LS-INGRID: A Pre-Processor And Three-Dimensional Mesh Generator For The Programs LS-DYNA, LS-NIKE3D And TOPAZ3D

Version 3.5

Livermore Software Technology Corporation
2876 Waverley Way
Livermore, CA 94550

August 1998

Mailing Address:

Livermore Software Technology Coporation
2876 Waverley Way
Livermore, California 94550-1740

Support Address:

Livermore Software Technology Corporation
97 Rickenbacker Circle
Livermore, California 94550-7612

FAX: 510-449-2507
TEL: 510-449-2500
EMAIL: sales@lstc.com

TABLE OF CONTENTS

LS-INGRID: A Pre-Processor and Three Dimensional Mesh Generator for the Programs LS-DYNA, LS-NIKE3D and TOPAZ3D.

ABSTRACT

LS-INGRID is a general purpose pre-processor for the programs, LS-NIKE2D[1], LS-NIKE3D[2], LS-DYNA2D[3], LS-DYNA3D[4], TOPAZ2D[5] and TOPAZ3D[6]. It can be used as a simple translator to convert various databases to these programs. In addition, it is a general purpose three-dimensional mesh generator with considerable capability to deal with complex geometries and allows for parametric geometric modeling.

PREFACE

LS-INGRID is an alternative mesh generator for finite element modeling which is principally intended as "research" program or one that focuses on various capabilities and techniques which are not addressed by commercial mesh generators. As a general purpose mesh generator, the capabilities are fairly complete with a wide range of geometric capabilities. An extensive parametric modeling capability is also support. LS-INGRID is most effective in combination with NIKExx and DYNAxx. A considerable amount of effort has gone into making LS-INGRID support virtually every feature of these programs (an almost impossible task given the rate that LS-DYNA3D improves). Although the usage of LS-INGRID can seem somewhat combersome relative to more traditional mesh generation schemes, the productivity of users performing parametric modeling tasks with LS-INGRID can much higher in some cases.

Unlike most general purpose mesh generators, LS-INGRID was developed by the Author for the sole purpose of aiding them in their routine analysis tasks. The original code was developed to assist in the preparation of SAP5 models at the University of Tennessee beginning in 1978. The 1978 program was loosely based on index space ideas obtained from the INGEN[7] program which was developed at Los Alamost National Laboratory. In 1981, the author moved to Lawrence Livermore National Laboratory. INGRID developments continued at LLNL on the side because LLNL was committed to the development of MAZE3D, but did not have any supported three-dimensional mesh generator. In 1985, the MAZE3D effort was finally abandoned and INGRID became the principal mesh generator of LLNL by default. At this time, the program was merged with the MAZE[8] program to produce a version similar to the current LS-INGRID.

After 1985, development work continued at SPARTA with a steady evolution and the

addition of the parametric modeling capabilities. LS-INGRID is the latest version supported by LSTC. Development on LS-INGRID is continuing in the directions which proved most popular in the past. The emphasis will continue to be providing a general purpose capability focused on NIKExx and DYNAxx with much work being done to support advanced modeling capabilities which are not found in any other program.

1. LS-INGRID BASICS

The LS-INGRID input file is an ASCII datafile which contains a complete description of the analysis. The commands are input using a parser which is simple and efficient, but also has a considerable amount of flexibility for dealing with complex situation.

1.1 THE PARSER

The parser basically takes a stream of blank delimited character strings and number and decodes them for the program. The character strings are for commands or parameters and are arbitrary in length. Normally, only the first four characters are significant. Deviations from this rule are described in the documentation. Numbers can be input in a variety of formats ranging from simple integers to floating point numbers specified with an "*E*" format. If an error is detected in the decoding of a number, the user will be notified.

All character input for commands or numbers is automatically converted to lower case for processing. Thus case selection can be performed strictly for the purpose of enhancing readability. Any number of commands and numbers can be placed on a single line of input with the only constraint being the 80 character input line limit. In the commands description, upper case characters or characters enclosed within quotes are commands which are to be typed exactly as lower (ignoring case). Lower case items represent variables which require input.

Comments may be included by using a "*c*" anywhere in the input followed by a blank and the comment. If the comment does not begin in column 1, then the "*c*" must be preceded by a blank. Blocks of input lines can be commented by preceding the block with the character "{" and ending it with "}".

Although items are normally blank delimited, commas can also be used to separate items. Two commas which are separated by blanks are treated as having the number 0 between them. Lists of numbers or character strings are input and terminated normally by a ";". This ";" does not necessarily need a blank between it and the last item. If the list is a list of numbers, then the list can be terminated by simply beginning the next command and eliminating the semicolon.

A function calculator is also built into the parser to permit advanced programming techniques to be used. This calculator is invoked by placing the calculator command within two square brackets. If the parser is expecting a character string, then the function will be processed without any other effect on the command stream. If a number is expected, then the calculator will send whatever value it calculates to the program. See Section 1.3 for a detailed description of the calculator functions.

1.2 COMMAND FILE FORMAT

The LS-INGRID input file "ingridi" has a relatively free flowing input format with few restrictions, some of which are:

- Define an item before using it; e.g. a line definition must occur before applying it to a part.
- Materials data and code execution options cannot be input until a code output option has been selected.
- Some commands have order dependent effects, e.g. rotating local coordinate systems successively about different axes.
- Many items which have names in LS-INGRID are assigned numbers for the analysis program. These numbers are assigned sequentially starting from one based on the order of first occurrence of names.

The form of ingridi is as follows:

Title line	(format is 80al)
Control commands	(Section 2)
Part definition	(Section 3 - 8)
Control commands	
Part definition	
.	
.	
.	
END	

1.3 The Calculator

The calculator is used to insert expressions into LS-INGRID input descriptions and is particularly useful for developing parametric models. When used in conjunction with the "include" command, it is possible to write programs for individual parts which can then be assembled into larger models. The calculator capabilities are invoked by inserting an expression anywhere in the input between two square brackets (e.g. [5*sin(30)]). If at that point in the input, LS-INGRID is expecting an integer or a floating point number, then the expression is evaluated and the results passed to LS-INGRID as either the nearest integer or floating point number. If a character string is expected, then the expression is evaluated and skipped over as if it were just a comment.

Within the calculator variables may be created and they will remain in effect until the program completes. Thus the expression [length=5*5] would store 25.0 into a variable named "length" and return 25.0 to LS-INGRID if a number is expected. This variable could be recalled later by [length].

Separate from the variable capability is a function capability. The function capability stores an expression which may consist of variables and other functions into a particular name for future evaluations. An example follows:

[*a*=1] [*b*=2] [*c*=1]	Set some variables so this won't evaluate improperly.
[def root1(*a*,*b*,*c*)=(-*b*+sqrt(*b***b*-4**a***c*))/2**a*]	Define the function.
[root1(2,1,0)]	Evaluate the function.

The general form of the calculator's capabilities is as follows:

[option name = expression]

Following is a summary of the calculator capabilities:

1.4 BUILT IN VARIABLES

Variable	Value
pi	p
e	e
#	Result of last operation
nnode	The current node number is set to nnode when outside of a part. Until the first part is complete, nnode is zero.
nbeam	The current beam element number is set to nbeam outside of a part. Until the first part is complete, nbeam is zero.
nbrick	The current brick element number is set to nbrick outside of a part. Until the first part is complete, nbrick is zero. nbrick includes both regular bricks and 8-node shell elements.
nshell	The current shell element number is set to nshell outside of a part. Until the first part is complete, nshell is zero.
npart	This variable is set inside parts and is set to the current part number.

1.5 BASIC ARITHMETIC OPTIONS

Operator	Purpose	Example
+	Addition	3+4
-	Subtraction	4-1
*	Multiplication	5.5*7.6
/	Division	7.5/length
^	Exponentiation	10^3
%	Modulo arithmetic	5%2

1.6 LOGICAL OPERATIONS

The result of a logical operator is 1.0 if true and 0.0 if false. These may be used either as expressions or as part of "if-then-else-endif" constructs.

Function	Purpose
==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
&&	Logical and
||	Logical or
!	Negation

1.7 FUNCTIONS

The angles in the following trigonometric functions are all in radians by default. This can be controlled by the "deg" and "rad" options listed in the "options" section below.

Function	Purpose
sin(angle)	Trigonometric sine.
cos(angle)	Trigonometric cosine.
tan(angle)	Trigonometric tangent.
asin(*x*)	Inverse trigonometric sine.
acos(*x*)	Inverse of trigonometric cosine.
atan(*x*)	Inverse of trigonometric tangent.
atan2(*y*,*x*)	Two argument inverse tangent.
sinh(*x*)	Hyperbolic sine.
cosh(*x*)	Hyperbolic cosine.
tanh(*x*)	Hyperbolic tangent.
exp(*x*)	Exponential.

ln(x)	Natural logarithm.
ln2(x)	Logarithm base 2.
log(x)	Logarithm base 10.
min(x_1,x_2, . . .)	Minimum of arbitrary number of parameters.
max(x_1,x_2, . . .)	Maximum of arbitrary number of parameters.
gcd(x_1,x_2, . . .)	Greatest common denominator.
lcm(x_1,x_2, . . .)	Least common multiple.
asa(angle,side,angle)	Evaluate the triangle and return largest angle.
ass(angle,side,side)	Evaluate the triangle and return largest angle.
sas(side,angle,side)	Evaluate the triangle and return largest angle.
sss(side,side,side)	Evaluate the triangle and return largest angle.
rnd	Return a random number.
rnd2	Return a random number but do not update the seed.

1.8 OPTIONS

Function	Purpose
help	Print the help message.
help subject	Print help for any of the calculator functions or options.
def name=expression	Define a function "name". Any time name is encountered in future expressions, it will be recursively evaluated.
save filename	Save all variables to file "filename".
load filename	Load variables from file "filename".
quit	Exit calculator (this will shut down LS-INGRID).
rad	All angles for trigonometric functions are assumed to be defined in radians (default).
deg	All angles for trigonometric functions are assumed to be defined in degrees.
list	List current active variables.
flist	List current definitions of functions.
root(c_n, . . .,c_1,c_0)	Determine the roots of the nth degree polynomial with coefficients c_0 through c_n.
factor(x)	Factor x into prime coefficients.
integral(e_1,e_2,f,v)	Determine the integral of the function f with respect to the variable v. The limits are from e_1 to e_2 which may be expressions. A Romberg integration rule is used.
degree n	The degree of Romberg integration for the "integral" command is n (default=4). Simpson's rule corresponds to n=1 and the trapezoidal rule is n=0.
solve (f_1,f_2, . . .,f_n) for (x_1=v_1+d_1, . . . x_n=v_n+d_n)	Solve a system of nonlinear equations. The equations are previously defined functions, f_1 through f_n. Variables x_1 through x_n must be listed and the calculator will attempt to determine them. Optional inputs include v_i and d_i. The initial starting guess is v_i and the initial increments for iterations are d_i.

maxits *n*	Set the maximum number of iterations for the "solve" command to *n* (default=30).
tol *t*	Set the convergence tolerance for the "solve" command to *t* (default=1e-6).
display	The display command is a brute force method for improving results of a divergent solve command. Results are displayed after every iteration.
if expr1 then expr2 else expr3 endif	If expression 1 is true than evaluate expression 2. Otherwise, evaluate expression 3.

1.9 DIRECTIVES

LS-INGRID provides directives to control the flow of logic in command file descriptions. Directives begin in the first column of a line and no other commands are allowed on the same line as the directive. This capability is patterned similar to the pre-processor used in the C-programming language.

DIRECTIVE	FUNCTION
#ELSE	This is for #IF - #ELSE - #ENDIF constructs.
#ELSEIF *expression*	Perform conditional execution of the following input lines as part of an #IF - #ELSEIF - #ENDIF construct.The results of *expression* should be either true (1) or false (0).
#ENDIF	This signifies the end of an #IF - #ENDIF block.
#ENDMACRO	End definition of macro initiated by #MACRO.
#IF *expression*	Conditionally execute the following lines of input. The results of *expression* should be either true (1) or false (0).
#INCLUDE *name*	Begin execution of commands in file *name*. When a #RETURN or an end-of-file is encountered, control is returned to the original file.
#MACRO *name*	Begin definition of macro *name*. The definition ends when an #ENDMACRO is encountered.
#RETURN	Cease reading input from the current input file and return control back to the next higher level file. See also #INCLUDE.

2. Control Commands

Control commands are optional and can be input in any order. They must not be placed inside a part. The following control commands are available:

ARRI *lc*

Generate arrival times for pressure surfaces associated with load curve *lc*. Arrival times are generated by assuming that the loads are caused by a wave. This wave starts from a three-dimensional point, line, or surface and travels with a velocity. The arrival time is the time required for the wave to travel from the source to an individual pressure segment.

*Options***:**

CG *p*
Pressure cannot exceed *p*.

CL *p*
Pressure cannot be less than *p*. (This option is selected with *p*=0.0 when "COSI" is executed).

COSINE
The pressure varies as a function of the angle between the pressure segments normal and the direction of the wave from the source.

DECAY *d* type [*a*]
The pressure wave decays as a function of the distance from the source. The distance at which the scale factor for the input pressure equals 1.0 is *d*. The type of decay is specified by type.

type = "*R*": relationship is $1.0/R$
= "*R*2": relationship is $1.0/R^2$
= "*R*3": relationship is $1.0/R^3$
= "CONSTANT": no decay
= "EXP": relationship is $1.0/R^a$

LINE $p_x\ p_y\ p_z\ v_x\ v_y\ v_z$
The source is a line. $(p_x\ p_y,p_z)$ is any point on the line and (v_x,v_y,v_z) is any vector along the line.

PLANE $p_x\ p_y\ p_z\ v_x\ v_y\ v_z$
The source is a plane. (p_x,p_y,p_z) is any point on the plane and (v_x,v_y,v_z) is any vector normal to the plane.

POINT $p_x\ p_y\ p_z$
The source is a point located at $(p_x\ p_y,p_z)$.

TOFF *dt*
Add *dt* to the arrival time.

VELO vel
The wave travels with velocity vel.

;
Terminate this command.

BATCH	LS-INGRID is to operate in batch mode. The interactive commands are placed at the end of the LS-INGRID model description so that they can be read automatically. A graphics device will still be requested since one of the batch output devices may be desired. If no graphics are needed use the **NOPL** command.
BELT	This section defines the properties of seat belt systems, but possibly has other applications. The seat belt capability is supported in LS-920 and later. A detailed description is included in the LS-920 manual.
SLIPRING name	Define a slipring.

Slip rings provide for a contiuous feeding of material through a pully. One node for the slip ring is fixed to a support structure. The slip ring logic works with seatbelt elements. Two seatbelt elements must also be identified which touch the slip ring. The friction coefficient, f, determines the resisting force to the belt being pulled through the slipring.

FRIC *f*	Friction coefficient for material sliding through the slip ring.
;	End of slipring definition.
RETRACTOR name	Define a retractor.

Retractor elements simulate the normal function of retrator systems for seat belts within an automobile.

DELAY *dt*	Time delay for retractor operation.
FEDL feed_length	
LCL lcl	Load curve for loading.
LCU lcu	Load curve for unloading.
PULL pullout	Amount of pull-out between time delay ending and retractor locking.
SENSOR name	Sensor for trigering retractor. At least one must be specified and no more than four.
;	End of retractor definition.
PRETENSIONER name	Define pretensioner.

During an automobile accident, pretensioners are frequently employed to automatically increase the tension on a seatbelt. Both pyrotechnic and spring type systems are supported. Usually, a sensor triggers the event.

PYROTECHNIC	Use a pyrotechnic pretensioner.
LCP lcp	Load curve for pretensioner.
RETR name	Retractor name effected.
TIME *t*	Time between sensor triggering and pretensioner acting.

PRELOAD	The pretensioner consists of a preloaded spring.
DELAY *dt*	Time between sensor triggering and pretensioner acting.
SPRING ispd	Spring element number.
LOCK	The pretensioner consists of a lock spring which is removed.
DELAY *dt*	Time between sensor triggering and pretensioner acting.
SPRING ispd	Spring element number.
DISTANCE	The distance between nodes is locked.
SENSOR name	Pretensioner is activated by one to four sensors.
;	End of pretensioner definition.
SENSOR name	Define sensor.

A variety of sensor systems are incorporated into automobiles to sense the onset of a crash. The accelerometers are simply used for saving output to an ASCII file. The other sensors are used to initiate the retractors and pretensioners.

ACCE *a*	The sensor is an accelerometer.
X	The acceleration is measured in the *x*-direction.
Y	The acceleration is measured in the *y*-direction.
Z	The acceleration is measured in the *z*-direction.
TIME *dt*	The sensor is triggered if a is exceeded for duration *dt*.
RETR	The sensor triggers based on the retractor pullout rate.
RETR name	Retractor name.
RATE *r*	Pullout rate.
TIME *t*	Time over which rate of pull-out must be exceeded.
TIME *t*	The sensor triggers after time *t*.
DIST	The sensor triggers based on the distance between two nodes.
DMAX $d_{\max}$	Maximum distance.
DMIN $d_{\min}$	Minimum distance.
;	End of Sensor definition.
;	End of BELT command.
CNV *n*	Define control volume *n*. (MVMA/DYNA3D, LS-910 and later.)

Options:

DAMP *d*	Set airbag damping constant to *d*.
MATE $m_1 \ldots m_n$;	The airbag consists of material subset $m_1 \ldots m_n$.
PSCA psca	Pressure scale factor used for converting pressures calculated by the thermodynamic control volume to pressures which will be applied to the finite element model. (default=1.0)
REVERSE	Reverse normals.
TYPE *m*	Control volume is of type *m*. Input for type *m* control volume begins immediately.
VINI vini	Initial filled volume. (default = 0.0)
VSCA vsca	Scale factor for converting calculated volume to volume used for thermodynamic calculations. (default = 1.0)

Type 1:

The pressure-volume relationship is of the form:

Pressure = p_0 * *s*/*Relative volume*

P0 p_0	Initial pressure.
SCAL *s*	Scale factor.
;	Terminate control volume input.

Type 3:

The pressure-volume relationship is of the form:

$$\dot{m}_{out} = A\sqrt{2\rho_0\mu}\sqrt{\frac{\gamma}{\gamma-1}\left(Q^{2/\gamma} - Q^{(\gamma+1)/\gamma}\right)}$$

where

$Q = \f(p_e,p)$

$\gamma = \f(c_p,c_v)$

$p = \f((\gamma - 1)E,V)$

Options:

CV c_v	Heat capacity at constant volume.
CP c_p	Heat capacity at constant pressure.
TIN *t*	Input gas temperature.
LCM lcm	Load curve defining input mass flow rate.
MU m	Shape factor for exit area. If m is negative, then |m| is the number of a load curve which defines the shape factor as a function of pressure.

A *a*	Exit Area. If a is less than zero, then |*a*| is the number of a load curve which defines the area as a function of pressure.
PE p_e	Ambient pressure.
RHO r	Ambient density.
GRAV *g*	Gravitational constant. If the ambient density is defined in units of weight per volume, then the actual gravitational constant must be used. Otherwise, *g* is set to 1.
;	Terminate control volume input.

Type 4:

Type 4 applies a constant internal pressure scaled by *s* until a point in time. A load curve is used to cause a change in behavior at some point in time. When the change occurs, the volume of the control volume is first calculated and used to initialize an adiabatic gas relationship.

PINT p_{int}	Interior pressure.
LC *lc*	Load curve.
SCAL *s*	Scale factor for pressure.
PE p_e	Ambient pressure.
RHO r	Density of gas when initialized.
GAMM g	Ratio of specific heats.
;	Terminate control volume input.

Type 5:

Type 5 is an implementation of the Wang-Nefske airbag model.

CV c_v	Heat capacity at constant volume.
CP c_p	Heat capacity at constant pressure.
TIN *t*	Input gas temperature.
LCM lcm	Load curve defining input mass flow rate.
C23 *c*23	Shape factor for exit hole.
A23 *a*23	Exit hole area.
CP23 *c*23¢	Shape factor for exit porisity.
AP23 *a*23¢	Exit hole porosity.
PEXT p_e	Ambient pressure.
RHO r	Ambient density.
GRAV *g*	Gravitational constant.
VOLT *v*	Optional tank volume.
LCOUT *lc*	Optional load curve specifying exit flow as a function of pressure.
PINI p_0	Optional initial overpressure (gauge).
PPOP ppop	Optional pressure where a plug is assumed to pop and venting begins.
;	Terminate control volume input.

COOR nc data	Input nc global coordinate systems. Global coordinate systems remain in effect until reset using this command. Coordinate system data is

described in detail in the section on *Coordinate Transformations*.

CSCA *s* — Scale all nodal coordinates by *s*.

CSYM — Define cyclic symmetry interface.

Options:

AXIS p_x p_y p_z — The vector which orients the axis for rotational cyclic symmetry is (p_x,p_y,p_z).

; — Terminate the CSYM command.

DEFAULT dir — The default directory for finding include files is dir. Currently, this is only supported on CONVEX computers.

DETP mat — Define detonation point for material mat. If mat is zero then all materials are detonated.

Options:

LNPT p_x p_y p_z q_x q_y q_z n — Generate *n* equals spaced detonation points on the line from (p_x,p_y,p_z) to (q_x,q_y,q_z).

POINT p_x p_y p_z — Detonate the point (p_x,p_y,p_z).

TIME *t* — Lighting time for detonation point.

; — Terminate this command.

DN2D *i j* — Output generated is compatible with LS-DYNA2D *i* and *j* flags specifying which 3-D coordinates correspond to the LS-DYNA2D *r* and *z* coordinates. *i* and *j* can have values "*x*", "*y*", or "*z*". This activates additional commands which are described in *LS-DYNA2D Options and Materials*.

DN3D — Output is generated for LS-DYNA3D. This activates additional control commands which are described in *LS-DYNA3D Options and Materials*.

DS *n* — Input digitized 3-D surface number *n*. Digitized surfaces consist of a surface defined by triangles. This is not smooth for coarse meshes; however, 10,000 and more triangles are common in defining surfaces to achieve

reasonably accurate definitions. This command consist of "DS *n*" optionally followed by a coordinate transformation and then one of the digitized surface types is input to complete the command.

Option:

MOVE data — Move the surface definition by data. data is described in the section on *Coordinate Transformations*.

Type 1: **GRID** n_i n_j — The surface is defined by a logically regular set of points in three dimensions. n_i*n_j points must be input in the following order. x_{11} y_{11} z_{11} ... x_{i1} y_{i1} z_{i1} ... x_{ij} y_{ij} z_{ij}.

Type 2: **FEM** *m n* — The surface is a grid of finite element quadrilaterals. It has *m* nodes and *n* elements.

x_1 y_1 z_1 x_2 y_2 z_2 ... — Input *m* nodal points.

i_1 j_1 k_1 l_1 i_2 j_2 k_2 l_2 ... — Input *n* four node elements.

Type 3: **CONT** ncont — The surface is defined by ncont contours that each have an arbitrary number of points.

Options:

XLE *x* — *X*-coordinate of leading edge.

YLE *y* — *Y*-coordinate of leading edge.

ZLE *z* — *Z*-coordinate of leading edge.

CHORD scal — Scale factor for the chord length.

FNU n_u **FNL** n_1 — The contour is defined by n_u points on the upper surface followed by n_1 points on the lower surface.

XF x_f — Contour points are in the plane $X=x_f$.

YF y_f — Contour points are in the plane $Y=y_f$.

ZF z_f — Contour points are in the plane $Z=z_f$.

L3D *l* — Use three-dimensional line definition *l*. The number of points on the contour is the number of points used to define the line definition.

L3E *l n* — Use three-dimensional line definition *l* with *n* equal spaced points.

; — Terminate option and read required data.

Data: — Skip this section if L3D or L3E is requested.

n — Number of points on contour. (Input only if *nu*=*n*1=0).

x_1 y_1 z_1 ... x_n y_n z_n — Contour coordinates. If *XF* was used, do not input any *X* coordinates, and similarly for *YF* and *ZF*.

Type 4: **FUNC** n_s n_t *x y z*

Define a parametric surface using parameters *s* and *t*. The number of points for making a grid in the *s* direction is *ns* and the number of points in the *t* direction is *nt*. *s* and *t* are assumed to range from 0 to 1. *x*, *y* and *z* are input as functions of *s* and *t*.

Type 5: **PROJ** *m* offset

The current digitized surface is formed by taking digitized surface *m* and projecting *m* a distance "offset" in the normal direction.

Options:

XSYM xsym xtol

Digitized surface *m* is symmetric about *X*=xsym. This forces normal components of points within xtol of the symmetry plane to be in the *Y*-*Z* plane only.

YSYM ysym ytol

Digitized surface *m* is symmetric about *Y*=ysym. This forces normal components of points within ytol of the symmetry plane to be in the *X*-*Z* plane only.

ZSYM zsym ztol

Digitized surface *m* is symmetric about *Z*=zsym. This forces normal components of points within ztol of the symmetry plane to be in the *X*-*Y* plane only.

;

End digitized surface definition.

END

Terminate the model description.

FDEF *n* v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9

Define fold plane number *n*. Fold planes are used later in the interactive phase to generate folded models of meshes such as airbags. The nine parameters have the following meanings:

v_1 = *x* or *y* position of fold relative to the unfolded mesh.

v_2 = *x* or *y* position of fold relative to the folded mesh.

v_3 = postive / negative fold flag. If v_3=+1, then the folded portion lies along the positive coordinate. If v_3=-1, then the folded portion lies along the negative coordinate.

v_4 = fold thickness desired.

v_5 = Up / down flag. If v_5=+1, then the material is folded onto the top of the mesh. If v_5=-1, then the material is folded under the mesh.

v_6 = Direction. The fold plane is normal to the *X*-axis if v_6=0 and normal to the *Y*-axis if v_6=1.

v_7 = Fold logic flag. The thin fold logic is used if v_7 = 0 and the thick fold logic is used if v_7=1.

v_8 = Scale factor. This can be used to scale the normal thickness offsets at the fold point. This will increase the separation between layers, but possibly cause unacceptable mesh distortion.

v_9 = Fold radius. This will take precedence over the internally computed fold radius.

See also FOLD and PFOLD under *Interactive Commands.*

FIGN *f* expr — When performing fold definition *f*, ignore nodes with undeformed coordinates that result in expr being true. (e.g. fign 6 [*y*<=0]).

FLEX name — Begin definition of flexure-torsion or cardan joints (LS-920 and later.)

Options:

CARDAN — This joint is a cardan joint.

FLEXION — This joint is a flexion-torsion joint.

LC1 lc1 — First torque-twist load curve.

LC2 lc2 — Second torque-twist load curve.

LC3 lc3 — Third torque-twist load curve.

LC4 lc4 — Fourth torque-twist load curve.

LC5 lc5 — Fifth torque-twist load curve.

LC6 lc6 — Sixth torque-twist load curve.

MATM *m* — Define master material.

MATS *m* — Define slave material.

MSYS *s* — Define master side local system.

PHIF *s*	First angle friction.
PHIS *s*	First angle stiffness.
PSIF *s*	Third angle friction.
PSIS *s*	Third angle stiffness.
SSYS *s*	Define slave side local system.
STOPA- *s*	Negative stop for first angle.
STOPA+ *s*	Positive stop for first angle.
STOPB- *s*	Negative stop for second angle.
STOPB+ *s*	Positive stop for second angle.
STOPC- *s*	Negative stop for third angle.
STOPC+ *s*	Positive stop for third angle.
THEF *s*	Second angle friction.
THES *s*	Second angle stiffness.

FMOV *f* data — After performing fold definition *f*, affected nodes are moved by the transformation described in data (see *Coordinate Transformations*).

FOPT *f options* — Input additional parameters for airbag folding. .

Options:

L3D	Fold abour 3-D line denfinition *ldnum*.
ANGLEθ	Fold the material θ degrees.
SCALE scale	The folded section will become thicker by the factor *scale*.
;	End of FOPT command.

FSYM *m x y z* n_x n_y n_z s_f — Define failing symmetry plane m. (x,y,z) is any point on the plane and (n_x,n_y,n_z) is any normal vector. Solid element faces are slaved to the symmetry plane and failure occurs when the normal stress exceeds s_f.

GEOC igeo mat — Geometric contact entity definition. A geometric contact entity is an analytical surface type which can be attached to a rigid body of

material mat. (MVMA/DYNA3D, LS-910 and later.)

Options:

COUPLE type *n*	The contact is between a CAL3D/MADYMO coupled rigid body and a deformable body. The rigid body type is either "ELLIPSE" or "PLANE" and n is the number of the shape in either CAL3D or MADYMO.
FRIC *f*	Set friction coefficient to *f*.
INSIDE	
MATE $m_1 \ldots m_n$;	The slaved mesh is the material subset $m_1 \ldots m_n$.
MOVE *n*	Move the entity using the global transformation number *n*.
OUTSIDE	
PNLT *p*	Penalty *p*.
QUAD *q*	Quadrature rule. q=0: Nodes only. q=1: Element centers. q=2: 2«2 quadrature on segments.
SD *n*	Use surface definition *n*. Valid surface types include planes, ellipsoids and spheres.
;	Terminate this command.

GMI *n* — Increment the default material number by *n* for each global copy of a part. This number is initially set to zero.

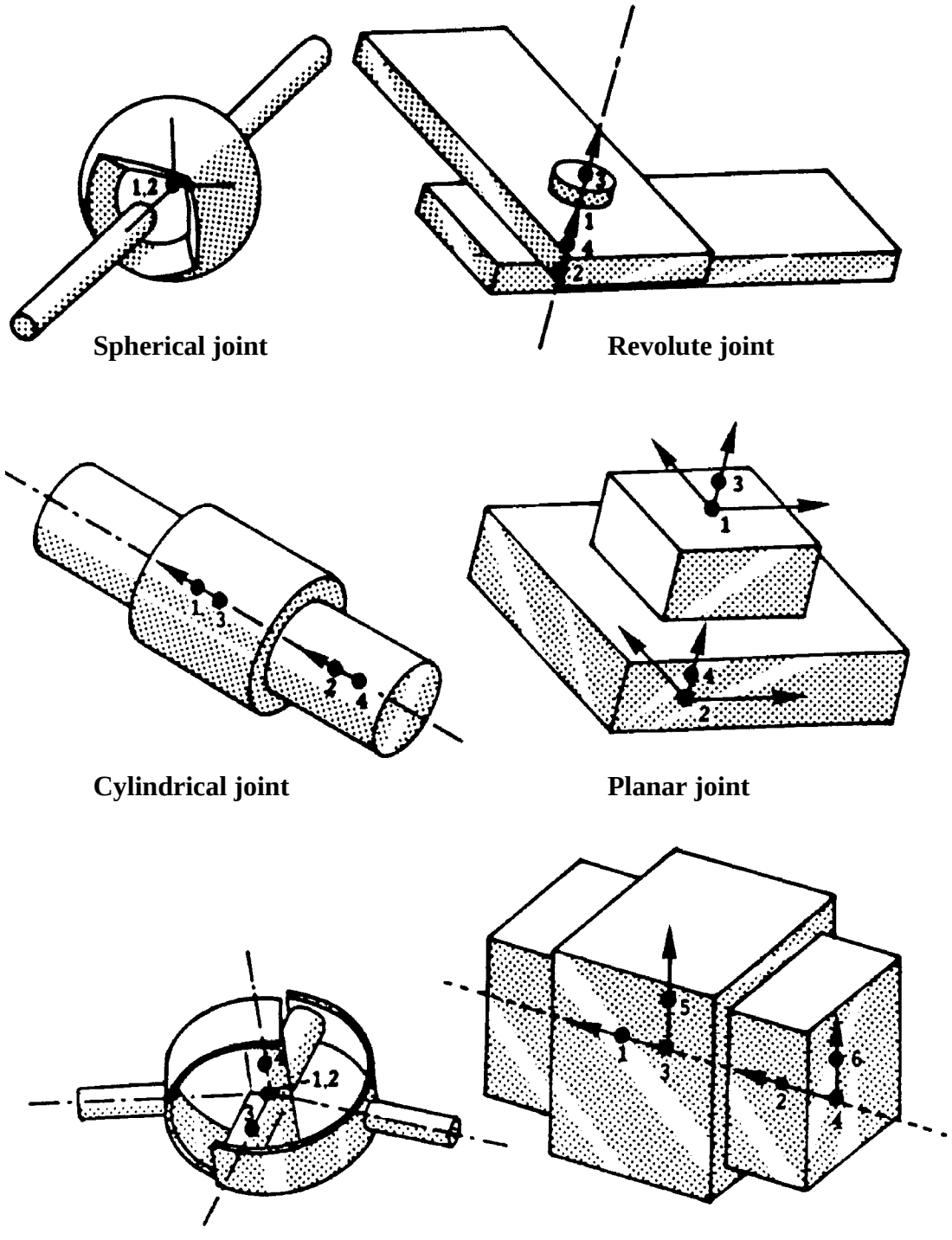
INCLUDE fname — Include the information in file fname in the command stream. The INCLUDE command can perform to 20 levels deep.

JD *j* — Begin joint definition for joint *j*. Diagrams of the types of joints are shown in Figure 2-1. Nodes are assigned to joint definitions within parts.

Options:

SJ	Spherical joint.
RJ	Revolute joint.
CJ	Cylindrical joint.
PJ	Planar joint.
UJ	Universal joint.
TJ	Translational joint.
PNLT *p*	Joint penalty.
NC icode	This joint is a simple nodal constraint. The common translational degrees of freedom are specified by icode: =-1: Rigid Massless Beam (LS-902 and later)

	= 0: Rigid Massless Truss (LS-902 and later) =1: *X* =2: *Y* =3: *Z* =4: *X* and *Y* =5: *Y* and *Z* =6: *Z* and *X* =7: *X*, *Y*, and *Z*
RC icode	This joint is a simple nodal constraint. The common rotational degrees of freedom are specified by icode: =0: none =1: *X* =2: *Y* =3: *Z* =4: *X* and *Y* =5: *Y* and *Z* =6: *Z* and *X* =7: *X*, *Y*, and *Z*
REPE *n*	Repeat the current joint definition for a total of *n* joints.
;	Terminate this command.



Spherical joint **Revolute joint**

Cylindrical joint **Planar joint**

Universal joint **Translational joint**

Figure 2-1. Joint definitions.

L3D *n* data

Begin definition of three-dimensional line *n*. If line *n* has been previously defined, this command has the effect of destroying the old definition. See *Three Dimensional Line Definitions* for a description of the data for this command.

LABELS

Define offsets for node, element and other item numbering. This applies to meshes which are imported after this command.

Options:

ELEMENT *m*

Offset element labels by m.

MAT *m*

Offset material labels by m.

NODE *m*

Offset node labels by m.

;

End of LABELS command.

LCD *n m* t_1 f_1 ... t_m f_m

Define load curve *n* with *m* pairs of time function points.

LCDF *n m f* t_1 t_n

Define load curve *n* with *m* pairs of time function points. *f* is input as an analytical function of variable *t* which ranges from t_1 to t_n. Thus to input one period of a sine wave with 100 points:

LCDF 1 100 [sin(t)] [0] [2*pi]

LD *n* data

Begin definition of two-dimensional line *n*. If line *n* has been previously defined, this command has the effect of destroying the old definition. See *Two Dimensional Line Definitions* for a description of the data for this command.

LEV *n*

Define part transformation sequence *n*. This defines a series of operations which can be performed on groups of parts.

Options:

ADD *m*

Add the list of transformations in sequence number *m* to the current sequence.

COOR *n* data

Add *n* coordinate transformations to the current sequence. The data for this command is described in the section, *Coordinate Transformations*.

CYLI	Perform a cylindrical coordinate transformation.
PROD *i j*	Form the product of sequence *i* with sequence *j*. If sequence *i* has *1* transformations, and sequence *j* has *m* transformations,then this option produces *1*m* transformations and adds them to the current sequence.
REPE l_1 l_2 ...;	Copy parts in global coordinate systems l_1, l_2, ...
SPHE	Perform a spherical coordinate transformation.
;	Terminate this command.
LMI *n*	Increment the default material number by *n* for each local copy of a part. This number is initially set to zero.
LSYS name	Define local system name for single point constraints, etc.
Options:	
PLANE p_x p_y p_z r_x r_y r_z	The local *X*-axis is parallel to (p_x,p_y,p_z) and (r_x,r_y,r_z) is a vector in the *XY*-plane.
2 c_x c_y c_z p_x p_y p_z r_x r_y r_z	The center of the local system is (c_x,c_y,c_z). (p_x,p_y,p_z) is a point along the local *X*-axis and (r_x,r_y,r_z) is a point in the *XY*-plane.
NODE n_1 n_2 n_3	The center of the local system is node n_1. Node n_2 is a point along the local *X*-axis and node n_3 is a point in the *XY*-plane.
;	Terminate LSYS command.
MAT *n* data	Code-dependent material data can be input. See the chapter on the specific computer program for input related to the MAT command.
MATE *m*	The default material name for the following parts is set to *m*. This name is initially set to 1.
MAZT tol	Set the MAZE tolerance to tol. This is used for a variety of two-dimensional line definitions and the MAZE parts.
MDBC *m* lc amp f_x f_y f_z	Material displacment boundary condition. This command is used only for rigid body materials in DYNA3D. The load curve number is lc, amp is the scale factor and (f_x,f_y,f_z) is in the load direction.

MKDS	Make a binary database of digitized 3-D surfaces. Digitized surfaces are generated using the **DS** command and they are read back in using the **RDDS** command. This command is primarily intended to allow fast reinitialization during restarts of LS-INGRID.
MFBC mat lc amp f_X f_Y f_Z	Apply force to rigid body material *mat*. The force is scale by load curve lc and factor amp and is in direction (f_X,f_Y,f_Z)..
MVBC *m* lc amp f_x f_y f_z	Material velocity boundary condition. This command is used only for rigid body materials in DYNA3D. The load curve number is lc, amp is the scale factor and (f_x,f_y,f_z) is in the load direction.
NFG name	Define nodal force group name.
Options:	
LSYS name	The nodal force group is defined relative to local system name. (default=global)
;	End of nodal force groups.
NIP m_1 m_2	Non-interacting pairs of materials. This is used to determine lists of noninteracting segments for use by FACET to determine radiation view factors.
NK2D *i j*	Output generated is compatible with LS-NIKE2D *i* and *j* flags specifying which 3-D coordinates correspond to the LS-NIKE2D *r* and *z* coordinates. *i* and *j* can have values "*x*", "*y*", or "z". This command activates additional control commands which are described in *LS-NIKE2D Options and Materials*.
NK3D	Output is generated for LS-NIKE3D. This command activates additional control commands which are described in *LS-NIKE3D Options and Materials*.
NOPL	Do not perform plotting. This command suppresses the normal prompting for a graphics device and is useful in combination with the **BATC**H command.
NOTE	Input a not eto be included into the output file. Example:

NOTE "Copyright 1985"

NSMOOTH *n* — Perform *n* smoothing operations on surfaces when using the standard part. The default is zero since this can be costly and is usually necessary only for complex free form surfaces.

ORV n {options} — Input orientation vector n.

Options:

PLANE v_x v_y v_z — The sping/damper distances are measured in the plane defined by normal vector (v_x,v_y,v_z).

VECTOR v_x v_y v_z — The spring/damper distances are measured along the vector defined by (v_x,v_y,v_z).

; — End the ORV command.

PAUSE — Execute a FORTRAN pause statement.

PLANE nplane — Input nplane plane definition. These planes are for applying boundary conditions only. Do not try to use this command more than once in the same input file.

Repeat the following information for each plane:

p_x p_y p_z — Global coordinates of any point on the plane.

q_x q_y q_z — Any vector normal to the plane.

Tolerance — All nodes within a distance less than tolerance from the plane are included in the definition. If the tolerance is negative, no nodes will be found. The "SW" command in the standard part can also be used to include nodes in the definition.

Options:

CYLI *radius len* — The stonewall is a cylindrical surface. The radius is *radius* and the length is *len*. If *len*=0, then an infinite cylinder is assumed. (LS-910 and later).

FRIC *m* — Specify stonewall friction properties.
- *m*=0: Frictionless sliding occurs.
- 0<*m*<1: *m* is colomb friction coefficient (LS-910 and later).
- *m*=1: No tangential motion allowed during contact.

LCD *lc* v_x v_y v_z — Load curve *lc* specifies the displacement history of the stone wall in the direction (v_x,v_y,v_z).

LCV *lc* v_x v_y v_z	Load curve *lc* specifies the velocity history of the stone wall in the direction (v_x,v_y,v_z).
MASS *mass*	The stonewall has mass *mass*. (LS-910 and later).
OVERRIDE *i*	If a node is also on plane *i*, then this plane takes precedence.
PLANE a_x a_y a_z alen blen	The stonewall is a finite plane. (a_x,a_y,a_z) is a vector which specifies an in-plane *a*-axis. The *b*-axis is determined from the cross-product of the *a*-axis with the normal vector. alen is the extent of the plane along the *a*-axis and blen is the extent along the *b*-axis. (LS-910 and later).
PRISM a_x a_y a_z alen blen clen	The stonewall is a prism. (a_x,a_y,a_z) is a vector which specifies an in-plane a-axis. The *b*-axis is determined from the cross-product of the *a*-axis with the normal vector. alen is the extent of the plane along the *a*-axis and blen is the extent along the *b*-axis. *c*-len is the extent along the normal axis. (LS-910 and later).
SPHE *radius*	The stonewall is a spherical surface. The radius is *radius*. (LS-910 and later).
VELOCITY *v*	The stonewall has a initial velocity *v* normal to the surface. (LS-910 and later).

One of the following three options is required to terminate the plane definition:

ASYM	Asymmetric boundary conditions are applied to the nodes.
STONE or **SW**	The boundary condition is a stonewall.
SYMM	Symmetric boundary conditions are applied to the nodes.
PPLV	Eliminate the part transformation sequence at the top of the stack. See also "PPLV" and "LEV" in this section.
PRINT *v*	Echo the value of *v* to the terminal. This is primarily used with the calculator functions to verify calculations.
PSCALE m_1 m_2	Scale properties. Materials from m_1 to m_2 are treated by this command. If m_1 and m_2 are numbers, than standard numeric comparisons are used to determine if materials are within the range. Otherwise, string comparisons are used.
SECTION *scale*	Scale all section properties by *scale*. This allows for a general unit conversion on section

properties.

; Terminate PSCALE command.

PSLV *n* Begin performing part transformation sequence *n* on all following parts. This remains in effect until a PPLV command is given. A stack is used for performing transformation sequences. PSLV adds a transformation sequence to the top of the stack and PPLV eliminates the top sequence on the stack.

QUAD Turn on generation of elements with quadratic shape functions in standard part. This causes 8-node shells and 20-node bricks to be generated.

RBMG m_1 m_2 Merge rigid body m_1 to rigid body m_2. m_1 is the master and m_2 is the slave. For a group of merged rigid bodies, there can be only one master. (DYNA3D only.)

READ Read external database.

Options:

NDIV *n* Number of subdivisions for internal NURB surface processing. (Default=1) Setting this to 2 or 3 can improve the reliability of intersections calculated from NURB surfaces, however, costs and memory requirements will increase roughly proportional to the square of this number.

NURB *name* Read a NURB surface database in file *name*. This ends the READ command.

SC03 *name* Read a SC03 database in file *name*. This ends the READ command.

ROTATION p_x p_y p_z v_x v_y v_z w Assign an initial rigid body rotation to all parts defined after this command. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w.

RVBC *m* lc idof amp f_x f_y f_z Rigid body velocity boundary condition. This command is used only for rigid body materials in DYNA3D. The load curve number is lc, amp is the scale factor and (f_x,f_y,f_z) is in the load direction. idof can be:

=1: *X*-translational degree-of-freedom
=2: *Y*-translational degree-of-freedom
=3: *Z*-translational degree-of-freedom

=.4: translational velocity in direction of vector (f_x,f_y,f_z).
=5: *X*-rotational degree-of-freedom
=6: *Y*-rotational degree-of-freedom
=7: *Z*-rotational degree-of-freedom
=8: rotational velocity in direction of vector (f_x,f_y,f_z).
=9: *Y* and *Z* degrees-of-freedom for node rotating about the global *X*-axis
=10: *Z* and *X* degrees-of-freedom for node rotating about the global *Y*-axis
=11: *X* and *Y* degrees-of-freedom for node rotating about the global *Z*-axis

SD *n* data — Begin definition of surface *n*. If surface n has been previously defined, this command has the effect of destroying the old definition. See *Surface Definitions* for a description of the additional input for this command.

SDMV s_1 s_2 data — Move surface definitions s_1 through s_2. data is described in the section on *Coordinate Transformations*.

SI islide — Define sliding interface islide. These options apply to both slide surfaces and slide lines.

Options:

A3 — Select contact interface type a3 (LS-920). Contact type a3 is insensitive to orientation of the contact segments.

A5 — Select contact interface type a5 (LS-920). Contact type a5 is insensitive to orientation of the contact segments.

A13 — Select contact interface type a13 (LS-920). This model is a single surface method which is principally used for inflating folded airbags.

BIRTH *t* — Birth time for interface (LS-910).

BOND *t* — GA slideline option: Bond shear modulus.

BOXM x_m x_x y_m y_x z_m z_x — Define box for master side of sliding interfaces (LS-910, VECDYNA).

BOXS x_m x_x y_m y_x z_m z_x — Define box for slave side of sliding interfaces. (LS-910, VECDYNA)

COMP *t* — GA slideline option: compressive strength of concrete.

DAMP *d*	Damping coefficient (percent of critical) (LS-910).
DEATH *t*	Death time for interface (LS-910).
DNIS	Discrete nodes impacting surface.
DNTS	Discrete nodes tied to surface.
DUMMY	Dummy slide surface. This option can be used to allow distinct but coincident nodes.
FAIL *e*	Tied slide surface with failure when volume weighted strain exceeds *e* (LS-DYNA3D).
FD *fd*	Dynamic friction coefficient.
FE *d*	Exponential decay coefficient.
FFN *f*	Normal failure force.
FFNE *f*	Normal failure exponent.
FFS *f*	Shear failure force.
FFSE *f*	Shear failure exponent.
FRIC *f*	Set static and dynamic friction to *f*. (default=0)
FS *fs*	Static friction coefficient.
GA	Select General Atomic's 1-D rebar slideline.
HDMG t	GA slideline option: Exponent in damage curve.
LCV	Load curve for force-penetration in types 19 and 20 contact. (LS-920 and later).
LS	Turn on limited search flag. (Default is off).
MATERIAL **MAST** m_1 m_2 ... ;	The master side of the interface consists of material subset m_1, m_2, ... (VEC/DYNA3D, LS-920 and later).
MATERIAL **SLAV** m_1 m_2 ... ;	The slave side of the interface consists of material subset m_1, m_2, ... (VEC/DYNA3D, LS-920 and later).
MAXS *t*	GA slideline option: maximum shear displacement.
MERGE	Coincident nodes are merged.

