GETTING STARTED

DESCRIPTION OF KEYWORD INPUT

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

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

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

Alternatively, acceptable input could also be of the form:

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

```
$       DEFINE ONE MORE SHELL ELEMENT
$
*ELEMENT_SHELL
       10301          pid     n1      n2      n3      n4
```

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

Figure 2-1 attempts to show the general philosophy of the input organization and how various entities relate to each other. In this figure the data included for the keyword, *ELEMENT, is the element identifier, EID, the part identifier, PID, and the nodal points identifiers, the NID's, defining the element connectivity: N1, N2, N3, and N4. The nodal point identifiers are defined in the *NODE section where each NID should be defined just once. A part defined with the *PART keyword has a unique part identifier, PID, a section identifier, SID, a material or constitutive model identifier, MID, an equation of state identifier, EOSID, and the hourglass control identifier, HGID. The *SECTION keyword defines the section identifier, SID, where a section has an element formulation specified, a shear factor, SHRF, a numerical integration rule, NIP, and so on. The constitutive constants are defined in the *MAT section where constitutive data is defined for all element types including solids, beams, shells, thick shells, seat belts, springs, and dampers. Equations of state, which are used only with certain *MAT materials for solid elements, are defined in the *EOS section. Since many elements in LS-DYNA use uniformly reduced numerical integration, zero energy deformation modes may develop. These modes are controlled numerically by either an artificial stiffness or viscosity which resists the formation of these undesirable modes. The hourglass control can optionally be user specified using the input in the *HOURGLASS section.

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

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

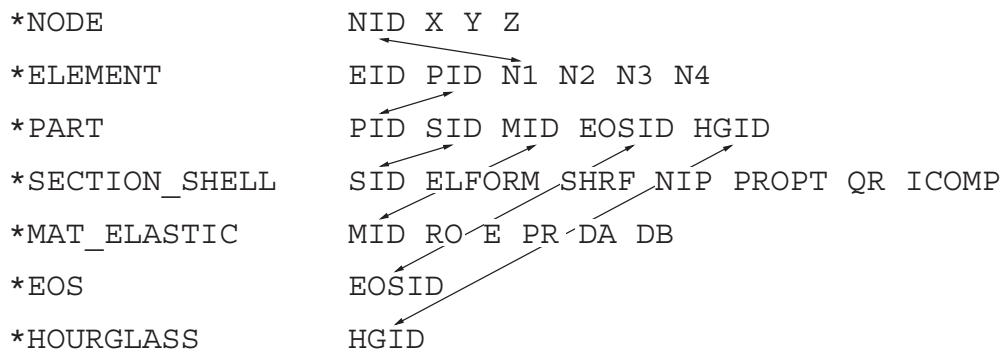

Figure 2-1. Organization of the keyword input.

```
*NODE
10101,x ,y ,z
10201,x ,y ,z

*ELEMENT_SHELL
10201,pid,n1,n2,n3,n4
10301,pid,n1,n2,n3,n4
```

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

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

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

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

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

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

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

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

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

STRUCTURED INPUT TYPE ID	KEYWORD NAME
1	SLIDING_ONLY
p 1	SLIDING_ONLY_PENALTY
2	TIED_SURFACE_TO_SURFACE
3	SURFACE_TO_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE
4	SINGLE_SURFACE
5	NODES_TO_SURFACE
a 5	AUTOMATIC_NODES_TO_SURFACE
6	TIED_NODES_TO_SURFACE
7	TIED_SHELL_EDGE_TO_SURFACE
8	TIEBREAK_NODES_TO_SURFACE
9	TIEBREAK_SURFACE_TO_SURFACE
10	ONE_WAY_SURFACE_TO_SURFACE
a 10	AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
13	AUTOMATIC_SINGLE_SURFACE
a 13	AIRBAG_SINGLE_SURFACE

STRUCTURED INPUT TYPE ID	KEYWORD NAME
14	ERODING_SURFACE_TO_SURFACE
15	ERODING_SINGLE_SURFACE
16	ERODING_NODES_TO_SURFACE
17	CONSTRAINT_SURFACE_TO_SURFACE
18	CONSTRAINT_NODES_TO_SURFACE
19	RIGID_BODY_TWO_WAY_TO_RIGID_BODY
20	RIGID_NODES_TO_RIGID_BODY
21	RIGID_BODY_ONE_WAY_TO_RIGID_BODY
22	SINGLE_EDGE
23	DRAWBEAD

