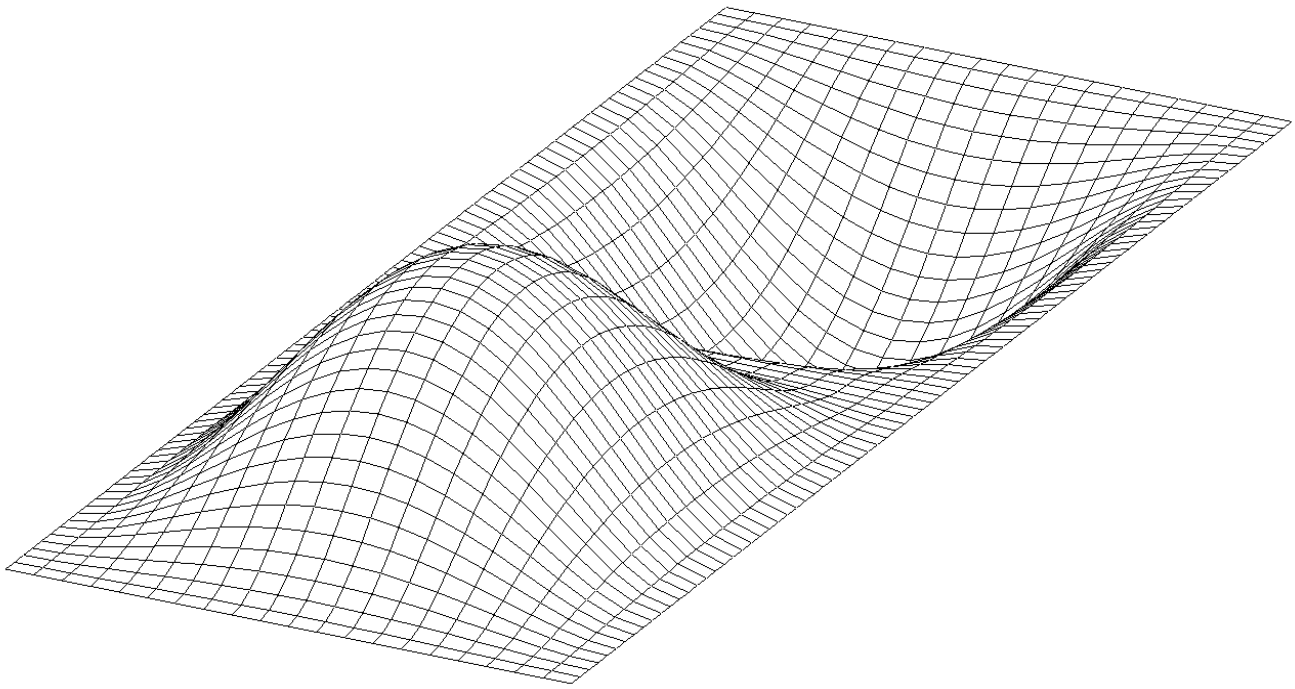
Autodesk Explicit 2021

User's Manual



AUTODESK.

User's Manual

Printed and bound in the United States of America.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 About This Manual

Autodesk Explicit is an explicit dynamics module that has been structured to use standard Nastran input files. For the most part, experienced Nastran users will find that there are very minor differences in the input file for a Autodesk Explicit run compared to a Nastran run. The differences mostly appear in the Case Control where we define parameters that are unique to the explicit dynamics module. Autodesk Explicit takes as standard Nastran input file with file extension of either ".nas" or ".NAS" and translates the Nastran input into two xml files that are used as input files for the explicit dynamics module. These two files are an input file with file extension ".inp.xml" and a mesh file with file extension (.tsm.xml). The mesh file contains all mesh definition (i.e. grid points, elements, node sets and element sets) while the input file contains all the model definitions such as boundary conditions, loads, material parameters, etc.

1.2 Running Autodesk Explicit

Autodesk Explicit is run by executing the file *aexp_app.exe*. The syntax for this along with the optional command line arguments are shown below. The *aexp_app.exe* should have the following lines preceding it:

```
cscript /nologo
```

This tells the script to output the information to the console. The entire line would appear as follows:

```
cscript /nologo  "C:\Program Files\Autodesk\Inventor Nastran
       2021\aexp_app.exe" -i inputfile.nas -dc

aexp_app.exe" -i inputFile -dc
```

The command line arguments are defined as follows:

`[-i inputFile]`	(REQUIRED) The name of an input file for the application. The input file should be a NASTRAN input file. The file extension ".nas" or ".NAS" is required.
`[-dc]`	(OPTIONAL) Data check flag: If this flag appears, the application will perform a data check run only. In this case the application will run the mesh translator through the point where the explicit module has read all of the input file, computed the stable time increment for the model, and checked the contact conditions. The explicit module will then stop without any time integration.

2. AUTODESK EXPLICIT FILE SYSTEM

2.1 Model Input

To illustrate the format of the input and output files, we have chosen the simply supported beam problem shown in Figure 2-1.

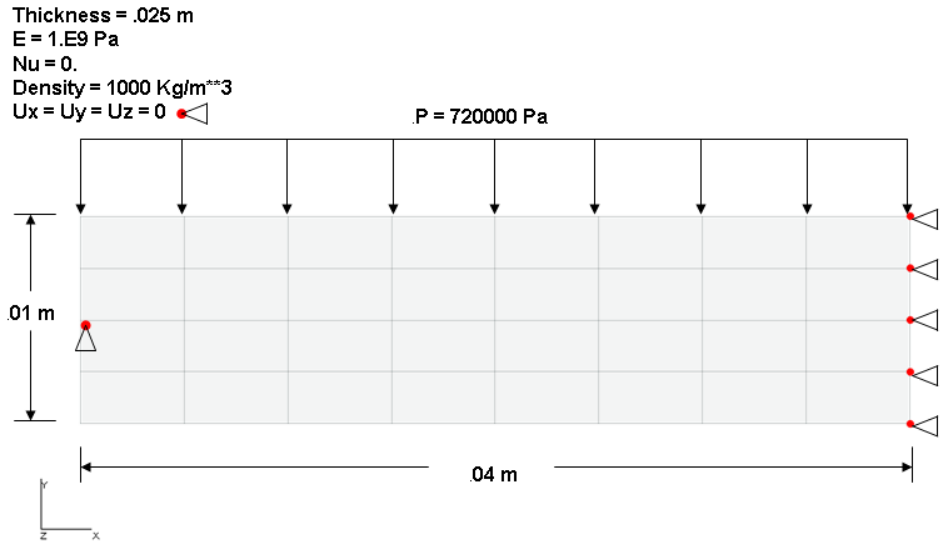


Figure 2-1. Simply Supported Beam Example Problem.

The structure represents a very simple model of a simply supported beam using 32 CHEXA elements. A suddenly applied pressure of magnitude 720000 is applied at time zero and the beam oscillates with an amplitude of twice the static displacement. The magnitude of the load is chosen so that the maximum amplitude is approximately the depth of the beam resulting in large displacement effects. This model was first proposed by Flanagan and Belytschko [1] to demonstrate the effectiveness of the hourglass control algorithm used for the uniform gradient element formulations.

Generating the Model Input File is the first step in performing an analysis using Autodesk Explicit. The Model Input File defines the structure's geometry, material properties, boundary conditions and loads. In addition, it specifies how the analysis is to be performed and what output is to be included in the Model Results Output File. The Model Input File is an 80 column ASCII text file and can be created using any text editor or one of the many pre-processors that interface with Autodesk Explicit.

The problem we are analyzing is shown in Figure 2-1 and the corresponding Autodesk Explicit Model Input File in Listing 2-1. Like most NASTRAN Model Input Files it can be divided into two distinct sections: the Case Control section and the Bulk Data section. Input in the Case Control section is referred to as a command and in the Bulk Data Section as an entry. The Case Control and Bulk Data sections must be assembled in the following sequence (BEGIN BULK and ENDDATA are required delimiters):

1. *Case Control Commands*

BEGIN BULK

2. *Bulk Data Entries*

ENDDATA

The Case Control section begins with the first command and ends with the command, BEGIN BULK. It defines the subcase structure for the problem, makes selections from the Bulk Data section, defines the output coordinate system for element and grid point results, and makes output requests for the Model Results Output File. Case Control commands are described in the Autodesk Explicit Reference Manual, Section 2, *Case Control*.

The Bulk Data section begins with the entry following BEGIN BULK and ends with the entry ENDDATA. It contains all of the details of the structural model and the conditions for the solution. BEGIN BULK and ENDDATA must be present even though no new bulk data is being introduced into the problem or all of the bulk data is coming from an alternate source, such as user-generated input. The format of the BEGIN BULK entry is free field. Generally speaking, only one structural model can be defined in the Bulk Data section. However, some of the bulk data, such as entries associated with loading conditions and constraints, may exist in multiple sets. Only sets selected in the Case Control section will be used in any particular solution. Bulk Data entries are described in the Autodesk Explicit Reference Manual, Section 3, *Bulk Data*.

Comments may be inserted in either section of the Model Input File. They are identified by a "$" in Column 1 with columns 2-72 containing any desired text. Comments may also be added to Case Control commands and free field Bulk Data entries with a "$" after the last character of data.

Once the generation of the Model Input File is complete, you can analyze your model by executing the Nastran command. See the *Autodesk Explicit Reference Manual*, Section 1, NASTRAN Command Line for more information.

2.1.1 Case Control

The Case Control section consists of commands which are used to:

1. Define the subcase structure for the analysis.
2. Select loads and constraints.
3. Define the contents of the Model Results Output File.
4. Define the output coordinate system for element and grid point results.

The Case Control section starts with the first line in the Model Input File and ends with the BEGIN BULK command. The output requests include displacement, velocity, acceleration, reaction forces, and applied forces at all grid points as well as stress and strain output for all elements in the model.

Reactions and stresses will be included for all grid points and elements. Displacements will be included by default.

The SPC command directs Autodesk Explicit to apply constraints defined by the SPC entry with an identification number (ID) of 1 in the Bulk Data section. The LOADSET command directs Autodesk Explicit to apply loading defined by the PLOAD4 entry with an ID of 1 in the Bulk Data section.