MSCA *s*	Scale factor for master thicknesses. (LS-910 and later).
MTHI *t*	Master side thickness. (LS-910 and later).
NFAIL *fs*	Normal failure stress.
NOMERGE	Coincident nodes are not merged.
NSWS	Nodes spot welded to surface.
PNLM *p*	Change master side penalty to *p*.
PNLS *p*	Change slave side penalty to *p*.
PNLT *p*	Change penalty to *p*.
RADIUS *t*	GA slideline option: Radius of rebar.
SETS	Shell edge tied to shell surface.
SFAIL *fs*	Shear failure stress.
SINGLE	Single sided slide surface.
SL	Sliding only.
SSCA *s*	Scale factor for slave thicknesses. (LS-910 and later).
STHI *t*	Slave side thickness. (LS-910 and later).
SV	Sliding with voids (default).
T10	Type 10 interface.
T11	Type 11 interface. This is the box/material limited automatic contact for shells in LS-910 and later. It is the single surface airbag contact for MVMA/DYNA3D.
T12	Type 12 interface. Automatic contact for shells. (LS-910 and later).
T13	Type 13 interface. (LS-920 and later. Converts to the similar type 11 in MVMA/DYNA3D).
T14	Type 14 interface. Surface to surface eroding contact. (LS-920 and later).
T15	Type 15 interface. Node to surface eroding contact. (LS-920 and later).
T16	Type 16 interface. Single surface eroding

contact. (LS-920 and later).

T17 Type 17 interface. Surface to surface symmetric/asymetric constraint method. (LS-920 and later).

T18 Type 18 interface. Taylor and Flanagan contact force calculation technique from PRONTO3D. (LS-920 and later).

T19 Type 19 interface. Rigid body to rigid body with specified force deflection curve. (LS-920 and later.)

T20 Type 20 interface. Node to rigid body with specified force deflection curve. (LS-920 and later.)

TBI Tie-break interface.

TCRS *r* Thermal contact resistance is *r*.

TIED Tied slide surface

VFRI *v* Viscous friction coefficient is *v*.
; Terminate this slide surface definition.

SPD *n* Input definition for spring/damper.

Options:

ROTA The spring/damper is rotary and not translational.

*Options (the following options end the **SPD** command):*

LE *e* Define a linear elastic spring with stiffness *e*. (force/displacement).

LV *d* Define a linear damper with damping constant *d*. (force/velocity).

IEP *e t y* Define an elastic-plastic spring with stiffness *e* (force/displacement), tangent stiffness *t* (force/displacement), and yield *y* (force).

NE *l* Define a nonlinear spring using load curve *l*. *l* represents force versus displacement.

NV *l* Define a nonlinear damper using load curve *l*. *l* represents force versus velocity.

GN l_l l_{ul} b Y_t Y_c

Define a general nonlinear spring. The spring loads along load curve l_l and unloads along l_{ul} with hardening parameter b. The initial yield in tension is Y_t and Y_c for compression. (LS-910 and later).

VE K_0 K_i b T_c F_c iopt

Three-parameter Maxwell viscoelastic spring. K_0 is the short time stiffness, K_i is the long time stiffness with decay parameter b. T_c is a cutoff time and F_c is the force after cutoff. iopt is zero for an incremental treatment and nonzero for a continuous treatment. (LS-910 and later).

TCO l K_{ul} flag

Inelastic tension or compression only. The spring loads along load curve l. K_{ul} is an optional unloading stiffness and flag is -1.0 for tension only and +1.0 for compression only. (LS-910 and later).

SLVM m_1 m_2

This command applies to the DYNA3D coupling with CAL3D or MADYMO3D. Deformable materials can be identified as being slaved to rigid bodies which are coupled to CAL3D or MADYMO3D. During the DYNA3D initialization, the deformable materials will be repositioned to reflect the shifting to global coordinates performed by CAL3D or MADYMO3D. The master rigid body is material m_1 and the slaved deformable material is m_2. (LS-920 and later.)

STOL t

Set the tolerance for surface intersections to t. (Default=1.0e-6.)

STOP

Execute a FORTRAN stop statement.

SYNTAX

Command for redefining the syntax of various part options.

Options:

REGION

Set syntax for <Region> in part definitions.

Options:

STANDARD

Set syntax for <Region> in standard part.

Options:

STANDARD

Use the standard syntax for <Region> in the

	standard part.
;	Terminate SYNTAX command.
MAZE	Set syntax for <Region> in the MAZE part.

Options:

STANDARD	The syntax for the <MRegion> is according to this manual and the 1985 INGRID manual from LLNL.
PD	The syntax for the <MRegion> has 6 indices according to the modification to INGRID by LLNL after 1986.
;	Terminate SYNTAX command.
OLD	Set syntax for <Region> in the OLD, BEAM and other low level input parts.

Options:

STANDARD	Use the standard definition of <Region>. This assumes <Region> = imin imax, unless the first item encountered is an expression in brackets. For an expression, it will only read one parameter.
1	Use a one paramter definition of <Region>. <Region> = imin=imax normally identifies single nodes and elements only, but can also identify a range with an expression.
2	Use a two parameter definition of <Region>. This does not allow an expression to specify the range, but is necessary for the following: B [nodebeg] [nodeend] 111000. This is because the standard method would see [nodebeg] and convert to a one parameter method.
;	Terminate SYNTAX command.
SYSEND	Terminate existing subsystem definition.
SYSTEM name	Begin definition of subsystem *name*. This remains in effect until a SYSEND or another SYSTEM command is encountered or another

	SYSTEM.
T12	This command must be typed just prior to the use of the MAZE part. The third side, L_3, of the next part will have exactly two times as many elements as side L_1. The transition is accomplished with quadrilateral elements. This command does not apply to triangular parts.
T13	This command must be typed just prior to the use of the MAZE part. The third side, L_3, of the next part will have exactly three times as many elements as side L_1. The transition is accomplished with quadrilateral elements. This command does not apply to triangular parts.
TEMP t	All parts defined after this command have initial temperature *t*. This remains in effect until reset with another "Temp" command. This can be overridden with an individual part. *t* can be a single number or it can be an expression of the form *t*(*x*,*y*,*z*). This allows nodes to be assigned temperatures based on an analytical expression of a temperature distribution based on the nodal coordinates.
THIC *t*	The default thickness for shells is *t*.
TIND mat I_{xx} I_{xy} I_{xz} I_{yy} I_{yz} I_{zz}	Specify inertia tensor.
TINE mat I_{xx} I_{yy} I_{zz} I_{xy} I_{yz} I_{zx}	Specify inertia tensor.
TIVE mat v_x v_y v_x w_x w_y w_z	Initial velocities (global), translational and rotational
TMCG mat c_x c_y c_z	Specify center of gravity.
TMM *n t*	The total mass of material *n* is *t*. The density of the material is determined by dividing the total mass of the material by the calculated volume.
TMSM *m* s_1 s_2 ... ;	The inertial properties which are input for material m include the masses of deformable materials s_1, s_2, ... The properties of *m* are computed such that the total mass properties of *m*, s_1, s_2 ... is equal to the input values.
TMVP mat (transformation)	Move center of gravity and inertias. Transformation refers to the section, *Coordinate Transformations*.
TRACER	Define tracer particles for material.

Options:

LNPT p_x p_y p_z q_x q_y q_z n	Generate n equals spaced tracer particles on the line from (p_x,p_y,p_z) to (q_x,q_y,q_z).
MATERIAL t	The tracer particle is fixed to a material point.
POINT p_x p_y p_z	Define a tracer particle starting at point (p_x,p_y,p_z).
SPACE t	The tracer particle is fixed in space.
TIME t	Activation time for tracer particle.
;	Terminate this command.
TRANS	This command must be typed just prior to a MAZE part and changes the command such that k+m elements are generated along sides L_1 and L_2 and m elements are generated along sides L_3 and L_4. This command does not apply to triangular parts or parts with variable zoning.
TZ2D i j	Output generated is compatible with TOPAZ2D. i and j flags specifying which 3-D coordinates correspond to the TOPAZ2D r and z coordinates. i and j can have values "x", "y", or "z". This command activates additional control commands which are described in *TOPAZ Options and Materials*.
TZ3D	Output is generated for TOPAZ3D. A FACET input deck will also be created if necessary. This command activates additional control commands which are described in *TOPAZ Options and Materials*.
WRITE *format* v_1 v_2 . . ;	Issue a Fortran write statement. variables v_1, v_2, ... are written to standard out and *format* is the Fortran format statement. Example: WRITE "('I =',e13.5)" [i] ;
XOFF d_x	Global X-offset.
XSCA s	Scale all X-coordinates.
VD n data	Begin definition of volume n. If volume n has been previously defined, this command has the effect of destroying the old definition. *Volume Definitions* describes the data for this command.

YOFF d_y	Global *Y*-offset.
YSCA *s*	Scale all *Y*-coordinates.
VELOCITY v_x v_y v_z	Assign initial rigid body velocity (v_x, v_y, v_z) to all parts defined after this command. v_x, v_y and v_z can be functions of (x,y,z) to allow initial velocity distributions.
ZOFF d_z	Global Z-offset.
ZSCA *s*	Scale all Z-coordinates.

3. IDEAS Part

The IDEAS part provides for importing SDRC/IDEAS neutral files into LS-INGRID. The form of the part is as follows:

IDEA filename
<optional functions>
END

filename is the name of the IDEAS neutral file.

3.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of:

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M          C  Elements of material 2 are assigned to
                         C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data — Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.

CYLI — Nodes are converted from cylindrical to rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta$
$Y = R \sin \theta$

LREP l_1 l_2 ... l_n	Repeat command. This command makes copies of the part in each of the **local** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
MATE matnum	The part has material number matnum.
REPE l_1 l_2 ... l_n	Repeat command. This command makes copies of the part in each of the **global** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
ROTA p_x p_y p_z v_x v_y v_z w	Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per second.
SPHE	Nodes are converted from spherical to rectangular coordinates. The equations for this transformation are: $X = R \cos\theta \sin\varphi$ $Y = R \sin\theta \sin\varphi$ $Z = R \cos\varphi$
TEMP t	The initial temperature of this part is t and it can be expressed as a function of x, y, z coordinates.
THIC thic	Plates have the thickness thic for this part.
VELO v_x v_y v_z	Assign initial rigid body velocity to all nodes within this parts. (V_x, V_y, V_z) is the global velocity vector. (V_x, V_y, V_z) can be expressed as a function of x, y, z coordinates.

4. PATRAN Part

The PATRAN part provides for importing PATRAN neutral files into LS-INGRID. The form of the part is as follows:

PATRAN filename
<optional functions>
END

filename is the name of the PATRAN neutral file.

4.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M          C  Elements of material 2 are assigned to
                         C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data — Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.

CYLI — Nodes are converted from cylindrical to rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta$

$Y = R \sin \theta$

LREP l_1 l_2 ... l_n	Repeat command. This command makes copies of the part in each of the **local** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
MATE matnum	The part has material number matnum.
REPE l_1 l_2 ... l_n	Repeat command. This command makes copies of the part in each of the **global** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
ROTA p_x p_y p_z v_x v_y v_z w	Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per second.
SPHE	Nodes are converted from spherical to rectangular coordinates. The equations for this transformation are: $X = R \cos\theta \sin\varphi$ $Y = R \sin\theta \sin\varphi$ $Z = R \cos\varphi$
TEMP t	The initial temperature of this part is t and it can be expressed as a function of x, y, z coordinates.
THIC thic	Plates have the thickness thic for this part.
VELO v_x v_y v_z	Assign initial rigid body velocity to all nodes within this parts. (V_x, V_y, V_z) is the global velocity vector. (V_x, V_y, V_z) can be expressed as a function of x, y, z coordinates.

5. NASTRAN Part

The NASTRAN part provides for importing NASTRAN input files into LS-INGRID. The form of the part is as follows:

NASTRAN filename
<optional functions>
END

filename is the name of the NASTRAN input file.

5.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of:

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M          C  Elements of material 2 are assigned to
                         C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data — Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.

CYLI	Nodes are converted from cylindrical to rectangular coordinates. The equations for this transformation are: $X = R \cos \theta$ $Y = R \sin \theta$
LREP l_1 l_2 ... l_n	Repeat command. This command makes copies of the part in each of the **local** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
MATE matnum	The part has material number matnum.
REPE l_1 l_2 ... l_n	Repeat command. This command makes copies of the part in each of the **global** coordinate systems l_1 to l_n If the coordinate system number is zero, the part is repeated with no transformation.
ROTA p_x p_y p_z v_x v_y v_z w	Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per second.
SPHE	Nodes are converted from spherical to rectangular coordinates. The equations for this transformation are: $X = R \cos \theta \sin \varphi$ $Y = R \sin \theta \sin \varphi$ $Z = R \cos \varphi$
TEMP t	The initial temperature of this part is t and it can be expressed as a function of x, y, z coordinates.
THIC thic	Plates have the thickness thic for this part.
VELO v_x v_y v_z	Assign initial rigid body velocity to all nodes within this parts. (V_x, V_y, V_z) is the global velocity vector. (V_x, V_y, V_z) can be expressed as a function of x, y, z coordinates.

Notes:

1. The following NASTRAN keywords are supported:

CBAR	CBEAM	CDAMP
CELAS2	CHEXA	CMASS2
CONM2	CORD1C	CORD1R
CORD1S	CORD2C	CORD2R
CORD2S	CPENTA	CQUAD4
CTETRA	CTRIA3	FORCE
GRAV	GRID	MAT1
MPC	PBAR	PBEAM
PLOAD2	PLOAD4	PSHELL
PSOLID	RBE2	SPC
SPC1		

2. The following keywords from MSC/DYNA are also supported:

DYMAT24	MATRIG

3. To preserve the arbitrary node, element and material numbering of NASTRAN input files, use the ARBITRARY command. (See *LS-DYNA3D Commands and Materials*.)

4. The material properties from the NASTRAN input are generally not used for LS-DYNAxx calculations. To assign properties from LS-INGRID, the materials may be defined either before or after the NASTRAN part. The LS-INGRID material ID's are input the same as those of the NASTRAN input. If the LS-INGRID materials are defined prior to the NASTAN model input, but the sections are not input, then LS-INGRID will try to use the section property data from the NASTRAN input. Section properties may be scaled using the global command: PSCALE.

5. Node, element and material ID's can be shifted using the global command, LABELS.

6. Standard Part

6.1 DEFINITIONS

Index Space: An index space is a three-dimensional discrete coordinate system with integer values greater than or equal to 1 in each of the three directions. The three discrete coordinates are labeled, *I, J,* and *K* axes respectively. Each point in the index space (*i, j, k*), represents a nodal point. Elements are defined as groups of adjacent nodes in the index space.

Region: A region is any rectangular or cubic block of nodes. A region is usually defined by a block in an index space.

Part: A part is a collection of regions which can be grouped and generated conveniently in an index space. Beginning users will typically use one region per part while more experienced users will be able to group numerous regions together into complex parts.

Model: The final model is a collection of parts. Each part has its own index space and is independent of other parts. Parts are connected together either by global coincident node removal, slide surfaces, or other constraints.

The standard part in LS-INGRID is based on a three-dimensional index space which is commonly used for finite difference mesh generation. Although this can be somewhat awkward for finite element meshes, proper usage technique and some enhancements have made this quite effective for certain geometries including some that are difficult for standard finite element mesh generators. The principal enhancement to the three-dimensional index space is an additional type of index notion, the "Index Progression". Index progressions provide a concise and simple method for describing complex structures, and are used to input data to LS-INGRID. The following is a detailed description of the index space notion and the index progression. This information provides the user with the concepts necessary to use LS-INGRID effectively.

Index Space

Node generation in LS-INGRID is done by a mapping from Index space onto the object of interest as is shown in Figure 6-1. Each region of the object is referenced by a set of six indices; (IMIN, JMIN, KMIN) specify the minimum indices for a region in the index space and (IMAX, JMAX, KMAX) specify the maximum indices. For a solid region, all eight corner nodes are defined by combinations of minimum and maximum indices. Table 6-1 lists the indices of the vertices in the example of Figure 6-1. We assume that any set of three indices, (*I, J, K*), defines a region in space.

If KMIN is set equal to KMAX, the resulting region is a plane of constant *K* as shown

in Figure 6-2a. Similarly, a plane of constant *I* is defined when IMIN is set equal to IMAX and a plane of constant *J* for JMIN equal to JMAX. A line in the index space is defined by holding two indices constant while the third index varies as shown in Figure 6-2b.

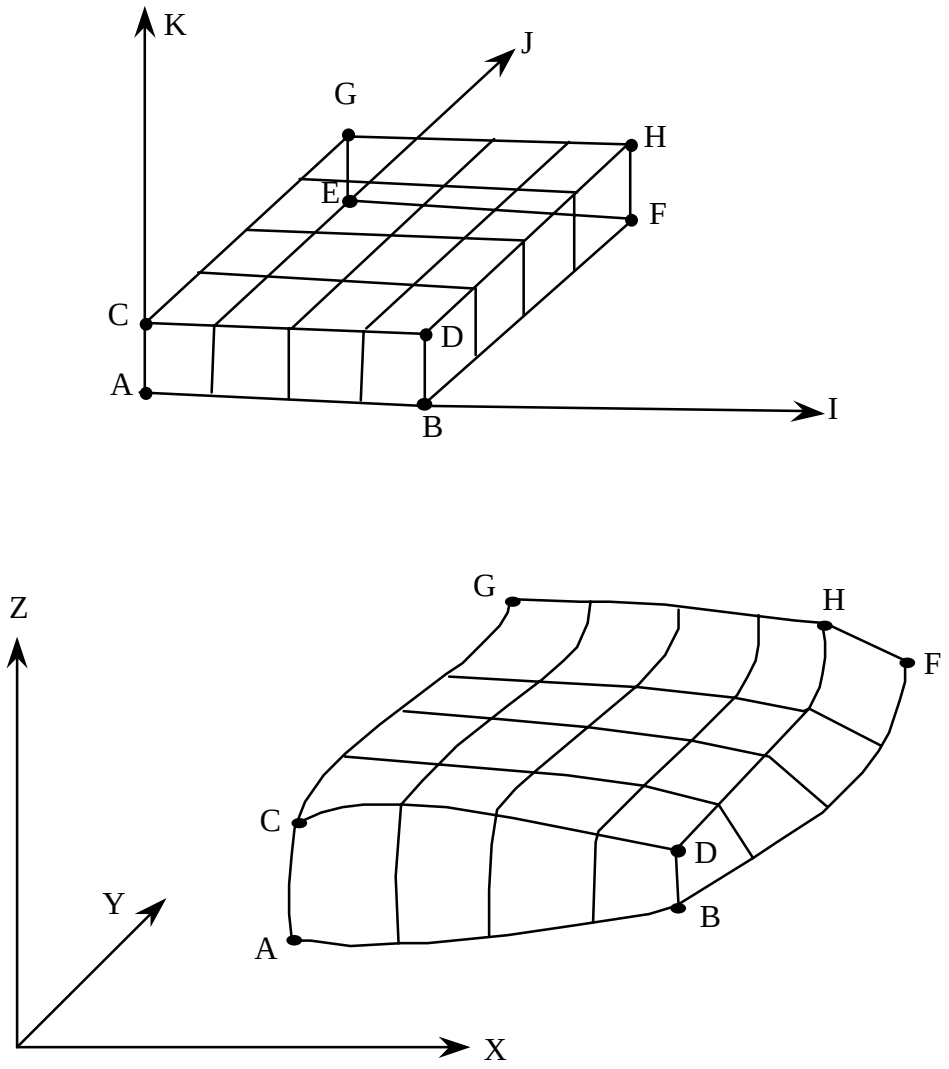


Figure 6-1. Mapping from index space to object space.

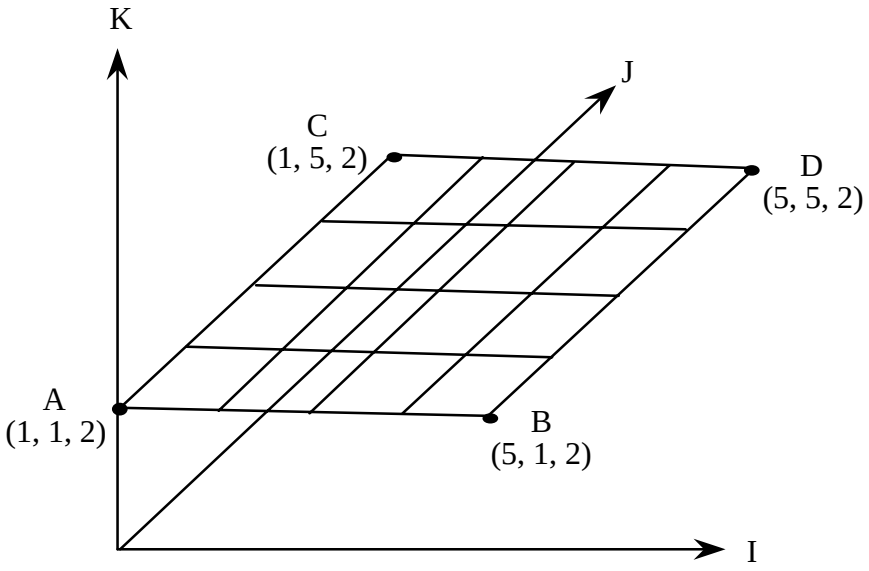


Figure 6-2a. Planes in index space.

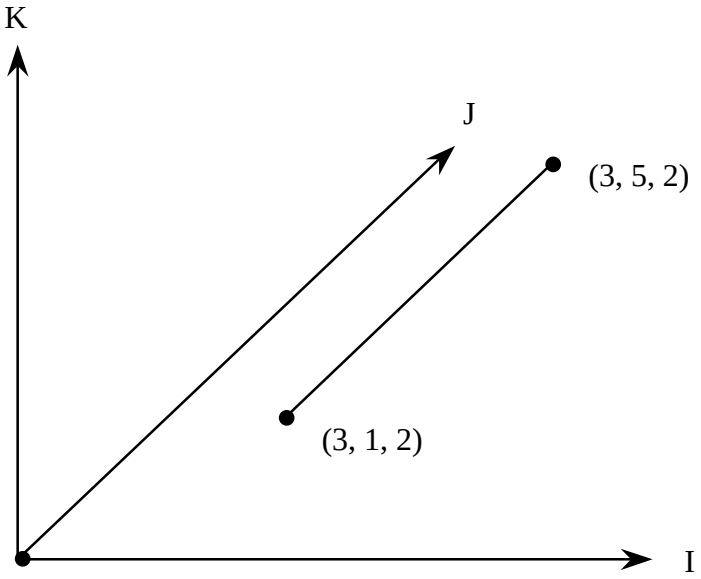


Figure 6-2b. Lines in index space.

TABLE 6-1. Indices associated with the vertices of a region.

Node	Indices	Position
A	(1, 1, 1)	(IMIN, JMIN, KMIN)
B	(5, 1, 1)	(IMAX, JMAX, KMAX)
C	(1, 1, 2)	(IMIN, JMIN, KMIN)
D	(5, 1, 2)	(IMAX, JMAX, KMAX)
E	(1, 5, 1)	(IMIN, JMIN, KMIN)
F	(5, 5, 1)	(IMAX, JMAX, KMAX)
G	(1, 5, 2)	(IMIN, JMIN, KMIN)
H	(5, 5, 1)	(IMAX, JMAX, KMAX)

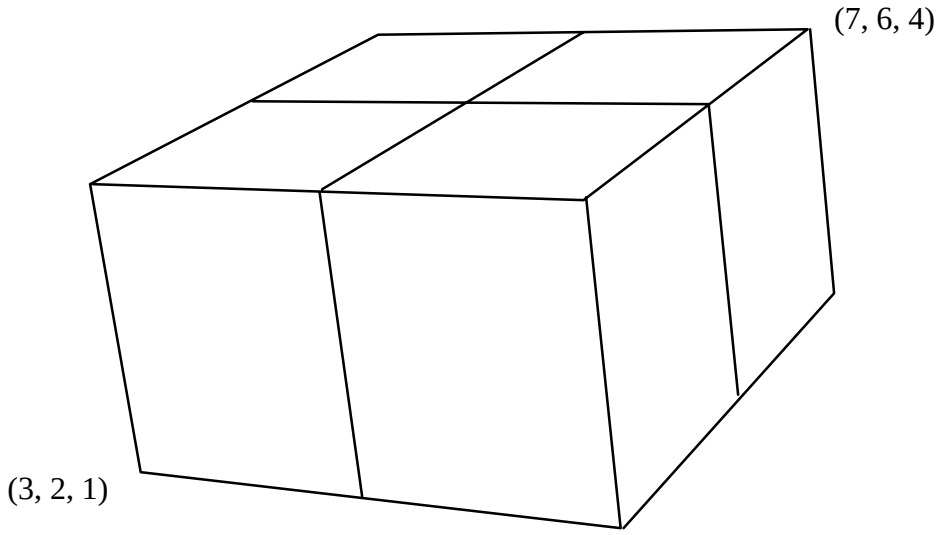
An index space is defined as the set of all possible indices < <IMAX, < <JMAX, 1 < <KMAX. If an index is zero, then it varies over all possible indices. Thus, the indices (3, 0, 2) defines a line which extends across the index space, and (0, 0, 2) defines a plane which divides the index space into two regions. (0, 0, 0) defines the entire index space.

Index Progressions

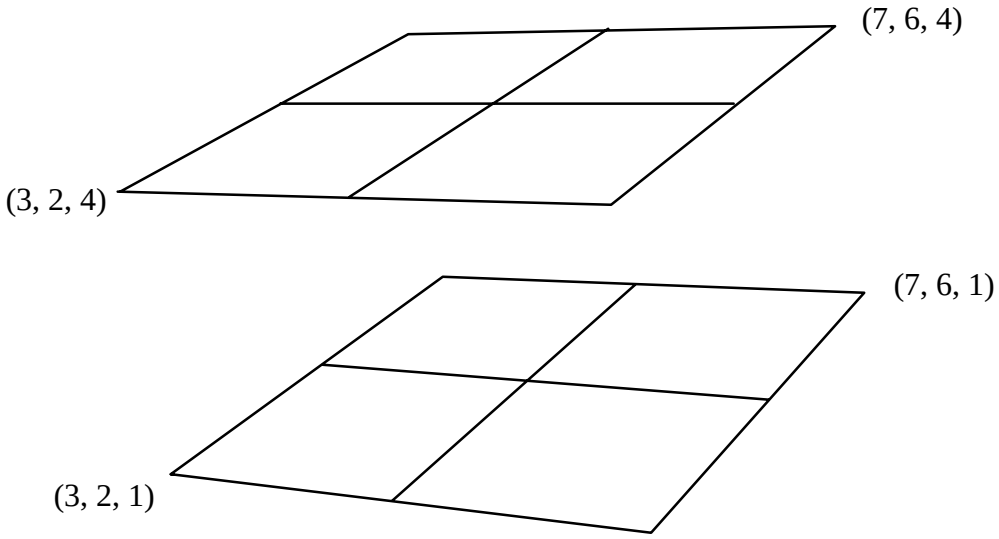
Index progressions were developed to facilitate the defining of multiple regions in index space. Rather than specifying the minimum and maximum indices for a region, one simply specifies the progression in indices along the *I*, *J*, and *K* directions respectively. For example, the region (2, 7, 6, 8, 9, 10) is represented as the progression (2, 8); (7, 9); (6, 10). If there is a region adjacent to (2, 7, 6, 8, 9, 20) such as (2, 7, 2, 8, 9, 6), the two regions are defined together by a new progression (2, 8); (7, 9); (2, 6, 10). To define the four solids regions shown in Figure 6-3a requires the progression (3, 5, 7); (2, 4, 6); (1, 4).

Index progressions for planes are defined in a similar manner. The index which remains constant throughout a plane is indicated by a negative sign so the plane (2, 5, 5, 2, 7, 8) is represented as (-2); (5, 7); (5, 8).

In Figure 6-3b there are eight planes which can be represented by the progression (3, 5, 7); (2, 4, 6); (-1, -4). The savings by this notation is apparent since specifying separately the eight regions in Figure 6-3b requires 49 number where as the index progression requires only 8 numbers.



(a) Index Space



(b) Object Space

Figure 6-3. Index progressions for planes and solids.

Another addition to the index progression notation is the zero index. The two solids regions shown in Figure 6-4 could be represented as an index progression except that they are not connected. In this case, a zero index is used along the I direction to indicate that the structure is discontinuous. This gives the progression (2, 4, 0, 6, 8); (3, 7); (4, 5). Plane regions can be separated by the zero index in a manner similar to solid regions.

More complicated regions can be represented by combining index progressions. An example of this is in Figure 6-5. The open box could be represented by two index progressions (-2, -5); (1, 7); (3, 5); and (2, 5); (1, 7); (-3, -5), but they can also be combined to give (-2, -5); (1, 7); (-3, -5). Figure 6-6 shows several more structures and their index progression representation.

LS-INGRID uses the index progression notation to set up regions in the index space which are to be mapped onto the object of interest. This notation has the advantage that it requires little input data and with less than 20 indices can represent thousands of configurations in index space. In practice, not all configurations in index space can be defined by an index progression so a command is added to allow deletion of regions in the index space. The delete command along with the index progression is enough to produce almost any conceivable region in the index space and is used as the central part of LS-INGRID's mesh generation.

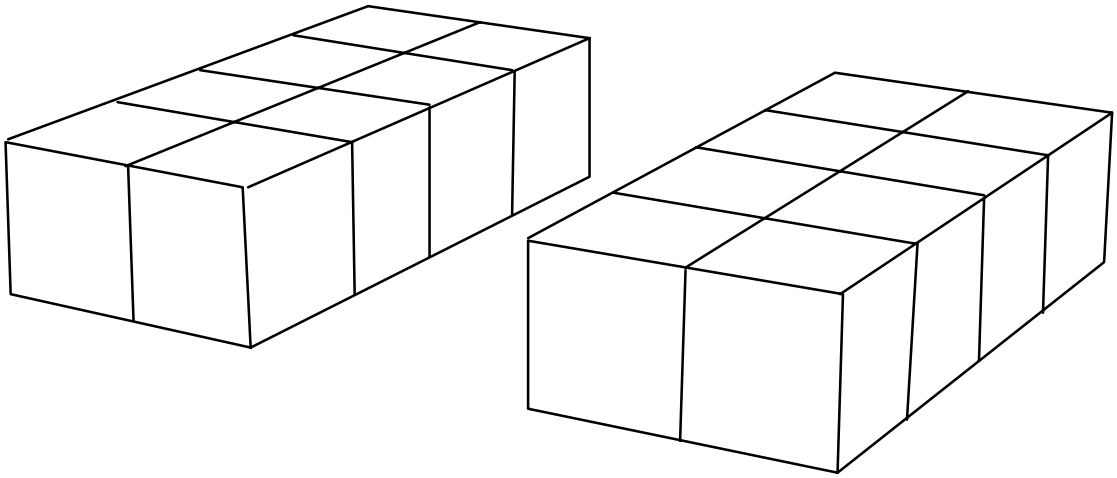


Figure 6-4. Separated solid regions.

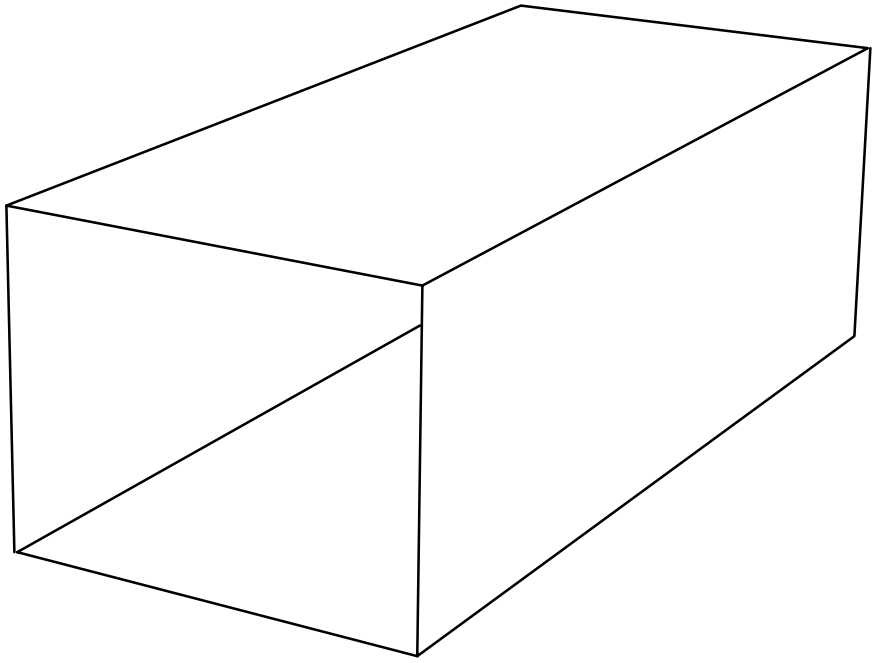
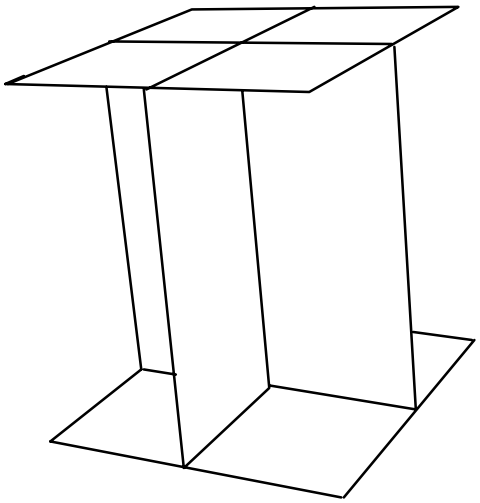
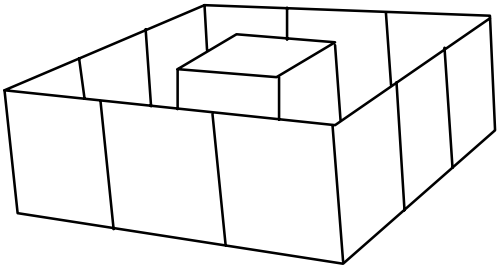


Figure 6-5. Open Box.



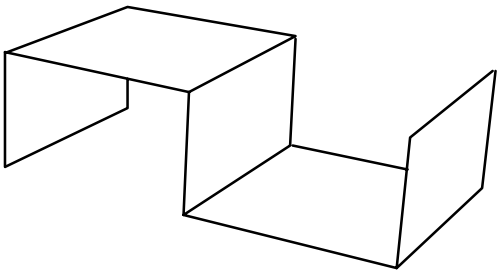
(a) Intersecting Plates

((2, -4, 6); (2, -4, 6); (-3, -7))



(c) Cube in a Box

((-2, 4, 6, -8); (-2, 4, 6, -8); (3, 5))

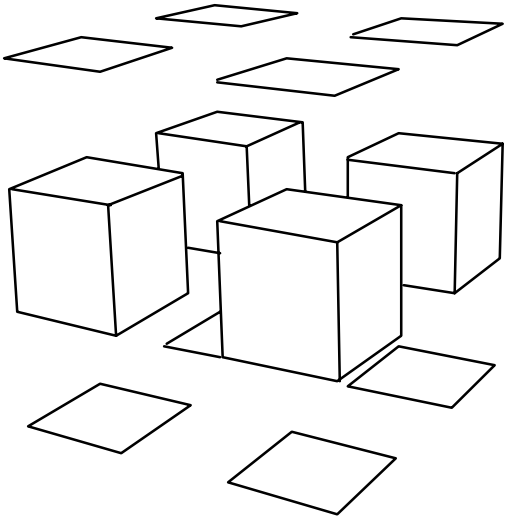


(c) Examples of Region Deletion

((-2, -6, -10); (3, 7); (-2, -4))

Deleted Regions:

(2, 3, 2, 6, 7, 2) and (6, 3, 4, 10, 7, 4)



(d) Planes and Solids with Gaps

((2, 4, 0, 6, -8); (2, 4, 0, 6, 8); (-2, 4, 6, -8))

Figure 6-6. Examples of index progressions.

Each part definition consists of the following data:

START
Index progression
Part control commands and functions
Loads and Boundary Conditions
END

"START" signifies the beginning of a part definition and is require as the first card in each part. Part control commands affect properties of the mesh. Following is a list of the default properties for a part.

4 node plate elements
8 node solid element
rectangular coordinates
material property = 1
plate thickness = 0.0

The dimension of the index space along with all plane and solid regions are defined by the index progression. Function cards manipulate the mesh defined by the index progression and an "END" signifies the end of a part. Following are some important definitions in addition to those previously given.

Index Space

The set of all indices defined by an index progression. For example, the progression (2, 3, -5, 10), (4, 5), (2, 6) defines the index space 2^|I|^10, 4^|J|^5, 2^|K|^6.

Reduced Index Space

The reduced index space references positions in an index progression. The point, *I*, *J*, *K* in the reduced index space refers to the point in the index progression defined by the *I*th integer in the *I*-progression, the *J*th integer in the *J*-progression and the *K*th integer in the *K*-progression. For the progression (2, 3, -5, 10), (4, 5), (2, 6) the relationship between the reduced index space and the index space is shown in Table 6-3.

Unless otherwise noted, all points and regions are defined in the reduced index space. Since the reduced index space is independent of the actual values of the index progression, the mesh can be refined or contracted only by changing the index progression.

Table 6-3. Comparison of the Reduced Index Space and the Index Space for the Index Progression (2, 3, -5, 10), (4, 5), (2, 6)

Reduced Index Space	Index Space
1, 1, 1	2, 4, 2
1, 1, 2	2, 4, 6
1, 2, 1	2, 5, 2
2, 2, 2	3, 5, 6
3, 1, 2	5, 4, 6
4, 1, 1	10, 4, 2

6.2 Index Progression

Each part must have an index progression. The following input is required.

$i_1\ i_2 \ldots i_i$	Progression in *I* direction
$j_1\ j_2 \ldots j_i$;	Progression in *J* direction
$k_1\ k_2 \ldots k_i$	Progression in *K* direction
$x_1\ x_2 \ldots x_i$	Initial *X*-coordinates
$y_1\ y_2 \ldots y_i$	Initial *Y*-coordinates
$z1\ z_2 \ldots z_i$	Initial *Z*-coordinates

6.3 Part Commands and Functions

All functions have the following form:

Keyword - index specification - parameters

Index specifications have three types which are abbreviated as <Point> , <Region>, or <Index Progression>. All index specifications are applied in the reduced index space. The input is defined as follows:

<Point>	Input for <point> consists only of the three indices (i,j,k). If any index is input as zero, then the index varies from the smallest to the largest possible value.
<Region>	The function locates the region defined by (i_m,j_m,k_m,i_x,j_x,k_x). If i_m, j_m, or k_m is input as zero, the zero index is given the minimum possible value. If i_x, j_x, or k_x is input as zero, the zero index is set to the maximum possible value.
<Index Progression> i_1 i_2 i_3... ; j_1 j_2... ; k_1 k_2... ;	This is used to define multiple regions according to the rules for index progression. If no indices are found for a list in a direction, then the function is assumed to go all the way through the index space in that direction.

Functions which use **<Region>** or **<Index Progression>** for index specification can be repeated and shifted to other parts of the index space. The general form of these commands is as follows:

Keyword - **<Region>** or **<Index Progression>** parameters - first offset - parameters - second offset - parameters.

The offset information is as follows:

+ *-or-* **+o**	Either a "+" or a "+o" is required as the first information for the offset. If "+" is used then the offset occurs from the region defined by the last offset. If "+O" is used then the offset is relative to the region defined by <Region> or <Index Progression>.

One and only one of the following commands must be input following "+" or "+o".

I *di*	Increment *I* indices by *di*.
J *dj*	Increment *J* indices by *dj*.
K dk	Increment *K* indices by *dk*.
IJ *di dj*	Increment *I* and *J* indices by *di* and *dj*.
JK *dj dk*	Increment *J* and *K* indices by *dj* and *dk*.
KI *dk di*	Increment *K* and *I* indices by *dk* and *di*.`
IJK *di dj dk*	Increment *I*, *J*, and *K* indices by *di*, *dj*, and *dk*.
SIJ	Switch *I* indices with *J* indices.
SJK	Switch *J* indices with *K* indices.
SKI	Switch *K* indices with *I* indices.
A <Region>	Form a curved edge between nodes A and B. The region is a line in the reduced index space.
ityp	Flag specifying type of curve. =1: A parabola through point P1 (See Figure 6-7). =2: A circular arc through point P1. =3: A circular arc with center P2.
x	*X*-coordinate of point P1 or P2
y	*Y*-coordinate of point P1 or P2
z	*Z*-coordinate of point P1 or P2
r	Radius

If the radius is non-zero for a circular arc with center P2, then nodes A and B (See Figure 6-7) are moved radially from P2 until they are a distance equal to the radius from P2. An arc is formed through the nodes at their final location.

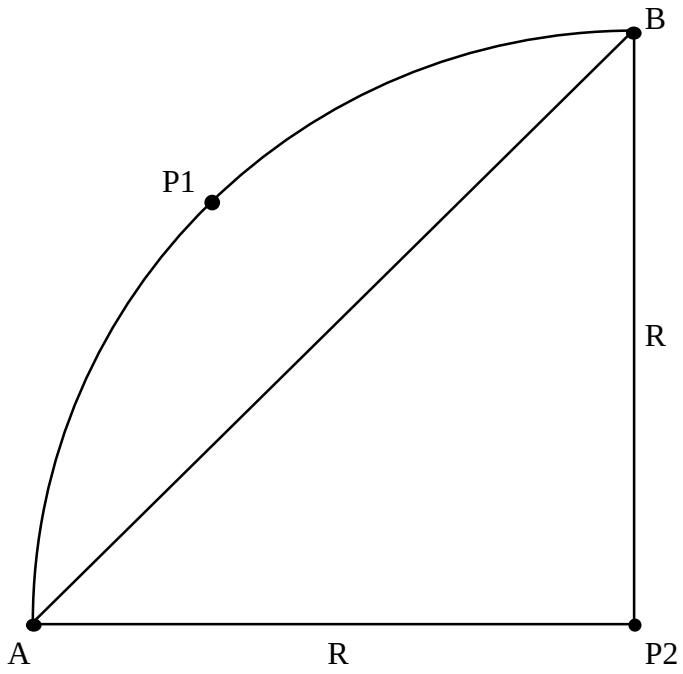


Figure 6-7. Curved boundaries.

A, **AE** <Region> — Arc keyword. The region is a plane or a solid in the reduced index space with an arbitrary length.

idir — Flag specifying axis of rotation in the index space.
="*I*": *I*-axis is axis of rotation
="*J*": *J*-axis is axis of rotation
="*K*": *K*-axis is axis of rotation

r — Radius

For any plane normal to the axis of rotation such as ABCD in Figure 6-8, a point 0 on the axis of rotation is located in the center of the plane. If the radius of the cylinder is not zero, then the points *A, B, C* and *D* are moved radially from 0 until they are a distance, *R*, from point 0. Curved boundaries are then formed for the segments *AB, BD, AC* and *CD* using center.0. This is done for each plane normal to the axis of rotation in the reduced index space.

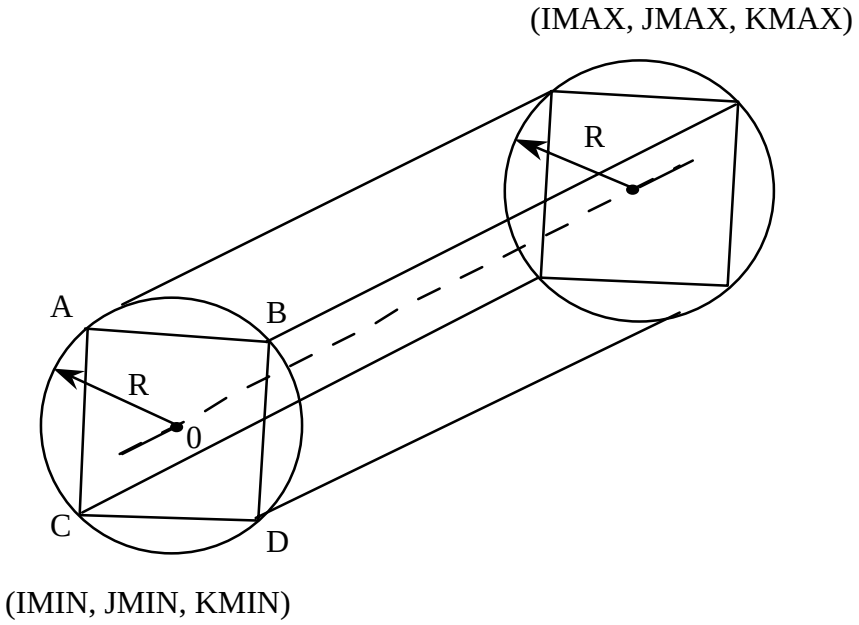


Figure 6-8. Cylindrical region.

AC, **ACE** <Region> — <Region> is a surface in the index space.

idir	Flag specifying axis of rotation in the index space. ="*I*": *I*-axis is axis of rotation ="*J*": *J*-axis is axis of rotation ="*K*": *K*-axis is axis of rotation
$p_x\ p_y\ p_z$	Any point on axis of cylinder (See Figure 6-9).
r	Radius of the cylinder.
$q_x\ q_y\ q_z$	Any vector parallel to the axis of the cylinder.

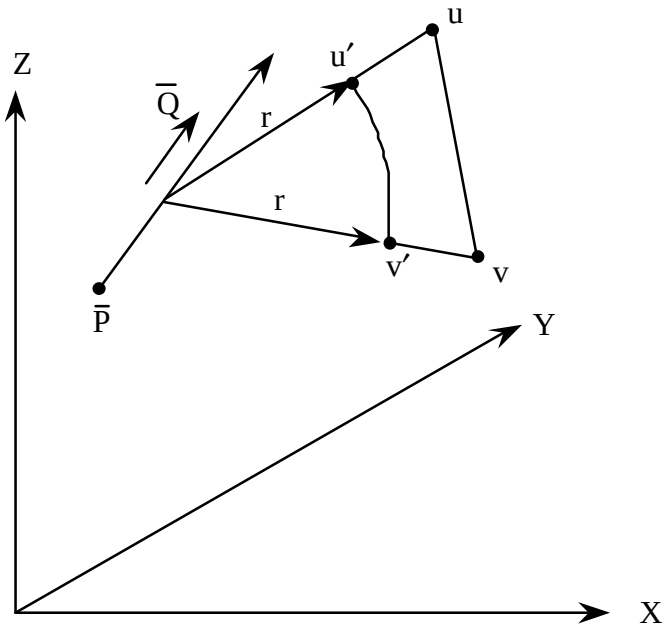


Figure 6-9. Cylindrical surface.

AUTO	Perform automatic smoothing of edges and surfaces which represent continuous surface definitions..
BG	Beam generation command.

The BG command permits beam elements to be defined within parts defined using an index space. If only beam elements are desired for the part, then all of the shell and solid elements can be deleted.

Options:

MT *m*	Beams have material number *m*.
SC *n*	Beams have section number *n*.
NGEN *n*	Generate n beams from point 1 to point 2.
N1 <Point>	Point 1 is located at <Point>
P1 p_x p_y p_z	Point 1 is located at (p_x, p_y, p_z)
N2 <Point>	Point 2 is located at <Point>
P2 p_x p_y p_z	Point 2 is located at (p_x, p_y, p_z).
B1 *n*	Set boundary code for point 1. *n* is a six digit binary number which specifies degrees of freedom which are to be constrained. Numbering the digits from left to right, they affect the following degree-of-freedom.