The *CONTACT_ENTITY section treats contact between a rigid surface, usually defined as an analytical surface, and a deformable structure. Applications of this type of contact exist in the metal forming area where the punch and die surface geometries can be input as VDA surfaces which are treated as rigid. Another application is treating contact between rigid body occupant dummy hyper-ellipsoids and deformable structures such as airbags and instrument panels. This option is particularly valuable in coupling with the rigid body occupant modeling codes MADYMO and CAL3D. The *CONTACT_1D is for modeling rebars in concrete structure.

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

***DAMPING**
Defines damping either globally or by part identifier.

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

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

***DEFORMABLE_TO_RIGID**
This section allows the user to switch parts that are defined as deformable to rigid at the start of the analysis. This capability provides a cost efficient method for simulating events such as rollover events. While the vehicle is rotating the computation cost can be reduced significantly by

switching deformable parts that are not expected to deform to rigid parts. Just before the vehicle comes in contact with ground, the analysis can be stopped and restarted with the part switched back to deformable.

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

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

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

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

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

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

***INTEGRATION**
In this section the user defined integration rules for beam and shell elements are specified. IRID refers to integration rule number IRID on *SECTION_BEAM and *SECTION_SHELL cards

respectively. Quadrature rules in the *SECTION_SHELL and *SECTION_BEAM cards need to be specified as a negative number. The absolute value of the negative number refers to user defined integration rule number. Positive rule numbers refer to the built in quadrature rules within LS-DYNA.

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

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

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

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

***NODE**
Define nodal point identifiers and their coordinates.

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

***PART**
This keyword serves two purposes.

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

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

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

***RIGIDWALL**
Rigid wall definitions have been divided into two separate sections, _PLANAR and _GEOMETRIC. Planar walls can be either stationary or moving in translational motion with mass and initial velocity. The planar wall can be either finite or infinite. Geometric walls can be planar as well as have the geometric shapes such as rectangular prism, cylindrical prism and sphere. By default, these walls are stationary unless the option MOTION is invoked for either prescribed translational velocity or displacement. Unlike the planar walls, the motion of the geometric wall is governed by a load curve. Multiple geometric walls can be defined to model combinations of geometric shapes available. For example, a wall defined with the _CYLINDER option can be combined with two walls defined with the _SPHERICAL option to model hemispherical surface caps on the two ends of a cylinder. Contact entities are also analytical surfaces but have the significant advantage that the motion can be influenced by the contact to other bodies, or prescribed with six full degrees-of-freedom.

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

***SENSOR**
This keyword provides a convenient way of activating and deactivating boundary conditions, airbags, discrete elements, joints, contact, rigid walls, single point constraints, and constrained

nodes. The sensor capability is new in the second release of version 971 and will evolve in later releases to encompass many more LS-DYNA capabilities and replace some of the existing capabilities such as the airbag sensor logic.

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

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

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

***TITLE**
In this section a title for the analysis is defined.

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

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

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

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

SUMMARY OF COMMONLY USED OPTIONS

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

Topic	Component	Keyword
Geometry	Nodes	*NODE
	Elements	*ELEMENT_BEAM
		*ELEMENT_SHELL
		*ELEMENT_SOLID
		*ELEMENT_TSHELL
	Discrete Elements	*ELEMENT_DISCRETE
		*ELEMENT_MASS
		ELEMENT_SEATBELT_*Option
Materials	Part (which is composed of Material and Section, equation of state and hourglass data)	*PART
	Material	
	Sections	*MAT_*Option*
		*SECTION_BEAM
		*SECTION_SHELL
		*SECTION_SOLID
	Discrete sections	*SECTION_TSHELL
		*SECTION_DISCRETE
	Equation of state	*SECTION_SEATBELT
	Hourglass	*EOS_*Option*
		*CONTROL_HOURGLASS
		*HOURGLASS
Contacts and Rigid walls	Defaults for contacts	*CONTROL_CONTACT
	Definition of contacts	*CONTACT_*Option*
	Definition of rigid walls	*RIGIDWALL_*Option*
Boundary Conditions & Loadings	Restraints	*NODE
		BOUNDARY_SPC_*Option
	Gravity (body) load	*LOAD_BODY_*Option*
	Point load	*LOAD_NODE_*Option*
	Pressure load	*LOAD_SEGMENT_*Option*
		LOAD_SHELL_*Option
	Thermal load	*LOAD_THERMAL_*Option*
	Load curves	*DEFINE_CURVE
Constraints and spot welds	Constrained nodes	*CONSTRAINED_NODE_SET
	Welds	*CONSTRAINED_GENERALIZED_WELD_*Option*
		*CONSTRAINED_SPOT_WELD
	Rivet	*CONSTRAINED_RIVET

