True*Grid*®Output Manual For DYNA3D

A Guide and a Reference

by

Robert Rainsberger

Version 2.3.0

XYZ Scientific Applications, Inc.

October 12, 2006

Table of Contents

Table of Contents 3

I. DYNA3D Output Guide 5
Introduction 5
Font Conventions 5
Supported Features 5
Sliding (or Contact) Surfaces 7
Initial and Boundary Conditions 8
Loads 9
Load Curves 10
Stone Walls 11
Bricks 11
Shells 11
Thick Shells 12
Beams 12
Joints and Rigid Bodies 13
Springs and Dampers and Point Masses 13
Temperatures 13
Shared Constraints 13
Tied with Failure 14
Post Processing 14

II. DYNA3D Output Example 15

III. DYNA3D Output Reference 19
Command Syntax Conventions 19
bsd global beam cross section definition 19
sid sliding interface definition 21
si select nodes or faces for sliding interface in merge phase 23
si assign sliding interface to region in part phase 23
sii assign sliding interfaces to progression in part phase 23
spd define the properties of a set of springs or dampers 25
dynaeos DYNA3D equation of state 26
dynamats DYNA3D materials 30
dynaopts DYNA3D analysis options 47

IV. Frequently Asked Questions 59

V. INDEX 61

I. DYNA3D Output Guide

Introduction

DYNA3D is a nonlinear, explicit, three-dimensional finite element code for solid and structural mechanics developed at Lawrence Livermore National Laboratories. The focus in this manual will be on those features in **True*Grid*®** that are specific to creating a DYNA3D input file. The **True*Grid*®** User Manual covers the creation of a mesh and will not be covered in this manual. This manual is incomplete in another way because it cannot be used as a substitute for the DYNA3D manual. For a full understanding of the use of these features, the user must have a working knowledge of DYNA3D and be familiarity with a DYNA3D User Manual and, in particular, the one written by Jerry I. Lin, dated January, 2005, UCRL-MA-107254.

Font Conventions

Different fonts are used through out this manual to indicate their meaning. A literal is highlighted in bold. A symbol to be substituted with a literal or a number is *italicized*. A computer example uses the `Courier font`.

Supported Features

There are many features in **True*Grid*®** to create a model for DYNA3D. The table below shows the commands that are used for each feature. Sometimes there are several commands listed. For example, shells can be generated using both the **block** and **cylinder** commands. The **n** and **th** are used to set the properties of these shells. In another example, the **si** and **sii** commands are used to identify the faces of the mesh that form the sliding (or contact) surfaces. The associated **sid** command is used to assign properties to the sliding surface.

DYNA3D feature	**True*Grid*®** commands
parameters in the control cards	**dynaopts**
materials	**dynamats, mate, mt, mti**
equation of state	**dynaeos**
truss and beam element cross section properties	**dynamats, bsd**
shell element cross section properties	**dynamats**
thick shell element cross section properties	**dynamats**
beam user defined integration rules	**bind**
shell user defined integration rules	**sind**
delamination elements	(under development)
cohesive elements	(under development)
nodal boundary conditions	**b, bi**

symmetry	**plane**
solid elements	**block, cylinder, or**
beam and truss elements	**block, cylinder, ibm, jbm, kbm, bm**
shell and membrane elements	**block, cylinder, n, th, thi, ssf, ssfi, or**
thick shell elements	**block, cylinder, or**
interface save segment definition	**iss**
nodal arbitrary single point constraints	**lsys, lb, sfb**
sliding boundary planes	**plane, sfb**
symmetry planes with failure	**plane, syf**
node time history blocks	**npb**
element time history blocks	**epb**
gravity stress initialization	**dynaopts** (gvst)
brode functions	**dynaopts**
cross section definitions for force output	**csf**
load curves	**lcd, flcd**
nodal forces and follower forces	**fc, mom, ndl, ffc, fmom**
pressure loads	**pr, pramp, dom, arri, dist**
prescribed velocities/accelerations/displacements	**fv, acc, fd, frb, dynamats** (bpm)
rigid or stone walls	**sw**, **swi**
coupled or shared nodal constraints	**mpc**
spot welds	(under development)
rigid node set	(under development)
initial velocity conditions	**rotation, velocity, ve**
material initial rotation	(under development)
sliding (or contact) surface	**sid, si, sii**
tie-breaking shell slide line	(under development)
tied node sets with failure	**fn**, **fni**
rigid body merges	**rigbm**
extra nodes with rigid bodies	**jt**
rigid body joints	**jd, jt**
prescribed base accelerations	**dynaopts** (grav)
prescribed angular velocities	**dynaopts** (xvel, yvel, zvel)
momentum deposition in solid elements	**mdep**
detonation points	**detp**
shell-solid interfaces	(under development)
discrete springs, dampers, and masses	**spd, spdp, spring, pm, npm**
rigid body inertial properties	(under development)
nonreflecting boundary segments	**nr**, **nri**
temperature input option I	**tepro**
temperature input option II	**temp, te, tei**
one dimensional slide line	**sid, si, sii**
material initialization for rotational motion	(under development)

body force by material	(under development)
CVS (MADTMO/ATB) Coupling	(under development)
Air Bag Gas Flow Definitions	(under development)
slide surface activation/deactivation times	(under development)
fiber orientation	(under development)
set the problem title	**title**
comments for specific sections of the output	**comment**
merge parts into a single connected model	**stp, tp, t**
view properties in the merge phase	**co**
select the DYNA3D output format	**dyna3d**
write a DYNA3D input file	**write**

You may want to view some of the properties graphically using the **condition** (**co**) command in the merge phase. The **tmm** command can be used to calculate the mass of each part. Be sure to merge the nodes using one of the merging commands such as **stp** and, finally, use the **dyna3d** command to select DYNA3D as the output option and the **write** command to actually create the input deck forDYNA3D.

The file produced by **True*Grid*®** is an ASCII file that can be examined or modified using any text editor. Some experienced users always inspect the file and modify it instead to rerunning **True*Grid*®** when make easy changes to the model such as changing a material model parameter or the time step. For this reason, the output file has helpful comments. However, this file can be very large and it might be easier to modify the **True*Grid*®** session file and rerun **True*Grid*®**. One of the comments that is automatically written has a time stamp for archiving purposes. The **title** is also helpful for archiving. You can insert your own comments with the use of the **comment** command.

The **readmesh** command has a dyna3d option so that you can import a DYNA3D model into **True*Grid*®**.This is intended to be used to translate a DYNA3D input file into another format or to make small modifications to a model when a session file does not exist. This feature does not replace the session file because the block structure of the mesh cannot be reconstructed. This also means there are no block boundary interfaces (**bb** command) to utilize. If there is a block structure underlying the mesh, you can form a block boundary interface using the **mbb** command, but this can be tedious. As a cautionary note, because DYNA3D has header data that prescribe the bulk data that follows it in the input file, when the format is changed due to additions, until the **readmesh** has been updated, the additional data in the DYNA3D file will cause errors in the **readmesh** command. You may have to experiment or check the documentation on **readmesh** to determine which features are support in **readmesh**.

Sliding (or Contact) Surfaces

To form a contact surface, use the **sid** command to define the surface type. Some types have only one side. Some are formed from faces of bricks or shells. Others are formed partially from nodes. The

sid command also has optional parameters such as friction. Be sure to choose a DYNA3D type, since other types will not be recognized when writing the output file.

While in the part phase use the **si** or **sii** commands to select faces of that part for inclusion in the surface definition. If the face is from a shell element, be sure to use the **orpt** orientation command prior to issuing the **si** or **sii** command so that the orientation of the face is towards the opposing face in the sliding surface. If you are using part replication (**lrep**, **grep**, or **pslv**), then you may want to use the **lsii** or the **gsii** to increment the sliding interface command for each replication. You must use the **sid** command for each sliding surface that is referenced when the **lsii** or **gsii** commands are used with replication.

You can use sets in the merge phase to add faces or nodes to a sliding surface. These sets can be formed with the combined use of the **fset** (for faces) and the **nset** (for nodes) commands in the part and merge phase. Only use node sets when defining a sliding surface where nodes are on the slave side and otherwise only use face sets. The node density between the master and slave sides of the interface should be roughly equal. When forming the mesh in the part phase, it may be necessary to build into the mesh a small gap between the master and slave sides of the contact surfaces, depending on the mesh density and the curvature to avoid initial penetration of the slave side into the master side.

When you merge the nodes (in the merge phase), the nodes from the slave side will not be allowed to merge with the nodes on the master side. Use the **mns** command in the merge phase to override this condition. When you fist merge the nodes, a table will be printed to the text window and the tsave file listing the number of faces and nodes associated with each sliding surface. Check this table carefully. You can also see the faces and nodes of either side of the sliding surfaces using the **co** command. When using this in combination with the hide graphics option, you can see the orientation of the faces. Use **labels** command to show how the nodes have merged graphically.

Initial and Boundary Conditions

There are several ways to constrain nodes. The **b** and **bi** commands in the part phase or the **b** command in the merge phase will constrain nodes in the global coordinate system. Use the **plane** command to specify symmetry plane constraints including symmetry planes with failure. Nodes in the model will be assigned to these symmetry planes based on the tolerance you specify in the **plane** command. The **lb** (and the associated **lsys**) command can be used to set the constraints in any coordinate system. The **sfb** command can also be used to do this. Be sure that something in the model has been constrained or the entire model might fly off.

To set non-reflective (or transmitting) boundary conditions, use the **nr** and **nri** commands in the part phase or the **nr** command in the merge phase. Special care is needed when developing a model using this type of boundary condition. See the DYNA3D User Manual for details.

If you use the **velocity** or **rotation** command in the control phase, then all subsequent parts will be assigned this initial velocity. This can be over ridden using the **velocity** or **rotation** command within a part. Both of these conditions can be over ridden for specific regions of the mesh using the **ve** or **vei** commands in the part phase or the **ve** command in the merge phase. Velocities are not accumulative. Care is needed when assigning initial velocities so that when two nodes are merged, the velocities of those two nodes match. Only one of the velocities will be used and if they do not match, you may get an unexpect result. Usually, if the velocities of two merged nodes do not match, this indicates an error in the model. There are other concerns regarding incompatible initial velocities and prescriptions and the DYNA3D User Manual discusses this issue. **True*Grid*®** does not protect you from or identify these incompatibilities.