1st digit: *x*-displacement
=0: free
=1: fixed
2nd digit: *y*-displacement
3rd digit: *z*-displacement
4th digit: *x*-rotation
5th digit: *y*-rotation
6th digit: *z*-rotation

V2 v_x v_y v_z — Point 2 is offset from point 1 by the vector (v_x, v_y, v_z).

B2 *n* — Set boundary code for point 2. *n* is has the same meaning as for the "B1" option in this command.

NO <Point> — The point defining the orientation of the local 2-axis is located at <Point>.

PO p_x p_y p_z — The point defining the orientation of the local 2-axis is located at (p_x p_y p_z).

VOv_x v_y v_z — The local 2-axis is defined by vector (v_x, v_y, v_z).

CO p_x, p_y, p_z — Same as "PO" except the point is in cylindrical coordinates.

SO p_x, p_y, p_z — Same as "PO" except the point is in spherical coordinates.

; — Terminate this command.

BIAS x_0 y_0 z_0 v_x v_y v_z r_0 r_1 f — Bias mesh. This command is experimental. (x_0,y_0,z_0,v_x,v_y,v_z) represents a line towards which the elements are biased. A transition distance is defined beginning at r_0 and ending at r_1 and f is a factor for adjacent element scaling.

COOR nc data — Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.

CPL <Region> — Center points along line. If <Region> is a line, then this command forces elements to be equally spaced from the beginning point to the ending point. If <Region> is a surface or a volume then the command is subdivided into lines in the direction specified by the direction flag.

dir — Direction flag (do not input if <Region> is a line).
="*I*": Equal space along *I*-index
="*J*": Equal space along *J*-index
="*K*": Equal space along *K*-index

CYLI — Nodes are converted from cylindrical to rectangular coordinates. The equations for this

transformation are:
$X = R \cos \theta$
$Y = R \sin \theta$

D <Region> or — Region deletion keyword.

DI <Index Progression>

EQSP — Equal space along arc. This applies to the "AC" and "A" functions.

FIND <Point> exp1 exp2 exp3 exp4 — The FIND command places the generated coordinates of <Point> into the variables [cenx] [ceny] [cenz] and the node number into [node]. Four expressions must be input as part of this command.

Example:

```
FIND 1 2 1 [bp3x=cenx] [bp3y=ceny] [bp3z=cenz] [bp3n=node]
```

I, **J** or **K** <Point> — Specify independent variable for the function.
="*I*": coordinates vary as a function of the *I*-index.
=*J*": coordinates vary as a function of the *J*-index.
="*K*": coordinates vary as a function of the *K*-index.

dir2 — Flag specifying which coordinate is modified.
="*X*": *X*-coordinate is modified.
="*Y*": *Y*-coordinate is modified.
="*Z*": *Z*-coordinate is modified.

c_1 c_2 c_3 ... — New progression of coordinates along index *dir1*.

INT <Region> s_1 s_2 — Nodes within <Region> lie on the intersection of surface s_1 and s_2. Surfaces are defined using the "SD" command in the control section. These commands will be generated automatically if two "SF" commands result in an intersection surface in the index space.

LORI v_x v_y v_z — Specify local axis for orthotropic shell elements. The vector in the local part system is (v_x,v_y,v_z). The vector (v_x,v_y,v_z) may be specified as a function of the local *x*,*y*,*z* coordinates. For example:

LORI [-*y*] [*x*] 0

LREP l_1 l_2 ... l_n

Repeat part command. This command makes copies of the part in each of the local coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

MA or **MB**

Point functions. These commands are used to modify 1, 2, or 3 coordinates of groups of nodes.

<POINT>

For "MA" only.

<REGION>

For "MB" only.

n

Flag indicating which coordinates to change.
="*X*": *x* coordinate is changed
="*Y*": *y* coordinate
="*Z*": *z* coordinate
="*XY*": *x* and *y* coordinates
="*XZ*": *x* and *z* coordinates
="*YZ*": *y* and *z* coordinates
="*XYZ*": *x*, *y*, and *z* coordinates

d_x d_y d_z

New coordinates. Only the coordinates required by flag *n* need to be input. The new coordinates are added to the old coordinates.

MATE *m*

The part has material number *m*.

MS <Region>

Apply multiple surface equations to <Region>. This command permits the identification of parallel index planes for the purpose of applying surface equations. The function of this command is similar to the "SF" command; however, this command can result in considerable reduction in input for many common cases.

idir

<Region> is divided into a series of parallel planes normal to the axis in index space specified by idir.
="*I*": *I*-axis
="*J*": *J*-axis
="*K*": *K*-axis

Next, one surface equation must be input for each of the index planes in <Region> normal to the specified axis. One of the following options may be used.

Option 1:

sf_1	Data for first surface equation (See *Surface Definitions*).
sf_2	Data for second surface equation.
.	
.	
.	

Option 2:

PPX, **PPY**, or **PPZ**	Parallel planes normal to *x, y,* or *z* axes respectively.
u_1 u_2 u_3 ...	The point along the specified axis where the planes intercept. One value must be input for each plane.

Option 3:

CNSP	
p_x p_y p_z	Center of the spheres.
r_1 r_2 r_3 ...	Radii.

Option 4:

CNCY	
p_x p_y p_z	Any point on the axis of the cylinder.
v_x v_y v_z	Any vector parallel to the axis.
r_1 r_2 r_3 ...	Radii.

Option 5:

PON, **POX**, **POY**, **POZ**	Planes offset normal or in the *x, y,* or *z* direction, respectively.
p_x p_y p_z	Any point on the plane.
q_x q_y q_z	Any vector normal to the plane.
o_1 o_2 o_3 ...	Offsets in the requested direction.

MT <Region> or **MTI** <Index Progression>	Signifies material command.
mat	Material number.
MTV *m n*	All elements contained within volume definition

m are assigned material number *n*.

OR <Region> Specify orientation of local axes relative to the index space. This is necessary when orthotropic materials are used and/or if 8-node shells are requested.

l_1 Local *r* axis.
="*I*": *I*-axis
="*J*": *J*-axis
="*K*": *K*-axis

l_2 Local *s* axis.
="*I*": *I*-axis
="*J*": *J*-axis
="*K*": *K*-axis

ORDER *d1 d2 d3* *Order of writing nodes in index space (d = "I", "J", or "K").*

PA.i.Commands:PA; or PB.i.Commands:PB; *Point functions. These commands are used to modify 1, 2, or 3 coordinates of groups of nodes.*

<POINT> *For "PA" only.*

<REGION> *For "PB" only.*

n *Flag indicating which coordinates to change.*
="X": x coordinate is changed
="Y": y coordinate
="Z": z coordinate
="XY": x and y coordinates
="XZ": x and z coordinates
="YZ": y and z coordinates
="XYZ": x, y, and z coordinates

d_x d_y d_z New coordinates. Only the coordinates required by flag n need to be input. The old coordinates are replaced by the new coordinates.

REPE l_1 l_2 ...l_n Repeat command. This command makes copies of the part in each of the global coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

RES <Region> Use unequal element spacing.

idir Direction of sides to be operated on in <Region>.
="*I*": *I*-direction

	="*J*": *J*-direction ="*K*": *K*-direction
r	The ratio of the length of one element side to the next element side as the *I, J,* or *K* index increases is *r*.
REST name	Restore the nodal coordinates of the existing standard part from file "name". For complicated parts, this can save considerable amounts of computing. See also the SAVE command.
ROTATION p_x p_y p_z v_x v_y v_z w	Assign an initial rigid body rotation to all parts defined after this command. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per time unit.
RR <Region>	Rotate region.
data	Data for this command is described in the section on *Coordinate Transformations*.
SAVE name	Save the nodal coordinates of the existing standard part to file "name". For complicated parts, this can save considerable amounts of computing by using the REST command.
SF <Region> ityp -or- **SFI** <Index Progression> ityp	Surface command. This command allows for the exact equation specification for 3-D surfaces. The command operates by moving nodes from an initial location to the closest point on the surface. Intersections of surfaces in the index space are detected and calculated. Since intersections are rarely unique the user must define initial coordinates which are near the final configuration using the initial coordinates and/or point functions. This is often necessary for LS-INGRID to converge to the correct geometry. If a part is generated in cylindrical coordinates the surfaces are still assumed to be in rectangular coordinates. This permits non-axisymmetric surfaces to be generated on primarily axisymmetric parts. ityp= "SD *n*". If itype = "SD" then the surface is defined using the command "SD" in the control section. The surface name must be input to complete this option.

	ityp\o(=,/) "SD". ityp refers to an option in *Surface Definitions*. See *Surface Definitions* for the remaining input.
SFE <Region> dir ityp -or-	Surface command. These commands are similar to
SFEI <Index Progression> dir ityp	the and SFI commands. The primary difference is that only edges of blocks oriented in direction dir (dir = "*I*", "*J*" or "*K*") are projected onto the surface rather than all nodes within the region. ityp= "SD *n*". If itype = "SD" then the surface is defined using the command "SD" in the control section. The surface name must be input to complete this option. ityp\o(=,/) "SD". ityp refers to an option in *Surface Definitions*. See *Surface Definitions* for the remaining input.
SFV <Region> -or-	Surface command. These commands are similar to
SFVI <Index Progression>	the SF and SFI commands. The primary difference is that only vertices of blocks are projected to the nearest point on a surface rather than all nodes within a region. ityp= "SD *n*". If itype = "SD" then the surface is defined using the command "SD" in the control section. The surface name must be input to complete this option. ityp\o(=,/) "SD". ityp refers to an option in *Surface Definitions*. See *Surface Definitions* for the remaining input.
SPHE	Nodes are converted from spherical to rectangular coordinates. The equations for this transformation are: $X = R \cos\theta \sin\varphi$ $Y = R \sin\theta \sin\varphi$ $Z = R \cos\varphi$
TEMP *t*	The initial temperature of this part is *t* and it can be expressed as a function of *x, y, z* coordinates.
THIC *t*	Plates have the thickness *t* for this part. The thickness, *t*, may be specified as a function of the part local coordinates to permit thickness distributions.

TRI2 *t*

All quadrilateral shell elements in this part will be converted to triangular shells. The attached pressure segments, contact segments, etc. will remain as quadrilaterals.

TRIA *t*

All quadrilateral shell elements in this part will be converted to triangular shells. The attached pressure segments, contact segments, etc. will also be converted to triangles.

VELOCITY v_x v_y v_z

Assign initial rigid body velocity (V_x, V_y, V_z) to all
parts defined after this command. (V_x, V_y, V_z) can be expressed as a function of x, y, z coordinates to allow for velocity distributions.

VTSP

Equal space along chord. This applies to the "AC" and "A" functions.

7. Beam Part

Beam generation in LS-INGRID is performed by a special part. The data in the part is as follows:

BEAM
Local nodal point input.
0 (zero)
Element generation commands
0 (zero)
Optional functions
END

Local Node Point Input

Important vertices are listed in this section. All points in this section are assigned node numbers in the global system and output even if there are no elements connected. Nodes are input sequentially and assigned local node numbers starting from one. These numbers are used later for generating elements.

ctype	Coordinate transformation to be performed on nodal coordinates. ="RT": rectangular coordinates (no transformation) ="CY": cylindrical coordinates ="SP": spherical coordinates
n	*n* is a six digit binary number which specifies degrees of freedom to be constrained. Numbering the digits from left to right they affect the following degrees of freedom. 1st digit: *x*-displacement =0 free =1 fixed 2nd digit: *y*-displacement 3rd digit: *z*-displacement 4th digit: *x*-rotation 5th digit: *y*-rotation 6th digit: z-rotation
x_1 y_1 z_1	Nodal coordinates.

Element Generation Commands

is	First local node number in a beam sequence.
if	Last local node number in a beam sequence.
nel	Number of elements to be generated from is to if.
mat	Material number for the beams.
isect	Section property number for the beams.
normal	Third local node for defining the orientation of the beams. Note: this node can be moved by the "REPEAT" command and is not necessarily in global coordinates.

7.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M      C  Elements of material 2 are assigned to
                     C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data	Input nc global coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.

CYLI

Nodes are converted from cylindrical to rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta$

$Y = R \sin \theta$

LREP l_1 l_2 ... l_n

Repeat command. This command makes copies of the part in each of the local coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

REPE l_1 l_2 ...l_n

Repeat command. This command makes copies of the part in each of the global coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

ROTA p_x p_y p_z v_x v_y v_z w

Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per time unit.

SPHE

Nodes are converted from spherical to rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta \sin \varphi$

$Y = R \sin \theta \sin \varphi$

$Z = R \cos \varphi$

TEMP t

The initial temperature of this part is t and it can be expressed as a function of x, y, z coordinates.

VELO v_x v_y v_z

Assign an initial rigid body velocity to all nodeswithin this part. V_x, V_y, V_z is the global velocity vector and it can be expressed as a function of x, y, z coordinates.

8. Old Data Part

This part permits the user to input and manipulate models which were generated by other mesh generators. It can also be used to take old finite element models and update them The data in the part is as follows:

OLD
Commands
END

The "commands" include the input nodes and elements in the form of tables. These tables may be either in free format or formatted. Before a table is input, a list of keywords is input which tells what the columns correspond to. After the data is input, the part may then be moved or otherwise modified before inclusion with the rest of the LS-INGRID model.

NODES *n*	*n* nodal points are input.

Options:

BCND	LS-NIKE3D displacement boundary codes are input.
BCNR	LS-NIKE3D rotational boundary codes are input.
BCSP	SAP boundary codes are input.
DUMMY	Read and ignore this item. (Must be a number).
FORM *f*	Nodal points are read using format *f*. *f* is a character string up to 80 characters long which has the correct FORTRAN format. All items must be read in floating point format. No more than one node point can be specified on a card. If this option is not used then nodal point data is input free format.
INCLUDE	Nodes are read from *file*. This option terminates the NODES command and reads the nodes.
K	Node point increment *k* is input.
NUMBER	Node numbers are to be read. If this option is not used, then node numbers are assigned sequentially.
T	Temperature.

X	*X*-coordinate.
Y	*Y*-coordinate.
Z	*Z*-coordinate.
;	Terminate options and read the nodal points. This is done automatically if an include file is specified.

BEAMS *n* — *n* beam elements are input.

Options:

FORM *f*	Beam elements are read using format *f*. *f* is a character string up to 80 characters long which has the correct FORTRAN format. All items must be read in floating point format. No more than one element can be specified on a card. If this option is not used then nodal point data is input free format.
NUMBER	Element numbers are to be read. If this option is not used, then element numbers are assigned sequentially.
K	Element increment K is input.
MATERIAL	Material numbers are input.
SECTION	Section property numbers are input.
INCLUDE	Beam elements are read from *file*. This option terminates the BEAMS command and reads the beam elements.
NODES	Three nodes are input (first node, second node, and node defining local two axis).
N1	First node.
N2	Second node.
N3	Node defining local two axis.
DUMMY	Read and ignore this item. (Must be a number).
;	Terminate options and read the element data.

SHELLS *n* — *n* shell elements are input.

Options:

FORM *f*	Elements are read using format *f*. *f* is a

	character string up to 80 characters long which has the correct FORTRAN format. All items must be read in floating point format. No more than one element can be specified on a card. If this option is not used then nodal point data is input free format.
NUMBER	Element numbers are to be read. If this option is not used, then element numbers are assigned sequentially.
K	Element increment *K* is input.
MATERIAL	Material numbers are input.
THICKNESS	Thickness of element.
INCLUDE	Shell elements are read from *file*. This option terminates the SHELLS command and reads the shells.
NODES	Four nodes are input.
N1	Node 1.
N2	Node 2.
N3	Node 3.
N4	Node 4.
DUMMY	Read and ignore this item.
;	Terminate options and read the element data.

BRICKS *n* — *n* brick elements are input.

Options:

FORM *f*	Elements are read using format *f*. *f* is a character string up to 80 characters long which has the correct FORTRAN format. All items must be read in floating point format. No more than one element can be specified on a card. If this option is not used then nodal point data is input free format.
NUMBER	Element numbers are to be read. If this option is not used, then element numbers are assigned sequentially.
K	Element increment *K* is input.
MATERIAL	Material numbers are input.

INCLUDE	Brick elements are read from *file*. This option terminates the BRICKS command and reads the brick elements.
NODES	Input 8 node numbers.
N1	Node 1.
N2	Node 2.
N3	Node 3.
N4	Node 4.
N5	Node 5.
N6	Node 6.
N7	Node 7.
N8	Node 8.
DUMMY	Read and ignore this item.
;	Terminate option and read the element data.

8.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M        C  Elements of material 2 are assigned to
                       C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data — Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.

CYLI — Nodes are converted from cylindrical to rectangular coordinates. The equations are:

$X = R \cos \theta$

$Y = R \sin \theta$

LREP l_1 l_2 ... l_n ; — Repeat command. This command makes copies of the part in each of the **local** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

MATE matnum — The part has material number matnum.

REDUCE — Eliminate unattached nodes which are input in this part.

REPE l_1 l_2 ... l_n

Repeat command. This command makes copies of the part in each of the **global** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

ROTA p_x p_y p_z v_x v_y v_z w

Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per second.

SPHE

Nodes are converted from spherical to rectangular coordinates. The equations are:

$X = R \cos \theta \sin \varphi$

$Y = R \sin \theta \sin \varphi$

$Z = R \cos \varphi$

TEMP *t*

The initial temperature of this part is *t* and it can be expressed as a function of *x, y, z* coordinates.

THIC thic

Plates have the thickness thic for this part.

VELO v_x v_y v_z

Assign initial rigid body velocity to all nodes withinthis parts. (V_x, V_y, V_z) is the global velocity vector. (V_x, V_y, V_z) can be expressed as a function of *x, y, z* coordinates.

9. MAZE Part

MAZE parts provide simple methods for generating two dimensional cross sections. These sections can then be used as shell elements or as 3-D solids using drag mesh operations. The data in the part is as follows:

PART
Required part data (9.1)
Optional part control commands (9.2)
Optional functions (9.3)
END

9.1 Required Part Data

Each MAZE part requires a set of line definitions followed by a material number and mesh density information. There are many possible methods for describing MAZE parts as:

$L_1\ L_2\ L_3\ L_4\ mt\ k\ m$	Define four sided region edges consisting of the intersection lines L_1, L_2, L_3,and L_4. This region will have material name mt and will be subdivided in a $k\ x\ m$ element mesh with k elements lying along edges L_1 and L_3 and m elements lying along edges L_2 and L_4. Edges must be listed in a counterclockwise order. If k or m are zero, the number of elements are assumed to be one less than the number of points in lines L_1 or L_2, respectively. Points defining the lines then become nodal coordinates.
$L_1\ L_2\ L_3\ L_3\ mt\ k\ m$	Define three sided region having edges consisting of the intersecting lines L_1, L_2, and L_3. This region will have material name mt and will be subdivided into $m\ x\ (2k+m)$ elements with $k+m$ elements along edges L_1 and L_2 and $2m$ elements along edge L_3. Edges must be listed in a counterclockwise order.
$L_1\ L_2\ L_3\ 0\ mt\ k\ m$	Define three sided region having edges consisting of the intersecting lines L_1, L_2, and L_3. This region will have material name mt and will be subdivided into $m\ x\ (2k+m)$ elements with $k+m$ elements along edges L_1 and L_2 and $2m$ elements along edge L_3. Edges must be listed in a counterclockwise order.

$L_1\ L_2\ L_3\ L_4$ *mt -k-m* $r_1\ r_2$ *-or-* $L_1\ L_2\ L_3\ L_4$ *mt -k m* r_1 $L_1\ L_2\ L_3\ L_4$ *mt k-m* r_2	Define four sided region as described above but with variable zoning. Parameters r_1 and r_2 are the ratios of the first segment length to the last segment length along edges 1,3, and 2,4 respectively.
$L_1\ L_2\ L_3\ L_4$ *-mt k m* $r_1\ r_2\ r_3\ r_4$	Define four sided region as described above but with variable zoning. Parameters r_1 to r_4 are the ratios of the first segment length to the last segment length along edges 1 to 4, respectively.
-$L_1\ L_2\ L_3\ L_4$ *mt k m* $L_1\ L_2\ L_3\ L_4$ *mt k m* . . . etc.	Define four sided region as described above but with a specified number of elements between consecutive points defining the lines whose line numbers are proceeded by a minus sign. For this option to work properly, the first intersection point must lie either on the first and second point of the line being subdivided. The total number of points used to define the line is equal to *p*. If desired, not all subdivisions need to be defined. For example, if it is desired to specify the number of subdivisions between the first three points of the first line type. The other segments are equally spaced over the balance of the line.

9.2 OPTIONS AND FUNCTIONS

The following part control commands are allowed.

COOR nc data	Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.
DRAG	Perform a drag mesh operation to make solid elements from plane elements.
Options:	
MOVE *n* data	Form *n* layers of solid elements by moving the original plane elements to the new location specified by data. Data is described in detail in *Coordinate Transformations*.
ROTA $n\ p_x\ p_y\ p_z\ q_x\ q_y\ q_z\ \alpha$	Form *n* layers of solid elements by rotating the original plane elements about an axis. (p_x, p_y, p_z) is any point on the axis of rotation and (q_x, q_y, q_z) is a vector parallel to the axis. The angle of rotation in degrees is α.
RES *r*	The ratio of one element length to the next is *r*. This applies only to the previous drag operation.
;	Terminate this command.
LREP $l_1\ l_2\ ...\ l_n$	Repeat command. This command makes copies of the part in each of the local coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
REPE $l_1\ l_2\ ...\ l_n$	Repeat command. This command makes copies of the part in each of the global coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.
ROTA $p_x\ p_y\ p_z\ w_x\ w_y\ w_z$	Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (w_x, w_y, w_z) is the rotation vector in radians per second.
SPIN $n\ \theta$	Perform spin operation. The number of layers of nodes is *n* and the total angle of the part is q in degrees.

STACK *n* q	Perform stack operation. The number of layers of nodes is *n* and the total length of the part is θ.
TEMP *t*	The initial temperature of this part is *t* and it can be expressed as a function of *x, y, z* coordinates.
THICK *t*	Plates have the thickness *t* for this part.
TRI2 *t*	All quadrilateral shell elements in this part will be converted to triangular shells. The attached pressure segments, contact segments, etc. will remain as quadrilaterals.
TRIA *t*	All quadrilateral shell elements in this part will be converted to triangular shells. The attached pressure segments, contact segments, etc. will also be converted to triangles.
VELO v_x v_y v_z	Assign initial rigid body velocity to all nodes within this part. (V_x, V_y, V_z) is the global velocity vector and it can be expressed as a function of *x, y, z* coordinates.

9.3 FUNCTIONS

All MAZE part functions have the following form:

Keyword - index specification - parameters

MAZE parts have one type of index specification which is abbreviated as <Mregion>. The input for this index specification is as follows:

c_1 c_2 p_1 p_2	Four indices can identify any vertex, edge, or surface in the MAZE part. Each MAZE part has either 3 or 4 corners. The first corner is the intersection of the first line and the last line that makes up the part. The second corner is the intersection of the first and second lines. Further corners are defined similarly around the part. The part also has several planes including the original cross section and one more plane for each drag operation. The first corner node reference by <Mregion> is c_1 and the last corner is c_2. The first plane is p_1 and the last plane is p_2. If c_1 or c_2 is zero, they take on the minimum and maximum corner numbers respectively. Similarly, if p_1 or p_2 is zero, they are assigned the minimum and maximum plane numbers, respectively.

10. EDIT Part

The EDIT part allows loads and boundary conditions to be applied to previously defined parts. It also provides for the performing of system assembly operation from subsystems using system joint commands. The general for of this part is:

EDIT filename
<optional functions>
END

10.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M          C  Elements of material 2 are assigned to
                         C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data	Input nc local coordinate systems. Coordinate system data is described in detail in the section on *Coordinate Transformations*.
CYLI	Nodes are converted from cylindrical to

rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta$

$Y = R \sin \theta$

LREP l_1 l_2 ... l_n

Repeat command. This command makes copies of the part in each of the **local** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

MATE matnum

The part has material number matnum.

REPE l_1 l_2 ... l_n

Repeat command. This command makes copies of the part in each of the **global** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

ROTA p_x p_y p_z v_x v_y v_z w

Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per second.

SPHE

Nodes are converted from be to rectangular coordinates. The equations for these transformation are:

$X = R \cos \theta \sin \varphi$

$Y = R \sin \theta \sin \varphi$

$Z = R \cos \varphi$

TEMP t

The initial temperature of this part is t and it can be expressed as a function of x, y, z coordinates.

THIC thic

Plates have the thickness thic for this part.

VELO v_x v_y v_z

Assign initial rigid body velocity to all nodes within this parts. (V_x, V_y, V_z) is the global velocity vector. (V_x, V_y, V_z) can be expressed as a function of x, y, z coordinates.

11. DYNA3D Part

The DYNA3D part provides for importing existing DYNA3D input files into LS-INGRID. The form of the part is as follows:

For DYNA3D or LS-DYNA3D input files, use:

DYNA3D filename
<optional functions>
END

For VEC-DYNA3D input files, use:

VECDYNA filename
<optional functions>
END

filename is the name of the DYNA3D input file.

11.1 OPTIONS AND FUNCTIONS

Functions require the ability to identify groups of nodes and elements in a part and assign various properties. These have the general form of

Keyword <region> function data

Where <region> is a part specific description of where the function is to be applied. For the current part, the nodes or elements through either node or element numbers or through analytical expressions. As an example:

```
SI [mat==2] 1 M          C  Elements of material 2 are assigned to
                         C  the master side of contact interface 1.
```

Variables available for function application are as follows:

Variable	Description
x y z	Part local coordinates of node or element center.
xg yg zg	Global coordinates of node or element center.
node	Node number.
mat	Material number.
elem	Element number.

The following options are allowed in any order. Additional functions can be applied and are described in the section on *Loads and Boundary Conditions*.

COOR nc data — Input nc local coordinate systems. Coordinate

system data is described in detail in the section on *Coordinate Transformations*.

CYLI

Nodes are converted from cylindrical to rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta$

$Y = R \sin \theta$

LREP l_1 l_2 ... l_n

Repeat command. This command makes copies of the part in each of the **local** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

MATE *matnum*

The part has material number *matnum*.

REPE l_1 l_2 ... l_n

Repeat command. This command makes copies of the part in each of the **global** coordinate systems l_1 to l_n. If the coordinate system number is zero, the part is repeated with no transformation.

ROTA p_x p_y p_z v_x v_y v_z w

Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and (v_x, v_y, v_z) defines the axis direction. The angular velocity is w in radians per second.

SPHE

Nodes are converted from spherical to rectangular coordinates. The equations for this transformation are:

$X = R \cos \theta \sin \varphi$

$Y = R \sin \theta \sin \varphi$

$Z = R \cos \varphi$

TEMP *t*

The initial temperature of this part is *t* and it can be expressed as a function of *x, y, z* coordinates.

THIC *thic*

Plates have the thickness *thic* for this part.

VELO v_x v_y v_z

Assign initial rigid body velocity to all nodes within this parts. (V_x, V_y, V_z) is the global velocity vector. (V_x, V_y, V_z) can be expressed as a function of *x, y, z* coordinates.

12. Loads and Boundary Conditions

Loads and boundary can be applied as optional functions within any of the previously describe parts. The syntax of <Region> can be slightly different depending on which part it is applied in. Refer to the appropriate part for a description of the meaning of <Region>. Alternately, <Region> may be an expression. The local part coordinates for a node are stored as internal variables, ***x***, ***y*** and ***z***. The current global coordinates of the same node are **xg**, **yg** and **zg**. The node number is stored as variable **node**. Thus, to create slave nodes for sliding interface 6, the following command may be used:

SFC [(node<=55)$$(x>5.0)] 6

ACC <Region> lc amp f_x f_y f_z *-or-*
ACCI <Index Progression> lc amp f_x f_y f_z

Acceleration boundary condition. The load curve number is lc. amp is a scale factor and (f_x,f_y,f_z) indicates the load direction.

ACCE acc_name local_node

Definition of accelerometer *acc_name*. The local node numbers are 1 through 3.

Options:

N <Point>

The local node is defined by <Point>.

P p_x p_y p_z m

The local node is at point (p_x,p_y,p_z) in the local coordinate system. m is the rigid body number which is attached to the node.

B n

The local joint node has boundary constraint n. n is a six digit binary number which specifies degrees of freedom to be constrained. Numbering digits from left to right they affect the following degrees of freedom.

1st digit: *x*-displacement
=0 free
=1 fixed
2nd digit: *y*-displacement
3rd digit: *z*-displacement
4th digit: *x*-rotation
5th digit: *y*-rotation
6th digit: *z*-rotation

;

Terminate ACCE command.

B <Region> code

Boundary condition keyword. Code is a six digit binary number which specifies degrees of freedom to be constrained. Numbering the digits from left to right they affect the following

degrees of freedom
1st digit: *x*-displacement
=0: free
=1: fixed
2nd digit: *y*-displacement
3rd digit: *z*-displacement
4th digit: *x*-rotation
5th digit: *y*-rotation
6th digit: *z*-rotation

BELT *type belt_name local_node*

Seat belt definition. This command identifies local node number *local_node* for item *belt_name*. Values for *type* are as follows:
=RETRACTOR: Retractor definition.
=SENSOR: Sensor definition.
=SLIPRING: Slipring definition.

Options:

N <Point>

The local node is defined by <Point>.

P p_x p_y p_z *m*

The local node is at point (p_x,p_y,p_z) in the local coordinate system. *m* is the rigid body number which is attached to the node.

B *n*

The local node has boundary constraint *n*. *n* is a six digit binary number which specifies degrees of freedom to be constrained. Numbering digits from left to right they affect the following degrees of freedom.
1st digit: *x*-displacement
=0: free
=1: fixed
2nd digit: *y*-displacement
3rd digit: *z*-displacement
4th digit: *x*-rotation
5th digit: *y*-rotation
6th digit: z-rotation

;

Terminate BELT command.

CNV <Region> icv p_x p_y p_z *-or-*
CNVI <Index Progression> icv p_x p_y p_z

Control volume. This command defines segments for control volume number *icv*. The segments are facing towards point (p_x,p_y,p_z). (MVMA/DYNA3D, LS-910 and later).

CSE <Point> *n*

Identify elements for LS-DYNA3D cross-section resultant force calculations on interface *n*.

PO *i j k*

Use the element offset from <Point> by (*i*,*j*,*k*).

RA <Region>

Ignore <Point> and grab the elements identified

by <Region>.

RO *im jm km ix jx kx* — Use the block of elements offset from <Point>.

; — Terminate this command.

CSN <Region> *n* — Identify nodes for LS-DYNA3D cross-section resultant force calculations on interface *n*.

CSY <Region> *side* *-or-*
CSYI <Index Progression> *side* — Cyclic symmetry interface nodes. *side* can be MASTER or SLAVE. (VEC/DYNA3D, LS-920 and later).

CV <Region> lc_1 *h* lc_2 T_{inf} *a* *-or-*
CVI <Index Progression> lc_1 *h* lc_2 T_{inf} *a* — Convection boundary condition for surface segments. lc_1 is the load curve for the convection coefficient with scale factor *h*. lc_2 is the load curve for the ambient temperature with scale factor T_{in} . *a* is the exponent in the equation $q=h(T-T_{inf})^a$.

CVL <Region> lc_1 *h* lc_2 T_{inf} *a* — Convection boundary condition for edge segments.

EDR *i j k n* — Identify elements for deletion during restart number *n*. (*i*,*j*,*k*) are the minimum indices of the region in which elements are to be deleted.

Options:

RO i_m j_m k_m i_x j_x k_x — The elements to be deleted are the absolute indices $i_m<i<i_x$,$j_m<j<j_x$, $k_m<k<k_x$, but offset by (*i*,*j*,*k*).

; — Terminates this function.

EPB <Point> — Element Print Blocks.

PO *i j k* — Use the element offset from <Point> by (*i*,*j*,*k*).

RO i_m j_m k_m i_x j_x k_x — Use the block of elements offset from <Point>.

; — Terminate this command.

FC <Region> lc amp f_x f_y f_z *-or-*
FCI <Index Progression> lc amp f_x f_y f_z — Point force. The load curve number is lc. amp is a scale factor and (f_x,f_y,f_z) indicates the load direction.

FD <Region> lc amp f_x f_y f_z *-or-*
FDI <Index Progression> lc amp f_x f_y f_z — Displacement boundary condition. The load curve number is lc. amp is a scale factor and (f_x,f_y,f_z) indicates the load direction.

FIND <Point> expr1 expr2 expr3 expr4

The FIND command places the generated coordinates of <Point> into the variables [cenx] [ceny] [cenz] and the node number into [node]. Example:

FIND 1 2 1 [bp3x=cenx] [bp3y=ceny] [bp3z=cenz] [bp3n=node]

FL <Region> lc scal *-or-*
FLI <Index Progression> lc scal

Flux boundary condition.

FN <Region> e_{fail}

All nodes within <Region> are failure nodes and will fail at strain e_{fail}. Additional nodes are created, shell elements, pressure surfaces, and slide surfaces are renumbered to permit independent motion of adjacent elements (LS-DYNA3D).

Failure nodes are a simple method for allowing fracture. Each adjacent element has completely independent nodes. Groups of nodes are initially constrained to move together. When the average strain of adjacent elements reaches the failure strain, the constraint is eliminated and the elements separate.

FRV <Region> lc amp v_X v_Y v_Z

Set rotational velocity boundary conditions.

FT <Region> lc T *-or-*
FTI <Index Progression> *lc* T

Set temperature boundary condition to T and scaleby load curve *lc*.

FTB <Region> lc T Tbase *-or-*
FTBI <Index Progression> *lc* T Tbase

Set temperature boundary condition to T and scale by load curve *lc*. The scaling is:
(T-Tbase)*f(lc,time).

FV <Region> *lc amp* f_x f_y f_z *-or-*
FVI <Region> *lc amp* f_x f_y f_z

Velocity boundary condition. The load curve number is *lc*. *amp* is a scale factor and (f_x,f_y,f_z) indicates the load direction.

GEOC <Region> *igeo*

Geometric contact slave nodes. All identified nodes are slaved to geometric contact entity *igeo*. (LS-910 and later.)

IN <Region> *name* *-or-*
INI <Index Progression> *name*

Define nodes associated with component interface *name*. (LS-920 and later.)

IS <Region> *name* *-or-*
ISI <Index Progression> *name*

Define segments associated with component interface *name*. (LS-920 and later.)

JOY <Region> *-or-*

Identify JOY interface nodes. (JOY is an

	Eulerian
JOYI <Index Progression>	hydrocode which exists only at Lawrence Livermore National Laboratory).
JT	Joint command.
jn	Joint definition number.
l	Local node number (See Figure 2-1).
Options:	
N <Point>	The local joint node, *l*, is defined by <Point> in the index space.
P p_x p_y p_z *m*	The local joint node, *l*, is at point (p_x,p_y,p_z) in the local coordinate system. m is the rigid body number which is attached to the node.
INC *i*	Increment *jn* by *i* for each copy of the part. (default = 1).
B *n*	The local joint node has boundary constraint *n*. *n* is a six digit binary number which specifies degrees of freedom to be constrained. Numbering digits from left to right they affect the following degrees of freedom. 1st digit: *x*-displacement =0: free =1: fixed 2nd digit: *y*-displacement 3rd digit: *z*-displacement 4th digit: *x*-rotation 5th digit: *y*-rotation 6th digit: *z*-rotation
;	Terminate joint command.
MKI, **MKI+**, **MKI-** <Index Progression> *m* p_x p_y p_z *-or-* **MK**, **MK+**, **MK-** <Region> *m* p_x p_y p_z	Surfaces in <Region> or <Index Progression> are assigned marked surface number *m*. For MK+ and MKI+ the surface points toward this point. For MK- and MKI- the surface points away from this point. Do not input (p_x, p_y, p_z) for MK or MKI.
MKL <Region> *m*	Identify marked line number *m*.
N+ <Region> p_x p_y p_z *-or-* **NI+** <Region> p_x p_y p_z	Shell normal orientation command. (p_x,p_y,p_z) is a vector along the element normal vector.
N- <Region> p_x p_y p_z *-or-* **NI-** <Region> p_x p_y p_z	Shell normal orientation command. (p_x,p_y,p_z) is a vector reverse to the element normal vector.

NPB <Point> Nodal Print Block.

Options:

PO *i j k* Use the node offset from <Point> by (*i*,*j*,*k*).

RO $i_m\ j_m\ k_m\ i_x\ j_x\ k_x$ Use the block of nodes offset for <Point>.

; Terminate this command.

NFG <Region> *name* *-or-*
NFGI <Index Progression> *name*
Identify nodal force group *name*.

NRB <Region> *-or-*
NRBI <Index Progression>
Identify non-reflecting boundaries.

ORV Define orientation vectors. The orientation vectors are used to specify which axis is to be used for determining the effects of springs and dampers. This particular option is used when two nodes are required for specifying an axis slaved to a body.

jn Orientation vector number.
l Local node number (either 1 or 2).

Options:

N <Point> The local orientation vector node, *l*, is defined by <Point> in the index space.

P $p_x\ p_y\ p_z\ m$ The local orientation vector node, *l*, is at point (p_x,p_y,p_z) in the local coordinate system. *m* is the rigid body number which is attached to the node.

INC *i* Increment *jn* by *i* for each copy of the part. (default = 1).

B *n* The local orientation vector node has boundary constraint *n*. *n* is a six digit binary number which specifies degree of freedom to be constrained. Numbering digits from left to right they affect the following degrees of freedom.

1st digit: *x*-displacement
=0: free
=1: fixed
2nd digit: *y*-displacement
3rd digit: *z*-displacement
4th digit: *x*-rotation
5th digit: *y*-rotation
6th digit: *z*-rotation

; Terminate ORV command.

PM <Region> *m*	All nodes within <Region> have mass *m* attached to them.
PR <Region> *-or-*	Signifies pressure load command for surface
PRI <Index Progression>	segments.
lc	Load curve or load curve number.
p	Pressure magnitude. Spatial variations may be obtained by inputting *p* as a function of global coordinates (*x,y,z*).
a_x a_y a_z	(a_x,a_y,a_z) is a point in the local coordinate system toward which the pressure acts. By specifying (a_x,a_y,a_z) LS-INGRID knows in which direction the pressure is acting and numbers the pressure card node accordingly.
PRL <Region>	Signifies pressure load command for edge segments.
lc	Load curve or load curve number.
p	Pressure magnitude.
a_x a_y a_z	(a_x,a_y,a_z) is a point in the local coordinate system toward which the pressure acts. By specifying (a_x,a_y,a_z) LS-INGRID knows in which direction the pressure is acting and numbers the pressure card node accordingly.
RB <Region> lc1 f lc2 t_{inf} *-or-* **RBI** <Index Progression> lc1 f lc2 t_{inf}	Radiation boundary condition.
RBN <Region> set_name	Assign nodes to rigid body node set *set_name*.
RE <Region> lc1 T flag *-or-*	Define a radiation enclosure. *lc1* is a load curve for emissivity. The temperature of the segment is T if lc1=0. flag is "YES" if this surface is to be included in obstructing surface calculations and "NO" otherwise.

Note: If this option is used, segments are oriented so that they face outward from the adjacent conduction elements. An error occurs if radiation segments defined by this command are not adjacent to a conduction element because the outward normal would be indeterminate.

RE+ <Region> lc1 T flag p_x p_y p_z *-or-* **RE-** <Region> lc1 T flag p_x p_y p_z	Define a radiation enclosure. *lc1* is a load curve for emissivity. The temperature of the segment is T if lc1=0. flag is "YES" if this surface is to be included in obstructing surface calculations

and "NO" otherwise. (p_x,p_y,p_z) is a point in the local coordinate system toward which the radiation occurs (RE+), or facing the opposite direction
(RE-).

RXN <Region> mat *-or-*
RXNI <Index Progression> mat — Extra nodes for rigid body of material mat.

SBI <Region> — Define a shell/brick interface. <Region> must be a point or a line in the index space.side
="*b*": We are identifying nodes on the brick side of the interface.
="*s*": We are identifying nodes on the shell side of the interface.

idir — Nodes on an interface are in a line parallel to axis idir.
=*I:* *I*-axis
=*J:* *J*-axis
=*K*: *K*-axis

ioff1 ioff2 — Increments for determining the nodes to be selected along direction *dir*.

SC <Region> idir*{options}* — ALE smoothing constraints. idir Smoothing constraints are generated along the line defined by axis *idir*.
=*I:* *I*-axis
=*J:* *J*-axis
=*K*: *K*-axis

Options:

1A i j k — The first point of the smoothing constraint is located at absolute indices (i,j,k).

1R i j k — The first point of the smoothing constraint is located at absolute indices (i,j,k).

2A i j k — The last point of the smoothing constraint is located at absolute indices (i,j,k).

2R i j k — The last point of the smoothing constraint is located at absolute indices (i,j,k).

PRE — Constraints are applied before ALE iterative smoothing is done. (The default requires that the constraints be performed after the smoothing is done.)

; — Terminate smoothing constraint command.

SFC <Region> *n*	Identify slave nodes for sliding interface *n*. This is used for interfaces which involve nodes impacting surfaces or to make more precise distinctions between master and slave sides for the merging algorithms.
SI <Region> *-or-* **SII** <Index Progression>	Identify sliding interfaces.
islid	Sliding interface number
mslid	Master/slave flag ="*M*": master surface ="*S*": slave surface
SI+ <Region> *-or-* **SII+** <Index Progression>	Identify sliding interface.
islid1	Sliding interface number
mslid	Master/slave flag. ="*M*": master surface. ="*S*": slave surface.
$p_x\ p_y\ p_z$	A point in the local coordinate system toward which the sliding interface faces.
SI- <Region> *-or-* **SII-** <Index Progression>	Signifies sliding interface command.
islid1	Sliding interface number.
mslid	Master/slave flag. ="*M*": master surface. ="*S*": slave surface.
$p_x\ p_y\ p_z$	A point in the local coordinate system which the sliding interface faces away from.
SL <Region> *n* isid	Define nodes on slide line *n*. <Region> should be a line in the index space. *isid* is either "master" or "slave". This command is sometimes useful in conjunction with SI to fix node tolerance problems.
SPC <Region> *name xyzxyz* *-or-* **SPCI** <Index Progression> *name xyzxyz*	Single point constraints to plane *name*. *xyzxyz* is a binary number which is zero for an unconstrained degree-of-freedom and 1 for a constrained degree-of-freedom. The left three digits are for the translational dof's and the right three are for the rotational.
SPDP <Region> {options} *n*1 isid	Define springs or dampers on all nodes within

	<Region>. They behave according to spring/damper definition *n*. *isid* is used to force nodes to be on opposite side of the definition. isid="*m*" for the master side and "*s*" for the slave side. {options} are as follows:
ORV *n*	This spring/damper acts along orientation vector *n*.
POFF	Turn element printing off (default).
PON	Turn element printing on.
SCAL *s*	The spring/damper force is scaled by *s*.
SW <Region> *n* -*or*- **SWI** <Index Progression> *n*	Slave nodes to stonewall number *n*.
SYSJ	Joint command.
jn	Joint definition name.
l	Local node number. Nodes 1 through 3 define the local system for the master side of a joint. Nodes 4 through 6 define the slave side of a joint.

Options:

INC *i*	Increment *jn* by *i* for each copy of the part. (default = 1).
B *n*	The local joint node has boundary constraint *n*. *n* is a six digit binary number which specifies degrees of freedom to be constrained. Numbering digits from left to right they affect the following degrees of freedom. 1st digit: *x*-displacement =0: free =1: fixed 2nd digit: *y*-displacement 3rd digit: *z*-displacement 4th digit: *x*-rotation 5th digit: *y*-rotation 6th digit: *z*-rotation
N <Point>	The local joint node, *l*, is defined by <Point> in the index space.
P p_x p_y p_z *m*	The local joint node, *l*, is at point (p_x,p_y,p_z) in the local coordinate system. *m* is the rigid body number which is attached to the node.

SYSTEM name	The local joint node, *l*, is defined for system name. (Default is the current active system.)
;	Terminate joint command.
TH <Region> *-or-* **THI** <Index Progression> thick	Specify thickness command. Thickness of plates within the region.
TM <Region> *t* *-or-* **TMI** <Index Progression> *t*	Set initial temperature (TOPAZ) or steady State temperature (DYNA/NIKE) to *t*.
TN <Region> *n* *-or-* **TNI** <Index Progression> *n*	Set thickness number to *n*.
VE <Region>	Set edge visibility on for outline and phantom edge plotting. <Region> must be a line in the reduced index space.

13. Interactive Commands

After the model is generated, LS-INGRID enters the interactive graphics phase of the program. The *x*-axis in screen coordinates is fixed relative to the screen and extends horizontally to the viewers right. The *y*-axis is positive up. The *z*-axis extends out of the screen towards the viewer.

The following commands are allowed in this phase.

AJNP $p_x\ p_y\ p_z$	Print the nodal point which is nearest to point (p_x,p_y,p_z).
AM $m_1\ m_2$...,	Add materials m_1, m_2,... (by number) to the active list.
AMN $m_1\ m_2$...,	Add materials m_1, m_2,... (by name) to the active list.
AP $p_1\ p_2$ ...;	Add parts p_1, p_2, ... to the active list.
ARROW	Toggle arrow plotting on or off. This allows the direction of the tool path to be visualized.
ASCII	Read ASCII tracer particle file.
BPTOL $p_1\ p_2\ t$	The tolerance to be used when merging part p_1 to p_2 is t.
CCEN	Select the center of the picture using the mouse.
CCOL *i r g b*	Change the (red,green,blue) values of color number *i* to (*r*,*g*,*b*).
CENT	Moments and products of inertia are determined relative to the centroid and global axes (default).
CHUE *m h*	Change hue of material *m* to *h*.
CONT	Finish model generation and format the output file.
CSAT *m s*	Change saturation of material *m* to *s*.
D *x*	Move down distance *x* relative to the structure.
DI ACCE	Display accelerometers.
DI BELT	Display seat belts.

DI CNV *n*	Display DYNA3D control volume *n*.
DI CSEC *n*	Display DYNA3D force output cross section *n*.
DI CSYM *n*	Display cyclic symmetry boundary conditions.
DI CV	Display convection boundary condition. (surface segments).
DI CVL	Display convection boundary condition. (edge segments).
DI D *lc*	Display forced displacements associated with load case (or load curve) *lc*.
DI DETP	Display detonation points.
DI DS n_1 n_2... ;	Display digitized surface definitions n_1, n_2...
DI DSAD n_1 n_2... ;	Display digitized surfaces and add definitions n_1, n_2 ... to the display list.
DI DSRM n_1 n_2 ... ;	Display digitized surfaces and remove definitions n_1, n_2 ... from the display list.
DI DX	Display *X*-translational boundary conditions.
DI DY	Display *Y*-translational boundary conditions.
DI DZ	Display *Z*-translational boundary conditions..
DI EDR *n*	Display elements to be deleted on restart *n*.
DI EPB	Display element print blocks.
DI F *lc*	Display point loads associated with load case (or load curve) *lc*.
DI FL	Display flux boundary conditions; (edge segments).
DI FLUX	Display flux boundary condition; (surface segments).
DI FSYM	Display failing symmetry planes.
DI INTF *name*	Display component substructure *name*.
DI JOY	Display joy interface nodes.
DI JTS	Display joints.
DI L3D	Display three-dimensional line definitions.