Output Control	Defaults ASCII time history files Binary plot, time history and restart files Items in time history blocks Nodes for nodal reaction output	*CONTROL_OUTPUT *DATABASE_*Option* *DATABASE_BINARY_*Option* *DATABASE_HISTORY_*Option* *DATABASE_NODAL_FORCE_GROUP
Termination	Termination time Termination cycle CPU termination Degree of freedom	*CONTROL_TERMINATION *CONTROL_TERMINATION *CONTROL_CPU *TERMINATION_NODE

Table 2.1. Keywords for the most commonly used options.

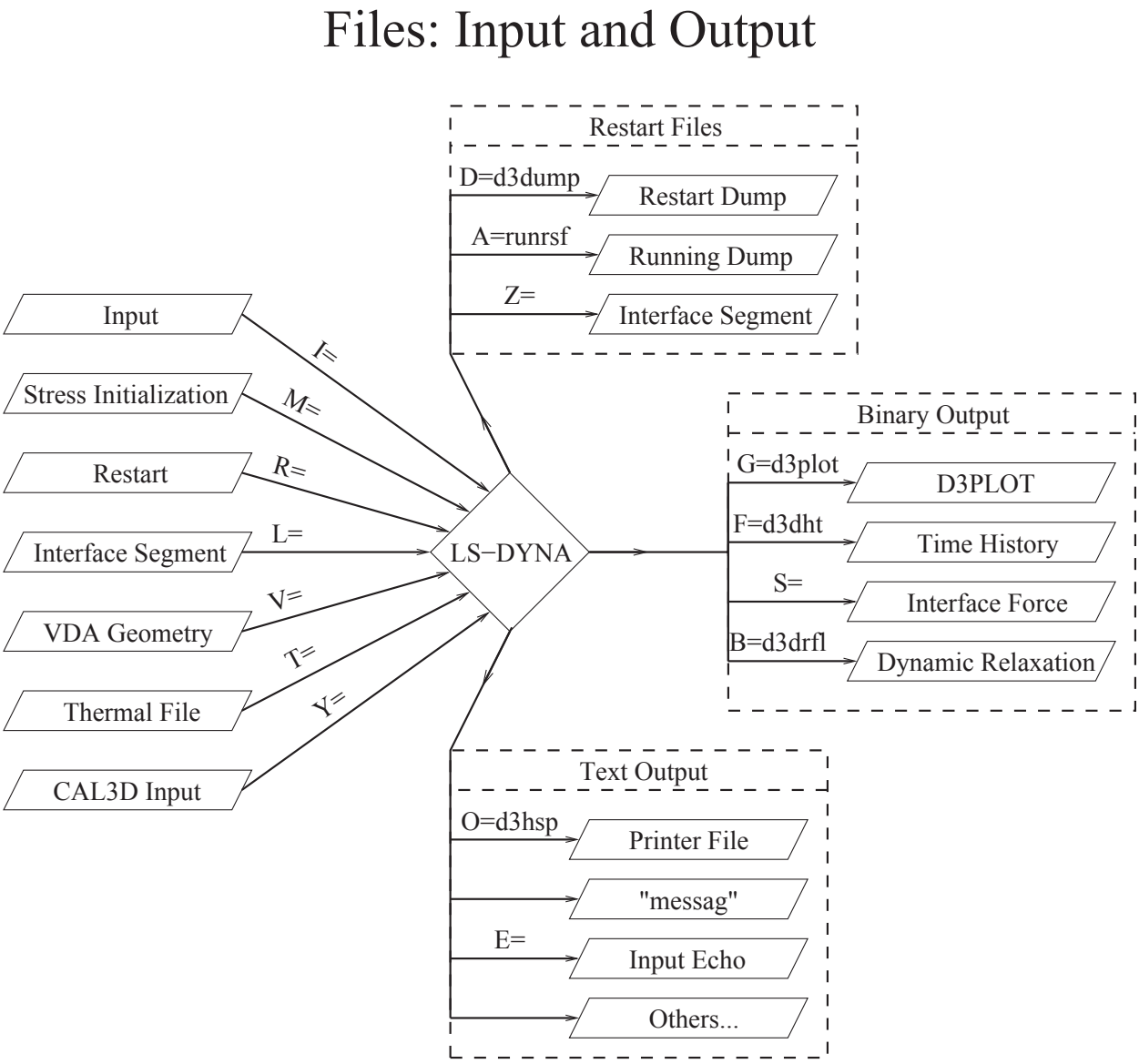

Figure 2-2. Files Input and Output.

EXECUTION SYNTAX

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

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

NCYCLE=**ncycle** JOBID=**jobid** D3PROP=**d3prop** GMINP=**gminp** GMOUT=**gmout** MCHECK=**y**

where

inf = input file (user specified)
otf = high speed printer file (default=D3HSP)
ptf = binary plot file for graphics (default=D3PLOT)
dpf = dump file for restarting (default=D3DUMP). This file is written at the end of every run and during the run as requested in the input. To stop the generation of this file set the file name to NODUMP.
thf = binary plot file for time histories of selected data (default=D3THDT)
tpf = optional temperature file
rrd = running restart dump file (default=RUNRSF)
sif = stress initialization file (user specified)
iff = interface force file (user specified)
isf1 = interface segment save file to be created (user specified)
isf2 = existing interface segment save file to be used (user specified)
rlf = binary plot file for dynamic relaxation (default=D3DRFL)
efl = echo file containing optional input echo with or without node/element data
root = root file name for general print option
scl = scale factor for binary file sizes (default=70 with the exception of LS-DYNA Version R7.0.0 for which the default=1024)
cpu = cumulative cpu time limit in seconds for the entire simulation, including all restarts, if **cpu** is positive. If **cpu** is negative, the absolute value of **cpu** is the cpu time limit in seconds for the first run and for each subsequent restart run.
kill = if LS-DYNA encounters this file name it will terminate with a restart file (default=D3KIL)
