output shear stresses: EQ.-1: No, EQ.1: Shear stress in *r*-direction and *s*-direction (default).
NFORC	Output forces: EQ.-1: No, EQ.1: *x*-, *y*-, *z*-force at all nodes (default).
NGAPC	Output contact gaps at all nodes and surface energy density. For Mortar contact, see Remark 1. EQ.-1: No, EQ.1: Yes (default).
NFAIL	Flag for display of deleted contact segments: EQ.0: All segments are displayed (default). EQ.1: Remove deleted contact segments from display.
IEVERF	Every interface force state for the intfor database is written to a separate file: EQ.0: More than one interface force state can be on each intfor file (default). EQ.1: One interface force output state only on each intfor file.
NWEAR	Output contact wear data; see *CONTACT_ADD_WEAR. EQ.0: No output (default). GE.1: Output wear depth. GE.2: Output sliding distance.
NWUSR	Number of user wear history variables to output from user defined wear routines; see *CONTACT_ADD_WEAR. See Remark 2.

VARIABLE	DESCRIPTION
NHUF	Number of user friction history variables to output from user defined friction routines; see *USER_INTERFACE_FRICTION (MPP only). See Remark 2.
NTIED	Output tied segments for Mortar contact. See Remark 3. EQ.0: No output EQ.1: Output
NENG	Output contact energy density for Mortar contact and SOFT = 2 contact (see also ENGOUT on *CONTROL_OUTPUT): EQ.0: No output EQ.1: Output
NPEN	Output penetration information for sliding contacts (non-TIED or TIEBREAK) when SOFT ≠ 2. A nodal field gives the penetration for each node (magnitude and direction) in the sliding interface; see also PENOUT on *CONTROL_OUTPUT. EQ.0: No output EQ.1: Output absolute penetration EQ.2: Output absolute and relative penetrations. Relative penetration is output as a percentage of the penetration at which the contact is released.

Remarks:

1. **Gaps in mortar contact.** Gaps in Mortar contact are measured with respect to the *nominal* contact surfaces of the two interacting segments. For instance, if IGNORE = 2 on *CONTACT_...MORTAR, then an initial penetration d will dislocate the tracked contact surface in the negative direction of the tracked surface normal $\mathbf{n}$. If NGAPC = 1, the gap g reported to the intfor file is still measured between the tracked and reference surface, neglecting this dislocation; thus, only *physical* gaps are reported.

2. **Wear outputs.** Wear outputs are governed by NWEAR and NWUSR and require the use of a wear model associated with the contact interface. For NWEAR the "wear depth" (NWEAR ≥ 1) and "sliding distance" (NWEAR ≥ 2) are listed under the Nodal fringe menu in LS-PrePost. Following this, NWUSR user defined history variables are listed, corresponding to user wear history variables

in a user wear routine. These are listed in the order that they are stored in the wear routine; see WTYPE ≤ 0 on *CONTACT_ADD_WEAR.

3. **Mortar tied contacts.** For all Mortar tied contacts, that is,

   *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_TIED,

   *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_TIED_WELD, and

   *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_MORTAR.

   NTIED will activate the output of an indicator stating whether a tracked segment is tied (= 1) or not (= 0); nothing is shown on the reference side. When opening the intfor file in LS-PrePost, the fringe menu will have two entries titled "tied at top" and "tied at bottom," corresponding to the top and bottom of a shell segment, respectively. The top and bottom labels are with respect to the shell normal. For obvious reasons, solid segments will only have nonzero values for the "tied at top" field. For tiebreak contact, these variables represent the relative adhesive strength in the contact, that is, $1 - D$, where D is the damage, so fully damaged segments will show 0 and undamaged segments will show 1. For the tied weld contact the entries will give the segments that have been welded to the corresponding reference side of the contact and can thus have only 0 (not welded) or 1 (welded).

*DATABASE_EXTENT_MOVIE

Purpose: Control the content written to the MOVIE databases. See movie option on *DATABASE manual entry.

Variable Cards. Define as many cards as needed. Input ends at next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	VTYPE	COMP						
Type	I	I						

VARIABLE	DESCRIPTION
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 16-1 (see DATABASE_EXTENT_AVS), VTYPE.EQ.1: Table 16-2 (see DATABASE_EXTENT_AVS), VTYPE.EQ.2: not supported, VTYPE.EQ.3: Table 16-3 (see DATABASE_EXTENT_AVS), VTYPE.EQ.4: not supported.

*DATABASE_EXTENT_MPGS

Purpose: Control the content written to the MPGS databases. The created MPGS databases consist of a geometry file and one file for each output database. See MPGS option to *DATABASE keyword.

Variable Cards. Define as many cards as needed. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	VTYPE	COMP						
Type	I	I						

VARIABLE	DESCRIPTION
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 16-1 (see DATABASE_EXTENT_AVS), VTYPE.EQ.1: Table 16-2 (see DATABASE_EXTENT_AVS), VTYPE.EQ.2: not supported, VTYPE.EQ.3: Table 16-3 (see DATABASE_EXTENT_AVS), VTYPE.EQ.4: not supported.

DATABASE_EXTENT_SSSTAT_*OPTION

The only *OPTION* is:

ID

The ID option allows the definition of a heading which will be written at the beginning of the ASCII file ssstat.

Purpose: This command defines one or more subsystems. A subsystem is simply a set of parts, grouped for convenience. The ASCII output file ssstat provides histories of energy (kinetic, internal, hourglass) and momentum (x, y, and z) for each subsystem. The ssstat file is thus similar to glstat and matsum. But whereas glstat provides data for the whole model and matsum provides data for each individual part, ssstat provides data for each subsystem. The output interval for the ssstat file is given using *DATABASE_SSSTAT. To also include histories of subsystem mass properties in the ssstat file, use *DATABASE_SSSTAT_MASS_PROPERTIES.

For *DATABASE_EXTENT_SSSTAT without the ID option, the following card(s) applies. Define as many cards as necessary. Define one part set ID per subsystem, up to 8 subsystems per card.

Card 1	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

For *DATABASE_EXTENT_SSSTAT_ID, the following card(s) applies. Define as many cards as necessary. Define one part set ID per subsystem, 1 subsystem per card.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID1	HEADING1						
Type	I	A70						

VARIABLE	DESCRIPTION
PSID*n*	Part set ID for subsystem *n*; see *SET_PART.
HEADING*n*	Heading for subsystem *n*.

*DATABASE_FATIGUE_STRESS_CYCLE

Purpose: Activate outputting number of stress cycles versus stress range in random vibration fatigue analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	BINARY							
Type	I							
Default	1							

VARIABLE	DESCRIPTION
BINARY	Flag for outputting ASCII files providing the number of stress cycles versus stress range: EQ.0: Off EQ.1: Output stress_cycle_[EID] ASCII files.

Remarks:

1. **Stress cycle versus stress range.** This keyword instructs LS-DYNA to write out ASCII files stress_cycle_[EID] which record the number of stress cycles versus stress range in random vibration fatigue analysis. EID is the element ID. The elements included in this output are defined by keywords *DATABASE_HISTORY_SOLID / SHELL / TSHELL_*{SET}*. The stress_cycle_[EID] file is in xyplot curve format and can be plotted using LS-PrePost.

.

*DATABASE_FATXML

Purpose: Process FATXML data. FATXML is an open, standardized data format based on the Extensible Markup Language (XML). It is developed by the German Research Association of Automotive Technology (Forschungsvereinigung Automobiltechnik - FAT). FATXML is designed for consistent data management in the overall CAE process chain. Schulte-Frankenfeld and Deiters [2016] give a comprehensive explanation of the FATXML data format specification.

LS-DYNA reads all lines between this keyword and the next keyword recognized by the star (*) sign, processes the data with respect to the include file structure and writes everything together in one output file called d3plot.xml.

Remarks:

We intend for a corresponding FE model to consist of one primary file with several associated include files. Each include file should contain a description with *DATABASE_FATXML data, usually at the end of the file. The primary file loads the include files via *INCLUDE_TRANSFORM with potential offset values for nodes, elements, parts, etc. (IDNOFF, IDEOFF, IDPOFF, …). All data from different include files with different offsets are collected and then summarized in d3plot.xml. Since the resulting data format is public domain, post-processors can read that data and correlate it with the associated CAE model.

Example:

```
...
*DATABASE_FATXML
<?xml version="1.0"?>
<CAE_META_DATA>
  < PART_ID NAME="TestCase">
    <PDM_DATA>
      ...
      <PDD_THICKNESS>
        < THICKNESS ID="123">1.0</THICKNESS >
        < THICKNESS ID="124">1.1</THICKNESS >
        ...
      </PDD_THICKNESS >
      ...
    </PDM_DATA >
  </PART_ID >
</CAE_META_DATA >
*END
```

*DATABASE_FORMAT

Purpose: Define the output format for binary files.

Card 1	1	2	3	4	5	6	7	8
Variable	IFORM	IBINARY						
Type	I	I						
Default	0	0						
Remarks	1	2						

VARIABLE	DESCRIPTION
IFORM	Output format for d3plot and d3thdt files: EQ.0: LS-DYNA database format (default), EQ.1: Ansys database format (generally disabled since R12; see Remark 1), EQ.2: Both LS-DYNA and Ansys database formats (generally disabled since R12; see Remark 1).
IBINARY	Flag to control the word size in the binary output files, such as d3plot and d3drlf. This variable applies only to double precision LS-DYNA executables. EQ.0: 64 bit format for binary output (default for double precision), EQ.1: 32 bit IEEE format for binary output. This reduces the volume of binary output from double precision executables by a factor of two.

