



LSTC
Livermore Software
Technology Corp.

LS-DYNA[®] Database Binary Output Files

Revised January 2011

**Copyright ©, 1989-2011
LIVERMORE SOFTWARE
TECHNOLOGY CORPORATION
All Rights Reserved**

Support Address
Livermore Software Technology Corporation
7374 Las Positas Road
Livermore, California 94551

Tel: 925 449 2500

Fax: 925 449 2507

Email: support@lstc.com

Copyright © 1989-2011 by Livermore Software Technology Corporation
All rights Reserved

LS-DYNA, LS-OPT and LS-PREPOST are registered trademarks of Livermore Software Technology Corporation.

TABLE OF CONTENTS

INTRODUCTION.....1

 CONTROL DATA.....3

 MATERIAL TYPE DATA.....7

 FLUID MATERIAL ID DATA.....8

 SMOOTH PARTICLE HYDRODYNAMICS ELEMENT DATA FLAGS.....9

 GEOMETRY DATA.....10

 USER MATERIAL, NODE, AND ELEMENT IDENTIFICATION NUMBERS.....11

 SMOOTH PARTICLE HYDRODYNAMICS NODE AND MATERIAL LIST.....12

 RIGID ROAD SURFACE DATA.....14

 EXTRA DATA TYPES.....17

 STATE DATA.....24

 ELEMENT DELETION OPTION.....31

 SMOOTH PARTICLE HYDRODYNAMICS NODE/ELEMENT STATE DATA.....32

 ROAD SURFACE MOTION.....34

 EXTRA DATA.....35

 END OF FILE MARKER.....39

TIME HISTORY DATABASE (d3thdt)40

 CONTROL DATA.....40

 SMOOTH PARTICLE HYDRODYNAMICS ELEMENT DATA FLAGS.....44

 GEOMETRY DATA.....45

 USER MATERIAL, NODE, AND ELEMENT IDENTIFICATION NUMBERS.....46

 TIME HISTORY DATA.....48

INTERFACE FORCE DATABASE.....58

 CONTROL DATA.....58

 GEOMETRY DATA.....60

 USER MATERIAL, NODE, AND ELEMENT IDENTIFICATION NUMBERS.....61

 STATE DATA.....64

INTRODUCTION

Three databases are discussed, these are:

1. State Database (G=ptf, default name d3plot and d3part)
2. Time History Database (F=thf, default name d3thdt)
3. Interface Force Database (S=iff, no default name, typically: intfor)

The purpose of this information is to give guidance on how to access and read the various databases.

The databases are written as word addressable fixed length binary files. The actual length depends on the amount of data saved, but will always be a multiple of 512 words (4 or 8 bytes each). Since it is likely that the database cannot be contained in a single file of length, FEMLEN, the data will spread over several files known as a family of files. Having a set of files enables them to be handled more easily than a single very large file. The root name for a family is the name of the first file member. Successive member names are compiled by appending a two or three digit number to the root name starting with 01, 02, through to 99, then 100 and ending with 999. Root names are limited to 75 characters. The original reason for a family of files was because the hard disks used for dyna3d runs could not cope with a single contiguous files large enough to contain all the data output. Subsequently, it has been found that splitting the output into separate files allows some unwanted data to be discarded and aids the copying, saving and movement of the data. Very large files can be impossible to transfer reliably over networks.

Total output data can amount to several gigabytes or more depending on the model size.

For ls-dyna runs with mesh adaptivity, the root name has a two letter appendage for each adapted mesh. Starting from 'aa' through 'az', then 'ba' through 'bz' and continuing up to 'zz', this gives a maximum of 676 possible adaptations. For example if the root name is 'd3plot' the subsequent files related to the original mesh are 'd3plot01', d3plot02, ..., after adaption the new mesh and undeformed geometry is put in 'd3plotaa' and the subsequent files for the new mesh are: 'd3plotaa01', d3plotaa02, ... The next adapted mesh will be in file 'd3plotab' and so on. A set of files at a particular adaption can be read separately by giving the root name with appendage, as the base file name.

For example command: 'lsprepost d3plot' will read in all the file with root name 'd3plot'

While: 'lsprepost d3plotab' will read in only files have 'd3plotab' in the name.

LSPREPOST will read the binary databases separately or combined.

Eg: lsprepost d3plot, lsprepost d3thdt, lsprepost iffname, lsprepost d3plot h=d3thdt f=iffname

The file length used is set in the ls-dyna run as the default size of 7x512x512 words. The size can be changed on the command line with the 'x=*factor*' parameter giving a size of: *factor*x512x512 words.

If the initial data or state data is larger than the given file length, the data will automatically split across files. This condition is not desirable because it is not clear whether any non root file can be discarded without destroying the continuity of the data. Ls-dyna checks before writing to a file, to ensure that there is room left in the file to contain the data at a particular state time. If not, it closes the current family member and starts writing the state data in the next file member. The files are written with a block size of 512 words, and if the data does not complete the last block it is padded out. This means that files cannot be concatenated and read together. The word size is 4 bytes for the single precision version of ls-dyna and 8 bytes for the double precision version, unless 32bit ieee format is defined, see *DATABASE_FORMAT, IBINARY.

FILE GENERAL STRUCTURE

The root file starts with a control words section, followed by node coordinates, then element connectivity for solids, thick shells, beams, and shells. Next are lists to reference the sequential internal numbering to the users number. State data is output next always starting with the time word. Data is of fixed length through the file members except where the mesh is adapted. The length of each area can be calculated from the information in the control words. The first file at adaption is like the root file in structure, so the new control words are used to recalculate the size of the subsequent data. The root file contains the initial data and also state data if there is room to write it. Further state data is written to the family members and each file will start with the time word provided data from the previous state did not overflow onto the file.

If the disk address, DA, of the data being written exceeds the maximum file length, then data is written into file number $\text{int}(DA/FAMLEN)$ at location $DA - FAMLEN * \text{int}(DA/FAMLEN)$. If the state length is greater than the remaining length, the disk address is increased to start the writing at the beginning of the next file.

STATE DATABASE (d3plot and d3part)

There are three sections in this database. The first contains 64 words of control information plus extensions. The second contains geometric information including the nodal coordinates and element connectivities and user numbering lists. The third section contains the results of the analysis at sequential output intervals. The output at a given time, called a state, contains a time word, global variables such as total energies and momenta for the whole model and each material (part), node data consisting of displacements, velocities, accelerations, and optionally temperatures, and finally element data that can include stresses and strains at integration points, and element deletion flags. The control data provides information about what is in the file and is used to calculate the various data length.

There are two other state database files, namely: d3drfl and d3part, these are similar to d3plot but contain less data. The dynamic relaxation file, d3drfl, provides the state at the end of the DR process, while d3part is state output for a reduced number of parts in the model.

CONTROL DATA

VALUE	#WORDS	DISK ADDRESS	DESCRIPTION
Title	10	0	Model identification
Run time	1	10	time in seconds since 00:00:00 UTC, January 1, 1970
File type	1	11	d3plot=1 1=d3plot, 2=d3drfl, 3=d3thdt, 4=intfor, 5=d3part 6=blstfor, 7=d3cpm, 8=d3ale, 11=d3eigv, 12=d3mode, 13=d3iter, 21=d3ssd, 22=d3spcm, 23=d3psd, 24=d3rms, 25=d3ftg
Source version	1	12	ls-dyna version *1000000 + svn number
Release number	1	13	Release number in character*4 form 50 for R5.0 511c for R5.1.1c
Version	1	14	Code version, floating number, eg 960.0 it is used to distinguish the floating point format, like cray, ieee, and dpiece
NDIM	1	15	Number of dimensions (2 or 3). If 5 or 7 then an array of material types is read (MAT TYP=1), element connectivities are unpacked and NDIM=3. If 4 then element connectivities are unpacked in the DYNA3D

LS-DYNA DATABASE

			database and NDIM is reset to 3. If >5 then state data contains movement of rigid road surface.
NUMNP	1	16	Number of nodal points
ICODE	1	17	Flag to identify finite element code=2: old DYNA3D, code=6: NIKE3D, LS-DYNA/3D, LS-NIKE3D database
NGLBV	1	18	Number of global variable to be read with each state NUMRW=number of rigid walls. NUMRBS=number of rigid body sets. = 6 + 6 * (NUMMAT8 + NUMMAT2 + NUMMAT4 + NUMATT+NUMRBS) + NUMRW * N N = 1 for DYNA3D and LS-DYNA3D N = 4 for LS-DYNA >= version 971
IT	1	19	Flag for temperatures = 0, none, = 1, read in a temperature for each node = 2, read temperature for each node and heat flux for each node. = 3, read thermal shell middle temperature, thermal shell inner temperature, thermal shell outer temperature, and heat flux for each node. Solid node temperatures are repeated +=10, read mass scaling value for each node
IU	1	20	Flag for current geometry (=1 or 0)
IV	1	21	Flag for velocities (=1 or 0)
IA	1	22	Flag for accelerations (=1 or 0)
NEL8	1	23	Number of 8 node solid elements If NEL8 < 0, 2 extra nodes are output for ten node solids. Array is 2 * abs(NEL8), and is read after 8 node connectivity array.
NUMMAT8	1	24	Number of materials used by the 8 node solids
BLANK	1	25	Insert zero
BLANK	1	26	Insert zero
NV3D	1	27	Number of values in database for each solid element. =7+NEIPH If NV3D is 8 * (7+NEIPH), each solid element has values at each Gauss point.
NEL2	1	28	Number of 2 node one-dimensional elements
NUMMAT2	1	29	Number of materials used by the 2 node 1D elements

NV1D	1	30	Number of values in database for each 1D element = 6 + BEAMIP * 5
NEL4	1	31	Number of four node two-dimensional elements
NUMMAT4	1	32	Number of materials used by the 4 node 2D elements
NV2D	1	33	Number of values in database for each 2D element Are: $MAXINT*(6*IOSHL(1)+IOSHL(2)+NEIPS)+8$ $*IOSHL(3)+4*IOSHL(4)+12*ISTRN$
NEIPH	1	34	Number of additional values per solid element to be written in the type 6 database, $NV3D=7+NEIPH$, Actual number of history variables= $NEIPH-6*ISTRN$
NEIPS	1	35	Number of additional values per integration point to be written into the type 6 database for shell elements.
MAXINT	1	36	Number of integration points dumped for each shell. The magnitude of MAXINT must be greater than or equal to 3. if $MAXINT \geq 0$, then MDLOPT=0 $MAXINT=MAXINT$ elseif $MAXINT < 0$, then MDLOPT=1 $MAXINT=abs(MAXINT)$ elseif $MAXINT < 10,000$, then MDLOPT=2 $MAXINT=abs(MAXINT)-10,000$ endif MDLOPT controls the element deletion table (see below). The increase in state lengths allows deletion by nodes or elements.