Loads

There are numerous ways to assign loads. The list of commands that can be used to assign loads in the part phase includes:

fc	Cartesian concentrated nodal loads
fci	Cartesian concentrated nodal loads
fcc	cylindrical concentrated nodal loads
fcci	cylindrical concentrated nodal loads
fcs	spherical concentrated nodal loads
fcsi	spherical concentrated nodal loads
mom	nodal moment about one of the nodal axis in the global coordinate system
momi	nodal moment about one of the nodal axis in the global coordinate system
ndl	pressure converted to distributed nodal loads
ndli	pressure converted to distributed nodal loads
pr	pressure loads on element faces
pri	pressure loads on element faces
pramp	pressure loads on element faces
fv	Cartesian prescribed nodal velocities
fvi	Cartesian prescribed nodal velocities
fvc	cylindrical prescribed nodal velocities
fvci	cylindrical prescribed nodal velocities
fvs	spherical prescribed nodal velocities
fvsi	spherical prescribed nodal velocities
fvv	Cartesian variable prescribed nodal velocities
fvvi	Cartesian variable prescribed nodal velocities
fvvc	cylindrical variable prescribed nodal velocities
fvvci	cylindrical variable prescribed nodal velocities
fvvs	spherical variable prescribed nodal velocities
fvvsi	spherical variable prescribed nodal velocities
acc	Cartesian prescribed nodal acceleration

acci	Cartesian prescribed nodal acceleration
accc	cylindrical prescribed nodal acceleration
accci	cylindrical prescribed nodal acceleration
accs	spherical prescribed nodal acceleration
accsi	spherical prescribed nodal acceleration
vacc	Cartesian variable prescribed nodal acceleration
vacci	Cartesian variable prescribed nodal acceleration
vaccc	cylindrical variable prescribed nodal acceleration
vaccci	cylindrical variable prescribed nodal acceleration
vaccs	spherical variable prescribed nodal acceleration
vaccsi	spherical variable prescribed nodal acceleration
fd	Cartesian displacement
fdi	Cartesian displacement
fdc	cylindrical displacement
fdci	cylindrical displacement
fds	spherical displacement
fdsi	spherical displacement
frb	prescribed rotation
frbi	prescribed rotation

The list of commands that can be used to assign loads in the merge phase includes:

fc	Cartesian concentrated nodal loads
mom	nodal moment about one of the nodal axis in the global coordinate system
ndl	pressure converted to distributed nodal loads
pr	pressure loads on element faces
pramp	pressure loads on element faces
fv	Cartesian prescribed nodal velocities
fvv	Cartesian variable prescribed nodal velocities
vacc	Cartesian variable prescribed nodal acceleration
fd	Cartesian displacement
frb	prescribed rotation
ffc	concentrated nodal load with a follower force
fmom	nodal moment with a follower force

The **pramp** command is used with either **pr** or **pri**. It applies a pressure based on a function for all nodes that have a zero pressure. In most cases, the magnitude of the load is specified using a load curve. This varies the amplitude of the load with respect to time.

Load Curves

Load curves are 2D polygonal curves that can be created using the **lcd** and **flcd** commands. Load curves are typically used to define the relative amplitude of a load with respect to time. They can be used to relate any two variables. Almost all prescribed loads require a load curve in time so that the amplitude of the load can vary. It is best to define a load curve before it is referenced in a load or material model to avoid a warning message. When the output file is written, if a load curve is referenced but not defined, you will also receive a warning message. Then a simple load curve will be used in the output file so that a valid DYNA3D file is written. It is advised that you correct this by defining the appropriate load curve for the problem. Do not rely on the load curve that is automatically generated.

In some dialogue boxes you might be prompted for a load curve or a set id. This is because such commands can be used to define, for example, a dynamic load for DYNA3D or a static load for another output option that has the option to turn loads on or off depending on the set id. Simply ignore the set id portion of the prompt and supply the load curve number.

Stone Walls

A stone wall is defined with two commands. Use the **plane** command to set the properties of the stone wall. Nodes to react to the stone wall will not be selected automatically based on the tolerance. Use the **sw** and **swi** commands in the part phase to assign faces of the model to react to the stone wall. You can use the **sw** command in the merge phase as well to assign faces from a face set to react to the stone wall.

Bricks

Brick elements refer to hexahedral, prism (wedge), and tetrahedral elements and are considered the same type in DYNA3D. Only one element type can associated with a material definition. If you want two different element types with the same material properties, you must define two materials. Most, but not all, materials support the different brick element types. There are no section properties for bricks. Be sure to use the **mate**, **mt**, or **mti** command to assign the proper material to each section of the mesh.

The element local coordinate system used in an orthotropic or anisotropic material is imposed by the order of the nodes that define the element. You can flip the nodal ordering to switch the orientation of this local coordinate system using the **or** command in the part phase.

Shells

Shell elements refer to both quadrilateral and triangular elements and sometimes referred to as structural elements. Cross sectional properties are included in the material model when the shell type is selected. There are no section properties for bricks. Be sure to use the **mate**, **mt**, or **mti** command to assign the proper material to each section of the mesh. The default shell thicknesses are included

as part of the cross sectional properties. These default thicknesses can be over ridden with the use of the **thic** command in the part phase. Both can be over ridden for a region of the part using the **th** and **thi** commands. If you have two surfaces that represent the inner and outer surfaces of a structure that is to be modeled using shell elements, than you can use the **ssf** and **ssfi** commands in the part phase to create shells with variable thickness.

The orientation of the positive normal direction to the shell is dictated by the nodal ordering of the nodes that define the shell. This positive direction is used, for example, to determine the direction of a positive pressure. This direction can be flipped using the **n** command in the part phase. The order of the nodes also dictate the local material coordinate system which can be important when using an orthotropic or anisotropic material. Use the **or** command to flip the coordinate system to the desired direction. When an angle is specified for the orientation of a composite material, it is with respect to this orientation.

You may need to specify the through thickness integration points when defining a composite material. This can be done by defining an integration rule with the **sind** command. Then identify this rule in the material definition.

Thick Shells

Thick shells are generated and look like hexahedral elements. Only a few materials support thick shells. Be sure to use the **mate**, **mt**, or **mti** command to assign the proper material to each section of the mesh.

The element local coordinate system used in an orthotropic or anisotropic material is imposed by the order of the nodes that define the element. You can flip the nodal ordering to switch the orientation of this local coordinate system using the **or** command in the part phase.

You may need to specify the through thickness integration points when defining a composite material. This can be done by defining an integration rule with the **sind** command. Then identify this rule in the material definition. All other cross section information is specified in the material definition.

Beams

Two nodes are required to form a beam element. In many cases, a third node is needed to define the local coordinate system used to form the cross sectional properties. These element are sometimes referred to as structural elements. Use the **ibm**, **ibmi**, **jbm**, **jbmi**, **kbm**, and **kbmi** commands to form beam elements with shell or brick structures while in the part phase. If the material of the shell or brick structure is set to zero using the **mt**, **mti**, or **mate** command, then the shells or bricks will be ignored, but the embedded beams will not be ignored. This is a convenient way to build an array of beams using block structured methods. You can also use the **bm** command in the merge phase to

build a string of beams that can be made to follow a 3D curve. The **beam** command (this command has been denigrated) can also be used to form beam elements, but the command is not interactive.

Both the element type and the default cross section properties are defined in the material definition. You can also use the **bsd** command to define cross sectional properties to over ride the material default cross sectional properties. When you create a beam, refer to the **bsd** number to assign these cross sectional properties to the beam. Use the **bind** command to define beam integration rule, if needed. Then refer to this integration rule when defining the material.

Joints and Rigid Bodies

A rigid body is formed using shells and bricks that are assigned the rigid body material. Each rigid body can be attached to other parts of the model using joints. A joint is defined in two steps. The **jd** command is used to define the properties of a joint. Then the **jt** command is used to identify which nodes are used to form the joint. Nodes within a joint are not merged.

Springs and Dampers and Point Masses

Springs and dampers are treated the same in **True*Grid*®**. They are only distinguishable by the material properties assigned them. Use the **spd** command to define the properties of the spring or damper. Then use the **spring** command to assign nodes to a numbered spring. Alternatively, the **spdp** command can be used in the part phase to create an array of springs between two parts, analogous to a contact surface.

Point masses can be generated in the part or merge phase. There are two types of point masses. The **pm** command will assign a mass to an existing node. The **npm** will create a new node and assign it a mass. The latter must then be connected either to a spring or beam.

Temperatures

There are two methods in DYNA3D to set the temperatures for material properties. For the first option, use the **tepro** command in the part or merge phase. You can use the **temp** command to set a default constant temperature. Then use the **te** and **tei** commands in the part phase or the **te** command in the merge phase to vary the temperature in different regions of the mesh.

Shared Constraints

Use the **mpc** command to couple a set of nodes. This requires that you create a node set first. The **nset** or **nseti** command can be used in the part phase and the **nset** command in the merge phase to create a node set. Also, click on the pick button in the environment window during the merge phase. Then you can use the mouse to modify or create a node set. The nodes sharing a set of constraints will not be merged together.

Tied with Failure

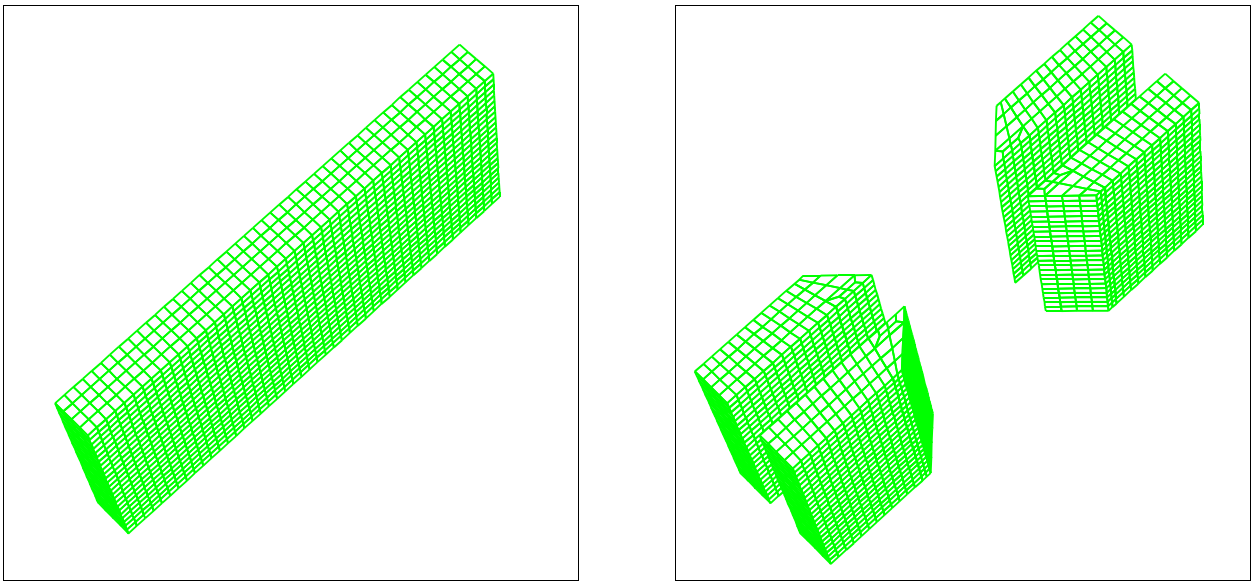
The **fn** and **fni** commands in the part phase will generate a shell mesh where there are 4 unique nodes for each shell element. These are automatically tied together with a failure criteria where shell elements meet. The nodes that are tied together are merged together in the merge phase.

Post Processing

There are a number of options of the **dynaopts** command to control the data saved in the database by DYNA3D for post processing. You can get more information from the reference section on the **iff**, **prti**, **plti**, **ssdm**, **drtflg** options of **dynaopts**. You may wish to analyze in greater detail the evolution of certain nodes or elements. Use the **npb** and **epb** commands (referred to as time history blocks), respectively, to identify areas of the mesh requiring a more detailed accumulation of data by DYNA3D.