Remarks:

1. **Availability restrictions.** The Ansys output option is not available in MPP and is not universally available in SMP. The LS-DYNA banner in d3hsp includes "ANSYS database format" under the list of "Features enabled" if the option is available. By default, our most recent executables do not have this feature enabled.

2. **32 bit IEEE format.** As an alternative to setting IBINARY = 1, the user may set the system environment variable LSTC_BINARY to 32ieee.

DATABASE_FREQUENCY_ASCII_*OPTION1_{OPTION2}

OPTION1 specifies one of the following frequency domain ASCII databases:

NODOUT_SSD ASCII database for nodal results for SSD (displacement, velocity, and acceleration). See also *FREQUENCY_DOMAIN_SSD.

ELOUT_SSD ASCII database for element results for SSD (stress and strain components). See also *FREQUENCY_DOMAIN_SSD.

NODFOR_SSD ASCII database for nodal force group for SSD (nodal forces for each node and reaction forces for the group). See also *FREQUENCY_DOMAIN_SSD.

SECFORC_SSD ASCII database for the cross-sectional force for SSD. See also *FREQUENCY_DOMAIN_SSD. See Remark 5.

NODOUT_PSD ASCII database for nodal PSD results (displacement, velocity, and acceleration). See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

ELOUT_PSD ASCII database for element PSD results (stress and strain components). See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

OPTION2:

MODAL_CONTRIBUTION (see Remark 7)

SUBCASE (see Remark 8)

<BLANK>

Purpose: Define output frequencies for the specified ASCII databases above. The frequencies can be different from the output frequencies for the binary databases D3SSD and D3PSD, which are defined by the keywords *DATABASE_FREQUENCY_BINARY_D3SSD and *DATABASE_FREQUENCY_BINARY_D3PSD, respectively.

Card 1	1	2	3	4	5	6	7	8
Variable	FMIN	FMAX	NFREQ	FSPACE	LCFREQ			
Type	F	F	I	I	I			
Default	0.0	0.0	0	0	0			

VARIABLE	DESCRIPTION
FMIN	Minimum frequency for output (cycles/time)
FMAX	Maximum frequency for output (cycles/time)
NFREQ	Number of frequencies for output
FSPACE	Frequency spacing option for output: EQ.0: Linear EQ.1: Logarithmic EQ.2: Biased EQ.3: Eigenfrequencies only
LCFREQ	Load curve ID defining the frequencies for output

Remarks:

1. **Output files.** The ASCII databases NODOUT_SSD, ELOUT_SSD, NODFOR_-SSD, NODOUT_PSD, and ELOUT_PSD are saved in binout files. LS-PrePost can read the binout files directly. The files can be converted to the ASCII format by feeding them to the l2a program like this:

   l2a binout*

2. **Nodal data.** The nodes to be output to NODOUT_SSD and NODOUT_PSD databases are specified by card *DATABASE_HISTORY_NODE.

3. **Element data.** The solid, beam, shell, and thick shell elements to be output to ELOUT_SSD and ELOUT_PSD are specified by the following cards:

   DATABASE_HISTORY_SOLID_*{OPTION}

   DATABASE_HISTORY_BEAM_*{OPTION}

   DATABASE_HISTORY_SHELL_*{OPTION}

   DATABASE_HISTORY_TSHELL_*{OPTION}

4. **Nodal set data.** The nodal sets to be output to NODFOR_SSD are specified by card *DATABASE_NODAL_FORCE_GROUP.

5. **Cross-sectional plane data.** Use *DATABASE_CROSS_SECTION_PLANE or *DATABASE_CROSS_SECTION_SET to specify the cross-sectional plane(s) output to SECFORC_SSD.

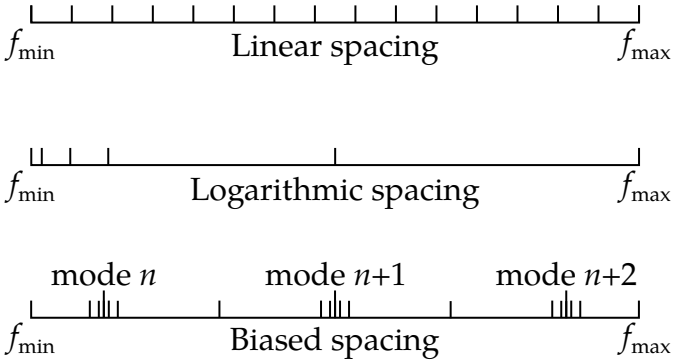

Figure 16-3. Spacing options of the frequency points.

6. **Output frequencies.** There are two methods to define the output frequencies.

   a) The first method is to define FMIN, FMAX, NFREQ, and FSPACE. FMIN and FMAX specify the frequency range of interest, and NFREQ specifies the number of frequencies at which results are required. FSPACE specifies the type of frequency spacing (linear, logarithmic, biased, or eigenfrequencies only) to be used. These frequency points for which results are required can be spaced equally along the frequency axis (on a linear or logarithmic scale), be biased toward the eigenfrequencies (the frequency points are placed closer together at eigenfrequencies in the frequency range) to obtain the detailed definition of the response close to resonance frequencies, or be just the eigenfrequencies in the frequency range defined by FMIN and FMAX. See Figure 16-3.

   b) The second method involves using a load curve (LCFREQ) to define the frequencies of interest.

7. **MODAL_CONTRIBUTION.** When the keyword option MODAL_CONTRIBUTION is present, LS-DYNA also computes the modal contribution fraction for each of the involved vibration modes. The modal contribution fraction shows the percentage of the response contributed by one particular mode over the total response. The modal contribution fraction values are saved in binout and can be processed using LS-PrePost. This keyword option is only valid for the SSD computation, that is, OPTION1 is NODOUT_SSD or ELOUT_SSD.

8. **SUBCASE.** The option SUBCASE only works when the keyword *FREQUENCY_DOMAIN_SSD_SUBCASE is present. It can only be used with *OPTION1* set to NODOUT_SSD, ELOUT_SSD, or NODFOR_SSD. With SUBCASE, Card 1 can be repeated to define individual output frequencies for each loading case defined in *FREQUENCY_DOMAIN_SSD_SUBCASE.

DATABASE_FREQUENCY_BINARY_*OPTION1_{OPTION2}

Purpose: Options for frequency domain binary output files.

OPTION1 specifies one of the following output file names:

D3ACC Binary output file for BEM acoustics (element acoustic pressure contribution and contribution percentage). See also *FREQUENCY_DOMAIN_ACOUSTIC_BEM.

D3ACS Binary output file for FEM acoustics (acoustic pressure and sound pressure level in the acoustic volume) and BEM (collocation BEM) acoustics (acoustic pressure, sound pressure level, and normal velocity on the surface of the acoustic volume). See also *FREQUENCY_DOMAIN_ACOUSTIC_FEM and *FREQUENCY_DOMAIN_ACOUSTIC_BEM.

D3ATV Binary output file for acoustic transfer vectors given by BEM acoustic analysis. See also *FREQUENCY_DOMAIN_ACOUSTIC_BEM_ATV.

D3ERP Binary output file for ERP (Equivalent Radiated Power) density and some other data (e.g., real and imaginary parts of normal velocity at surface nodes). See also *FREQUENCY_DOMAIN_SSD_ERP.

D3FTG Binary output file for random vibration fatigue analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE.

D3PSD Binary Power Spectral Density output file for random vibration analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

D3RMS Binary Root Mean Square output file for random vibration analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

D3SPCM Binary output file for response spectrum analysis. See also *FREQUENCY_DOMAIN_RESPONSE_SPECTRUM.

D3SSD Binary output file for steady-state dynamics. See also *FREQUENCY_DOMAIN_SSD.

D3ZCF Binary Zero-Crossing Frequency output file for random vibration analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

OPTION2 takes one of the following options:

<BLANK>

SUMMATION

SUBCASE

SUMMATION only works with *OPTION1* set to either D3PSD or D3RMS. With SUMMATION, the existing PSD and RMS databases from multiple random vibration analyses will be summed up to provide the total response of the same model subjected to multiple loading sources.

SUBCASE only works with D3SSD. With SUBCASE, an individual set of output frequencies is specified for each loading case included in *FREQUENCY_DOMAIN_SSD_SUBCASE.

The D3ACC, D3ACS, D3ATV, D3ERP, D3FTG, D3PSD, D3RMS, D3SPCM, D3SSD, and D3ZCF files contain plotting information to plot data over the three-dimensional geometry of the model. LS-PrePost can plot data from these databases.

- The D3ACC file contains each boundary element's acoustic pressure contribution (and contribution percentage) for a range of frequencies defined in *FREQUENCY_DOMAIN_ACOUSTIC_BEM.
- The D3ACS file contains acoustic results, including acoustic pressure and sound pressure level, for a range of frequencies defined in *FREQUENCY_DOMAIN_ACOUSTIC_FEM. Alternatively, it can be used to plot acoustic pressure, sound pressure level, and normal velocity on the surface of an acoustic volume for a range of frequencies defined in *FREQUENCY_DOMAIN_ACOUSTIC_BEM. Please note that only collocation BEM (METHOD = 3 or 4 in *FREQUENCY_DOMAIN_ACOUSTIC_BEM) can output the acoustic pressure, sound pressure level, and normal velocity on the surface of the acoustic volume to D3ACS.
- The D3ATV file contains NFIELD × NFREQ states, where NFIELD is the number of acoustic field points, and NFREQ is the number of output frequencies.
- The D3ERP file contains state data for a range of frequencies. These frequencies are the same as those used in D3SSD output.
- The D3FTG, D3RMS, and D3ZCF files contain only one state each as they are the data for cumulative fatigue damage ratio, root mean square of response for random vibration, peak response for response spectrum analysis, and zero-crossing frequency of response for random vibration separately.
- The D3PSD file contains PSD state data for a range of frequencies.
- When BINARY = 1, the D3SPCM file contains only one state: peak response for response spectrum analysis. When BINARY = 2, D3SPCM contains multiple states. The first state is peak response, like for BINARY = 1. The other states are the individual mode response sequentially for the normal modes involved in the mode combination.
- The D3SSD file contains state data for a range of frequencies. The data can be real or complex, depending on the variable BINARY defined below.