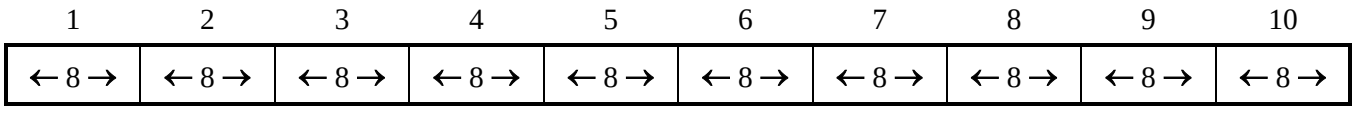
2.1.2 Bulk Data

The Bulk Data section defines your model by allowing you to specify geometry (grid points, element connectivity, etc.), material properties, boundary conditions (constraints) and loading (forces, moments, pressures, etc.). The start of the Bulk Data section is denoted by the BEGIN BULK delimiter and the end, the ENDDATA delimiter. Both delimiters are required.

The Case Control section has control over entries that describe boundary conditions and loading. In the cantilever beam example that would mean the SPC and PLOAD4 entries only. All other entries are always included in the model regardless of what the Case Control section specifies. This allows you to have multiple load cases and control what load cases are used for a given analysis. Constraint and load entries can exist in the model and not be called unless needed. In addition, material property and coordinate system entries can exist that are never referenced. An error message will result, however, if an element or grid point references a material property or coordinate system that does not exist.

In the beam example, the model's geometry is defined via the GRID entry. Each grid point coordinate is defined in the default basic coordinate system. You may pick whatever units you like as long as you are consistent. Element connectivity is defined via the CHEX8 entries. The material property is defined on the PSOLID entry. The isotropic material that the beam is made from is defined using the MAT1 entry.

Bulk data entries may be entered either in fixed or free field format. Free field format will be discussed at the end of this section. Fixed field format is divided into small and large field formats. Large field format can be used when small field does not provide enough significant digits. For small field format a data line is divided into 10 fields, each with eight characters as shown below:



The following is an example of small field fixed format:

1	2	3	4	5	6	7	8	9	10
GRID	100	20	1.0	10.5	0.0	17			

Large field format requires (at least) two lines for each entry: The first and last field of each data line contains eight characters and the four fields between contain 16 characters as shown below:

Line 1:

1A	2	3	4	5	6
← 8 →	← 16 →	← 16 →	← 16 →	← 16 →	← 8 →

Line 2:

1B	6	7	8	9	10B
← 8 →	← 16 →	← 16 →	← 16 →	← 16 →	← 8 →

The following is an example of large field fixed format:

Line 1:

1A	2	3	4	5	6
GRID*	100	20	1.0	10.5	*C0001

Line 2:

1B	6	7	8	9	10
*C0001	0.0	17			

Large field entries are denoted by an asterisk (*) immediately following the character string in field 1A of the first line and immediately preceding the character string in field 1B of the second line.

For all formats the name of the Bulk Data entry is input in field 1 beginning in column 1. Fields 2-9 are for data items. The only limitations in data items are that they must lie completely within the designated field, have no embedded blanks, and must be of the proper type (i.e., blank, integer, real, or character). All real numbers, including zero, must contain a decimal point. A blank will be interpreted as a real zero or integer zero, as required. Real numbers may be encoded in various ways. For example, the real number 7.0 may be encoded as 7.0, .7E1, 0.7+1, 70.-1, .70+1, 7+0, etc.

Field 10 of the Bulk Data entry is used for two purposes. If the Bulk Data entry does not have a continuation line, field 10 may be used as an optional comment field. A ";" is still required as the first character of the comment. If the Bulk Data entry has a continuation line, field 10 is used for the continuation identifier. The continuation contains the symbol + in column 1 followed by the same seven characters that appeared in columns 74-80 of field 10 of the entry that is being continued. Character strings used as continuation identifiers cannot contain the symbol "$" in column 1 or ";" in any column. The continuation identifier must be unique with respect to all the other identifiers in your Bulk Data. Continuation fields can also be generated automatically by Autodesk Explicit. To automatically generate a continuation, the continuation line (or lines) must immediately follow the parent Bulk Data entry. In addition, fields 1 and 10 of the continuation line (or lines) must be left blank.

Free field format provides an easier method for inputting data manually. In the free field format, commas are used to separate the fields. An entry in free field format is identified by a comma or equal sign in any of the first nine columns. The following rules apply to the use of the free field format:

- Free field data entries must start in column 1.
- A comma must separate data items.
- Data must be eight characters or less.
- If automatic continuation is to be used, the continuation line starts with a comma in field 1.

The following is an example of free field format:

```
GRID, 100, 20, 1., 10.5, , 17
```

The following is an example of free field format with automatic continuation.

```
CBAR, 10, 100, 201, 202, 0., 0., 1., ,
, , ,1., 0., 0., 1., 0., 0.
```

2.1.3 Output File

Autodesk Explicit writes an output file that contains a summary of the input data file and a "heart beat" history of the analysis as the time integration is performed. This heart beat consists of the energy balance for the model every 50 time increments. The file extension ".out" is used for the output file.

2.1.4 Selected Results

Autodesk Explicit writes an optional selected results file when directed to do so by the user. The selected results file contains fields defined over the entire mesh for nodal and element results. These are the results that you use to create deformed mesh images and contours of fields over the mesh. You define the selected results output by using keywords in the case options. FEMAP output files are used for the selected results file and the ".fno" will be used. The frequency of output to the selected results file is specified on the TSTEPNL option.

2.1.4.1 Nodal Output Choices

DISPLACEMENT	Write nodal displacements and rotations for every node in the mesh. This option is always included by default.
VELOCITY	Write nodal velocities and angular rotations for every node in the mesh.
ACCELERATION	Write nodal accelerations and angular accelerations for every node in the mesh.
REACTIONS	Write nodal reaction forces and reaction moments for the nodes in the mesh that have boundary conditions (SPCs) applied to them.
CONTACTFORCES	Write nodal contact forces for the nodes in contact.
APPLIEDLOAD	Write nodal applied loads (from PLOADs, FORCEs, and MOMENTs.).

2.1.4.2 Element Output Choices

STRESS	Write stresses for all elements in the mesh. This option will automatically write output to the file for each stress tensor component, the von Mises equivalent stress value, and the mean stress value.
STRAIN	Write strains for all elements in the mesh. This option will automatically write output to the file for each strain tensor component, the von effective strain value, and the equivalent plastic strain value (if the material is nonlinear).
PLASTICSTRAIN	Write plastic strains for all elements in the mesh that have a plasticity model defined. If the material model does not have plasticity, a value of zero is written.

2.1.5 Restart

Autodesk Explicit always writes a restart file so that you can recover an analysis and continue further analysis. At a minimum Autodesk Explicit will write a restart checkpoint at the end of every case in the analysis procedure sequence.

You can specify more frequent checkpoints in each procedure by defining the number of restart writes that you want to have. To specify the number of restart increments during each explicit case use the RESTARTWRITE parameter. If no RESTARTWRITE parameter appears in the case options, the default is to write only one restart checkpoint at the end of the duration so that you can continue the run.

The file extension ".rst.xml" is used for the restart files.

2.1.6 History

Autodesk Explicit allows the user to define a set of tracer points at which the history of user specified nodal variables are written to a history file. This is the "time history" at a specified tracer point. You create history output requests using the XYPLOT options in the Case Control. The file extension ".his.xml" is used for the history files.

3. EXPLICIT DYNAMICS

3.1 Explicit Dynamics Integration

A formal definition of the central difference time integration algorithm is given by:

$$\ddot{u}_k = \frac{\dot{u}_{k+\frac{1}{2}} - \dot{u}_{k-\frac{1}{2}}}{t_{k+1} - t_{k-1}}$$

$$\dot{u}_{k+\frac{1}{2}} = \frac{u_{k+1} - u_k}{t_{k+1} - t_k}$$

$$\ddot{u}_{k-\frac{1}{2}} = \frac{u_k - u_{k-1}}{t_k - t_{k-1}}$$

The *k* subscripts refer to the increment index. The "half" subscripts signify center of the time increment and indicate mean values for the increment. Rearranging the above equations, we arrive at the classic kinematic update formula for the central difference integration scheme.

$$t_{k+1} = t_k + \Delta t_{k+1}$$

$$\dot{u}_{k+\frac{1}{2}} = \dot{u}_{k-\frac{1}{2}} + \frac{\Delta t_{k+1} + \Delta t_k}{2} \ddot{u}_k$$

$$u_{k+1} = u_k + \Delta t_{k+1} \dot{u}_{k+\frac{1}{2}}$$

This central difference integration scheme is optimal for two reasons. First, it defines "mean velocities" that are natural for computing the incremental displacement and, second, it integrates the motion of a rigid body under constant acceleration exactly. It is natural, though, to want velocities defined at the beginning and end of the time step since all other kinematic quantities are defined there and we would like all kinematic quantities to be at the same time when we write output for the user. We define the "state velocities" as the velocity at the integral multiples of *k* and express them in terms of the mean velocities as:

$$\dot{u}_{k+\frac{1}{2}} = \frac{\dot{u}_{k+1} + \dot{u}_k}{2}$$

$$\dot{u}_{k-\frac{1}{2}} = \frac{\dot{u}_k + \dot{u}_{k-1}}{2}$$

Eliminating the mean velocities allows us to generate an update scheme in terms of the state velocities:

$$\dot{u}_{k+1} = \dot{u}_{k-1} + \left(\Delta t_k + \Delta t_{k+1}\right) \ddot{u}_k$$

$$u_{k+1} = u_k + \Delta t_{k+1} \frac{\dot{u}_{k+1} + \dot{u}_k}{2}$$

Unfortunately, the above update scheme requires two previous states to update the current state velocity. We can do better than that and derive a "two-state" update formula as long as we are willing to store both the state velocities (those at the beginning and end of the time increments) and the mean velocities in the state. By definition, the mean velocities are simply the average of the two state velocities and we can write a simple update formula for the state velocity in terms of the mean velocity and the previous state velocity as:

$$\dot{u}_{k+1} = 2\dot{u}_{k-\frac{1}{2}} - \dot{u}_k$$

Unfortunately, this update formula for the state velocities suffers from severe oscillations of the state velocities about the mean velocities in the presence of discontinuities in the velocity field. Such sharp discontinuities arise most often in impact problems where a node experiences a sudden change in velocity. It also arises from single point constraints where the user imposes a jump in displacement and in element deletion problems where a free surface suddenly appears. These severe oscillations are not self correcting and can grow without bound under certain circumstances. Hence an alternative update formula is required for the state velocities. Define the state velocities as a linear interpolation of the mean velocities in the form:

$$\dot{u}_{k+1} = \dot{u}_{k+\frac{1}{2}} + \frac{1}{2}\left[\frac{\dot{u}_{k+\frac{3}{2}} - \dot{u}_{k+\frac{1}{2}}}{\frac{1}{2}\left(\Delta t_k + \Delta t_{k+1}\right)}\right]$$

However, we see from our original classical definition of the central difference operator (above), this can simply be written as:

$$\dot{u}_{k+1} = \dot{u}_{k+\frac{1}{2}} + \frac{1}{2}\Delta t_{k+1}\ddot{u}_k$$

which is algebraically equivalent to a Newark-β scheme.