LS-DYNA DATABASE

EDLOPT	1	37	Element deletion flag (not standard) =xxx1 Solids deleted =xx1x Beams deleted =x1xx Shells deleted =1xxx Thick Shells deleted (Not used in LS-DYNA)
NMSPH	1	37	Number of SPH Nodes
NGSPH	1	38	Number of SPH materials
NARBS	1	39	Additional storage required for arbitrary node and element numbering in type 6 database =0 Sequential numbering =(10+NUMNP+ NEL8+NEL2+NEL4+NELT).
NELT	1	40	Number of 8 node thick shell elements. $MAXINT*(6*IOSHL(1)+IOSHL(2)+NEIPS)+12*ISTRN$
NUMMATT	1	41	Number of materials used for the 8 node thick shell element.
NV3DT	1	42	Number of values in database for each thick shell
IOSHL(1)	1	43	6 stress components flag, if 1000 =1 else =0
IOSHL(2)	1	44	Plastic strain flag, if 1000 =1 else =0
IOSHL(3)	1	45	Shell force resultants flag, if 1000 =1 else =0
IOSHL(4)	1	46	Shell thickness, energy+2 others, if 1000 =1 else =0
IALEMAT	1	47	Size of array containing solid element parts numbers used as ALE material
NCFDV1	1	48	Bit flags for CFD nodal values. If = 67108864, then state contains CFD extra data – see below (ls980 version)
NCFDV2	1	49	Further bit flags for CFD nodal values. If extra data, then value equals number of data domains (ls980 version)
NADAPT	1	50	Number of adapted element to parent pairs (not implemented)
NMMAT	1	51	Total number of materials – not set in LS-DYNA/3D
NUMFLUID	1	52	Total number of ALE fluid groups. Fluid density and volume fractions output as history variables, and a flag for the dominant group. If negative multi-material species mass for each group is also output. Order is: rho, vf1, ... vfn, dvf flag, m1, ... mn. Density is at position 8

after the location for plastic strain. Any element material history variables are written before the Ale variables, and the six element strains components after these if ISTRN=1.

ID3PROP	1	53	Number of material data sets. (no longer used)
NPEFG	1	54	Number of particle method data sets.
NVEFG	1	55	Not used.
IDTDT	1	56	Rate of change of temperature per node flag. An array of dT/dt values of length NUMNP. Array is written after node temperature arrays.
BLANK	7	57	Unused space.

The value of ISTRN must be computed, it is not output in the control data

ISTRN can only be computed as follows and if NV2D > 0.

If $NV2D - MAXINT * (6 * IOSHL(1) + IOSHL(2) + NEIPS) + 8 * IOSHL(3) + 4 * IOSHL(4) > 1$

Then ISTRN = 1, else ISTRN = 0

If ISTRN=1, and NEIPH>=6, last the 6 additional values are the six strain components.

Or NELT > 0

If $NV3DT - MAXINT * (6 * IOSHL(1) + IOSHL(2) + NEIPS) > 1$

Then ISTRN = 1, else ISTRN = 0

MATERIAL TYPE DATA

The material section contains the material type numbers. This section is skipped if MATTYP is zero.

This data is required because those shell elements that are in a rigid body have no element data output in the state data section. The normal length of the shell element state data is:

NEL4 * NV2D, when the MATTYP flag is set the length is: (NEL4 – NUMRBE) * NV2D.

When reading the shell element data, the material number must be check against IRBRTYP list to find the element's material type. If the type = 20, then all the values for the element to zero.

This option is set in *DATABASE_EXTENT_BINARY, with DCOMP=2

VALUE	LENGTH	DESCRIPTION
NUMRBE	1	Number of rigid body shell elements.

LS-DYNA DATABASE

NUMMAT	1	Number of materials in the database.
IRBTYP	NUMMAT	Material type numbers

FLUID MATERIAL ID DATA

The fluid material section contains the material numbers for solid elements that are used to define an Euler grid or Arbitrary Lagrangian Euler mesh. This section is skipped if IALEMAT is zero.

VALUE	LENGTH	DESCRIPTION
FLUIDID	IALEMAT	Fluid material number used in solid element mesh

SMOOTH PARTICLE HYDRODYNAMICS ELEMENT DATA FLAGS

This section is only output if NMSPH > 0. The section is a list of flags to indicate what SPH data is output for each SPH node/element. The first number is the length in words for this array, currently = 10.

SPH elements are centered at nodes, and cover a spherical volume defined by the radius of influence. They do not have a connectivity with other SPH elements. They should be displayed as a dot or a spherical surface, with radius scaling to reduce the size and enable each element to be distinguishable.

As follows:

- isphfg(1) = 10 - length of sph flags array
- isphfg(2) = 1 - radius of influence
- isphfg(3) = 1 - pressure in particle
- isphfg(4) = 6 - 6 true stress components
- isphfg(5) = 1 - plastic strain, > 0.0 if effective stress exceeds yield strength
- isphfg(6) = 1 - density of particle material
- isphfg(7) = 1 - internal energy (strain)
- isphfg(8) = 1 - number of neighbors affecting particle
- isphfg(9) = 6 - 6 true strain components
- isphfg(10)=1 - mass of element (>= ls971)

If the value of isphfg(2-10) = 0, then the particular data item is not output for the particle. To calculate the size of data add the isphfg values from isphfg(2) through isphfg(10) and add one. One value is always output which is the material number as a floating point number for each particle.

If this value is negative then the particle has been deleted from the model.

Full output for each particle is:

mat#, radius, pressure, {sx, sy, sz, sxy, syz, sxz} ps, rho, ie, nn, {ex, ey, ez, exy, eyz, exz}, mass.

Hence total size is 20.

When a particle is deleted from the model, data is still output for it because the length of data must always be the same for each state.

GEOMETRY DATA

The geometry section contains the nodal coordinates and the element connectivities. The ordering of the nodal points is the same as the ordering of the nodal data in the state data that follows. If NDIM=3 the connectivities are assumed to be packed with 3 integers per word, if NDIM>3, then connectivities are not pack, (the default for LS-DYNA, LS-DYNA3D and LS-NIKE3D. The order of the elements are 3, 2, and 1 dimensional elements if the database is ICODE=2 or 6.

VALUE	LENGTH	DESCRIPTION
X(3,1)	NDIM*NUMNP	Array of nodal coordinates X1,Y1,Z1, X2,Y2,Z2, X3,Y3,Z3, ... ,Xn,Yn,Zn
IX8(9,1)	9*NEL8	Connectivity and material number for each 8 node solid element.
If NEL8 < 0	2*abs(NEL8)	Extra nodes for ten node solids.
IXT(9,1)	9*NELT	Connectivity and material number for each 8 node thick shell element.
IX2(6,1)	6*NEL2	Connectivity, orientation node, two null entries, and the material number for each 2 node beam element.
IX4(5,1)	5*NEL4	Connectivity and material number for each 4 node shell element

Note the node numbers are the LS-DYNA internal numbers for nodes, these will be the same as the user's numbers if NARBS = 0, otherwise, the arbitrary number lists are used to find the user's numbers, similarly, for element numbers and material numbers.

USER MATERIAL, NODE, AND ELEMENT IDENTIFICATION NUMBERS

Skip this section if NARBS (disk address 39) is zero. The user node and element numbers must be in ascending order. *It assumed that if this option is used all node and element data anywhere in the databases is in ascending order in relation to the user numbering.* The total length of the data in this data is equal to:

$$\text{NARBS} = 10 + \text{NUMNP} + \text{NEL8} + \text{NEL2} + \text{NEL4} + \text{NELT},$$

if sequential numbering is used of the materials. For arbitrary material numbering the total length is increased by $6 + \text{NUMMAT8} + \text{NUMMAT4} + \text{NUMMAT2} + \text{NUMMATT}$. Material numbers are not in ascending order.

VALUE	LENGTH	DESCRIPTION
NSORT	1	Pointer to arbitrary node numbers in LS-DYNA source code, a negative sign flags that arbitrary material identification numbers are also used.
NSRH	1	Pointer to arbitrary solid element numbers in LS-DYNA source code: =NSORT+NUMNP
NSRB	1	Pointer to arbitrary beam element numbers in LS-DYNA source code: =NSRH+NEL8
NSRS	1	Pointer to arbitrary shell element numbers in LS-DYNA source code: =NSRB+NEL2
NSRT	1	Pointer to arbitrary thick shell element numbers in LS-DYNA source code: =NSRS+NEL4
NSORTD	1	Number of nodal points
NSRHD	1	Number of 8 node solid elements
NSRBD	1	Number of 2 node beam elements
NSRSD	1	Number of 4 node shell elements
NSRTD	1	Number of 8 node thick shell elements
NSRMA	1	Pointer to an array in the LS-DYNA source code that list the material ID's in ascending order.
NSRMU	1	Pointer to an array in the LS-DYNA source code that gives the material ID's in the actual order that they are defined in the user input.

LS-DYNA DATABASE

VALUE	LENGTH	DESCRIPTION
NSRMP	1	Pointer to an array in the LS-DYNA source code that gives the location of a member in the array originating at NSRMU for each member in the array starting at NSRMA.
NSRTM	1	Total number of materials
NUMRBS	1	Total number of nodal rigid body constraint sets
NMMAT	1	Total number of materials
NUSERN	NSORTD	Array of user defined node numbers
NUSERH	NSORTH	Array of user defined solid element numbers
NUSERB	NSORTB	Array of user defined beam element numbers
NUSERS	NSORTS	Array of user defined shell element numbers
NUSERT	NSORTT	Array of user defined thick shell numbers
NORDER	NMMAT	Ordered array of user defined material ID's
NSRMU	NMMAT	Unordered array of user material ID's
NSRMP	NMMAT	Cross reference array

ADAPTED ELEMENT PARENT LIST (not implemented)

List of element id pairs for H-type shell element adaptivity.

Length of data is $2 * NADAPT$, pairs are element number and element parent number

SMOOTH PARTICLE HYDRODYNAMICS NODE AND MATERIAL LIST

If $NMSPH > 0$ List of sph node and its material number

Length of data $2 * NUMSPH$

PARTICLE DATA (NPEFG > 0)

Control block

If NPEFG > 0 airbag particles are output

The first three digits of NPEFG are the number of airbags in the database = NPARTGAS

NPARTGAS = NPRFG % 1000

SUBVER = NPEFG / 1000

In the extended control block:

The first four words in the block are:

1. NGEOM number of geometry variables
2. NVAR number of state variables
3. NPART number of particles
4. NSTGEOM number of state geometry variables

NLIST = NGEOM + NVAR + NSTGEOM

NLIST words of output for variables listed to define the type of each variable, =1 for integer
and 2= for floating point

2 * NLIST words of variable names (8 bytes per name or 16 for double precision output).

GEOMETRY DATA

NPARTGAS blocks of NGEOM data to describe the geometry for each airbag:

1. first particle ID for the airbag
2. number of particles in the airbag
3. ID for the airbag
4. number of gas mixtures in the airbag

RIGID ROAD SURFACE DATA

If NDIM > 5

NNODE Number of nodes in road surface
NSEG Total number of 4 noded road surface segments
NSURF Number of road surfaces
MOTION Flag to indicate motion data is output for each state
NODEID NNODE list of IDs
SURFNODE XYZ Coordinate for each node
Lists of 4 noded segments for each surface
SURFID Surface ID Number
SURFNSEG Number of segments in surface
SURFSEGS SURFNSEG of 4 node ids for each segment

Length of data = 4 + NNODE + 3 * NNODE + NSURF * (2 + 4 * SURFNSEG)

PART TITLES

At the end of the first binary files, eg d3plot, the part and model titles are appended.
 If the model input includes *DATABASE_BINARY_D3PROP, all the d3prop part data is included.

At the end of the first interface force file, titles and contact id are appended.

This extra data is written at the end of the following files:
 d3plot, d3part and intfor files, and the header and part titles are written directly after the EOF (= -999999.0) marker.

Part title output

Value	Length	Description
NTYPE	1	entity type = 90001
NUMPROP	1	number of parts

For NUMPROP parts:

IDP	1	part id
PTITLE	18	Part title (72 characters)

Header output

NTYPE	1	entity type = 90000
HEAD	18	Header title (72 characters)

For the interface force file (intfor), header and contact titles are written at the end of first file after the EOF (= -999999.0) marker.

Contact title output

NTYPE	1	entity type = 90002
NUMCON	1	number of contacts

For NUMCON contacts:

IDC	1	contact id
CTITLE	18	Contact title (72 characters)

Header output

NTYPE	1	entity type = 90000
HEAD	18	Header title (72 characters)

The d3prop data is written to the d3plot file only if it is requested.