vda = VDA/IGES database for geometrical surfaces
c3d = CAL3D input file
bof = *FREQUENCY_DOMAIN_ACOUSTIC_BEM output file
nwds = Number of words to be allocated. On engineering workstations a word is usually 32bits. **This number overwrites the memory size specified on the *KEYWORD card at the beginning of the input deck.**
ncpu = Overrides **NCPU** and **CONST** defined in *CONTROL_PARALLEL. A positive value sets CONST=2 and a negative values sets CONST=1. See the *CONTROL_PARALLEL command for an explanation of these parameters. The *KEYWORD command provides an alternative way to set the number of CPUs.
para = Overrides **PARA** defined in *CONTROL_PARALLEL.
time = Overrides **ENDTIM** defined in *CONTROL_TERMINATION.
ncycle = Overrides **ENDCYC** defined in *CONTROL_TERMINATION.
jobid = Character string which acts as a prefix for all output files. Maximum length is 72 characters. **Do not** include the following characters:) (* / ? \.
d3prop = See *DATABASE_BINARY_D3PROP input parameter IFILE for options.
gminp = Input file for reading recorded motions in *INTERFACE_SSI (default=GMBIN).
gmout = Output file for writing recorded motions in *INTERFACE_SSI_AUX (default=GMBIN).

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

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

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

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

If the word **CASE** appears on the command line, then ***CASE** statements will be handled by the built in driver routines. Otherwise they should be processed by the external lscasedriver program, and if any ***CASE** statements are encountered it will cause an error.

If **MCHECK**=y is given on the command line, the program switches to "model check" mode. In this mode the program will run only 10 cycles – just enough to verify that the model will start. For implicit problems, all initialization is performed, but execution halts before the first cycle. If the network license is being used, the program will attempt to check out a license under the program name "LS-DYNAMC" so as not to use up one of the normal DYNA licenses. If this fails, a normal execution license will be used.