II. DYNA3D Output Example

The following example was provided by Dr. Richard J. Fields at National Institute of Standards and Technology of the United States Department of Commerce. It has been modified from a model developed for another code.



This model forms a pair of clamps and a rectangular block of metal to be drawn as the clamps move apart. The first part forms both the rectangular metal to be drawn as well as the majority of the clamps. This removes the need of merging the nodes of the clamp with the nodes of the drawn material. The wedge portion is made as a separate part to take advantage of the transitional block boundary. The clamps are made rigid. The entire bottom face is constrained in the z-direction. Displacements are applied to the end faces of the clamps.

```
title Metal Drawing Process using DYNA3D by NIST Metallurgy Div.

c choose the output format
dyna3d

c set the termination time and the plot interval
dynaopts term 1.e-3 plti 1.e-4 ; ;

c load curve
lcd 1 0 0 1 1000.;
```

```
c elastic material for the sample material being drawn
dynamats 1 24 rho 7.e-4 mhead workpiece
         e 3.e7 pr 0.3 sigy 75000 et 1.e5 efp 0.2 ;

c rigid material for the clamps
dynamats 2 20 rho 7.e-4 e 3.e7 pr 0.3 ;

c main part with both elastic and rigid materials
block 1 13 17 31 35 47;1 5 9 13;1 25;
      1 13 17 31 35 47;1 5 9 13;1 25;

c remove some of the unneeded regions between the clamps
dei 2 5; 1 2 0 3 4;;

c set the default material to elastic
mate 1

c set the clamp regions of the part to rigid material
mti 1 3 0 4 6; 1 2 0 3 4; ; 2

c position some of the key nodes
mbi -2; -1 0 -4; 1 2;x -1
mbi -5; -1 0 -4; 1 2;x 1

c save the interface for the second part
bb 2 2 1 3 2 2 1;

c nodal constraints
bi ;; -1;dz 1;

c displacements
fdi -6;1 2 0 3 4;1 2;1 1 1 0 0
fdi -1;1 2 0 3 4;1 2;1 1 -1 0 0

endpart

c part to form the transitional region of the clamps
block 1 3;1 5;1 25; 13 17 1 5 1 25

c all rigid material
mate 2

c position some of the nodes
mb 1 1 1 1 1 2 x -1
mb 2 1 1 2 1 2 x -4
```

```
c transitional interface
trbb 1 2 1 2 2 2 1;

c replicate this part
lct 3 rzx my 14;ryz mx 48;rzx ryz my 14 mx 48;
lrep 0:3;

c nodal constraints
bi ;; -1;dz 1;

endpart

c enter the merge phase to write the output file
merge

c merge the nodes at the interfaces
stp .001

c write the output file
write
```

III. DYNA3D Output Reference

The commands found here are provided to the user so that a complete input file can be generated by **True*Grid*®**. This manual does not try to explain the meaning of these parameters. For this, the user is referred to the DYNA3D User Manual.

Command Syntax Conventions

When an arbitrarily long list of arguments are required, a semi-colon terminates the list. Sometimes the abbreviation *#_things* is used to mean "number of things". Each command is described by an entry like the following:

command summary description

command *arguments* brief description of functionality
with brief descriptions of what the *arguments* should be.
indentation is used to indicate a list of options to the *arguments*

Remarks

When present, the Remarks section describes the command in even greater detail. It may describe the context in which the command is normally used, and other commands used in association with this command. It may describe side effects. It may describe other, similar commands. In many cases, it includes a description of where to find the command in the menus.

Example

When present, this shows the exact use of the command. If you use the dialogues, this command will be generated by simple selection options with the mouse and entering data where indicated. The command, as shown here, will appear in the session file for later reuse and possible modification. You can also enter the command into the text window or insert it into a command file to be run in batch mode.

bsd global beam cross section definition

bsd *option_list ;*
where an *option* can be:

for the Hughes-Liu beam with constant thickness
sthi *thickness* s-thickness at both ends

tthi *thickness* t-thickness at both ends

for the Hughes-Liu beam with variable thickness

sthi1 *thickness* s-thickness at beginning
sthi2 *thickness* s-thickness at ending
tthi1 *thickness* t-thickness at beginning
tthi2 *thickness* t-thickness at ending

for the Belytschko-Schwer beam

carea *area* cross section area
iss *iss* area moment of inertia about s-axis
itt *itt* area moment of inertia about t-axis
irr *irr* area moment of inertia about r-axis
sarea *area* shear area of cross section

for the truss

carea *area* cross section area

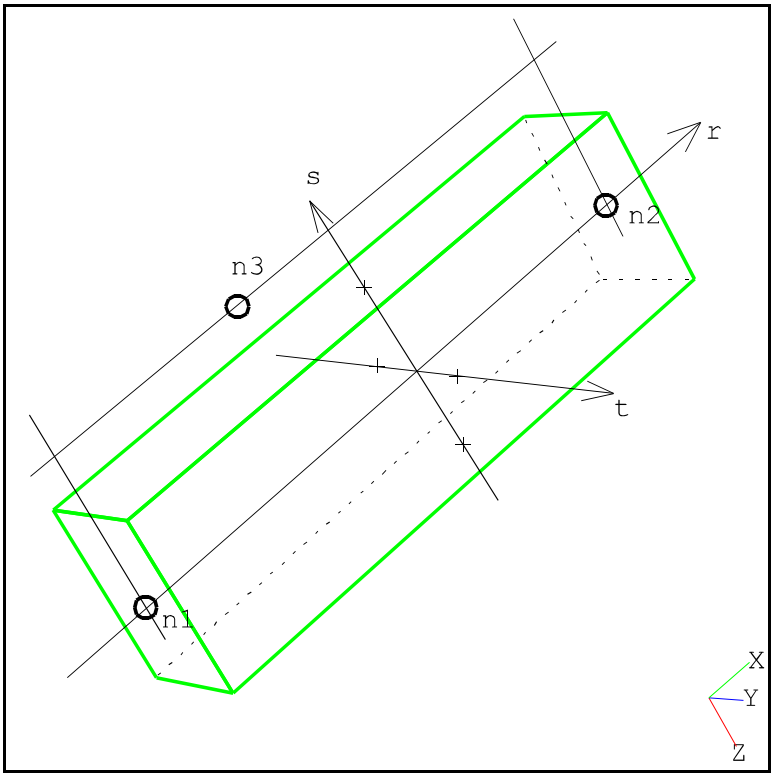


Figure 3 Beam Local Coordinate System for DYNA3D

Remarks

A third node is always required but is not significant for a truss. For other beam types, the third node is used to define the cross section orientation.

There are other options to this command, but they are not shown here because they do not apply to the DYNA3D output. For a complete description of the **bsd** command, see the **True*Grid*®** User Manual.

Since there are default cross section properties provided in the definition of the material using the **dynamats** command, not all of the parameters need to be assigned through the **bsd** command. Only use the parameters appropriate for the beam type selected in the **dynamats** command. Inappropriate parameters will be ignored.

Each cross section definition is assigned a number by you so that you can reference it when defining a set of beams with the **ibm**, **ibmi**, **jbm**, **jbmi**, **kbm**, **kbmi**, **bm**, and **beam** commands.

Example

```
bsd 2 sthi .03 tthi .03 ; ;
```

sid sliding interface definition

sid *slide_# option_list* ;

where an *option* can be

tied	tied sliding surface
sl	sliding only
sv	sliding with voids
single	single sided slide surface
dni	discrete nodes impacting surface
dnt	discrete nodes tied to surface
sets	hell element edge tied to shell element surface
nsw	nodes spot welded
break	tie-break interface
owsv	one way sliding with voids
dummy	is only used to insure that nodes in this interface will not be merged
sand *type*	Slide Surface with Adaptive New Definitions

where *type* can be

sms *slave_material_list* ;

mms *master_material_list* ;

auto for automatic contact
rebar *type* to define properties of REBAR 1D sliding interface
where type can be any of the following:
rbrad radius
rbstr strength
rbshr modulus
rbumax displacement
rbexp exponent
rbibond non-negative_number
fric *friction_factor* for static coefficient of friction
kfric *kinetic_coefficient_of_friction* for kinetic coefficient of friction
decay *exponential_decay_coefficient* for exponential decay coefficient
pen for small penetration flag
sfif for slave to be printed in force file
mfif for master to be printed in force file
pnlts *slave_penalty_factor* for slave penalty factor
pnltm *master_penalty_factor* for master penalty factor
pnlt *penalty_factor* for sliding penalty

Remarks

Sliding interfaces or contact surfaces are constructed in 3 steps. These steps can be done in any order.

1. define the properties
2. select the slave side
3. select the master side, if applicable

The **sid** command is used to define the properties. The **si** and **sii** commands are used in the part phase or the merge phase to select the nodes or faces that form the master and slave sides of the interface.

When nodes are merged, nodes across a sliding interface will not be merged. When a merge command is first issued in the merge phase, a table is written listing the number of nodes and faces associated with each sliding interface.

The **dummy** type interface is actually used to avoid merging of nodes. A sliding interface of this type is not written to the output file.

The nodes and faces of a sliding interface or contact surface can be viewed in the merge phase using the **si** option of the **co** command.

If the output option has been selected prior to using the dialogue box to make a selection, only the options available to that output option will be displayed in the dialogue box.

si select nodes or faces for sliding interface in merge phase

si *type interface_# boundary parameters ;*
where *type* and *parameters* can be one of:

n *node_numbe*r	to select a single node
rt *x y z*	to select a node close to a Cartesian point
cy *rho theta z*	to select a node close to a cylindrical point
sp *rho theta phi*	to select a node close to a spherical point
nset *name_of_set*	to select an entire node set
fset *face_set*	to select a face set

where *boundary* can be one of

m	master side of the interface
s	slave side of the interface

si assign sliding interface to region in part phase

si *region sliding_# type options*
where

sliding_#	reference number for the interface
type	**m** for master and **s** for slave
options	this depends on the *type*.

sii assign sliding interfaces to progression in part phase

sii *progression sliding_# type options*
where

sliding_#	sliding interface reference number
type	**m** for master and **s** for slave
options	this depends on the *type*.