D3PROP output

Values Length Description

NTYPE 1 entity type = 900100
NLINE 1 number of keyword lines

For NLINE keyword lines:

KEYWORD 20 keyword line (80 characters)

DESCRIPTION OF BINARY FILE TYPES

Control word 11

File type:

1=d3plot	plot file of model and state data
2=d3drlf	plot file of model and state data from a dynamic relaxation analysis
3=d3thdt	time history plot file for a set of nodes and elements
4=intfor	plot file of contact interfaces
5=d3part	plot file of model and state data for a set of parts
6=blstfor	plot file for a blast wave analysis
7=d3cpm	
8=d3ale	plot file for ale fluid-structure interface
or fsifor	
11=d3eigv	plot file for an eigen value analysis
12=d3mode	
13=d3iter	
21=d3ssd	plot file for steady state dynamic response.
22=d3spcm	plot file for response spectrum analysis.
23=d3psd	plot file for power spectral density of response, in random vibration.
24=d3rms	plot file for root mean square of response, in random vibration.
25=d3ftg	plot file for random fatigue analysis.

EXTRA DATA TYPES

If NCFDV1 = 67108864, then NCFDV2 will be the number of additional datasets from different solver-mesh combinations that are included in the d3plot file. One of each of the solver-mesh combinations listed below can be among the NCFDV2 datasets. Currently defined solver-mesh combinations follow.

For the following domain, the mesh can be completely different for each output state, so no mesh is output in this control block.

```

solver and domain ID:          PFEM_IF

number of volume vars output:  nvolvar_pfem
first volume variable ID:      ID 1
...
last volume variable ID:       ID nvolvar_pfem

number of PFEM parts           nPFEM_parts
first internal part ID:        partID 1
...
last internal part ID:         partID nPFEM_parts
first user part ID:            user_partID 1
...
last user part ID:             user_partID nPFEM_parts

```

For the following domain, the mesh can be completely different for each output state, so no mesh is output in this control block.

```

solver and domain ID:          PFEM_IF_SURFACE

number of surface vars output:  nsurfvar_pfem
first surface variable ID:      ID 1
...
last surface variable ID:       ID nsurfvar_pfem

number of PFEM parts           nPFEM_surfparts
first internal part ID:        partID 1
...
last internal part ID:         partID nPFEM_surfparts
first user part ID:            user_partID 1
...
last user part ID:             user_partID nPFEM_surfparts

```

For the following domain, the mesh can be completely different for each output state, so no mesh is output in this control block.

```

solver and domain ID:          CESE

number of volume vars output:  nvolvar_cese
first volume variable ID:      ID 1
...
last volume variable ID:       ID nvolvar_cese

number of CESE parts           nCESE_parts
first internal part ID:        partID 1
...
last internal part ID:         partID nCESE_parts

```

LS-DYNA DATABASE

```
first user part ID:          user_partID  1
...
last user part ID:          user_partID  nCESE_parts
```

For the following domain, the mesh can be completely different for each output state, so no mesh is output in this control block.

```
solver and domain ID:      CESE_SURFACE

number of surface vars output:  nsurfvar_cese
first surface variable ID:    ID 1
...
last surface variable ID:    ID nsurfvar_cese

number of CESE parts        nCESE_surfparts
first internal part ID:    partID  1
...
last internal part ID:    partID  nCESE_surfparts
first user part ID:       user_partID  1
...
last user part ID:       user_partID  nCESE_surfparts
```

For the following domain, the mesh can be completely different for each output state, so no mesh is output in this control block.

```
solver and domain ID:      EM

number of volume vars output:  nvolvar_EM
first volume variable ID:    ID 1
...
last volume variable ID:    ID nvolvar_EM

number of EM parts          nEM_parts
first internal part ID:    partID  1
...
last internal part ID:    partID  nEM_parts
first user part ID:       user_partID  1
...
last user part ID:       user_partID  nEM_parts
```

For the following domain, the mesh can be completely different for each output state, so no mesh is output in this control block.

```
solver and domain ID:      EM_SURFACE

number of surface vars output:  nsurfvar_EM
first surface variable ID:    ID 1
...
last surface variable ID:    ID nsurfvar_EM

number of EM parts          nEM_surfparts
first internal part ID:    partID  1
...
last internal part ID:    partID  nEM_surfparts
first user part ID:       user_partID  1
...
last user part ID:       user_partID  nEM_surfparts
```

In this domain, the variables are defined at the mesh nodes.

```

solver and domain ID:          CESE_CFD_NODE
size of each variable component: numnp_cese
number of nodes:              numnp_cese
number of elements:          numelh_cese
user node numbers:           nodes_cese_cfd(numnp_cese)
array of nodal coordinates:  x_cese_cfd(3, numnp_cese)
element connectivity:        ix8_cese_cfd(9, numelh_cese)
number of output vars:       nv_cese_cfd_node
first variable ID:          ID 1
...
last variable ID:           ID nv_cese_cfd_node
number of CESE parts        ncese_parts
first internal part ID:     partID 1
...
last internal part ID:     partID ncese_parts
first user part ID:        user_partID 1
...
last user part ID:         user_partID ncese_parts

```

In this domain, the variables are defined at the element centroid.

```

solver and domain ID:          CESE_CFD_ELEMENT
size of each variable component: numelh_cese
number of nodes:              numnp_cese
number of elements:          numelh_cese
user node numbers:           nodes_cese_cfd(numnp_cese)
array of nodal coordinates:  x_cese_cfd(3, numnp_cese)
element connectivity:        ix8_ins_cfd(9, numelh_cese)
number of output vars:       nv_cese_cfd_ele
first variable ID:          ID 1
...
last variable ID:           ID nv_cese_cfd_ele
number of CESE parts        ncese_parts
first internal part ID:     partID 1
...
last internal part ID:     partID ncese_parts
first user part ID:        user_partID 1
...
last user part ID:         user_partID ncese_parts

```

In this domain, the variables are defined by Taylor series expanded around the element centroid.

```

solver and domain ID:          CESE_CFD_ELEMENT_TS
size of each variable component: 4*numelh_cese
number of nodes:              numnp_cese
number of elements:          numelh_cese
user node numbers:           nodes_cese_cfd(numnp_cese)
array of nodal coordinates:  x_cese_cfd(3, numnp_cese)
element connectivity:        ix8_ins_cfd(9, numelh_cese)
number of output vars:       nv_cese_cfd_ts
first variable ID:          ID 1
...
last variable ID:           ID nv_cese_cfd_ts
number of CESE parts        ncese_parts
first internal part ID:     partID 1

```

```
...
last internal part ID:      partID ncse_parts
first user part ID:       user_partID 1
...
last user part ID:        user_partID ncse_parts
```

In this domain, the variables are defined on structural solid elements.

```
solver and domain ID:      EM_FEMSTER_SOLID_INTEG_PTS
size of each variable component: nip_solid_em * numelh
number of nodes:          numnp
number of elements:       numelh
number of integration points: nip_solid_em
number of output vars:    nv_em_solid_integ
first variable ID:        ID 1
...
last variable ID:         ID nv_em_solid_integ
```

In this domain, the variables are defined on structural thick shell elements.

```
solver and domain ID:      EM_FEMSTER_TSHELL_INTEG_PTS
size of each variable component: nip_tshell_em * numelh
number of nodes:          numnp
number of elements:       numelh
number of integration points: nip_tshell_em
number of output vars:    nv_em_tshell_integ
first variable ID:        ID 1
...
last variable ID:         ID nv_em_tshell_integ
```

In this domain, the variables are defined on structural thin shell elements.

```
solver and domain ID:      EM_FEMSTER_SHELL_INTEG_PTS
size of each variable component: nip_shell_em * numelh
number of nodes:          numnp
number of elements:       numelh
number of integration points: nip_shell_em
number of output vars:    nv_em_shell_integ
first variable ID:        ID 1
...
last variable ID:         ID nv_em_shell_integ
```

In this domain, the variables are defined at the centroids of structural solid elements.

```
solver and domain ID:      EM_FEMSTER_SOLID_CENTROID
size of each variable component: numelh
number of nodes:          numnp
number of elements:       numelh
number of output vars:    nv_em_solid_cent
first variable ID:        ID 1
...
last variable ID:         ID nv_em_solid_cent
```

In this domain, the variables are defined at the centroids of structural thick shell elements.

```

solver and domain ID:          EM_FEMSTER_TSHELL_CENTROID
size of each variable component: numelt
number of nodes:              numnp
number of elements:           numelt
number of output vars:        nv_em_tshell_cent
first variable ID:            ID 1
...
last variable ID:             ID nv_em_tshell_cent

```

In this domain, the variables are defined at the centroids of structural thin shell elements.

```

solver and domain ID:          EM_FEMSTER_SHELL_CENTROID
size of each variable component: numels
number of nodes:              numnp
number of elements:           numels
number of output vars:        nv_em_shell_cent
first variable ID:            ID 1
...
last variable ID:             ID nv_em_shell_cent

```

In this domain, the variables are defined at the mesh nodes.

```

solver and domain ID:          EM_FEMSTER_AIR
size of each variable component: nip_air_em * numelh_air_em
number of nodes:              numnp_air_em
number of elements:           numelh_air_em
number of integration points: nip_air_em
user node numbers:            nodes_air_em(numnp_air_em)
array of nodal coordinates:    x_air_em(3,numnp_air_em)
element connectivity:          ix8_air_em(8, numelh_air_em)
number of output vars:        nv_em_air_integ
first variable ID:            ID 1
...
last variable ID:             ID nv_em_air_integ

```

In this domain, the variables are defined at the nodes of the implied rectangular mesh.

```

solver and domain ID:          RECT_AIR_EM_NODE
size of each variable component: nx_rect_air_em * ny_rect_air_em
                                * nz_rect_air_em
number of x nodes:             nx_rect_air_em
number of y nodes:             ny_rect_air_em
number of z nodes:             nz_rect_air_em
minimum x coordinate:          xmin_rect_air_em
minimum y coordinate:          ymin_rect_air_em
minimum z coordinate:          zmin_rect_air_em
maximum x coordinate:          xmax_rect_air_em
maximum y coordinate:          ymax_rect_air_em
maximum z coordinate:          zmax_rect_air_em
number of output vars:        nv_em_air_nd
first variable ID:            ID 1
...
last variable ID:             ID nv_em_air_nd

```

In this domain, the variables are defined on faces of structural elements.

```

solver and domain ID:          EM_FEMSTER_BEM
size of each variable component: nip_bem_em * nfaces_bem_em

```

LS-DYNA DATABASE

```
number of nodes:                numnp_bem_em
number of elements:             nfaces_bem_em
number of integration points:    nip_bem_em

number of BEM parts:            em_numPartBem

flag for BEM mesh:              nBEMflag
    (first bit =0 if no motion,=1 if motion)
    (second bit =0 if no edge domain,=1 if edge domain)
    (third bit =0 if no node domain,=1 if node domain)

user node numbers:              nodes_bem_em(numnp_bem_em)
array of nodal coordinates:     x_bem_em(3, numnp_bem_em)
element connectivity:           ix4_bem_em(5, nfaces_bem_em)

if (second bit(nBEMflag) = 1)
    number of edges                nedges_bem_em
    number of edge domains          nedgedomain_bem_em
    node edge connectivity          edgex2_bem_em(3, nedges_bem_em)
                                    (internal node1,internal node2,partId)
    number of edges per domain      numEdgesPerDomain(nedgedomain_bem_em)
    edge domain list                edgeDomainList(sum(numEdgePerDomain(i)))
endif

if (third bit(nBEMflag) = 1)
    node element connectivity       nodex4_bem_em(5, nfaces_bem_em)
    number of node domains          nnodedomain_bem_em
    node domain array               nodeDomain(numnp_bem_em)
endif

number of output vars:          nv_em_bem_integ
first variable ID:              ID 1
...
last variable ID:               ID nv_em_bem_integ
```

In this domain, the variables are defined at the particle positions.

```
solver and domain ID:          STOCHASTIC_PARTICLES
number of output vars:          n_prtcl_vars
first variable ID:              ID 1
...
last variable ID:               ID n_prtcl_vars
```

Notes:

The variable IDs are grouped into three groups:

- 1) D3PL_FIRST_SCALAR_ID <= ID < D3PL_FIRST_VECTOR_ID are scalar variables
- 2) D3PL_FIRST_VECTOR_ID <= ID < D3PL_FIRST_TENSOR_ID
are vector variables (3 components per entry)
- 3) D3PL_FIRST_TENSOR_ID <= ID < D3PL_END_IDS
are symmetric tensor variables (6 component per entry)

When a number of integration points are specified, it is assumed that they are distributed at the Gauss points of the given element type based upon how many are output. That is, for shell or face elements,

4 output points would imply the 2x2 Gauss points are used, while 9 output points would imply the 3x3 Gauss points are used, and so forth. Similarly, for volume elements, 8 output points would imply the 2x2x2 Gauss points are used, while 27 output points would imply the 3x3x3 Gauss points are used, and so forth.