DI LAX code

Display local axes.
code = "*R*": local *R*-axis
code = "*S*": local *S*-axis
code = "*T*": local *T*-axis
code = "*RS*": local *RS*-axes
code = "*ST*": local *ST*-axes
code = "*TR*": local *TR*-axes
code = "*RST*": local *RST*-axes
code = "*CORI*": local composite angles

DI M m_1 m_2 ... ;

Materials m_1, m_2, ... are to be highlighted during plotting.

DI MCG *m*

Display mass properties of individual materials.

DI MK *m*

Display marked surface *m*.

DI NCV

Display NURB curves.

DI NFG

Display nodal force groups.

DI NRB

Display non-reflecting boundaries.

DI NPB

Display nodal print blocks.

DI NSF

Display NURB surfaces.

DI NV

Display shell element normal vectors.

DI ORV

Display orientation vectors.

DI OUTL

Display free edges of shells.

DI P p_1 p_2 ... ;

Parts p_1, p_2, ... are to be highlighted during plotting.

DI PL *lc*

Display pressure surfaces associated with load case (or load curve) *lc*; (edge segments).

DI PM

Display point masses.

DI PR *lc*

Display pressure surfaces associated with load case (or load curve) *lc*; (surface segments).

DI PV *n*

Display tool path *n*.

DI RB

Display radiation boundary conditions. (surface segments).

DI RBL

Display radiation boundary conditions; (edge segments).

DI RBN

Display nodal rigid bodies. (LS-910 and later).

DI RE	Display radiation enclosure; (surface segments).
DI REL	Display radiation enclosure. (edge segments)
DI RX	Display *X*-rotational boundary conditions.
DI RXN *m*	Display extra nodes slaved to rigid body material *m*.
DI RY	Display *Y*-rotational boundary conditions.
DI RZ	Display *Z*-rotational boundary conditions.
DI SBI	Display shell/brick interfaces.
DI SFC islid mslid	Display nodes which are part of sliding interface definition islid. mslid = "*M*": display master side. mslid = "*S*": display slave side. mslid = "*B*": display both sides.
DI SI islid mslid	Display slide surface islid; (surface segments). mslid = "*M*": display master side mslid = "*S*": display slave side mslid = "*B*": display both sides
DI SL *n* isid	Display slide line *n*; (edge segments). mslid = "*M*": display master side. mslid = "*S"*: display slave side. mslid = "*B*": display both sides.
DI SPC	Display single point constraints.
DI SPD	Display springs and dampers.
DI SW *s*	Display stonewall *s*.
DI SY isym	Display symmetry plane isym.
DI SYSJ isym	Display symmetry plane isym.
DI TB	Display temperature boundary conditions.
DI TI	Display temperature initial conditions.
DI TRACER	Display tracer particles.
DI VB *lc*	Display velocity boundary conditions associated with load curve *lc*.
DI VECT *c*	Display vectors of component *c*. *c* = IV: initial velocity *c* = IR: initial rotational velocity *c* = IDV: initial distortional velocity

	c = IRB: initial rigid body velocity *c* = V: current velocity *c* = DV: current distortional velocity *c* = RB: current rigid body velocity *c* = A: current acceleration
DI WARP *ang*	Display shells with warp angles that exceed *ang*.
DIAD	This can be used in any of the above commands instead of DI. If this is used, then the display request is in addition to the previous ones rather than replacing them.
DICOL l	Following DI and DIAD options are to be performed using color number l. Valid numbers for l are 1 through 7.
DIOFF	Turn off display options.
DM m_1 m_2...,	Delete materials m_1, m_2, ... (by number) from active display list.
DMN m_1 m_2...,	Delete materials m_1, m_2, ... (by name) from active display list.
DMEM	Dump memory allocations statistics.
DRAW	Draw the mesh. All mesh lines are plotted.
DSV	View three-dimensional digitized surfaces.
DSVS d_1, d_2, ...d_n ;	View digitized surfaces d_1, d_2 ...
DSAD d_1, d_2, ...d_n ;	Add digitized surfaces d_1, d_2 ... to the active display list.
DSRM d_1, d_2, ...d_n ;	Remove digitized surfaces d_1, d_2 ... from the active display list.
ELPLT on/off	Turn element number plotting on or off. The default is off.
EXIT	Exit LS-INGRID now.
FOLD *n d*	Airbag folding. All nodes of the model are included in the fold operation. Fold definitions 1 through *n* are applied in ascending order. *d* is an optional thickness which can be used to increase the fold thicknesses. The maximum of *d* and the fold definition specified thickness is used.
FRAME	Reference frame with tick marks plotted

	(default).
GRID	Displays will be overlaid by a grid of orthogonal lines. (Two dimensional plots only).
INFO	Print information on the mesh size.
L x	Move left a distance x relative to the structure.
L3V	View three dimensional digitized surfaces.
L3VS l_1, l_2, ...l_n ;	View all three-dimensional lines l_1, l_2, ...
LCV n	View load curve n.
LIGHT p_x p_y p_z	Locate the light source for continuous color plots at (p_x,p_y,p_z).
LMIN l	Set minimum luminosity for continuous color plots to l.
LSIZE	On/off switch for printing extent of active three-dimensional line definitions during plotting.
LV	Display all two-dimensional line definitions.
LVI m n	Display lines m to n.
LVS l_1, l_2, ... l_n ;	Display lines l_1, l_2, ...
M m_1 m_2 ...,	Display materials m_1, m_2, ... (by number).
MCOL	Color plots based on element materials. (See also PCOL).
MMASS	This is the same as TMASS except that the calculation is only performed for the active materials.
MN m_1 m_2 ...,	Display materials m_1, m_2, ... (by name).
MPLT	Plot mass properties on screen.
MSIZ	Print the extent of the current active material subset.
NCV ;	View all NURB curves.
NCV d_1, d_2, ... d_n ;	View NURB curves d_1, d_2 ...
NCAD d_1, d_2, ... d_n ;	Add NURB curves d_1, d_2 ... to the active display list.

NCRM d_1, d_2, ... d_n ;	Remove NURB curves d_1, d_2 ... from the active display list.
NDPLT on/off	Turn node number plotting on or off. The default is off.
NOFRAME	No reference frame is plotted.
NOGRID	Displays will not be overlaid by a grid of orthogonal lines. (Default).
NSET *n x y z*	Set the coordinates of node *n* to (*x*,*y*,*z*).
NSV ;	View all NURB surfaces.
NSV d_1, d_2, ...d_n ;	View NURB surfaces d_1, d_2 ...
NSAD d_1, d_2, ... d_n ;	Add NURB surfaces d_1, d_2 ... to the active display list.
NSRM d_1, d_2, ... d_n ;	Remove NURB surfaces d_1, d_2 ... from the active display list.
OVERLAY *n x y z*	Stop screen erasing of previous picture so that the next picture is overlaid.
P p_1 p_2 ... ;	Display parts p_1, p_2, ... To display all parts simply type "*P* ;". The ";" is also optional so that the command "P 1 VIEW" would show part one on the screen.
PCHK	Turn on checking of penetrations in the single surface contact algorithms. Currently, this is only designed to work with the airbag folding capability. Repeating this command will turn the option off. Penetrations are graphically displayed.
PCOL	Color plots based on element parts. This is the default. (See also MCOL).
PFOLD *n d* p_1 p_2	This is the same as the FOLD command except that only parts p_1 through p_2 are treated rather than the entire mesh.
PINF	Print information on each part.
PMASS	This is the same as TMASS except that the calculation is only performed for the active parts.
POOR	Poor man's hidden line algorithm.

PRINT *v*	Echo the value of *v* back to the terminal. This is most frequently used with the calculator program. (e.g. "PRINT [SQRT(27)*24.3]")
PSRGB	Create a RGB Postscript file.
PTOL *n t*	Set the tolerance for part *n* to *t*. See also "*T*" and "*TP*".
PV	View tool paths.
PVS p_1 p_2 ... p_n ;	View tool paths p_1, p_2, ...
QUIT	Quit LS-INGRID now.
R *x*	Move right a distance *x* relative to the structure.
REDUCE	Eliminate exterior faces which have become interior faces due to the tolerance command.
REFP r_x r_y r_z	Moments and products of inertia are determined relative to the point (r_x,r_y,r_z) and global axes.
RESO ires	Set the Z-buffer resolution to ires for the VIEW command. ires is limited to one of 256, 512, 1024, 2048, 4096, 8192. (The default is 1024).
REST	Restore all rotations to their initial settings.

Note: The local coordinates are fixed to the model and rotate as the model rotates.

RLX θ	Rotate the body θ degrees about the local *x*-axis.
RLY θ	Rotate the body θ degrees about the local *y*-axis.
RLZ θ	Rotate the body θ degrees about the local *z*-axis.
RM m_1 m_2 ... ,	Remove materials m_1, m_2, ... (by number) from the active list.
RMN m_1 m_2 ... ,	Remove materials m_1, m_2, ... (by name) from the active list.
RP p_1 p_2 ... ;	Remove parts p_1, p_2, ... from the active list.
RX θ	Rotate body θ degrees about the *x*-axis in the screen coordinates. A positive rotation is counterclockwise.
RY θ	Rotate body θ degrees about the *y*-axis in the screen coordinates. A positive rotation is counterclockwise.

RZ θ	Rotate body θ degrees about the *z*-axis in the screen coordinates. A positive rotation is counterclockwise.
SCALE *s*	Multiply the mesh size by *s*. Default is 1.0.
SCOL	Color plots based on system name; (see also MCOL and PCOL).
SEAL name	Seal airbag edges which are marked with name.
SEAL CIRCLE	Seal the airbag periphery. The airbag mesh is assumed to be circular in the *x*-*y* plane and centered along the *z*-axis at *z*=0; (default).
SEAL OFF	Turn off airbag sealing options.
SEAL OUTLINE	Seal the free edges of an airbag.
SHRINK *s*	Shrink individual elements by *s* when plotting. This is used to see if there are any holes in the mesh.
SIZE	Print the range of coordinates in the current active part list.
STOP	Exit the program immediately.
T tol	Remove duplicate nodes within a distance tol. This command will not eliminate coincident nodes on opposite sides of slide surfaces or joints.
TMASS	Calculate the total mass of the model. Mass densities must be input using the MAT command. This command also calculates kinetic energy, linear momentum, volume, moments of inertia, and the centroid.
TP tol	Remove duplicate nodes within a distance tol and print the number of nodes merged between any two parts.
TRIAD on/off	Show the coordinate system triad on the screen when doing three-dimensional plots. The default is on.
TRPT	On/off flag for printing timing statistics from plot commands.
TTIME	Plot time histories of tracer particles. (See also ASCII TRACER.) comp is one of the following components: = SX --> σ_{xx}

	= SY --> σ_{yy}
	= SZ --> σ_{zz}
	= SXY --> σ_{xy}
	= SYZ --> σ_{yz}
	= SZX --> σ_{zx}
	= P --> Pressure
	= EFP --> Effective Plastic Strain
	= RHO --> ρ
	= RVOL --> Relative Volume
TV *n*	Select graphics device *n*. Available graphics devices are dependent on the installation. When typing this option, LS-INGRID will prompt the user for the correct device and provide a list of available devices.
U *x*	Move up a distance *x* relative to the structure.
UPDATE	Re-read the LS-INGRID input deck and return to the interactive phase for continued plotting.
VEOS *n* V_1 V_2	View equation of state for material n from relative volume V_1 to relative volume V_2.
VIEW or **G**	View the mesh. An algorithm based on a *z*-buffer method is used for hidden line processing.
WBGR	Write the boundary grid into the LS-INGRID output file. For a solid grid, all internal polygons are removed and the external polygons are written out.
WBIF	Write the boundary information file, grfinfo for use by POST.
WRDB	Write reduced TAURUS database. This option only writes out surface polygons.
WTDB	Write TAURUS database.
ZIN	Zoom in on the picture by selecting the upper and lower corners with the mouse.
ZOUT	Zoom out on the picture by selecting the upper and lower corners with the mouse.

13.1 Exploded View Commands

Exploded view commands permit collections of parts or materials to be moved from their generated locations. Exploding a model will affect the graphics and mass property calculations, but will not affect the output computational model. Exploding a model with respect to parts will only affect the TMASS and PMASS commands, while exploding with respect to materials will only affect the MMASS command.

MEXP	Exploded views are performed with respect to materials. This command is automatically invoked by all other material explode commands.
MLOC data	Set position of material subset to the position specified in data. Data is described in the section on *Coordinate Transformations*.
MMOV data	Shift the position of material subset by the transformation specified in data. Data is described in the section on *Coordinate Transformations*.
MSEL m_1 m_2 ... ;	Select material subset m_1, m_2, ... ; for explode operations.
NEXP	Turn off exploded view option.
PEXP	Exploded views are performed with respect to parts. This command is automatically invoked by all other part explode commands.
PLOC data	Set position of part subset to the position specified in data. Data is described in the section on *Coordinate Transformations*.
PMOV data	Shift the position of part subset by the transformation specified in data. Data is described in the section on *Coordinate Transformations*.
PSEL p_1, p_2, ... ;	Select part subset p_1, p_2, ... ; for explode operations.

13.2 TAURUS/Post-Processing Commands

The post-processing phase of LS-INGRID allows for the generated models to be displayed in their deformed configurations with a variety of boundary conditions or other graphical information superimposed. Some time history facilities are also included.

TAURUS file	Read TAURUS database file.
ALAB label	Set abscissa label.
ALAB OFF	Use default abscissa label.
ASCL scale	Set abscissa scale factor.
ASET min max	Set abscissa range.
DSF *d*	Set displacement scale factor to *d*; (default=1.0).
DTS s_1 s_2	Delete states s_1 through s_2.
GTIME comp ;	Plot global component comp. Components are defined in table 13.2.1.
LS	Last state.
LTS	List time states.
NTIME comp n_1 n_2 ... ;	Plot nodal component comp for nodes n_1, n_2, ... Components are defined in table 13.2.2.
NTIME comp ;	Plot nodal component comp for the previously used nodal list. Components are defined in table 13.2.2.
OLAB label	Set ordinate label.
OLAB OFF	Use default ordinate label.
OSCL scale	Set ordinate scale factor.
OSET min max	Set ordinate range.
REP s_1 s_2 *k*	Execute the next plot command from states s_1 to s_2 by increment state increment *k*.
S *n*	Select state number *n*.
S 0	Plot the generated geometry.
SI *n*	Increment the state number by *n*.

UDEF *n* The undeformed state is number *n*; (default=1).

TABLE 13.2.1
GLOBAL TIME HISTORY COMPONENTS

ENERGY	Plot the total, kinetic and internal enegy.
MOMENTUM	Plot the momentum vector.
XVEL	*X*-momentum/total mass.
YVEL	*Y*-momentum/total mass.
ZVEL	*Z*-momentum/total mass.
INTERNAL	Internal enegy.
KINETIC	Kinetic enegy.
TOTAL	Total enegy.

TABLE 13.2.2
VECTOR PLOT COMPONENTS

A	Current acceleration.
DV	Current distortional velocity.
IDV	Initial distortional velocity.
IR	Initial rotational velocity.
IRB	Initial rigid body velocity.
IV	Initial velocity.
RB	Current rigid body velocity.
V	Current velocity.

TABLE 13.2.3
NODAL TIME HISTORY COMPONENTS

Rectangular coordinates:	
X	*X*-position
Y	*Y*-position
Z	*Z*-position
AX	*X*-acceleration
AY	*Y*-acceleration
AZ	*Z*-acceleration
DX	*X*-displacement
DY	*Y*-displacement
DZ	*Z*-displacement
VX	*X*-velocity
VY	*Y*-velocity
VZ	*Z*-velocity
Cylindrical coordinates:	
CR	Radial-position
CT	Circumfirential-position
CZ	Axial-position.
CAR	Radial-acceleration
CAT	Circumfirential-acceleration
CAZ	Axial-acceleration
CDR	Radial-displacement
CDT	Circumfirential-displacement
CDZ	Axial-displacement
CVR	Radial-velocity
CVT	Circumfirential-velocity
CVZ	Axial-velocity
Spherical coordinates:	
SR	Radial-position
ST	Theta-position
SP	Phi-position
SAR	Radial-acceleration
SAT	Theta-acceleration
SAP	Phi-acceleration
SDR	Radial-displacement
SDT	Theta-displacement
SDP	Phi-displacement
SVR	Radial-velocity
SVT	Theta-velocity
SVP	Phi-velocity
Special components:	
TEMP	Temperature
TIME	Time
TOTP	Total position
TOTA	Total acceleration
TOTD	Total displacement
TOTV	Total velocity

14. Two-Dimensional Line Definitions

Two-dimensional line definitions are lists of *r,z* (*x,y*) points which form a piecewise linear curve. Each line definitions has a number.

LAD r_c z_c t	Define a circular arc centered at point (r_c,z_c), beginning at the last point defined and sweeping through t degrees. Positive *t* is assumed to be counterclockwise.
LADD l_1 sl_2 t	Define line definition *l* as a linear combination of line definitions l_1 and l_2. $l = s* l_1 + t * l_2$.
LADV *ldr dz*	Add vector (*dr,dz*) to line definition *l*.
LAP r_1 z_1 r_c z_c	Define a circular arc by specifying points. The arc is assumed to begin at the last point defined and to end at (r_1,z_1). Point (r_c,z_c) lies at the center of the arc.
LAR r_1 z_1 R	Define a circular arc by specifying radius. An arc of radius *R* is assumed to begin at the last point defined and to end at (r_1,z_1). If *R* is positive, the center of the arc lies to the left as one moves from the last point defined to (r_1,z_1). If *R* is negative, the center of the arc will be to the right.
LAT r_1 z_1 r_2 z_2 R	Define a circular arc of radius *R* tangent to the last line segment and a line segment joining point (r_1,z_1) to point (r_2,z_2). These line segments will be automatically extended or truncated at the tangency point.
LCC n r_c z_c t_1 t_2 r_1 ... r_n	Define *n* lines consisting of circular arcs centered at point (r_c,z_c) that sweep from angle t_1 to t_2. r_1 r_2, ... r_n are the radii of the next *n* lines. Line numbers are assigned by LS-INGRID beginning with the next available number.
LD *n*	Begin line definition *n*.

LEP $a\ b\ r_c\ z_c\ t_1\ t_2\ p$

Define an elliptical arc by the semi-major and semi-minor axes, a and b respectively, the center point (r_c,z_c), and a sweep from angle t_1 to angle t_2 as measured from the major axis. Angle p is the angle between the major axis and the r-axis. A circular arc is generated by setting $a=b$. Positive angles represent counterclockwise rotations.

LEXP $r(s)\ z(s)\ n$

Define a line definition using expression $r(s)$ and $z(s)$ where $0 \leq s \leq 1$. The number of points to be used is n.

LINT $l_1\ l_2\ s$

The current line definition is formed by interpolation between line definition l_1 and line definition l_2. The equation is $l = s*l_1 + (1-s) * l_2$.

LO $m\ r_1\ z_1\ r_2\ z_2$

Define a line segment for line n by offsetting a segment of line m such that the first point of the new segment begins at (r_1,z_1) and the last point terminates at (r_2,z_2).

LOD $m\ d$

Define a line segment for line n by offsetting the entire line m a distance d. Positive d offsets the line segments to the left as one moves along line m in the direction that was originally defined. Negative d offsets the segment to the right.

LP $n\ r_1\ z_1\ ...\ r_n\ z_n$

The line definition consist of n points.

LPIL $l_1\ l_2$

Define point for line n at the intersection point of lines l_1 and l_2.

LPT $r_1\ z_1\ r_2\ z_2\ R$

Define a circular arc of radius R beginning at the last point defined and tangent to a line segment joining point (r_1,z_1) to point (r_2,z_2). This line segment will be extended or truncated to begin at the tangency point.

LPTA $r_c\ z_c\ R$

Define a line segment beginning at the last point defined and terminating at the tangency point on an arc of radius R centered at (r_c,z_c). The first tangency point encountered as the arc is generated by a counterclockwise rotation from the r-axis will become the end point. If R is given as a negative number, a clockwise rotation from the r-axis will determine the first tangency point.

LRL $n\ r_c\ z_c\ L\ t_1 \ldots t_n$

Define n lines consisting of radial lines of length L originating at point (r_c,z_c) and oriented at angles $t_1 \ldots t_n$. Positive angles are measured counterclockwise from r-axis.

LROT $l\ t$

Rotate line definition l about the origin t degrees.

LSCA $l\ s$

Scale line definition l by s.

LSCR $l\ s$

Scale r-coordinates of line definition l by s.

LSCZ $l\ s$

Scale z-coordinates of line definition l by s.

LSTL $m\ d_r\ d_z$

Define a line segment for line n by translating line m an increment (d_r,d_z).

LT $n\ d_r\ d_z$

Translate line n by the increment (d_r,d_z).

LTAS $r_1\ z_1$ rot $r_2\ z_2\ R$

Define a line segment tangent to a circular arc centered at point (r_1,z_1), beginning at the last point defined, and sweeping counterclockwise if rot=1 and clockwise if rot =-1. The line segment terminates at its tangency point on a second arc of radius R centered at (r_2,z_2). The first tangency point encountered as the second arc is generated by a counterclockwise rotation from the r-axis will become the end point. If R is given as a negative number; a clockwise rotation from the r-axis will determine the tangency point.

LTBC $m\ t\ dt\ s\ r_1\ r_2 \ldots r_m$

Define a line segment for line n with tab cell data. Tab cell data is often used in drafting programs, and consist of m radii each dt degrees apart starting at angle t. Each radius is scaled by s. Positive angles represent counterclockwise rotations.

LTBO $m_1\ d_1\ m_2\ d_2 \ldots m_m\ d_m$

Define a line segment for line n by offsetting last line defined with the "LTBC" or "LTBO" command. The radii of the first m_1 points are offset d_1, the next m_2 by d_2, and so on. Note that $m=m_1 + m_2 + \ldots m_m$ where m comes from the last "LTBC" command.

LTP $r\ z\ R$

Define a circular arc of radius R tangent to the last line segment and terminating at point (r,z). The last line segment will be automatically extended or truncated to the tangency point.

LVC $t\ l$

Define a line segment vector of length l,

	oriented
or	at t degrees (positive counterclockwise from the
LVC r_1 z_1 t l	*r*-axis). If this is the first command in
LVC r_2 z_2 t -l	a new line, the origin (*r*,*z*) must be given (second or third forms). A negative l indicates that the second point is defined, i.e., that the vector points towards the first point.
RLN	Read next line definition in operational input file and assign the next available line number.
RLNS	Read all line definitions in operational input file and assign the next available line numbers.

15. Three-Dimensional Line Definitions

Three-dimensional line definitions are lists of *x,y,z* points which form a piecewise linear curve. Each line definition has a name which is a character string with up to eight characters.

AVGN l_1 l_2...l_n ;	Average *n* line definitions.
BLEN s_1 s_2 s_3 s_4 s_5 p	Determine a line interpolated between surface definition s_1 and s_2 by a ratio p. s_3 and s_4 determine the end points of the line and the line lies on s_5. Convergence can be improved by using the following: PO for (s_1,s_3,s_5), P1 for (s_1,s_4,s_5), P2 for (s_2,s_4,s_5), P3 for (s_2,s_3,s_5). Refer to Figure 14-1.
COMP l_1 l_2...l_n ;	Form a single line definition by placing line definitions l_1 l_2 ... l_n end-to-end.
COPY *n* data	Move line definition *n* using the transformation defined by data. Input for data is described in the section on *Coordinate Transformations.*
L2D *n* data	Turn two-dimensional line definition *n* into a three dimensional line definition. The line definition is initially assumed to lie in the *x-y* plane and can be moved anywhere in space using data which is described in *Coordinate Transformations.*
L3D *n*	Begin line definition *n*.
LAD x_c y_c a	Form an arc about a *z*-vector located at (x_c,y_c) beginning at the last point defined and sweeping through a degrees.
LADD l_1 s_1 l_2 s_2	Make a linear combination: l_{new}=l_1*s_1+l_2*s_2.
LBCX *l r f*	Ball-correct line definition *l* with a cylinder parallel to the *x*-axis with radius *r*. If the correction is to the left then *f* is 'left', otherwise f is 'right'.
LBCY *l r f*	Ball-correct line definition *l* with a cylinder parallel to the *y*-axis with radius *r*. If the correction is to the left then *f* is 'left', otherwise f is 'right'.
LBCZ *l r f*	Ball-correct line definition *l* with a cylinder

parallel to the *z*-axis with radius *r*. If the correction is to the left then *f* is 'left', otherwise *f* is 'right'.

LBCV *l r* v_x v_y v_z — Ball-correct line definition *l* with a cylinder parallel to the vector, (v_x,v_y,v_z), with radius *r*. If the correction is to the left then *f* is 'left', otherwise *f* is 'right'.

LCUT *l* opt dist — Cut line definition *l* with a plane normal to axis opt (opt = *X*, *Y* or *Z*) at a distance of dist from the origin. The results are stored into calculator variables (l3cenx, l3ceny, l3cenz). They may then be accessed and used as necessary.

LEXP *x*(*s*) *y*(*s*) *z*(*s*) *n* — Define a line definition using expression *x*(*s*), *y*(*s*) and *z*(*s*) where $0 \le s \le 1$. The number of points to be used is *n*.

LFOR *l* opt vd sd dir — Form line definition *l*. If a point on *l* is inside (opt=IN) or outside (opt=OUT) of surface definition vd, then it is projected onto surface definition sd. The projection is constrained by dir:

- dir=0: project to nearest point.
- dir=1: project along *X*-axis
- dir=2: project along *Y*-axis
- dir=3: project along *Z*-axis

LINT l_1 l_2 *s* — Form a line by linear interpolation between l_1 and l_2 with parameter *s*.

LLCM l_1 s_1 ... l_n s_n ; — Form a linear combination of *n* lines where lnew(*r*)=$l_1(r)*s_1$+ ... +$l_n(r)*s_n$.

LP *n* x_1 y_1 z_1 ... x_n y_n z_n — The line definition consists of *n* points.

LPN *n* p_x p_y p_z — The next point on the line definition is at (p_x,p_y,p_z) but *n* equal spaced points in a straight line are added to make up this segment.

LPRJ *l s* — Project line definition *l* onto surface definition *s*.

LREV — Reverse the direction of the line. Additions to the line definition will occur at the beginning of the line rather than the end of the line.

LRNX *l r* — Sharp corners on line definitions *l* are rounded by a cylinder parallel to the *x*-axis with radius *r*. The coordinates of the center of rotation of the last round are returned to calculator variables (l3cenx,l3ceny,l3cenz) and the last angle of sweep

	is returned to l3angle.
LRNY *l r*	Sharp corners on line definitions *l* are rounded by a cylinder parallel to the *y*-axis with radius *r*. The coordinates of the center of rotation of the last round are returned to calculator variables (l3cenx,l3ceny,l3cenz) and the last angle of sweep is returned to l3angle.
LRNZ *l r*	Sharp corners on line definitions *l* are rounded by a cylinder parallel to the z-axis with radius *r*. The coordinates of the center of rotation of the last round are returned to calculator variables (l3cenx,l3ceny,l3cenz) and the last angle of sweep is returned to l3angle.
LRNV *l r* v_x v_y v_z	Sharp corners on line definition l are rounded by a cylinder parallel to the vector, (v_x ,v_y,v_z), with radius *r*. The coordinates of the center of rotation of the last round are returned to calculator variables (l3cenx,l3ceny,l3cenz) and the last angle of sweep is returned to l3angle.
LROT p_x p_y p_z v_x v_y v_z w	Form an arc by taking the last point and rotating it an angle w (in degrees) about the axis defined by point (p_x,p_y,p_z) and orientation vector (v_x,v_y,v_z).
LVT *d*	Add a vector tangent to the last line segment with length *d*.
LVTB *d*	Add a vector tangent to the first line segment with length *d*.
P0 x_0 y_0 z_0	Set point P0 for intersection.
P1 x_1 y_1 z_1	Set point P1 for intersection determination on the next command.
P2 x_2 y_2 z_2	Set point P2 for intersection determination on the next command.
P3 x_3 y_3 z_3	Set point P3 for intersection determination on the next command.
PINT s_1 s_2 s_3	The next point on the line is at the intersection of s_1, s_2 and s_3. P0 can be used to improve convergence. The results of the projection are returned to the calculator variables (l3cenx,l3ceny,l3cenz).
PPRJ p_x p_y p_z surf	The next point on the line definition is formed

by projecting (p_x,p_y,p_z) to the nearest point on surface surf.

SINT s_1 s_2 s_3 s_4

Determine the curve formed by the intersection of s_1 and s_2 beginning at s_3 and terminating at s_4. If this is not the first point on the line, then s_3 is not input and LS-INGRID assumes that the last point defined lies on the intersection of s_1 and s_2. The convergence can be improved by using P0 for (s_1,s_2,s_3) and P1 for (s_2,s_3,s_4).

16. Surface Definitions

This section describes options for defining three-dimensional curved surfaces. Analytical representations of the surfaces are stored, if possible, so that exact projections can be made.

BLND d_1 l_1 d_2 l_2

The surface is blended between line definition l_1 and line definition l_2. Initially the line definitions are in the *x*-*z* plane at *y*=0. Line definitions are moved by d_1 and d_2 which are described in *Coordinate Transformations*.

CN2P p_x p_y p_z v_x v_y v_z r_1 z_1 r_2 z_2

Define a conical surface by specifying the axis and two points. (p_x,p_y,p_z) is a point on the axis and (v_x,v_y,v_z) is a vector which orients the axis. radial and axial positions relative to the center point are (r_1,z_1) and (r_2,z_2).

CONE p_x p_y p_z v_x v_y v_z r θ

Define a conical surface by specifying an axis, a radius and an angle. (p_x,p_y,p_z) is a point on the axis where the cone has radius r and (v_x,v_y,v_z) is a vector along the axis. The angle of the cone relative to the positive axis is θ.

CP data l

Form an infinite surface from line definition *l*. Initially the *r*-coordinate of the line definition is the *x*-coordinate of the part and the *z*-coordinate of the line definition is the y-coordinate of the part. The surface is the same curve in any *x*-*y* plane along the part's *z*-axis. *Coordinate Transformations* describes data which can be used to move the surface anywhere in space.

CR p_x p_y p_z v_x v_y v_z l

Spin two-dimensional line definition, *l*, about an axis to form a line. (p_x,p_y,p_z) is the center point on the axis and (v_x,v_y,v_z) is a vector which orients the axis.

CRX, **CRY**, or **CRZ** l

Spin two-dimensional line definition, *l*, about the *X*, *Y*, or *Z* axes, respectively.

CYLI p_x p_y p_z v_x v_y v_z r

Define a cylindrical surface with a point on the axis at (p_x,p_y,p_z), an orientation vector (v_x,v_y,v_z), and radius *r*.

ER p_x p_y p_z v_x v_y v_z r_1 r_2

Define an elliptical surface revolved about an

axis. (p_x,p_y,p_z) is the center point, (v_x,v_y,v_z) is a vector which orients the ellipse. The radius in the plane normal to the axis of rotation is r_1 and the intercept along the axis of rotation is at +/- r_2.

GELN *a b c n* data

Define a general ellipsoid with the equation:
$(x/a)^n+(y/b)^n+(z/c)^n=1$

The ellipsoid can be positioned anywhere in space with data which is described in the section on *Coordinate Transformations.*

GELS *a b c* data

Define an ellipsoid with the equation:
$(x/a)^2+(y/b)^2+(z/c)^2=1$

The ellipsoid can be positioned anywhere in space with data which is described in the section on *Coordinate Transformations.*

GS *n*

Use general 3-D digitized surface number *n*.

GS1 *n*

Use lower side of general 3-D surface.

GS2 *n*

Use upper side of general 3-D surface.

GSM data *n*

Digitized surface *n* is moved by data which is described in *Coordinate Transformations.*

GSN offset *n*

Use digitized surface *n* but offset the surface by offset in the normal direction.

L3 *l*

The surface is actually just three-dimensional line definition *l*.

L3P *l* v_x v_y v_z

Three-dimensional line definition *l* is projected along (v_x,v_y,v_z) to form a surface.

L3R *l r*

The surface is a circular tube of radius *r* about three-dimensional line definition *l*.

L3S p_x p_y p_z v_x v_y v_z *l*

Spin three-dimensional line definition *l* about the axis defined by point (p_x,p_y,p_z) and orientation vector (v_x,v_y,v_z).

NCV data *n*

NURB curve defined by entity number *n* in the NURB geometry database is used. This curve is moved by data which is described in *Coordinate Transformations.*

NSF data *n*

NURB surface defined by entity number *n* in the NURB geometry database is used. This surface

	is moved by data which is described in *Coordinate Transformations.*
NSFN data *n*	NURB surface defined by entity name *n* in the SC03 geometry database is used. This surface is moved by data which is described in *Coordinate Transformations.*
PL3 f1 p1x p1y p1z f2 p2x p2y p2z f3 p3x p3y p3z offset	Plane Defined by three points. f1, f2 and f3 specify the coordinate system which is RT for rectangular, CY for cylindrical, or SP for spherical. P1, P2, and P3 must be three non-collinear points in the plane. f2 and f3 can also be V to indicate that the input points are vectors relative to P1. The surface definition is offset from the three points by the distance offset.
PLAN $p_x\ p_y\ p_z\ v_x\ v_y\ v_z$	Define a plane. (p_x,p_y,p_z) is any point on the plane and (v_x,v_y,v_z) is a normal vector.
POLY $p_x\ p_y\ p_z\ v_x\ v_y\ v_z$ $n\ a_0\ a_1\ \ldots\ a_n$	Define a surface as a planar polynomial which is then spun about an axis. (p_x,p_y,p_z) is a point on the axis of rotation and (v_x,v_y,v_z) is a vector which orients the axis of rotation. The polynomial is of degree n with coefficients a_0, a_1, ... a_n.
PR $p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ r_1\ t_1\ r_2\ t_2\ r_3\ t_3$	Parabolic surface of revolution. (p_x,p_y,p_z) is a point on the axis of revolution and (v_x,v_y,v_z) is a vector orienting the axis of revolution. (r_1,t_1), (r_2,t_2) and (r_3,t_3) are radial and axial positions of three points which are fit with a parabola.
SP $p_x\ p_y\ p_z\ r$	Define a sphere with center (p_x,p_y,p_z) and radius r.
T1 $l_1\ l_2\ l_3\ l_4\ l_5$	This is a special purpose surface. l_1, l_2, l_3, l_4 and l_5 refer to two-dimensional line definitions. The surface is axisymmetric about the *Z*-axis and performs *Z*-projections only. The equation for the surface is as follows: $z = l_3(r)+(l_2(r)-l_1(r))*(1+l_4(q)*l_5(r))+l_1(r)$
T2 *n* opt	Project along an axis onto digitized surface definition *n*. Values for opt are as follows: =MINX: project along the *X*-axis to the minimum *X*- intercept. =MAXX: project along the *X*-axis to the maximum *X*- intercept. =MINY: project along the *Y*-axis to the minimum *Y*- intercept.

=MAXY: project along the *Y*-axis to the maximum *Y*- intercept.
=MINZ: project along the *Z*-axis to the minimum *Z*- intercept.
=MAXZ: project along the *Z*-axis to the maximum *Z*- intercept.

TS $p_x\ p_y\ p_z\ q_x\ q_y\ q_z\ r_1\ t_1\ r_2$

Define a torus. (p_x,p_y,p_z) is a point on the primary axis of rotation and (v_x,v_y,v_z) is a vector which orients this axis. r_1 is the radius to the secondary axis, t_1 is an axial offset relative to (p_x,p_y,p_z) and r_2 is the radius from the secondary axis to the torus surface.

TS2P $p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ r_1\ z_1\ r_2\ z_2\ r_3$

Define a torus with two points on the surface. (p_x,p_y,p_z) is a point on the primary axis of rotation and (v_x,v_y,v_z) is a vector which orients this axis. r_3 is the radius of the surface from the secondary axis. If $r_3 > 0$ then the secondary axis lies to the left as one moves from (r_1, z_1) to (r_2, z_2). Otherwise, the axis is to the right.

17. Volume Definitions

This section documents the available solid geometric objects. Solid objects are used by the "VD" command in the control section. The following types are available.

CR $p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ l$	Form a solid by spinning two-dimensional line definition l about the axis defined by point (p_x,p_y,p_z) and orientation vector (v_x,v_y,v_z).
CYF $p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ r\ t_{\min}\ t_{\max}$	Define a cylinder of radius r and axis defined by point (p_x,p_y,p_z) and orientation vector (v_x,v_y,v_z). The cylinder extends along the axis from $t_{\min}$ to $t_{\max}$.
CYLI $p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ r$	Define a cylinder of radius r and axis defined by point (p_x,p_y,p_z) and orientation vector (v_x,v_y,v_z). The cylinder has infinite length.
DS n	Digitized surface n is a closed surface which defines a volume.
RECT $n\ x_{\min}\ x_{\max}\ y_{\min}\ y_{\max}\ z_{\min}\ z_{\max}$	Define a rectangular solid with $x_{\min}\leq x\leq x_{\max}$, $y_{\min}\leq y\leq y_{\max}$ and $z_{\min}\leq z\leq z_{\max}$. This can be positioned anywhere in space using global coordinate transformation number n.
SD $n\ t$	The surface is defined by surface definition n and thickness t.
SPHE $p_x\ p_y\ p_z\ r$	Define a sphere of radius r and centered at (p_x,p_y,p_z).
TRIA $n\ x_1\ y_1\ x_2\ y_2\ x_3\ y_3\ z_{\min}\ z_{\max}$	The solid is a triangular section in the X-Y plane which runs from $z_{\min}$ to $z_{\max}$ in the Z-direction. (x_1,y_1), (x_2,y_2), and (x_3,y_3) are the three corner points. This can be moved anywhere in space using global coordinate transformation n.

18. Coordinate Transformations

Option 1:

For Option 1, three nodal points must be input. Figure 18-1 shows the orientation of the local coordinate system defined by the three points.

f_1	Flag describing coordinate type for point 1 ="*RT*": rectangular coordinates ="*CY*": cylindrical coordinates (R, θ, Z). ="*SP*": spherical coordinates (R, θ, ψ)
$p_{1x}\ p_{1y}\ p_{1z}$	Coordinates for point 1.
f_2	Flag describing coordinate type for point 2. ="*RT*": rectangular coordinates ="*CY*": cylindrical coordinates (R, θ, Z). ="*SP*": spherical coordinates (R, θ, ψ) ="*V*": point 2 is offset from point 1 by the vector.
$p_{2x}\ p_{2y}\ p_{2z}$	Coordinates or vector for point 2
f_3	Flag describing coordinate type for point 3. ="*RT*": rectangular coordinates ="*CY*": cylindrical coordinates (R, θ, Z). ="*SP*": spherical coordinates (R, θ, ψ) ="*V*": point 3 is offset from point 1 by the vector.
$p_{3x}\ p_{3y}\ p_{3z}$	Coordinates or vector for point 3.

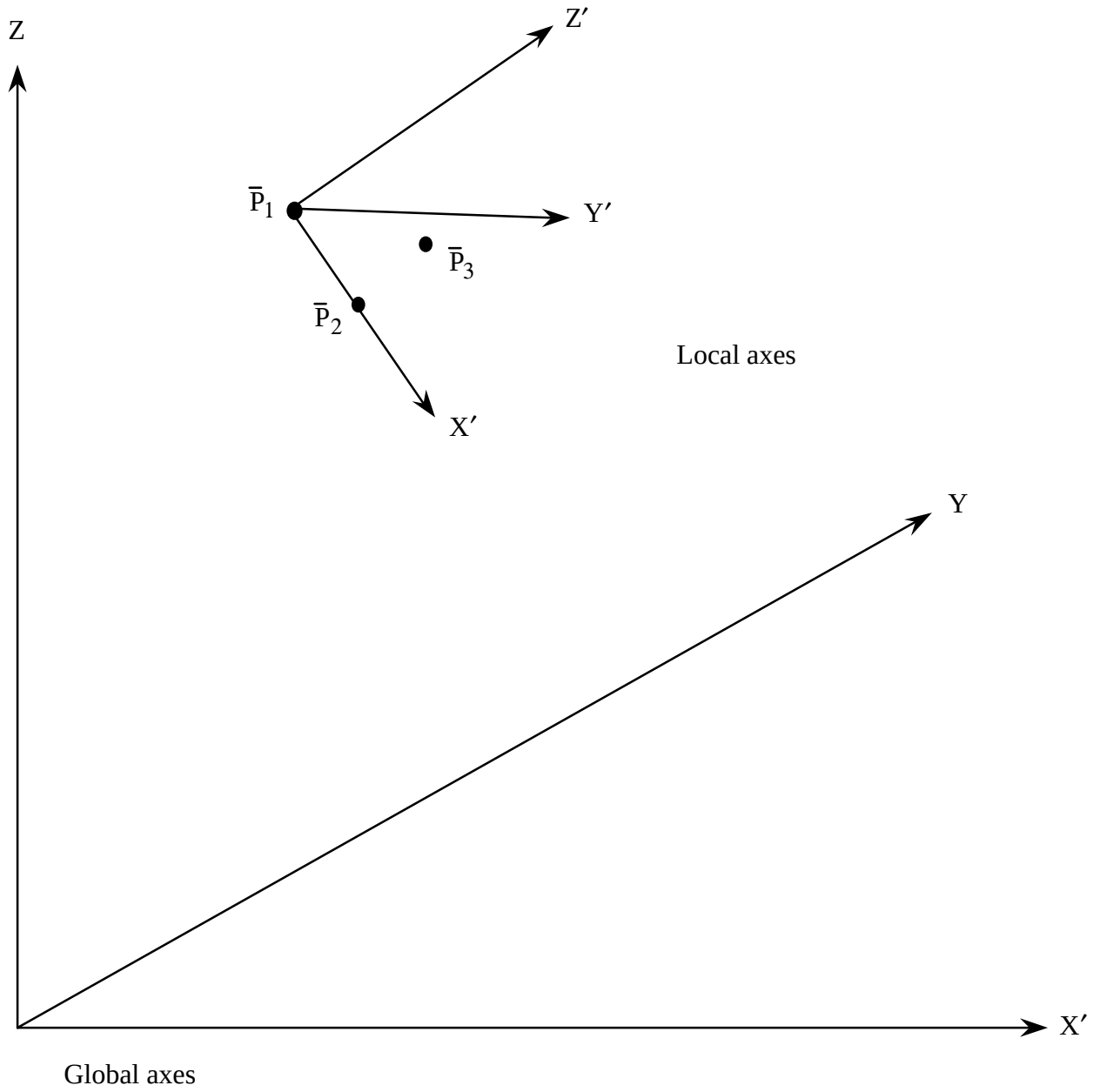


Figure 18-1. Coordinate Transformations.

Option 2:

Option 2 allows the following commands in any order.

CSCA *s*	Scale coordinates by *s*.
D1	Save the current offset position and perform rotations relative to this point.
D2	Restore the offset position.
L	Copy the previous transformation and begin defining the next system.
MATRIX a_{11} a_{21} a_{31} a_{12} a_{22} a_{23} a_{31} a_{32} a_{33}	Set the transformation to the input 3«3 matrix.
MX Δx	Move Δx in the *x*-direction.
MY Δy	Move Δy in the *y*-direction.
MZ Δz	Move Δz in the *z*-direction.
REPE *n*	Repeat the current transformation n times.
RX θ	Rotate θ degrees about the *X*-axis.
RY θ	Rotate θ degrees about the *Y*-axis.
RZ θ	Rotate θ degrees about the *Z*-axis.
RXY	Reflect about the *XY* plane.
RYZ	Reflect about the *YZ* plane.
RZX	Reflect about the *ZX* plane.
SAVE *n*	The sequence of coordinate transformations is generated starting from coordinate system *n* when using the "REPE" command.
SCALE *s*	Scale coordinates by *s*.
V Δx Δy Δz	Move Δx, Δy, and Δz.
XSCA *s*	Scale *X*-coordinates.
YSCA *s*	Scale *Y*-coordinates.
ZSCA *s*	Scale *Z*-coordinates.
;	Terminate Option 2.

19. LS-DYNA2D Commands and Materials

Analysis options are code dependent. They can be set either in the control section of the LS-INGRID input file or in the graphics phase. These commands become active when LS-DYNA2D output is selected with the DN2D command (see *Control Commands*).

BRODE — Define Brode function parameters.

Options:

YLD	yld	Yield (Ktons)
HEIGHT	*h*	Height of burst
XBO	*x*	Coordinates of Brode origin (space, time) in LS-INGRID units.
YBO	*y*	
ZBO	*z*	
TBO	*t*	
CL	*cl*	Conversion factor - ft. to DYNA length units (default = meters).
CT	*ct*	Conversion factor - ms to DYNA time units (default - seconds).
CP	*cp*	Conversion factor - psi to DYNA pressure units (default = Pascals).
;		Terminate Brode function input.

Note: If "RANG" "COEF", and "GFUN" are specified, a "modified" Brode function will be used in DYNA; otherwise, straight Brode is used.

RANG	$r_1 \ldots r_5$	Range values for Brode function.
COEF	$c_1 \ldots c_8$	Coefficient values for Brode function.
GFUN	$g_1 \ldots g_7$	GFUNC values for Brode Function.

The Brode function is applied to pressure surfaces with load curve number -1.

DBQT *i* — Change default bulk viscosity type from 1 to *i*:
=1: standard LS-DYNA2D
=2: Richards-Wilkins

DHGQ Q_h — Change default hourglass viscosity from .10 to Q_h.

DHQT *i* — Change default hourglass viscosity type from 1 to *i*:

=1: standard LS-DYNA2D
=2: rotational
=3: Flanagan-Belytschko viscous form.
=4: Hancock

DQL Q_l — Change default linear bulk viscosity from .06 to Q_l.

DQQ Q_q — Change default quadratic bulk viscosity from 1.5 to Q_q.

GEOM — Select geometry type.
=AXIS: axisymmetric (default)
=PLAN: plane strain

GRAV $g_x\ g_y\ g_z$ — Gravity acceleration vector.

ITSS t_o — Initial time step size. This is optional input for LS-DYNA3D. If t_o is zero, LS-DYNA3D picks the initial time step size.

PLTI D*t* — Node and element data dump interval for TAURUS post-processing.

PRTI D*t* — Node and element data dump interval for high speed printer.

RDMT *m* — Delete material *m*. This applies to the restart number selected by the RNUM command.

RDSI *s* — Delete sliding interface *s*. This applies to the restart number selected by the RNUM command.

REST name — Set the family name for restart input file generation to name.

REZO $t_1\ t_2$ D*t* — Periodic rezones begin at time t_1 and end at time t_2. Rezones are performed after every time interval of D*t*.

RHVC *h* — The default hourglass viscosity for restart is set to *h*. This applies to the restart number selected by the RNUM command.

RLBV *l* — The default linear bulk viscosity for restart is set to *l*. This applies to the restart number selected by the RNUM command.

RNUM *n* — Restart commands apply to restart number *n*.

RPLT *t* — The plot interval for restart is set to *t*. This applies to the restart number selected by the RNUM command.

RPRT *t* The print interval for restart is set to *t*. This applies to the restart number selected by the RNUM command.