Card Summary:

Card 1a. Include this card if *OPTION2* is not used or *OPTION2* is SUBCASE.

BINARY				PSETID			

Card 1b. Include this card if *OPTION2* is SUMMATION. This card can be repeated if multiple binary databases exist.

FILENAME

Card 2a. Include this card if *OPTION1* is set to D3ACC. Include as many of this card as needed. Up to 10 NIDs are allowed.

NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8

Card 2b. Include this card if *OPTION1* is set to D3PSD or D3SSD. For D3SSD, when *OPTION2* is set to SUBCASE, this card can be repeated to define the output frequencies for each of the loading cases included in *FREQUENCY_DOMAIN_SSD_SUBCASE (in the same order).

FMIN	FMAX	NFREQ	FSPACE	LCFREQ			

Card 2c. Include this card if *OPTION1* is set to D3SPCM or D3RMS and BINARY = 3.

ISTATE	FILENAME

Data Card Definitions:

Include this card if *OPTION2* is unset or *OPTION2* is set to SUBCASE.

Card 1a	1	2	3	4	5	6	7	8
Variable	BINARY	SF			PSETID			
Type	I	F			I			
Default	0	3.0			0			

VARIABLE	DESCRIPTION
BINARY	Flag for writing the binary plot file. See Remark 1. EQ.0: Off

VARIABLE	DESCRIPTION
	EQ.1: Write the binary plot file. EQ.2: Write the complex variable binary plot file D3SSD (*OPTION1* = D3SSD) or include the individual mode response in the binary plot file D3SPCM (*OPTION1* = D3SPCM). EQ.3: Write the binary plot file, which combines response spectrum analysis or RMS results, depending on the choice of *OPTION1*, with other structural analysis results provided by the file specified with Card 2c It is available for *OPTION1* = D3SPCM or D3RMS. See Remarks 4 and 5. EQ.90: Write only the real part of the frequency response (D3SSD only). EQ.91: Write only the imaginary part of the frequency response (D3SSD only).
SF	Scale factor on the RMS response when using *OPTION1* = D3RMS with BINARY = 3. See Remark 5.
PSETID	Part set ID for the parts to be included in the database. If not defined, all the parts are included in the database. Currently it works for D3SSD only.

Summation Card. Card 1 when *OPTION1* = D3PSD or D3RMS and *OPTION2* = SUMMATION. This card can be repeated if multiple binary databases exist.

Card 1b	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

VARIABLE	DESCRIPTION
FILENAME	Path and file name of precomputed PSD or RMS binary databases (see Remark 3)

D3ACC Card. Additional card for D3ACC keyword option. Include as many of this card as needed. Up to 10 NIDs are allowed.

Card 2a	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

VARIABLE	DESCRIPTION
NID*i*	Field point node ID for writing D3ACC file

D3PSD and D3SSD Card. Additional card for D3PSD and D3SSD keyword options.

Card 2b	1	2	3	4	5	6	7	8
Variable	FMIN	FMAX	NFREQ	FSPACE	LCFREQ			
Type	F	F	I	I	I			
Default	0.0	0.0	0	0	0			

VARIABLE	DESCRIPTION
FMIN	Minimum frequency for output (cycles/time). See Remark 2.
FMAX	Maximum frequency for output (cycles/time). See Remark 2.
NFREQ	Number of frequencies for output. See Remark 2.
FSPACE	Frequency spacing option for output (See Remark 2): EQ.0: Linear EQ.1: Logarithmic EQ.2: Biased EQ.3: Eigenfrequencies only
LCFREQ	Load curve ID defining the frequencies for output. See Remark 2.

D3SPCM and D3RMS Card. Additional card for BINARY = 3.

Card 2c	1	2	3	4	5	6	7	8
Variable	ISTATE	FILENAME						
Type	I	C						

VARIABLE	DESCRIPTION
ISTATE	State number in a binary plot file with the name FILENAME. The structural analysis results at this state are combined with the results from the current run.
FILENAME	Path and file name of precomputed structural response binary plot file from which the structural response at a specific state will be combined with the current run's spectrum or RMS results (see Remarks 4 and 5)

Remarks:

1. **Binary data written for D3SSD.** For *OPTION1* = D3SSD, for BINARY = 1, the binary database d3ssd only contains the magnitude of the displacement, velocity, acceleration, and stress response. LS-PrePost 3.0 or older versions can access this data. For customers using LS-PrePost 3.0 or older versions, we suggest setting BINARY = 1. For BINARY = 2, d3ssd includes both the response's magnitude and phase angle so that LS-PrePost (3.1 or higher versions) can run modal expansion (to show the cyclic time history fringe plot) on each output frequency. If BINARY = 90 or 91, only the real or imaginary part of the response is written into d3ssd.

2. **Output frequencies.** There are two methods to define the output frequencies.

   a) The first method is to define FMIN, FMAX, NFREQ, and FSPACE. FMIN and FMAX specify the frequency range of interest, and NFREQ specifies the number of frequencies at which results are required. FSPACE specifies the type of frequency spacing (linear, logarithmic, biased, or eigenfrequencies only). These frequency points for which results are required can be spaced equally along the frequency axis (on a linear or logarithmic scale). Or they can be biased toward the eigenfrequencies (the frequency points are placed closer together at eigenfrequencies in the frequency range) to obtain a detailed response close resonance frequencies. Or they can be just the

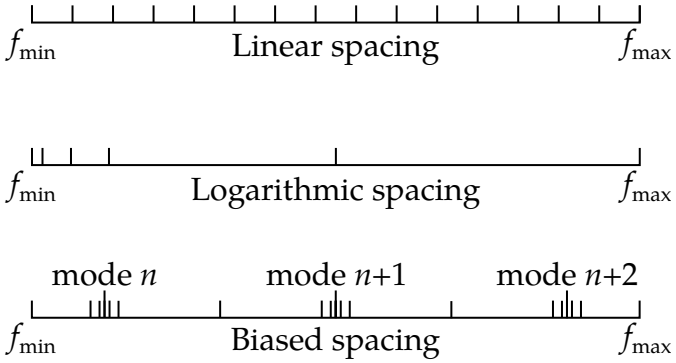

Figure 16-4. Spacing options of the frequency points.

eigenfrequencies in the frequency range defined by FMIN and FMAX. See Figure 16-4.

b) The second method uses a load curve (LCFREQ) to define the frequencies of interest.

3. **Summation for D3PSD and D3RMS.** For *OPTION1* = D3PSD and D3RMS, *OPTION2* = SUMMATION sums up and dumps to a new binary plot database d3psd and d3rms all the precomputed PSD and RMS results saved in existing binary databases defined in Card 1 (by default, d3psd and d3rms are under a different path or prefix). For d3psd, the new psd results are simply the summation of the psd results from each d3psd database (please note that the output frequencies of the existing d3psd databases must be the same, and the new d3psd provides psd results at the same set of frequencies). For d3rms, the new rms results are computed as the square root of the summation of squares of the rms values from each d3rms database. This provides the total PSD and RMS response for the same model, subjected to multiple loading resources. RESTRT = 2 in *FREQUENCY_DOMAIN_RANDOM_VIBRATION activates this operation.

4. **Combined results for response spectrum analysis with other structural analyses.** For *OPTION1* = D3SPCM, if BINARY = 3 in Card 1a, LS-DYNA combines the response spectrum analysis results with other structural analysis results. These other results come from a file specified in Card 2c and correspond to a specific state in that file. This feature allows obtaining the maximum value of the total response from multiple loadings.

d3spcm includes four states when BINARY = 3:

- u
- $r + u$
- $r - u$

- $|r| + |u|$

where u is the response spectrum analysis result and r is the other structural analysis result.

5. **Combined RMS results with other structural analyses.** For *OPTION1* = D3RMS with BINARY = 3 on Card 1a causes outputting the combination of RMS results with prestress from other structural analyses. These other results come from a file specified in Card 2c and correspond to a specific state in that file. This feature enables using stress from a prestressed random vibration in a failure analysis.

   d3rms includes three states when BINARY = 3:

   - u
   - $r + \mathrm{SF} \times u$
   - $|r| + \mathrm{SF} \times u$

   where u is the RMS response, r is the prestress from the structural analyses, and SF is the scale factor input on Card 1a.

***DATABASE_FSI** *{OPTION}*

Available options include:

<BLANK>

DR

Purpose: When a Lagrangian mesh overlaps with an Eulerian or ALE mesh, the fluid-structure (or ALE-Lagrangian) interaction is often modeled using the *CONSTRAINED_LAGRANGE_IN_SOLID or *ALE_STRUCTURED_FSI card. This keyword (*DATABASE_FSI) causes certain coupling information related to the flux through and load on selected Lagrangian surfaces defined in corresponding *CONSTRAINED_LAGRANGE_IN_SOLID or *ALE_STRUCTURED_FSI card to be written to the ASCII-based dbfsi file or in the case of MPP-DYNA the binout file.

A run involving *CONTROL_DYNAMIC_RELAXATION has two phases: a relaxation phase (to prestress a structure, for instance) and the actual run. The DR option enables saving the dbfsi data of the first phase if needed. If dbfsi is output as an ASCII file (BINARY = 1 or 3) during the first phase, the file is renamed dbfsi_dr before launching the second phase. If dbfsi is output in binout, the second phase data are concatenated to the first phase ones in binout.