STATE DATA

The state data has three parts:

- Time word and global data
- Node data
- Element data for solids, shell, and beams, respectively

VALUE	LENGTH	DESCRIPTION
TIME	1	Time word
GLOBAL	NGLBV	Global variables for this state. LS-DYNA Global Variables: KE, IE, TE, X, Y, and Z velocity, External Work IE for each material KE for each material X, Y, and Z velocity for mat 1 ... X, Y, and Z velocity for mat n Mass for each material Force for each rigid wall Xyz position of wall (ls971) = 6 + 6 * (NUMMAT8 + NUMMAT2 + NUMMAT4 + NUMMATT+NUMRBS) + N * NUMRW, N=1, for ls-dyna(ls971) N=4
NODEDATA	NND	Total nodal values for state. FOR LS-DYNA3D and LS-DYNA IT=1, node temperatures only, N=0 IT=2, node temperature and node flux, N=2 IT=3, 3 temperature per node and node flux, N=3 Temperature for shell node at inner, middle and outer layer, inner array, middle array, outer array. IT/10=1, mass scaling value at node. N+=1 =((IT+N)+NDIM*(IU+IV+IA))*NUMNP where IT=temperature flag, IU=coordinates flag, IV=velocities flag, and IA=accelerations flag.
CFDDATA	CFD	Bit flag: NCFDV1, bits from right to left eg, Pressure, Resultant Vorticity, and Density NCFDV1=2+32+1024=1058 2 Pressure 3 X Vorticity 4 Y Vorticity 5 Z Vorticity 6 Resultant Vorticity 7 Enstrophy 8 Helicity

- 9 Stream Function
- 10 Enthalpy
- 11 Density
- 12 Turbulent KE
- 13 Dissipation
- 14-20 Eddy Viscosity
- Bit flag: NCFDV2
- 2-11 Species 1 through 10

ELEMDATA

ENN

Total element data for state.
 =NEL8*NV3D+NEL2*NV1D+NEL4*NV2D
 The organization of the element data for each element type is described below. The data for the solid elements (7 values/element) is printed first, followed by the data for the beam elements (6 values/element), and then the data for the shell elements (typical 33 or 45 values/element depending on whether the strains are included).

This state data is repeated for each state in the database.

Element data is defined at the integration points within the element. Contour and fringe plots require that the data be extrapolated to the nodal points. In LS_PREPOST the element values are averaged at the nodes. Element strains are not output by default, these are only output for solids, shell, and thick shell when *DATABASE_EXTENT_BINARY, STRFLG=1

SOLID ELEMENTS – 8 node Hexahedron, other solid elements like wedge, pyramid, and tetrahedron are identified by repeated final connectivities. Eg tetrahedron = 1,2,3,4,4,4,4,4

The database for solid elements consists of 7+NEIPH values per element. NEIPH extra values are defined if and only if NEIPH is greater than zero or if the model is an ALE analysis. If strain components are output, then the last 6 neiph values are true strains: ex, ey, ez, exy, eyz, exz, in the global system.

They are:

1. Sigma-x (true stress in the global system)
 2. Sigma-y
 3. Sigma-z
 4. Sigma-xy
 5. Sigma-yz
 6. Sigma-zx
 7. Effective plastic strain or material dependent variable
 8. First extra value (if NEIPH>0)
 9. Second extra value (if NEIPH >1)
 10. Etc. until NEIPH extra values are defined if ISTRN=1
- 7+NEIPH-5. Epsilon-x
7+NEIPH-4. Epsilon-y
7+NEIPH-3. Epsilon-z
7+NEIPH-2. Epsilon-xy
7+NEIPH-1. Epsilon-yz
7+NEIPH. Epsilon-zx

For thick shell elements the database contains $NV3DT = MAXINT * (6 * IOSHL(1) + IOSHL(2) + NEIPS) + 12 * ISTRN$ values per element. Three sets of global stresses are always put into the database for each thick shell and are located at the mid surface, the inner integration point surface, and the outer integration point surface, respectively. If one integration point is used the same through the thickness stress state is outputted three times. If two integration points are used then the mid surface value is taken as the average value. The inner values of the stress are always set to the values at the innermost integration point and likewise for outer values. If the integration point does not lie at the center, ie, an even number of integration points through the thickness, a value is computed that is an average of the two integration point nearest the mid surface.

The IOSHL flags indicate which shell element data is included which is suppressed.

The flags are set in ls-dyna by *DATABASE_EXTENT_BINARY, SIGFLG, EPSFLG, RLFLG, and ENGFLG

The ordering of the data follows:

1. Sigma-x (mid surface true stress in global system)
2. Sigma-y
3. Sigma-z
4. Sigma-xy
5. Sigma-yz
6. Sigma-zx
7. Effective plastic strain or material dependent variable
- *. **Define NEIPS additional history values here for mid surface**
8. Sigma-x (inner surface true stress in global system)
9. Sigma-y
10. Sigma-z
11. Sigma-xy
12. Sigma-yz
13. Sigma-zx
14. Effective plastic strain or material dependent variable
- *. **Define NEIPS additional history values here for inner surface**
15. Sigma-x (outer surface true stress in global system)
16. Sigma-y
17. Sigma-z
18. Sigma-xy
19. Sigma-yz
20. Sigma-zx

21. Effective plastic strain or material dependent variable
- *. **Define NEIPS additional history values here for outer surface**
- *. **If ISTRN=1, then define strain components Epsilon (x, y, z, xy, yz, zx) here for inner surface and outer surface**

If $MAXINT > 3$ then define an additional $(MAXINT-3) * (6 * IOSHL(1) + 1 * IOSHL(2) + NEIPS)$ quantities here.

For beam elements the database contains $NV1D=6$ values per element. They are:

1. Axial force
2. S shear resultant
3. T shear resultant
4. S bending moment
5. T bending moment
6. Torsional resultant

If there are values output at beam integration points, then $NV1D = 6 + 5 * BEAMIP$

1. RS shear stress
2. TR shear stress
3. Axial stress
4. Plastic strain
5. Axial strain

BEAMIP is set in *DATABASE_EXTENT_BINARY

For shell elements the database contains NV2D values, where:
 $NV2D = MAXINT * (6 * IOSHL(1) + 1 * IOSHL(2) + NEIPS) + 8 * IOSHL(3) + 4 * IOSHL(4) + 12 * ISTRN$ values per deformable element. If MATTYP=1 and IRBTYP(I)=20, where I=internal element number, then the material is rigid and the compressed database contains no data for the element. If the minimum value of *MAXINT* is used, i.e., 3, the stresses are typically located at the mid surface, the inner surface, and the outer surface, respectively. If one integration point is used the stress is written three times. If two integration points are used then the mid surface value is taken as the average value. The inner values of the stress are always set to the values at the innermost integration point and likewise for outer values. If no integration point lies at the center, i.e., an even number of integration points through the thickness, a value is computed that is an average of the two integration point lying nearest the mid surface.

The ordering of the data follows:

1. Sigma-x (mid surface true stress in global system)
2. Sigma-y
3. Sigma-z
4. Sigma-x
5. Sigma-yz
6. Sigma-zx
7. Effective plastic strain or material dependent variable
- *. **Define NEIPS additional history values here for mid surface**
8. Sigma-x (inner surface true stress in global system)
9. Sigma-y
10. Sigma-z
11. Sigma-xy
12. Sigma-yz
13. Sigma-zx
14. Effective plastic strain or material dependent variable
- *. **Define NEIPS additional history values here for inner surface**

- 15. Sigma-x (outer surface true stress in global system)
- 16. Sigma-y
- 17. Sigma-z
- 18. Sigma-xy
- 19. Sigma-yz
- 20. Sigma-zx
- 21. Effective plastic strain or material dependent variable

*. **Define NEIPS additional history values here for outer surface**

If *MAXINT* >3 then define an additional (*MAXINT*-3)* (6**IOSHL*(1) + 1**IOSHL*(2) + 8**IOSHL*(3) + 4**IOSHL*(4) + *NEIPS*) quantities here

- 22. Bending moment-Mx (local shell coordinate system)
- 23. Bending moment-My
- 24. Bending moment-Mxy
- 25. Shear resultant-Qx
- 26. Shear resultant-Qy
- 27. Normal resultant-Nx
- 28. Normal resultant-Ny
- 29. Normal resultant-Nxy
- 30. Thickness
- 31. Element dependent variable
- 32. Element dependent variable
- 33. Internal energy (if and only if *ISTRN*=0)

The following quantities are expected if and only if *ISTRN*=1

- 33. eps-x (inner surface strain in global system)
- 34. eps-y
- 35. eps-z
- 36. eps-xy
- 37. eps-yz
- 38. eps-zx
- 39. eps-x (outer surface strain in global system)
- 40. eps-y
- 41. eps-z
- 42. eps-xy
- 43. eps-yz
- 44. eps-zx
- 45. Internal energy (if and only if *NV2D*>=45)

ELEMENT DELETION OPTION

Skip this section if the word MAXINT is greater than or equal to zero, (MDLOPT=0). If MDLOPT=1, then the list is equal to the number of nodal points (NUMNP) and contains a one if the node is visible and a zero if the node is not visible, (only used in vec-dyna3d). If MDLOPT=2, then the list equals the total number of elements (NEL8 + NELT + NEL4 + NEL2), in this order, and each value is set to the element material number or =0, if the element is deleted. All these numbers are output as floating point values and not integers.

SMOOTH PARTICLE HYDRODYNAMICS NODE/ELEMENT STATE DATA

This section is only output if NMSPH>0

For each SPH node the follow values are output:

NUM_SPH_DATA = 1 + \sum isphfg(i), i=2:10

Length of data = NUM_SPH_DATA * NUMSPH

Material number, if <=0 then element is deleted.

Currently isphfg(1) = 10, ie number of sph data flags, this could be changed in the future.

If isphfg(2) =1, radius of particle influence

If isphfg(3) =1, pressure in particle

If isphfg(4) =6, stress components for particle, sx, sy, sz, sxy, syz, sxz

If isphfg(5) =1, plastic strain for particle

If isphfg(6) =1, density of particle material

If isphfg(7) =1, internal energy of particle

If isphfg(8) =1, number of particle neighbors

If isphfg(9) =6, strain components for particle, ex, ey, ez, exy, eyz, exz

If isphfg(10)=1, mass of element (ls971)

Note: it is possible a SPH element could be deleted, or be none active in the initial states, and become active in later states.