Since our central difference integration rule is a two step rule (old state, new state), it is not self starting and requires an initialization algorithm. We assert the following condition:

$$\dot{u}_{+\frac{1}{2}} = \dot{u}_0 + \frac{\Delta t_1}{2}\ddot{u}_0$$

And assuming a reasonable (small) initial time increment (Δt_0), we construct the initial mean velocity which enforces the above condition as:

$$\dot{u}_{-\frac{1}{2}} = \dot{u}_0 - \frac{\Delta t_0}{2}\ddot{u}_0$$

In summary, the kinematic update algorithm in Autodesk Explicit proceeds as:

$$\dot{u}_{k+\frac{1}{2}} = \dot{u}_{k-\frac{1}{2}} + \frac{\Delta t_{k+1} + \Delta t_k}{2}\ddot{u}_k$$

$$\dot{u}_{k+1} = \dot{u}_{k+\frac{1}{2}} + \frac{1}{2}\Delta t_{k+1}\ddot{u}_k$$

$$\Delta u_{k+1} = \Delta t_{k+1}\,\dot{u}_{k+\frac{1}{2}}$$

$$u_{k+1} = u_k + \Delta u_{k+1}$$

$$x_{k+1} = X_0 + u_{k+1}$$

3.2 Stability Limits

The central difference integration rule is only conditionally stable. That means that there is some minimum time step that we may not exceed or the calculations will blow up. This stability limit is tied to the highest natural frequency in the mesh. If ω_{max} is the highest natural frequency, the critical time step we cannot exceed is defined by the Courant stability limit:

$$\Delta t_{cr} \le \frac{2}{\omega_{max}}$$

The stability limit generally limits the explicit dynamics procedure to very small time increments. However, since the integration rule is "explicit" each time step requires small computational effort. It is not unusual for an explicit dynamics procedure to take thousands if not millions of time increments. The stability limit given above is for the undamped case.

There are damping terms introduced in the Autodesk Explicit dynamics procedure. All of the element formulations also contain a so called "Bulk Viscosity" that is provided to damp out high frequency noise in the solution. Moreover, you can include Rayleigh Damping in the model using parameters ALPHA and BETA in the case control. If ξ is the fraction of critical damping in the highest mode, the damped stability limit is given by:

$$\Delta t_{cr} \le \frac{2}{\omega_{max}}\left(\sqrt{1+\xi^2} - \xi\right)$$

Contrary to intuition, the stability limit is actually reduced by introducing damping. Hence, you cannot damp out numerical instability by increasing the damping in the system. That just makes matters worse. We will see in the discussion below of Rayleigh Damping with stiffness proportional damping (BETA) that can have an extremely deleterious effect on the time step computed by Autodesk Explicit. For this reason stiffness proportional damping is highly discouraged.

Fortunately, we have methods for estimating the highest natural frequency of each element in the mesh and also for estimating the amount of numerical damping in those modes (see Flanagan and Belytschko[2]). This allows us to go through all the elements of the mesh and find the smallest frequency and base our time step on that value using the damped Courant stability limit. .The stability limit of each individual element in the mesh can be generalized as the time it takes a dilatational wave to propagate across the element. We refer to this as the "transit time" for the element. Hence, the element stability limit can be written (for the undamped case) as:

$$\Delta t \le \frac{L_C}{C_d}$$

Where L_C is the element characteristic length and $C_d = \sqrt{\frac{\lambda + 2\mu}{\rho}}$ is the dilatation wave speed defined in terms of the Lame constants for the material and the density

For one dimensional elements such as rods or beams, the transit time is defined by the length of the element and the material wave speed. For solid elements, the elements have some "characteristic length" that we could use for the transit time. This could be the shortest edge length or the shortest diagonal length across the element. Autodesk Explicit contains algorithms to determine this value very precisely. However, the important consideration is that the smaller you make your elements (i.e. the more mesh refinement), the smaller the time step size. Likewise, if you have very stiff materials (steel) you will have a high wave speed in the material and a resultant small time step. Often in explicit dynamics applications you will have time steps smaller than a microsecond.

Fortunately, Autodesk Explicit will compute the stability limit for the mesh and ensure that we remain within the Courant stability limit at all times.

4. AUTOMATIC MASS SCALING

The courant stability limit of the smallest element in the entire model will control the overall global stability of the model. It does not matter if you have a million perfectly formed elements and one badly formed or extremely small element – the small element will control the time step of the problem. It is quite common in meshes comprised of tetrahedral elements that there are a few small badly formed sliver-like elements. This is particularly true for meshes created using geometry-based automatic mesh generation packages which often generate tiny sliver-like elements around regions of geometric detail.

The automatic mass scaling option in Autodesk Explicit (AUTOMASSSCALE) provides a tool for overcoming these kinds of deficiencies. Remember that the Courant stability limit of each element can be considered to be the transit time of a dilatational wave across some characteristic element length for the element as:

$$\Delta t \le \frac{L_C}{C_d}$$

where L_C is the element characteristic length and $C_d = \sqrt{\frac{\lambda + 2\mu}{\rho}}$ is the dilatation wave speed. The element stability limit is proportional to the square root of the density of the material. Clearly if we simply increase the density of the material we will achieve a larger stability limit. For example, increasing the material density by a factor of 100 would increase the stability limit by a factor of 10 and our explicit analysis would run 10 times faster. Of course, the dynamic response of the model might be significantly different given that the inertia effects are 100 times larger. However, suppose we only mass scale the tiny elements in the model and leave the others alone. This would be as if the material had some dense tiny inclusions inside the material. Using the AUTOMASSSCALE option the user can provide a minimum for the time step size and Autodesk Explicit will scale the density for each element with a stability limit below this value so that the element will have precisely the user specified stability limit. This means that a collection of elements (those below the user specified value) will have their own increased density. Generally, this option is very useful when the minimum element stability limit is an order of magnitude smaller than the average element stability limit in a part.

Autodesk Explicit provides a summary of the element statistics (before any auto mass scaling is applied) for each part of the body. This summary gives the minimum, average and maximum element stability limits and characteristic element lengths for the part. A data check run can be used to examine these limits to decide if automatic mass scaling can be used to achieve a speed-up in the run without appreciable affecting the results. The code will then provide a summary of the effects of any auto mass scaling applied by the user where it gives the number of elements scaled and the percent change in the overall mass of the part due to automatic mass scaling. For a case where the minimum stability limit is 20 times smaller than the average stability limit it is over the case that setting the auto mass scaling time step size to say half the average value will result in scaling perhaps 10 percent of the elements or less but increases the total mass of the part by less than 1 percent. A few iterations on the model using the data check option and modifying the automatic mass scaling parameter can quickly determine an acceptable value for the scaling parameter. This powerful feature can often times speed up run-times by factors of 3 or 4.

5. DAMPING

5.1 Damped Vibrations

For a 1-DOF system,

$$M\ddot{y} + C\dot{y} + Ky = F(t)$$

We can easily show that for the undamped system, the period of the system, τ, and the frequency of the system (Hz), f, are given by:

$$\omega = \sqrt{\frac{K}{M}}, \quad \tau = \frac{2\pi}{\omega}, \quad f = \frac{1}{\tau}$$

Then, for the damped case, the response is given by:

$$y = e^{-\beta t}\left(C_1 \sin \omega_d t + C_2 \sin \omega_d t\right)$$

where

$$\beta = \frac{C}{2M} \qquad \omega_d = \sqrt{w^2 - \beta^2}$$

We define critical damping, C_{cr}, as the damping level for which there is no oscillatory motion:

$$C_{cr} = 2M\omega = 2\sqrt{KM}$$

and the damping ratio also known as fraction of critical damping is defined as:

$$\xi = \frac{C}{C_{cr}} = \frac{2M\beta}{2M\omega} = \frac{\beta}{\omega}$$

5.2 Amplitude Decrement

The amplitude decrement is defined as the ratio of the amplitude for each successive cycle of the motion.

$$\frac{y(t+\tau)}{y(t)} e^{-\beta\tau} = e^{-\xi\omega\left(\frac{2\pi}{\omega}\right)} = e^{-2\pi\xi}$$

For six percent of critical damping, $\xi = .06$, we obtain an amplitude decrement of:

$$\frac{y(t+\tau)}{y(t)} e^{-2\pi(.06)} = .686$$

5.3 Rayleigh Damping

In a multiple DOF system,

$$M\ddot{y} + C\dot{y} + Ky = F(t)$$

where M, C, and K represent mass, damping and stiffness matrices, respectively. Rayleigh damping represents the special case where we choose to form the damping matrix from a linear combination of the mass and stiffness matrices:

$$C = \alpha_R M + \beta_R K$$

The damping ratio as a function of frequency for Rayleigh damping is:

$$\xi = \frac{\alpha_R}{2\omega} + \frac{\beta_R \omega}{2}$$

Thus, the mass or alpha damping increases with decreasing frequency, and the stiffness or beta damping increases linearly with frequency. Figure 5-1 below shows the damping ratio as a function of frequency. Curves are shown for pure mass proportional damping (Beta=0) and pure stiffness proportional damping (Alpha=0). For combined mass and stiffness proportional damping (Alpha+Beta), the curves are simply added.

In general, we do not combine the effects of mass and stiffness proportional damping. We will use ξ_M to denote the separate mass-induced critical damping coefficient and ξ_K to denote the separate stiffness-induced critical damping coefficient, and we will describe these in terms of frequency.