NOTE: If used with *CONSTRAINED_LAGRANGE_IN_SOLID, penalty method coupling must be used (CTYPE = 4 or 5). This card is *not* compatible with constraint-based coupling.

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT	BINARY						
Type	F	I						

Surface Card. Add one card per surface. This input terminates at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	DBFSI_ID	SID	SIDTYPE	SWID	CONVID	NDSETID	CID	
Type	I	I	I	I	I	I	I	

VARIABLE	DESCRIPTION
DTOUT	Output interval time step
BINARY	Flag for binary output. See remarks under "Output Files and Post-Processing" in Appendix O, "LS-DYNA MPP User Guide." EQ.1: ASCII file is written. This is the default for shared memory parallel (SMP) LS-DYNA executables. EQ.2: Data written to binary database binout, 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 for MPP LS-DYNA executables. EQ.3: ASCII file is written, and the data is also written to the binary database. NOTE: MPP LS-DYNA executables only produce the binary database.
DBFSI_ID	Surface ID (for reference purposes only) or a *DATABASE_FSI entity ID. It consists of a geometric entity defined by the SID below.
SID	Set ID defining the geometrical surface(s) through which or upon which some data is to be tracked and output to a 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 structure 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
SWID	This is an ID from a corresponding *ALE_FSI_SWITCH_MMG_ID card. This card allows for the AMMG ID of an ALE material to be

VARIABLE	DESCRIPTION
	switched as it passes across a monitoring surface. If defined, the accumulative mass of the "switched" ALE multi-material group (AMMG) is written out under the "mout" parameter in the "dbfsi" file.
CONVID	This is used mostly for airbag application only. CONVID is an ID from a corresponding *LOAD_ALE_CONVECTION_ID card which computes the heat transfer between inflator gas (ALE material) and the inflator canister (Lagrangian part). If defined, the temperature of the Lagrangian part having heat transfer with the gas, and its change in temperature as function of time are output in the "dbfsi" file.
NDSETID	Set ID consisting of the nodes on which the moments of the forces applied on SID are computed. See Remark 3.
CID	Coordinate system ID, see *DEFINE_COORDINATE_SYSTEM.

Remarks:

1. **Overview of dbfsi File.** The dbfsi parameters output are enumerated below.

pres = Averaged estimated coupling pressure over each surface entity being monitored. For example, if using SI base units for mass-length-time-temperature, this pressure would then be in Pascal.

fx, fy, fz = Averaged total estimated coupling force components (N in metric units) along the global coordinate directions, over each surface entity defined, and acting at the centroid of each surface.

mout = Accumulated mass (kg in metric units) passing through each DBFS_ID surface entity. See Remark 2 below. (This parameter used to be called "pleak").

obsolete = (This parameter used to be called "mflux").

gx, gy, gz = Average estimated leakage-control force component over the surface entity. This data is useful for debugging. Leakage control forces are too large (relative to the main coupling forces, fx, fy and fz) may indicate that alternate coupling approach should be considered since the main coupling force is putting out too little resistance to leakage. (These parameters used to be called fx-lc, fy-lc and fz-lc).

Ptmp = Lagrangian part Temperature (Activated only when the *LOAD_ALE_CONVECTION card is used).

PDt = Lagrangian part Temperature change (Activated only when the *LOAD_ALE_CONVECTION card is used).

2. **MOUT.** "mout" parameter in the "dbfsi" output from this keyword contains the accumulated mass passing through each DBFS_ID surface entity. For 4 different cases:

   a) When LCIDPOR is defined in the coupling card (CLIS), porous accumulated mass transport across a Lagrangian shell surface may be monitored and output in "mout".

   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. This is an alternate form of (a).

   c) When NVENT in the CLIS card is defined (isentropic venting), the venting mass transport across the isentropic vent hole surface may be output in "mout".

   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 passing across a monitoring surface is output.

3. **Calculation of Moments for NDSETID.** A geometrical surface SID has a centroid where the coupling forces are averaged. The distances between this centroid and the nodes defined by the set NDSETID are the lever arms. The moments are the cross-products of these distances with the averaged coupling forces. For each node in the set NDSETID, a new line in the "dbfsi" file is inserted after each output for the corresponding coupling forces (see Remark 1). These additional lines have the format following the template established by the example in Remark 1 where the forces are replaced by the moments, and the node ID replaces the DBFSI_ID values.

4. **2D axisymmetric models.** The forces from a 2D axisymmetric simulation are reported per radian, regardless of the element formulation used. Multiply the 2D force by 2π to get the 3D force.

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          pres            fx            fy            fz          mout
                obsolete            gx            gy            gz          Ptmp
PDt
         time=  0.00000E+00
        11    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
              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    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    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    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    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    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*DATABASE_FSI_SENSOR

Purpose: Activate the output of an ASCII file called dbsensor. This keyword's input defines the pressure sensors' locations which follow the positions of some Lagrangian segments during the simulation. The 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 or *ALE_STRUCTURED_FSI card is used and the Lagrangian shell elements defining the locations of the sensors must be included in the structure coupling set.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY						
Type	F	I						

Surface Card. Add one card per surface. This input terminates at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	DBFSI_ID	NID	SEGMID	OFFSET	ND1	ND2	ND3	
Type	I	I	I	F	I	I	I	

VARIABLE	DESCRIPTION
DT	Output interval
BINARY	Flag for binary output. See "Output Files and Post-Processing" in Appendix O, "LS-DYNA MPP User Guide."

EQ.1: ASCII file is written. This is the default for shared memory parallel (SMP) LS-DYNA executables.

EQ.2: Data written to binary database binout, 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 for MPP LS-DYNA executables.

EQ.3: ASCII file is written, and the data is also written to the binary database (NOTE: MPP LS-DYNA executables will only produce the binary database).

VARIABLE	DESCRIPTION
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 element ID for locating the pressure sensor. If NID = 0 or is blank, the sensor will be automatically placed in the center of this SEGMID, accounting for the offset distance. If the model is three-dimensional, the Lagrangian element can be a shell or solid (for the solid element, ND1 and ND2 are required to define the face). If the model is two-dimensional, the Lagrangian element can be a beam or shell (for the shell element, ND1 and ND2 are required to define the side).
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; otherwise, vice versa.
ND1, ND2, ND3	Nodes defining the solid face for the solid element in the three-dimensional model or shell side for the shell element in the two-dimensional model, from which the sensor is located. In three dimensions, if the solid face has 4 nodes, only the diagonal opposites ND1 and ND2 are required. If the solid face is triangular, a third node ND3 should be provided. In two dimensions, only ND1 and ND2 are required to define the shell side.

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.

   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

DES (see Remark 4)

DES_SET

DISCRETE

DISCRETE_ID

DISCRETE_SET

NODE

NODE_ID

NODE_LOCAL

NODE_LOCAL_ID

NODE_SET

NODE_SET_LOCAL

SEATBELT (see Remark 2)

SEATBELT_ID

SEATBELT_RETRACTOR

SEATBELT_RETRACTOR_ID (see Remark 3)

SEATBELTSLIPRING

SEATBELT_SLIPRING_ID (see Remark 3)

SHELL

SHELL_ID

SHELL_SET

SOLID

SOLID_ID

SOLID_SET

SPH

SPH_SET

TSHELL

TSHELL_ID

TSHELL_SET

Purpose: Control which nodes, elements, or bonds are output into the binary history file d3thdt, the ASCII file nodout, the ASCII file elout, the ASCII file sphout, and the ASCII file debond. See also *DATABASE_BINARY_*OPTION* and *DATABASE_*OPTION*.

Card Summary:

Card 1a. Include this card if the keyword option is BEAM, BEAM_SET, DES, DES_SET, DISCRETE, DISCRETE_SET, NODE, NODE_SET, SEATBELT, SEATBELT_RETRACTOR, SEATBELT_SLIPRING, SHELL, SHELL_SET, SOLID, SOLID_SET, SPH, SPH_SET, TSHELL, or TSHELL_SET. Include as many as needed. Input terminates at the next keyword ("*") card.

ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8

Card 1b. Include this card if the keyword option is BEAM_ID, NODE_ID, SEATBELT_ID, SEATBELT_RETRACTOR_ID, SEATBELT_SLIPRING_ID, SHELL_ID, SOLID_ID, and TSHELL_ID. Include as many as needed. Input terminates at the next keyword ("*") card.

ID	HEADING

Card 1c. Include this card if the keyword option is NODE_LOCAL, NODE_LOCAL_ID, and NODE_SET_LOCAL. If the keyword option is NODE_LOCAL_ID, see Card 1c.1 as well. Include as many cards as needed to specify all the nodes. This input terminates at the next keyword ("*") card.

ID	CID	REF	HFO				

Card 1c.1. This card is only used for the NODE_LOCAL_ID keyword option. When activated, each node is specified by a pair of cards consisting of Card 1c, and, secondly, this card. Include as many pairs as needed to specify all the nodes. This input terminates at the next keyword ("*") card.

HEADING	

Data Card Definitions:

Node/Element Cards for Case I (no "ID", and no "LOCAL"). Cards for keyword options BEAM, BEAM_SET, DES, DES_SET, DISCRETE, DISCRETE_SET, NODE, NODE_SET, SEATBELT, SEATBELT_RETRACTOR, SEATBELT_SLIPRING, SHELL, SHELL_SET, SOLID, SOLID_SET, SPH, SPH_SET, TSHELL, and TSHELL_SET. Include as many as needed. Input terminates at the next keyword ("*") card.