PARTICLE STATE DATA (NPEFG > 0)

STATE DATA

NPARTGAS blocks of NSTGEOM data to describe the state geometry for each bag:

1. number of active particles
2. current bag volume

PARTICLE DATA

NVAR words of data output for each particle:

1. gas ID
2. leakage flag, 0 active, -1 fabric, -2 vent hole, -3 mistracked
3. mass
4. radius
5. spin energy
6. translational energy
7. distance from particle to nearest segment
8. x position
9. y position
10. z position
11. x velocity
12. y velocity
13. z velocity

ROAD SURFACE MOTION

If $NDIM > 5$ output rigid body displacement, dx, dy, dz and velocity, vx, vy, vz of each road surface.

Length of data = $6 * NSURF$

EXTRA DATA

If NCFDV1 = 67108864, then the state data includes NCFDV2 additional datasets from solver-mesh combinations specified after the "User material, node, and element identification numbers" for the structural mesh.

State data of the first solver-mesh combination

...

State data of the last (NCFDV2-th) solver-mesh combination

When the state data comes from the PFEM_IF domain, then the mesh is output first, followed by the data. Currently, the mesh is entirely tetrahedral, but we anticipate users will also specify mixed meshes in the near future:

size of each volume variable component:	nnpvol_pfem
number of volume nodes:	nnpvol_pfem
number of tetrahedral elements:	ntet_pfem
number of pyramid elements:	npyr_pfem
number of wedge elements:	nwdg_pfem
number of hexahedral elements:	nhex_pfem
user volume node numbers:	volnodes_pfem(nnpvol_pfem)
array of volume nodal coordinates:	xvol_pfem(3, nnpvol_pfem)
tetrahedral element connectivity:	ix4_pfem(5, ntet_pfem)
pyramid element connectivity:	ix5_pfem(6, npyr_pfem)
wedge element connectivity:	ix6_pfem(7, nwdg_pfem)
hexahedral element connectivity:	ix8_pfem(9, nhex_pfem)

data for 1st volume variable (size is nnpvol_pfem)

...

data for nvolvar_pfem-th volume variable (size is nnpvol_pfem)

Notes:

- (1) the first four entries of ix4_pfem(5, ntet_pfem) for each element are the volume node numbers, while the last entry is the volume part number.
- (2) the first five entries of ix5_pfem(6, npyr_pfem) for each element are the volume node numbers, while the last entry is the volume part number.
- (3) the first six entries of ix6_pfem(7, nwdg_pfem) for each element are the volume node numbers, while the last entry is the volume part number.
- (4) the first eight entries of ix8_pfem(9, nhex_pfem) for each element are the volume node numbers, while the last entry is the volume part number.

When the state data comes from the `PFEM_IF_SURFACE` domain, then the surface mesh is output first, followed by the data. Currently, the surface mesh is entirely triangular, but we anticipate users will also specify mixed triangle-quadrilateral meshes in the near future:

size of each surface variable component: `nnpsurf_pfem`
number of surface nodes: `nnpsurf_pfem`
number of surface elements: `nelsurf_pfem`
user surface node numbers: `surfnodes_pfem(nnpsurf_pfem)`
surface element connectivity: `ixsurf_pfem(5, nelsurf_pfem)`

data for 1st surface variable (size is `nnpsurf_pfem`)

...

data for `nsurfvar_pfem`-th surface variable (size is `nnpsurf_pfem`)

Notes:

- (1) the first four entries of `ixsurf_pfem(5, nelsurf_pfem)` for each surface element are the surface node numbers, while the last entry is the surface part number. The 3rd and 4th node numbers are the same for triangles.

When the state data comes from the `CESE` domain, then the mesh is output first, followed by the data. Currently, the mesh is entirely tetrahedral, but we anticipate users will also specify mixed meshes in the near future:

size of each volume variable component: `nele_cese`
number of volume nodes: `nnpvol_cese`
number of tetrahedral elements: `ntet_cese`
number of pyramid elements: `npyr_cese`
number of wedge elements: `nwdg_cese`
number of hexahedral elements: `nhex_cese`
user volume node numbers: `volnodes_cese(nnpvol_cese)`
array of volume nodal coordinates: `xvol_cese(3, nnpvol_cese)`
tetrahedral element connectivity: `ix4_cese(5, ntet_cese)`
pyramid element connectivity: `ix5_cese(6, npyr_cese)`
wedge element connectivity: `ix6_cese(7, nwdg_cese)`
hexahedral element connectivity: `ix8_cese(9, nhex_cese)`

data for 1st volume variable (size is `nele_cese`)

...

data for `nvolvar_cese`-th volume variable (size is `nele_cese`)

Notes:

- (1) `nele_cese = ntet_cese + npyr_cese + nwdg_cese + nhex_cese`
- (2) the first four entries of `ix4_cese(5, ntet_cese)` for each element are the volume node numbers, while the last entry is the volume part number.
- (3) the first five entries of `ix5_cese(6, npyr_cese)` for each element are the volume node numbers, while the last entry is

- the volume part number.
- (4) the first six entries of `ix6_cese(7, nwdg_cese)` for each element are the volume node numbers, while the last entry is the volume part number.
- (5) the first eight entries of `ix8_cese(9, nhex_cese)` for each element are the volume node numbers, while the last entry is the volume part number.

When the state data comes from the `CESE_SURFACE` domain, then the surface mesh is output first, followed by the data. Currently, the surface mesh is entirely triangular, but we anticipate users will also specify mixed triangle-quadrilateral meshes in the near future:

```

size of each surface variable component: nnpsurf_cese
number of surface nodes:                nnpsurf_cese
number of surface elements:             nelsurf_cese
user surface node numbers:              surfnodes_cese(nnpsurf_cese)
surface element connectivity:           ixsurf_cese(5, nelsurf_cese)

```

data for 1st surface variable (size is nnpsurf_cese)

...

data for nsurfvar_cese-th surface variable (size is nnpsurf_cese)

Notes:

- (1) the first four entries of `ixsurf_cese(5, nelsurf_cese)` for each surface element are the surface node numbers, while the last entry is the surface part number. The 3rd and 4th node numbers are the same for triangles.

When the state data comes from the `EM` domain, then the mesh is output first, followed by the data. Currently, the mesh is entirely tetrahedral, but we anticipate users will also specify mixed meshes in the near future:

```

size of each volume variable component: nnpvol_EM
number of volume nodes:                nnpvol_EM
number of tetrahedral elements:        ntet_EM
number of pyramid elements:            npyr_EM
number of wedge elements:              nwdg_EM
number of hexahedral elements:         nhex_EM
user volume node numbers:              volnodes_EM(nnpvol_EM)
array of volume nodal coordinates:     xvol_EM(3, nnpvol_EM)
tetrahedral element connectivity:      ix4_EM(5, ntet_EM)
pyramid element connectivity:          ix5_EM(6, npyr_EM)
wedge element connectivity:            ix6_EM(7, nwdg_EM)
hexahedral element connectivity:       ix8_EM(9, nhex_EM)

```

data for 1st volume variable (size is nnpvol_EM)

...

data for nvolvar_EM-th volume variable (size is nnpvol_EM)

Notes:

- (1) the first four entries of `ix4_EM(5, ntet_EM)` for each element are the volume node numbers, while the last entry is the volume part number.
- (2) the first five entries of `ix5_EM(6, npyr_EM)` for each element are the volume node numbers, while the last entry is the volume part number.
- (3) the first six entries of `ix6_EM(7, nwdg_EM)` for each element are the volume node numbers, while the last entry is the volume part number.
- (4) the first eight entries of `ix8_EM(9, nhex_EM)` for each element are the volume node numbers, while the last entry is the volume part number.

When the state data comes from the `EM_SURFACE` domain, then the surface mesh is output first, followed by the data. Currently, the surface mesh is entirely triangular, but we anticipate users will also specify mixed triangle-quadrilateral meshes in the near future:

```
size of each surface variable component: nnpsurf_EM
number of surface nodes:                nnpsurf_EM
number of surface elements:             nelsurf_EM
user surface node numbers:              surfnodes_EM(nnpsurf_EM)
surface element connectivity:           ixsurf_EM(5, nelsurf_EM)
```

```
data for 1st surface variable           (size is nnpsurf_EM)
...
data for nsurfvar_EM-th surface variable (size is nnpsurf_EM)
```

Notes:

- (1) the first four entries of `ixsurf_EM(5, nelsurf_EM)` for each surface element are the surface node numbers, while the last entry is the surface part number. The 3rd and 4th node numbers are the same for triangles.

When the state data comes from the `STOCHASTIC_PARTICLES` domain, then the

```
size of each variable component:  n_particles
array of particle positions:       x_particles(3, n_particles)
data for 1st output variable
...
data for n_prctl_vars-th output variable
```

There will always be at least the following two variables output for each particle domain: `PARTICLE_SIZES` and `PARTICLE_VELOCITIES`. That is, `n_prctl_vars` \geq 2. For each particle, both the position and velocity are a 3-component vector.

END OF FILE MARKER

Value = -999999.0 (a floating point number)

TIME HISTORY DATABASE (d3thdt)

There are three sections in the LS-DYNA time history database. The first used to contain 144 words of control information, but now depends upon the number of node and elements the user defines in LS-DYNA. The second contains geometric information including the nodal coordinates and element connectivities. The third section contains the results of the analysis at sequential output intervals for a subset of solids, beams, and shells. The output at a given time, called a state, contains a time word, global variables such as total energies and momenta, nodal data consisting of accelerations, velocities, and displacements, and finally element data is written that may include stresses and strains at integration points. The control information provides information on what is in the file and which database is contained.

CONTROL DATA

VALUE	#WORDS	DISK ADDRESS	DESCRIPTION
Title	10	0	Model identification
Run time	1	10	time in seconds since 00:00:00 UTC, January 1, 1970
File type	1	11	d3thdt=3 1=d3plot, 2=d3drlf, 3=d3thdt, 4=intfor, 5=d3part 6=blstfor, 7=d3cpm, 8=d3ale, 11=d3eigv, 12=d3mode, 13=d3iter, 21=d3ssd, 22=d3spcm, 23=d3psd, 24=d3rms, 25=d3ftg
Source version	1	12	ls-dyna version *1000000 + svn number
Release number	1	13	Release number in character*4 form 50 for R5.0 511c for R5.1.1c
Version	1	14	Code version, a real number, not integer
NDIM	1	15	Number of dimensions (2 or 3) is set to 4 if element connectivities are unpacked in the LS-DYNA/3D database and NDIM is reset to 3.
NUMNP	1	16	Number of nodal points
ICODE	1	17	Flag to identify finite element code =2 old DYNA3D, NIKE3D database =6 new LS-NIKE3D, LS-DYNA/3D database
NGLBV	1	18	Number of global variables to be read in each state

IT	1	19	Flag for temperatures =0 none, =1 read in a temperature for each node
IU	1	20	Flag for current geometry (=1)
IV	1	21	Flag for velocities (=1)
IA	1	22	Flag for accelerations (=1)
NEL8	1	23	Number of 8 node solid elements
NUMMAT8	1	24	Number of materials used by the 8 node solids
NDS	1	25	Number of node blocks for plotting
NST	1	26	Number of element blocks for plotting. =NSTH + NSTB + NSTS +NSTT
NV3D	1	27	Number of values in database for each solid element
NEL2	1	28	Number of 2 node one-dimensional elements
NUMMAT2	1	29	Number of materials used by the 2 node 1D elements
NV1D	1	30	Number of values in database for each 1D element
NEL4	1	31	Number of four node two-dimensional elements
NUMMAT4	1	32	Number of materials used by the 4 node 2D elements
NV2D	1	33	Number of values in database for each 2D element
NEIPH	1	34	Number of additional values per solid element to be written in the type 6 database =NEIPH-6*ISTRN
NEIPS	1	35	Number of additional values per integration point to be written into the type 6 database for shell elements
MAXINT	1	36	Number of integration points dumped for each shell element.
NMSPH	1	37	Number of SPH Nodes
NGSPH	1	38	Number of SPH materials
NARBS	1	39	Additional storage required for arbitrary node and element numbering in type 6 database
BLANK	3	40	Unused space
IOSHL(1)	1	43	stress components flag (=1000 yes)
IOSHL(2)	1	44	Strain components, ISTRN (=1000 yes)