RQBV *q* The default quadratic bulk viscosity for restart is set to *q*. This applies to the restart number selected by the RNUM command.

RTERM *t* The termination time for this restart is *t*. This applies to the restart number selected by the RNUM command.

RTSF *s* The time step scale factor for restart is set to *s*. This applies to the restart number selected by the RNUM command.

SBRF *n* Number of time steps between restart dumps is *n*.

SFSI *s* Sliding interface scale factor (default = 1.0).

TEO *i* Thermal effects option

=0: no thermal effects.

=N: nodal temperatures are defined in input and are scaled according to a time function. N is the load curve number.

=-1: each time step a new temperature state is read from a disk file. The time word at the beginning of each temperature state is ignored.

=-2: each time step a temperature state is interpolated from the temperature state in a disk file. Therefore the time words at the beginning of each temperature state are used.

=-3: the disk file containing temperatures has only one state. The initial state is assumed to be zero.

TERM *t* Terminate dynamic time integration at time *t*.

TINV *n* Number of time steps between dumps of reaction history blocks.

TSSF *s* Scale factor on time step size.

19.1 LS-DYNA2D MATERIAL INPUT

LS-DYNA2D material input is possible after the DN2D command is input (see *Control Commands*). The form of this input is: MAT *n* TYPE *m* {options specific to material type *m*} {general material options} ENDMAT. *n* is a material name which is assigned an input number. Therefore, the materials should be defined in order before any additional use of materials is made.

19.2 GENERAL MATERIAL OPTIONS

BQL Q_l	Change linear bulk viscosity for .06 to Q_l.
BQQ Q_q	Change quadratic bulk viscosity from 1.5 to Q_q.
BQT *i*	Change bulk viscosity type from 1 to *i*: =1: standard LS-DYNA2D =2: Richards-Wilkins
EOS eost	Begin defining equation-of-state type eost for the current material definition. Each equation-of-state is terminated by the ENDEOS command.
HEAD	Replace default heading (typed on the next line).
HGQ Q_h	Change value of hourglass viscosity from .10 to Q_h.
HGQT	Change value of hourglass viscosity type from 1 to *i*: =1: standard LS-DYNA2D =2: rotational =3: Flanagan-Belytschko viscous form =4: Hancock
IMGL	Initialize material for gravity loads.
MAT *m*	Begin material definition *m*. Each material definition is terminated by the ENDMAT command.
RO *m*	Density (required - no default).
TYPE *n*	The current material is of type *n*.

Material Type 1 (Elastic)

Default heading: Material Type #1 (Elastic)

Input any two of the following.

BULK *K*	Bulk modulus
E *E*	Young's modulus
G *G*	Shear modulus
PR ν	Poisson's ratio

Material Type 2 (Orthotropic Elastic)

EA E_a	See constitutive matrix below
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA $\upsilon_{\chi a}$	
PRCB υ_{cb}	
GAB G_{ab}	
AOPT aopt	Material axes option (Figure 19-1). =0.0: locally orthotropic with materials axes by j value specified on each element card and element nodes n_1 and n_2. (see Figure 19-1). =1.0: locally orthotropic with materials axes by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by j_G.
RP r_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
PSIG j_G	Define for AOPT = 2.

The material law that relates stresses to strains is defined as:

$$\underset{\sim}{C} = \underset{\sim}{T}^T \underset{\sim}{C}_L \underset{\sim}{T},$$

Where $\underset{\sim}{T}$ is a transformation matrix and $\underset{\sim}{C}_L$ is the constitutive matrix defined in terms of the material constants of the orthogonal material axes, *a*, *b*, and *c*. The inverse of $\underset{\sim}{C}_L$ is defined as

$$\underset{\sim L}{C^{-1}} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ba}}{E_b} & -\frac{\nu_{ca}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ab}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{cb}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ac}}{E_a} & -\frac{\nu_{bc}}{E_b} & \frac{1}{E_c} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{bc}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{ca}} \end{bmatrix}$$

Note that $\frac{\nu_{ab}}{E_a} = \frac{\nu_{ba}}{E_b}, \frac{\nu_{ca}}{E_c} = \frac{\nu_{ac}}{E_a}, \frac{\nu_{cb}}{E_c} = \frac{\nu_{bc}}{E_b}$.

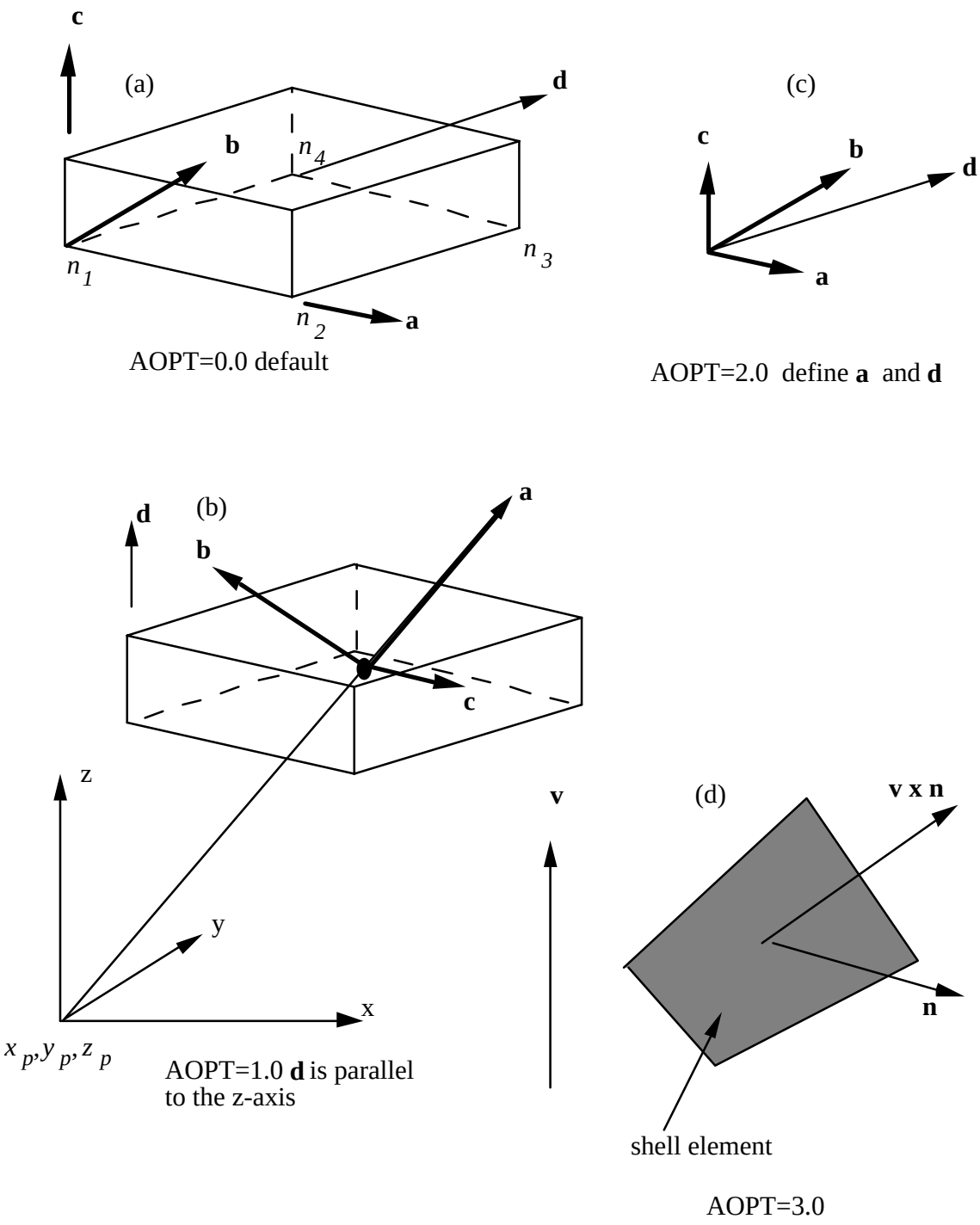


Figure 19-1. Options for determining principal materials axes: (a) AOPT = 0.0, (b) AOPT = 1.0, and (c) AOPT = 2.0.

Material Type 3 (Kinematic/Isotropic Elastic/Plastic)

Default heading: Material Type #3 (Elastic-Plastic)

Input any two of the following:

BULK *K*	Bulk modulus
E *E*	Young's modulus
G *G*	Shear modulus
PR ν	Poisson's ratio

Additional *Options*:

SIGY σ_y	Yield stress
ETAN E_t	Hardening modulus
BETA β'	Hardening parameter, $0 \leq \beta' \leq 1$

Isotropic, kinematic, or a combination of isotropic and kinematic hardening may be specified by varying β' between 0 and 1. For β' equal to 0 and 1, respectively kinematic and isotropic hardening are obtained as shown in Figure 19-2. Effective stress is defined in terms of the deviatoric stress tensor, s_{ij} as:

$$\bar{\sigma} = \left(\frac{3}{2} S_{ij} S_{ij} \right)^{1/2}$$

where,

$$S_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij}$$

and effective plastic strain by:

$$\bar{\varepsilon}^{\,p} = \int_0^t d\bar{\varepsilon}^{\,p}$$

where t denotes time and

$$d\bar{\varepsilon}^{\,p} = \left(\frac{2}{3} d\varepsilon_{ij}^{\,p} d\varepsilon_{ij}^{\,p} \right)^{1/2}$$

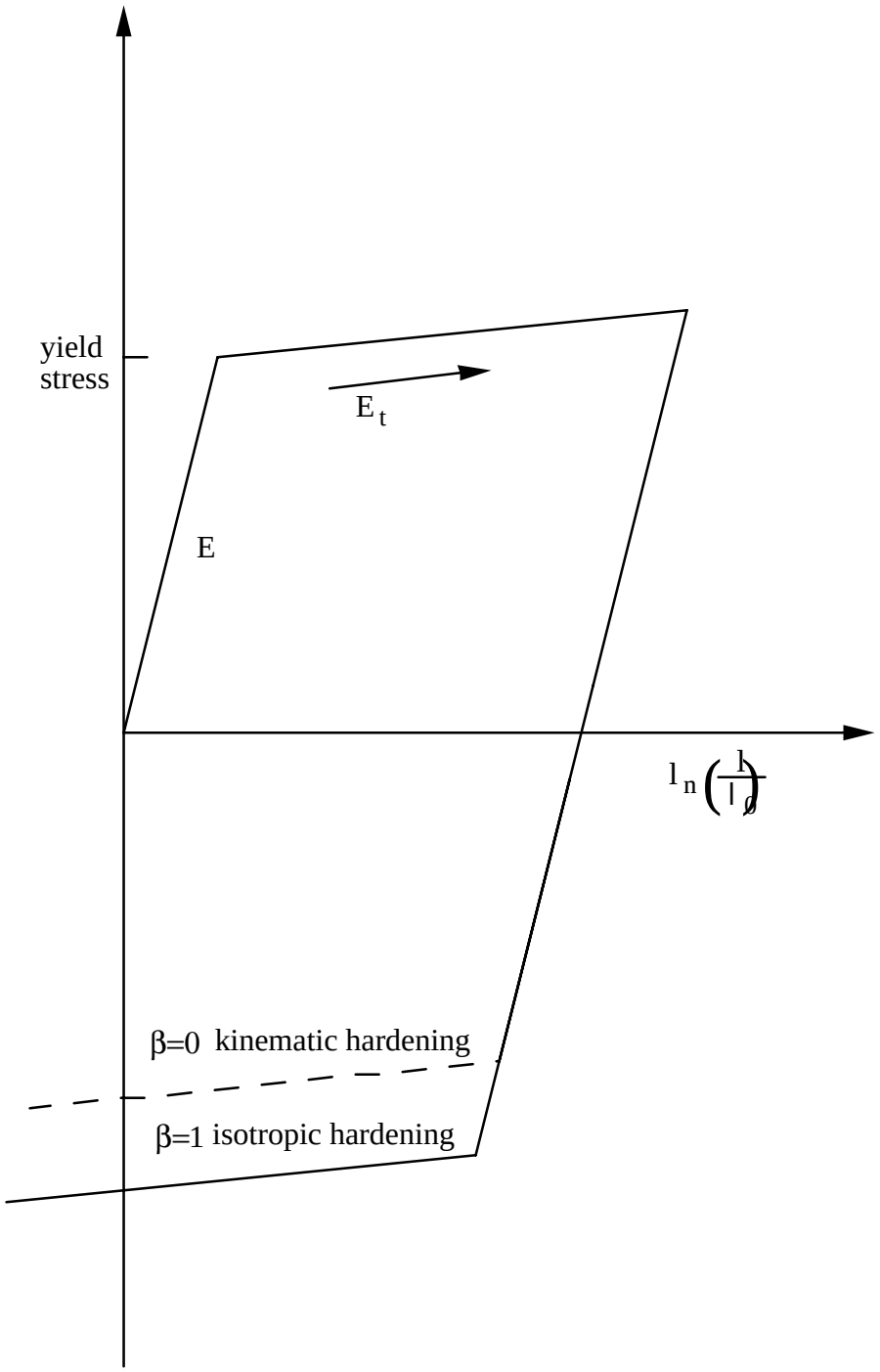


Figure 19-2. Elastic-plastic behavior with isotropic and kinematic hardening where l_0 and l are undeformed and deformed length of uniaxial tension specimen.

Material Type 4 (Thermo-Elastic-Plastic)

Default heading: Material Type 4 (Thermo-Elastic-Plastic)

NPTS *n*	Number of temperature values for which material constants are defined.
TEMP T_1 T_2...T_n	Temperatures
E E_1 E_2...E_n	Young's moduli
PR υ_1 υ_2...υ_n	Poisson's ratios
ALPHA α_1 α_2...α_n	Coefficients of thermal expansion.
SIGY σ_{y1} σ_{y2}...σ_{yn}	Yield stresses
ETAN E_{t1} E_{t2}...E_{tn}	Tangent moduli

Material Type 5 (Soil and Crushable Foam)

Default heading: Material Type 5 (Soil and Crushable Foam)

Input any two of the following:

BULK *K*	Bulk modulus
E *E*	Young's modulus
G *G*	Shear modulus
PR ν	Poisson's ratio

Additional *Options:*

AO a_0	Yield function constant.
A1 a_1	Yield function constant.
A2 a_2	Yield function constant.
PC P_c	Pressure cutoff for tensile fracture.
NPTS *n*	Number of points in volumetric strain versus pressure curve (*n*£10).
VS ε_{v1} ε_{v2} ... ε_{vn}	Volumetric strain values.
P p_1 p_2...p_n	Pressures corresponding to volumetric strain values.

The deviatoric yield function, ϕ, is described in terms of the second invariant J_2.

$$J_2 = \frac{1}{2} s_{ij} \; s_{ij}$$

Pressure, p, and constants a_0, a_1, and a_2 as:

$$\phi = J_2 - \left[a_0 + a_1 p + a_2 p^2\right].$$

On the yield surface, $J_2=1/3(\sigma_y)^{1/2}$, where σ_y is the yield stress, i.e.,

$$\sigma_y = \left[3\left(a_0 + a_1 p + a_2 p^2\right)\right]^{1/2}$$

For elastic-perfectly plastic behavior $a_1=a_2=0$, and $(3a_0)^{1/2}$ defines the yield strength. The volumetric strain is given by the natural logarithm of the relative volume V. If the pressure drops below the cutoff value, PC, then it is reset to that value.

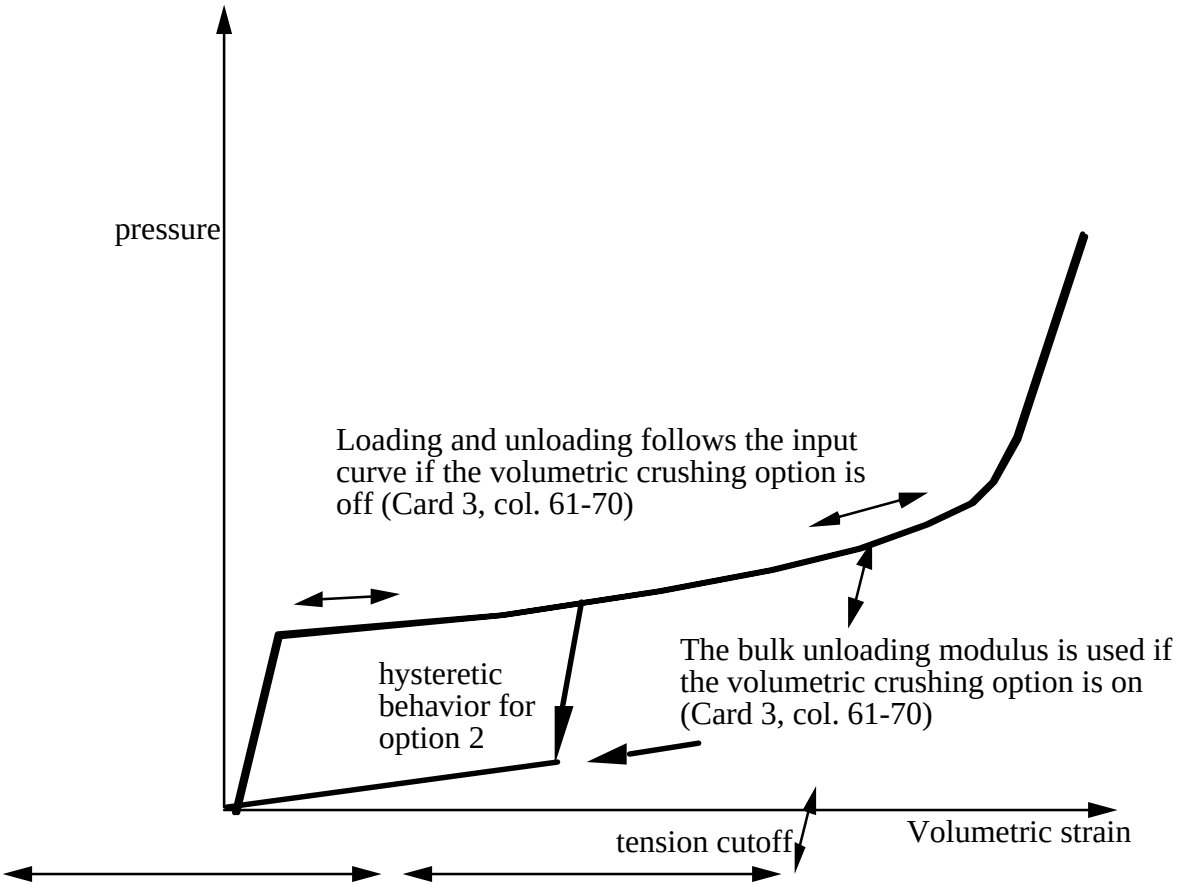


Figure 19-3. Volumetric strain versus pressure curve for soil and crushable foam model.

Material Type 6 (Viscoelastic)

G G_0	Short term shear modulus.
GI G_∞	Long term shear modulus.
K K	Bulk modulus.
BETA β	Decay constant.

The shear relaxation behavior is described by:

$$G(t) = G_\infty + (G_0 - G_\infty)\, e^{-\beta t}$$

A Jaumann rate formulation is used:

$$\overset{\nabla}{\sigma}'_{ij} = 2 \int_0^t G(t-\tau)\, D'_{ij}(\tau)\, dt$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{\sigma}_{ij}$, and the strain rate D_{IJ}.

Material Type 7 (Blatz-Ko Rubber)

Default heading: Material Type #7 (Rubber)

G m	Shear modulus

The second Piola-Kirchhoff stress is computed as

$$S_{ij} = \mu \left(\frac{1}{V} C_{ij} - V^{-1/1-2\upsilon} \delta_{ij} \right)$$

where V is the relative volume, C_{ij} is the right Cauchy-Green strain tensor, and ν is the Poisson's ratio which is set to .463 internally. This stress measure is transformed to the Cauchy stress, σ_{ij}, according to the relationship

$$\sigma_{ij} = V^{-1} F_{ik} F_{jl} S_{lk}$$

where F_{ij} is the deformation gradient tensor.

Material Type 8 (High Explosive Burn)

Default heading: Material Type #8 (High Explosive Burn)

D *D*	Detonation velocity.
PCJ P_{CJ}	Chapman-Jouget pressure.

This material model requires an equation-of-state.

Material Type 9 (Null Material)

Default heading: Material Type #9 (Null Material)

PC pc	Pressure cutoff.
MU μ	Viscosity.

The null material must be used with an equation-of-state. Pressure cutoff is negative in tension. A viscous stress of the form

$$\sigma_{ij} = \mu \dot{\varepsilon}_{ij}$$

is computed for nonzero μ where $\dot{\varepsilon}_{ij}$ is the deviatoric strain rate.

Material Type 10 (Isotropic-Elastic-Plastic-Hydrodynamic)

Default heading: Material Type #10 (Isotropic-Elastic-Plastic-Hydrodynamic)

G *G*	Shear modulus
SIGY σ_y	Yield strength
EH E_h	Plastic hardening modulus
PC p_c or -s_f	Pressure cutoff = 0: cutoff of -´ is assumed
A1 a_1	Yield function constant
A2 a_2	Yield function constant
NPTS *n*	Number of points in yield stress-effective plastic strain curve or yield stress-pressure curve.
ES σ_{y1} σ_{y2} ... σ_{yn}	Yield stress
EPS ε_{p1} ε_{p2} ... ε_{pn}	Effective plastic strain
P p_1 p_2...p_n	Pressure
FS *s*	Failure strain

If the yield stress-plastic strain curve is not defined and if $a_1 = a_2 = 0$, the bilinear stress-strain curve shown in Figure 19-2 is obtained with b =1. The yield strength is calculated as

$$\sigma_y = \sigma_0 + E_h \bar{\varepsilon}^p$$

The quantity E_h is the plastic hardening modulus defined in terms of Young's modulus, E, and the tangent modulus, E_t , as follows

$$E_h = \frac{E_t \, E}{E - E_t}$$

If Cards 5-8 are used, a curve like that shown in Figure 3.4 may be defined. Effective stress is defined in terms of the deviatoric stress tensor, s_{ij}, as:

$$\bar{\sigma} = \left(\frac{3}{2} s_{ij} \, s_{ij} \right)^{1/2} \tag{1}$$

and effective plastic strain by:

$$\bar{\varepsilon}^p = \int_0^t \left(\frac{2}{3} D_{ij}^p D_{ij}^p \right)^{1/2} dt, \tag{2}$$

where t denotes time and D_{ij}^p is the plastic component of the rate of deformation tensor. In this case the plastic hardening modulus on Card 3 is ignored and the yield stress is given as

$$\sigma_y = f\left(\bar{\varepsilon}^p\right),$$

where the value for $f\left(\bar{\varepsilon}^p\right)$ is found by interpolation from the data curve.

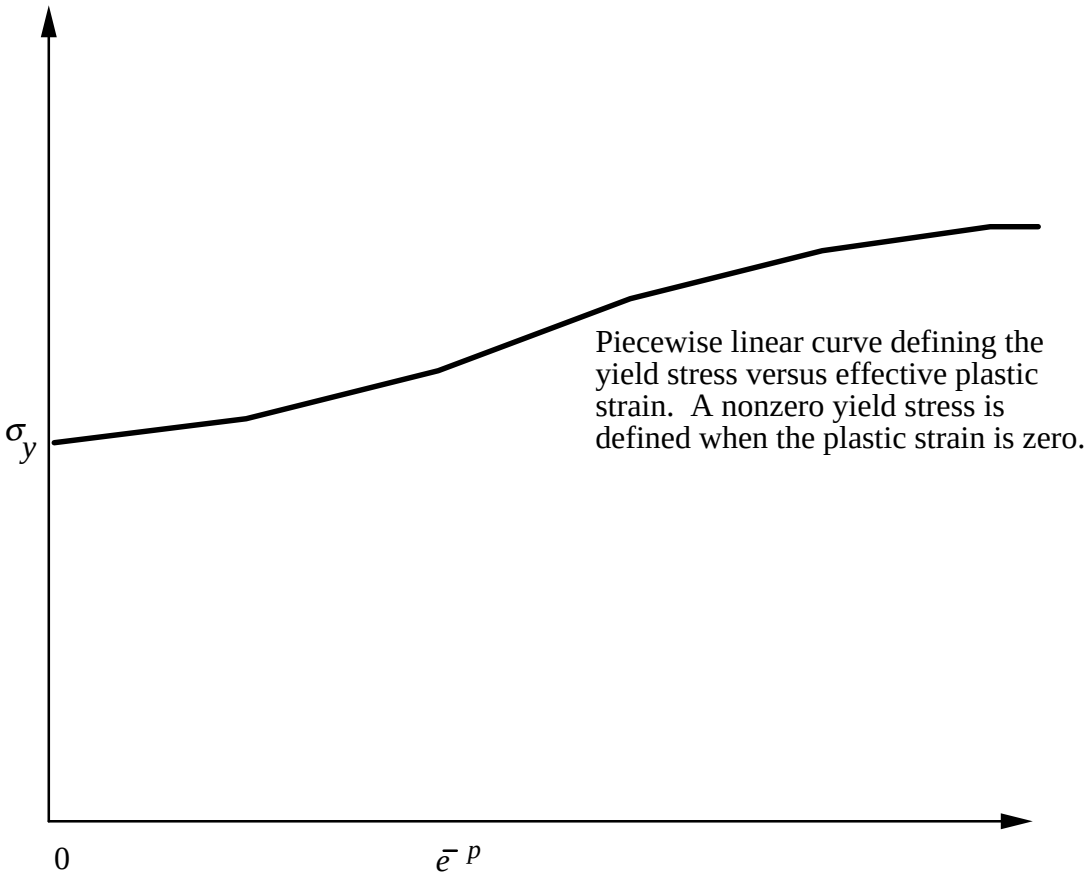


Figure 19-4. Effective stress versus effective plastic strain curve.

Material Type 11 (Temperature Dependent Elastic-Plastic Hydrodynamic)

Default heading: Material Type #11 (Temperature Dependent Elastic-Plastic Hydrodynamic)

G G_0	Shear modulus.
SIGO σ_o	See equations below.
BETA β	
N n	
GAMA g_i	
SIGM σ_m	
B b	
BP b'	
H h	
F f	
A A	
TO T_{mo}	
GAMO γ_0	
SA a	

PC pmin or -s_f
ECO EC_0 Cold compression energy coefficients (optional)
EC1 EC_1
EC2 EC_2
EC3 EC_3
EC4 EC_4
EC5 EC_5
EC6 EC_6
EC7 EC_7
EC8 EC_8
EC9 EC_9

If cold compression energy coefficients are not input, then LS-DYNA2D will calculate them based on the equation-of-state.

SPALL type Spall type
=0: default set to "2.0"
=1: p 3 $p_{\min}$
=2: if $\sigma_{max} \geq \sigma_f$ element spalls and tension $p < 0$, is never allowed; σ_{max} = maximum principal stress.
=3: if $p < p_{\min}$ element spalls and tension $p<0$, is never allowed.
=4: failure strain

Users who have an interest in this mode are encouraged to study the paper by Steinberg and Guinan [9] which provides the theoretical basis. Another useful reference is the KOVEC user's manual [10].

In terms of the foregoing input parameters, we define the shear modulus, G, before the material melts as:

$$G = G_0\left[1 + b p V^{1/3} - h\left(\frac{E_i - E_c}{3R'} - 300\right)\right] e^{-fE_i/(E_m - E_i)}$$

where p is the pressure, V is the relative volume, E_c is the cold compression energy:

$$\bar{\varepsilon}^{\,p} = \int_0^t \left(\frac{2}{3}\, D_{ij}^{p} D_{ij}^{p}\right)^{1/2} dt,$$

$$x = 1 - V$$

and E_m is the melting energy:

$$E_m(x) = E_c(x) + 3R'T_m(x)$$

which is in terms of the melting temperature $T_m(x)$:

$$T_m(x) = \frac{T_{mo}\exp(2ax)}{V^{2(\gamma_o - a - \frac{1}{3})}}$$

and the melting temperature at $r = r_0$, T_{mo}.

In the above equation, R' is defined by

$$R' = \frac{R\rho}{A}$$

where R is the gas constant and A is the atomic weight. If R' is not defined, LS-DYNA2D computes it with R in the cm-gram-microsecond system of units.

The yield strength σ_y is given by:

$$\sigma_y = \sigma_0' \left[1 + b'pV^{1/3} - h\left(\frac{E_i - E_c}{3R'} - 300\right)\right] e^{-fE_i/(E_m - E_i)}$$

if E_m exceeds E_i. Here, σ_0' is given by:

$$\sigma_y = \sigma_0' \left|1 + \beta(\gamma_i + \varepsilon^{-p})^{n}\right.$$

where ε_i is the initial plastic strain. Whenever σ_0' exceeds σ_m, σ_0' is set equal to σ_m. After the material melts, σ_y and G are set to zero.

If the coefficients ECO,...EC9 are not defined above, LS-DYNA2D will fit the cold compression energy to the ten term polynomial expansion:

$$E_c = \sum_{i=0}^{9} EC_i \eta^i$$

where EC_i is the ith coefficient and $\eta=\rho/\rho_0-1$. The least square method is used to perform the fit.

Material Type 12 (Johnson/Cook Plasticity Model)

G *G*	Shear modulus.
A *A*	See equation (1).
B *B*	See equation (1).
N *n*	See equation (1).
R *r*	See equation (1).

M *m*	See equation (1).
TM T_{melt}	Melt temperature
TO *To*	Room temperature.
EPSO *Eo*	Effective plastic strain rate.
HCP *c*	Specific heat.
PC *pc*	Pressure cutoff (pc < 0.0).
D1 d_1	See equation (2).
D2 d_2	See equation (2).
D3 d_3	See equation (2).
D4 d_4	See equation (2).
D5 d_5	See equation (2).
IT *i*	Iteration options: =0: no iterations. =1: LS-DYNA2D iterates to determine a more accurate point on the stress-strain curve.

The Johnson/Cook model is described in reference [11]. This includes strain rate hardening, thermal softening, and a complex damage model. The equations describing the flow stress versus effective plastic strain and failure strain are as follows:

$$\sigma_y = \left(A + B \bar{\varepsilon}^{p^n} \right) \left(1 + c \ln \dot{\varepsilon}^*\right) \left(1 - T^{*m}\right)$$

where *A, B, C, n,* and *m* are input constants,

$\bar{\varepsilon}^p$ effective plastic strain

$\dot{\varepsilon}^* = \dfrac{\dot{\bar{\varepsilon}}^p}{\dot{\varepsilon}_0}$ effective plastic strain rate for $\dot{\varepsilon}_0 = 1\ s^{-1}$

$T^*=T_r/T_m$=homologous temperature

Constants for a variety of materials are also provided in [11].

Due to the nonlinearity in the dependence of flow stress on plastic strain, an accurate value of the flow stress requires iteration for the increment in plastic strain. However, by using a Taylor series expansion with linearization about the current time, we can solve for σ_y with sufficient accuracy to avoid iteration.

The strain at fracture is given by

$$\varepsilon^f = \left[D_1 + D_2 \exp D_3 \sigma^*\right] \left[1 + D_4 \ln \dot{\varepsilon}^*\right] \left[1 + D_5 T^*\right]$$

where σ^* is the ratio of pressure divided by effective stress:

$$\sigma^* = \frac{p}{\sigma_{eff}}$$

Fracture occurs when the damage parameter

$$D = \sum \frac{\Delta \bar{\varepsilon}^p}{\varepsilon^f}$$

reaches the value of 1.

Material Type 13 (Power Law Plasticity)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

K *k*	See equation below.
M *m*	See equation below.

Elastoplastic behavior with isotropic hardening is provided by this model. The yield stress, σ_y, is a function of plastic strain and obeys the equation:

$$\sigma_y = k\left(\varepsilon_e + \bar{\varepsilon}^p\right)^n$$

ε_e is the elastic strain to yield and where $\bar{\varepsilon}^p$ is the effective plastic strain. ***missing***

.***missing***

Material Type 16 (Pseudo Tensor Geological Model)

Default heading: Material Type #16 (Pseudo Tensor Geological Model)

G *G*	Shear modulus (constant Shear modulus model).

PR ν	Poisson's ratio (constant Poisson's ratio model).
SIGF sigf	Tensile cutoff. (Maximum principal stress for failure.)
A0 a_0	Cohesion.
A1 a_1	Yield function constant.
A2 a_2	Yield function constant.
A0F a_{0f}	Cohesion for failed material.
A1F a_{1f}	Pressure hardening coefficient for failed material.
B1 b_1	Damage scaling factor.
PER p	Percent reinforcement.
ER E_r	Elastic modulus for reinforcement.
PR ν_r	Poisson's ratio for reinforcement.
SIGY σ_y	Initial yield strength.
ETAN E_t	Tangent modulus.
LCP lc_1	Load curve giving rate sensitivity for principal material.
LCR lc_2	Load curve giving rate sensitivity for reinforcement.
NPTS n	Number of points in yield stress-effective plastic strain curve or yield stress-pressure curve; (n£16).
ES σ_1 σ_2 ... σ_n	Yield stress.
EPS ε_{p1} ε_{p2} ... ε_{p3}	Effective plastic strain.
P p_1 p_2 ... p_n	Pressure.

See the LS-DYNA2D manual for a description of this model.

Material Type 25 (Inviscid Two Invariant Geologic Cap Model)

G G	Shear Modulus.
K K	Bulk Modulus.
ALPHA α	α.
BETA β	β.
GAMMA γ	γ.
THETA θ	θ.

R R	R.
D D	D.
X0 X_0	X_0.
CC C	C.
T T	Tension cutoff.
NPLOT nplot	Save the following variable for plotting in ORION: =1: k =2: X =3: e_v^p =4: J_1 =5: $(J_2)^{1/2}$ =6: $(J2\|_L)^{1/2}$ =7: $(J2\|_{tr})^{1/2}$ =8: MTYPE =9: number of iterations
LTYPE ltype	Variable ltype. =1: soil/concrete (cap contracts) =2: rock (cap doesn't contract)

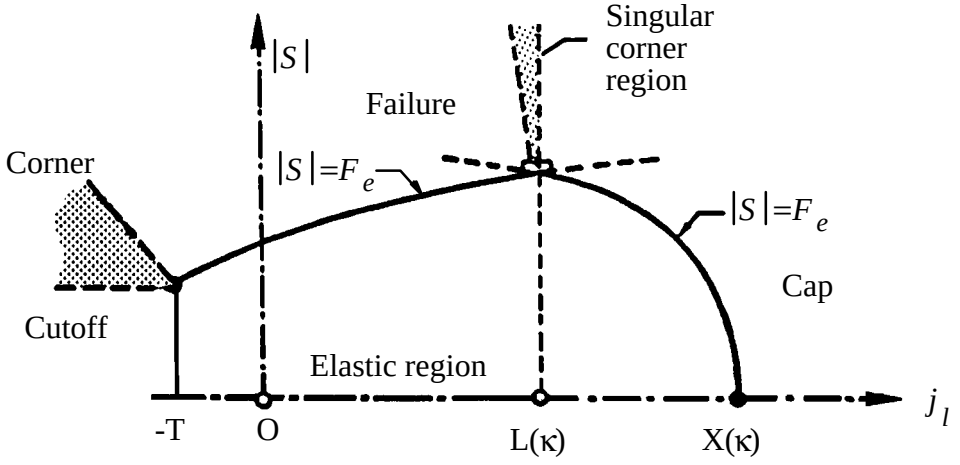


Figure 19-5. The yield surface of the two-invariant cap model in pressure/J_2-deviator space. $f_1 = 0$, $f_2 = 0$, and $f_3 = 0$, denote the failure envelope, the hardening cap surface, and the tension cut-off surface, respectively.

The shaded area in Figure 19-5 is the “compressive corner regions”.

$$f_1\left(\underset{\sim}{\sigma}\right) := \left\| \underset{\sim}{s} \right\| - F_e(J_1) \qquad \text{for} \quad -T \le J_1 < \kappa$$

$$f_2\left(\underset{\sim}{\sigma}, \kappa\right) := \left\| \underset{\sim}{s} \right\| - F_c(J_1, \kappa) \qquad \text{for} \quad \kappa \le J_1 < X(\kappa)$$

$$f_3\left(\underset{\sim}{\sigma}\right) := -T - J_1 \qquad \text{for} \quad -T \le J_1 < (\kappa),$$

where $J_1 := \operatorname{tr} \underset{\sim}{\sigma}$, $\left\| \underset{\sim}{s} \right\| := \sqrt{\underset{\sim}{s} : \underset{\sim}{s}}$, with $\underset{\sim}{s} := \underset{\sim}{\sigma} - \frac{1}{3}\left(\operatorname{tr} \underset{\sim}{\sigma}\right)\underset{\sim}{1}$.

In addition, T > 0 is an material constant referred to as the **tension cutoff**. Note that the following standard conventions in soil mechanics, we have **assumed compression and compaction positive.** Functional forms for F_e and F_c used are:

$$F_e(J_1) := \left[\alpha - \lambda \exp(-\beta J_1) + \theta J_1\right]$$

$$F_c(J_1, \kappa) := \sqrt{F_e^2(\kappa) - \frac{[J_1 - \kappa]^2}{R^2}},$$

where a > 0, λ > 0, β > 0, θ > 0, and R > 0 are material parameters. In addition, X(κ) is a function of the hardening parameter κ defined as

$$X(\kappa) := \kappa + R F_e(\kappa).$$

$$\dot{\kappa} = \begin{cases} 0 & \text{if } \operatorname{tr} \dot{\varepsilon}^p < 0,\ J_1 = \kappa \text{ and } \dot{f}_1 = \dot{f}_2 = 0 \\ \max\left[h'(\kappa) \operatorname{tr} \dot{\varepsilon}^p ; \dfrac{\frac{d|s|}{dt}}{F_e(\kappa)} \right] & \text{if } J_1 = \kappa,\ \dot{f}_1 = 0, \text{ and } \dot{f}_2 < 0 \\ h'(\kappa) \operatorname{tr} \dot{\varepsilon}^p & \end{cases}$$

otherwise

$$\bar{h}(\kappa) := W\left\{1 - \exp\left[-D\left(X(\kappa) - X_o\right)\right]\right.$$

20. LS-DYNA3D Commands and Materials

Analysis options are code dependent. They can be set either in the control section of the LS-INGRID input file or in the graphics phase. These commands become active when LS-DYNA3D output is selected with the DN3D command (see *Control Commands*).

ARBITRARY	Node and element numbering is arbitrary. (LS-902 and later).
BRODE	Define Brode function parameters.

Options:

YLD yld	Yield (Ktons)
HEIGHT *h*	Height of burst
XBO *x*	Coordinates of Brode origin (space, time) in LS-INGRID units.
YBO *y*	
ZBO *z*	
TBO *t*	
CL *cl*	Conversion factor - ft. to DYNA length units (default = meters).
CT *ct*	Conversion factor - ms to DYNA time units (default - seconds).
CP *cp*	Conversion factor - psi to DYNA pressure units (default = Pascals).
;	Terminate Brode function input.

Note: If "RANG" "COEF", and "GFUN" are specified, a "modified" Brode function will be used in DYNA; otherwise, straight Brode is used.

RANG r_1 ... r_5	Range values for Brode function.
COEF c_1 ... c_8	Coefficient values for Brode function.
GFUN g_1 ... g_7	GFUNC values for Brode Function.

The Brode function is applied to pressure surfaces with load curve number -1.

BRUL *n*	Begin definition of user specified integration rule for beams number *n*.

Options:

NPTS n s_1 t_1 w_1 <m_1> ... s_n t_n w_n <m_n>	Input n integration points with the parametric coordinate, (s,t) and the weight, w. This terminates the rule.
BUPD opt	Flag for updating coordinates of reference node for beam elements. Values of opt are "on" or "off". (LS-910 and later).
CUNI length time force	Unit conversion factors for coupling between LS-DYNA3D and CAL3D or MADYMO3D. (LS-910 and later).
D2R m_1 m_2	Convert material m_1 from deformable to rigid. If m_2 is 0, then this is an independent rigid body. Otherwise, m_2 is the master rigid body material. If a restart file definition has been initiated, then this command applies to the restart. Otherwise, it applies to the main DYNA3D input. (LS-920 and later.)
D3HSP	Additional output options for the D3HSP and message files.

Options:

DEBUG opt	Option for producing debug output on calculation progress in the message file. Values for opt are either "on" or "off". (LS-910 and later).
ECHO opt	Additional suppression options for printout. (LS-910 and later). =0: all data is printed. =1: nodal printing is suppressed. =2: element printing is suppressed. =3: both node and element printing are suppressed.
IKEDIT n	Number of time steps between writing global statistics data to D3HSP file. (Default = 1000.)
SUPP opt	Performs suppression of output echo. Values for opt are either "on" or "off". (LS-910 and later).
TSTEP opt	Print flag for element time step sizes on first cycle. Values for opt are either "on" or "off". (LS-910 and later).
;	Terminate the D3HSP command.

DBQT *i*

Change default bulk viscosity type from 1 to *i*:
=1: standard LS-DYNA3D

DELT Δt

Set time step for mass scaled calculations to Δt. Note that this is an advanced option. Normally LS-DYNA3D sets the time step. Study the mass scaling option in LS-DYNA3D before using this option. (LS-910 and later).

DHGQ *Qh*

Set default hourglass viscosity from .10 to *Qh*.

DHQT *i*

Set default hourglass viscosity type from 1 to *i*:
=1: standard LS-DYNA3D
=2: Flanagan-Belytschko viscous form
=3: Flanagan-Belytschko viscous form with exact volume integration.
=4: Flanagan-Belytschko stiffness form
=5: full Flanagan-Belytschko stiffness form with exact volume integration.

DQL *Ql*

Set default linear bulk viscosity for .06 to *Ql*.

DQQ *Qq*

Set default quadratic bulk viscosity from 1.5 to *Qq*.

DROPTS

Select dynamic relaxation options. There are three separate methods in LS-910 and later for performing dynamic relaxation. The first uses the SYSD or LCDAMP commands and is designated "Method A". This is the recommended method. The second is designated "Method B" and requires the options below which must be set with care. The third is a method due to Papadrakakis and is designated "Method C". Only one method should be used at a time to avoid confusion.

Options:

DRFCTR *d*

Damping factor expressed as $V_{n+1}=d*V_n$. This should be set with care based on the formulas in the DYNA3D Course Notes. (Method B)

DRTOL tol

Tolerance on distortional kinetic energy for determining convergence. (Method B)

NRCYCK *n*

Number of time steps between convergence checks. (Method B)

TSSFDR tssfdr

Time step scale factor during dynamic relaxation. (Method B)

DRTERM *t*

Termination time for dynamic relaxation simulation should convergence not be obtained.

(default = infinity). (LS-910 and later). (Method B)

; End of dynamic relaxation options.

FLUID Set ALE and Eulerian options. (VECALE, LS-930 and later.)

Options:

ADVECTION *opt* Set the advection formulation.
opt = 1: first order. (SALE Method)
opt = 2: second order. (Benson HIS)
opt = 3: second order. (Van-Leer)

ALE The element formulation is Arbitrary Lagrangian-Eulerian.

EULERIAN The element formulation is Eulerian.

LAGRANGIAN The element formulation is Lagrangian (default).

MAT *m* These options apply to material *m*. (The default is that the specified fluid options apply globally to the model.)

NCYCLES *n* The number of cycles between smoothing and advection (ALE) or smoothing (Eulerian) is *n*.

RELAX1 r_1 Weight for simple average relaxation method.

RELAX2 r_2 Weight for Kikuchi relaxation method.

RELAX3 r_3 Weight for isoparametric relaxation method.

RELAX4 r_4 Weight for equipotential relaxation method.

START t_s Start time for ALE.

STOP t_e Stop time for ALE.

; End of dynamic relaxation options.

FRES m_1 m_2 ... ; Perform a full restart. Materials m_1, m_2, ... are to be remapped. If "FRES ;" is input, then all materials will be remapped.

GMPRT	Input general printing option intervals. (LS-910 and later).
Options:	
ABSTAT *t*	Airbag statistics.
AVSFLT *t*	AVS filter.
BCOUT *t*	Boundary condition forces.
BELT *t*	Seat belt output file.
DEFGEO *t*	Smug animator file.
DEFORC *t*	Discrete element.
ELOUT *t*	Element data.
GEFORC *t*	Geometric entity resultants.
GLSTAT *t*	Global data.
JOINTS *t*	Joint file.
MATSUM *t*	Material energies.
MOVIE *t*	Movie-BYU output file.
MPGS *t*	MPGS output.
NCFORCE *t*	Nodal interface forces.
NFG *t*	Nodel force groups.
NODOUT *t*	Nodal point data.
RBOUT *t*	Rigid body acceleration output.
RCFORC *t*	Resultant interface forces.
RWFORC *t*	Rigid wall forces.
SECFORCE *t*	Section forces.
SIDB *t*	Sliding interface database.
SPCFORC *t*	SPC reaction forces.
SWFORC *t*	Nodal constant resultants.
TRACER	
VARIABLE *typ icomp*	Components for ASCII state output. *typ* can be **AVS**, **MOVI**E or **MPGS**. The component number is *icomp*.
;	Terminate this command.
GRAV g_x g_y g_z	Gravity acceleration vector.
HGENERGY on/off	Option for computing hourglass energy dissipation. (Default=Off.) (LS-910 and later).
IARB on/off	Selection for material input method. If on, then the material input is broken into separate constitutive model, equation-of-state and section property sections. LS-INGRID can convert from one method to another during generation. The last method selected applies to the output file. (Default=Off.) (LS-910 and later).
IRDMS on/off	Turn on deformable to rigid switching. (LS-920 and later.)
ITSS t_o	Initial time step size. This is optional input for LS-DYNA3D. If t_o is zero, LS-DYNA3D picks the initial time step size.