Example

A model was created by the use of the following command file. Some normals are displayed as circular arcs with arrows. This is caused by the normals pointing almost orthogonally from the screen. There is an angle parameter in the **co** command setting the range of such behavior. You can modify it, or totally disable it.

```
c Sliding Interface -> master side
c Part definition -> shells.
```

```
block 1 3 5 7 9;-1;1 3 5 7 9; 0 2 4 6 8; 9 ; 0 2 4 6 8;
c Definition of orientation point in Cartesian coord x,y, and z
orpt - 9 0 9
c Definition of the type of the sliding interface
sid 1 sv ;
c Assignment of region (;-1;;) and type(1 m)of slid. interf.
sii ;-1;; 1 m
c Definition of 3 global transformations around y-axis
gct 3 ry 90; ry 180; ry 270 ;  c for 90,180 and 270 degrees.

c Global replication 3 times by rotation for
grep 0 1 2 3;  90,180 and 270 degrees
c Cylinder part -> tube made from hexahedrons.
cylinder 1 6; 1 3 5 7 9 11 13 15 17 19 21 23 25; 1 10;
     2 4;0 30 60 90 120 150 180 210 240 270 300 330 360;0 20;
c Definition of the orientation point in the default coordinate
c system of the part (in Cylindrical coordinates r,eta,z).
orpt - 0 20 5
c Assignment of region (-2;;) and type(1 s) of slid. interf.
sii -2;;; 1 s
c Assignment of region (;;-1;) and type(1 s) of slid. interf.
sii ;;-1; 1 s
c Assignment of region (;;-1;) and type(1 s) of slid. interf.
sii ;;-2; 1 s
lct 1 my 20 ;  c Definition of the local transformation
lrep 1;        c Transformation 1 is applied.
merge
labels size 3       c Scale the size of arrows.
rx 20 ry 20 rz 20   c Rotate mesh in window.
center              c Center picture in window.
set tv disp         c Set hide display option.
co si 1 m;     c Display of master side of sliding interface 1
co si 1 s;     c Display of slave side of sliding interface 1
```

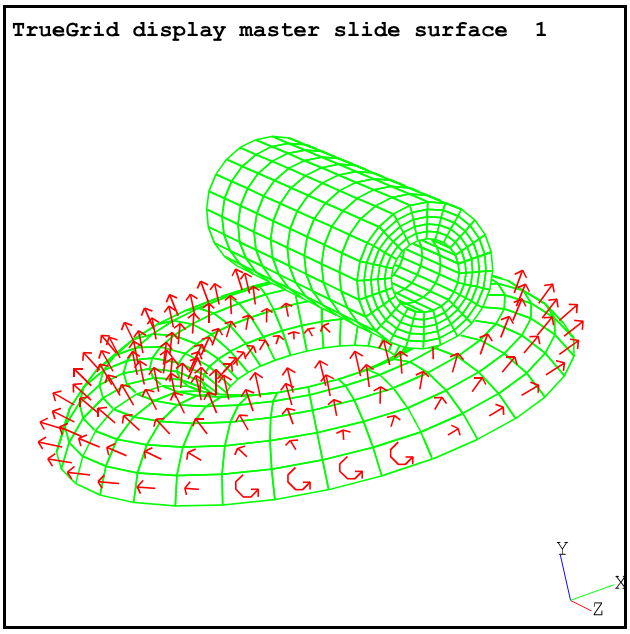


Figure 4 master side of interface

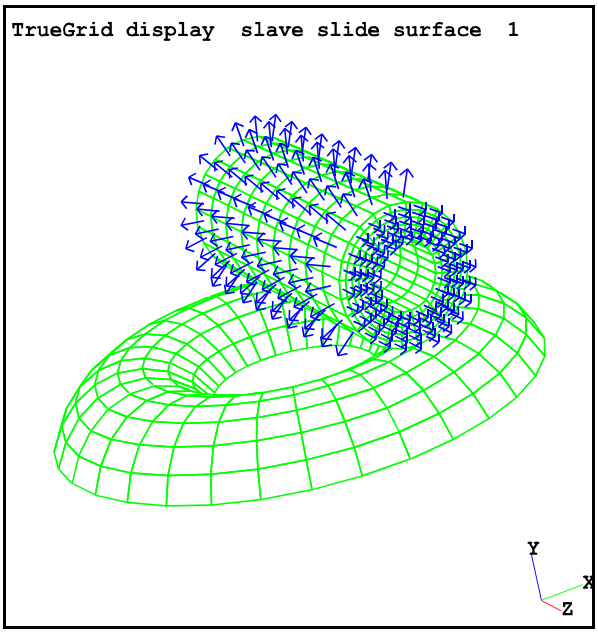


Figure 5 slave side of interface

Remarks

The *options* for the **si** and **sii** commands are not used for DYNA3D.

spd define the properties of a set of springs or dampers

spd *spring/damper_# type parameters*
where *type* is the spring or damper's material model and
parameters is one of the following:

le *stiffness*	linear elastic
lv *damping*	linear viscous
iep *elastic tangent yield*	isotropic elastic
ne *ld_curve_#*	nonlinear elastic
nv *ld_curve_#*	nonlinear viscous
nesf *ld_curve_# force_curve_#*	nonlinear elastic w/ force load curve
gn *loading_# unloading_# hardening tension compression*	general nonlinear

Remarks

A spring or damper is defined using either the **spdp** command forming a set of springs/dampers between two surfaces, or using the **spring** command to create a single spring at a time. In each case, the definition of a spring includes a reference to a material definition **spd** number.

DYNA3D, use linear elastic, linear viscous (damper), isotropic elastoplastic, nonlinear elastic, nonlinear viscous, general tabulated nonlinear, and dashpot.

If the output option has been selected prior to using the dialogue box to make a selection, only the options available to that output option will be displayed in the dialogue box.

dynaeos DYNA3D equation of state

dynaeos *material_# eos_type parameters_list* ;

where the *eos_type* can be

1	linear polynomial
2	JWL
3	Sack
4	Gruneisen
5	ratio of polynomials
6	linear polynomial with energy deposition
7	ignition and growth of reaction in HE
8	tabulated model with compaction
9	tabulated
11	pore collapse

where the *parameters_list* for EOS linear polynomial (type 1) is:

c0 *constant*
c1 *coefficient*
c2 *coefficien*t
c3 *coefficient*
c4 *coefficient*
c5 *coefficient*
c6 *coefficient*
e0 *energy*
v0 *volume*

where the *parameters_list* for EOS JWL (type 2) is:

a *constant*
b *constant*

r1 *constant*
r2 *constant*
omega *constant*
e0 *energy*
v0 *volume*

where the *parameters_list* for EOS Sack (type 3) is:

a1 constant
a2 constant
a3 constant
b1 constant
b2 constant
b0 constant
v0 constant

where the *parameters_list* for EOS Gruneisen (type 4) is:

vci intercept
s1 coefficient
s2 coefficient
s3 coefficient
gamma coefficient
sa coefficient
b0 energy
v0 volume

where the *parameters_list* for EOS ratio of polynomials (type 5) is:

a10 constant
a11 constant
a12 constant
a13 constant
a20 constant
a21 constant
a22 constant
a23 constant
a30 constant
a31 constant
a32 constant
a33 constant
a40 constant
a41 constant
a42 constant
a43 constant

a50 constant
a51 constant
a52 constant
a53 constant
a60 constant
a61 constant
a62 constant
a63 constant
a70 constant
a71 *constant*
a72 *constant*
a73 *constant*
alpha *constant*
beta *constant*
a14 *constant*
a24 *constant*
e0 *energy*
v0 *volume*

where the *parameters_list* for EOS linear polynomial with energy deposition (type 6) is:

c0 *constant*
c1 *coefficient*
c2 *coefficient*
c3 *coefficient*
c4 *coefficient*
c5 *coefficient*
c6 *coefficient*
e0 *energy*
v0 *volume*
lc *load_curve*

where the *parameters_list* for EOS ignition and growth of reaction in HE (type 7) is:

ap *constant*
bp *constant*
r1p *constant*
r2p *constant*
g *coefficient*
wpcp *constant*
ae *constant*
be *constant*
wece *constant*
r1e *constant*

r2e *constant*
fcrit *fraction*
i *coefficient*
h *coefficient*
z *exponent*
x *exponent*
y *exponent*
cp *heat_capacity*
ce *heat_capacity*
m *exponent*
e0 *energy*
t0 *temperature*

where the *parameters_list* for EOS tabulated model with compaction (type 8) is::
eps *list_strains* ;
pc *list_constants* ;
t *list_temperatures* ;
ku *list_modulus* ;*compression*
gamma *gamma*
e0 *energy*
v0 *volume*

where the *parameters_list* for EOS tabulated (type 9) is:
eps *list_strains* ;
pc *list_constants* ;
t *list_temperatures* ;
gamma *gamma*
e0 *energy*
v0 *volume*

where the *parameters_list* for EOS pore collapse (type 11) is:
mu1 *compression*
mu2 *compresion*
e0 *energy*
mu0 *compression*
virgin *load_curve_pairs ;*
crushed *load_curve_pairs ;*

NOTE: The following EOS models are under development

where the *parameters_list* for EOS Ignition and Growth of Reaction in HE 3-Term (type 13) is:

where the *parameters_list* for EOS Self-Generated Table with Compaction (type 14) is:
(No arguments)

Remarks

The material in this command assigns this equation of state to a specific material. An equation of state is not required by DYNA3D. When using an equation of state, be sure to choose a material, using the **dynamats** command, that supports an equation of state.

You must be in the control phase to issue this command. When you first start **True*Grid*®** you are in the control phase and it is convenient to issue this command at that time. However, one can return to the control phase anytime using the **control** command.

Example

```
dynaeos 1 4 vci 159634 s1 19400 s2 -1992.2 s3 92.33
            gamma 1.69 sa 0.9976 b0 0 v0 1.0 ;
```

dynamats DYNA3D materials

dynamats *material_# material_type options_lis*t *properties_list* ;
where the following *options* are available for all materials:

mhead *text* comment w/ max. of 80 characters (to end of line)

shell *features_list*
with the following *features*

elfor *option*
where the *option* can be

hl	for Hughes-Lui shell
bt	for Belytschko-Tsay shell
bciz	for triangular shells
c0	for triangular shells
membrane	
yase	

shear *factor*

tsti *#_points* for through shell thickness integration

propt *option* for print out options
where *option* can be

1	for element center
2	for plan integration points
3	for through thickness and plan integration points

quad *integration_rule_#*

where the *integration_rule_#* can be

n	positive for the number of points using the trapezoidal rule
0	Gauss
-n	negative of the user specified rule number (from **sind** command)

shth *thickness* default shell thickness
shth1 *thickness* default shell thickness at the first node
shth2 *thickness* default shell thickness at the second node
shth3 *thickness* default shell thickness at the third node
shth4 *thickness* default shell thickness at the fourth node
shloc *location* through thickness location of the shell

where *location* can be

1	top surface
0	middle surface
-1	bottom surface

beam *features_list*

with the following *features*

elfom *option*

where the *option* can be

hl	Hughes-Lui beams
bt	Belytschko-Tsay beams
truss	

shear *factor*
quad *option*

where the *option* can be

1	for a truss
2	for 2x2 Gauss quadrature
3	for 3x3 Gauss quadrature
4	for 3x3 Lobatto integration
5	for 4x4 Gauss quadrature

bmcross *shape*

where the *shape* can be

0	for rectangular
1	for tubular

sthi *thickness*
tthi *thickness*
sthi1 *thickness*
sthi2 *thickness*
tthi1 *thickness*
tthi2 *thickness*
sloc *location*

where *location* can be

1 meaning the side where s is 1
0 meaning centered
-1 meaning the side where s is -1

tloc *location*
where *location* can be
1 meaning the side where t is 1
0 meaning centered
-1 meaning the side where t is -1

tshell *features_list*
with the following *features*
shear *shear*
tsti *#_points*
quad *integration_rule_#*

rho *density*

where the *properties_list* is specific to the selected material type:

Experimental Material Model - Material type 0

p*ij value* repeat as often as is needed
where *i* can be from 3 to 8 (index to the record number of the material definition)
where *j* can be from 1 to 8 (index to the field number of the material definition)