LS-DYNA DATABASE

IOSHL(3)	1	45	Shell force resultants (=1000 yes)
IOSHL(4)	1	46	Shell thickness, energy + 2 others (=1000 yes)
BLANK	1	47	Unused space, ignore value
NCFDV1	1	48	Bit flags for CFD nodal values
NCFDV2	1	49	Further bit flags for CFD nodal values
BLANK	8	50	Unused space
NSTP	1	58	Number of SPH element blocks
IFLAGD	1	59	Number of node blocks + 1000 flag
NSTH	1	60	Number of solid element blocks
NSTB	1	61	Number of beam element blocks
NSTS	1	62	Number of shell element blocks
NSTT	1	63	Number of thick shell element block
NDSB	2*NDS	64	Node blocks for which time histories are output. The locations 2n-1, where n=1 through NDS correspond to the first node in the block and locations 2n correspond to the last node in the block.
NSTHB	2*NSTH	64+2*NDS	Solid element blocks start and end numbers These are defined in a similar manner to the nodal time history blocks
NSTBB	2*NSTB	64+2*NDS +2*NSTH	Beam element block start and end numbers
NSTSB	2*NSTS	64+2*NDS +2*NSTH +2*NSTB	Shell element block start and end numbers
NSTTB	2*NSTT	64+2*NDS +2*NSTH +2*NSTB +2*NSTS	Thick shell element block start and end numbers
NSTPB	2*NSTP	64+2*NDS +2*NSTH +2*NSTB +2*NSTS +2*NSTT	SPH element block start and end numbers

ISTRN can only be computed as follows and if NSTS > 0.

If $NV2D-MAXINT*(6*IOSHL(1)+IOSHL(2)+NEIPS)+8*IOSHL(3)+4*IOSHL(4) > 10$

Then ISTRN = 1, else ISTRN = 0

Or NSTT > 0

If $NV3DT-MAXINT*(6*IOSHL(1)+IOSHL(2)+NEIPS) > 10$

Then ISTRN = 1, else ISTRN = 0

SMOOTH PARTICLE HYDRODYNAMICS ELEMENT DATA FLAGS

This section is only output if $NMSPH > 0$. The section is a list of flags to indicate what SPH data is output for each SPH node/element. The first number is the length in words for this array, currently = 10.

SPH elements are centered at nodes, and cover a spherical volume defined by the radius of influence. They do not have a connectivity with other SPH elements. They should be displayed as a dot or a spherical surface, with radius scaling to reduce the size and enable each element to be distinguishable.

As follows:

- isphfg(1) = 10 - length of sph flags array
- isphfg(2) = 1 - radius of influence
- isphfg(3) = 1 - pressure in particle
- isphfg(4) = 6 - 6 true stress components
- isphfg(5) = 1 - plastic strain, > 0.0 if effective stress exceeds yield strength
- isphfg(6) = 1 - density of particle material
- isphfg(7) = 1 - internal energy (strain)
- isphfg(8) = 1 - number of neighbors affecting particle
- isphfg(9) = 6 - 6 true strain components
- isphfg(10)=1 - mass of element

If the value of $isphfg(2-10) = 0$, then the particular data item is not output for the particle. To calculate the size of data add the $isphfg$ values from $isphfg(2)$ through $isphfg(10)$ and add one. One value is always output which is the material number as a floating point number for each particle.

If this value is negative then the particle has been deleted from the model.

Full output for each particle is:

mat#, radius, pressure, {sx, sy, sz, sxy, syz, sxz} ps, rho, ie, nn, {ex, ey, ez, exy, eyz, exz}, mass.

Hence total size is 20.

When a particle is deleted from the model, data is still output for it because the length of data must always be the same for each state.

GEOMETRY DATA

The geometry section contains the nodal coordinates and the element connectivities. The ordering of the nodal points is assumed to be the same as the ordering of the nodal data in the state data that follows. The connectivities are assumed to be packed with 3 integers per word unless NDIM is set to 4 as in the new LS-DYNA/3D, LS-NIKE3D databases. The order of the elements are 3, 2, and 1 dimensional elements if the database is ICODE=2 or 6.

VALUE	LENGTH	DESCRIPTION
X(3,1)	NDIM*NUMNP	Array of nodal coordinates X1,Y1,Z1, X2,Y2,Z2,X3,Y3,Z3, ... ,Xn,Yn,Zn
IX8(9,1)	9*NEL8	Connectivity and material number for each 8 node solid element
IXT(9,1)	9*NELT	Connectivity and material number for each 8 node thick shell element
IX2(6,1)	6*NEL2	Connectivity, orientation node, two null entries, and the material number for each 2 node beam element
IX4(5,1)	5*NEL4	Connectivity and material number for each 4 node shell element

USER MATERIAL, NODE, AND ELEMENT IDENTIFICATION NUMBERS

Skip this section if NARBS (disk address 39) is zero. The user node and element numbers must be in ascending order. *It is assume that if this option is used all the node and element data in the databases is in ascending order in relation to the user numbering.* The total length of the data in this database is equal to

$$\text{NARBS} = 10 + \text{NUMNP} + \text{NEL8} + \text{NEL2} + \text{NEL4} + \text{NELT}$$

if sequential numbering is used of the materials. For arbitrary material numbering the total length is increased by $6 + \text{NUMMAT8} + \text{NUMMAT4} + \text{NUMMAT2} + \text{NUMMATT}$. Material numbers are not in ascending order.

VALUE	LENGTH	DESCRIPTION
NSORT	1	Pointer to arbitrary node numbers in LS-DYNA/3D source code, a negative sign flags that arbitrary material identification numbers are also used.
NSRH	1	Pointer to arbitrary solid element numbers in LS-DYNA source code: =NSORT+NUMNP
NSRB	1	Pointer to arbitrary beam element numbers in LS-DYNA source code: =NSRH+NEL8
NSRS	1	Pointer to arbitrary shell element numbers in LS-DYNA source code: =NSRB+NEL2
NSRT	1	Pointer to arbitrary thick shell element numbers in LS-DYNA source code: =NSRS+NEL4
NSORTD	1	Number of nodal points
NSRHD	1	Number of 8 node solid elements
NSRBD	1	Number of 2 node beam elements
NSRSD	1	Number of 4 node shell elements
NSRTD	1	Number of 8 node thick shell elements
NSRMA	1	Pointer to an array in the LS-DYNA source code that list the material ID's in ascending order.
NSRMU	1	Pointer to an array in the LS-DYNA source code that gives the material ID's in the actual order that they are defined in the user input

VALUE	LENGTH	DESCRIPTION
NSRMP	1	Pointer to an array in the LS-DYNA source code that gives the location of a member in the array originating at NSRMU for each member in the array starting at NSRMA.
NSRTM	1	Total number of materials
NUMRBS	1	Total number of nodal rigid body constraint sets.
NMMAT	1	Total number of materials
NUSERN	NSORTD	Array of user defined node numbers
NUSERH	NSORTH	Array of user defined solid element numbers
NUSERB	NSORTB	Array of user defined beam element numbers
NUSERS	NSORTS	Array of user defined shell element numbers
NUSERT	NSORTT	Array of user defined solid shell numbers
NORDER	NMMAT	Ordered array of user defined material ID's
NSRMU	NMMAT	Unordered array of user material ID's
NSRMP	NMMAT	Cross reference array

TIME HISTORY DATA

The time database contains the following data:

- Time word
- Node data
- Node data for solids, thick shells, and shells, respectively
- Element data for solids, thick shells, beams, and shells, respectively

VALUE	LENGTH	DESCRIPTION
TIME	1	Time word
GLOBAL	NGLBV	Global variables for this state. LS-DYNA Global Variables: KE, IE, TE, X, Y, and Z velocity IE for each material KE for each material X, Y, and Z velocity for mat 1 ... X, Y, and Z velocity for mat n Mass for each material Force for each rigid wall = 6 + 6 * (NUMMAT8+ NUMMAT2 + NUMMAT4 + NUMMATT + NUMRBS) + NUMRW

SKIP THE FOLLOWING DATA IF THE NUMBER OF NODE BLOCKS FOR PLOTTING IS ZERO (VALUE NUMDS AT DISK ADDRESS 25)

TIME	1	Time word
NODEDATA	NND	Total nodal values for state where NLN=10*TNODS where TNODS is the number of nodes put into database. The database contains TNODS vectors each with up to 10 components: temperature (if IT=1); x, y, and z coordinates; x, y, and z velocities; and x, y, and z accelerations.

CFDDATA

CFD

Bit flag: NCFDV1, bits from right to left
Eg Pressure, Resultant Vorticity, and Density

$NCFDV1=2+32+1024=1058$

- 14 Pressure
- 15 X Vorticity
- 16 Y Vorticity
- 17 Z Vorticity
- 18 Resultant Vorticity
- 19 Enstrophy
- 20 Helicity
- 21 Stream Function
- 22 Enthalpy
- 23 Density
- 24 Turbulent KE
- 25 Dissipation
- 14-20 Eddy Viscosity

Bit flag: NCFDV2

2-11 Species 1 through 10

Count number of bits on * NUMNP

SKIP THE FOLLOWING DATA IF THE NUMBER OF ELEMENT BLOCKS FOR IS ZERO
(VALUE NUMDS AT DISK ADDRESS 26)

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR SOLID ELEMENTS ****

TIME	1	Time word
SOLIDDATA	ENV	Total nodal values for solid elements where ENV=56*TBELM where THELM is the total number of solid elements to be put into the database. The data contains THELM vectors each with 56 components ordered as follows: 8 connectivities: x,y,z coordinates for each of the 8 nodes; and, lastly, x,y,z velocities for each of the 8 nodes.

For solid elements the database contains $(7+NEIPH-6*ISTRN)$ values per element. One set of global stresses are always put into the database for each solid element followed by NEIPH history values. Only data for elements defined in the time history blocks is output. The ordering of the data follows:

1. Sigma-x (true stress in the global system)
 2. Sigma-y
 3. Sigma-z
 4. Sigma-xy
 5. Sigma-yz
 6. Sigma-zx
 7. Effective plastic strain or material dependent variable
 8. First extra value (if NEIPH>0)
 9. Second extra value (if NEIPH >1)
 10. Etc. until NEIPH extra values are defined if ISTRN=1
- 7+NEIPH-5. Epsilon-x
7+NEIPH-4. Epsilon-y
7+NEIPH-3. Epsilon-z
7+NEIPH-2. Epsilon-xy
7+NEIPH-1. Epsilon-yz
7+NEIPH. Epsilon-zx

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR THICK SHELLS ****

TIME	1	Time word
TSHELLDATA	ENV	Total nodal values for thick shell elements where $ENV=56*TBSEL$ where TBSEL is the total number of thick shell elements in the database. The data contains TBSEL vectors each with 56 components ordered as follows: 8 connectivities: x,y,z coordinates for each of the 8 nodes; and, lastly, x,y,z velocities for each of the 8 nodes.