Card 1a	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*	NODE/NODE_SET or element/element set ID *n*. Elements may be BEAM/BEAM_SET, DES/DES_SET, DISCRETE/DISCRETE_SET, SEATBELT, SEATBELT_RETRACTOR, SEATBELT_SLIPRING, SHELL/SHELL_SET, SOLID/SOLID_SET, SPH/SPH_SET or TSHELL/TSHELL_SET. ID*n* for NODE_SET, SPH_SET, and DES_SET refers to node set ID *n* defined using the *SET_NODE_*{OPTION}*. The contents of the files are given in Table 16-1 for nodes, Table 16-2 for solid elements, Table 16-3 for shells and thick shells, and Table 16-4 for beam elements. The contents of binary file d3thdt may be extended or reduced with *DATABASE_EXTENT_BINARY.

Node/Element Cards for Case II ("ID" option, but no "LOCAL"). Cards for keyword options BEAM_ID, NODE_ID, SEATBELT_ID, SEATBELT_RETRACTOR_ID, SEATBELT_SLIPRING_ID, SHELL_ID, SOLID_ID, and TSHELL_ID. Include as many as needed. Input terminates at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

VARIABLE	DESCRIPTION
ID	Node or element ID
HEADING	A description of the node or element. Using a unique description is recommended. This description is written into the d3hsp file and into the ASCII databases nodout and elout.

Node Cards for Case III ("LOCAL" option). Card 1 for keyword options NODE_LOCAL, NODE_LOCAL_ID, and NODE_SET_LOCAL. Include as many cards as needed to specify all the nodes. This input terminates at the next keyword ("*") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	ID	CID	REF	HFO				
Type	I	I	I	I				

ID Card for Case III. Additional card for ID option. This card is only used for the NODE_LOCAL_ID keyword option. When activated, each node is specified by a pair of cards consisting of Card 1c and, secondly, this card. Include as many pairs as needed to specify all the nodes. This input terminates at the next keyword ("*") card.

Card 1c.1	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	A70							

VARIABLE	DESCRIPTION
ID	NODE/NODE_SET set ID. The contents of the files are given in Table 16-1 for nodes. See Remark 1 concerning accelerometer nodes.
CID	Coordinate system ID for nodal output. See *DEFINE_COORDINATE options.
REF	Output coordinate system for displacements, velocities, and accelerations. (Nodal coordinates are always in the global coordinate system.) EQ.0: Output is in the local system fixed for all time from the beginning of the calculation. If CID is nonzero, FLAG in the corresponding *DEFINE_COORDINATE_NODES command must be set to 0. FLAG has no bearing on results when REF is set to 1 or 2. EQ.1: Translational output is the projection of the node's absolute translational motion onto the local system. The local system is defined by the *DEFINE_COORDINATE_NODES command and can change orientation according to the movement of the three defining nodes. The defining nodes can belong to either deformable or rigid parts. EQ.2: Translational output is the motion of the node, expressed in the local system attached to node N1 of CID. The local system is defined as described in REF = 1 above. For displacements, the fixed reference location is defined to contain the initial coordinates of the history node in the local coordinate system. During the analysis, the coordinates of the history node are first expressed in the translating and rotating local coordinate system and then the local displacement vector is determined by subtracting the reference location from these coordinates. If dynamic relaxation is used, the reference location is reset when convergence is achieved. Rotational output is truly relative to the updated location coordinate system only if REF = 2.
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

VARIABLE	DESCRIPTION
	description be used. This description is written into the d3hsp file and into the ASCII database nodout.

Remarks:

1. **Accelerometer.** If a node belongs to an accelerometer (see *ELEMENT_SEATBELT_ACCELEROMETER) and also appears as an active node in the NODE_LOCAL or NODE_SET_LOCAL keyword, the coordinate system transformations are skipped, and the LOCAL option has no effect.

2. **Seat belt.** The SEATBELT keyword option is only for 1D (bar-type) seat belts, not 2D (shell-type).

3. **Legend for slip ring/retractor data.** To define legends for slip ring and retractor data, append ID to the SEATBELT_SLIPRING and SEATBELT_RETRACTOR keyword options. The HEADING field gives the legend name.

4. **DES.** By default, the debond ASCII file (see *DATABASE_DEBOND) monitors all DES bonds for breakage. Defining *DATABASE_HISTORY_DES_*{OPTION}* causes debond to only monitor the bonds associated with the node IDs defined by ID*n* . For example, if a single node appears in *DATABASE_HISTORY_DES, all bonds that bind the discrete sphere defined by that node will be monitored for breakage type.

*DATABASE_HISTORY_ACOUSTIC

Purpose: Identify acoustic nodes for time history output.

Card 1	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

VARIABLE	DESCRIPTION
NID*n*	Node ID of the acoustic node.

Remarks:

1. **Results File Name.** In explicit, transient solutions, the histories are written to file ACEOUT. In implicit, direct SSD solutions, the histories are written to ACEOUT_SSD.

*DATABASE_ISPHHTC

Purpose: Create Heat Transfer Coefficient output for Incompressible SPH simulations. HTC values are time-averaged over the defined output time interval.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY	LCDT	IOOPT	PSET			
Type	F	I	I	I	I			

VARIABLE	DESCRIPTION
DT	Time interval between output states. If DT is zero, the output will be skipped even if LCDT is defined.
BINARY	Currently unused. The output format is currently a profile file that can be directly imported inside Ansys Fluent or Ansys Mechanical.
LCDT	Optional load curve ID specifying the output time interval as a function of time.
IOOPT	This input field governs how the plot state frequency is determined from curve LCDT: EQ.1: When a plot is generated at time t_n, the next plot time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCDT}(t_n)$. This is the default behavior. EQ.2: When a plot is generated at time t_n, the next plot time t_{n+1} is computed as $t_{n+1} = t_n + \mathrm{LCDT}(t_{n+1})$. EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.
PSET	Optional part set. If defined, only parts contained in this part set will be included in the HTC output file.

*DATABASE_MASSOUT

Purpose: Output nodal masses into ASCII file MASSOUT.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	NDFLG	RBFLG					
Type	I	I	I					
Default	0	1	0					

VARIABLE	DESCRIPTION
SETID	Optional set ID. EQ.0: mass output for all nodes, LT.0: no output, GT.0: set ID identifying nodes whose mass will be output.
NDFLG	Database extent: EQ.1: output translational mass for deformable nodes identified by SETID (default), EQ.2: output translational mass and rotary inertias for the deformable nodes identified by the SETID. EQ.3: output translational mass for deformable and rigid nodes identified by SETID (default), EQ.4: output translational mass and rotary inertias for the deformable and rigid nodes identified by the SETID.
RBFLG	Rigid body data: EQ.0: no output for rigid bodies, EQ.1: output rigid body mass and inertia.

Remarks:

1. **Massless Nodes and Rigid Bodies.** Nodes and rigid bodies with no mass are not output. By inference, when the set ID is zero and no output shows up for a node, then the mass of that node is zero.

DATABASE_MAX_*OPTION

Available options include:

BEAM

BEAM_ID

BEAM_SET

SHELL

SHELL_ID

SHELL_SET

SOLID

SOLID_ID

SOLID_SET

TSHELL

TSHELL_ID

TSHELL_SET

Purpose: Control which elements have data output to the binary history file d3max. See also *DATABASE_D3MAX.

Card Summary:

Card 1a. This card is included if the keyword option is BEAM, BEAM_SET, SHELL, SHELL_SET, SOLID, SOLID_SET, TSHELL, or TSHELL_SET. Include as many as needed. Input terminates at the next keyword ("*") card.

ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8

Card 1b. This card is included if the keyword option is BEAM_ID, SHELL_ID, SOLID_-ID, and TSHELL_ID. Include as many as needed. Input terminates at the next keyword ("*") card.

ID	HEADING

Data Card Definitions:

Element Cards for Case I (no "ID"). Cards for keyword options BEAM, BEAM_SET, SHELL, SHELL_SET, SOLID, SOLID_SET, TSHELL, and TSHELL_SET. Include as many as needed. Input terminates at the next keyword ("*") card.

Card 1a	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*	Element/element set ID *n*. The elements may be beams, shells, solids, or tshells depending on the keyword option.

Element Cards for Case II ("ID" option"). Cards for keyword options BEAM_ID, SHELL_ID, SOLID_ID, and TSHELL_ID. Include as many as needed. Input terminates at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

VARIABLE	DESCRIPTION
ID	Element ID
HEADING	A description of the element. We suggest using unique descriptions.

*DATABASE_NODAL_FORCE_GROUP

Purpose: Define a nodal force group for output into the ASCII file nodfor. The output interval must be specified using *DATABASE_NODFOR (see *DATABASE_*OPTION*).