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

SENSE SWITCH CONTROLS

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

Type	Response
SW1.	A restart file is written and LS-DYNA terminates.
SW2.	LS-DYNA responds with time and cycle numbers.
SW3.	A restart file is written and LS-DYNA continues.
SW4.	A plot state is written and LS-DYNA continues.
SW5.	Enter interactive graphics phase and real time visualization.
SW7.	Turn off real time visualization.
SW8.	Interactive 2D rezoner for solid elements and real time visualization.
SW9.	Turn off real time visualization (for option SW8).
SWA.	Flush ASCII file buffers.
lprint	Enable/Disable printing of equation solver memory, cpu requirements.
nlprint	Enable/Disable printing of nonlinear equilibrium iteration information.
iter	Enable/Disable output of binary plot database "d3iter" showing mesh after each equilibrium iteration. Useful for debugging convergence problems.
conv	Temporarily override nonlinear convergence tolerances.
stop	Halt execution immediately, closing open files.

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

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

Procedure for LS-DYNA/MPP

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

rsh <machine name> kill -INT <PID>

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

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

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

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

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

where

rtf = restart filename.

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

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

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

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

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

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

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

and in the second by:

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

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

I=inf

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

I=inf R=rtf

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

RESTART ANALYSIS

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

The format of the restart input file is described in this manual. If, for example, the user wishes to restart the analysis from dump state *nn*, contained in file *D3DUMPnn*, then the following procedure is followed:

1. Create the restart input deck, if required, as described in the Restart Section of this manual. Call this file *restartinput.*

2. By invoking the execution line:

   LS-DYNA I=*restartinput* R=*D3DUMPnn*

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

   LS-DYNA R=*D3DUMPnn*

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

The R=D3DUMPnn on the status line informs the program that this is a restart analysis.

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

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

Assume an analysis is run using the input file, job1.inf, and a restart dump named d3dump01 is created. A new input file job2.inf is generated and submitted as a restart with R=d3dump01 as the dump file. The input file job2.inf contains the entire model in its original undeformed state but with more contact surfaces, new output databases, and so on. Since this is a restart job, information must be given to tell LS-DYNA which parts of the model should be initialized in the full deck restart. When the calculation begins the restart database contained in the file d3dump01 is read, and a new database is created to initialize the model in the input file, job2.inf. The data in file job2.inf is read and the LS-DYNA proceeds through the entire input deck and initialization. At the end of the initialization process, all the parts selected are initialized from the data saved from d3dump01. This means that the deformed position and velocities of the nodes on the elements of each part, and the stresses and strains in the elements (and, if the material of the part is rigid, the rigid body properties) will be assigned.

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

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

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

Points to note are:

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

VDA/IGES DATABASES

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

The advantage of using VDA surfaces is twofold. First, the problem of meshing the surface of the geometric entities is avoided and, second, smooth surfaces can be achieved which are very

important in metalforming. With smooth surfaces, artificial friction introduced by standard faceted meshes with corners and edges can be avoided. This is a big advantage in springback calculations.

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

LS-PrePost®

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

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

Some of the capabilities available in LS-PrePost are:

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

LS-PrePost processes output from LS-DYNA. LS-PrePost reads the binary plot-files generated by LS-DYNA and plots contours, fringes, time histories, and deformed shapes. Color contours