Elastic - Material type 1

e *modulus*
pr *ratio*

Orthotropic Elastic - Material type 2

ea *ea*
eb *eb*
ec *ec*
prba *vba*
prca *vca*
prcb *vcb*
gab *gab*
gbc *gbc*
gca *gca*
aopt *option parameters* for material orientation
where the *option* can be one of
0 for by nodes
1 for by point and element center
2 for by normal vectors
3 for by cross product with shell normal (shell elements only)
where the *parameters* can be

xp *x-coordinate* for **aopt 1**
yp *y-coordinate* for **aopt 1**
zp *z-coordinate* for **aopt 1**
ax *x-component* for **aopt 2**
ay *y-component* for **aopt 2**
az *z-component* for **aopt 2**
dx *x-component* for **aopt 2**
dy *y-component* for **aopt 2**
dz *z-component* for **aopt 2**
vx *x-component* for **aopt 3**
vy *y-component* for **aopt 3**
vz *z-component* for **aopt 3**
beta *angle*

Kinematic/Isotropic Elastic-Plastic - Material type 3

e *modulus*
pr *ratio*
sigy *stress*
etan *modulus*
beta *parameter*

Thermo-Elastic-Plastic - Material type 4

temp *temperature_list* ;
e *modulus_list* ;
pr *ratio_list* ;
alpha *secant_list* ;
sigy *stress_list* ;
etan *modulus_list* ;

Soil And Crushable Foam - Material type 5

g *modulus*
ku *modulus*
a0 *yield*
a1 *yield*
a2 *yield*
pc *pressure*
vs *strain_list* ;
p *pressure_list* ;

Viscoelastic - Material type 6

k *modulus*
g0 *modulus*

gi *modulus*
beta *constant*
mflag *option*
where the *option* can be

0	which means that beta is the delay constant
1	which means that beta is the time relaxation constant

Blatz-Ko Hyperelastic Rubber - Material type 7
g *modulus*

High Explosive Burn - Material type 8
d *velocity*
pcj *pressure*

Fluid - Material type 9
pc *pressure*
mu *coefficient*

Isotropic-Elastic-Plastic-Hydrodynamic - Material type 10
g *modulus*
sigy *stress*
etan *modulus*
pc *pressure*
a1 *coefficient*
a2 *coefficient*
ispall *model*
where the *model* can be

pl	for pressure limit
max	for maximum principal stress spall criterion
hydro	for hydrostatic tension spall criterion
eps *strain_list* ;	up to 16 values
es *stress_list* ;	up to 16 values

Steinberg-Guinan High Rate Elastic-Plastic - Material type 11
g0 *modulus*
sig0 *stress*
beta *constant*
n *exponent*
gama *strain*
sigm *stress*
b *modulus*
bpm *stress*

h *coefficient*
f *exponent*
t0 *temperature*
gam0 *gamma*
sa *constant*
pc *pressure*
ispall *model*
where the *model* can be

pl	for pressure limit
max	for maximum principal stress spall criterion
hydro	for hydrostatic tension spall criterion

a *atomic_weight*
r *r_prime*
spall
ivar *option*
where the *option* can be

0	for cold compression polynomial coefficient in eta
1	for cold compression polynomial coefficient in mu

ec0 *coefficient*
ec1 *coefficient*
ec2 *coefficient*
ec3 *coefficient*
ec4 *coefficient*
ec5 *coefficient*
ec6 *coefficient*
ec7 *coefficient*
ec8 *coefficient*
ec9 *coefficient*

Isotropic-Elastic-Plastic - Material type 12
g *modulus*
sigy *stress*
eh *modulus*
k *modulus*

Elastic-Plastic With Failure - Material type 13
g *modulus*
sigy *stress*
eh *modulus*
fs *strain*
fp *pressure*
k *modulus*

Soil And Crushable Foam With Failure - Material type 14

g *modulus*
ku *modulus*
a0 *constant*
a1 *constant*
a2 *constant*
pf *pressure*
iflag *flag*
where the *flag* can be

0	for hydrostatic tension
1	for maximum principal stress

sigmaf *stress*
vs *strain_list* ;
ps *pressure_lis* ;

Johnson/Cook Elastic-Plastic - Material type 15

g *modulus*
a *stress*
b *coefficient*
n *exponent*
sc *coefficient*
m *exponent*
tm *temperature*
tr *temperature*
x0 *rate*
sh *heat*
ispall *model*
where the *model* can be

pl	for pressure limit
max	for maximum principal stress spall criterion
hydro	for hydrostatic tension spall criterion

iter *flag*
where the *flag* can be

0	for fast approximate solution
1	for accurate iterative solution

d1 *parameter*
d2 *parameter*
d3 *parameter*
d4 *parameter*
d5 *parameter*
e *modulus*
pr *ratio*

dtcrit *step_size*

Concrete/Geological Model - Material type 16

pr *ratio*
g *modulus*
sigy *stress*
a0 *cohesion*
a1 *coefficient*
a2 *coefficient*
b1 *factor*
a0f *cohesion*
a1f *coefficient*
r *percent*
emr *modulus*
prr *ratio*
sigma0 *stress*
tm *modulus*
lc *load_curve*
lcr *load_curve*
eps *list_strain* ;
es *list_stress* ;
p *list_pressure* ;

Isotropic Elastic-Plastic With Oriented Crack - Material type 17

e *modulus*
pr *ratio*
sigy *stress*
eh *modulus*
fs *strength*
pc *pressure*

Power Law Isotropic Elastic-Plastic - Material type 18

e *modulus*
pr *ratio*
k *coefficient*
n *exponent*

Strain Rate Dependent Isotropic Elastic-Plastic - Material type 19

e *modulus*
pr *ratio*
lcs0 *load_curve*
etan *modulus*

lce *load_curve*
lce *load_curve*
lcfs *load_curve*
tss *step_size*

Rigid - Material type 20

e *modulus*
pr *ratio*
bpm *options* ;
where an *option* can be
dof *flag*
where *flag* can be

1	x-translational degree-of-freedom
2	y-translational degree-of-freedom
3	z-translational degree-of-freedom
4	translational motion in the given vector direction (use v below)
5	x-rotational degree-of-freedom
6	y-rotational degree-of-freedom
7	z-rotational degree-of-freedom
8	rotational motion about the given vector (use v below)

lcid *load_curve_#*
sf *scale_factor*
v x_0 y_0 z_0
rbv *load_curve amplitude fx fy fz* (obsolete)

Thermal Orthotropic Elastic - Material type 21

ea *modulus*
eb *modulus*
ec *modulus*
prba *ratio*
prca *ratio*
prcb *ratio*
alpa *coefficient*
alpb *coefficient*
alpc *coefficient*
gab *modulus*
gbc *modulus*
gca *modulus*
aopt *option parameters* for material orientation
where the *option* can be one of

0	for by nodes

1 for by point and element center
2 for by normal vectors
3 for by cross product with shell normal (shell elements only)

where parameters can be

xp *x-coordinate* for **aopt 1**
yp *y-coordinate* for **aopt 1**
zp *z-coordinate* for **aopt 1**
ax *x-component* for **aopt 2**
ay *y-component* for **aopt 2**
az *z-component* for **aopt 2**
dx *x-component* for **aopt 2**
dy *y-component* for **aopt 2**
dz *z-component* for **aopt 2**
vx *x-component* for **aopt 3**
vy *y-component* for **aopt 3**
vz *z-component* for **aopt 3**
beta *angle*

Fiber Composite With Damage - Material type 22

ro *density*
ea *modulus*
eb *modulus*
ec *modulus*
k *modulus*
sn *strength*
syz *strength*
szx *strength*
prba *ratio*
prcb *ratio*
prca *ratio*
gab *modulus*
gbc *modulus*
gca *modulus*
aopt *option parameters* for material orientation

where the *option* can be one of

0 for by nodes
1 for by point and element center
2 for by normal vectors
3 for by cross product with shell normal (shell elements only)

where the *parameters* can be

xp *x-coordinate* for **aopt 1**

yp *y-coordinate* for **aopt 1**
zp *z-coordinate* for **aopt 1**
ax *x-component* for **aopt 2**
ay *y-component* for **aopt 2**
az *z-component* for **aopt 2**
dx *x-component* for **aopt 2**
dy *y-component* for **aopt 2**
dz *z-component* for **aopt 2**
vx *x-component* for **aopt 3**
vy *y-component* for **aopt 3**
vz *z-component* for **aopt 3**

axes *flag*
where the *flag* can be
1 for the default
2 for switch material axes a and b
3 for switch material axes a and c

sc *strength*
xt *strength*
yt *strength*
yc *strength*
alpha *parameter*
beta *list_angles* ;

Thermal Orthotropic Elastic With Variable Properties - Material type 23

ea *ea_list* ;
eb *eb_list* ;
ec *ec_list* ;
vba *vba_list* ;
vca *vca_list* ;
vcb *vcb_list* ;
aa *aa_list* ;
ab *ab_list* ;
ac *ac_list* ;
gab *gab_list* ;
gbc *gbc_list* ;
gca *gca_list* ;
t *temperature_list* ;
angles *angle_list* ;
aopt *option parameters* for material orientation
where the *option* can be one of
0 for by nodes
1 for by point and element center

2 for by normal vectors
3 for by cross product with shell normal (shell elements only)

where the *parameters* can be

xp *x-coordinate* for **aopt 1**
yp *y-coordinate* for **aopt 1**
zp *z-coordinate* for **aopt 1**
ax *x-component* for **aopt 2**
ay *y-component* for **aopt 2**
az *z-component* for **aopt 2**
dx *x-component* for **aopt 2**
dy *y-component* for **aopt 2**
dz *z-component* for **aopt 2**
vx *x-component* for **aopt 3**
vy *y-component* for **aopt 3**
vz *z-component* for **aopt 3**

Rate-Dependent Tabular Isotropic Elastic-Plastic - Material type 24

e *modulus*
pr *ratio*
sigy *stress*
et *modulus*
efp *strain*
dtcrit *time*
lc *load_curve*
eps *strain*
es *stress*

Extended Two Invariant Geologic Cap - Material type 25

k *modulus*
g *modulus*
alpha *parameters*
theta *coefficient*
gamma *coefficient*
beta *exponent*
r *ratio*
d *exponent*
w *coefficient*
x0 *parameter*
cbar *coefficient*
n *parameter*
nplot *option*

where the *options* are

1 hardening variable, k
2 cap - j1 axis intercept, x(k)
3 volumetric plastic strain
4 first stress invariant, j1
5 second stress invariant, square root of j2
8 response mode number
9 number of iterations

ltype *option*
where the *options* are
1 soil or concrete (cap surface may contract)
2 rock (cap surface does not contract)

ivec *option*
where the *options* are
0 vectorization (fixed number of iterations)
1 fully iterative

t *cutoff*

Metallic Honeycomb - Material type 26

e *modulus*
pr *ratio*
sigy *stress*
sigaa *load_curve*
sigbb *load_curve*
sigcc *load_curve*
ssrv *load_curve*
crv *volume*
ea *modulus*
eb *modulus*
ec *modulus*
gab *modulus*
gbc *modulus*
gca *modulus*