Card 1	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
CID	Coordinate system ID for output of data in local system

Remarks:

1. The reaction forces in the global x, y, and z directions (and local x, y, and z directions if CID is defined above) for the nodal force group are written to the nodfor file (see *DATABASE_NODFOR) along with the external work done by these reaction forces. The reaction forces in the global x, y, and z directions for each node in the nodal force group are also written to nodfor. 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_PAP_OUTPUT

Purpose: Set contents of output files for pore air pressure calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	IVEL	IACCX	IACCY	IACCZ	NCYOUT			
Type	I	I	I	I	I			
Default	0	0	0	0	100			

VARIABLE	DESCRIPTION
IVEL	Meaning of "Velocity" in d3plot and d3thdt output files EQ.0: Nodal velocity vector EQ.1: Seepage velocity vector
IACCX, Y, Z	Meaning of "X/Y/Z-Acceleration" in d3plot and d3thdt output files EQ.0: Not written EQ.21: Nodal air density EQ.22: Nodal pore air pressure EQ.24: Nodal air mass EQ.25: Nodal air mass flow rate
NCYOUT	Number of cycles between outputs of calculation status to d3hsp and log files

*DATABASE_PBLAST_SENSOR

Purpose: This keyword causes LS-DYNA to output a directory called pblast_sensor to the file, binout. The input for this keyword defines the sensors' locations based on the positions of some Lagrangian shell elements. With this keyword, LS-DYNA outputs the history of the position, temperature, density, and pressure averaged over the number of particles contained in the sensors to binout. This keyword is activated only when the *DEFINE_PARTICLE_BLAST card is used.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY						
Type	F	I						
Default	none	3						

Sensor Definition Cards. Each card defines one sensor. This card may be repeated to define multiple sensors. Input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	ID	ITYPE	OFFSET	RADIUS				
Type	I	I	F	F	F			

VARIABLE	DESCRIPTION
DT	Output interval
BINARY	Flag for the binary file: EQ.3: Data is written to the binary file binout.
ID	Set ID
ITYPE	Type for ID: EQ.0: *SET_SHELL ID EQ.1: Shell ID

VARIABLE	DESCRIPTION
OFFSET	Offset distance, d, between the sensor and the segment center. Here $d > 0$ is along the shell normal and $d < 0$ is against the shell normal.
RADIUS	Radius of the spherical sensor. Please see *DATABASE_CPM_-SENSOR for more details about spherical sensors.

Remarks:

1. **Output.** The output parameters to the pblast_sensor directory are:

x	=	x-coordinate
y	=	y-coordinate
z	=	z-coordinate
temp	=	temperature
dens	=	density
pres	=	pressure
counts	=	number of particles in the sensor

These values are averaged over the number of particles in the sensor. The sensor should be large enough to contain a reasonable number of particles for the averages.

*DATABASE_PROFILE

Purpose: Plot the distribution or profile of a data along x, y, or z-direction.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	ID	TYPE	DATA	DIR	UPDLOC	MMG	
Type	I	I	I	I	I	I	I	
Default	none	none	none	none	none	0	0	

VARIABLE	DESCRIPTION
DT	Interval time.
ID	Set ID.
TYPE	Set type: EQ.1: node set EQ.2: solid set EQ.3: shell set EQ.4: segment set EQ.5: beam set EQ.6: tshell set
DATA	Data type: EQ.1: x-velocity, EQ.2: y-velocity, EQ.3: z-velocity, EQ.4: velocity magnitude, EQ.5: x-acceleration, EQ.6: y-acceleration, EQ.7: z-acceleration, EQ.8: acceleration magnitude, EQ.9: pressure,

VARIABLE	DESCRIPTION
	EQ.10: *xx*-stress, EQ.11: *yy*-stress, EQ.12: *zz*-stress, EQ.13: *xy*-stress, EQ.14: *yz*-stress, EQ.15: *zx*-stress, EQ.16: temperature, EQ.17: volume fraction, EQ.18: kinetic energy, EQ.19: internal energy, EQ.20: density, EQ.21: *xx*-strain, EQ.22: *yy*-strain, EQ.23: *zz*-strain, EQ.24: *xy*-strain, EQ.25: *yz*-strain, EQ.26: *zx*-strain. EQ.27: effective plastic strain
DIR	Direction: EQ.1: *x*-direction EQ.2: *y*-direction EQ.3: *z*-direction EQ.4: curvilinear (relative distances between elements of set ID are added up in the order defined by the set)
UPDLOC	Flag to update the set location: EQ.0: only the initial position of set ID is considered. EQ.1: the positions of the elements composing the set are updated each DT.
MMG	Multi-Material ALE group ID. See Remark 2. GT.0: Multi-Material ALE group ID

VARIABLE	DESCRIPTION
LT.0:	|MMG| is the ID of a *SET_MULTI-MATERIAL_GROUP_-LIST that can list several Multi-Material ALE group IDs.

Remarks:

1. **File Description.** At a given time T the profile is written in a file named profile_DATA_DIR_time$T$.xy (DATA and DIR are replaced by the data and direction names respectively). The file has a xyplot format that LS-PrePost can read and plot. For example, DATA = 9, DIR = 2, and DT = 0.1 sec will save a pressure profile at $t = 0.0$ sec in profile_pressure_y_time0.0.xy, at $t = 0.1$ sec in profile_pressure_y_time0.1.xy, at $t = 0.2$ sec in profile_pressure_y_time0.2.xy. The stresses are output for each integration point. The shell strains are output for the innermost and outermost integration points if STRFLG = 1 in *DATABASE_-EXTENT_BINARY. For stresses and shell strains, the integration point rank is appended to the data names in the filename.

2. **Multi-Material ALE Model.** For the case of a multi-material ALE model (elform = 11 in *SECTION_SOLID or *SECTION_ALE2D or *SECTION_-ALE1D), an element can contain several materials with each material being associated with its own pressures and stresses. By default element data is volume averaged before being written out; however, when the multi-material group field, MMG, is set, then element data are output only for the specified materials.

*DATABASE_PWP_FLOW

Purpose: Request output containing the net inflow of fluid at a set of nodes.

Node Sets. Include as many cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NSET							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
NSET	Node set ID

Remarks:

1. **Inflow.** Net inflow or outflow occurs when maintaining an applied PWP boundary condition causes the addition or removal of water. See *BOUNDARY_PWP.

2. **Output file.** Output is written to a file named database_pwp_flow.csv, a comma-separated ASCII file. Each line consists of (time, flow1, flow2, …) where flow1 is the total inflow at the node set for the first *DATABASE_PWP_FLOW request, flow2 is for the second, etc.

3. **Output time interval.** For versions up to and including R15, the output is written at the time interval DT on *DATABASE_BINARY_D3THDT. If this card is absent, no output is written.

*DATABASE_PWP_OUTPUT

Purpose: Set contents of output files for pore pressure calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	IVEL	IACCX	IACCY	IACCZ	NCYOUT			
Type	I	I	I	I	I			
Default	0	0	0	0	100			

VARIABLE	DESCRIPTION
IVEL	Meaning of nodal "Velocity" in d3plot: EQ.0: Nodal velocity vector EQ.1: Seepage velocity vector
IACCX, Y, Z	Meaning of nodal "X/Y/Z-Acceleration" in d3plot, d3thdt, and nodout: EQ.0: Not written EQ.1: Total pwp head EQ.2: Excess pwp head (this is also written as temperature) EQ.3: Target rate of volume change EQ.4: Actual rate of volume change EQ.5: Part whose pore pressure properties are used for this node (internal ID) EQ.7: Hydraulic pwp head EQ.8: Error in rate of volume change (calculated from seepage minus actual) EQ.9: Volume at node EQ.10: Rate of volume change calculated from seepage EQ.14: Void volume (generated at suction limit) EQ.17: Status flag (e.g: +4/-4 for nodes on suction limit)

VARIABLE	DESCRIPTION
NCYOUT	Number of cycles between outputs of calculation status to d3hsp, log, and tdc_control_output.csv files (time-dependent and steady-state analysis types).

*DATABASE_RCFORC_MOMENT

Purpose: Define contact ID and nodes for moment calculations. Moments are written to rcforc according to output interval given in *DATABASE_RCFORC. If *DATABASE_RCFORC_MOMENT is not used, the moments reported to rcforc are about the origin (0,0,0).

Card 1	1	2	3	4	5	6	7	8
Variable	CID	NODESA	NODESB					
Type	I	I	I					

VARIABLE	DESCRIPTION
CID	Contact ID
NODESA	Node about which moments are calculated due to contact forces on SURFA surface
NODESB	Node about which moments are calculated due to contact forces on SURFB surface

*DATABASE_RECOVER_NODE

Purpose: Recovers the stresses at nodal points of solid or thin shell elements by using either Zienkiewicz-Zhu's Superconvergent Patch Recovery (SPR) method or an elemental extrapolation method.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	IAX	IAY	IAZ	METHOD	IVX	IVY	IVZ
Type	I	A	A	A	I	A	A	A
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
PSID	Part set ID of solid or thin shell elements whose nodal stress will be recovered
IAX, IAY, IAZ, IVX, IVY, IVZ	Meaning of "$x/y/z$-Acceleration" or "$x/y/z$-Velocity" in d3plot and d3thdt output files EQ.SMNPD: the minimum principal deviator stress EQ.SMNPR: the minimum principal stress EQ.SMXPD: the maximum principal deviator stress EQ.SMXPR: the maximum principal stress EQ.SMXSH: the maximum shear stress EQ.SPR: nodal pressure EQ.SVM: nodal von Mises stress EQ.SXX: nodal normal stress along x direction EQ.SYY: nodal normal stress along y direction EQ.SZZ: nodal normal stress along z direction EQ.SXY: nodal shear stress along x-y direction EQ.SYZ: nodal shear stress along y-z direction EQ.SZX: nodal shear stress along z-x direction For shell elements append either "B" or "T" to the input string to recover nodal stresses at the bottom or top layer of shell elements. For example, SPRT recovers the nodal pressure at the top layer.

VARIABLE	DESCRIPTION
METHOD	Method used to recover the nodal stress: EQ.0: Zienkiewicz-Zhu's Superconvergent Patch Recovery method EQ.1: elemental extrapolation method

Remarks:

1. **Coordinate System.** Recovered stresses are in global coordinate system.
2. **SPR Method.** The SPR method is based on the extrapolation of an element patch, a group of elements surrounding a node. Therefore, it should be more accurate than the elemental extrapolation method which is based on the extrapolation of a single element. This is especially true for under-integrated elements where the integration points are the super convergent points. However, since SPR requires an element patch, it cannot recover the nodal stresses correctly when a mesh is too course to form a patch (for example, through the thickness direction of a beam, only a single layer of elements is used). Also, when used for shell elements, it can only provide a good solution in two-dimensional analysis or in small strain analysis where most elements remain on the same plane.
3. **Elemental Extrapolation Method.** The elemental extrapolation method is also used for detailed element out, eloutdet. It is more robust and provides acceptable solutions for elements of multiple integration points.