LCDAMP lc	System damping is specified by load curve lc. (LS-902 and later).
LCGX lcgx	Load curve number for *X*-body load. (default=1).
LCGY lcgy	Load curve number for *Y*-body load. (default=1).
LCGZ lcgz	Load curve number for *Z*-body load. (default=1).
LCRX lcrx	Load curve number for *X*-centrifugal load. (default=1).
LCRY lcry	Load curve number for *Y*-centrifugal load. (default=1).
LCRZ lcrz	Load curve number for *Z*-centrifugal load. (default=1).
LCMAX *lc*	*lc* is a load curve which specifies the maximum time step as a function of time.
MVMA	Output is generated which is compatible with MVMA-DYNA3D.
NCPU *n*	Use n CPU's for parallel processing. (LS-920 and later.)
NEWC	Use new contact formulations. (LS-902, VEC/DYNA3D). This turns on the eroding contact in VEC/DYNA3D.
NSTEP *n*	The number of time steps for mass scaled calculations is *n*. Note that this is an advanced option and normally LS-DYNA3D sets the time step. Study the mass scaling option in LS-DYNA3D before using this option. (LS-910 and later).
OPIFS *n*	Output interval for interface file.
PASS opt	Option for sorting parallel assembly of the right hand side. Values for opt are "on" or "off". (LS-920 and later).
PERCENT *n*	Maximum allowable change in total energy in percent.
PLTI Δt	Node and element data dump interval for TAURUS post-processing.
PRTI Δt	Node and element data dump interval for high

	speed printer.
PSPO iopt	Plane stress iteration flag. =1: iterative plasticity with 3 secant iterations (default). =2: full iterative plasticity. =3: radial return non-iterative plasticity. (quick and very dirty.)
R2D m_1 m_2	Convert material m_1 from rigid to deformable. If a restart file definition has been initiated, then this command applies to the restart. Otherwise, it applies to the main DYNA3D input. (LS-920 and later.)
RDENERGY on/off	Option for computing stone wall energy dissipation. (Default=Off.) (LS-910 and later).
RDMT *m*	Delete material *m*. This applies to the restart number selected by the RNUM command.
RDSI *s*	Delete sliding interface *s*. This applies to the restart number selected by the RNUM command.
REIN *i*	Hughes-Liu shell normal initialization count. *i* = -2: unique nodal fibers per Hughes-Liu =-1: compute normals each cycle (default). =1: compute on restart. =*n*: compute on restart and every nth cycle.
REST name	Set the family name for restart input file generation to name.
RHVC *h*	The default hourglass viscosity for restart is set to *h*. This applies to the restart number selected by the RNUM command.
RIRDMS on/off	Turn on rigid to deformable switching. (LS-920 and later.)
RLBV *l*	The default linear bulk viscosity for restart is set to *l*. This applies to the restart number selected by the RNUM command.
RNUM *n*	Restart commands apply to restart number *n*.
RPLT *t*	The plot interval for restart is set to *t*. This applies to the restart number selected by the RNUM command.
RPRT *t*	The print interval for restart is set to *t*. This applies to the restart number selected by the RNUM command.

RQBV *q*
The default quadratic bulk viscosity for restart is set to *q*. This applies to the restart number selected by the RNUM command.

RTERM *t*
The termination time for this restart is *t*. This applies to the restart number selected by the RNUM command.

RTSF *s*
The time step scale factor for restart is set to *s*. This applies to the restart number selected by the RNUM command.

RWPNAL *p*
Scale factor for rigid body nodes impacting rigid walls. If *p*=0.0, then this capability is ignored.

SBRF *n*
Number of time steps between restart dumps is *n*.

SEQUENTIAL
Use sequential node, element and material numbering. (Default)

SFSI *s*
Sliding interface scale factor (default = 0.1).

SIOPT
Additional sliding interface options. (LS-910 and later).

Options:

ENER opt
Option for determining sliding interface energy dissipation. Values for opt are "on" and "off".

CHECK opt
Option for performing initial penetration checks on contact interfaces. Values for opt are "on" and "off".

OFFSET *n*
Set shell thickness offset option to n:
=0: thickness is not considered in two surface contacts.
=1: thickness is considered but rigid bodies are excluded.
=2: thickness is considered including rigid bodies.

ORIE opt
Option for automatically reorienting normals of shell contact segments during initialization. Values for opt are "on" and "off".

PSOPT *n*
Penalty stiffness option:
=1: use minimum of master segment and slave node (default).
=2: use master segment stiffness (old way).
=3: use slave node value.
=4: use slave node value area or mass

weighted.
=5: same as 4 but inversly proportional to the shell thickness.

THIN opt — Option for including thinning of shells in thickness offsets. Values for opt are "on" and "off".

; — Terminate SIOPT command.

SRUL n — Begin definition of user specified integration rule for shell number n.

Options:

MATE — Include optional material selection. (default=off)

NPTS *n* t_1 w_1 <m_1> ... t_n w_n <m_n> — Input *n* integration points with the parametric coordinate, *t*, the weight, *w* and the optional material number, *m*. This terminates the rule.

STYP *s* — Default shell formulation type
s = **HUGH**ES: use Hughes-Liu shell formulation (default).
= **BELY**TSCHKO: use Belytschko-Lin-Tsay shell theory.

SWENERGY on/off — Option for computing stone wall energy dissipation. (Default=Off.) (LS-910 and later).

SYSD *d* — System damping constant *d*. (MVMA/DYNA3D, VEC/DYNA3D, LS-902 and later.)

TAURUS — Additional ouput options for the D3PLOT, D3IFF and D3THDT files.

Options:

AVER opt — Output averaged accelerations from velocities in file "nodout" and the time history database file, "d3thdt". (LS-910 and later).

CMSO opt — Composite material stress output option. Values for opt are "global" and "local". (LS-910 and later).

DRDB opt — Produce a separate TAURUS database for the dynamic relaxation option. Values for opt are "on" or "off". (LS-910 and later).

IFDT D*t* — Output interval for interface force database. If zero, the default is the same as for complete state dumps.

INT4 *n*	Number of additional integration point history variables written to the TAURUS database for shell elements.
INT8 *n*	Number of additional integration point history variables written to the TAURUS database for solid elements.
SEPARATE opt	Option for separating D3PLOT file into one state per family output member. Values for opt are "on" or "off". (LS-910 and later).
TINT *n*	Number of through thickness integration points written to TAURUS database. (default=3).
;	Terminate TAURUS command options.
TCYCLE *n*	The termination cycle is *n*. (LS-910 and later).
TEO *i*	Thermal effects option =0: no thermal effects. =N: nodal temperatures are defined in input and are scaled according to a time function. N is the load curve number. =-1: each time step a new temperature state is read from a disk file. The time word at the beginning of each temperature state is ignored. = -2: at each time step a temperature state is interpolated from the temperature state in a disk file. Therefore the time words at the beginning of each temperature state is used. =-3: the disk file containing temperatures has only one state. The initial state is assumed to be zero.
TERM *t*	Terminate dynamic time integration at time *t*.
TINV *n*	Number of time steps between dumps of reaction history blocks.
TSLIMIT Δt	The minimum time step for shell elements of type 3, 18, 19 and 24 cannot go below Δt. To enforce this condition, the element stiffness is artificially softened. This is useful when pretty pictures are more important than good results.
TSORT opt	Sort triangular elements to treat degenerate quadrilateral elements with the C_0 triangular shell formulation. Values for opt are "on" or "off". (LS910 and later).

TSSF *s*	Scale factor on time step size.
TUPD	Modify shell thickness based on membrane strains (default doe not modify shell thickness).
V90	Output is compatible with LS-DYNA3D version 902.
V91	Output is compatible with LS-DYNA3D version 910.
V92	Output is compatible with LS-DYNA3D version 920.
V93	Output is compatible with LS-DYNA3D version 930. This produces the LS-DYNA3D keyword based input.
VEC	Output is compatible with VEC-DYNA3D.
VEC92	Output is compatible with VECALE.
WARP ang	Shell element warpage angle in degrees. If a warpage greater than this angle is found, a warning message is printed. (default=20.0) (LS-902 and later).
WEDGE	Normally, LS-INGRID does not allow the generation of wedge elements. This command turns on the support for 6-node and 4-node solid elements.

20.1 LS-DYNA3D MATERIAL INPUT

LS-DYNA3D material input is possible after the DN3D command is input (see *Control Commands*). The form of this input is: MAT *n* TYPE *m* {options specific to material type *m*} {general material options} ENDMAT. *n* is a material name which is assigned a number as input. Therefore, the materials should be defined in order before any additional use of materials is made.

20.2 GENERAL MATERIAL OPTIONS

ANGLES β_1 β_2 ... β_n	Input angles for laminated materials. *n* is the number of integration points; thus, this command cannot be used until after the QUAD command has been used to specify the number of integration points for the current material.

BEAM	This material is defined for two node beam elements only.
BFORM *s*	Beam formulation type s = "HUGH": Hughes-Liu s = "BELY": Belytschko-Schwer s = "TRUS": Truss
BQL Q_l	Change linear bulk viscosity for .06 to Q_l.
BQQ Q_q	Change quadratic bulk viscosity from 1.5 to Q_q.
BRFORM *s*	Brick element formulation type (LS-920 and later): s = 1: standard single point brick s = 2: fully integrated brick element
BQT *i*	Change bulk viscosity type from 1 to *i*: =1: standard LS-DYNA3D (not much choice)
CAREH *a*	Cross sectional area for Belytscko-Schwer beam.
EOS eost	Begin defining equation-of-state type eost for the current material definition. Each equation-of-state is terminated by the ENDEOS command.
HEAD	Replace default heading (typed on the next line).
HGQ Q_h	Change hourglass viscosity from .10 to Q_h.
HGQT *i*	Change type of hourglass viscosity from 1 to *i*: =1: standard LS-DYNA3D =2: Flanagan-Belytschko viscous form =3: Flanagan-Belytschko viscous form with exact volume integration. =4: Flanagan-Belytschko stiffness form =5: full Flanagan-Belytschko stiffness form with exact volume integration.
IMGL	Initialize material for gravity loads.
IRR *Irr*	Moment of area along *r*-axis for Belytschko-Schwer beam.
IRULE **GAUS**S	Use gauss quadrature (default).
IRULE **TRAP**EZOIDAL	Use trapezoidal integration.
IRULE **USER** *n*	Use user defined integration rule *n*.
ISS *Iss*	Area moment of inertia along *s*-axis for

Belytschko-Schwer beam.

ITT *Itt* — Area moment of inertia along *t*-axis for Belytschko-Schwer beam.

LTMN — The local *t*-axis for thick shell elements of this material is the shortest direction through the brick.

LTMX — The local *t*-axis for thick shell elements of this material is the longest direction through the brick.

MAT *m* — Begin material definition *m*. Each material definition is terminated by the ENDMAT command.

MDMP lc scale — Apply mass weighted damping to material mat. The magnitude is scale which is multiplied by load curve *lc*. (LS-920 and later.)

QUADRATURE *i* — Select quadrature rule *i*.

For beams the rules are:
- *i* = 1: truss element
- *i* = 2: 2 X 2 Gauss (default)
- *i* = 3: 3 X 3 Gauss
- *i* = 4: 3 X 3 Lobatto
- *i* = 5: 4 X 4 Gauss

For four node shells the rules are:
- *i* = 1: membrane element
- *i* = 2: 2 point Gauss (default)
- *i* = 3: 3 point Gauss
- *i* = 4: 4 point Gauss
- *i* = 5: 5 point Gauss

For eight node thick shells the rules are:
- *i* = 1: membrane element
- *i* = 2: 2 point Gauss (default)
- *i* = 3: 3 point Gauss
- *i* = 4: 4 point Gauss
- *i* = 5: 5 point Gauss

RAYD b — Rayleigh (stiffness proportional) damping coefficient. (LS-920 and later.)

REPOSITION — Reposition deformable materials which are positioned relative to CAL3D/MADYMO3D bodies at initialization time. (LS-920 and later)

Options:

ELLIPSE *n* — Slave to MADYMO3D ellipse *n*.

PLANE *n*	Slave to MADYMO3D plane *n*.
SEGMENT *n*	Slave to CAL3D segment *n*.
SYSTEM *n*	Slave to MADYMO3D system *n*.
RO *m*	Density (required - no default).
SAREA *a*	Shear area for Belytschko-Schwer beam.
SFORM *s*	Shell formulation type *s* = "HUGH": Hughes-Liu. *s* = "BELY": Belytschko-Lin-Tsay *s* = "BCZ": BCIZ triangular shell *s* = "C0T": C_0 triangular shell. *s* = "MEMB": B-L-T membrane. *s* = "SRHL": S/R Hughes-Liu *s* = "CRHL": Corotational Hughes-Liu. *s* = "YASE": Engelmann-Whirley's "YASE" shell. (Not recommended.)
SHELL	This material is defined for four node shell elements only.
SLOC *s*	Factor specifying offset of the local *s* axis. =-1: reference surface is at bottom plane of shell. =0: reference surface is at center plane of shell =1: reference surface is at upper plane of shell
STHICK thick	The default thickness along the element local *s* axis is thick (beams and shell).
TLOC *t*	Factor specifying offset of the local *t* axis. =-1: reference surface is at bottom plane of shell =0: reference surface is at center plane of shell. =1: reference surface is at upper plane of shell.
TSHELL	This material is defined for thick (8-node solid) shell elements only.
TTHICK thick	The default thickness along the element local *t* axis is thick (beams only).
TYPE *n*	The current material is of type *n*.

Material Type 1 (Elastic)

Default heading: Material Type #1 (Elastic)

Input any two of the following.

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR ν	Poisson's ratio.

Material Type 2 (Orthotropic Elastic)

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA υ_{ca}	
PRCB υ_{cb}	
GAB G_{ab}	
GBC G_{bc}	
GCA G_{ca}	
AOPT aopt	Material axes option (Figure 20-1). =0.0: locally orthotropic with materials axes by element nodes n_1, n_2, and n_4, (see Figure 20-1). =1.0: locally orthotropic with materials axes by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below. =3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.

ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

The material law that relates stresses to strains is defined as:

$$\underset{\sim}{C} = \underset{\sim}{T}^T \underset{\sim}{C}_L \underset{\sim}{T},$$

where \o(T, ~) is a transformation matrix, and \o(C, ~)L is the constitutive matrix defined in terms of the material constants of the orthogonal material axes, a, b, and c. The inverse of \o(C, ~)L is defined as

$$\underset{\sim L}{C^{-1}} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ba}}{E_b} & -\frac{\nu_{ca}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ab}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{cb}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ac}}{E_a} & -\frac{\nu_{bc}}{E_b} & \frac{1}{E_c} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{bc}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{ca}} \end{bmatrix}$$

Note that $\frac{\nu_{ab}}{E_a} = \frac{\nu_{ba}}{E_b}, \frac{\nu_{ca}}{E_c} = \frac{\nu_{ac}}{E_a}, \frac{\nu_{cb}}{E_c} = \frac{\nu_{bc}}{E_b}$.

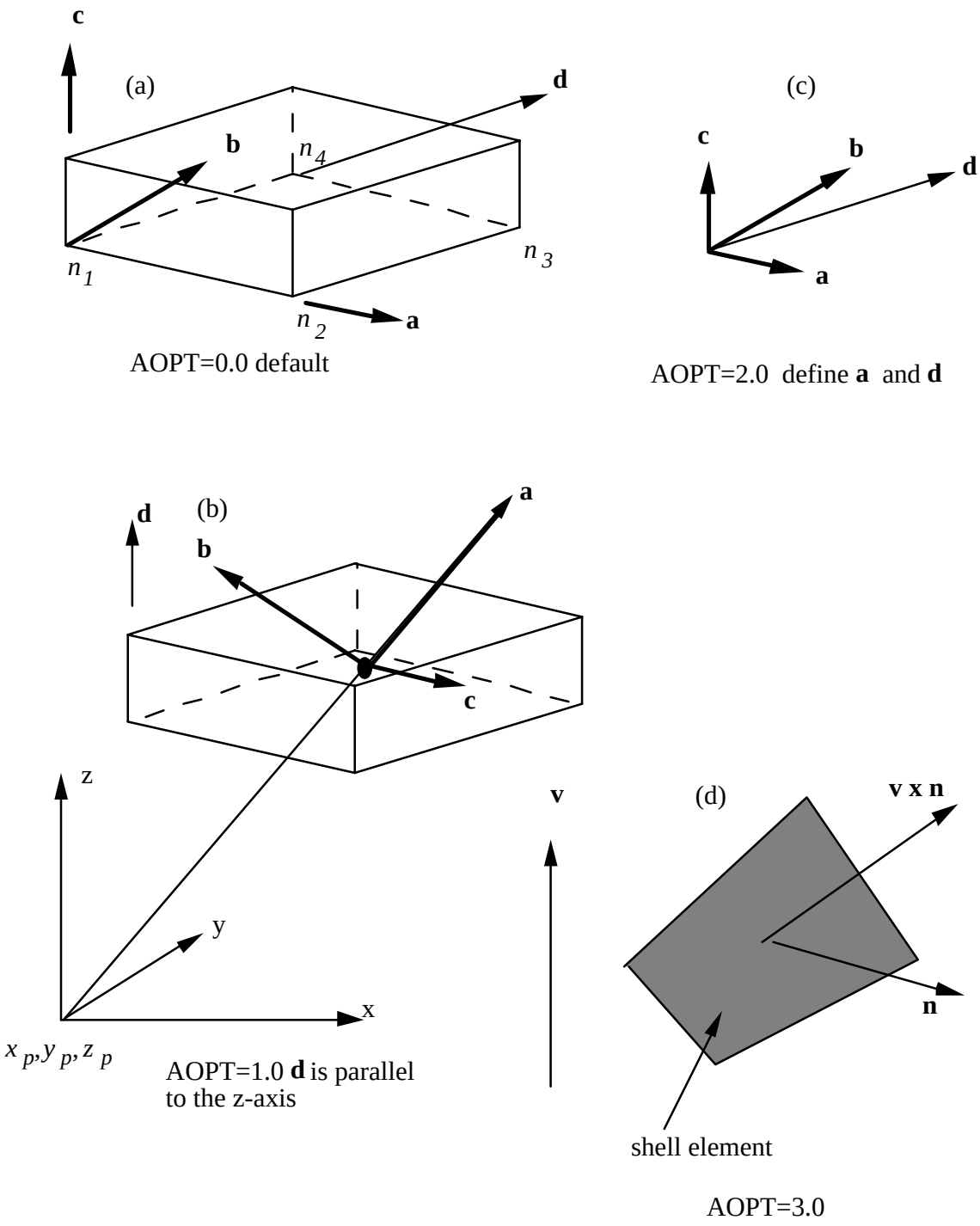


Figure 20-1. Options for determining principal materials axes: (a) AOPT = 0.0, (b) AOPT = 1.0, and (c) AOPT = 2.0, (d) AOPT = 3.0.

Material Type 3 (Kinematic/Isotropic Elastic/Plastic)

Default heading: Material Type #3 (Elastic-Plastic)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

SIGY σ_y	Yield stress
ETAN E_t	Hardening modulus
BETA β'	Hardening parameter, $0 \le \beta' \le 1$
SC *c*	Strain rate parameter, C
SP *p*	Strain rate parameter, *p*

Strain rate is accounted for using the Cowper and Symonds model which scales the yield stress with the factor

$$1 + \left(\frac{\dot{\varepsilon}}{C} \right)^{1/p}$$

where $\dot{\varepsilon}$ is the strain rate.

Isotropic, kinematic, or a combination of isotropic and kinematic hardening may be specified by varying β' between 0 and 1. For β' equal to 0 and 1, respectively kinematic and isotropic hardening are obtained as shown in Figure 20-2. Effective stress is defined in terms of the deviatoric stress tensor, S_{ij} as:

$$\bar{\sigma} = \left(\frac{3}{2} S_{ij} S_{ij} \right)^{1/2}$$

where,

$$S_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij}$$

and effective plastic strain by:

$$\bar{\varepsilon}^p = \int_0^t d\bar{\varepsilon}^p$$

where t denotes time and

$$d\bar{\varepsilon}^{p} = \left(\frac{2}{3} d\varepsilon_{ij}^{p} d\varepsilon_{ij}^{p} \right)^{1/2}$$

For isotropic hardening ($\beta' = 1$) material model 12 requires less storage and is more efficient.

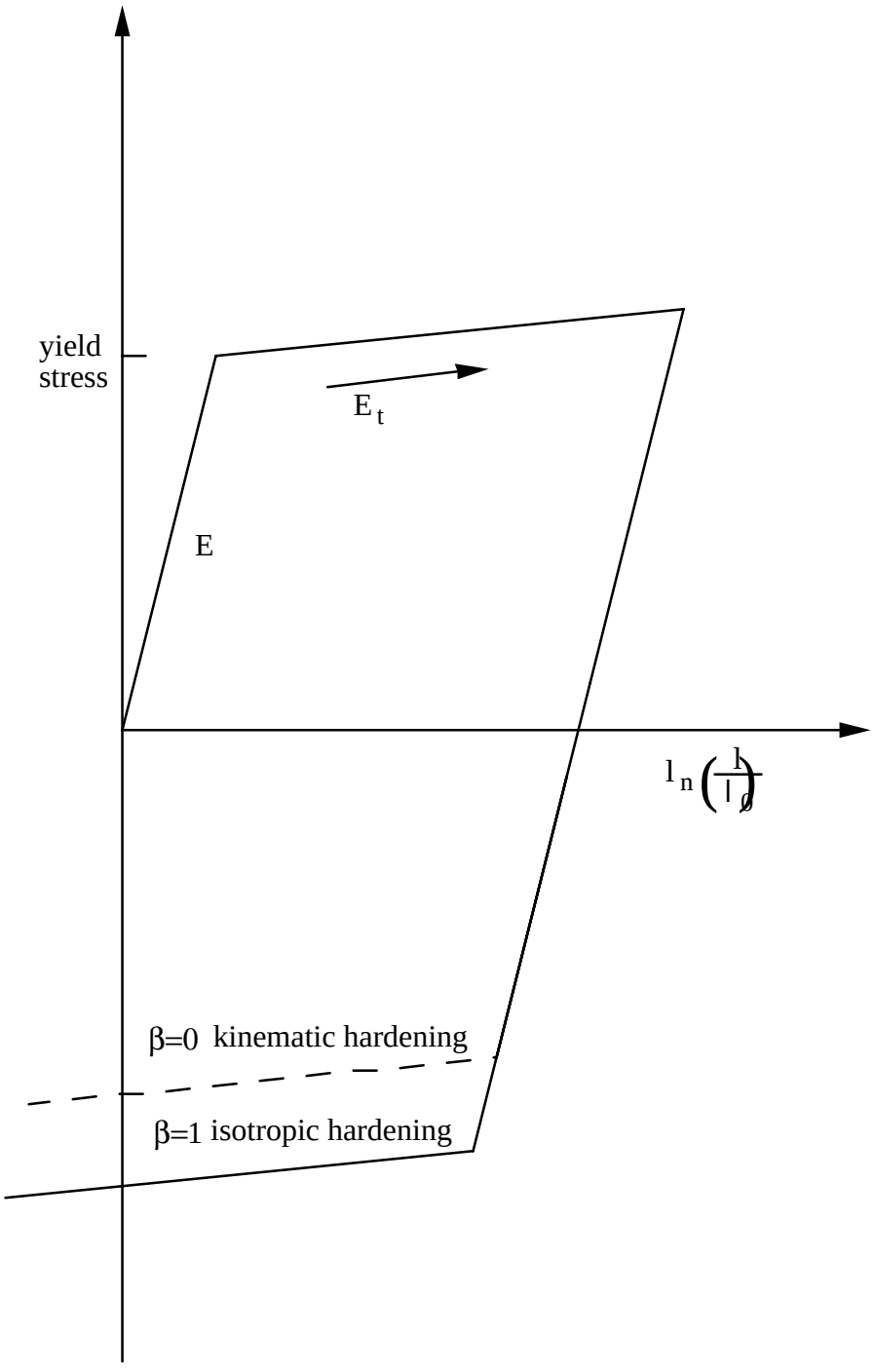


Figure 20-2. Elastic-plastic behavior with isotropic and kinematic hardening where l_0 and l are undeformed and deformed length of uniaxial tension specimen.

Material Type 4 (Thermo-Elastic-Plastic)

Default heading: Material Type 4 (Thermo-Elastic-Plastic)

NPTS n	Number of temperature values for which material constants are defined.
TEMP T_1 T_2...T_n	Temperatures.
E E_1 E_2...E_n	Young's moduli.
PR υ_1 υ_2...υ_n	Poisson's ratios.
ALPHA α_1 α_2...α_n	Coefficients of thermal expansion.
SIGY σ_{y1} σ_{y2}...σ_{yn}	Yield stresses.
ETAN E_{t1} E_{t2}...E_{tn}	Tangent moduli.

Material Type 5 (Soil and Crushable Foam)

Default heading: Material Type 5 (Soil and Crushable Foam)

Input any two of the following:

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options:*

AO a_0	Yield function constant
A1 a_1	Yield function constant
A2 a_2	Yield function constant
PC P_c	Pressure cutoff for tensile fracture
UL uopt	Unloading option =0: volumetric crushing =1: loading and unloading are the same. =2: hysteretic behavior.
ULD d	Unloading distance for option 2 above.
NPTS n	Number of points in volumetric strain versus pressure curve (n£10).
VS ε_{v1} ε_{v2} ... ε_{vn}	Volumetric strain values
P p_1 p_2...p_n	Pressures corresponding to volumetric strain values

The deviatoric, perfectly yield function, ϕ, is described in terms of the second invariant J_2.

$$J_2 = \frac{1}{2} s_{ij} \; s_{ij}$$

Pressure, p, and constants a_0, a_1, and a_2 as:

$$\phi = J_2 - \left[a_0 + a_1 p + a_2 p^2 \right].$$

On the yield surface, $J_2 = \frac{1}{3} \sigma_y^2$, where s_y is the yield stress, i.e.,

$$\sigma_y = \left| 3 \left(a_0 + a_1 p + a_2 p^2 \right) \right|^{1/2}$$

For elastic-perfectly plastic behavior a_1=a_2=0, and $(3a_0)^{1/2}$ defines the yield strength. The volumetric strain is given by the natural logarithm of the relative volume V. If the pressure drops below the cutoff value, PC, then it is reset to that value.

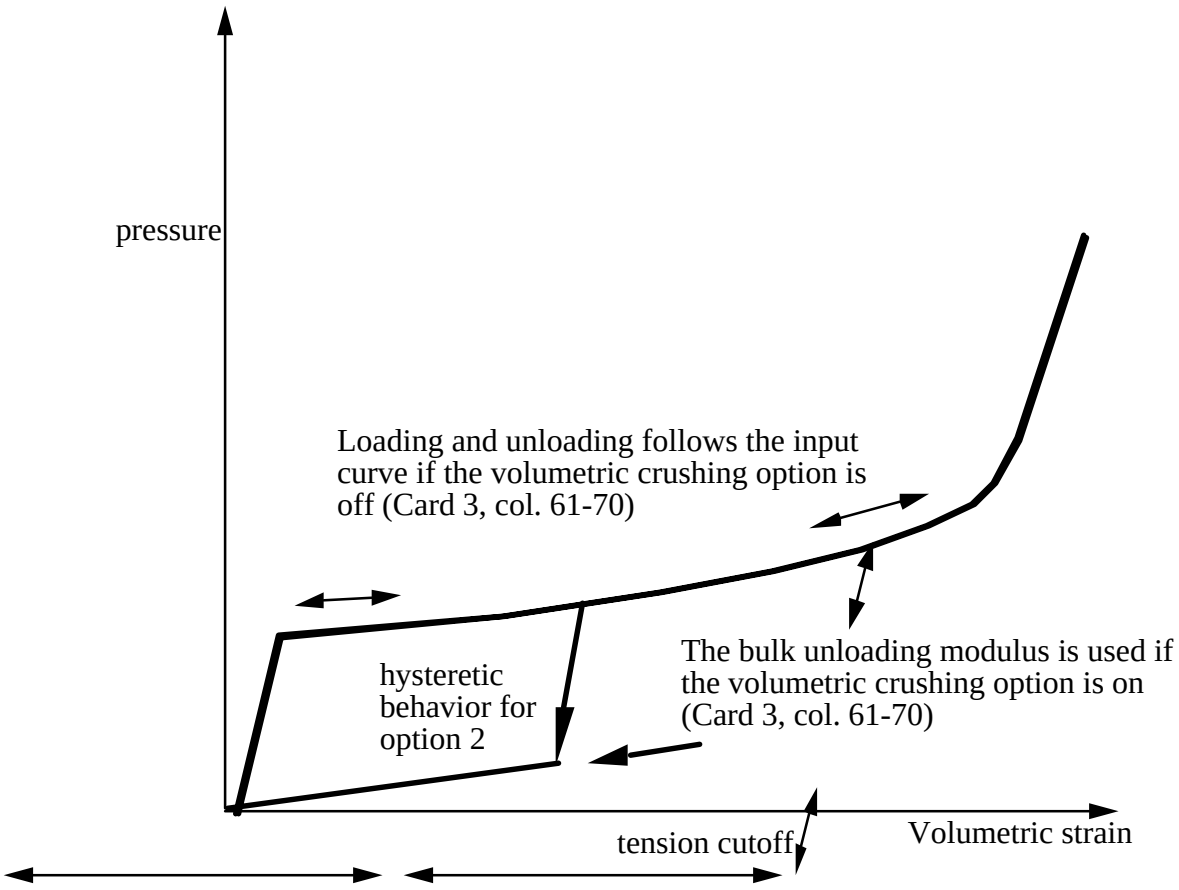


Figure 20-3. Volumetric strain versus pressure curve for soil and crushable foam model.

Material Type 6 (Viscoelastic)

G G_0	Short term shear modulus.
GI G_∞	Long term shear modulus.
K K	Bulk modulus.
BETA β	Decay constant.

The shear relaxation behavior is described by:

$$G(t) = G_\infty + (G_0 - G_\infty)\, e^{-\beta t}$$

A Jaumann rate formulation is used:

$$\overset{\nabla}{\sigma}'_{ij} = 2 \int_0^t G(t-\tau)\, D'_{ij}(\tau)\, dt$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{\sigma}_{ij}$, and the strain rate D_{IJ}.

Material Type 7 (Blatz-Ko Rubber)

Default heading: Material Type #7 (Rubber)

G G	Shear modulus.

The second Piola-Kirchhoff stress is computed as

$$S_{ij} = \mu \left(\frac{1}{V} C_{ij} - V^{-1/1-2\upsilon} \delta_{ij} \right)$$

where V is the relative volume, C_{ij} is the right Cauchy-Green strain tensor, and n is the Poisson's ratio which is set to .463 internally. This stress measure is transformed to the Cauchy stress, s_{ij}, according to the relationship

$$\sigma_{ij} = V^{-1} F_{ik} F_{jl} S_{lk}$$

where F_{ij} is the deformation gradient tensor.

Material Type 8 (High Explosive Burn)

Default heading: Material Type #8 (High Explosive Burn)

D D	Detonation velocity.
PCJ P_{CJ}	Chapman-Jouget pressure.

This material model requires an equation-of-state.

Material Type 9 (Null Material)

Default heading: Material Type #9 (Null Material)

PC pc	Pressure cutoff.
MU μ	μ.

The null material must be used with an equation-of-state. Pressure cutoff is negative in tension. A viscous stress of the form

$$\sigma_{ij} = \mu \dot{\varepsilon}^{V}_{ij}$$

is computed for nonzero m where $\dot{\varepsilon}^{V}_{ij}$ is the deviatoric strain rate.

Material Type 10 (Isotropic-Elastic-Plastic-Hydrodynamic)

Default heading: Material Type #10 (Isotropic-Elastic-Plastic-Hydrodynamic)

G G	Shear modulus
SIGY σ_y	Yield strength
EH E_h	Plastic hardening modulus
PC p_c or $-\sigma_f$	Pressure cutoff = 0: cutoff of -´ is assumed
A1 a_1	Yield function constant
A2 a_2	Yield function constant
NPTS n	Number of points in yield stress-effective plastic strain curve or yield stress-pressure curve.
ES σ_{y1} σ_{y2} ... σ_{yn}	Yield stress

EPS ε_{p1} ε_{p2} ... ε_{pn}	Effective plastic strain
P p_1 p_2 ... p_n	Pressure
FS ε_f	Failure strain

If the yield stress-plastic strain curve is not defined and if $a_1 = a_2 = 0$, the bilinear stress-strain curve shown in Figure 20-2 is obtained with b =1. The yield strength is calculated as

$$\sigma_y = \sigma_0 + E_h \bar{\varepsilon}^p$$

where p is the pressure. The quantity E_h is the plastic hardening modulus defined in terms of Young's modulus, E, and the tangent modulus, E_t, as follows

$$E_h = \frac{E_t\ E}{E - E_t}$$

If the yield stress-plastic strain (pressure) curve is defined, a curve like that shown in Figure 20-4 may be defined. In this latter case, the yield stress and plastic hardening modulus, a_1 and a_2 are ignored. Effective stress is defined in terms of the deviatoric stress tensor, s_{ij}, as:

$$\bar{\sigma} = \left(\frac{3}{2} s_{ij}\ s_{ij} \right)^{1/2} \tag{1}$$

and effective plastic strain by:

$$\bar{\varepsilon}^p = \int_0^t \left(\frac{2}{3} D_{ij}^p D_{ij}^p \right)^{1/2} dt, \tag{2}$$

where t denotes time and D_{ij}^p is the plastic component of the rate of deformation tensor. Yield stress may be defined as a function of plastic strain or pressure but not both.

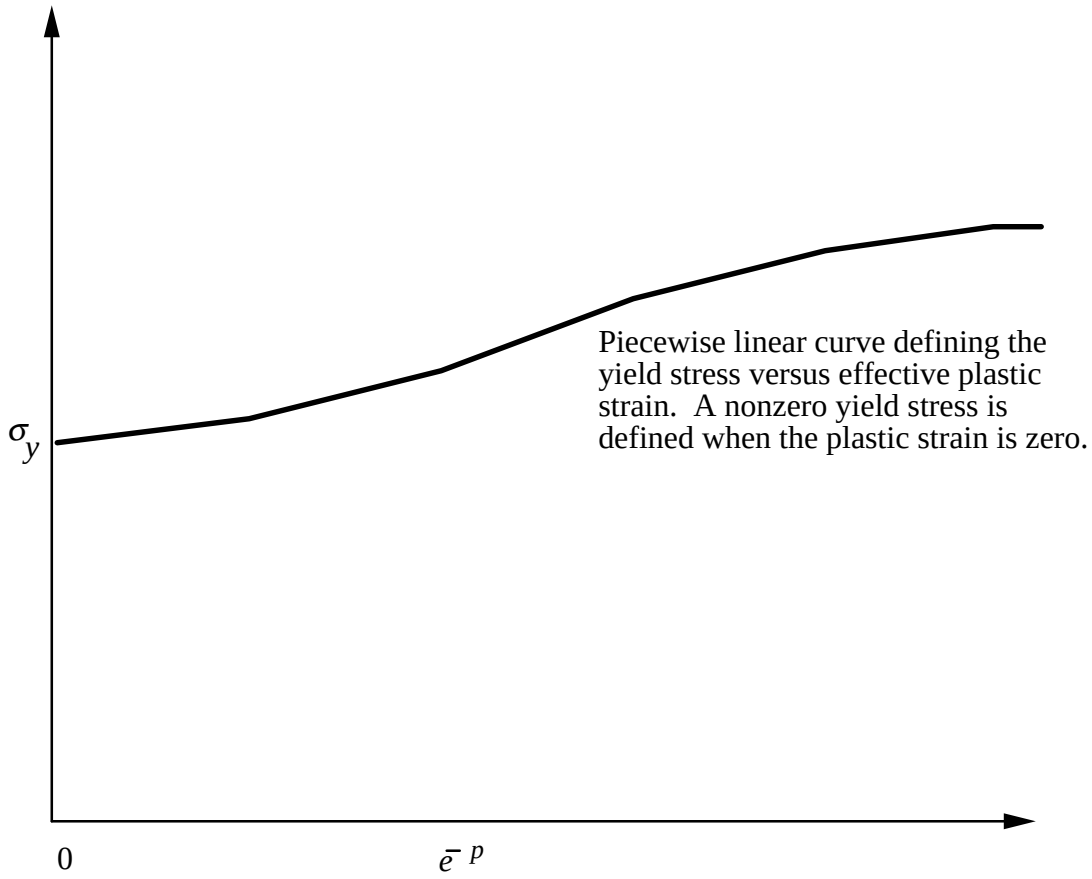


Figure 20-4. Effective stress versus effective plastic strain curve.

Material Type 11 (Temperature Dependent Elastic-Plastic Hydrodynamic)

Default heading: Material Type #11 (Temperature Dependent Elastic-Plastic Hydrodynamic)

G G_0	Shear modulus.
SIGO σ_o	See equations below.
BETA β	
N n	
GAMA γ_i	
SIGM σ_m	
B b	
BP b'	
H h	
F f	
A a	
TO T_{mo}	
GAMO g_0	
SA a	

PC pmin or -s_f
ECO EC_0 Cold compression energy coefficients (optional)
EC1 EC_1
EC2 EC_2
EC3 EC_3
EC4 EC_4
EC5 EC_5
EC6 EC_6
EC7 EC_7
EC8 EC_8
EC9 EC_9

If cold compression energy coefficients are not input, then LS-DYNA3D will calculate them based on the equation-of-state.

SPALL type Spall type
=0: default set to "2.0"
=1: p 3 $p_{\min}$
=2: if $s_{max} \geq s_f$ element spalls and tension $p < 0$, is never allowed; s_{max} = maximum principal stress.
=3: if $p < p_{\min}$ element spalls and tension $p<0$, is never allowed.
=4: failure strain

Users who have an interest in this mode are encouraged to study the paper by Steinberg and Guinan [9] which provides the theoretical basis. Another useful reference is the KOVEC user's manual [10].

In terms of the foregoing input parameters, we define the shear modulus, G, before the material melts as:

$$G = G_0\left[1 + bpV^{1/3} - h\left(\frac{E_i - E_c}{3R'} - 300\right)\right]e^{-fE_i/E_m - E_i}$$

where p is the pressure, V is the relative volume, E_c is the cold compression energy:

$$\bar{\varepsilon}^{\,p} = \int_0^t \left(\frac{2}{3}\,D_{ij}^{p}D_{ij}^{p}\right)^{1/2} dt,$$

$$x = 1 - V$$

and E_m is the melting energy:

$$E_m(x) = E_c(x) + 3R'T_m(x)$$

which is in terms of the melting temperature $T_m(x)$:

$$T_m(x) = \frac{T_{mo}\exp(2ax)}{V^{2(\gamma_o - a - \frac{1}{3})}}$$

and the melting temperature at $r = r_0$, T_{mo}.

In the above equation, R' is defined by

$$R' = \frac{R\rho}{A}$$

where R is the gas constant and A is the atomic weight. If R' is not defined, LS-DYNA3D computes it with R in the cm-gram-microsecond system of units.

The yield strength σ_y is given by:

$$\sigma_y = \sigma_0' \left[1 + b'pV^{1/3} - h\left(\frac{E_i - E_c}{3R'} - 300\right)\right] e^{-fE_i/(E_m - E_i)}$$

if E_m exceeds E_i. Here, σ_0' is given by:

$$\sigma_y = \sigma_0' \left|1 + \beta(\gamma_i + \varepsilon^{-p})^{n}\right.$$

where ε_i is the initial plastic strain. Whenever σ_0' exceeds σ_m, σ_0' is set equal to σ_m. After the material melts, σ_y and G are set to zero.

If the coefficients ECO,...EC9 are not defined above, LS-DYNA3D will fit the cold compression energy to the ten term polynomial expansion:

$$E_c = \sum_{i=0}^{9} EC_i \eta^i$$

where EC_i is the ith coefficient and $h=r/r_0-1$. A least square method is used to perform the fit.

Material Type 12 (Isotropic-Elastic-Plastic)

Default heading: Material Type #12 (Isotropic-Elastic-Plastic)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.

PR ν	Poisson's ratio.

Additional Options:

SIGY σ_y	Yield strength.
EH E_h	Hardening modulus.

Pressure is integrated in time

$$\dot{p} = -K\frac{\dot{V}}{V}$$

where V is the relative volume. This model is recommended for brick elements but not for shell elements since it is not too accurate.

Material Type 13 (Elastic-Plastic with Failure Model)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

SIGY σ_y	Yield strength.
EH E_h	Hardening modulus.
FS ε_f	Failure strain.
FP p_f	Failure pressure (£0.0).

When the effective plastic strain reaches the failure strain or when the pressure reaches the failure pressure, the material loses its ability to carry tension and the deviatoric stresses are set to zero, i.e., the material behaves like a fluid.

Material Type 14 (Soil and Crushable Foam with Failure Model)

The input for this model is the same as for material type 5; however, when the pressure reaches the failure pressure, the element loses its ability to carry tension.

Material Type 15 (Johnson/Cook Plasticity Model)

G *G*	Shear modulus.
A *A*	See equation (1).
B *B*	See equation (1).
N *n*	See equation (1).
R *r*	See equation (1).
M *m*	See equation (1).
TM T_{melt}	Melt temperature
TO *To*	Room temperature.
EPSO *Eo*	Effective plastic strain rate.
HCP *c*	Specific heat.
PC *pc*	Pressure cutoff (pc < 0.0).
D1 d_1	See equation (2).
D2 d_2	See equation (2).
D3 d_3	See equation (2).
D4 d_4	See equation (2).
D5 d_5	See equation (2).
IT *i*	Iteration options: =0: no iterations. =1: LS-DYNA3D iterates to determine a more accurate point on the stress-strain curve.

The Johnson/Cook model is described in reference [11]. This model includes strain rate hardening, thermal softening, and has a complex damage model. The equations describing the flow stress vs. effective plastic strain and failure strain are as follows:

$$\sigma_y = \left(A + B \bar{\varepsilon}^{p^n} \right) \left(1 + c \ln \dot{\varepsilon}^*\right) \left(1 - T^{*^m}\right)$$

where *A, B, C, n,* and *m* are input constants,

$\bar{\varepsilon}^p$ effective plastic strain

$\dot{\varepsilon}^* = \frac{\dot{\bar{\varepsilon}}^p}{\dot{\varepsilon}_0}$ effective plastic strain rate for $\dot{\varepsilon}_0 = 1\ s^{-1}$

$T^*=T_r/T_m$=homologous temperature

Constants for a variety of materials are also provided in [11].

Due to the nonlinearity in the dependence of flow stress on plastic strain, an accurate value of the flow stress requires iteration for the increment in plastic strain. However, by using a Taylor series expansion with linearization about the current time, we can solve for s_y with sufficient accuracy to avoid iteration.

The strain at fracture is given by

$$\varepsilon^f = \left[D_1 + D_2 \exp D_3 \sigma^* \right] \left[1 + D_4 \ln \dot{\varepsilon}^*\right] \left[1 + D_5 T^*\right]$$

where s* is the ratio of pressure divided by effective stress:

$$\sigma^* = \frac{p}{\sigma_{eff}}$$

Fracture occurs when the damage parameter

$$D = \sum \frac{\Delta \bar{\varepsilon}^p}{\varepsilon^f}$$

reaches the value of 1.

Material Type 16 (Pseudo Tensor Geological Model)

Default heading: Material Type #16 (Pseudo Tensor Geological Model)

G G	Shear modulus (constant Shear modulus model).
PR ν	Poisson's ratio (constant Poisson's ratio model).
SIGF sigf	Tensile cutoff. (Maximum principal stress for failure.)
A0 a_0	Cohesion.
A1 a_1	Yield function constant.
A2 a_2	Yield function constant.
A0F a_{0f}	Cohesion for failed material.

A1F a_{1f}	Pressure hardening coefficient for failed material.
B1 b_1	Damage scaling factor.
PER p	Percent reinforcement.
ER E_r	Elastic modulus for reinforcement.
PR ν_r	Poisson's ratio for reinforcement.
SIGY σ_y	Initial yield strength.
ETAN E_t	Tangent modulus.
LCP lc_1	Load curve giving rate sensitivity for principal material.
LCR lc_2	Load curve giving rate sensitivity for reinforcement.
NPTS n	Number of points in yield stress-effective plastic strain curve or yield stress-pressure curve; (n£16).
ES σ_1 σ_2 ... σ_n	Yield stress.
EPS ε_{p1} ε_{p2} ... ε_{p3}	Effective plastic strain.
P p_1 p_2 ... p_n	Pressure.

See the LS-DYNA3D manual for a description of this model.

Material Type 17 (Elastic Plastic with Failure Model)

Input any two of the following:

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options:*

SIGY σ_y	Yield strength.
EH E_h	Plastic hardening modulus.
FS ε_f	Failure stress.

Model 17 can fail in two ways. In hydrostatic tension, the element will fail when the failure stress is exceeded. The element will then allow hydrostatic compressive loads only.

If the effective stress exceeds the failure stress, the element will form a fracture plane and retain part of its strength.

Material Type 18 (Power Law Plasticity)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

K *k*	See equation below
M *m*	See equation below.
SC *c*	Strain rate parameter, *C*.
SP *p*	Strain rate parameter, *p*.

missing

Elastoplastic behavior with isotropic hardening is provided by this model. The yield stress, σ_y, is a function of plastic strain and obeys the equation:

$$\sigma_y = k(\varepsilon_e + \bar{\varepsilon}^p)^n$$

ε_e is the elastic strain to yield and where $\bar{\varepsilon}^p$ is the effective plastic strain. The strain-rate parameters are defined in material type 3.

Material Type 19 (Strain Rate Sensitive Plasticity)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

ECRV *lc*	Load curve describing Young's modulus as a function of strain rate.
ETAN e_{tan}	Tangent hardening modulus.
FCRV *lc*	Load curve describing failure stress as a function of strain rate.
SIGY *lc*	Load curve describing yield as a function of strain rate.
TCRV *lc*	Load curve describing tangent modulus as a function of strain rate.
TDEL D*t*	Minimum time step. (This is for element deletion).

In this model, a load curve is used to describe the yield strength, s_0, as a function of effective strain rate,

$$\dot{\bar{\varepsilon}} = \left(\frac{2}{3} \dot{\varepsilon}'_{ij} \dot{\varepsilon}'_{ij} \right)^{1/2}$$

and the prime denotes the deviatoric component. The yield stress is defined as

$$\sigma_y = \sigma_0 \left(\dot{\bar{\varepsilon}} \right) + E_h \bar{\varepsilon}^p$$

where $\bar{\varepsilon}^p$ is the effective plastic strain and E_h is given by

$$E_h = \frac{E E_t}{E - E_t} \quad .$$

Material Type 20 (Rigid Body)

All elements with the same material number become a single rigid body if the material is type 20 whether the elements are connected or not. Density and two independent material strength constants are required to establish penalties for contact surfaces and joints.

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

DEFG	The rigid body is defined in the global system used by CAL3D/MADYMO3D. (LS-920)

DEFL	The rigid body is defined in the local system used by CAL3D/MADYMO3D. (LS-920)
ELLIPSE *m*	The rigid body is slaved to MADYMO3D ellipsoid *m*. (LS-920)
MESH	Generate a mesh for the CAL3D/MADYMO3D coupled rigid body. (LS-920)
MSLAV *m*	The rigid body is slaved to CAL3D rigid body number *m*. (LS-920)
PLANE *m*	The rigid body is slaved to MADYMO3D plane *m*. (LS-920)
SYSTEM *n*	The rigid body is slaved to MADYMO3D system *n*. (LS-920)
VDA	The rigid body is characterized by a VDA surface geometry. (LS-920)
AVEC a_x a_y a_z	Define the vector **a** for the rigid body local system.
VVEC v_x v_y v_z	Define the vector **v** for the rigid body local system.

Material Type 21 (Thermal Orthotropic)

EA E_a	See constitutive matrix for material 2.
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA υ_{ca}	
PRCB υ_{cb}	
GAB G_{ab}	
GBC G_{bc}	
GCA G_{ca}	
AA α_a	α_a.
AB α_b	α_b.
AC α_c	α_c.
AOPT aopt	Material axes option (Figure 20-1). =0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1). =1.0: locally orthotropic with materials axes

	determined by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below. =3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 22 (Orthotropic Damage Model)

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA υ_{ca}	
PRCB υ_{cb}	
GAB G_{ab}	
GBC G_{bc}	
GCA G_{ca}	
K K_f	Bulk modulus of failed material.
SC S_c	Shear strength, *ab* plane.
XT x_t	Longitudinal tensile strength, *a*-axis.
YT y_t	Transverse tensile strength, *b*-axis.
YC y_c	Transverse compressive strength.
ALPH α	Non-linear shear stress parameter.