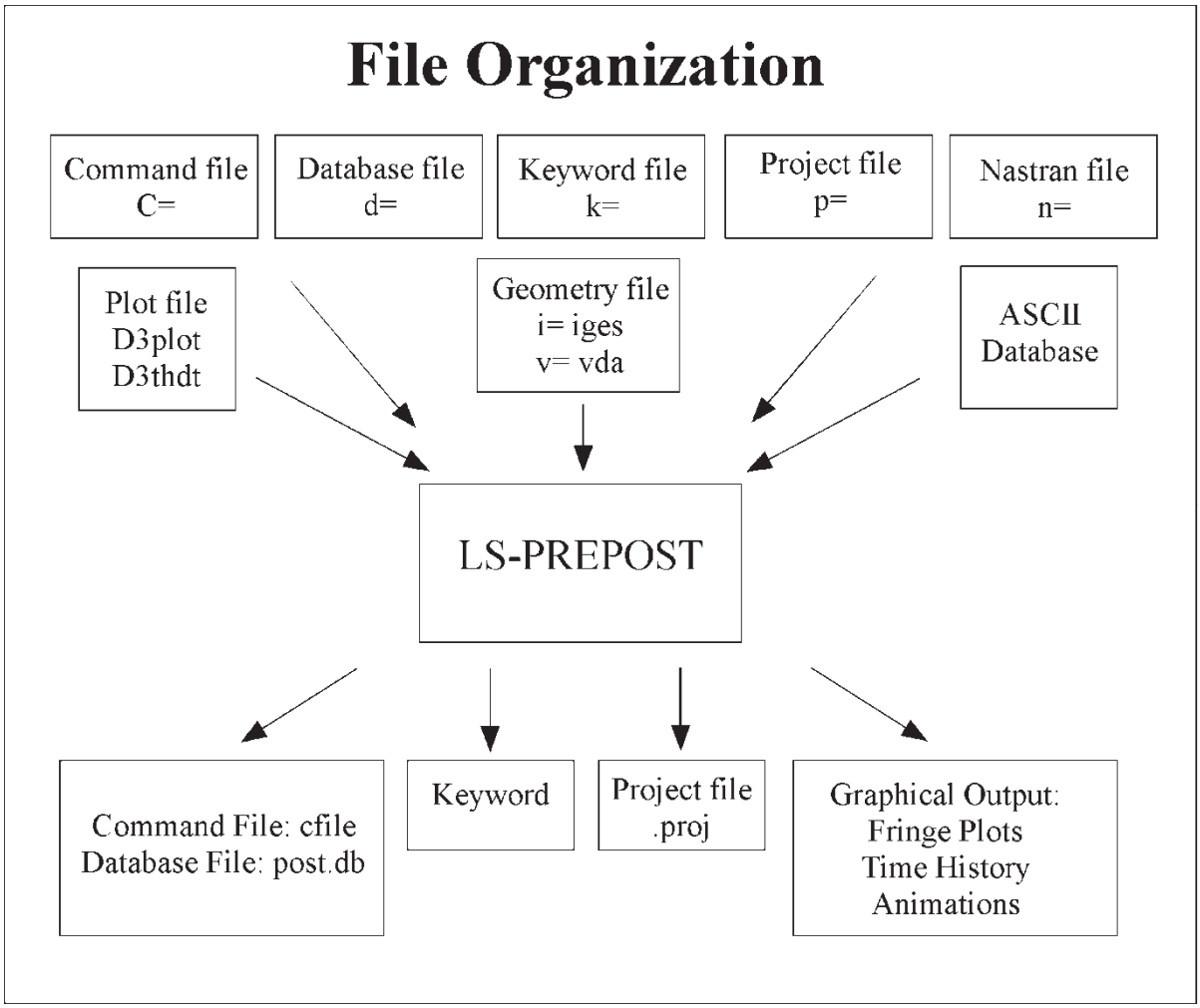

Figure 2-3. File Organization

and fringes of a large number of quantities may be interactively plotted on meshes consisting of plate, shell, and solid type elements. LS-PrePost can compute a variety of strain measures, reaction forces along constrained boundaries.

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

EXECUTION SPEEDS

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

Element Type	Relative Cost
8 node solid with 1 point integration and default hourglass control	4
as above but with Flanagan-Belytschko hourglass control	5
constant stress and Flanagan-Belytschko hourglass control, i.e., the Flanagan-Belytschko element	7
4 node Belytschko-Tsay shell with four thickness integration points	4
4 node Belytschko-Tsay shell with resultant plasticity	3
BCIZ triangular shell with four thickness integration points	7
C^0 triangular shell with four thickness integration points	4
2 node Hughes-Liu beam with four integration points	9
2 node Belytschko-Schwer beam	2
2 node simple truss elements	1
8 node solid-shell with four thickness integration points	11

These relative timings are very approximate. Each interface node of the sliding interfaces is roughly equivalent to one-half zone cycle in cost. Figure 2-4. illustrates the relative cost of the various shell formulations in LS-DYNA.