*DATABASE_RVE

Purpose: Output of the RVE homogenization results to the rveout file. Double precision SMP/MPP LS-DYNA version R13 and newer versions support this RVE analysis function.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY						
Type	F	I						
Default	0.0	0						

VARIABLE	DESCRIPTION
DT	Time interval for the output of RVE homogenization results to the rveout file.
BINARY	Type of the output file: EQ.0: ASCII database file named rveout

Remarks:

At each output time t, LS-DYNA records the macroscopic material responses to the rveout file. The responses include the macroscopic deformation gradient $\tilde{\mathbf{F}}^t$, Green strain $\tilde{\mathbf{E}}^t$, Cauchy stress $\tilde{\boldsymbol{\tau}}^t$, and the 1st-Piola-Kirchhoff (PK1) stress $\tilde{\mathbf{P}}^t$. These results are obtained through nonlinear computational homogenization of RVE (*Representative Volume Element*) for composite materials. Please refer to the keyword *RVE_ANALYSIS_FEM for more details.

*DATABASE_SALE

Purpose: Control how the S-ALE solver outputs the post-processing database and whether it outputs the generated S-ALE mesh or not.

Card 1	1	2	3	4	5	6	7	8
Variable	D3SALE	DMPMSH						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
D3SALE	Flag for where to output the S-ALE part data (see Remarks 1 and 2): EQ.0: Output S-ALE part data to d3plot. EQ.1: Write S-ALE part data to binary, LSDA-format database d3sale.
DMPMSH	Flag for outputting the generated S-ALE mesh (see Remark 3): EQ.0: Do not output the S-ALE mesh. EQ.1: Output the S-ALE mesh into a file named salemsh.inc.

Remarks:

1. **S-ALE data output interval.** *DATABASE_BINARY_D3PLOT specifies the output interval of S-ALE data for either choice of D3SALE.

2. **D3SALE.** The d3sale file is an LSDA-formatted database for flexibility and file size reduction. When turned on, the output d3sale contains S-ALE post-processing data, while d3plot contains the data for Lagrange parts.

3. **Outputting S-ALE mesh.** The keyword input phase generates the S-ALE mesh (elements/nodes) based on information provided in *ALE_STRUCTURED_MESH and *ALE_STRUCTURED_MESH_CONTROL_POINTS. DMPMSH = 1 invokes outputting the generated mesh keywords into a file named salemsh.inc. The file can be used to validate the mesh in a pre-processor.

*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	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_SUPERPLASTIC_FORMING

Purpose: Specify the output intervals to the superplastic forming output files. The option *LOAD_SUPERPLASTIC_FORMING must be active.

Card 1	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 as a function of 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_{OPTION}

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, SPH, and DEM (Discrete Element Method) problems.

Available options are:

<BLANK>

DE

The DE option defines a tracer corresponding to discrete elements (*ELEMENT_DISCRETE_SPHERE). See Remarks 2 and 4.

Card	1	2	3	4	5	6	7	8
Variable	TIME	TRACK	X	Y	Z	AMMGID	NID	RADIUS
Type	F	I	F	F	F	I	I	F
Default	0.0	0	0.0	0.0	0.0	0	0	0.0

VARIABLE	DESCRIPTION
TIME	Start time for tracer particle
TRACK	Tracking option: EQ.0: Particle follows material (see Remark 5). EQ.1: Particle is fixed in space. EQ.2: Particle follows the mesh
X	Initial *x*-coordinate
Y	Initial *y*-coordinate
Z	Initial *z*-coordinate
AMMGID	The ALE multi-material referenced with its AMMGID for general ALE or with either its AMMGID or AMMG name (AMMGNM) for S-ALE, of the material being tracked in a multi-material ALE element. See Remark 1.

VARIABLE	DESCRIPTION
NID	An optional node ID defining the initial position of a tracer particle. If defined, its coordinates will overwrite the x, y, and z coordinates above. This feature is for TRACK = 0 only and can be applied to ALE tracers and DE tracers. See Remark 2.
RADIUS	Radius is used only for the DE option to indicate whether the tracer follows and monitors a single discrete element or multiple discrete elements. GT.0.0: The tracer takes the average results of all discrete elements located inside a sphere with radius = RADIUS. That sphere stays centered on the DE tracer. LT.0.0: The discrete element closest to the tracer is used. The magnitude of RADIUS, in this case, is unimportant.

Remarks:

1. **Multi-material groups.** ALE elements can contain multi-materials. Each material is referred to as an ALE multi-material group or AMMG. Each AMMG has its list of history variables that can be output. For example, if a tracer is in a mixed element consisting of 2 AMMGs, and the history variables of AMMG 1 are to be output or tracked, the AMMGID should be defined as AMMGID = 1. If AMMGID = 0, a volume-fraction-weighted-averaged pressure will be reported instead.

2. **NID description.** For ALE, NID is a massless dummy node. Its location will be updated according to the motion of the ALE material.

   For the DE option, NID is a discrete element node that defines the initial location of the tracer. The DE tracer continues to follow that node if RADIUS < 0. On the other hand, the DE tracer's location is updated according to the average motion of the group of DE nodes inside the sphere defined by RADIUS when RADIUS > 0.

3. **Tracer particles in ambient ALE elements.** Since the auxiliary variables (6 stresses, plastic strain, internal energy, …) for ambient elements are reset to their initial values before and after advection and tracer data are stored in trhist during the advection cycle, tracers in ambient elements show the initial stresses, not the current ones.

4. **Discrete elements.** If the DE keyword option is used, tracer particles will save a history of either a material point or a spatial point into an ASCII file: demtrh. This history includes positions, velocities components, stress components,

porosity, void ratio, and coordination number. *DATABASE_TRHIST must be active.

5. **Mapping.** Lagrange (TRACK = 0) tracers with mapping (map =) produce a keyword file tracers_final_positions.k, containing the final positions of the tracer particles. This file should be included in the mapped simulation to correctly locate the Lagrange tracers.

*DATABASE_TRACER_ALE

Purpose: Save a history of either a material point or a spatial point using tracer particles into an ASCII file: traleh. This history includes positions, velocities, and stress and other user-specified element history variables. *DATABASE_TRHIST must be active. This keyword is only to be used with ALE problems.

Card	1	2	3	4	5	6	7	8
Variable	NID	TRACK	AMMGID	HVBEG	HVEND	TIME		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0.0		

VARIABLE	DESCRIPTION
NID	Node ID defining the initial position of a tracer particle. See Remark 1.
TRACK	Tracking option: EQ.0: Particle follows material EQ.1: Particle is fixed in space
AMMGID	ALE multi-material referenced with its AMMGID for general ALE or with either its AMMGID or AMMG name (AMMGNM) for S-ALE, of the material being tracked in a multi-material ALE element. See Remark 2.
HVBEG	The beginning index of element history variables to be output. See Remark 3.
HVEND	The ending index of element history variables to be output. The number of extra history variables must be no more than 15, meaning HVEND -HVBEG = 15.
TIME	Start time for tracer particle activation

Remarks:

1. **NID description.** NID is a massless dummy node. Its location will be updated according to the motion of the ALE material.

2. **Multi-material groups.** ALE elements can contain multi-materials. Each material is referred to as an ALE multi-material group or AMMG. Each AMMG has its list of history variables that can be output. For example, if a tracer is in a mixed element consisting of 2 AMMGs, and the history variables of AMMG 1 are to be output or tracked, the AMMGID should be defined as AMMGID = 1. If AMMGID = 0, a volume-fraction-weighted-averaged pressure will be reported instead.

3. **History variables.** This keyword provides a way to output specific element history variables. By default, there are 9 element properties output for each particle: 6 stresses, plastic strain, volume fraction, and density. For example, if you want to know the air temperature of AMMG 2 which is material *MAT_148, you set AMMGID to 2 and HVBEG = HVEND = 16 since history variable 16 of *MAT_148 is temperature. At most, data for 15 history variables can be output this way.

4. **Tracer particles in ambient ALE elements.** Since the auxiliary variables (6 stresses, plastic strain, internal energy, …) for ambient elements are reset to their initial values before and after advection, and tracer data are stored in trhist during the advection cycle, tracers in ambient elements show the initial stresses, not the current ones.

Table 16-5. Variables output to traleh

Variable	Description
elemID	ID of ALE element in which the tracer is currently located
x, y, z	Position
vx, vy, vz	Velocity
sx, sy, sz, sxy, syz, szx	Stress
efp	Effective plastic strain
rho	Density
rvol	Volume fraction
hisvar*i*	History variable *i*

Example File:

Below is an example traleh file. 7 and 30 in the second line tells the reader that there are 7 tracers and each tracer has 30 output variables. The next six lines list the output variables as they are output in the file. These output variables are described in Table 16-5.

```
ALE Tracer particle file
    7   30
          elemID
```

```
          x           y           z          vx          vy          vz
         sx          sy          sz         sxy         syz         szx
        efp         rho        rvol       hisv1       hisv2       hisv3
      hisv4       hisv5       hisv6       hisv7       hisv8       hisv9
     hisv10      hisv11      hisv12      hisv13      hisv14      hisv15
 0.00000E+00
       5002767
 5.01000E+02  1.00100E+03  1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
-9.99400E-02 -9.99400E-02 -9.99400E-02  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
       5002780
 1.00100E+03  1.00100E+03  1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
-9.99400E-02 -9.99400E-02 -9.99400E-02  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
       5002792
 1.50100E+03  1.00100E+03  1.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
-9.99400E-02 -9.99400E-02 -9.99400E-02  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
```

*DATABASE_TRACER_GENERAL

Purpose: Output histories for any solids, beams, shells, or thick shells in which the tracer, which is a node, is located. The histories are saved into a binary file: trcrgal_binout. These histories include positions, velocities, and stress components. The data structure is identical to the one output by *DATABASE_TRACER_ALE into traleh file. Except for the positions and element ID specifying where the tracer is, the output can be controlled with the VARLOC and VAREPL fields.