Thus, for mass proportional damping:

$$\xi_M = \frac{\alpha_R}{4\pi f} \quad or \quad \alpha_R = 4\pi f \xi_M$$

and for stiffness proportional damping:

$$\xi_K = \pi \beta_R f \quad or \quad \beta_R = \frac{\xi_K}{\pi f}$$

Consider the example of a simply supported beam where we reduced the amplitude of the applied pressure to a value of 10000. We reduce the amplitude of the pressure so that the displacements are several orders of magnitude less than in the original problem definition. If we do not reduce the displacements to what is essentially small deformations, the large deformation effect of stretching the mid-plane of the beam has a dramatic effect on the natural frequency of the beam and the equations we have derived above for Rayleigh damping are not appropriate.

The period of oscillation of the beam undergoing small deformations is .01463 which gives a frequency of 68.35 Hz. Suppose we want to induce 6% of critical damping at the frequency of 68.35 Hz. We can impose this damping as mass proportional damping or as stiffness proportional damping. For mass and stiffness proportional damping, the alpha and beta parameters would be:

$$\alpha_R = 4\pi(68.35)(.06) = 51.53 \qquad \beta_R = \frac{(.06)}{\pi(68.35)} = 2.794x10^{-4}$$

When we run our simply supported beam problem with no damping and with an instantaneously applied pressure of 10000, the beam oscillates about the static displacement of 6.818e-4. The analysis takes 1801 increments with an average time increment size of 1.65e-5.

If we run the same model using these alpha and beta parameters we obtain the following results:

	Mass Proportional	**Stiffness Proportional**
$y(t_1) - y_{static}$	5.637e-4	5.658e-4
$y(t_1 + \tau) - y_{static}$	3.837e-4	3.923e-4
$\frac{y(t_1 + \tau)}{y(t_1)}$	.6807	.693
Number Time Increments	1801	23507
Average Time Increment	1.65e-5	1.27e-6

In both cases the results match the analytical prediction for the amplitude ratio value for 6% critical damping computed above (.686). In fact the solutions to both cases are almost indistinguishable from one another except for one very startling fact: **the stiffness proportional damping case takes more than an order of magnitude more time steps!** The mass proportional damping case took exactly the same time sizes as the undamped case. This rather startling result shows that you should seldom use stiffness proportional damping in an explicit dynamic analysis.

Why does the stiffness proportional damping take such small time increments? Remember the Courant stability limit is based upon the highest natural frequency in the mesh. When you introduce stiffness proportional damping you are increasing the element natural frequencies dramatically. The automatic time incrementation algorithm is monitoring the natural frequencies of all the elements and correctly identifies this increase in natural frequency.

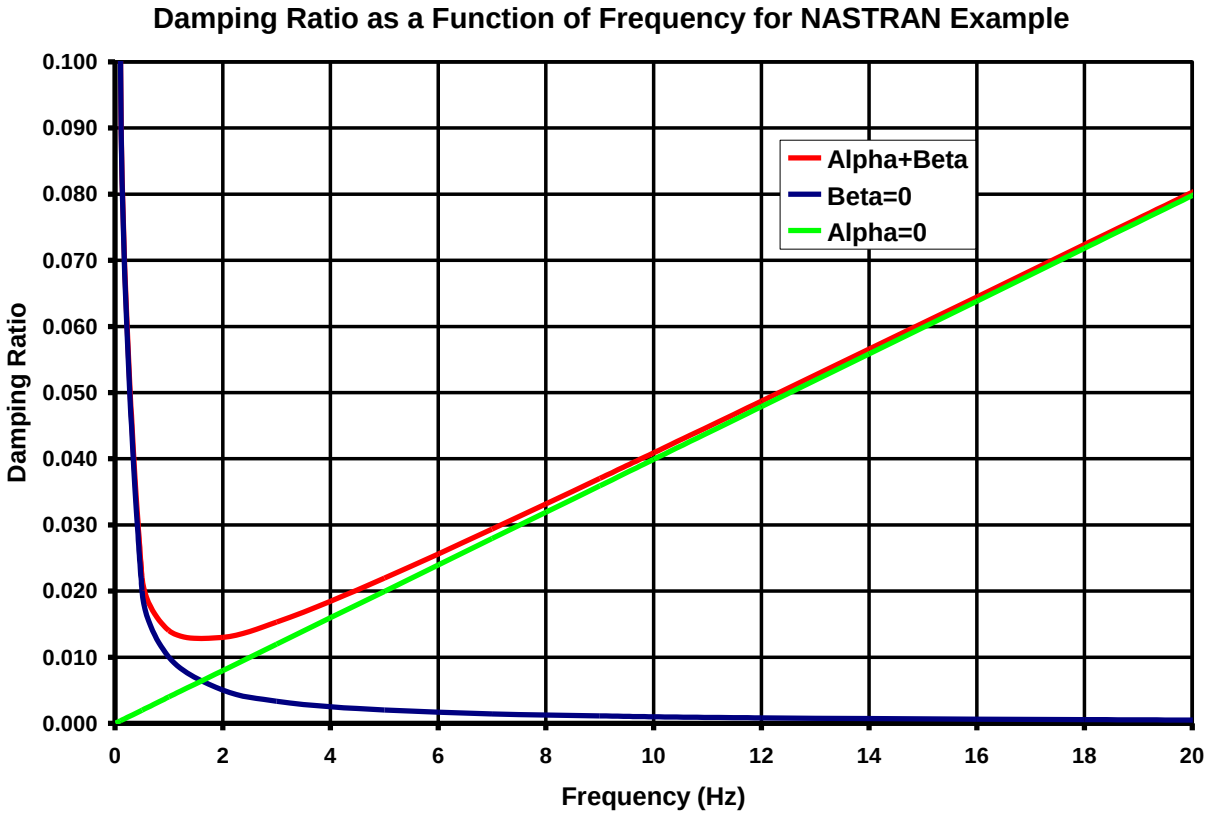


Figure 5-1. Damping Ratio as a Function of Frequency.

6. PROPERTIES

All elements in the Nastran input model must be assigned a property reference (e.g. PSOLID, PSHELL, etc.). The mesh translator that creates the input file (.inp.xml) and the mesh file (.tsm.xml) will sort the elements of the mesh into "element blocks" which are sets of elements that share the same property PID in the Nastran input file. These element blocks will fall into different categories (e.g. solid, shell, etc.).

The different types of element block categories that will be produced from the mesh translator are:

Category	Nastran Property Option
Solid	The Solid category will be assigned according to the following rules: • PSOLID – element type CHEXA, CTETRA, or CPRISM elements • PSHELL – element type CQUAD4, CQUADR, CTRIA3, or CTRIAR with MID2 = -1 (plane strain).
Shell	The Shell category will be assigned if the element type is the CQUAD4, CQUADR, CTRIA3, or CTRIAR and the property ID is one of: • PSHELL with MID2 greater than 0. • PCOMP.
Membrane	The Membrane category will be assigned if the element type is the CQUAD4, CQUADR, CTRIA3, or CTRIAR the property ID PCOMP with MID2 greater equal 0 (plane stress).
Beam	Property ID is PBEAM.
Truss	Property ID is PBAR or PROD.
Spring	Property ID is CELAST1 or CELAS2.

7. ELEMENTS

All of the element formulations in Autodesk Explicit are lower order, reduced integration elements (i.e. there are no quadratic isoparametric elements). All of the elements use the Uniform Gradient formulation developed by Flanagan and Belytschko [1], [2].

7.1 Plane Stress/Strain Uniform Gradient 4-Node Quadrilateral with Hourglass Control

7.1.1 Element Kinematics

The four-node, two-dimensional, isoparametric element is widely used in computational mechanics. Optimum integration schemes for these elements present a difficult dilemma. A one point integration of the element under-integrates the element resulting in rank deficiency for the element which manifests itself in spurious zero energy modes, commonly referred to as "hour glass" modes. A 2x2 integration of the element over-integrates the element and can lead to serious problems of element locking in fully plastic and incompressible material representations. The four-point integration also carries a tremendous computational and memory footprint penalty compared to the one point rule. We use a "uniform gradient" formulation that while similar to the one point integration rule provides a precise mathematical formulation and a complete theory for eliminating the spurious energy modes. This formulation is due to Flanagan & Belytschko.

The velocity gradient L, is defined in terms of the nodal velocities as:

$$L_{ij} = \dot{u}_{i,j} = \dot{u}_{il}\Phi_{I,j} \qquad (0.1)$$

By convention, a comma preceding a lowercase subscript denotes differentiation with respect to the spatial coordinates (e.g. $\dot{u}_{i,j}$ denotes $\partial \dot{u}_i / \partial x_j$).

The 2D isoparametric shape functions map the unit square in $\xi - \eta$ to an arbitrary quadrilateral in $x - y$, as shown in Figure 7-1. We choose to center the unit square at the origin in $\xi - \eta$ space that the shape functions may be conveniently expanded in terms of an orthogonal set of basis vectors, given in Table 7-1, as follows:

$$\Phi_I = \frac{1}{4}\sum_I + \frac{1}{2}\xi\Lambda_{1I} + \frac{1}{2}\eta\Lambda_{2I} + \xi\eta\Gamma_I \qquad (0.2)$$

The basis vectors represent the displacement modes of a unit square. The first vector $\sum_I$ represents a rigid body translation. We call $\sum_I$ the summation vector since it may be employed in indicial notation to represent the algebraic sum of a vector.

The linear base vectors Λ_{iI} may be readily combined to define the uniform normal strains and shear strain in the element. We refer to the Λ_{iI} as the volumetric base vectors since, as we will illustrate below, they are the only base vectors that appear in the element area expression.

The last base vector Γ_I gives rise to the linear strain modes which are neglected in the uniform strain formulation. This vector defines the hourglass patterns for the unit square.

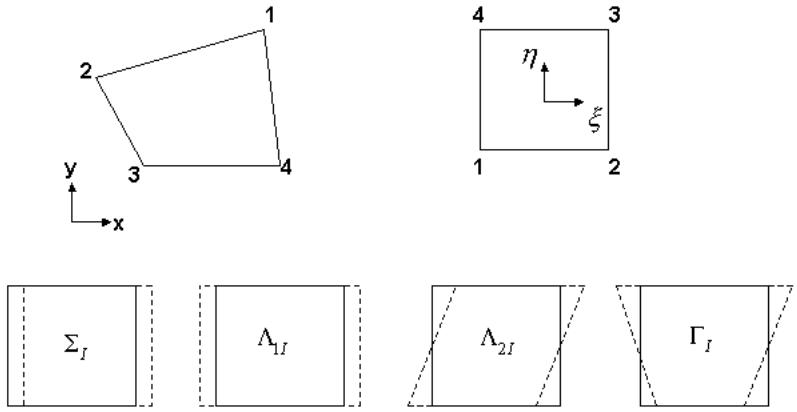


Figure 7.1. Mode Shapes for the Four-Node Constant Strain Quadrilateral Element.