Compressible Mooney-Rivlin Hyperelastic Rubber - Material type 27

a *coefficient*
b *coefficient*
pr *ratio*

Resultant Plasticity - Material type 28

e *modulus*
pr *ratio*
sigy *stress*

etan *modulus*

Closed-Form Update Elastic-Plastic For Shells - Material type 30

e *modulus*
pr *ratio*
sigy *stress*
etan *modulus*

Frazer-Nash Hyperelastic Rubber - Material type 31

g001 *coefficient*
g010 *coefficient*
g020 *coefficient*
g100 *coefficient*
g101 *coefficient*
g110 *coefficient*
g200 *coefficient*
g210 *coefficient*
g300 *coefficient*
g400 *coefficient*
ilimit *option*
where the *option* can be

0	to stop if strain limits are exceeded
1	to continue if strain limits are exceeded

stmx *strain*
stmn *strain*

Ramberg-Osgood Elastic-Plastic - Material type 32

gammay *strain*
tauy *stress*
alpha *coefficient*
r *exponent*
k *modulus*

General Anisotropic Elastic-Plastic - Material type 33

ea *modulus*
eb *modulus*
ec *modulus*
r *coefficient*
acp *coefficient*
qbc *coefficient*
qab *coefficient*
qac *coefficient*

prba *ratio*
prca *ratio*
prcb *ratio*
aopt *option parameters* for material orientation
where the *option* can be one of
0 for by nodes
1 for by point and element center
2 for by normal vectors
3 for by cross product with shell normal (shell elements only)
where the *parameters* can be
xp *x-coordinate* for **aopt 1**
yp *y-coordinate* for **aopt 1**
zp *z-coordinate* for **aopt 1**
ax *x-component* for **aopt 2**
ay *y-component* for **aopt 2**
az *z-component* for **aopt 2**
dx *x-component* for **aopt 2**
dy *y-component* for **aopt 2**
dz *z-component* for **aopt 2**
vx *x-component* for **aopt 3**
vy *y-component* for **aopt 3**
vz *z-component* for **aopt 3**
sigya *stress*
beta *angle*
eap *modulus*
gbc *modulus*
gab *modulus*
gac *modulus*
npss *substeps*
epsap *list_strain* ;
sigmaya *list_stress* ;

Normal Anisotropic Elastic-Plastic For Shells - Material type 34
e *modulus*
pr *ratio*
sigy *stress*
etan *modulus*
r *parameter*

Elastic-Plastic With Forming Limit Diagram - Material type 35
e *modulus*
ifld *option*

where the *option* can be

1	for total
2	for incremental
3	for damage

pr *ratio*
lclh *load_curve*
lcrh *load_curve*
lcpx *load_curve*
lcedf *load_curve*
lcedm *load_curve*
sigy *stress*
scldev *factor*
etan *modulus*
beta *parameter*
ep *list_strain* ;
sigmay *list_stress* ;

Brittle Damage Model - Material type 36
e *modulus*
pr *ratio*
ft0 *limit*
fs0 *limit*
gc *toughness*
beta *retention*
nu *viscosity*

NOTE: The following material models are under development.

Three-Invariant Viscoplastic Cap Model - Material type 37

Bammann Plasticity Model - Material type 38

Snadia Damage Model - Material type 39

Fahrenthold Brittle Damage - Material type 40

Fabric with Damage - Material type 41

Multi-Material Shell Element Model - Material type 42

Transversely Isotropic Visco-Hyperelasticity - Material type 43

Rigid Foam - Material type 44

DTRA Concrete/Geological - Material type 45

Anisotropic Elastic - Material type 46

MIG - Material Interface Guide - Material type 47

Visco-Elastic with Statistical Crack Mechanics - Material 48

LANL Hyperfoam - Material type 49

Braided Composite Model with Damage - Material type 50

Uni-Directional Elasto-Plastic Composite - Material type 56

Uni-Directional Elasto-Plastic Composite - Material 62

Visco-Hyper Elastic - Material type 63

Steinberg-Guinan with 3-D Failure - Material 64

K&C Concrete/Geological - Material type 65

Brittle Damage Model with Power-law Plasticity - Material 70

Delamination Element Material Model

Cohesive Element Material Model

Remarks

You must be in the control phase to issue this command. When you first start **True*Grid*®** you are in the control phase and it is convenient to issue this command at that time. However, one can return to the control phase anytime using the **control** command.

Example

```
para
c sheet metal
  e_stl      200.0e3      c modulus of elasticity for steel
  v_stl        0.33       c poisson ratio for steel
  yld_stl    207          c yield stress for steel
```

```
  tan_stl    200          c tangent modulus for steel
  den_stl      7.86e-9 ;  c density of steel
dynamats 12 3  shell
  e %e_stl pr %v_stl sigy %yld_stl etan %tan_stl
  beta 0. rho %den_stl tsti 3;
dynamats 18 3 beam
  e %e_stl pr %v_stl rho %den_stl sigy %yld_stl
  etan %tan_stl beta 0.
  bmcross 1 elfom hl sthi 25 tthi 10 quad 3 sloc 0  tloc 0 ;
dynamats 9 1 e 1 pr 0.35 ;
```

dynaopts DYNA3D analysis options

dynaopts *options*

where any or all of the following *options* can be invoked:

iif *interval* — time interval between writes of the interface segment save file

stsm *factor* — minimum time step size for thin shell element using the materials kinematic/isotropic elastic-plastic, strain rate dependent elastic-plastic, or rate dependent tabular isotropic elastic-plastic

rfpf *flag* — set the reaction force print flag

where the *flag* can be

- **0** — no printing
- **1** — reactions are printed

defpf *flag* — set the discrete (lumped parameter) element forces print flag

where the *flag* can be

- **0** — no printing
- **1** — forces in all elements are written

edsdf *flag* — set the element delete (SAND database flag)

where the *flag* can be

- **0** — failed elements not deleted
- **1** — failed elements deleted
- **2** — triggers a debug run
- **10** — same as 0 except failed shells not printed
- **11** — same as 1 except failed shells not printed
- **12** — same as 2 except failed shells not printed

gvst a*cceleration direction mat_1 mat_2 ...* ; *density_1 depth_1 density_2 depth_2 ...* ; — initialize gravity stress

stat *interval* — number of steps between problem status reports (default 1000)

yldb *yield* — brode function yield in ktons

hiteb *distance* — brode function height of burst

xb0 *x* model x-coordinate of brode origin

yb0 *y* model y-coordinate of brode origin

zb0 *z* model z-coordinate of brode origin

tb0 *time* model time of brode time origin

lcb1 *load_curve* brode function arrival versus range load curve relative to brode origin

lcb2 *load_curve* brode function yield scaling versus time load curve relative to brode

clb *factor* brode function conversion factor kft to DYNA3D length units

ctb *factor* brode function conversion factor milliseconds to DYNA3D time units

cpb *factor* brode function conversion factor psi to DYNA3D pressure units

ticsf *interval* time interval between output cross section forces

ngrav *x_acceleration load_curve* prescribed base acceleration
y_acceleration load_curve
z_acceleration load_curve

xvel *x_velocity load_curve* angular velocity about the x-axis

yvel *y_velocity load_curve* angular velocity about the y-axis

zvel *z_velocity load_curve* angular velocity about the z-axis

term *time* termination time

prti *interval* time interval between writes of time history node and element print block plot data

plti *interval* time interval between writes of state plot database for all nodes and elements and, optionally, the interface force database containing pressures and shear traction for all sliding interfaces

nrest *time_step* number of time steps between writes of the saved restart files

nrunr *time_step* number of time steps between writes of the continuously overwritten running restart file

itss *time_step* initial time step size

pnlt *factor* global scale factor for sliding interface penalty stiffness controlling inter-penetration and stability (default 0.1)

teo *load_curve* thermal effects option

where a non-positive value for *load_curve* means

- **0** no thermal effects
- **-1** nodal temperatures in TOPAZ3D generated plot files
- **-2** nodal temperatures use temperature option 2
- **-3** TOPAZ3D coupling
- **-4** nodal temperatures use temperature option2; subsequent temperature history are defined in TOPAZ3D generated plot files.

-9999	nodal temperatures use temperature option 1
tssf *factor*	time step scale factor (default is 0.67 for high explosives, 0.90 otherwise)
lcmax *load_curve*	load curve number that limits maximum time step size (optional)
ssdm	write shell strain tensor at inner and outer surface
snrs *option*	Hughes-Liu shell normal update option

where *option* is the number of time steps between computations or

-2	unique nodal fibers
-1	compute normals each time step
1	compute on restarts
n	compute every *n* steps
stup	shell thickness change due to membrane straining
sfor *option*	default shell element formulation

where the *option* can be

hl	Hughes-Liu
bt	Belytschko-Tsay
bciz	triangle
c0	triangle
membrane	
yase	
yase2	YASE with full in-plane integration
bd	Bath-Dvorkin (full integration)
blt	Belytschko-Lin-Tsay with selective-reduced integration
tsmin *factor*	reduction factor for initial time step size to determine minimum time step size. When this minimum time step is reached, DYNA 3D terminates with a restart dump.
itrx *#_iterations*	number of iterations between convergence checks for dynamic relaxation in quasi-static problems (default 250)
tolrx *tolerance*	convergence tolerance for dynamic relaxation option (default 0.0001)
facrx *factor*	dynamic relaxation static analysis velocity reduction factor (default 0.995)
scftrx *factor*	scale factor for computed time step during dynamic relaxation
stss *option*	alternative methods for approximating the maximum stable time step size for 4-node shell elements

where the *option* can be

0	characteristic length is area/longest side
1	characteristic length is area/longest diagonal
2	based on bar wave, shortest side, area/longest side
plas *option*	plane stress constitutive integration algorithms for elastoplastic shell material models

where the *option* can be

1	iterative plasticity with 3 secant iterations
2	full iterative plasticity
3	stress scaling non-iterative plasticity

prtflg print element time step sizes on the first cycle

drdb write the taurus database at every convergence check during dynamic relaxation

rayl *alpha* global generalized Rayleigh damping mass proportional coefficient

ihq *option* select an hourglass stabilization method

where the *option* can be

1	standard DYNA3D (viscous form)
2	Flanagan-Belytschko (viscous form)
3	Flanagan-Belytschko with exact volume integration (viscous form)
4	stiffness with Flanagan-Belytschko
5	stiffness with Flanagan-Belytschko with exact volume
6	bricks: selective-reduced 8-point hexahedral element (B-bar) shells: viscous form type 2 and stiffness form type 3
7	bricks: physical stabilization (models 1, 40, 56 & 62 only)
8	total displacement physical stabilization (models 2, 7, 21, 23, 27, 31, 43, 46, 60 & 63 only)
9	bricks: physical stabilization - exact volume (models 1, 40, 56 & 62 only)
10	bricks: total displacement physical stabilization - exact volume (models 2, 7, 21, 23, 27, 31, 43, 46, 60 & 63 only)
12	bricks: fully integrated, 8-pt. hexahedral element

qh *coefficient* hourglass stabilization coefficient

q1 *coefficient* quadratic bulk viscosity coefficient for added stability and resolution with shock waves

q2 *coefficient* linear bulk viscosity coefficient for added stability and resolution with shock waves

NOTE: The following options are under development.

debug DYNA3D places various debug information into the plot files and terminates.