AOPT aopt

Ma

terial axes option (Figure 20-1).

=0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1).

=1.0: locally orthotropic with materials axes determined by a point in space and global location of element center.

=2.0: globally orthotropic with materials axes determined by vectors defined below.

=3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.

XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 23 (Thermal Orthotropic with Curves)

NPTS npts	Number of points. (1<NPTS<50).
EA $(E_a)_1$ ... $(E_a)_n$	
EB $(E_b)_1$ ... $(E_b)_n$	
EC $(E_c)_1$ ... $(E_c)_n$	
PRBA $(\nu_{ba})_1$ ... $(\nu_{ba})_n$	
PRCA $(\nu_{ca})_1$ ... $(\nu_{ca})_n$	
PRCB $(\nu_{cb})_1$ ... $(\nu_{cb})_n$	
AA $(\alpha_a)_1$ ... $(\alpha_a)_n$	
AB $(\alpha_b)_1$ ... $(\alpha_b)_n$	
AC $(\alpha_c)_1$ ... $(\alpha_c)_n$	

GAB $(G_{ab})_1$ ... $(G_{ab})_n$
GBC $(G_{bc})_1$ ... $(G_{bc})_n$
GCA $(G_{ca})_1$ ... $(G_{ca})_n$

AOPT aopt — Material axes option (Figure 20-1).

=0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1).

=1.0: locally orthotropic with materials axes determined by a point in space and global location of element center.

=2.0: globally orthotropic with materials axes determined by vectors defined below.

=3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.

XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 24 (Elastic Plastic with Failure)

Input any two of the following:

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options:*

SIGY s_y	Yield strength.
ETAN E_t	Hardening modulus
NPTS n	Number of points in effective stress-effective plastic strain curve. Note that the first point on this curve must be $e_1 = 0.0$ and s_1 = yield stress.
ES σ_1 σ_2 ... σ_n	Effective stress.
EPS ε_1 ε_2 ... ε_n	Effective plastic strain.
TDEL Δt	Minimum time step. (This is for automatic element deletion).
FAIL ε_f	Failure strain
CSR lc	Load curve which describes strain-rate effects.

Strain rate is accounted for using the Cowper and Symonds model which scales the yield stress with the factor

$$1+\left(\frac{\dot{\varepsilon}}{C}\right)^{1/p}$$

where $\dot{\varepsilon}$ is the strain rate. For complete generality a load curve may be input instead. This latter option is quite expensive.

A curve similar to that shown in Figure 3.4 is expected. A load curve may be used with an arbitrary number of points if eight is not sufficient. The cost is roughly the same for either approach.

Material Type 25 (Inviscid Two Invariant Geologic Cap Model)

G *G*	Shear Modulus.
K *K*	Bulk Modulus.
ALPHA α	α.
BETA β	β.
GAMMA γ	γ.
THETA θ	θ.
R *R*	*R*.
D *D*	*D*.
X0 X_0	X_0.
CC *C*	*C*.
T *T*	Tension cutoff.
NPLOT nplot	Save the following variable for plotting in TAURUS: =1: k =2: *X*

=3: e_v^p
=.4: J_1
=5: $(J_2)^{1/2}$
=6: $(J2|_L)^{1/2}$
=7: $(J2|_{tr})^{1/2}$
=8: MTYPE
=9: number of iterations

LTYPE ltype — Variable ltype.
=1: soil/concrete (cap contracts)
=2: rock (cap doesn't contract)

For details of this model, please refer to the LS-DYNA3D User's Manual.

Material Type 26 (Metallic Honeycomb)

Model 26 provides a method for modeling the crushing of an anisotropic material which eventually compresses to a solid, isotropic mass. This model is valid for brick elements only. For more details, see the LS-DYNA3D manual.

E *E* — Young's modulus for fully compressed state.
PR ν — Poisson's ratio for fully compressed state.
SIGY σ_y — Yield stress for fully compressed state.
VF V_f — Relative volume at which the material is fully compacted.
LCA *lca* — Load curve for sigma-*aa* versus either relative volume or volumetric strain.
LCB *lcb* — Load curve for sigma-*bb* versus either relative volume or volumetric strain.
LCC *lcc* — Load curve for sigma-*cc* versus either relative volume or volumetric strain.
LCS *lcs* — Load curve for shear stress versus either relative volume or volumetric strain.
EAAU E_{aau} — Elastic modulus E_{aau} in uncompressed configuration.
EBBU E_{bbu} — Elastic modulus E_{bbu} in uncompressed configuration.
ECCU E_{ccu} — Elastic modulus E_{ccu} in uncompressed configuration.
GABU G_{abu} — Elastic shear modulus G_{abu} in uncompressed configuration.
GBCU G_{bcu} — Elastic shear modulus G_{bcu} in uncompressed configuration.
GCAU G_{cau} — Elastic shear modulus G_{cau} in uncompressed configuration.
LCAB *lcab* — Load curve number for s_{ab} versus either relative

	volume or volumetric strain. (default: *lcab=lcs*)
LCBC *lcbc*	Load curve number for s_{bc} versus either relative volume or volumetric strain. (default: *lcbc=lcs*)
LCCA *lcca*	Load curve number for s_{ca} versus either relative volume or volumetric strain. (default: *lcca=lcs*)
AOPT aopt	Material axes option (Figure 20-1). =0.0: locally orthotropic with materials axes by determined element nodes n_1, n_2, and n_4, (see Figure 20-1). =1.0: locally orthotropic with materials axes determined by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.

Material Type 27 (Compressible Mooney-Rivlin Rubber)

This material model provides an alternative to the Blatz-Ko rubber model. The implementation is due to Maker [12].

A *A*	Constant A.
B *B*	Constant B.
PR ν	Poisson's ratio.

The strain energy density function is defined as:

$$W=A(\text{I}-3)+B(\text{II}-3)+C(\text{III}^{-2}-1)+D(\text{III}-1)^2$$

where

C=0.5A+B.

$$D = \frac{A(5\upsilon - 2) + B(11\upsilon - 5)}{2(1 - 2\upsilon)}$$

ν=Poisson's ratio.

2(A+B)=G=shear modulus of linear elasticity.

I, II, III are invariants of the right Cauchy-Green Tensor $\underset{\sim}{C}$.

Material Type 28 (Resultant Plasticity)

Default heading: Material Type #28 (Resultant Plasticity)

This model is available for the Belytschko-Schwer beam and the Belytschko-Tsay shell and is still under development. For beams the treatment is elastic-perfectly plastic, but for shell elements isotropic hardening is approximately modeled.

Input any two of the following:

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

SIGY σ_y	Yield strength.
ET E_t	Hardening modulus (shells only).

Material Type 29 (Force Limited Resultant Formulation)

This model is valid for the Belytschko beam element only. Experimentally obtained force-deflection curves may be used to model buckling and plastic behavior. See the LS-DYNA3D manual for more details.

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

R R_1 lc_1... R_n lc_n ;	Applied moments for force deflection curves. *n*£8.
LPS1 *lps*1	Load curve for plastic moment versus rotation at node 1 in *s*-direction.
SFS1 s*fs1*	Scale factor for plastic moment versus rotation curve at node 1 in *s*-direction.
LPS2 *lps*2	Load curve for plastic moment versus rotation at node 2 in *s*-direction.
SFS2 s*fs2*	Scale factor for plastic moment versus rotation curve at node 2 in *s*-direction.
LPT1 *lpt*1	Load curve for plastic moment versus rotation at node 1 in *t*-direction.
SFT1 s*ft1*	Scale factor for plastic moment versus rotation curve at node 1 in *t*-direction.
LPT2 *lpt*2	Load curve for plastic moment versus rotation at node 2 in *t*-direction.
SFT2 s*ft2*	Scale factor for plastic moment versus rotation curve at node 2 in *t*-direction.

Material Type 30 (Closed-Form Update Shell Plasticity)

Default heading: Material Type #30 (Closed-Form Update Shell Plasticity)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

SIGY σ_y	Yield strength.
ET E_t	Hardening modulus.

This model is available for the Belytschko-Schwer beam and the Belytschko-Tsay shell and is still under development. For beams, the treatment is elastic-perfectly plastic, but for shell elements, isotropic harening is approximately modeled.

Material Type 31 (Frazer-Nash Rubber Model)

This model implements a hyperelastic constitutive law described in [13].

C001 C001	
C010 C010	
C020 C020	
C100 C100	
C101 C101	
C110 C110	
C200 C200	
C210 C210	
C300 C300	
C400 C400	
LIMIT l	Limit option =0.0: stop if strain limits are exceeded. 10.0: continue if strain limits are exceeded.
EMAX ε_{max}	Maximum strain limit.
EMIN ε_{min}	Minimum strain limit.

The strain energy function, *U*, is defined in terms of the input constants as:

$$U = C_{100} I_1 + C_{200} I_1^2 + C_{300} I_1^3 + C_{400} I_1^4 + C_{010} I_2 +$$

$$C_{020} I_2^2 + C_{110} I_1 I_2 + C_{210} I_1^2 I_2 + C_{001} I_3 + C_{101} I_1 I_3$$

The derivative of *U* with respect to a component of strain gives the corresponding component of stress:

$$S_{ij} = \frac{\partial U}{\partial E_{ij}} = 2\frac{\partial U}{\partial C_{ij}}$$

where S_{ij}, E_{ij}, and C_{ij} are the second Piola-Kirchhoff stress tensor, the Green-St. Venant strain tensor, and the right Cauchy-Green deformation tensor, respectively.

Material Type 32 (Laminated Glass Model)

EG E_g	Young's modulus for glass.
PRG ν_g	Poisson's ratio for glass.
SYG $(\sigma_y)_g$	Yield stress for glass.
ETG $(E_t)_g$	Hardening modulus for glass.
FSG $(\varepsilon_f)_g$	Failure strain.
EP E_p	Young's modulus for polymer.
PRP ν_p	Poisson's ratio for polymer.
SYP $(\sigma_y)_p$	Yield stress for polymer.
ETP $(E_t)_p$	Hardening modulus for polymer.
IOPT f_1 ... f_n ;	Integration point options. f_i=0: glass. f_i=1: polymer.

Isotropic hardening is assumed. The material to which the glass is bonded is assumed to stretch plastically without failure. A user defined integration rule is required which specifies the thickness of the layers making up the glass. There must be the same number of parameters for the IOPT command as integration points.

Material Type 34 (Fabric)

The fabric material is similar to the orthotropic composite model (22). It is designed to allow a fabric to be modeled as layers of orthotropic material. The principal characteristic of a fabric material is that it does not support compressive stresses. This is because it is usually modeled with elements that are at least an order-of-magnitude wider than the thickness of the material. This model is still somewhat experimental and model 22 is frequently substituted.

EA E_a	See constitutive matrix below.
EB E_b	

EC E_c
PRBA υ_{ba}
PRCA υ_{ca}
PRCB υ_{cb}
GAB G_{ab}
GBC G_{bc}
GCA G_{ca}

CSF *csf*	Compressive modulus scale factor.
TSF *tsf*	Tensile modulus scale factor.
EXP exp	Exponent.
CSEF *f*	Compressive stress elimination flag. =0: use the variable modulus method. =1: truncate stresses (recommended).

The material law that relates stresses to strains is defined as:

$$\underset{\sim}{C} = \underset{\sim}{T}^{T}\ \underset{\sim}{C}_{L}\underset{\sim}{T},$$

where \o(T, ~) is a transformation matrix, and \o(C, ~)L is the constitutive matrix defined in terms of the material constants of the orthogonal material axes, a, b, and c. The inverse of \o(C, ~)L is defined as

$$\underset{\sim L}{C^{-1}} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ba}}{E_b} & -\frac{\nu_{ca}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ab}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{cb}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ac}}{E_a} & -\frac{\nu_{bc}}{E_b} & \frac{1}{E_c} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{bc}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{ca}} \end{bmatrix}$$

Note that $\frac{\nu_{ab}}{E_a} = \frac{\nu_{ba}}{E_b}, \frac{\nu_{ca}}{E_c} = \frac{\nu_{ac}}{E_a}, \frac{\nu_{cb}}{E_c} = \frac{\nu_{bc}}{E_b}$.

AOPT aopt	Material axes option (Figure 20-1). =0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1). =1.0: locally orthotropic with materials axes determined by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below. =3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 35 (Kinematic/Isotropic Elastic-Plastic Green-Naghdi Rate)

Default heading: Material Type #35 (Green-Naghdi Rate Plasticity)

Input any two of the following:

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

SIGY σ_y	Yield strength
ET E_t	Hardening modulus

BETA β'	Hardening parameter, $0 \leq \beta' \leq 1$
SC *c*	Strain rate parameter, *C*
SP *p*	Strain rate parameter, *p*

This model is available only for brick elements and is similar to model 3 but uses the Green-Naghdi Rate formulation rather than the Jaumann rate.

Material Type #37 (Transversely Anisotropic Elastic-Plastic)

Default heading: Material Type #37 (Transversely Anisotropic Elastic-Plastic)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

ET E_t	Hardening modulus
LCSS *lc*	Load curve number for stress-strain curve.
R *R*	Anisotropic hardening parameter, *R*.
SIGY σ_y	Yield strength

This model is only available for shell elements and is intended for modeling sheet metal forming processes. This is a degenerate form of Hill's model which assumes similar in-plane flow characteristics in all directions but different through-thickness effects. See the LS-DYNA3D manual for more details.

Material Type 41-50 (User Defined Material Models)

NPTS npts	Number of material parameters.
PARAM parameter 1 ... parameter	Material parameters.
AOPT aopt	Material axes option (Figure 20-1).

	=0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1).
	=1.0: locally orthotropic with materials axes determined by a point in space and global location of element center.
	=2.0: globally orthotropic with materials axes determined by vectors defined below.
	=3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 51 (Temperature and Rate Dependent Plasticity)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional Options:

T *T*	Initial Temperature.
HC *HC*	Heat generation coefficient.
COEF C_1 ... C_{18}	Model Coefficients.
ALPHA α_1 α_2 α_4 α_5 α_6	Initial value of internal state variables.

KAPPA κ	κ.

See the LS-DYNA3D manual for a description of this model.

Material Type 52 (Sandia's Damage Model)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

T *T*	Initial Temperature.
HC *HC*	Heat generation coefficient.
COEF C_1 ... C_{18}	Model Coefficients.
ALPHA α_1 α_2 α_4 α_5 α_6	Initial value of internal state variables.
NEXP *n*	Exponent in damage evolution
D0 D0	Initial damage (porosity).

See the LS-DYNA3D manual for a description of this model.

Material Type 53 (Low Density Closed Cell Polyurethane Foam)

Options:

E *E*	Young's modulus.
GAM0 γ_0	Initial volumetric strain.
P0 p_0	Initial foam pressure.
PA a	a.
PB b	b.
PC *c*	*c*.
PHI ϕ	Ratio of foam to polymer density.

See the LS-DYNA3D manual for a description of this model.

Material Type 54 and 55 (Composite Damage Model)

Material 54 uses the Chang matrix failure criterion (the same as model 22). Material 55 uses the Tsay-Wu criterion. These models are for LS-920 and later.

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA υ_{ca}	
PRCB υ_{cb}	
GAB G_{ab}	
GBC G_{bc}	
GCA G_{ca}	
FBRT fbrt	Softening for fiber tensile strength =0.0: fiber rupture with tension cutoff. >0.0: stress=fbrt, X_c after failure.
SOFT soft	Softening reduction factor for material strength in crashfront elements (default=1.0)
K K_f	Bulk modulus of failed material.
SC S_c	Shear strength, *ab* plane.
XT x_t	Longitudinal tensile strength, *a*-axis.
YT y_t	Transverse tensile strength, *b*-axis.
YC y_c	Transverse compressive strength.
ALPH α	Non-linear shear stress parameter.
TFAIL t_f	Time step for automatic element deletion.

AOPT aopt	Material axes option (Figure 20-1). =0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1). =1.0: locally orthotropic with materials axes determined by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below. =3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.

XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 57 (Low Density Urethane Foam)

This model is for LS-920 and later.

Options:

BETA β	Decay constant.
E *E*	Young's modulus.
LC *l*	Load curve number of nominal stress versus strain.
TENSION *t*	Tension cut-off stress.
UNLOAD *d*	Hysteretic unloading factor between 0 and 1 (Default=1, i.e. no energy dissipation).

See the LS-DYNA3D manual for a description of this model.

Material Type 59 (Composite Failure Model - Plasticity Based)

This model is for LS-920 and later.

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA υ_{ca}	
PRCB υ_{cb}	
GAB G_{ab}	

GBC G_{bc}	
GCA G_{ca}	
FBRT fbrt	Softening for fiber tensile strength =0.0: fiber rupture with tension cutoff. >0.0: stress=fbrt, X_c after failure.
SOFT soft	Softening reduction factor for material strength in crashfront elements (default=1.0)
SF *sf*	Softening factor. (default=0.0).
SR *sr*	Reduction factor. (default=0.447).
K K_f	Bulk modulus of failed material.
SC S_c	Shear strength, *ab* plane.
XT x_t	Longitudinal tensile strength, *a*-axis.
YT y_t	Transverse tensile strength, *b*-axis.
YC y_c	Transverse compressive strength.
ALPH α	Non-linear shear stress parameter.
TFAIL t_f	Time step for automatic element deletion.

AOPT aopt	Material axes option (Figure 20-1). =0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 20-1). =1.0: locally orthotropic with materials axes determined by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below. =3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

Material Type 60 (Elastic with Viscosity)

This model is for LS-910 and later.

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

NPTS npts	Number of points. (npts£8). (Default=1).
T T_1 ... T_n	Temperatures. (input only if npts>1.)
VC v_1 ... v_n	Viscosity coefficients (at least one is input.)

Material Type 64 (Simple Creep Model)

This model is for LS-930 and later.

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

EI *ei*	Value for *ei*.
K *k*	Value for *k*.
LCK *lck*	Load curve for *k*.
LCM *lcm*	Load curve for *m*.
M *m*	Value for *m*.
N *n*	Value for *n*.

Material Type Belt

This is a special material which applies to beam elements only. When material type belt is specified, beams are converted to the special seat belt element in LS-920 and later. Dummy beam elements are output to LS-DYNA3D also for viewing as null materials.

LCL *lcl*	Load curve for loading.
LCU *lcu*	Load curve for unloading.
RO r	Mass per unit length.
MINIMUM *l*	Minimum allowable length. (This is used to determine the minimum element size before an element is passed through a slip ring.)

Example:

```
MAT 56 TYPE BELT LCL 24 LCU 24 RO [0.100/386.4]
        MINIMUM 0.2 BEAM ENDMAT
```

21. Equations-of-State

Equations-of-state are required by certain LS-DYNA2D and LS-DYNA3D material models. They provide a relationship between pressure, relative volume, and temperature (or internal energy) which is used in place of a bulk modulus. Equations-of-state are needed when significant volume changes occur during a deformation process. They are attached to a material model and the general form of the input is:

MAT *i* TYPE *j* {material options} ENDMAT
EOS *k* {equation-of-state options} ENDEOS

This will define material *i* as being of type *j* and having equation-of-state characteristics of type *k*.

Equation-of-State Form 1 (Linear Polynomial)

Default heading: Equation-of-State Form 1 (Linear Polynomial)

CO C_0	See equation below
C1 C_1	
C2 C_2	
C3 C_3	
C4 C_4	
C5 C_5	
C6 C_6	
E0 E_0	Initial internal energy
V0 V_0	Initial relative volume
ENDEOS	End equation-of-state definition.

The linear polynomial equation-of-state is linear in internal energy. The pressure is given by:

$$P = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2)E .$$

where terms $C_2\mu^2$ and $C_6\mu^2$ are set to zero if $\mu<0$, $\mu=\rho/\rho_0-1$ and ρ/ρ_0, is the ratio of current density to the initial density.

Equation-of-State Form 2 (JWL)

Default heading: Equation-of-State Form 2 (JWL High Explosive)

A A	See equation below.
B B	
R1 R_1	
R2 R_2	
OMEGA ω	
E0 E_0	Initial internal energy
V0 V_0	Initial relative volume
ENDEOS	End equation-of-state definition.

The JWL equation-of-state defines the pressure as

$$p = A\left(1 - \frac{\omega}{R_1 V}\right)e^{-R_1 V} + B\left(1 - \frac{\omega}{R_2 V}\right)e^{-R_2 V} + \frac{\omega E}{V},$$

and is usually used for detonation products of high explosives.

Equation-of-State Form 3 (Sack)

Default heading: Equation-of-State Form 3 (Sack Tuesday High Explosive)

A1 A_1	See equation below
A2 A_2	
A3 A_3	
B1 B_1	
B2 B_2	
E0 E_0	Initial internal energy
V0 V_0	Initial relative volume
ENDEOS	End equation-of-state definition.

The Sack equation-of-state defines the pressure as

$$p = \frac{A_3}{V^{A_1}} e^{-A_2 V}\left(1 - \frac{B_1}{V}\right) + \frac{B_2}{V} E$$

and is used for detonation products of high explosives.

Equation-of-State Form 4 (Gruneisen)

Default heading: Equation-of-State Form 4 (Gruneisen)

SP C	See equation below.
S1 S_1	
S2 S_2	
S3 S_3	
GAMMA g_0	
SA a	
E0 E_0	Initial internal energy.
V0 V_0	Initial relative volume.
ENDEOS	End equation-of-state definition.

The Gruneisen equation-of-state with cubic shock velocity-particle velocity defines pressure for compressed materials as

$$p = \frac{\rho_0 C^2 \mu \left[1 + \left(1 - \frac{\gamma_0}{2}\right)\mu - \frac{a}{2}\mu^2\right]}{\left[1 - (S_1 - 1)\mu - S_2 \frac{\mu^2}{\mu+1} S_3 \frac{\mu^3}{(\mu+1)}\right]^2} + (\gamma_0 + a\mu)E .$$

and for expanded materials as

$$p = \rho_0 C^2 \mu + (\gamma_0 + a\mu)E .$$

where C is the intercept of the u_s-u_p curve, S_1, S_2, and S_3 are the coefficients of the slope of the u_s-u_p curve, γ_0 is the Gruneisen gamma; and a is the first order volume correction to γ_0 and $\mu = \frac{\rho}{\rho_0} - 1$.

Equation-of-State Form 5 (Ratio of Polynomials)

Default heading: Equation-of-State Form 5 (Ratio of Polynomials)

A10 A_{10}
A11 A_{11}
A12 A_{12}
A13 A_{13}
A20 A_{20}
A21 A_{21}
A22 A_{22}
A23 A_{23}
A30 A_{30}
A31 A_{31}
A32 A_{32}
A33 A_{33}
A40 A_{40}
A41 A_{41}
A42 A_{42}
A43 A_{43}
A50 A_{50}
A51 A_{51}
A52 A_{52}
A53 A_{53}
A60 A_{60}
A61 A_{61}
A62 A_{62}
A63 A_{63}
A70 A_{70}
A71 A_{71}
A72 A_{72}
A73 A_{73}
ALPHA a
BETA b
A14 A_{14}
A24 A_{24}

COEF A10..A24	List the 32 above coefficients in the same order as they appear.
E0 E_0	Initial internal energy
V0 V_0	Initial relative volume
ENDEOS	End equation-of-state definition.

The ratio of polynomials equation-of-state defines the pressure as

$$p = \frac{F_1 + F_2 E + F_3 E^2 + F_4 E^3}{F_5 + F_6 E + F_7 E^2} (1 + \alpha \mu)$$

where

$$F_i = \sum_{j=0}^{n} A_{ij} \mu^j \qquad \text{n} = 4 \text{ if } i < 3$$

$$\mu = \frac{\rho}{\rho_0} - 1 \qquad \text{n} = 3 \text{ if } i \bullet 3$$

In expanded zones F_1 is replaced by $F'_1 = F_1 + \beta \mu^2$. By setting coefficient $A_{10} = 1.0$, the delta-phase pressure modeling for this material will be initiated. The code will reset it to 0.0 after setting flags.

Equation-of-State Form 6 (Linear Polynomial With Energy Leak)

Default heading: Equation-of-State Form 6 (Linear Polynomial with Energy Leak)

C1 C_1	See Equation-of-State Form 1.
C2 C_2	
C3 C_3	
C4 C_4	
C5 C_5	
C6 C_6	
E0 E_0	Initial internal energy.
V0 V_0	Initial relative volume.
CN *cn*	Curve number of time history that gives energy deposition rate.
ENDEOS	End equation-of-state definition.

Equation-of-State Form 7 (Ignition and Growth of Reaction in HE)

Default heading: Equation-of-State Form 7 (Ignition and Growth of Reaction in High Explosive)

AP A_p	See equations below
BP B_p	See equations below
R1P R_{1p}	See equations below
R2P R_{2p}	See equations below
G G	Second ignition coefficient
WPCP w_pC_p	See equations below
AE A_e	See equations below
BE B_e	See equations below
WECE w_eC_e	See equations below
R1E R_{1e}	See equations below
R2E R_{2e}	See equations below
FCRIT FCRIT	Critical fraction reached
I I	First ignition coefficient
H H	Growth coefficient
Z z	Pressure exponent
X x	See equations below
Y Y	See equations below
CP C_p	Heat capacity of reaction products
CE C_e	Heat capacity of unreacted HE
M m	(generally = 0)
T0 T_0	Initial temperature (^oK)
E0 E_0	Initial internal energy
ENDEOS	End equation-of-state definition.

A JWL equation-of-state defines the pressure in the unreacted HE as

$$P_e = A_e\left(1-\frac{\omega_e}{R1_eV_e}\right)e^{-R1_eV_e} + B_e\left(1-\frac{\omega_e}{R2_eV_e}\right)e^{-R2_eV_e} + \frac{\omega E_e}{V_e}$$

where V_e is the relative volume, E_e is the internal energy, and the constants A_e, B_e, w_e, R_{1e}, and R_{2e} are input constants. Similarly the pressure in the reaction products is defined by another JWL form

$$P_p = A_p\left(1-\frac{\omega_p}{R1_pV_p}\right)e^{-R1_pV_p} + B_e\left(1-\frac{\omega_p}{R2_pV_p}\right)e^{-R2_pV_p} + \frac{\omega E_p}{V_p}$$

The mixture of unreacted explosive and reaction products is defined by the fraction reacted F (F=0) ¨ no reaction, (F=1) ¨ complete conversion from explosive to products. The pressures and temperatures are assumed to be in equilibrium and the volumes are assumed to be additive.

$$V = (1\text{-}F)V_e + FV_p$$

The rate of reaction is

$$\frac{\partial F}{\partial t} = I\,(\text{FCRIT}-F)^y\left(V_e^{-1}-1\right)^3 + G\left(V_e^{-1}-1\right) + H\,(1-F)^y F^x P^z \left(V_p^{-1}-1\right)^m$$

where I, G, H, x, y, z, and m (generally m=0) are input constants.

The JWL equations of state and the reaction rates have been fitted to one- and two-dimensional shock initiation and detonation data for four explosives: PBX-9404, RX-03-BB, PETN, and cast TNT. The details of the calculational method are described by Cochran and Chan [14]. The detailed one-dimensional calculations and parameters for the four explosives are given by Lee and Tarver [15].

Equation-of-State Form 8 (Tabulated-Compaction)

Default heading: Equation-of-State Form 8(Tabulated-Compaction)

NPTS n	Number of points in tabulated curves.
LNV ε_{V_1} ε_{V_2} ε_{V_n}	Volumetric strain points, $\varepsilon_{V_i} = ln(V_i)$.
PC C_1 C_2...C_n	Points on the curve for $C(\varepsilon_V)$.
PT T_1 T_2...T_n	Points on the curve for $T(\varepsilon_V)$.
KU K_1 K_2...K_n	Points on the curve for the unloading bulk modulus.
GAMMA γ	See equation below.
E0 E_0	Initial internal energy.
V0 V_0	Initial relative volume.
ENDEOS	End equation-of-state definition.

The tabulated compaction model is linear in internal energy. Pressure is defined by

$$p = C(\varepsilon_V) + \gamma T(\varepsilon_V) E$$

in the loading phase. The volumetric strain ε_V, is given by the natural logarithm of the relative volume. Unloading occurs along the unloading bulk modulus to the pressure cutoff. Reloading always follows the unloading path to the point where unloading began, and continues on the loading path. See Figure 21-1. Up to 10 points and as few as 2 may be used when defining the tabulated function, LS-DYNA2D/3D will extrapolate to find the pressure

if necessary.

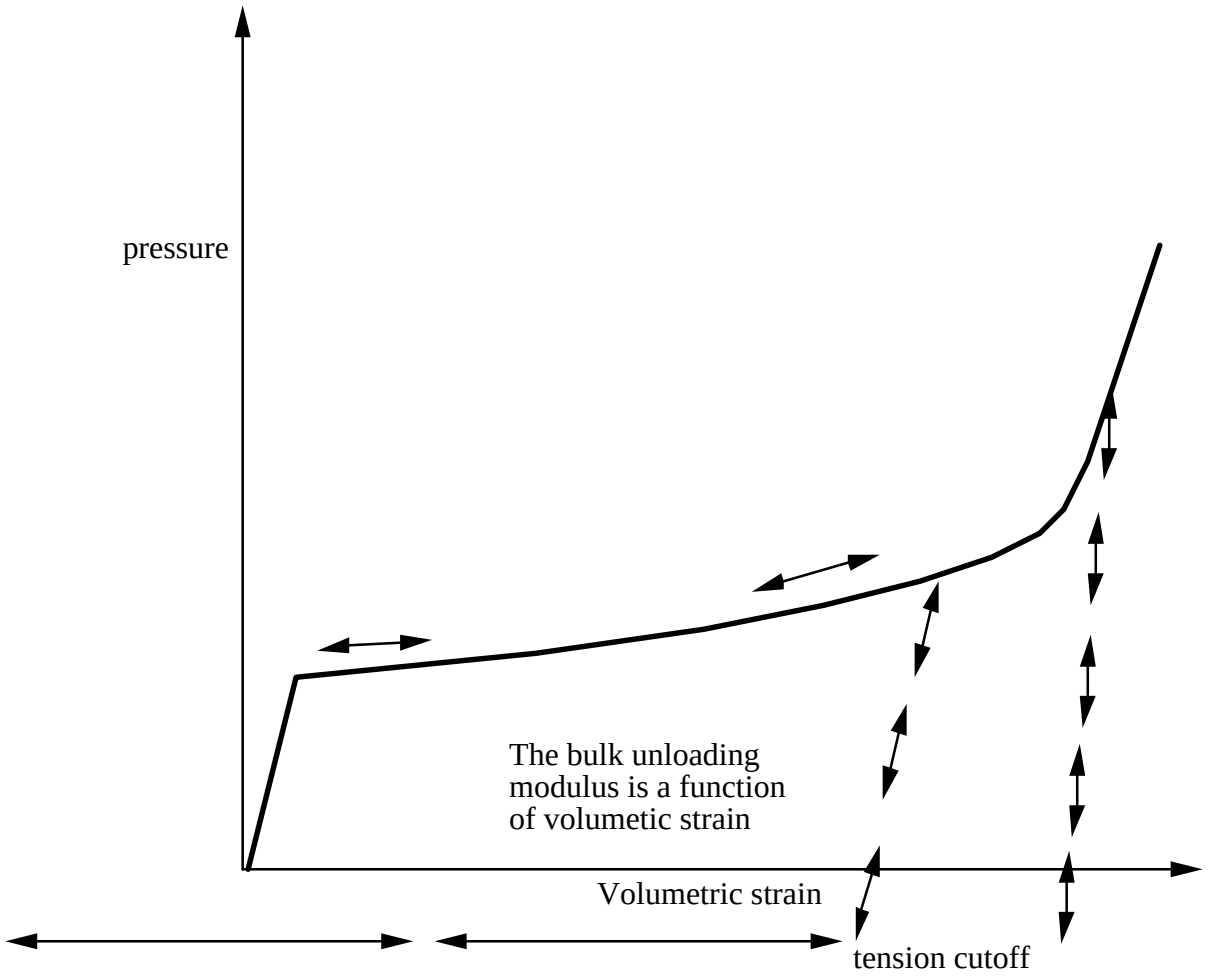


Figure 21-1. Pressure versues volumetric strain curve for equation-of-state form 8 with compaction. In the compacted states the bulk unloading modulus depend on the peak volumetric strain.

Equation-of-State Form 9 (Tabulated)

Default heading: Equation-of-State Form 9 (Tabulated)

NPTS *n*	Number of points in tabulated curves.
LNV e_{V_1} e_{V_2} e_{V_n}	Volumetric strain points, $e_{V_i} = ln(V_i)$.
PC C_1 C_2...C_n	Points on the curve for $C(e_V)$.
PT T_1 T_2...T_n	Points on the curve for $T(e_V)$.
GAMMA g	See equation below.

E0 E_0	Initial internal energy.
V0 V_0	Initial relative volume.
ENDEOS	End equation-of-state definition.

The tabulated compaction model is linear in internal energy. Pressure is defined by

$$P = C(\varepsilon_V) + \gamma T(\varepsilon_V)E$$

in the loading phase. The volumetric strain e_V, is given by the natural logarithm of the relative volume. Unloading occurs along the unloading bulk modulus to the pressure cutoff. Reloading always follows the unloading path to the point where unloading began, and continues on the loading path. See Figure 21-1. Up to 10 points and as few as 2 may be used when defining the tabulated function, LS-DYNA2D/3D will extrapolate to find the pressure if necessary.

22. LS-NIKE2D Commands and Materials

Analysis options are code dependent. They can be set either in the control section of the LS-INGRID input file or in the graphics phase. These commands become active when LS-NIKE2D output is selected with the NK2D command.

ANAL	*n*	Analysis type ="STAT": static analysis (default). ="DYN": direct time integration. ="DYNS": direct time integration with static initialization. ="EIGE": eigenvalue extraction.
BWMO	*n*	Bandwidth minimization option. ="ON": perform minimization in analysis code (default). ="OFF": don't minimize bandwidth.
DCTOL	tol	Convergence tolerance on displacements. LS-NIKE2D defaults to 0.001.
DELT	D*t*	Time step size for LS-NIKE2D.
DTMAX	*D*	Maximum step size permitted. If SSO = "AUTO" the default is set by LS-NIKE2D.
DTMN	*d*	Minimum step size permitted. If SSO = "AUTO" the default is set by LS-NIKE2D.
ECTOL	tol	Convergence tolerance on energy. LS-NIKE2D defaults to 0.01.
GEOM	*sn*	Node and element data dump interval for high speed printer. "PLAN" Plane strain "STRE" Plane stress "AXIS" Axisymmetric
GRAV	g_x g_y g_z	Gravity acceleration vector. The gravitational field is scaled in time by load curve one.
GSTIF	on/off	Geometric stiffness option. The default is off and generally gives the best results.
IPLT	*n*	Node and element data dump interval for TAURUS post-processing.
LST	tol	Line search tolerance.

MSRF	*n*	Maximum number of stiffness reformations per time step. LS-NIKE2D defaults to the recommended value of 15.
NBEI	*n*	The number of time steps between equilibrium iterations.
NBSR	*n*	The number of time steps between stiffness matrix reformation.
NEIG	*n*	Number of eigenvectors. This option turns on the subspace iteration eigenvalue/eigenvector solution method and overrides all other solution options. Eigenvectors are mass normalized and written into the graphics database. The time word corresponds to the frequency in radians/units of time.
NIBSR	*n*	Maximum number of equilibrium iterations permitted between stiffness matrix reformations. LS-NIKE2D defaults to the recommended value of 10.
NIP1	*s*	First Newmark integration parameter.
NIP2	*s*	Second Newmark integration parameter.
NSMD	*n*	Nonlinear solution method. ="BFGS": BFGS (default) ="BROY": Broyden's ="MODN": modified Newton To obtain a linear elastic solution, NBSR and NBEI should be larger than the number of time steps in the problem. The default parameters for nonlinear solution methods are near optimal. If a problem is having trouble converging the fixes include decreasing the time step, adding dynamic effects, or trying to eliminate some of the nonlinearities.
NSTEP	*n*	Number of desired time steps.
RFTS	*r*	Reduction factor for tangential stiffness. This is used for modeling the stick condition due to friction in the penalty formulation of contact.
SBRF	*n*	Number of time steps between restart file generations. If zero, LS-NIKE2D writes a restart file as it terminates.
SHIFT	w	Shift frequency in hertz. This option works with

		the eigenvalue/eigenvector solution method. Using this option, LS-NIKE2D will find the NEIG eigenvalues nearest to w. If the model has rigid body modes, a negative value for w should be used to make the run stable. If w is exactly the same value as an eigenvalue the system becomes singular.
SSIT	*s*	Slide surface insertion tolerance.
SSO	*u*	Step size option. "AUTO" "MANUAL"
SSOO	*n*	Optimal number of iterations per step.
TEO *i*		Thermal effects option =0: no thermal effects. =N: nodal temperatures are defined in input and are scaled according to a time function. N is the load curve number. =-1: each time step a new temperature state is read from a disk file. The time word at the beginning of each temperature state is ignored. =-2: each time step a temperature state is interpolated from the temperature state in a disk file. Therefore the time words at the beginning of each temperature state are used. =-3: the disk file containing temperatures has only one state. The initial state is assumed to be zero.
TERM	*t*	Terminate dynamic time integration at time *t*. The dynamic time step size will be computed if this command is used instead of the "DELT" command.

22.1 LS-NIKE2D MATERIAL INPUT

LS-NIKE2D material input is possible after the NK2D command has been input (see *Control Commands*). The form of this input is: MAT *n* TYPE *m* {options specific to material type m} {general material options} ENDMAT. *n* is a material name which is assigned a number in the order that they occur in the input. Therefore, the materials should be defined in order before any additional use of materials is made.

Material Type 1 (Elastic)

Default heading: Material Type #1 (Elastic)

Input any two of the following.

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR n	Poisson's ratio.

Material Type 2 (Orthotropic Elastic)

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA u_{ba}	
PRCA u_{ca}	
PRCB u_{cb}	
GAB G_{ab}	
AOPT aopt	Material axes option (Figure 22-1). =0.0: locally orthotropic with materials axes by y value specified on each element card and element nodes n_1 and n_2, (see Figure 22-1). =1.0: locally orthotropic with materials axes by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by y_G.
RP r_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
PSIG y_G	Define for AOPT = 2.

The material law that relates stresses to strains is defined as:

$$\underset{\sim}{C} = \underset{\sim}{T}^{T}\ \underset{\sim}{C}_{L}\underset{\sim}{T},$$

Where $\underset{\sim}{T}$ is a transformation matrix, and $\underset{\sim}{C}_{L}$ is the constitutive matrix defined in terms of the material constants of the orthogonal material axes, *a*, *b*, and *c*. The inverse of $\underset{\sim}{C}_{L}$ is defined as

$$\underset{\sim}{C}_{L}^{-1} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ba}}{E_b} & -\frac{\nu_{ca}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ab}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{cb}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ac}}{E_a} & -\frac{\nu_{bc}}{E_b} & \frac{1}{E_c} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{bc}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{ca}} \end{bmatrix}$$

Note that $\frac{\nu_{ab}}{E_a} = \frac{\nu_{ba}}{E_b}, \frac{\nu_{ca}}{E_c} = \frac{\nu_{ac}}{E_a}, \frac{\nu_{cb}}{E_c} = \frac{\nu_{bc}}{E_b}$.

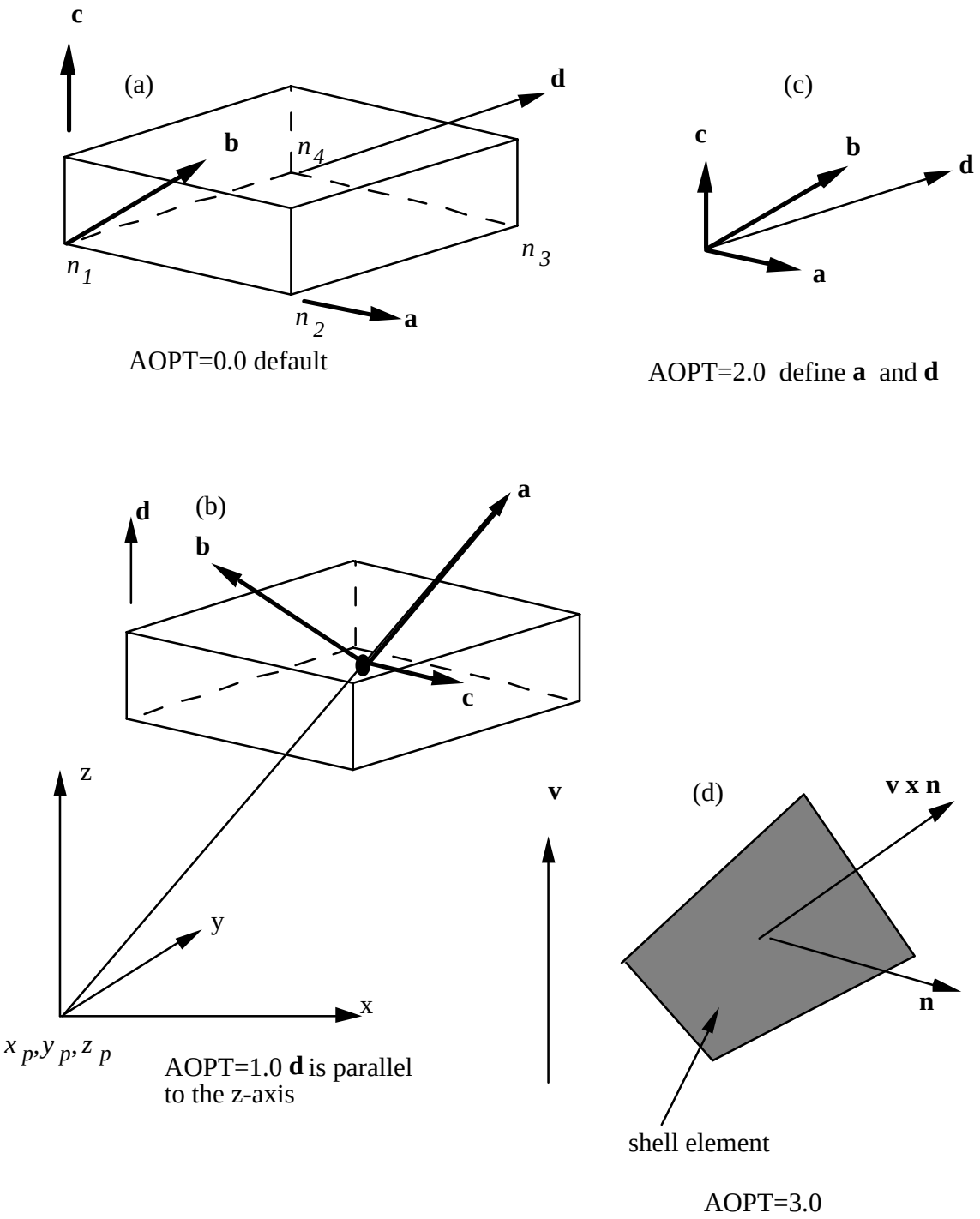


Figure 22-1. Options for determining principal materials axes: (a) AOPT = 0.0, (b) AOPT = 1.0, and (c) AOPT = 2.0.

Material Type 3 (Kinematic/Isotropic Elastic/Plastic)

Default heading: Material Type #3 (Elastic-Plastic)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR n	Poisson's ratio.

Additional *Options*

SIGY s_y	Yield strength.
ETAN E_t	Hardening modulus.
BETA b'	Hardening parameter, $0 \leq b' \leq 1$
NPTS *n*	Number of points on stress-effective plastic strain curve.
ES s_{y1} s_{y2} ... s_{yn}	Effective stress.
EPS e_{p1} e_{p2} ... e_{pn}	Effective plastic strain.

Isotropic, kinematic, or a combination of isotropic and kinematic hardening may be specified by varying b' between 0 and 1. For b' equal to 0 and 1, respectively, kinematic and isotropic hardening are obtained as shown in Figure 22-2. Effective stress is defined in terms of the deviatoric stress tensor, s_{ij} as:

$$\bar{\sigma} = \left(\frac{3}{2} S_{ij} S_{ij}\right)^{1/2}$$

where,

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$$

and effective plastic strain by:

$$\bar{\varepsilon}^{\,p} = \int_0^t d\bar{\varepsilon}^{\,p}$$

where t denotes time and

$$d\bar{\varepsilon}^{p} = \left(\frac{2}{3} d\varepsilon_{ij}^{p} d\varepsilon_{ij}^{p}\right)^{1/2}$$

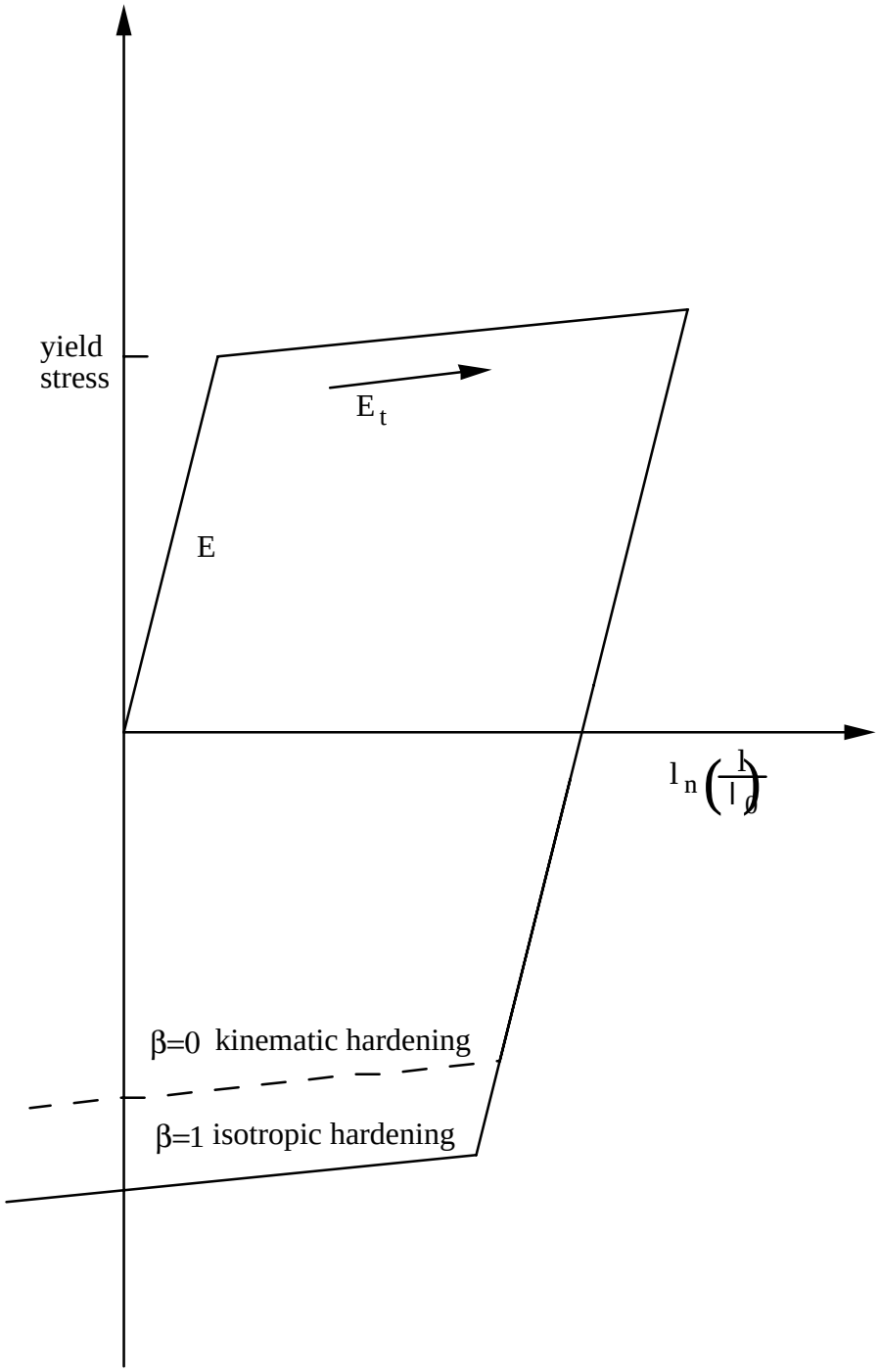


Figure 22-2. Elastic-plastic behavior with isotropic and kinematic hardening where l_0 and l are undeformed and deformed length of uniaxial tensions specimen.