Table 7-1. Base Vectors for 4-Node Quadrilateral.

Node	ξ	η	Σ_I	Λ_{1I}	Λ_{2I}	Γ_I
1	-.5	-.5	+1	-1	-1	+1
2	+.5	-.5	+1	+1	-1	-1
3	+.5	+.5	+1	+1	+1	+1
4	-.5	+.5	+1	-1	+1	-1

In the finite element method, we replace the momentum equation with the weak form of the equation. Using the principle of virtual work, we write the weak form of the equation as:

$$\sum_e \int_V (T_{ij,j} + \rho b_i - \rho \ddot{u}_i)\delta u_i dV = 0 \qquad (0.3)$$

In plane strain and plane stress, the thickness of the body is considered uniform and arbitrary, and therefore can be eliminated from the preceding expression. Integrating by parts and applying Gauss's Divergence Theorem gives:

$$\sum_e \left[\int_S T_{ij} n_j \delta u_i dS - \int_A T_{ij} \delta u_{i,j} dA + \int_A \rho b_i \delta u_i dA - \int_A \rho \ddot{u}_i \delta u_i dA \right] = 0 \qquad (0.4)$$

The second integral in the preceding equation is used to define the element internal force vector f_{iI} as:

$$\delta u_{iI} f_{iI} = \int_A T_{ij} \delta u_{i,j} dA \qquad (0.5)$$

The first and third integrals define the external force vector, and the fourth integral defines the inertial response.

7.1.2 Uniform Gradient Assumption

For our element formulation we concentrate on the element internal force vector f_{iI}. We perform the integration over the element by neglecting the nonlinear portion of the element displacement field, thereby considering a state of uniform strain and stress. Equation (0.5) is approximated by:

$$f_{iI} = \bar{T}_{ij} \int_A \Phi_{i,j} dA \qquad (0.6)$$

where we have eliminated the arbitrary virtual displacements, and $\bar{T}_{ij}$ represents the assumed uniform stress field which will be referred to as the mean stress tensor. By neglecting the nonlinear displacements, we have assumed that the mean stresses depend only on the mean strains. Mean kinematic quantities are defined by integrating over the element as follows:

$$\bar{\dot{u}}_{i,j} = \frac{1}{V} \int_V \dot{u}_{i,j} dV \qquad (0.7)$$

We now define the discrete gradient operator as:

$$G_{iI} = \int_A \Phi_{I,i} dA \qquad (0.8)$$

The mean velocity gradient defined in Equation (0.7) can be re-written using Equation (0.8) as:

$$\bar{\dot{u}}_{i,j} = \frac{1}{A} \dot{u}_{iI} G_{jI} \qquad (0.9)$$

Combining Equations (0.6) and (0.9), we may express the nodal forces by:

$$f_{iI} = \bar{T}_{ij} G_{jI} \qquad (0.10)$$

We see that the gradient operator defined by Equation (0.8) can also be used to compute the divergence of the mean stress tensor.

Computing the nodal forces with the uniform gradient formulation requires the evaluation of the element area and gradient operator. These two tasks are linked because:

$$x_{i,j} = \delta_{ij} \qquad (0.11)$$

where δ_{ij} is the Kronecker delta. Equations, (0.7), (0.8) and (0.11) yield:

$$x_{iI} G_{jI} = \int_A (x_{iI} \Phi_I)_{,j} dA = A \delta_{ij} \qquad (0.12)$$

Consequently, the gradient may be expressed by:

$$G_{iI} = \frac{\partial A}{\partial x_{iI}} \qquad (0.13)$$

To integrate the element area in closed form, we use the Jacobian of the isoparametric transformation to transform the integral in $x-y$ space to an integral over the unit square:

$$A = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} J d\eta d\xi \tag{0.14}$$

where:

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \tag{0.15}$$

Therefore, Equation (0.14) can be written as:

$$A = x_I y_J C_{IJ} \tag{0.16}$$

where:

$$C_{IJ} = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\left(\frac{\partial \Phi_I}{\partial \xi}\frac{\partial \Phi_J}{\partial \eta} - \frac{\partial \Phi_I}{\partial \eta}\frac{\partial \Phi_J}{\partial \xi}\right) \tag{0.17}$$

In light of Equation (0.2), the above integration involves at most bilinear functions. Therefore, only the constant term does not vanish and the integration yields:

$$C_{IJ} = \frac{1}{4}\left(\Lambda_{1I}\Lambda_{2J} - \Lambda_{2I}\Lambda_{1J}\right) \tag{0.18}$$

Note that C_{IJ} is anti-symmetric:

$$C_{IJ} = -C_{JI} \tag{0.19}$$

Evaluating Equation (0.18), we obtain the following explicit representation for C_{IJ}

$$C_{IJ} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \tag{0.20}$$

Substituting the above expression into Equation (0.16), we obtain the familiar expression for the area of a quadrilateral:

$$A = \frac{1}{2}\left[(x_3 - x_1)\cdot(y_4 - y_2) + (x_2 - x_4)\cdot(y_3 - y_2)\right] \tag{0.21}$$

Using this result in Equation (0.13), the gradient operator can be expressed as:

$$G_{IJ} = C_{IJ} \begin{Bmatrix} y_j \\ -x_j \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} (y_2 - y_4) & (y_3 - y_1) & (y_4 - y_2) & (y_1 - y_3) \\ (x_4 - x_2) & (x_1 - x_3) & (x_2 - x_4) & (x_3 - x_1) \end{bmatrix} \tag{0.22}$$

The mean stress approach used here gives the same result as using a one-point Gauss Quadrature rule to integrate the element. However, we will now see that there is an elegant and accurate mathematical formulation for the uniform gradient formulation to deal with the zero energy modes that were neglected in the integration (these very same zero energy modes are neglected in the one-point Gauss Quadrature rule as well).

7.1.3 Hourglass Control

Our uniform gradient formulation considers only a fully linear velocity field. The remaining portion of the nodal velocity field is the so-called "hourglass" field. Excitation of these modes may lead to severe unresisted mesh distortion. The hourglass control algorithm we will now describe isolates the hourglass modes so that they can be treated independently of the rigid body and uniform strain modes.

A fully linear velocity field for the quadrilateral can be described by:

$$\dot{u}_i^{LIN} = \dot{\bar{u}}_i + \dot{\bar{u}}_{i,j} \left(x_j - \bar{x}_j \right) \tag{0.23}$$

The mean coordinates $\bar{x}_i$ correspond to the center of the element and are defined as:

$$\bar{x}_i = \frac{1}{4} x_{iI} \sum_I \tag{0.24}$$

The mean translational velocity is similarly defined by:

$$\dot{\bar{u}}_i = \frac{1}{4} \dot{u}_{iI} \sum_I \tag{0.25}$$

The linear portion of the nodal velocity field may be expressed by specializing Equation (0.23) to the nodes as follows:

$$\dot{u}_{iI}^{LIN} = \dot{\bar{u}}_i \sum_I + \dot{\bar{u}}_{i,j} \left(x_{jI} - \bar{x}_j \sum_I \right) \tag{0.26}$$

where $\sum_I$ is used to maintain consistent index notation and indicates that $\dot{\bar{u}}_I$ and $\bar{x}_J$ are independent of position within the element. From Equations (0.12) and (0.26); and the orthogonality of the base vectors, it follows that:

$$\dot{u}_{iI} \sum_I = \dot{u}_{iI}^{LIN} \sum_I = 4 \dot{u}_i \tag{0.27}$$

and

$$\dot{u}_{iI} G_{jI} = \dot{u}_{iI}^{LIN} G_{jI} = A \dot{\bar{u}}_{i,j} \tag{0.28}$$

The hourglass field $\dot{u}_{iI}^{HG}$ may now be defined by removing the linear portion of the nodal velocity field:

$$\dot{u}_{iI}^{HG} = \dot{u}_{iI} - \dot{u}_{iI}^{LIN} \tag{0.29}$$

Equations (0.27) through (0.29) prove that $\sum_I$ and G_{jI} are orthogonal to the hourglass field:

$$\dot{u}_{iI}^{HG} \sum_I = 0 \qquad (0.30)$$

$$\dot{u}_{iI}^{HG} G_{jI} = 0 \qquad (0.31)$$

Furthermore, it can be shown that the gradient operator is a linear combination of the volumetric base vectors, Λ_{iI}, so Equation (0.31) can be written as:

$$\dot{u}_{iI}^{HG} \Lambda_{iI} = 0 \qquad (0.32)$$

where no sum is implied on the lower case indices of Equation (0.32).