ndacc high accuracy coordinates for MILI plot data

drstep *steps* maximum time steps allowed in dynamic relaxation phase. Execution ends or switches to transient phase after this step limit is reached regardless of convergence.

drtime *time* — Maximum time allowed in dynamic relaxation phase. Execution ends or switches to transient phase after this time limit is reached regardless of convergence.

hgmodes *modes* — Select the number of enhanced modes used in the physically stabilized brick element hourglass control (default=3). *Modes* can either be 3 or 6. This feature only applies to material models 3, 4, and 18 when hourglass type 7 or 9 are used.

iforce — activate the interface force output

irestt *flag* — start time designation

where *flag* can be

- 0 — simulation start time is set to 0. This is used for most cases.
- 1 — simulation start time is set to the time recorded in the stress/deformation initialization file.
- 2 — same as 1, but also read the initial time step from the stress/deformation initialization file.

mili *file_name* — family name of MILI format output files

normck *flag* — check for consistent slide surface normals

where *flag* can be

- 0 — slide surface normals are not checked.
- 1 — consistency check is performed and warning is printed if inconsistent normals are detected
- 2 — consistency check is performed and program terminates if inconsistent normals are detected

numrrf *n* — Number of running restart data dumps maintained

pressure_display — Activate the external pressure display for segments listed in the Pressure Loads section

reaction_force_out — Activate the nodal reaction forces output into the MILI plot database.

rotational_vel_out — Activate the rotational velocity/displacement output

smp_dynamic *smp* — Assign number of SMP processors used statically or dynamically

where *smp* can be

- **0** — Static - always use *threads* number of processors during run
- **1** — Dynamic - use *threads* or less number of processor during run to instantaneously maximize use of system resources.

taurus_plot off — Disable the TAURUS state plot output.

threads *threads* — Number of shared-memory processors to use

zero_init_vol *vol* — On initialization, reset the relative volume of brick elements to unity when an element's initial volumetric strain (in absolute value) is less than *vol*

ptsming *ptsming* — Default time-step size factor for element deletion

where *ptsming* can be

0	Option inactive
greater than **0**	Elements with non-zero *ptsming* will be deleted when their time-step size becomes less than *ptsmin* times the initial global time-step size. Brick elements will also be deleted when their volume becomes non-positive.

tsming *tsming* Default time-step size for element deletion

where *tsming* can be

0	Option inactive
greater than **0**	Elements with non-zero *tsmin*s will be deleted when their time-step size becomes less than *tsmin*. Brick elements will also be deleted when their volume becomes non-positive.

ascii_output_file *filename*	Rename the ASCII output file filename
rigid_wall *irigid*	Print all rigid-wall normal forces to FORCES file

where *irigid* can be

0	No forces print to file
1	Forces printed to file

verbose_hsp	Generate verbose input and initialization data in hsp file
pencfile *file_name*	PENCRV3D input file in a DYNA3D-PENC RV 3D analysis.
penhnose *hnose*	define the PENCRV3D nose height
pen_off *off_time*	turn off PENCRV3D when the analysis time exceeds *off_time*
pennose *node_num*	Designate *node_num* as the nose tip of the projectile in a DYNA 3D PENC RV3D analysis.
pensym *sym*	fraction of the penetrator represented by the mesh
pentail *node_num*	centroid of the projectile tail in a DYNA3D PENCRV3D analysis
beamfile *file_name*	Name of the file that contains beam element definitions
brickfile *file_name*	Name of the file that contains hex element definitions
include* *file_name*	Name of new input file to be read from.
infree *file_name*	Name of the file that contains auxiliary keyword definitions.
loadcurvefile *file_name*	Name of the file that contains the load curve definitions
matfile *file_name*	Name of the file that contains material definitions
nikefile *file_name*	Name of the stress/deformation file to be created at the end of either a regular or restart analysis.
nodefile *file_name*	Name of the file that contains nodal definitions
pressurefile *file_name*	Name of the file that contains pressure definitions
shellfile *file_name*	Name of the file that contains shell element definitions
slidefile *file_name*	Name of the file that contains the sliding interface definitions
tshellfile *file_name*	Name of the file that contains thick shell element definitions
velofile *file_name*	Name of the file that contains the initial nodal velocity definitions

ptsmin_ex *ptsmin mat_list ;* Set *ptsmin* for all other material numbers except those in the list

ptsmin_in *ptsmin mat_list ;* Set *ptsmin* for materials in the list

tsmin_ex *tsmin mat_list ;* Set *tsmin* for all other material numbers except those in the list

tsmin_in *tsmin mat_list ;* Set *tsmin* for materials in the list

verbose_hsp Generate verbose input and initialization data in hsp file

rigid_wall *irigid* Print all rigid-wall normal forces to FORCES file

where *irigid* can be

0 No forces print to file (default).

1 Forces printed to file.

dump_times *llist_times ;* write a restart file after the analysis reaches a time in the list

dt_plot *dt_list ;* write a plotfile for each time step

fdt_plot *fdt_list ;* write a plotfile for each factor multiplied by the time step in **dt_plot**

group_velocity_ex *mat_list ;* Include in the MILI time-history and plot files the mass averaged velocity or this group of materials.

group_velocity_in *mat_list ;* Include in the MILI time-history and plot files the mass averaged velocity for this group of materials.

mat_sv_ex *mat_list ; name_list ;* Alter which material model dependent state variables and peak derived variables are included in the plotfile for all brick-element materials in the model except for the materials in the list

mat_sv_in *mat_list ; name_list ;* Alter which material model dependent state variables and peak derived variables are included in the plotfile for the materials in the list

max_mises Track the maximum value of the von Mises stress with time.

min_press Track the minimum value of the pressure with time.

max_press Track the maximum value of the pressure with time.

min_prin1 Track the minimum value of the 1st principle stress with time.

max_prin1 Track the maximum value of the 1st principle stress with time.

min_prin3 Track the minimum value of the 3rd principle stress with time.

max_prin3 Track the maximum value of the 3rd principle stress with time.

plotall Select all state variables for this material model to be included in plotfile for the current material number chosen.

plotno *state_variable_name* Exclude the variable from the plotfile for the current material chosen.

plotnone Exclude all state variables from the plotfile for the current material chosen.

nip_thickness *num* For elements with more than one integration point, determine the number of integration points data is output for. For hex elements, the *num* is limited to 1 or the actual number of integration points.

nip_inplane *num* For multiple in-the-plane integrated shell & beam elements, output data at *num* in-plane locations. The number *num* is limit to 1 or the actual number of in-plane integration points.

maxpfile *file_name mat_list ;* Write the maximum pressure for the materials in the list at the time history output frequency to a text file.

pencvmat *lst* Provide a list of materials that are included in the projectile in a DYNA3D PENCRV3D analysis. Listed materials must only be associated with elements that are part of the projectile.

where *lst* can be

- **list** *mat_list* ; List of materials must be provided.
- **exclude** *mat_list* ; List of excluded materials must be provided.
- **all** All materials are considered as part of the projectile.

cavity_expansion *load_curve* Initialize a new cavity expansion model and designate the pressure segments associated with the load curve as belonging to this model.

cavity0 *c0* First cavity expansion coefficient for the current layer

cavity1 *c1* Second cavity expansion coefficient for the current layer

cavity2 *c2* Third cavity expansion coefficient for the current layer

cavitycyl cylindrical cavity expansion idealization for the present C.E. model.

fscale *fscale* current layer

load_curve *lc_num start_time* load curve to scale the C.E. pressures.

normal *nx ny nz* normal of the target surface

pen_off *off_time* turn off this cavity expansion model when the analysis time exceeds *off_time*

pennose *node_num* tip node of the body in this cavity expansion model.

pentail *node_num* tail node of the body in this cavity expansion mode.

point_fs *px py pz* reference point lies on the upper surface of the target's top layer

sc_radius *rad* minimum spherical/cylindrical radius

spherical spherical cavity expansion idealization for the present C.E. model.

thickness *thick* thickness of the current layer

velocity_min *velmin* velocity cutoff used in calculating the C. E. pressures

velocity_trans *veltrans* transition velocity used in calculating the optional decay factor

closed_volume *law cutoff ld_curve pressure power plane node*
pressure loads associated with load curve are determined by the gas pressure in a closed volume.

where *law* can be

- **ideal_gas** The current pressure is computed

cutoff Negative pressure cutoff. Only positive pressure (compression) is allowed on associated element facets.

no_cutoff Both positive and negative pressures are admissible.

where *plane* can be

x Project the closed volume to a plane normal to the global x-axis for volume calculation.

y Project the closed volume to a plane normal to the global y-axis for volume calculation.

z Project the closed volume to a plane normal to the global z-axis for volume calculation.

any Project the closed volume to an arbitrary plane for volume calculation.

where *node* can be

any DYNA3D chooses the node to define the plane of projection.

n Plane of projection passes through node n. n must be a positive integer.

dam_base *direction coordinate* must be used ifWestergaard hydrodynamic pressure is present in the Pressure Loads section

where *direction* can be

+x

-x

+y

-y

+z

-z

free_surface *direction coordinate* must be used if hydro static/hydrodynam ic pressure is present in the Pressure Loads section

where *direction* can be

+x

-x

+y

-y

+z

-z

gravity *g* acceleration of gravity, *g*, for hydrostatic or hydrodynamic pressure calculation use.

hydro_density *rho* water (or other fluid) density for hydrostatic or hydrodynamic pressure calculation use

hydro_cutoff *switch* hydrodynamic pressure

where *switch* can be

on	the hydrodynamic pressure will be set to zero if the instantaneous base acceleration generates negative pressure according to the Westergaard's formula
off	default
shear_traction *ld_curve option*	Designate the pressure loads associated with Load Curve to be applied in a tangential direction instead of the segment normal direction.

where the tangential direction is controlled by *option*

local_r	tangential traction is applied in direction of local r-direction
local_s	tangential traction is applied in direction of local s-direction
aux_vector *x y z*	tangential traction is applied in a direction that is determined by the segment normal vector and an auxiliary vector.
aux_nodes *x y z*	tangential traction is applied in a direction that is determined by the segment normal vector and an auxiliary vector
weibull *mat*	Calculate the Weibull statistic probability of failure (volume based) for material
wistrength *value*	specify the strength value as for the current Weibull material definition.
wi**exponent** *m*	specify the exponent value as *m* for the current Weibull material definition.
wigamma *offset*	specify the optional offset value of for the current Weibull material definition.

Remarks

This commands sets parameters that are found in the control cards for the DYNA3D input file format.

You must be in the control phase to issue this command. When you first start **True*Grid*®** you are in the control phase and it is convenient to issue this command at that time. However, one can return to the control phase anytime using the **control** command.

All keywords that have one indentation from the left are keyword options to this command. Many of these keywords are followed by additional data. If the additional data is one of several options, those options are further indented. Each further indentation indicates a set of options to the keyword above the list.

For example, the **iif** option must be followed by a number that is the factor. The **sfor** option must be followed by one of **hl**, **bt**, **bciz**, **c0**, **membrane**, **yase**, **yase2**, **bd**, or **blt**.

This command can be issued to set just one option or it can continue with as many options as desired until this list of options is terminated with a semi-colon. This command can be issued as many times as desired.