Material Type 4 (Thermo-Elastic-Plastic)

Default heading: Material Type 4 (Thermo-Elastic-Plastic)

NPTS *n*	Number of temperature values for which material constants are defined.
TEMP T_1 T_2...T_n	Temperatures.
E E_1 E_2...E_n	Young's moduli.
PR u_1 u_2...u_n	Poisson's ratios.
ALPHA a_1 a_2...a_n	Coefficients of thermal expansion.
SIGY σ_{y1} σ_{y2}...σ_{yn}	Yield stresses.
ETAN E_{t1} E_{t2}...E_{tn}	Tangent moduli.

Material Type 5 (Soil and Crushable Foam)

Default heading: Material Type 5 (Soil and Crushable Foam)

Input any two of the following:

BULK *K*	Bulk modulus.
E *E*	Young's modulus.
G *G*	Shear modulus.
PR ν	Poisson's ratio.

Additional *Options*:

AO *n*	Yield function constant
A1 a_1	Yield function constant
A2 a_2	Yield function constant
PC P_c	Pressure cutoff for tensile fracture
UL uopt	Unloading option =0: volumetric crushing =1: no volumetric crushing
NPTS *n*	Number of points in volumetric strain versus pressure curve (*n*£10).
VS e_{v1} e_{v2} ... e_{vn}	Volumetric strain values
P p_1 p_2 ... p_n	Pressures corresponding to volumetric strain values

The deviatoric yield function, ϕ, is described in terms of the second invariant J_2.

$$J_2 = \frac{1}{2} s_{ij} \; s_{ij}$$

Pressure, p, and constants a_0, a_1, and a_2 as:

$$\phi = J_2 - \left[a_0 + a_1 p + a_2 p^2\right].$$

On the yield surface, $J_2=1/3(s_y)^{1/2}$, where σ_y is the yield stress, i.e.,

$$\sigma_y = \left[3\left(a_0 + a_1 p + a_2 p^2\right)\right]^{1/2}$$

For elastic-perfectly plastic behavior $a_1=a_2=0$, and $(3a_0)^{1/2}$ defines the yield strength. The volumetric strain is given by the natural logarithm of the relative volume V. If the pressure drops below the cutoff value, PC, then it is reset to that value.

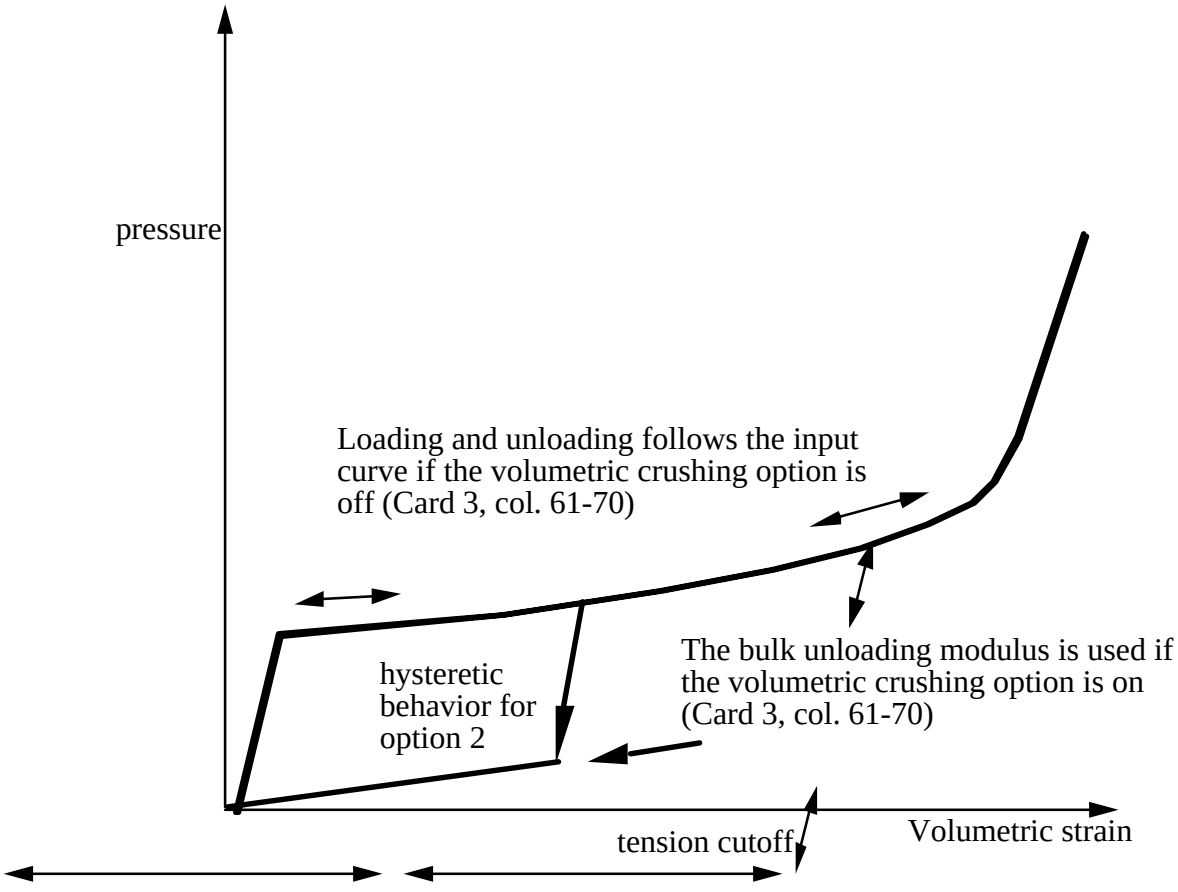


Figure 22-3. Volumetric strain versus pressure curve for soil and crushable foam model.

Material Type 6 (Viscoelastic)

G G_0	Short term shear modulus.
GI G_∞	Long term shear modulus.
K K	Bulk modulus.
BETA b	Decay constant.

The shear relaxation behavior is described by:

$$G(t) = G_\infty + (G_0 - G_\infty)\, e^{-\beta t}$$

A Jaumann rate formulation is used:

$$\overset{\nabla}{\sigma}'_{ij} = 2 \int_0^t G(t-\tau)\, D'_{ij}(\tau)\, dt$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{\sigma}_{ij}$, and the strain rate D_{IJ}.

Material Type 7 (Thermal Orthotropic Elastic)

Default heading: Material Type #7 (Thermal Orthotropic Elastic)

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA u_{ba}	
PRCA u_{ca}	
PRCB u_{cb}	
ALPA a_a	Thermal expansion coefficient along axis a.
ALPB a_b	Thermal expansion coefficient along axis b.
ALPC a_c	Thermal expansion coefficient along axis c.
GAB G_{ab}	
AOPT aopt	Material axes option (Figure 22-1). =0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 22-1). =1.0: locally orthotropic with materials axes determined by a point in space and global location of element center.

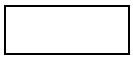
	=2.0: globally orthotropic with materials axes determined by y_G.
RP r_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
PSIG y_G	In radians, define for AOPT = 2.

Material Type 8 (Thermo-Elastic-Creep)

Default heading: Material Type #8 (Thermo-Elastic-Creep)

NPTS n	Number of temperature values for which material constants are defined.
TEMP T_1 T_2...T_n	Temperatures
G G_1 G_2...G_n	Shear moduli.
K K_1 K_2...K_n	Bulk moduli.
ALPHA a_1 a_2...a_n	Coefficients of thermal expansion.
A a_1 a_2...a_n	Creep parameters.
B b_1 b_2...b_n	Creep parameters.

In this model, G, is the shear modulus and the instantaneous creep is given by a power law of the form



where a and b are functions of temperature. This model was developed and provided for LS-NIKE2D by R. D. Krieg of Sandia National Laboratories.

Material Type 9 (Blatz-Ko Rubber)

Default heading: Material Type #9 (Rubber)

G m	Shear modulus

The second Piola-Kirchhoff stress is computed as

$$S_{ij} = \mu \left(\frac{1}{V} C_{ij} - V^{-1/1-2\upsilon} \delta_{ij} \right)$$

where V is the relative volume, C_{ij} is the right Cauchy-Green strain tensor, and n is the

Poisson's ratio which is set to .463 internally. This stress measure is transformed to the Cauchy stress, s_{ij}, according to the relationship

$$\sigma_{ij} = V^{-1} F_{ik} F_{jl} S_{lk}$$

where F_{ij} is the deformation gradient tensor.

Material Type 10 (Power Law Plasticity)

Input any two of the following:

BULK *K*	Bulk modulus
E *E*	Young's modulus
G *G*	Shear modulus
PR n	Poisson's ratio

Additional *Options:*

K *k*	See equation below
M *m*	See equation below
FC *f*	Failure criteria. =1: Mohr-Coulomb. =2: Drucker-Prager =3: check both
MPS s_{max}	Maximum pricipal stress (optional).
MSS t_{max}	Maximum shear stress (optional).
LC lc	Optional failure curve number.

The stress-strain curve for this model is based on the following equation:

$$\sigma_y = k(\varepsilon_e + \bar{\varepsilon}^p)^n$$

Material Type 12 (Power Law Thermo Plasticity)

NPTS n	Number of temperature points (£8).
T T_1 T_2...T_n	Temperatures
E E_1 E_2...E_n	Young's moduli
PR u_1 u_2...u_n	Poisson's ratios
K k_1...k_n	See equation below
M m_1...m_n	See equation below

The stress-strain curve for this model is based on the following equation:

$$\sigma_y = k(\varepsilon_e + \bar{\varepsilon}^p)^n$$

Material Type 22 (Frazer-Nash Rubber Model)

This model implements a hyperelastic constitutive law described in [13].

C001 C001
C010 C010
C020 C020
C100 C100
C101 C101
C110 C110
C200 C200
C210 C210
C300 C300
C400 C400

The strain energy function, U, is defined in terms of the input constants as:

$$U = C_{100} I_1 + C_{200} I_1^2 + C_{300} I_1^3 + C_{400} I_1^4 + C_{010} I_2 +$$

$$C_{020} I_2^2 + C_{110} I_1 I_2 + C_{210} I_1^2 I_2 + C_{001} I_3 + C_{101} I_1 I_3$$

The derivative of U with respect to a component of strain gives the corresponding component of stress:

$$S_{ij} = \frac{\partial U}{\partial E_{ij}} = 2\frac{\partial U}{\partial C_{ij}}$$

where, S_{ij}, E_{ij} and C_{ij} are the second Piola-Kirchhoff stress tensor, the Green-St. Venant strain tensor, and the right Cauchy-Green deformation tensor, respectively.

23. LS-NIKE3D Commands and Materials

Analysis options are code dependent. They can be set either in the control section of the LS-INGRID input file or in the graphics phase. These commands become active when LS-NIKE3D output is selected with the NK3D command.

ANAL	*n*	Analysis type ="STAT": static analysis (default) ="DYN": direct time integration ="DYNS": direct time integration with tatic initialization. ="EIGE": eigenvalue extraction.
BWMO	*n*	Bandwidth minimization option. ="ON": perform minimization in analysis code (default). ="OFF": don't minimize bandwidth.
DCTOL	tol	Convergence tolerance on displacements. LS-NIKE3D defaults to 0.001.
DELT	D*t*	Time step size for LS-NIKE3D.
DTMAX	*D*	Maximum step size permitted. If SSO = "AUTO" the default is set by LS-NIKE3D.
DTMN	*d*	Minimum step size permitted. If SSO = "AUTO" the default is set by LS-NIKE3D.
ECTOL	tol	Convergence tolerance on energy. LS-NIKE3D defaults to 0.01.
GRAV	$g_x\ g_y\ g_z$	Gravity acceleration vector. The gravitational field is scaled in time by load curve one.
GSTIF	on/off	Geometric stiffness option. The default is off and generally gives the best results.
IPLT	*n*	Node and element data dump interval for TAURUS post-processing.
LST	tol	Line search tolerance.
MSRF	*n*	Maximum number of stiffness reformations per time step. LS-NIKE3D defaults to the recommended value of 15.
NBEI	*n*	The number of time steps between equilibrium iterations.

NBSR	*n*	The number of time steps between stiffness matrix reformation.
NEIG	*n*	Number of eigenvectors. This option turns on the subspace iteration eigenvalue/eigenvector solution method and overrides all other solution options. Eigenvectors are mass normalized and written into the graphics database. The time word corresponds to the frequency in radians/units of time.
NIBSR	*n*	Maximum number of equilibrium iterations permitted between stiffness matrix reformation. LS-NIKE3D defaults to the recommended value of 10.
NIP1	*s*	First Newmark integration parameter.
NIP2	*s*	Second Newmark integration parameter.
NSMD	*n*	Nonlinear solution method. = "BFGS": BFGS (default) ="BROY": Broyden's ="MODN": modified Newton To obtain a linear elastic solution, NBSR and NBEI should be larger than the number of time steps in the problem. The default parameters for nonlinear solution methods are near optimal. If a problem is having trouble converging the fixes include decreasing the time step, adding dynamic effects, or trying to eliminate some of the nonlinearities.
NSTEP	*n*	Number of desired time steps.
RFTS	*r*	Reduction factor for tangential stiffness. This is used for modeling the stick condition due to friction in the penalty formulation of contact.
SBRF	*n*	Number of time steps between restart file generation. If zero, LS-NIKE3D writes a restart file as it terminates.
SHIFT	w	Shift frequency in hertz. This option works with the eigenvalue/eigenvector solution method. Using this option, NIKE will find the NEIG eigenvalues nearest to w. If the model has rigid body modes, a negative value for w should be used to make the run stable. If w is exactly the same value as an eigenvalue the system becomes singular.

SSIT	*s*	Slide surface insertion tolerance
SSO	*u*	Step size option. "AUTO" "MANUAL"
SSOO	*n*	Optimal number of iterations per step.
TEO	*i*	Thermal effects option = 0: no thermal effects. =*N*: nodal temperatures are defined in input and are scaled according to a time function. *N* is the load curve number. =-1: at each time step a new temperature state is read from a disk file. The time word at the beginning of each temperature state is ignored. =-2: at each time step a temperature state is interpolated from the temperature state in a disk file. Therefore the time words at the beginning of each temperature state is used. = -3: the disk file containing temperatures has only one state. The initial state is assumed to be zero.
TERM	*t*	Terminate dynamic time integration at time *t*. The dynamic time step size will be computed if this command is used instead of the "DELT" command.

23.1 LS-NIKE3D MATERIAL INPUT

LS-NIKE3D material input is possible after the NK3D command has been input (see *Control Commands*). The form of this input is: MAT *n* TYPE *m* {options specific to material type *m*} {general material options} ENDMAT. *n* is a material name which is assigned a number in the order that they occur in the input. Therefore, the materials should be defined in order before any additional use of materials is made.

Material Type 1 (Elastic)

Default heading: Material Type #1 (Elastic)

Input any two of the following.

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR n	Poisson's ratio.

Material Type 2 (Orthotropic Elastic)

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA u_{ba}	
PRCA u_{ca}	
PRCB u_{cb}	
GAB G_{ab}	
GBC G_{bc}	
GCA G_{ca}	
AOPT aopt	Material axes option (Figure 23-1). =0.0: locally orthotropic with materials axes by element nodes n_1, n_2, and n_4, (see Figure 23-1). =1.0: locally orthotropic with materials axes by a point in space and global location of element center. =2.0: globally orthotropic with materials axes determined by vectors defined below. =3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.

D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

The material law that relates stresses to strains is defined as:

$$\underset{\sim}{C} = \underset{\sim}{T}^{T}\ \underset{\sim}{C}_{L}\underset{\sim}{T},$$

Where \o(T,)$_{\sim}$ is a transformation matrix, and \o(C,)$_{L}$ is the constitutive matrix defined in terms of the material constants of the orthogonal material axes, *a*, *b*, and *c*. The inverse of \o(C,)$_{L}$ is defined as

$$\underset{\sim L}{C^{-1}} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ba}}{E_b} & -\frac{\nu_{ca}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ab}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{cb}}{E_c} & 0 & 0 & 0 \\ -\frac{\nu_{ac}}{E_a} & -\frac{\nu_{bc}}{E_b} & \frac{1}{E_c} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{bc}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{ca}} \end{bmatrix}$$

Note that $\frac{\nu_{ab}}{E_a} = \frac{\nu_{ba}}{E_b}, \frac{\nu_{ca}}{E_c} = \frac{\nu_{ac}}{E_a}, \frac{\nu_{cb}}{E_c} = \frac{\nu_{bc}}{E_b}$.

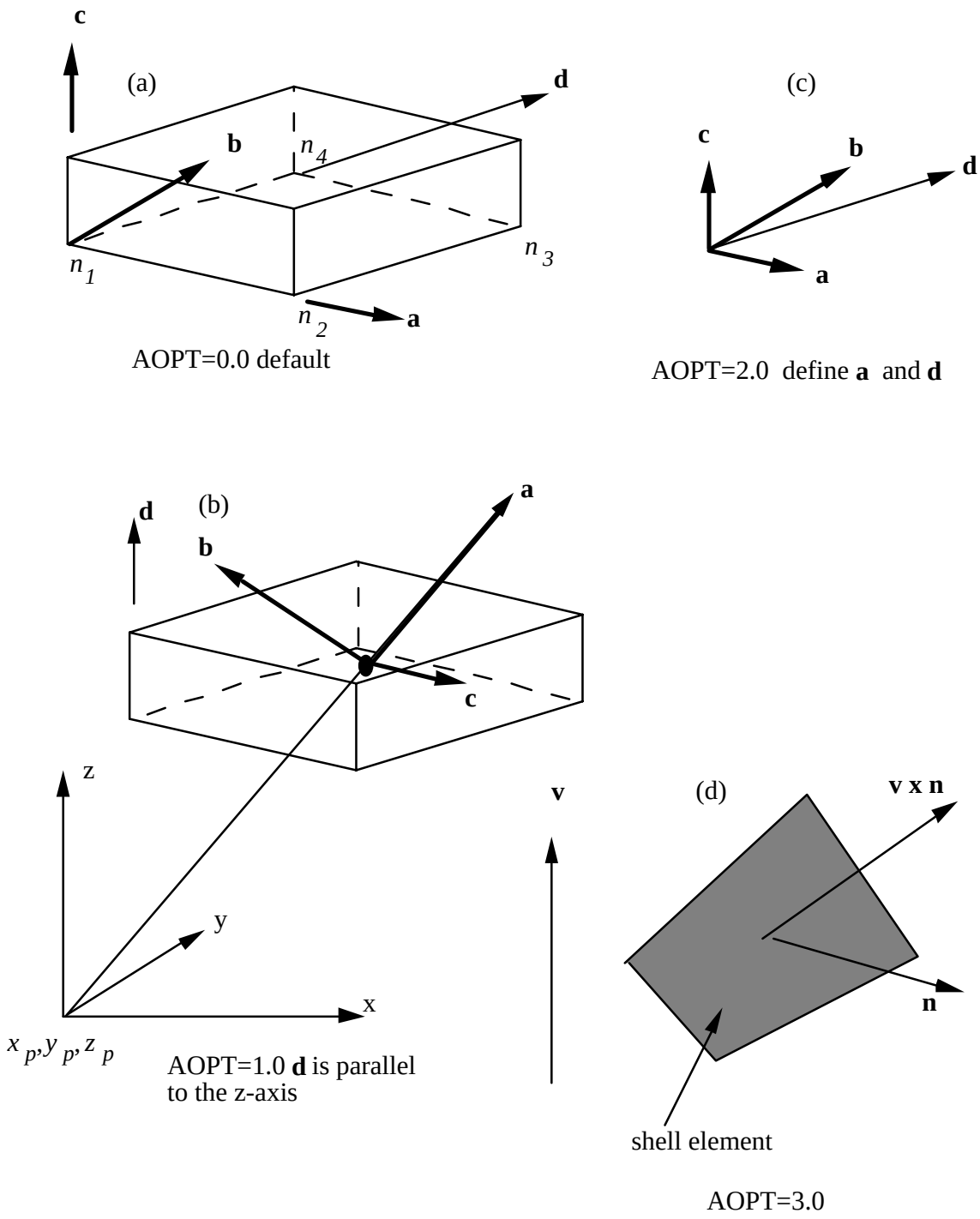


Figure 23-1. Options for determining principal materials axes: (a) AOPT = 0.0, (b) AOPT = 1.0, and (c) AOPT = 2.0.

Material Type 3 (Kinematic/Isotropic Elastic/Plastic)

Default heading: Material Type #3 (Elastic-Plastic)

Input any two of the following:

BULK *K*	Bulk modulus
E *E*	Young's modulus
G *G*	Shear modulus
PR n	Poisson's ratio

Additional *Options*:

SIGY s_y	Yield stress
ETAN E_t	Hardening modulus
BETA b'	Hardening parameter, $0 \leq b' \leq 1$
NPTS *n*	Number of points on stress-effective plastic strain curve.
ES s_{y1} s_{y2} ... s_{yn}	Effective stress.
EPS e_{p1} e_{p2} ... e_{pn}	Effective plastic strain.

Isotropic, kinematic, or a combination of isotropic and kinematic hardening may be specified by varying b' between 0 and 1. For b' equal to 0 and 1, respectively kinematic and isotropic hardening are obtained as shown in Figure 23-2. Effective stress is defined in terms of the deviatoric stress tensor, S_{ij} as:

$$\bar{\sigma} = \left(\frac{3}{2} S_{ij} S_{ij}\right)^{1/2}$$

where,

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$$

and effective plastic strain by:

$$\bar{\varepsilon}^{\,p} = \int_0^t d\bar{\varepsilon}^{\,p}$$

where *t* denotes time and

$$d\bar{\varepsilon}^{p} = \left(\frac{2}{3} d\varepsilon_{ij}^{p} d\varepsilon_{ij}^{p}\right)^{1/2}$$

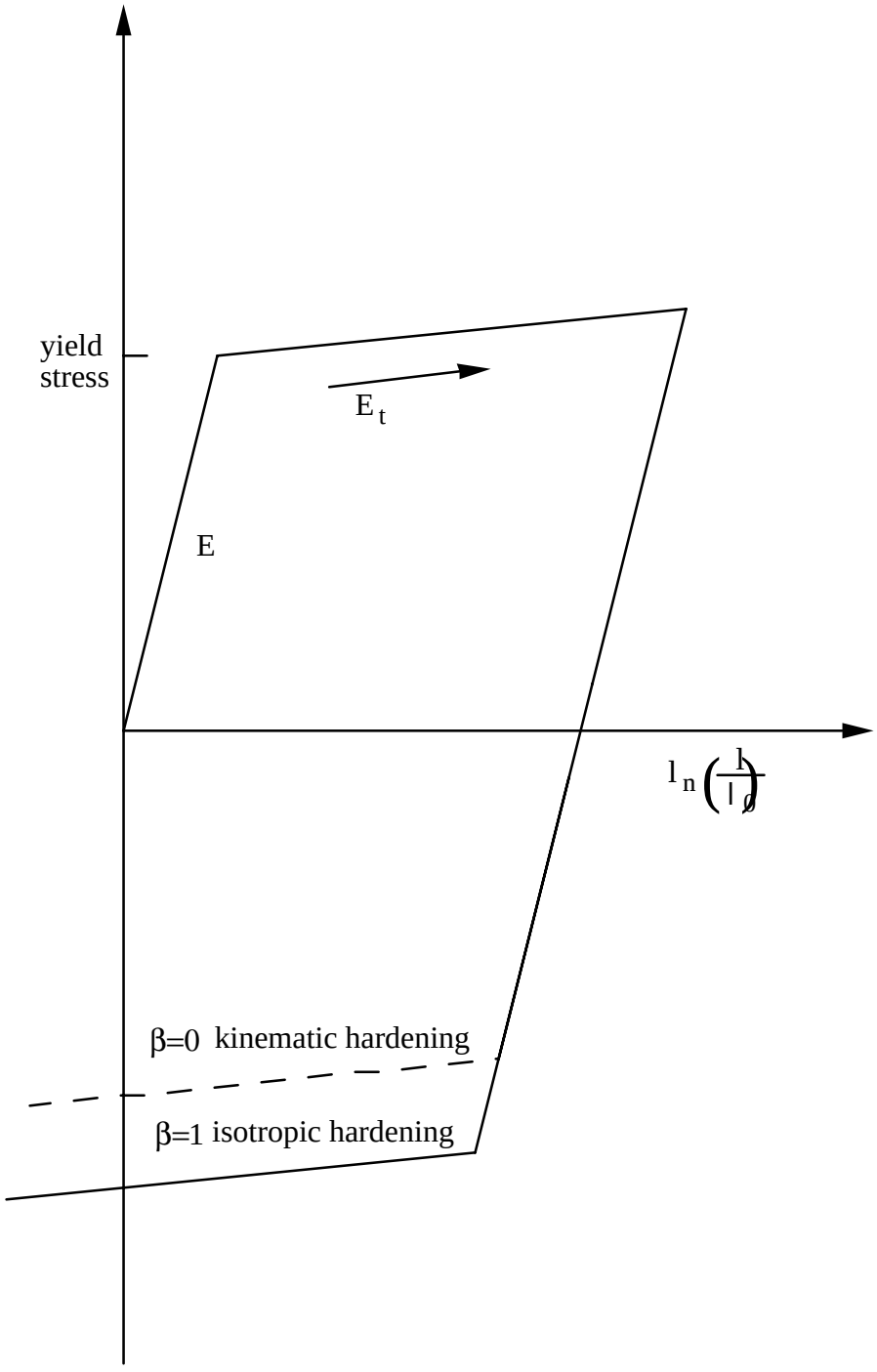


Figure 23-2. Elastic-plastic behavior with isotropic and kinematic hardening where l_0 and l are undeformed and deformed length of uniaxial tension specimen.

Material Type 4 (Thermo-Elastic-Plastic)

Default heading: Material Type 4 (Thermo-Elastic-Plastic)

NPTS *n*	Number of temperature values for which material constants are defined.
TEMP T_1 T_2...T_n	Temperatures
E E_1 E_2...E_n	Young's moduli
PR u_1 u_2...u_n	Poisson's ratios
ALPHA a_1 a_2...a_n	Coefficients of thermal expansion.
SIGY s_{y1} s_{y2}...s_{yn}	Yield stresses
ETAN E_{t1} E_{t2}...E_{tn}	Tangent moduli

Material Type 5 (Soil and Crushable Foam)

Default heading: Material Type 5 (Soil and Crushable Foam)

Input any two of the following:

BULK *K*	Bulk modulus
E *E*	Young's modulus
G *G*	Shear modulus
PR n	Poisson's ratio

Additional Options:

AO a_0	Yield function constant.
A1 a_1	Yield function constant
A2 a_2	Yield function constant
PC P_c	Pressure cutoff for tensile fracture
UL uopt	Unloading option =0: volumetric crushing =1: loading and unloading are the same.
NPTS *n*	Number of points in volumetric strain versus pressure curve (n£10).
VS e_{v1} e_{v2} ... e_{vn}	Volumetric strain values
P p_1 p_2...p_n	Pressures corresponding to volumetric strain values

The deviatoric yield function, ϕ, is described in terms of the second invariant J_2.

$$J_2 = \frac{1}{2} s_{ij} \; s_{ij}$$

Pressure, p, and constants a_0, a_1, and a_2 as:

$$\phi = J_2 - \left[a_0 + a_1 \, p + a_2 p^2 \right].$$

On the yield surface, $J_2=1/3(s_y)^{1/2}$, where s_y is the yield stress, i.e.,

$$\sigma_y = \left[3 \left(a_0 + a_1 p + a_2 p^2 \right) \right]^{1/2}$$

For elastic-perfectly plastic behavior $a_1=a_2=0$, and $(3a_0)^{1/2}$ defines the yield strength. The volumetric strain is given by the natural logarithm of the relative volume V. If the pressure drops below the cutoff value, PC, then it is reset to that value.

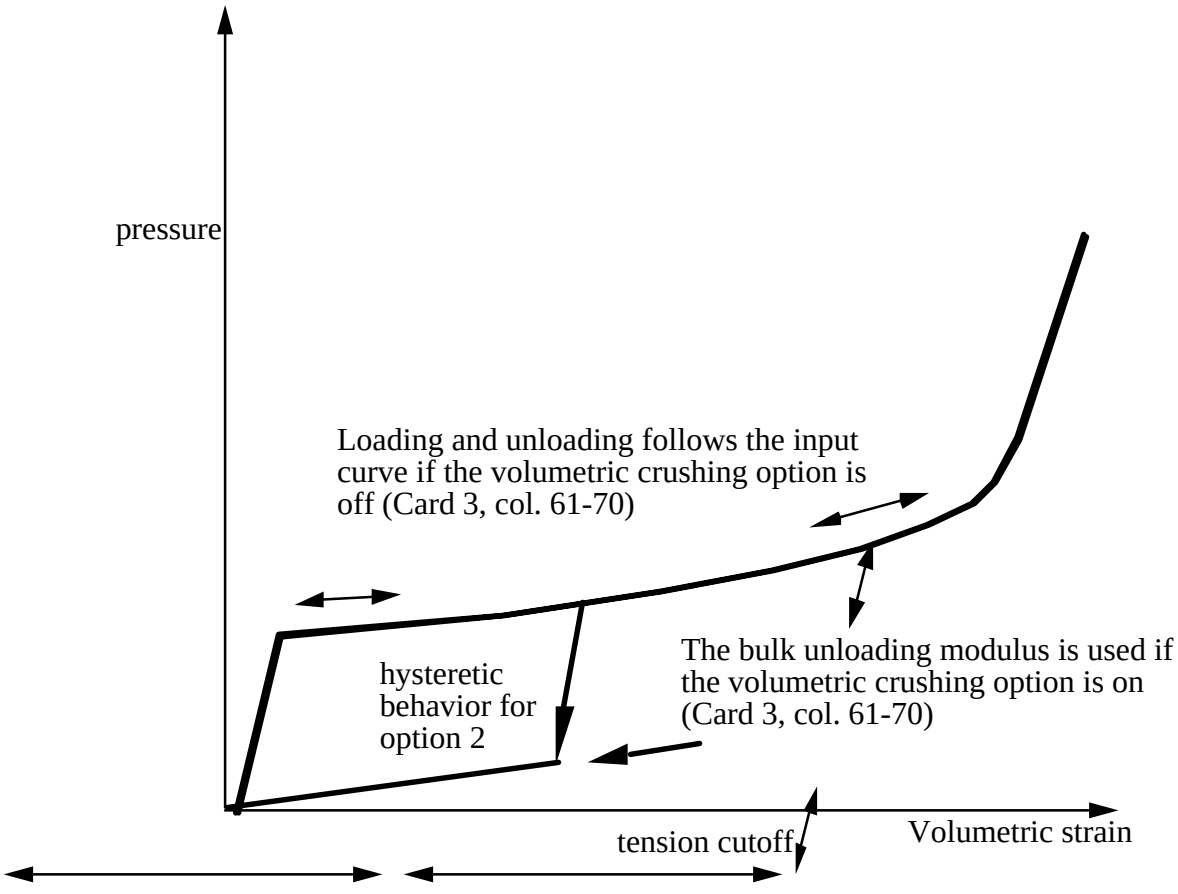


Figure 23-3. Volumetric strain versus pressure curve for soil and crushable foam model.

Material Type 6 (Viscoelastic)

G G_0	Short term shear modulus.
GI G_∞	Long term shear modulus.
K K	Bulk modulus.
BETA b	Decay constant.

The shear relaxation behavior is described by:

$$G(t) = G_\infty + (G_0 - G_\infty)\, e^{-\beta t}$$

A Jaumann rate formulation is used:

$$\overset{\nabla}{\sigma}'_{ij} = 2 \int_0^t G(t-\tau)\, D'_{ij}(\tau)\, dt$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{\sigma}_{ij}$, and the strain rate D_{IJ}.

Material Type 7 (Thermal Orthotropic Elastic)

Default heading: Material Type #7 (Thermal Orthotropic Elastic)

EA E_a	See constitutive matrix below.
EB E_b	
EC E_c	
PRBA υ_{ba}	
PRCA υ_{ca}	
PRCB υ_{cb}	
ALPA α_a	Thermal expansion coefficient along material axis a.
ALPB α_b	Thermal expansion coefficient along material axis b.
ALPC α_c	Thermal expansion coefficient along material axis c.
GAB G_{ab}	
AOPT aopt	Material axes option (Figure 22-1). =0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 22-1).

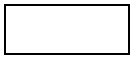
	=1.0: locally orthotropic with materials axes determined by a point in space and global location of element center.
	=2.0: globally orthotropic with materials axes determined by ψ_G.
RP r_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
PSIG ψ_G	In radians, define for AOPT = 2.

Material Type 8 (Thermo-Elastic-Creep)

Default heading: Material Type #8 (Thermo-Elastic-Creep)

NPTS n	Number of temperature values for which material constants are defined.
TEMP T_1 T_2...T_n	Temperatures
G G_1 G_2...G_n	Shear moduli
K K_1 K_2...K_n	Bulk moduli
ALPHA a_1 a_2...a_n	Coefficients of thermal expansion
A a_1 a_2...a_n	Creep parameters
B b_1 b_2...b_n	Creep parameters

In this model, *G*, is the shear modulus and the instantaneous creep is given by a power law of the form



where *a* and *b* are functions of temperature. This model was developed and provided for LS-NIKE3D by R. D. Krieg of Sandia National Laboratories.

Material Type 9 (Blatz-Ko Rubber)

Default heading: Material Type #9 (Rubber)

G μ	Shear modulus

The second Piola-Kirchhoff stress is computed as

where V is the relative volume, C_{ij} is the right Cauchy-Green strain tensor, and n is the Poisson's ratio which is set to .463 internally. This stress measure is transformed to the Cauchy stress, s_{ij}, according to the relationship

$$\sigma_{ij} = V^{-1} F_{ik} F_{jl} S_{kl}$$

where F_{ij} is the deformation gradient tensor.

Material Type 10 (Power Law Thermo Plasticity)

NPTS n	Number of temperature points (£8).
T T_1 T_2...T_n	Temperatures
E E_1 E_2...E_n	Young's moduli
PR u_1 u_2...u_n	Poisson's ratios
K k_1...k_n	See equation below
M m_1...m_n	See equation below

The stress-strain curve for this model is based on the following equation:

$$\sigma_y = k(\varepsilon_e + \bar{\varepsilon}^p)^n$$

Material Type 11 (Compressible Mooney-Rivlin Rubber)

This material model provides an alternative to the Blatz-Ko rubber model. The implementation is due to Maker [12].

A A	Constant A.
B B	Constant B.
PR n	Poisson's ratio.

The strain energy density function is defined as:

$$W=A(\text{I}-3)+B(\text{II}-3)+C(\text{III}^{-2}-1)+D(\text{III}-1)^2$$

where

C=0.5A+B.

$$D=\frac{A(5\nu-2)+B(11\nu-5)}{2(1-2\nu)}$$

n=Poisson's ratio.

2(A+B)=G=shear modulus of linear elasticity.

I, II, III are invariants of the right Cauchy-Green Tensor.

Material Type 20 (Rigid Body)

All elements with the same material number become a single rigid body if the material is type 20 whether the elements are connected or not. Density and two independent material strength constants are required to establish penalties for contact surfaces and joints.

Input any two of the following:

BULK K	Bulk modulus.
E E	Young's modulus.
G G	Shear modulus.
PR n	Poisson's ratio.

Additional *Options*:

LC lc	Load curve number for displacement control.
SCALE scale	Scale factor.
VVEC v_x v_y v_z	Define the vector **v** for the direction cosines.

Material Type 23 (Thermal Orthotropic with Curves)

NPTS npts	Number of points (1<NPTS<50).
EA $(E_a)_1$ ... $(E_a)_n$	
EB $(E_b)_1$ ... $(E_b)_n$	
EC $(E_c)_1$ ... $(E_c)_n$	
PRBA $(n_{ba})_1$ ... $(n_{ba})_n$	
PRCA $(n_{ca})_1$ ... $(n_{ca})_n$	

PRCB $(n_{cb})_1$ ... $(n_{cb})_n$
AA $(a_a)_1$ ... $(a_a)_n$
AB $(a_b)_1$ ... $(a_b)_n$
AC $(a_c)_1$ ... $(a_c)_n$
GAB $(G_{ab})_1$ ... $(G_{ab})_n$
GBC $(G_{bc})_1$ ... $(G_{bc})_n$
GCA $(G_{ca})_1$ ... $(G_{ca})_n$

AOPT aopt — Material axes option (Figure 23-1).

=0.0: locally orthotropic with materials axes determined by element nodes n_1, n_2, and n_4, (see Figure 23-1).

=1.0: locally orthotropic with materials axes determined by a point in space and global location of element center.

=2.0: globally orthotropic with materials axes determined by vectors defined below.

=3.0: SHELL ELEMENTS ONLY: The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.

XP x_p	Define for AOPT = 1.
YP y_p	Define for AOPT = 1.
ZP z_p	Define for AOPT = 1.
A1 a_1	Define for AOPT = 2.
A2 a_2	Define for AOPT = 2.
A3 a_3	Define for AOPT = 2.
D1 d_1	Define for AOPT = 2.
D2 d_2	Define for AOPT = 2.
D3 d_3	Define for AOPT = 2.
V1 v_1	Define for AOPT = 3.
V2 v_2	Define for AOPT = 3.
V3 v_3	Define for AOPT = 3.

24. TOPAZ Commands and Materials

Analysis options are code dependent. They can be set either in the control section of the LS-INGRID input file or in the interactive phase. These commands become active when TOPAZ2D or TOPAZ3D output is selected with the TZ2D or TZ3D commands respectively.

BWMO	*n*	Bandwidth minimization option. ="ON": minimize bandwidth (default). ="OFF": don't minimize bandwidth.
DCMX	d*t*	Desired maximum temperature change in each time step above which the time step will decrease.
DCTOL	tol	Convergence tolerance for equilibrium iterations (default = 0.0001).
DELT	D*t*	Time step size for fixed time step and initial time step for variable time step.
DTMAX	Dt_{max}	Maximum time step size.
DTMIN	Dt_{min}	Minimum time step size.
FLUX	*n*	Nodal heat flux calculations ="ON": perform calculations ="OFF": don't perform calculations (default).
IPLT	*n*	Number of time steps between output of graphics database.
IPRT	*n*	Number of time steps between output printouts.
IUNIT	*n*	Temperature units ="DIME": dimensionless ="CENT": centigrade ="FAHR": fahrenheit ="KELV": Kelvin =."RANK": Rankine
LINEAR		Problem is linear.
MFTS	*t*	Modification factor for increasing/decreasing time step.
MRDI	*m*	Maximum number of radiosity iterations.
MSRF	*n*	Maximum number of conductance matrix reformations per time step (default = 10).

NBEI	*n*	The number of time steps between equilibrium iterations (default =1).
NBSR	*n*	The number of time steps between conductance matrix reformation (default = 1).
NIBSR	*n*	Maximum number of equilibrium iterations permitted per conductance matrix reformation.
NIP1	*x*	First Newmark integration parameter. (default = 0.5). =1.0: fully implicit
NONLINEAR		Problem is non-linear.
NSSD	*n*	Number of surface subdivision for radiation view factor calculation (default = 5).
PHASE	*n*	Phase charge flag = "ON": perform phase change calculation = "OFF": no phase change calculation (default).
RADI	*n*	Radiation calculation type = "VIEW": view factors = "EXCH": exchange factors
RBAND	*m n*	Radiation bands. The number of wavelength break points is *m* and the number of curves is *n*.
	l_1 l_2...l_m	Wavelength breakpoints
	$(E_1)_l$ $(E_2)_l$ ...$(E_m)_l$	Emissivities for curve *l*
	$(E_1)_n$ $(E_2)_n$ ...$(E_m)_n$	Emissivities for curve *n*
RCTOL	*s*	Radiosity convergence tolerance.
RELAX	*r*	Relaxation parameter (default =1).
SBC	*s*	Stefan-Boltzmann constant.
SBRF	*n*	Number of time steps between restart dumps.
SOLUTION **STEA**DY		Analysis is steady state.
SOLUTION **TRAN**SIENT		Analysis is transient.
STEP	*n*	Time step code ="FIXE": fixed time step ="VARI": variable time step.
TERM	*t*	Final problem time.

TIMIN	*t*	Initial problem time.

24.1 TOPAZ MATERIAL INPUT

TOPAZ material input is possible after the TZ2D or TZ3D command has been input (see *Control Commands*). The form of this input is: MAT *n* TYPE *m* {options specific to material type *m*} {general material options} ENDMAT. *n* is a material name which is assigned a number in the order that they occur in the input. Therefore, the materials should be defined in order before any additional use of materials is made.

Material Type #1 (Isotropic)

RO r	Density
TLHA t	Temperature at which latent heat is absorbed or released.
LH h	Latent heat.
TGC lc	Thermal generation rate curve number.
TGM r	Thermal generation rate multiplier.
HCP c	Heat capacity.
K k	Thermal conductivity
ENDMAT	End this material model.

Material Type 2 (Orthotropic)

RO r	Density
TLHA *t*	Temperature at which latent heat is absorbed or released.
LH *h*	Latent heat.
TGC *lc*	Thermal generation rate curve number.
TGM *r*	Thermal generation rate multiplier.
HCP *c*	Heat capacity.
K1 k_1	Thermal conductivity in local 1 direction.
K2 k_2	Thermal conductivity in local 2 direction.
K3 k_3	Thermal conductivity in local 3 direction.
ENDMAT	End this material model.

Material Type 3 (Isotropic Temperature Dependent)

RO r	Density
TLHA *t*	Temperature at which latent heat is absorbed or released.
LH *h*	Latent heat.
TGC *lc*	Thermal generation rate curve number.
TGM *r*	Thermal generation rate multiplier.
NPTS *n*	Number of temperature points.
TEMP T_1 T_2...T_n	Temperatures.
CP C_1 C_2...C_n	Heat capacities
K K_1 K_2...K_n	Thermal conductivities.
ENDMAT	End this material model.

Material Type 4 (Orthotropic Temperature Dependent)

RO r	Density
TLHA t	Temperature at which latent heat is absorbed or released.
LH h	Latent heat.
TGC lc	Thermal generation rate curve number.
TGM *r*	Thermal generation rate multiplier.
NPTS *n*	Number of temperature points.
TEMP T_1 T_2...T_n	Temperatures.
CP C_1 C_2...C_n	Heat capacities
K1 $(K_1)_1$ $(K_1)_2$...$(K_1)_n$	Thermal conductivities in local 1 direction.
K2 $(K_2)_1$ $(K_2)_2$...$(K_2)_n$	Thermal conductivities in local 2 direction
K3 $(K_3)_1$ $(K_3)_2$...$(K_3)_n$	Thermal conductivities in local 3 direction.
ENDMAT	End this material model.

ACKNOWLEDGMENTS

Any work of this magnitude obviously was influenced by a large number of people who cannot possibly be given proper credit. The authors very much appreciate the all of the inputs whether positive or hostile which have aided in this work. The original work of Bill Cook on INGEN influenced LS-INGRID considerably. Special thanks must be given to Russ Rosinsky for his patience in finding bugs, recommending new capabilities and proof reading this manual. Steve Sackett, Greg Kay, and Tracy Glover also helped in providing sample problems, new ideas, and uncovering bugs.

Developments at SPARTA benefited from David Lichtblau's and Brian Wainscott's work on the calculator program. Sophie Tsui and Dawn Greayer made contributions to the materials processing portion and Bill Campbell provided some useful ideas for surface intersection algorithms. Sharon Kiefer made some important contributions in debugging and improving the manufacturing capabilities. The NURB curve and surface algorithms were developed by Alan Winslow.

Eunice Hinkle, Nikki Falco, Valli A. James and Debbie Aiken all suffered through the preparation of various versions of this manual. The University of Tennessee, Lawrence Livermore National Laboratory and SPARTA, Inc. all generously provided computer resources to allow LS-INGRID to be developed.

REFERENCES

[1] Hallquist, John O., "LS-NIKE2D User's Manual", LSTC Report 1006, 1990

[2] Hallquist, John O., "NIKE3D: An implicit, finite-deformation, finite element code for analyzing the static and dynamic response of three-dimensional solids", University of California, Lawrence Livermore National Laboratory, UCID-18822, 1984.

[3] Hallquist, John O., "LS-DYNA2D User's Manual", LSTC Report 1004, 1990

[4] Hallquist, John O., "LS-DYNA3D User's Manual", LSTC Report 1007, 1990

[5] Shapiro, Arthur B., "TOPAZ2D - A three-dimensional finite element heat transfer code", University of California, Lawrence Livermore National Laboratory, Rept. UCID-20484 (1985).

[6] Shapiro, Arthur B., "TOPAZ3D - A three-dimensional finite element heat transfer code", University of California, Lawrence Livermore National Laboratory, Rept. UCID-20484 (1985).

[7] Cook, William A., "INGEN: A General Purpose Mesh Generator for Finite Element Codes," Los Alamos Scientific Laboratory, Rept. LA-7135-MS (1978).

[8] Hallquist, John O., "LS-MAZE: An Input Generator for NIKE2D, DYNA2D, and TOPAZ2D", LSTC Report 1005, 1990.

[9] Steinberg, D.J. and M.W. Guinan, "A High-Strain-Rate Constitutive Model for Metals," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-80465 (1978).

[10] Woodruff, J.P. "KOVEC User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).

[11] Johnson, G.R. and W.H. Cook, "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures." Presented at the Seventh International Symposium on Ballistics, The Hague, Netherlands, April 1983.

[12] Maker, B.N., Private communication, Lawrence Livermore National Laboratory.

[13] Kenchington, G.J.,"A Non-Linear Elastic Material Model for DYNA3D, "Proceedings of the DYNA3D User's Group Conference, September 1988, published by Boeing Computer Services (Europe) Limited.

[14] Cochran, S.G., and J. Chan, "Shock Initiation and Detonation Models in One and Two Dimensions," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18024 (1979).

[15] Lee, E.L., and C.M. Tarver, "A Phenomenological Model of Shock Initiation in Heterogenous Explosives," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-83618 (1979).

Index