Equations (0.30) and (0.32) show that the hourglass field is orthogonal to all the base vectors in Table 7-1 except the hourglass base vectors. Therefore $\dot{u}_{iI}^{HG}$ may be expanded as a linear combination of the hourglass vectors as follows:

$$\dot{u}_{iI}^{HG} = \frac{1}{2} \dot{q}_i \Gamma_I \qquad (0.33)$$

The hourglass nodal velocities are represented by $\dot{q}_i$ above (the leading constant is added to normalize Γ_I). We now define the hourglass shape vector γ_I such that:

$$\dot{q}_i = \frac{1}{2} \dot{u}_{iI} \gamma_I \qquad (0.34)$$

By substituting Equations (0.26), (0.29), and (0.34) into Equation (0.33), then multiplying by Γ_I and using the orthogonality of the base vectors, we obtain the following:

$$\dot{\bar{u}}_{iI} \Gamma_I - \dot{\bar{u}}_{i,j} x_{jI} \Gamma_I = \dot{u}_{iI} \gamma_I \qquad (0.35)$$

With the definition of the mean velocity gradient, Equation (0.35), we can eliminate the nodal velocities above. As a result, we can compute γ_I from the following expression:

$$\gamma_I = \Gamma_I - \frac{1}{A} G_{iI} x_{iJ} \Gamma_J \qquad (0.36)$$

The difference between the hourglass base vectors Γ_I and the hourglass shape vectors γ_I is very important. They are identical if and only if the quadrilateral is a parallelogram (the second term on the right hand side of Equation (0.36) vanishes for a parallelogram). For a general shape quadrilateral, Γ_I is orthogonal to G_{jI} while γ_I is orthogonal to the linear velocity field $\dot{u}_{iI}^{LIN}$. While Γ_I defines the hourglass pattern, γ_I is necessary to accurately detect hourglassing. Equation (0.36) is simple enough for the quadrilateral that it can be written explicitly as:

$$\gamma_I = \frac{1}{A}\begin{bmatrix} x_2(y_3 - y_4) + x_3(y_4 - y_2) + x_4(y_2 - y_3) \\ x_3(y_1 - y_4) + x_4(y_3 - y_1) + x_1(y_4 - y_3) \\ x_4(y_1 - y_2) + x_1(y_2 - y_4) + x_2(y_4 - y_1) \\ x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1) \end{bmatrix} \qquad (0.37)$$

For the purpose of controlling the hourglass modes, we define generalized forces Q_i, which are conjugate to $\dot{q}_i$ so that the rate of work is:

$$\dot{u}_{iI} f_{iI}^{HG} = \frac{1}{2} Q_i \dot{q}_i \qquad (0.38)$$

for arbitrary $\dot{u}_{iI}$. Using Equation (0.34), it follows that the contribution of the hourglass resistance to the nodal forces is given by:

$$f_{iI}^{HG} = \frac{1}{2} Q_i \gamma_I \qquad (0.39)$$

The hourglass generalized forces Q_i represent element variables. There are two kinds of hourglass resistance algorithms that you can use, stiffness (K) or damping (V). Or, you can use a combination of the two. We express the combination of the two by:

$$Q_i = Q_i^K + Q_i^V \qquad (0.40)$$

In terms of adjustable stiffness (κ) and viscosity (ε) factors, these resistances are given by:

$$\dot{Q}_i^K = \frac{\kappa}{2} 2\hat{\mu} \frac{G_{kI} G_{kI}}{A} \dot{q}_i \qquad (0.41)$$

$$Q_i^V = \varepsilon \sqrt{min(0, 2\hat{\mu}) M} \cdot \dot{q}_i \qquad (0.42)$$

where M is the mass of the element and $\hat{\mu}$ is the Lame constant. We will show below how to determine the effective Lame constants for the element in a general manner. Note that the stiffness expression is a rate expression, implying that the stiffness hourglass resistance forces must be integrated and stored as an element state variable.

Observe that the nodal anti-hourglass forces of Equation (0.39) have the shape of γ_I rather than Γ_I. This fact is essential since the anti-hourglass forces should be orthogonal to the linear velocity field, so that no energy is transferred to or from the rigid body and uniform strain modes by the anti-hourglassing scheme.

The stiffness based hourglass approach is preferred to the viscous based approach. It generally proves to be more accurate and stable.

7.2 3D Uniform Gradient 8 Node Hexahedron with Hourglass Control

7.2.1 Element Kinematics

The eight-node, three-dimensional, isoparametric element is widely used in computational mechanics. Optimum integration schemes for these elements present a difficult dilemma. A one point integration of the element under-integrates the element resulting in rank deficiency for the element which manifests itself in spurious zero energy modes, commonly referred to as "hourglass" modes. A 2x2 integration of the element over-integrates the element and can lead to serious problems of element locking in fully plastic and incompressible material representations. The eight-point integration also carries a tremendous computational and memory footprint penalty compared to the one point rule. We use a "uniform gradient" formulation that while similar to the one point integration rule provides a precise mathematical formulation and a complete theory for eliminating the spurious energy modes. This formulation is due to Flanagan & Belytschko.

The velocity gradient L, is defined in terms of the nodal velocities as:

$$L_{ij} = \dot{u}_{i,j} = \dot{u}_{iI}\Phi_{I,j} \qquad (0.43)$$

By convention, a comma preceding a lowercase subscript denotes differentiation with respect to the spatial coordinates (e.g. $\dot{u}_{i,j}$ denotes $\partial \dot{u}_i / \partial x_j$).

The 3D isoparametric shape functions map the unit square in ξ_i-space (ξ_i is written explicitly as (ξ,η,ζ)) to an arbitrary hexahedron in 3D space, as shown in Figure 7-2. We choose to center the unit square at the origin in ξ_i space that the shape functions may be conveniently expanded in terms of an orthogonal set of basis vectors, given in Table 7-2, as follows:

$$\Phi_I = \frac{1}{8}\sum_I + \frac{1}{4}\xi\Lambda_{1I} + \frac{1}{4}\eta\Lambda_{2I} + \frac{1}{4}\zeta\Lambda_{3I} + \frac{1}{2}\eta\zeta\Gamma_{1I} + \frac{1}{2}\zeta\xi\Gamma_{2I} + \frac{1}{2}\xi\eta\Gamma_{3I} + \xi\eta\zeta\Gamma_{4I} \qquad (0.44)$$

The basis vectors represent the displacement modes of a unit cube. The first vector $\sum_I$ represents a rigid body translation. We call $\sum_I$ the summation vector since it may be employed in indicial notation to represent the algebraic sum of a vector.

The linear base vectors Λ_{iI} may be readily combined to define the three uniform normal strains and three rigid body rotation modes for the unit cube. We refer to the Λ_{iI} as the volumetric base vectors since, as we will illustrate below, they are the only base vectors that appear in the element volume expression.

The last base vectors Γ_{iI} give rise to the linear strain modes which are neglected in the uniform strain formulation. This vector defines the hourglass patterns for the unit square.

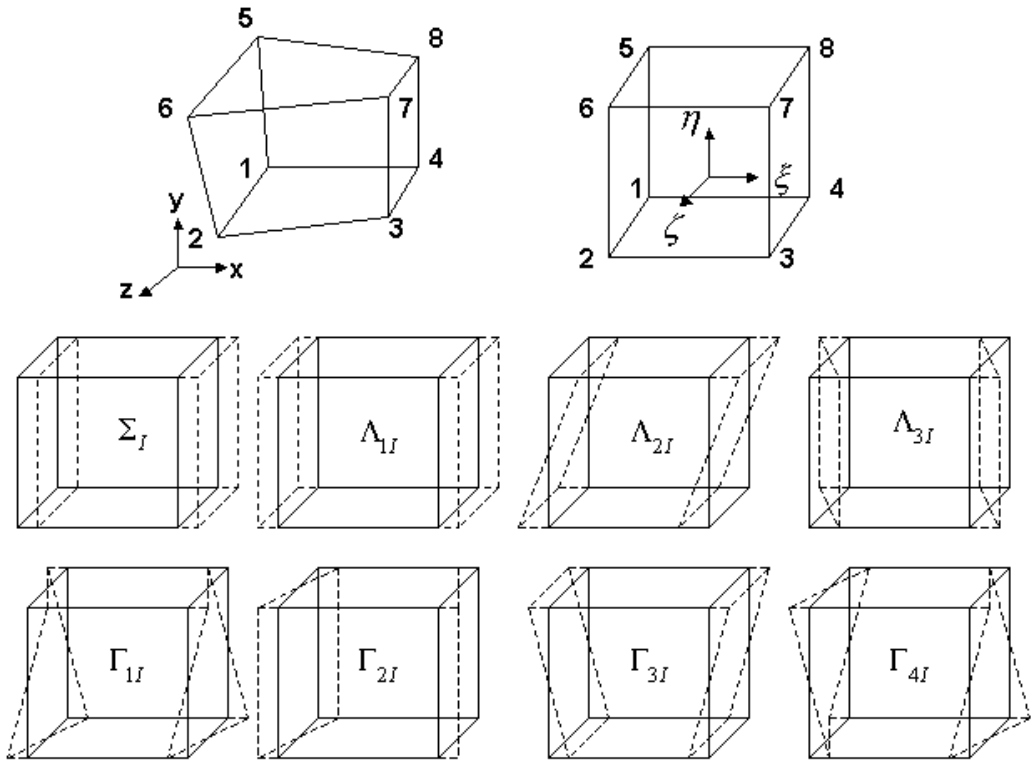


Figure 7-2. Mode Shapes for the Eight-Node Constant Strain Hexahedral Element.

Table 7-2. Base Vectors for 8-Node Hexahedral.

Node	ξ	η	ζ	Σ_I	Λ_{1I}	Λ_{2I}	Λ_{3I}	Γ_{1I}	Γ_{2I}	Γ_{3I}	Γ_{4I}
1	-.5	-.5	-.5	+1	-1	-1	-1	+1	+1	+1	-1
2	+.5	-.5	-.5	+1	+1	-1	-1	+1	-1	-1	+1
3	+.5	+.5	-.5	+1	+1	+1	-1	-1	-1	+1	-1
4	-.5	+.5	-.5	+1	-1	+1	-1	-1	+1	-1	+1
5	-.5	-.5	+.5	+1	-1	-1	+1	-1	-1	+1	+1
6	+.5	-.5	+.5	+1	+1	-1	+1	-1	+1	-1	-1
7	+.5	+.5	+.5	+1	+1	+1	+1	+1	+1	+1	+1
8	-.5	+.5	+.5	+1	-1	+1	+1	+1	-1	-1	-1

In the finite element method, we replace the momentum equation with the weak form of the equation. Using the principle of virtual work, we write the weak form of the equation as:

$$\sum_e \int_V \left(T_{ij,j} + \rho b_i - \rho \ddot{u}_i\right) \delta u_i \, dV = 0 \qquad (0.45)$$

Integrating by parts and applying Gauss's Divergence Theorem gives:

$$\sum_e \left[\int_{S_e} T_{ij} n_j \delta u_i \, dS - \int_V T_{ij} \delta u_{i,j} \, dV + \int_V \rho b_i \delta u_i \, dV - \int_V \rho \ddot{u}_i \delta u_i \, dV \right] = 0 \qquad (0.46)$$

The second integral in the preceding equation is used to define the element internal force vector f_{iI} as:

$$\delta u_{iI} f_{iI} = \int_V T_{ij} \delta u_{i,j} dV \qquad (0.47)$$

The first and third integrals define the external force vector, and the fourth integral defines the inertial response.