Because of the complexity of this command, it is advised that you use the dialogue box to select the options you require. The execution of the dialogue box will produce a command that follows the syntax above. Since this command will automatically be saved in the session file for future reruns, one can use this description of the command to make modifications to the options without having to use the dialogue box interactively.

Example

```
dynaopts term 1. velocity 0. 0. 0. plti 1.
         prti 1. nrest 20000 nrunr 1000 ;
```

IV. Frequently Asked Questions

1. How do I create an output deck for DYNA3D?

Go to the merge phase, issue a merge command, such as stp, issue the dyna3d command followed by the write command.

V. INDEX

a 26, 35, 36, 42
Acc 6, 9
Accc 10
Accci 10
Acceleration 9, 10, 48
    prescribed 6
    view 7
Acci 10
Accs 10
Accsi 10
alpha 28, 33, 40, 41, 43
angles 40
Angular velocity 48
Anisotropic 11, 12
aopt 32, 38, 39, 44
Archive 7
Area 20
Arri 6
Automatic contact 22
ax 33, 39-41, 44
ay 33, 39-41, 44
az 33, 39-41, 44
b 5, 8, 26, 34, 36, 42
Base acceleration 48
Base accelerations 6
Bb 7
    make 7
Bciz 30, 49
be 28
beam 5, 12, 13, 31
    beam cross section 21
Beams 12
Belytschko-Schwer 20, 31
beta 28, 33, 34, 39-41, 44, 45
Bi 5, 8
Bind 5, 13
Block 5, 6
Block boundary 7
Bm 6, 12
    beam cross section 21
Bmcross 31
Body force
    material 7
Bold
    syntax 5
Boundary conditions 5, 8
    nonreflecting 6
    si 23
    view 7
Brick elements 6
Bricks 11
Brode 47, 48
Brode functions 6
Bsd 5, 13, 19
Bt 30, 31, 49
C0 . . . . . 30, 49
Carea 20
Clb 48
Co 7, 8
    si 23
Comment 7
Concentrated loads 9, 10
Condition 7
    si 23
Constitutive 49
Constraints 8
Contact Surface 5-7, 21
    1D 6
    faces 8
    gap 8
    graphics 8
    initial penetration 8
    merged nodes 8
    mesh density 8
    orientation 8
    penalty 48
    replication 8
    sets 8
    table 8
    type 7
    view 7
Contact surfaces

sii 23
Control cards 5, 56
Control phase 30, 46, 56
Convergence 49
Cpb 48
Cross section 5, 6, 11-13, 48
crushed 29
Csf 6
Ctb 48
Cylinder 5, 6
d 34, 41
Damper 13
Dampers 6
Damping 50
Data base 14
Database 14, 47-50
Default 21
Defpf 47
Delete 47
Detonation 6
Detp 6
Dialogue box
sid 22
spd 26
Displacement 10
Displacements
prescribed 6
view 7
Dist 6
Distributed loads 9, 10
Dom 6
Drdb 50
Drtflg 14
Dummy interface 22
Dyna3D 7, 26, 30, 47
Dynaeos 5, 26
Dynamats 30
beam cross section 21
Dynamic relaxation 49, 50
Dynaopts 5, 6, 14
analysis option 47
e 32, 33, 36, 38, 41-44
ea 32, 38-40, 42, 43
eb 32, 38-40, 42, 43
Edsdf 47
Elastoplastic 49
Element
delete 47
sand 47
Element forces 47
Element history 6
Element type 49
Elfom 31
Elfor 30
EOS
Energy deposition 26, 28
Gruneisen 26, 27
Ignition and growth 26, 28
JWL 26
Linear Polynomial 26, 27, 29
pore collapse 26, 29
Ratio of polynomials 26, 27
Sack 26, 27
tabulated 26, 29
tabulated w/ compaction 26, 29
Epb 6, 14
Equation of state 5
Experimental material 32
Extra nodes 6
Face sets 8
Facrx 49
Failure 6, 14
Fc 6, 9, 10
Fcc 9
Fcci 9
Fci 9
Fcs 9
Fcsi 9
Fd 6, 10
Fdc 10
Fdci 10
Fdi 10
Fds 10
Fdsi 10
Ffc 10
Fields, Richard 15

Flcd 6, 11
Fmom 10
Fn 6
Fni 6
Follower force 10
Follower forces 6
    view 7
Force 47, 48
Force output 6
Format 7
Frb 6, 10
Frbi 10
Friction 22
Fset 8
Fv 6, 9, 10
Fvc 9
Fvci 9
Fvi 9
Fvs 9
Fvsi 9
Fvv 9, 10
Fvvc 9
Fvvci 9
Fvvi 9
Fvvs 9
Fvvsi 9
g 28, 33-37, 41
gab 32, 39, 40, 42, 44
gama 34
Gravity 47
Gravity stress initialization 6
Grep 8
Gsii 8
Gvst 47
Hexahedral 11
History 48
Hiteb 47
Hl 30, 31, 49
Hour glass 50
Hughes-Lui 20, 31
Ibm 6, 12
    beam cross section 21
Ibmi 12
    beam cross section 21
Iff 14
Ihq 50
Iif 47
Indentation 56
Inertia 20
Initial velocity 6
    view 7
Integration 49
Integration points 12, 13
Integration rules 5
Interface saved segment 47
Interface saved segments 6
Irr 20
Iss 6, 20
Italicized
    syntax 5
Itrx 49
Itss 48
Itt 20
Jbm 6, 12
    beam cross section 21
Jbmi 12
    beam cross section 21
Jd 6, 13
Joints 6, 13
Jt 6, 13
Kbm 6, 12
    beam cross section 21
Kbmi 12
    beam cross section 21
Lb 6
Lcb1 48
Lcb2 48
Lcd 6, 11
Lcmax 49
Load curve 6
Load curves 11
Local constraints 8
Local system 8
Lrep 8
Lsii 8
Lsys 6

Make BB 7
Mass points 6
Mate 5, 11, 12
Material
coordinate system 11, 12
Materials 5
body force 7
rotation 6
max 34-36
Mbb 7
Mdep 6
Membrane 6, 30, 49
Merge 6
dummy interface 22
nodes 7
Merged nodes 8, 13, 14
Mesh density 8
Mns 8
Mom 6, 9, 10
Moments 9, 10
Momentum
deposition 6
Momi 9
Mpc 6, 13
Mt 5, 11, 12
Mti 5, 11, 12
Multiple constraints 6
Multiple point constraints 13
N 6, 12
shell orientation 5
Ndl 6, 9, 10
Ndli 9
Ngrav 48
NIST 15
Nodal constraints 5
view 7
Nodal forces 6
view 7
Nodal history 6
Node set 13
Node sets 8
Non-reflecting 8
Nonreflecting boundaries 6
view 7
Normal vector 12
Npb 6, 14
Npm 6, 13
Nr 6, 8
Nrest 48
Nri 6, 8
Nrunr 48
Nset 8, 13
Nseti 13
omega 27
Or 6, 11, 12
Orientation 8, 11, 12, 21
view 7
Orpt 8
Orthotropic 11, 12
Outout format 7
Parts 5
Plane 6, 8, 11
Plane strain 49
Plas 49
Plate 5
Plot data 48
Plti 14, 48
Pm 6, 13
Pnlt 48
Point mass 13
Point masses 6
Post processing 14
Pr 6, 9, 10
Pramp 6, 9, 10
Prescribed 6
Pressure 6, 9, 10
view 7
Pri 9
Print flag 47
Prism 11
Propt 30
Prtflg 50
Prti 14, 48
Pslv 8
Q1 50
Q2 50

Qh 50
quad 30-32
Rayl 50
Reaction force 47
Readmesh 7
Rebar
sliding 22
Remarks 19
Replication 8
Restart 49
Rfpf 47
Rho 32
Rigbm 6
Rigid bodies 6
inertia 6
Rigid body 13
Rigid walls 6
Rotation 6, 9, 10
material 6
Sand 47
Sarea 20
Scftrx 49
Set ID 11
Sfb 6
Sfor 49
Shared constraints 6, 13
view 7
shear 30-32
Shear area 20
shell 5, 30
Shell elements 49
Shell normal 49
Shell orientation 5
view 7
Shell strain 49
Shell thickness 5, 49
Shell type 49
Shells 11
Shloc 31
Shth 31
Shth1 31
Shth2 31
Shth3 31
Shth4 31
Si 5, 6, 8, 23
sid 22
Sid 6, 7, 21
with si and sii 22
Sii 5, 6, 8, 23
Siid 5
Sind 5, 12, 31
Single point constraintsts 6
view 7
Slide lines 21
Sliding boundary planes 6
view 7
Sliding interface 21
display 23
dummy interface 22
si 23
sii 23
Sliding Surface 5-7
1D 6
faces 8
gap 8
graphics 8
initial penetration 8
merged nodes 8
mesh density 8
orientation 8
penalty 48
replication 8
sets 8
table 8
type 7
view 7
Sloc 31
Snrs 49
Solid to shell 6
spall 35
Spd 6, 13, 25
Spdp 6, 13
spd 26
Spring 6, 13
properties 25
spd 26

Springs 6
Ssdm 14, 49
Ssf 12
Ssfi 12
Stabilization 50
stat 47
Statis reports 47
Sthi 19, 31
Sthi1 . . . . . 20, 31
Sthi2 . . . . . 20, 31
Stone wall 11
Stone walls 6
    view 7
Stp 7
    dummy interface 22
Strain 49
Stress 47
Stsm 47
Stss 49
Stup 49
Surface constraint 8
Sw 6, 11
Swi 6, 11
Syf 6
Symmetry 6, 8
    view 7
Symmetry w/ failure 6
    view 7
Syntax 56
T 7
    dummy interface 22
Tb0 48
Te 6, 13
Tei 6, 13
temp 6, 13, 33
Temperature 6, 13
    view 7
Teo 48
Tepro 6, 13
term 48
Termination 48
Tetrahedral 11
Th 5, 6, 12
Thermal effects 48
Thi 12
Thic 12
Thick shells 5, 12
Thickness 12
    beam 19
Ticsf 48
Tie-breaking 6
Tied contact 21
Tied nodes 6, 14
    view 7
Time history 6, 14, 48
Time interval 48
Time stamp 7
Time step 47, 49, 50
Time steps
    initial 48
    restart file 48
    saved restart file 48
Title 7
Tloc 32
tmm 7
Tolrx 49
Tp 7
    dummy interface 22
Truss 5, 20, 31
Tshell 32
Tsmin 49
Tssf 49
Tsti 30
Tthi 20, 31
Tthi1 . . . . . 20, 31
Tthi2 . . . . . 20, 31
US Dept. Commerce 15
Vacc 10
Vaccc 10
Vaccci 10
Vacci 10
Vaccs 10
Vaccsi 10
Variable acceleration 10
Variable thickness 12
Variable velocity 9, 10

Ve 6, 9
Vei 9
Velocities 9, 10
Velocity 6, 9, 48, 49
    prescribed 6
    view 7
virgin 29
Wedge 11
Write 7
Xb0 48
Xvel 48
Yase 30, 49
Yb0 48
Ylbd 47
Yvel 48
Zb0 48
Zvel 48