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR SHELL ELEMENTS ****

TIME	1	Time word
SHELLDATA	ENVS	Total nodal values for shell elements where $ENVS=28*TSELM$ where TSELM is the total number of shell elements in the database. The data contains TSELM vectors each with 28 components ordered as follows: 4 connectivities: x,y,z coordinates for each of the 4 nodes; and, lastly, x,y,z velocities for each of the 4 nodes.

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR THICK SHELLS ****

For thick shell elements the database contains NV3DT values per element. Three sets of global stresses are always put into the database for each thick shell and are located at the mid surface, the inner surface, and the outer surface, respectively. If one integration point is used the single state is written three times. If two integration points are used then the mid surface value is taken as the average value. The inner values of the stress are always set to the values at the innermost integration point and likewise for outer values. If no integration point lies at the center, i. e. an even number of integration points through the thickness, a value is computed that is an average of the two integration point lying nearest the mid surface. Only data for elements defined in the time history blocks is output. The ordering of the data follows:

1. Sigma-x (mid surface true stress in global system)
2. Sigma-y
3. Sigma-z

- 4. Sigma-xy
 - 5. Sigma-yz
 - 6. Sigma-zx
 - 7. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for midsurface**
 - 8. Sigma-x (inner surface true stress in global system)
 - 9. Sigma-y
 - 10. Sigma-z
 - 11. Sigma-xy
 - 12. Sigma-yz
 - 13. Sigma-zx
 - 14. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for inner surface**
 - 15. Sigma-x (outer surface true stress in global system)
 - 16. Sigma-y
 - 17. Sigma-z
 - 18. Sigma-xy
 - 19. Sigma-yz
 - 20. Sigma-zx
 - 21. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for outer surface**
 - 21. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for outer surface**
- If MAXINT >3 then define an additional (MAXINT-3)* (6*IOSHL(1) + 1*IOSHL(2) + NEIPS) quantities here
- *. **If ISTRN=1, then define strain components Epsilon (x, y, z, xy, yz, zx) here for inner surface and outer surface**

LS-DYNA DATABASE

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR BEAM ELEMENTS ****

TIME	1	Time word
BEAMDATA	BESV	Total element values for beam elements where $BESV = NV1D * TBELM$. The data contains $TBELM$ vectors each with $NV2D$ values ordered as described below.

For beam elements the database contains $NV1D=6$ values per element. They are:

1. Axial force
2. S shear resultant
3. T shear resultant
4. S bending moment
5. T bending moment
6. Torsional resultant

If there are values output at beam integration points, then $NV1D = 6 + 5 * BEAMIP$

7. RS shear stress
8. TR shear stress
9. Axial stress
10. Plastic strain
11. Axial strain

BEAMIP is set in *DATABASE_EXTENT_BINARY

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR SHELL ELEMENTS ****

For shell elements the database contains NV2D values per element. If the minimum value of MAXINT is 3, then the stresses are typically located at the mid surface, the inner surface, and the outer surface, respectively. If one integration point is used the stress is written three times. If two integration points are used then the mid surface value is taken as the average value. The inner values of the stress are always set to the values at the innermost integration point and likewise for outer values. If no integration point lies at the center, i. e. an even number of integration points through the thickness, a value is computed that is an average of the two integration point lying nearest the mid surface. Only data for elements defined in the time history blocks is output. The ordering of the data follows:

1. Sigma-x (mid surface true stress in global system)
 2. Sigma-y
 3. Sigma-z
 4. Sigma-xy
 5. Sigma-yz
 6. Sigma-zx
 7. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for midsurface**
 8. Sigma-x (inner surface true stress in global system)
 9. Sigma-y
 10. Sigma-z
 11. Sigma-xy
 12. Sigma-yz
 13. Sigma-zx
 14. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for inner surface**
 15. Sigma-x (outer surface true stress in global system)
 16. Sigma-y
 17. Sigma-z
 18. Sigma-xy
 19. Sigma-yz
 20. Sigma-zx
 21. Effective plastic strain or material dependent variable
 - *. **Define NEIPS additional history values here for outer surface**
- If MAXINT >3 then define an additional (MAXINT-3)* (6*IOSHL(1) + 1*IOSHL(2) + 8*IOSHL(3) + 4*IOSHL(4) + NEIPS) quantities here

- 22. Bending moment-mx (local shell coordinate system)
- 23. Bending moment-my
- 24. Bending moment-mxy
- 25. Shear resultant-qx
- 26. Shear resultant-qy
- 27. Normal resultant-nx
- 28. Normal resultant-ny
- 29. Normal resultant-nxy
- 30. Thickness
- 31. Element dependent variable
- 32. Element dependent variable
- 33. Internal energy (if and only if ISTRN=0)

The following quantities are expected if and only if ISTRN=1

- 33. eps-x (inner surface strain in global system)
- 34. eps-y
- 35. eps-z
- 36. eps-xy
- 37. eps-yz
- 38. eps-zx
- 39. eps-x (outer surface strain in global system)
- 40. eps-y
- 41. eps-z
- 42. eps-xy
- 43. eps-yz
- 44. eps-zx
- 45. Internal energy (if and only if ISTRN=1)

****SKIP THE FOLLOWING IF THERE IS NO DATA FOR SPH ELEMENTS ****

TIME	1	Time word
SPHDATA	SPHV	Data for each sph element according to the sph flags SPHV=NSTP*NUM_SPH_DATA

INTERFACE FORCE DATABASE

There are three sections in the interface force database. The first contains 64 words of control information. The second contains geometric information, i.e. the nodal coordinates and segment connectivities for each segment contained in the master and slave surface definitions. The third section contains the results of the analysis at sequential output intervals. The output at a given time is called a state. The state contains a time word, global variables such as total energies and momenta, nodal data consisting of accelerations, velocities, and displacements, and finally segment data is written that include the pressure and shear stress acting on each segment and nodal forces for each node that defines the segment. The control information that follows provides information as to what is in the file and which database is being processed.

CONTROL DATA

VALUE	#WORDS	DISK ADDRESS	DESCRIPTION
Title	10	0	Problem identification
Run time	1	10	time in seconds since 00:00:00 UTC, January 1, 1970
File type	1	11	intfor=4 1=d3plot, 2=d3drif, 3=d3thdt, 4=intfor, 5=d3part 6=blstfor, 7=d3cpm, 8=d3ale, 11=d3eigv, 12=d3mode, 13=d3iter, 21=d3ssd, 22=d3spcm, 23=d3psd, 24=d3rms, 25=d3ftg
Source version	1	12	ls-dyna version *1000000 + svn number
Release number	1	13	Release number in character*4 form 50 for R5.0 511c for R5.1.1c
Version	1	14	Code version
NDIM	1	15	Insert 4 for LS-DYNA/3D database
NUMNP	1	16	Number of nodal points
ICODE	1	17	Insert 6 for LS-DYNA/3D database
NGLBV	1	18	Number of global variable to be read
BLANK	1	19	Insert zero
IU	1	20	Flag for current geometry (=1)

IV	1	21	Flag for velocities (default=1)
BLANK	1	22	Insert zero
BLANK	1	23	Insert zero
BLANK	1	24	Insert zero
BLANK	1	25	Insert zero
BLANK	1	26	Insert zero
BLANK	1	27	Insert zero
BLANK	1	28	Insert zero
BLANK	1	29	Insert zero
BLANK	1	30	Insert zero
NUMSG	1	31	Total number of slave and master segments in sliding interface definitions.
NUMMAT4	1	32	= 2 times the number of sliding interfaces.
NV2D	1	33	= 16, 17, 21 or 23 and see below values If NV2D is negative then the file is FSIFOR for an ALE model
BLANK	5	34	Unused space
NARBS	1	39	Additional storage required for arbitrary node and element numbering in type 6 database This number equals the sum of (10+ NUMNP+NEL8+NEL2+NEL4+ NELT)
BLANK	17	40	Unused space
NPRESU	1	57	1, 2, or 3 (default is 3)
NSHEAR	1	58	default is 1
NFORCE	1	59	default is 1
NGAPC	1	60	default is 1

If *DATABASE_EXTENT_BINARY is included in the model input the following values apply to the state output:

$$NV2D = \max(0, NPRESU) + \max(0, NSHEAR) * 3 + \max(0, NFORCE) * 12 + \max(0, NGAPC) * 5$$

NGLBV: output option for global variables

EQ,-1 no (NGLBV=0)

EQ,1 on

NVELO: output option for nodal velocities

EQ.-1 no (IV=0)
EQ.1 yes

NPRESU: output option for pressures

EQ.-1 no
EQ.1 output normal interface pressure only
EQ.2 output normal interface pressure and peak pressure
EQ.3 output normal interface pressure, peak pressure
and time to peak pressure

NSHEAR: output option for maximum interface shear stress,
shear stress in r-direction and s-direction

EQ.-1 no
EQ.1 yes

NFORCE: output option for X-, Y- and Z-force at all nodes

EQ.-1 no
EQ.1 yes

NGAPC: output option for contact gap at all nodes and surface energy density

EQ.-1 no
EQ.1 yes

GEOMETRY DATA

The geometry section contains the nodal coordinates and the element connectivities. The ordering of the nodal points is assumed to be the same as the ordering of the nodal data in the state data that follows.

VALUE	LENGTH	DESCRIPTION
X(3,1)	NDIM*NUMNP	Array of nodal coordinates X1,Y1,Z1, X2,Y2,Z2,X3,Y3,Z3, ... ,Xn,Yn,Zn
IX4(5,1)	5*NUMSG	Connectivity and identification number for each 3 or 4 node interface segment. For sliding interface n the identification number in 2n-1 for the slave surface and 2n for the master surface.

USER MATERIAL, NODE, AND ELEMENT IDENTIFICATION NUMBERS

Skip this section if NARBS (disk address 39) is zero. The user node and element numbers must be in ascending order. *It is assumed that if this option is used all node and element data anywhere in the databases is in ascending order based on user numbering.* The total length of the data in this database is equal to

$$\text{NARBS} = 10 + \text{NUMNP} + \text{NEL8} + \text{NEL2} + \text{NEL4} + \text{NELT}$$

if sequential numbering is used of the materials. For arbitrary material numbering the total length is increased by $6 + \text{NUMMAT8} + \text{NUMMAT4} + \text{NUMMAT2} + \text{NUMMATT}$. Material numbers are not in ascending order.

VALUE	LENGTH	DESCRIPTION
NSORT	1	Pointer to arbitrary node numbers in LS-DYNA source code, a negative sign flags that arbitrary material identification numbers are also used.
NSRH	1	Pointer to arbitrary solid element numbers in LS-DYNA source code: =NSORT+NUMNP
NSRB	1	Pointer to arbitrary beam element numbers in LS-DYNA source code: =NSRH+NEL8
NSRS	1	Pointer to arbitrary shell element numbers in LS-DYNA source code: =NSRB+NEL2
NSRT	1	Pointer to arbitrary thick shell element numbers in LS-DYNA source code: =NSRS+NEL4
NSORTD	1	Number of nodal points
NSRHD	1	Number of 8 node solid elements
NSRBD	1	Number of 2 node beam elements
NSRSD	1	Number of 4 node shell elements
NSRTD	1	Number of 8 node thick shell elements
NSRMA	1	Pointer to an array in the LS-DYNA source code that list the contact ID's in ascending order.
NSRMU	1	Pointer to an array in the LS-DYNA source code that gives the contact ID's in the actual order that they are defined in the user input.