7.2.2 Uniform Gradient Assumption

For our element formulation we concentrate on the element internal force vector f_{iI}. We perform the integration over the element by neglecting the nonlinear portion of the element displacement field, thereby considering a state of uniform strain and stress. Equation (0.47) is approximated by:

$$f_{iI} = \bar{T}_{ij} \int_{V_e} \Phi_{i,j} dV \qquad (0.48)$$

where we have eliminated the arbitrary virtual displacements, and $\bar{T}_{ij}$ represents the assumed uniform stress field that will be referred to as the mean stress tensor. By neglecting the nonlinear displacements, we have assumed that the mean stresses depend only on the mean strains. Mean kinematic quantities are defined by integrating over the element as follows:

$$\bar{\dot{u}}_{i,j} = \frac{1}{V} \int_V \dot{u}_{i,j} dV \qquad (0.49)$$

We now define the discrete gradient operator as:

$$G_{iI} = \int_V \Phi_{I,i} dV \qquad (0.50)$$

The mean velocity gradient defined in Equation (0.49) can be re-written using Equation (0.50) as:

$$\bar{\dot{u}}_{i,j} = \frac{1}{V} \dot{u}_{iI} G_{jI} \qquad (0.51)$$

Combining Equation (0.48) and (0.50), we may express the nodal forces by:

$$f_{iI} = \bar{T}_{ij} G_{jI} \qquad (0.52)$$

We see that the gradient operator defined by Equation (0.50) can also be used to compute the divergence of the mean stress tensor.

Computing the nodal forces with the uniform gradient formulation requires the evaluation of the element area and gradient operator. These two tasks are linked because

$$x_{i,j} = \delta_{ij} \qquad (0.53)$$

where δ_{ij} is the Kronecker delta. Equations (0.49), (0.50) and (0.53) yield:

$$x_{iI}G_{jI} = \int_{V_e} \left(x_{iI}\Phi_I\right)_{,j} dV = V\delta_{ij} \tag{0.54}$$

Consequently, the gradient may be expressed by:

$$G_{iI} = \frac{\partial V}{\partial x_{iI}} \tag{0.55}$$

To integrate the element volume in closed form, we use the Jacobian of the isoparametric transformation to transform the integral in $x-y$ space to an integral over the unit square:

$$V = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\int_{-1/2}^{1/2} J d\eta d\xi d\zeta \tag{0.56}$$

where

$$J = e_{ijk}\frac{\partial x}{\partial \xi_i}\frac{\partial y}{\partial \eta_j}\frac{\partial z}{\partial \zeta_k} \tag{0.57}$$

Therefore, Equation (0.56) can be written as:

$$V = x_I y_J z_K C_{IJK} \tag{0.58}$$

where

$$C_{IJK} = e_{ijk}\int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\int_{-1/2}^{1/2} \frac{\partial \Phi_I}{\partial \xi_i}\frac{\partial \Phi_J}{\partial \eta_j}\frac{\partial \Phi_K}{\partial \zeta_k} d\zeta d\eta d\xi \tag{0.59}$$

Observe that the coefficient array C_{IJK} is identical for all hexahedrons. Furthermore, it possess the alternator properties as given below:

$$C_{IJK} = C_{JKI} = C_{KIJ} = -C_{IKJ} = -C_{JIK} = -C_{KJI} \tag{0.60}$$

Applying Equations (0.55) and (0.60) to Equation (0.58) yields the following form for evaluating the gradient operator:

$$G_{iI} = \begin{bmatrix} y_J z_K \\ z_J x_K \\ x_J y_K \end{bmatrix} C_{IJK} \tag{0.61}$$

In light of Equation (0.44), it is evident that evaluating each component of C_{IJK} involves integrating a polynomial that is at most bi-quadratic. However, since we are integrating over a symmetric region, any term with a linear dependence will vanish. The only terms, which survive the integration, will be the constant, square, double square and triple square terms. The resulting expression for C_{IJK} is:

$$C_{IJK} = \frac{1}{192} e_{ijk}\left(3\Lambda_{iI}\Lambda_{jJ}\Lambda_{kK} + \Lambda_{iI}\Gamma_{kJ}\Gamma_{jK} + \Gamma_{kI}\Lambda_{jJ}\Gamma_{iK} + \Gamma_{jI}\Gamma_{iJ}\Lambda_{kK}\right) \tag{0.62}$$

Seven independent terms of C_{IJK} are listed in Table 7-3. These terms may be evaluated via Equation (0.62) and Table 7-2, after which practical formulas for computing the gradient operator are developed. Since C_{IJK} has the alternator properties given in Equation (0.60), only 56 (the combination of eight nodes taken three at a time) distinct nonzero terms are possible. However, the volume must be independent of the selection of node 1, which implies that C_{IJK} is invariant if the nodes are permuted according to Table 7-4. Therefore, only seven terms of C_{IJK} need to be evaluated. Only three of these seven terms do not vanish, as indicated in Table 7-3. All other nonzero terms of C_{IJK} are found by permuting the nodes according to Table 7-4 and using the alternator properties of Equation (0.60). Alternatively, the nonzero terms may be generated by applying anti-symmetry ($C_{IJK} = -C_{IKJ}$) to Table 7-3, then permuting according to Tables 7-5 and 7-4, successively. The latter scheme results in a formula for computing the gradient operator.

The first term of the gradient operator is:

$$\begin{aligned} G_{11} = \frac{1}{12}\big[& y_2\big((z_6 - z_3) - (z_4 - z_5)\big) + y_3(z_2 - z_4) \\ & + y_4\big((z_3 - z_8) - (z_5 - z_2)\big) + y_5\big((z_8 - z_6) - (z_2 - z_4)\big) \\ & + y_6(z_5 - z_2) + y_8(z_4 - z_5)\big] \end{aligned} \qquad (0.63)$$

Table 7-3. Nonzero Terms Generated by Applying Asymmetry.

I	J	K	C_{IJK}
1	2	3	-1/12
1	2	5	+1/12
1	2	6	+1/12
1	2	7	0
1	2	8	0
1	3	5	0
1	3	6	0

Table 7-4. Nodal Permutations.

1	2	3	4	5	6	7	8
2	3	4	1	6	7	8	5
3	4	1	2	7	8	5	6
4	1	2	3	8	5	6	7
5	8	7	6	1	4	3	2
6	5	8	7	2	1	4	3
7	6	5	8	3	2	1	4
8	7	6	5	4	3	2	1

Table 7-5. Three Possible Orientations of Node Numbering.

1	2	3	4	5	6	7	8
1	4	8	5	2	3	7	6
1	5	6	2	4	8	7	3

Table 7-6. Coordinate Axes Permutations.

1	2	3
2	3	1
3	1	2

Other terms of the gradient operator are evaluated using the same formula after permuting the nodes according to Table 7-4 and, subsequently, permuting the coordinate axes according to Table 7-6. The element volume is most easily computed by contracting the gradient operator and nodal coordinates as per Equation (0.54).

The mean stress approach used here gives the same result as using a one-point Gauss Quadrature rule to integrate the element. However, we will now see that there is an elegant and accurate mathematical formulation for the uniform gradient formulation to deal with the zero energy modes that were neglected in the integration (these very same zero energy modes are neglected in the one-point Gauss Quadrature rule as well).

7.2.3 Hourglass Control

Our uniform gradient formulation considers only a fully linear velocity field. The remaining portion of the nodal velocity field is the so-called "hourglass" field. Excitation of these modes may lead to sever, un-resisted mesh distortion. The hourglass control algorithm we will now describe isolates the hourglass modes so that they can be treated independently of the rigid body and uniform strain modes.

A fully linear velocity field for the quadrilateral can be described by:

$$\dot{u}_i^{LIN} = \dot{\bar{u}}_i + \dot{\bar{u}}_{i,j}\left(x_j - \bar{x}_j\right) \qquad (0.64)$$

The mean coordinates $\bar{x}_i$ correspond to the center of the element and are defined as:

$$\bar{x}_i = \frac{1}{8} x_{iI} \sum_I \qquad (0.65)$$

The mean translational velocity is similarly defined by:

$$\dot{\bar{u}}_i = \frac{1}{8} \dot{u}_{iI} \sum_I \qquad (0.66)$$

The linear portion of the nodal velocity field may be expressed by specializing Equation (0.64) to the nodes as follows:

$$\dot{u}_{iI}^{LIN} = \dot{\bar{u}}_i \sum_I + \dot{\bar{u}}_{i,j}\left(x_{jI} - \bar{x}_j \sum_I\right) \qquad (0.67)$$

where $\sum_I$ is used to maintain consistent index notation and indicates that $\dot{\bar{u}}_I$ and $\bar{x}_J$ are independent of position within the element. From Equations (0.54) and (0.67), and the orthogonality of the base vectors, it follows that:

$$\dot{u}_{iI} \sum_I = \dot{u}_{iI}^{LIN} \sum_I = 8\dot{u}_i \tag{0.68}$$

and

$$\dot{u}_{iI} G_{jI} = \dot{u}_{iI}^{LIN} G_{jI} = V\dot{\bar{u}}_{i,j} \tag{0.69}$$

The hourglass field $\dot{u}_{iI}^{HG}$ may now be defined by removing the linear portion of the nodal velocity field:

$$\dot{u}_{iI}^{HG} = \dot{u}_{iI} - \dot{u}_{iI}^{LIN} \tag{0.70}$$

Equations (0.68) through (0.70) prove that $\sum_I$ and G_{jI} are orthogonal to the hourglass field:

$$\dot{u}_{iI}^{HG} \sum_I = 0 \tag{0.71}$$

$$\dot{u}_{iI}^{HG} G_{jI} = 0 \tag{0.72}$$

Furthermore, it can be shown that the gradient operator is a linear combination of the volumetric base vectors, Λ_{iI}, so Equation (0.72) can be written as:

$$\dot{u}_{iI}^{HG} \Lambda_{iI} = 0 \tag{0.73}$$

where no sum is implied on the lower case indices of Equation (0.73).

Equation (0.71) and (0.73) show that the hourglass field is orthogonal to all the base vectors in Table 7-2 except the hourglass base vectors. Therefore $\dot{u}_{iI}^{HG}$ may be expanded as a linear combination of the hourglass vectors as follows:

$$\dot{u}_{iI}^{HG} = \frac{1}{\sqrt{8}} \dot{q}_{i\alpha} \Gamma_{\alpha I} \tag{0.74}$$

The hourglass nodal velocities are represented by $\dot{q}_{i\alpha}$ above (the leading constant is added to normalize $\Gamma_{\alpha I}$). The range on the lower case Greek subscript, α, is 1 to 4 (the number of hourglass modes).