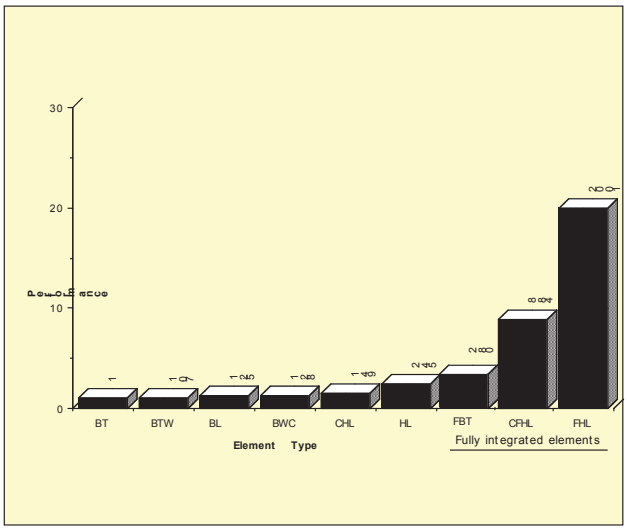

Figure 2-4. Relative cost of the four noded shells available in LS-DYNA where BT is the Belytschko-Tsay shell, BTW is the Belytschko-Tsay shell with the warping stiffness taken from the Belytschko-Wong-Chiang, BWC, shell. The BL shell is the Belytschko-Leviathan shell. CHL denotes the Hughes-Liu shell, HL, with one point quadrature and a co-rotational formulation. FBT is a Belytschko-Tsay like shell with full integration, FHL is the fully integrated Hughes-Liu shell, and the CFHL shell is its co-rotational version.

UNITS

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

1 (force unit) = 1 (mass unit) x 1 (acceleration unit)

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

Examples of sets of consistent units are:

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

Table 2.2.

GENERAL CARD FORMAT

The following sections specify for each keyword the cards that must be defined and those cards that are optional. Each card is described in its fixed format form and is shown as a number of fields in an 80 character string. With the exception of "long format" input as described below, most cards are 8 fields with a length of 10 and a sample card is shown below. The card format is clearly stated if it is other than eight fields of 10.

As an alternative to fixed format, a card may be free format with the values of the variables separated by commas. When using comma-delimited values on a card, the number of characters used to specify a value must not exceed the field length for fixed format. For example, an I8 number is limited to a number of 99999999 and larger numbers with more than eight characters are unacceptable. Fixed and free formats can be mixed throughout the deck and even within different cards of a single command but not within a card.

Card Format

	1	2	3	4	5	6	7	8
Variable	NSID	PSID	A1	A2	A3	KAT		
Type	I	I	F	F	F	I		
Default	none	none	1.0	1.0	0	1		
Remarks	1			2		3		

In the example shown above, the row labeled “Type” gives the variable type and is either F, for floating point or I, for an integer. The row labeled “Default” reveals the default value set for a variable if zero is specified, the field is left blank, or the card is not defined. The “Remarks” row refers to enumerated remarks at the end of the section.

Long Format Input:

To accommodate larger or more precise values for input variables than are allowed by the standard format input as described above, a “long format” input option is available. One way of invoking long format keyword input is by adding “long=y” to the execution line. A second way is to add “long=y” to the *KEYWORD command in the input deck.

long=y: read long keyword input deck; write long structured input deck.
long=s: read standard keyword input deck; write long structured input deck.
long=k: read long keyword input deck; write standard structured input deck.

The “long=s” option may be helpful in the rare event that the keyword input is of standard format but LS-DYNA reports an input error and the dyna.str file (see *CONTROL_STRUCTURED) reveals that one of more variables is incorrectly written to dyna.str as a series of asterisks due to inadequate field length(s) in dyna.str.

The “long=k” option really serves no practical purpose.

When long format is invoked for keyword input, input fields for each variable become 20 characters long with a maximum of four fields per line. In long format, only four variables can be defined per line, i.e., a “card” that requires eight variables and normally would fit on a single line must be spread over two lines of input.

You can mix long and standard format within one input deck by use of “+” or “-“ signs within the deck. If the execution line indicates standard format, you can add " +" at the end of any keywords to invoke long format just for those keywords. For example, “*NODE +” in place of

“*NODE” invokes a read format of two lines per node (I20,3E20.0 on the first line and 2F20.0 on the second line).

Similarly, if the execution line indicates long format, you can add "-" at the end of any keywords to invoke standard format for those keywords. For example, “*NODE –“ in place of “*NODE” invokes the standard read format of one line per node (I8,3E16.0,2F8.0).