Card 1	1	2	3	4	5	6	7	8
Variable	NODE	ELEM	TYPM	MOVE	SET	TYPS		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Optional Variable Cards. Cards defining the times to output data in trcrgal_binout.

Card 2	1	2	3	4	5	6	7	8
Variable	DT	TBEG	TEND	FID				
Type	F	F	F	I				
Default	0.0	0.0	10^{20}	0				

Optional Variable Cards. Cards defining new variables to be output to trcrgal_binout instead of the default ones. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	VARLOC	VAREPL						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
NODE	Node ID that locates the tracer (see Remark 1)
ELEM	Element ID that controls the tracer motion (see Remarks 1 and 2) GT.0: data are output for ELEM if tracer is located inside ELEM LT.0: data are not output for ELEM if tracer is located inside ELEM
TYPM	ELEM type: EQ.1: solid EQ.2: beam EQ.3: shell EQ.4: tshell
MOVE	Flag to define how the tracer moves (see Remark 1): EQ.0: the tracer does not move with ELEM EQ.1: the tracer velocity is interpolated from ELEM nodal velocities EQ.2: the tracer position is interpolated from ELEM nodal positions
SET	Element set for which the data are output by the tracer (see Remark 2)
TYPS	SET type: EQ.0: part EQ.1: solid EQ.2: beam EQ.3: shell EQ.4: tshell
DT	Time interval between outputs (see Remark 3)
TBEG	Time to start the output
TEND	Time to stop the output

VARIABLE	DESCRIPTION
FID	ID to be appended to trcrgal_binout (see Remark 3)
VARLOC	Variable location in trcrgal_binout to be replaced with the variable specified in the VAREPL field: EQ.4: *x*-velocity EQ.5: *y*-velocity EQ.6: *z*-velocity EQ.7: *xx*-stress EQ.8: *yy*-stress EQ.9: *zz*-stress EQ.10: *xy*-stress EQ.11: *yz*-stress EQ.12: *zx*-stress EQ.13: plastic strain EQ.14: nodal mass EQ.15: undefined GE.16 and LE.30: other auxiliary variables
VAREPL	Data to be output to the trcrgal_binout file instead of the variable located at VARLOC. The interpretation of VAREPL is enumerated in the following list: EQ.1: *x*-acceleration EQ.2: *y*- acceleration EQ.3: *z*- acceleration EQ.4: nodal temperature EQ.5: density EQ.6: compression ratio EQ.7: pressure

Remarks:

1. **NODE, ELEM and MOVE.** A node represents the tracer. The initial location of the tracer is given by the nodal coordinates of the node defined in *NODE like for any node. If NODE = 0 and ELEM is defined, a node is added at the center

of the element ELEM. If NODE is defined and ELEM = 0, the closest element to the tracer found during the initialization is used for ELEM. NODE and ELEM cannot both be zero. At least one of them should be provided to position the tracer. If MOVE > 0, the tracer will move with ELEM, even if NODE is not in ELEM.

2. **ELEM and SET.** If SET > 0, data will be output for the element in which the tracer is located if the element belongs to the set (if TYPS = 0, it will be the elements in the parts in SET). If SET = 0 and ELEM > 0, the elements of the same type (TYPM) as ELEM will be the output focus. If SET = 0 and ELEM ≤ 0, data will be output for any solid, beam, shell and thick shell.

3. **DT and FID.** DT is the time interval between data outputs to the file trcrgal_binout. If DT = 0.0 and *DATABASE_TRHIST is defined, the time step in this keyword will be used for DT. If DT = 0.0, data will be output every computational cycle. If there are several *DATABASE_TRACER_GENERAL keywords with different DT values, outputs from tracers with identical time steps will be grouped in the same file with a unique ID added to the end of trcrgen_binout. This ID can be provided by the user with FID. Two keywords with different DT but identical FID will terminate the job with an error.

4. **Post-Processing.** The output of *DATABASE_TRACER_GENERATE is written to a file named trcrgal_binout. To access the output in LS-PrePost: *Post* (or *TAB 2*) → *Binout* → *trcrgal_binout* → *traleh* → *"Traleh Branch"* window contains a list of variables output for each tracer.

5. **Binary to ASCII File Conversion.** The variables in traleh and trcrgal_binout are arranged in an identical order. Therefore, the traleh can be obtained from the trcrgal_binout file by using the l2a program located at http://ftp.lstc.com/-user/lsda.

*DATABASE_TRACER_GENERATE

Purpose: Generate tracer particles along an isosurface for a variable defined in the VALTYPE list. The tracer particles follow the motion of this surface and save data histories into a binary file called trcrgen_binout (see Remarks 4 and 5). These histories are identical to the ones output by *DATABASE_TRACER into the trhist file; they include positions, velocities, and stress components. Except for the positions and element ID specifying where the tracer is, the output can be controlled with the VARLOC and VALTYPE2 fields. This option applies to ALE problems.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	VALOW	VALUP	VALTYPE1	SET	SETYPE	MMGSET	UPDT
Type	F	F	F	I	I	I	I	F
Default	none	0.0	0.0	none	0	0	none	DT

Optional Variable Cards. Cards defining new variables to be output to trcrgen_binout instead of the default ones. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	VARLOC	VALTYPE2	MMGSET					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
DT	Interval time between each tracer generation and position update (See Remark 1).
VALOW, VALUP	Range of values between which the isosurface is defined. VALOW is the lower bound while VALUP is the upper bound. See Remark 2. The value at the isosurface is 0.5(VALOW + VALUP). The variable with this value is defined by VALTYPE.

VARIABLE	DESCRIPTION
VALTYPE1	The variable that will be used to generate the isosurfaces. See VALTYPE2 for enumeration of values.
VALTYPE2	Data to be output to the trcrgen_binout file. The interpretation of VALTYPE1 and VALTYPE2 is enumerated in the following list:

EQ.1:	*xx*-stress
EQ.2:	*yy*-stress
EQ.3:	*zz*-stress
EQ.4:	*xy*-stress
EQ.5:	*yz*-stress
EQ.6:	*zx*-stress
EQ.7:	plastic strain
EQ.8:	internal energy
EQ.9:	bulk viscosity
EQ.10:	relative volume
GE.11 and LE.19:	other auxiliary variables
EQ.20:	pressure
EQ.21:	density
EQ.22:	material volume
EQ.23:	compression ratio
EQ.24:	element volume fraction
EQ.25:	nodal volume fraction
EQ.26:	*x*-position
EQ.27:	*y*-position
EQ.28:	*z*-position
EQ.29:	*x*-velocity
EQ.30:	*y*-velocity
EQ.31:	*z*-velocity
EQ.31:	velocity
EQ.33:	*x*-acceleration
EQ.34:	*y*- acceleration

VARIABLE	DESCRIPTION
	EQ.35: *z*- acceleration
	EQ.36: acceleration
	EQ.37: nodal mass
	EQ.38: nodal temperature
SET	Set ID (See Remark 2)
SETYPE	Type of set (See Remark 2):
	EQ.0: solid set
	EQ.1: segment set
	EQ.2: node set
MMGSET	Multi-material group set (See Remark 3).
UPDT	Time interval between tracer position update (See Remark 1).
VARLOC	Variable location in trcrgen_binout to be replaced with the variable specified in the VALTYPE2 field:
	EQ.4: *x*-velocity
	EQ.5: *y*-velocity
	EQ.6: *z*-velocity
	EQ.7: *xx*-stress
	EQ.8: *yy*-stress
	EQ.9: *zz*-stress
	EQ.10: *xy*-stress
	EQ.11: *yz*-stress
	EQ.12: *zx*-stress
	EQ.13: plastic strain
	EQ.14: density
	EQ.15: relative volume

Remarks:

1. **DT.** The frequency to create tracers is defined by DT. The default value of UPDT, which is the time interval between updates to the tracer position, is also set

to DT. The default behavior, then, is to update tracer positions when a new tracer is created; however, by setting UPDT to a value less than DT tracer positions can be updated more frequently without creating new tracers.

2. **Tracing Algorithm.** When LS-DYNA adds new tracer particles (see DT), tracers are created at element centers, segment centers, or nodes depending on the set type (SETYPE). A new tracer particle is created when the value at the element center, segment center, or node center is in the bounding interval [VALOW, VALUP], provided that there is *not* already a nearby tracer particle. The tracer particles follow the isosurface defined by the midpoint of the bounding interval (VALOW + VALUP)/2.

3. **Multi-Material Groups.** ALE elements can contain several materials. Each material is referred to as an ALE multi-material group. The volume fractions define how much of the element volume is occupied by the groups. Each group has their own variables for 0 < VALTYPE < 26. If VALTYPE < 21 or VALTYPE = 23, the variable is volume averaged over the groups defined by MMGSET.

4. **Post-Procesing.** The output of *DATABASE_TRACER_GENERATE is written to a file named trcrgen_binout. To access the output in LS-PrePost: *TAB 2* → *LOAD* → *trcrgen_binout* → *trhist* → *"Trhist Data"* window contains a list of variables output for each tracer.

5. **Binary to ASCII File Conversion.** The variables in trhist and trcrgen_binout are arranged in an identical order. Therefore, the trhist can be obtained from the trcgen_binout file by using the l2a program located at http://ftp.lstc.com/-user/lsda.

.