VALUE	LENGTH	DESCRIPTION
NSRMP	1	Pointer to an array in the LS-DYNA source code that gives the location of a member in the array originating at NSRMU for each member in the array starting at NSRMA.
NSRTM	1	Total number of materials
NUMRBS	1	Total number of nodal rigid body constraint sets
NMMAT	1	Total number of materials
NUSERN	NSORTD	Array of user defined node numbers
NUSERH	NSORTH	Array of user defined solid element numbers
NUSERB	NSORTB	Array of user defined beam element numbers
NUSERS	NSORTS	Array of user defined shell element numbers
NUSERT	NSORTT	Array of user defined thick shell numbers
NORDER	NMMAT	Ordered array of user defined contact ID's
NSRMU	NMMAT	Unordered array of user contact ID's
NSRMP	NMMAT	Cross reference array

STATE DATA

The state data for the interface forces have three parts:

- Time word and global data
- Node data
- Force data for sliding interface segments

VALUE	LENGTH	DESCRIPTION
TIME	1	Time word
GLOBAL	NGLBV	Global variables for this state
NODEDATA	NND	Total nodal values for state where $NND=(IT+NDIM*(IU+IV))*NUMNP$ LS-DYNA/3D writes 6 values per node, i.e., the three coordinates and the translational velocities. The Data is put into the database as two vectors: first X(3,NUMNP) and then V(3,NUMNP), respectively.
SEGMDATA	ENN	Data for sliding interface segments where the quantity $ENN = (16,17,21, \text{ or } 23)*NV2D$. The organization of the segment data is described below.

This state data is repeated for each state in the database.

For each sliding interface segment the database contains $NV2D=16, 17, 21$ or 23 values per segment. The data order is:

1. Normal interface pressure acting on segment
2. Maximum interface shear stress acting on segment
3. Shear stress in local r-direction of segment
4. Shear stress in local s-direction of segment
5. X force at node n1 of segment
6. Y force at node n1 of segment
7. Z force at node n1 of segment
8. X force at node n2 of segment
9. Y force at node n2 of segment
10. Z force at node n2 of segment
11. X force at node n3 of segment
12. Y force at node n3 of segment

13. Z force at node n3 of segment
14. X force at node n4 of segment
15. Y force at node n4 of segment
16. Z force at node n4 of segment
17. contact gap at node n1
18. contact gap at node n2
19. contact gap at node n3
20. contact gap at node n4
21. surface energy density of segment
22. peak pressure
23. time to peak pressure

If contact gap at node n1 = -1.0, then no values are set. Similarly, if surface energy density is = -1.0, no value is set.

If contact gap is ≥ 999.0 ignore the value, this means that the interfaces are not in discernible contact.

Note: original interface force files have 16 variables, while recent ones, from ls-dyna ls970 and ls971, have up to 23. Check size of NV2D. If *database_extent_intfor is include in the model input then NV2D is according to the formula above.

FSIFOR file output:

1. Normal interface pressure acting on segment
2. X force on segment
3. Y force on segment
4. Z force on segment
5. relative interface velocity

BLSTFOR file (NV2D=16 or 7) output:

1. effective (combined incident and reflected) pressure applied to the segment
2. reflected wave (relevant only for BLAST=4)
 - = -1: segment is below ground level and not exposed to blast
 - = 0 : segment has not been subjected to blast waves
 - = 1 : segment has been subjected to the initial incident wave
 - = 2 : segment has been subjected to the ground reflected wave
 - = 3 : segment resides in the Mach stem region (Since known by geometrical considerations this value is fixed at time $t=0$. Thus, any non-zero reflected pressure on this segment is due to the Mach wave.)

3. incident pressure
4. mass density of air
5. global x-velocity of blast wind
6. global y-velocity of blast wind
7. global z-velocity of blast wind

CRACK FILE (d3crck)

The crack file is created in LS-DYNA when the Winfrith Concrete material model is used for solid elements. This model allows up to three orthogonal crack planes to develop each with an origin at the center of the element. The plane is assumed to project to the surface of the solid, and can be represented by drawing a line on any of the six solid element faces where it emerges. This cutting line can be found by considering the intersection of each plane and each solid face.

The crack file is written as a Fortran unformatted binary file, and each record in the file has a start record mark and an end record mark, each of which is 4 bytes for a single precision run and 8 bytes for a double precision run. The data is output for a state and contains:

VALUE	#WORDS	DESCRIPTION
TIME	1	State Time word
NUMBER OF CRACKS	1	NC, Number of sets of crack data
CRACK DATA	16 * NC	Sets of data for each crack

Data for each crack contains: Element ID, Flag for each crack plane, Normal vector for each crack plane and width for each crack plane.

Crack plane flags are as follows:

0 = no crack, 1 = cracked, but no sustaining tensile load, 2 = cracked but closed up, and 3 = fully cracked with no tensile strength.

Shown below is some C coding from LS-PREPOST to illustrate how the crack data is used and the crack lines established. The important coding is picked out in bold type. Also, after the crack data was read in, the crack flags (3 words) were stored as bits in one word. NG and MAT are the Group number for elements and the Material ID we assign in LS-PREPOST.

```

/* elm id, 3 2bit flags 0 -> 2, group no., mat no. */
int id, pflag, ng, mat;
float abc[9]; /* plane normals */
float cp[3]; /* crack width planes 1, 2 and 3 */

/* check brick element face against crack plane (nx,ny,nz) positioned
 * at element center (xe,ye,ze)
 */
int CrackPlane(float xe, float ye, float ze, float nx, float ny, float nz,
float xyz[][3], float xs[2], float ys[2], float zs[2])
{
    int i, k;
    float x0, y0, z0, x1, y1, z1;

```

```
float xp1, yp1, zp1, xp2, yp2, zp2, pn, dn, t;

k = 0;
x0 = xyz[3][0];
y0 = xyz[3][1];
z0 = xyz[3][2];
for (i=0; i<4; i++) {
    /* parametric line clip algorithm */
    xp1 = x0 - xe;
    yp1 = y0 - ye;
    zp1 = z0 - ze;
    x1 = xyz[i][0];
    y1 = xyz[i][1];
    z1 = xyz[i][2];
    xp2 = x1 - x0;
    yp2 = y1 - y0;
    zp2 = z1 - z0;
    x0 = x1;
    y0 = y1;
    z0 = z1;
    pn = nx*xp1 + ny*yp1 + nz*zp1;
    dn = nx*xp2 + ny*yp2 + nz*zp2;
    if (fabs(dn) < 1.0e-15) continue;
    t = -pn / dn;
    if (t < 0.0 || t > 1.0) continue;
    t = t - 1.0;
    xs[k] = x1 + t * xp2;
    ys[k] = y1 + t * yp2;
    zs[k] = z1 + t * zp2;
    k++;
    if (k > 1) break;
}
return k;
}
```

void SetCrackWidth(float v)

```
{
    min_crack_width = MAX(0.0, v);
}
```

void DrawCracks(int ist, float *bg_color)

```
{
    int i, k, m, n, nc, nd, kd, ip, ic, is;
    int id, facecode;
    int etype, nface, pflag, flag;
    unsigned int j;
    float xc, yc, zc, a, b, c;
    float xyz[24][3];
    float xi[2], yi[2], zi[2];
    int shrink, count, ns[2];
    float dx, dy, dz, ds, d;
    NDCOOR *nod;
    int ng=0;
```

```

float rd, gn, bu;

nod = node;
nod--;

rd = 1.0 - bg_color[0];
gn = 1.0 - bg_color[1];
bu = 1.0 - bg_color[2];
glDisable(GL_LIGHTING);
glColor3f(rd, gn, bu);
glLineWidth(2.0);
GetCrackData(ist);
glBegin(GL_LINES);
nc = cstate[ist].nc;
for (n=0; n<nc; n++) {
    pflag = crack[n].pflag;
    if (pflag == 0) continue;
    id = crack[n].id - 1;
    k = solid[id].mat & MASKBIT22;
    j = active_list[k].loc;
    facecode = (active_list[k].akey>>2) & FACEBITS1;
    if (j < BIT30 && facecode > 0) {
        id = j;
        ng = crack[n].ng;
        if (!part[ng].active) continue;
        dx = part[ng].dscale[0];
        dy = part[ng].dscale[1];
        dz = part[ng].dscale[2];
        ds = part[ng].dscale[3];
        shrink = part[ng].shrink_mode;
        etype = (active_list[k].akey) & 0x3;
        nface = FACE_NF[etype];
        xc = yc = zc = 0.0; ic = 0;

        for (i=0; i<nface; i++) {
            for (m=0; m<4; m++) {
                nd = FACE_P[etype][i][m];
                kd = solid[id].conn[nd];
                k = 4 * i + m;
                if (ds == 0.0) {
                    xyz[k][0] = (disp_state+kd)->xyz[0];
                    xyz[k][1] = (disp_state+kd)->xyz[1];
                    xyz[k][2] = (disp_state+kd)->xyz[2];
                }
                else {
                    GetScaledNodalCoord(kd, nod, disp_state, dx,dy,dz, xyz[k]);
                }
                xc += xyz[k][0];
                yc += xyz[k][1];
                zc += xyz[k][2];
                ic += 1;
            }
        }
    }
}

```

```

    if (ic == 0) continue;
    d = 1.0 / (float)ic; xc *= d; yc *= d; zc *= d;
    if (shrink) {
        for (i=0; i<24; i++) {
            xyz[i][0] = xc + (xyz[i][0] - xc) * shrink_factor;
            xyz[i][1] = yc + (xyz[i][1] - yc) * shrink_factor;
            xyz[i][2] = zc + (xyz[i][2] - zc) * shrink_factor;
        }
    }
    for (i=0; i<nface; i++) {
        if (shrink || (facecode & FACE_CODE1[i])) {
            k = 4 * i;
            /* for each active crack plane */
            for (ip=0, is=0, ic=0; ip<3; ip++, is+=2, ic+=3) {
                flag = (pflag >> is) & 3;
                if (flag == 0) continue;
                if ((min_crack_width < 0.5 && crack[n].cp[ip] >= min_crack_width)
                    || (min_crack_width >= 0.5 && flag == 3)) {
                    a = crack[n].abc[ic];
                    b = crack[n].abc[ic+1];
                    c = crack[n].abc[ic+2];
                    count = CrackPlane(xc, yc, zc, a, b, c, &xyz[k], xi, yi, zi);
                    if (count > 1) {
                        glVertex3f(xi[0], yi[0], zi[0]);
                        glVertex3f(xi[1], yi[1], zi[1]);
                    }
                }
            }
        }
    }
    glEnd();
    glLineWidth(1.0);
}

```

DYNAIN BINARY FILE FORMAT (dynain.bin)

/* Discription of Dynain binary format:

* In first 100 words (integers)

* head[0] = location of nodal data

* head[1] = number of nodes

* head[2] = location of solid element connectivities

* head[3] = number of solid elements

* head[4] = location of shell element connectivities + thicknesses

* head[5] = number of shell elements

* head[6] = location of adaptive constraints

* head[7] = number of adaptive constraints

* head[8] = location of initial stresses for solid elements

* head[9] = number of initial stress states defined for solids

* head[10] = location of initial stresses for shell elements

* head[11] = number of initial stress states defined for shells

* head[12] = location of initial strains for shell elements

* head[13] = number of initial strains states defined for shells

* head[14] = location of boundar spc's

* head[15] = number of boundary spc's

* head[16] = location of local coordinate systems by nodes

* head[17] = number of local coordinate systems by nodes

* head[18] = location of local coordinate systems by vector

* head[19] = number of local coordinate systems by vector

* head[20] = location of initial stress states for beams

* head[21] = number of initial stress states for beams

* head[22] = location of thick shell element connectivities

* head[23] = number of thick shell elements

* head[24] = location of initial stresses for thick shell elements

* head[25] = number of initial stress states defined for thick shells

* head[26] = location of beam element connectivities

* head[27] = number of beam elements

* head[28] = location of initial strains for solid elements

* head[29] = number of initial strain states defined for solids

*/