We now define the hourglass shape vector $\gamma_{\alpha I}$ such that:

$$\dot{q}_{i\alpha} = \frac{1}{\sqrt{8}} \dot{u}_{iI} \gamma_{\alpha I} \tag{0.75}$$

By substituting Equations (0.67), (0.70), and (0.75) into Equation (0.74), then multiplying by $\Gamma_{\alpha I}$ and using the orthogonality of the base vectors, we obtain the following:

$$\dot{\bar{u}}_{iI} \Gamma_{\alpha I} - \dot{\bar{u}}_{i,j} x_{jI} \Gamma_{\alpha I} = \dot{u}_{iI} \gamma_{\alpha I} \tag{0.76}$$

With the definition of the mean velocity gradient, Equation (0.76), we can eliminate the nodal velocities above. As a result, we can compute $\gamma_{\alpha I}$ from the following expression:

$$\gamma_{\alpha I} = \Gamma_{\alpha I} - \frac{1}{V} G_{iI} x_{iJ} \Gamma_{\alpha J} \tag{0.77}$$

The difference between the hourglass base vectors $\Gamma_{\alpha I}$ and the hourglass shape vectors $\gamma_{\alpha I}$ is very important. They are identical if and only if the hexahedron is a parallelepiped (the second term on the right hand side of Equation (0.77) vanishes for a parallelepiped). For a general shape hexahedron, $\Gamma_{\alpha I}$ is orthogonal to the gradient operator, G_{iI}, while $\gamma_{\alpha I}$ is orthogonal to the linear velocity field $\dot{u}_{iI}^{LIN}$. While $\Gamma_{\alpha I}$ defines the hourglass pattern, $\gamma_{\alpha I}$ is necessary to accurately detect hourglassing.

For the purpose of controlling the hourglass modes, we define generalized forces $Q_{i\alpha}$, which are conjugate to $\dot{q}_{i\alpha}$ so that the rate of work is:

$$\dot{u}_{iI} f_{iI}^{HG} = \frac{1}{2} Q_{i\alpha} \dot{q}_{i\alpha} \tag{0.78}$$

for arbitrary $\dot{u}_{iI}$. Using Equation (0.75), it follows that the contribution of the hourglass resistance to the nodal forces is given by:

$$f_{iI}^{HG} = \frac{1}{2} Q_{i\alpha} \gamma_{\alpha I} \tag{0.79}$$

The hourglass generalized forces $Q_{i\alpha}$ represent element state variables. There are two kinds of hourglass resistance algorithms that you can use, stiffness (K) or damping (V). Or, you can use a combination of the two. We express the combination of the two by:

$$Q_{i\alpha} = Q_{i\alpha}^{K} + Q_{i\alpha}^{V} \tag{0.80}$$

In terms of adjustable stiffness (κ) and viscosity (ε) factors, these resistances are given by:

$$\dot{Q}_{i\alpha}^{K} = \frac{\kappa}{2} 2\hat{\mu} \frac{G_{kI} G_{kI}}{V} \dot{q}_{i\alpha} \tag{0.81}$$

$$Q_{i\alpha}^{V} = \varepsilon \sqrt{min(0, 2\hat{\mu}) M} \cdot \dot{q}_{i\alpha} \tag{0.82}$$

where M is the mass of the element and $\hat{\mu}$ is an effective Lame constant. We discuss below how the effective Lame constants are determined in a general manner for the element. Note that the stiffness expression is a rate expression, implying that the stiffness hourglass resistance forces must be integrated and stored as an element state variable.

Observe that the nodal anti-hourglass forces of Equation (0.79) have the shape of $\gamma_{\alpha I}$ rather than $\Gamma_{\alpha I}$.

This fact is essential since the anti-hourglass forces should be orthogonal to the linear velocity field, so that no energy is transferred to or from the rigid body and uniform strain modes by the anti-hourglassing scheme.

The stiffness based hourglass approach is preferred to the viscous based approach. It generally proves to be more accurate and stable.

7.3 3D Uniform Gradient 4 Node Quadrilateral Membrane with Hourglass Control

7.3.1 Coordinate Systems

A local orthogonal coordinate system is defined for the membrane element and assumed to be uniform over the element. The base vectors in the local system are denoted by $\underset{\sim}{t}_i$. A compact indicial notation is used to define the local system: t_{ij}. The matrix t_{ij} gives the direction cosine matrix for the transformation from the global 3D Cartesian coordinate system to the local 2D element Cartesian coordinate system. The second index of t_{ij} indicates the base vector number and the first index indicates the component of the base vector in the global 3D coordinate system (i.e. the columns of the matrix are the base vectors). The third element base vector, t_{i3}, is always the outward normal for the element defined by the counter-clockwise numbering of the four nodes. Figure 7-3 shows a typical membrane element and the construction of the local coordinate system from the mid-chords of the element.

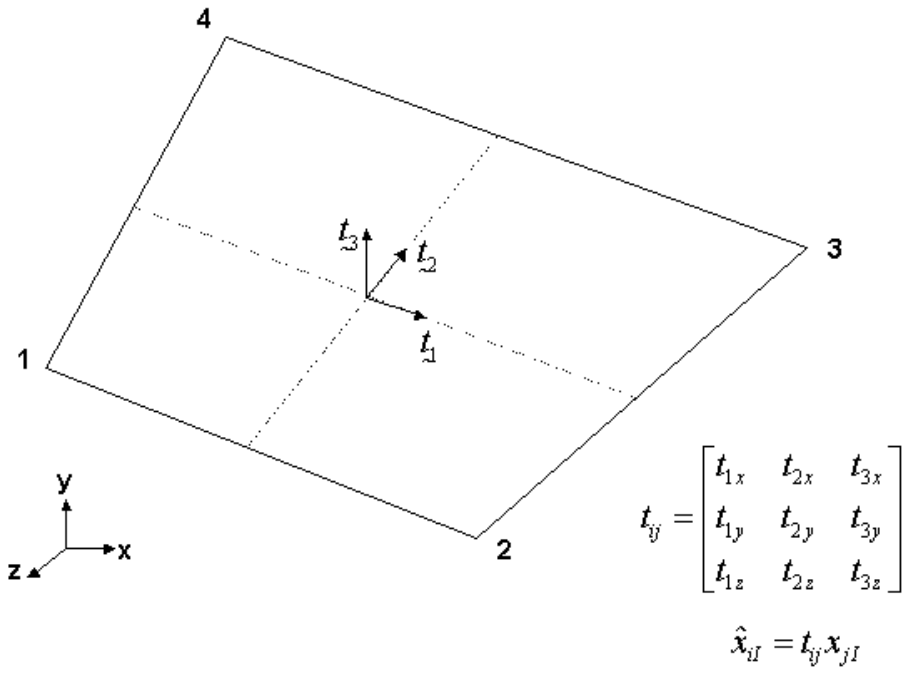


Figure 7-3. Membrane Local Coordinate System.

The first operation performed on the element in its initialization is to compute the local coordinate system of the element, t_{ij} based upon the mid-chords shown in Figure 7-3. We compute the initial coordinate transformation based purely upon the element geometry. This direction cosine matrix is used to compute the initial state quaternion, $\{q\}^{old}$ using the framework mathematical method to convert a direction cosine to a quaternion.

7.3.2 Projected Element

The membrane element is written as a uniform strain element. In general, the 4-Node, 3D, membrane element will have a warped geometry. The warped geometry vastly complicates integrals over the volume of the element. The fundamental simplifying assumption of this element is that we can project the warped three-dimensional element onto the plane defined by the uniform normal to the element, t_{i3}. The element domain is defined as the projection of the warped element geometry onto the plane defined by the mean normal.

The element formulation depends upon keeping track of the local coordinate system in the projected element. This coordinate system is stored in a state quaternion. In the initialization of the element we computed the initial orientation of the projected coordinate system based purely upon the element geometry. As we march through time and the element deforms, we update the coordinate system using the following algorithm.

1. Compute the current normal to the deformed element by crossing the mid-chords.
2. Recover the old direction cosine matrix t_{ij}^{old} from the old state quaternion $\{q\}^{old}$ using the framework mathematical method to compute a direction cosine matrix from a quaternion.
3. Calculate the cross product of the old normal (the 3rd column of t_{ij}^{old}) with the new normal computed in step 1 above. This represents the rotation vector for the local coordinate system over this time increment.
4. Update the state quaternion to obtain the new values $\{q\}^{new}$ using the framework mathematical method for quaternion updates.
5. Recover the new, update direction cosine matrix from the new quaternion.

Once we have computed the new local coordinate system of the element, t_{ij}, we rotate the gathered element coordinates and velocities into the local system:

$$\hat{x}_{iI} = t_{ij} x_{jI} \qquad (0.83)$$

$$\hat{\dot{u}}_{iI} = t_{ij} \dot{u}_{jI} \qquad (0.84)$$

Once the local kinematic variables are computed using Equations (0.83) and (0.84), the element formulation proceeds using the two dimensional, uniform gradient element formulation to compute the internal force contribution for the element. These element internal forces are in the local element coordinate system and must be transformed to the global, 3D, Cartesian coordinate system for assembly:

$$f_{iI} = t_{ji} \hat{f}_{jI} \qquad (0.85)$$

7.3.3 Element Kinematics and Interpolation

The element formulation follows exactly the description provided in the two dimensional uniform gradient formulation. The same set of nodal shape functions are used to define a position of a point within an element:

$$\Phi_I = \frac{1}{4}\sum_I + \frac{1}{2}\xi\Lambda_{1I} + \frac{1}{2}\eta\Lambda_{2I} + \xi\eta\Gamma_I$$

7.3.4 Gradient and Body Operators

We define the discrete gradient operator as in the two dimensional element (except all computations are performed in the local element coordinate system):

$$G_{iI} = \int_A \Phi_{I,i} dA \qquad (0.86)$$

The mean velocity gradient is given by:

$$\dot{\bar{u}}_{i,j} = \frac{1}{A}\dot{\hat{u}}_{iI} G_{jI} \qquad (0.87)$$

Note, the mean velocities are in the local coordinate system and we have dropped the hat "^" because the notation becomes too unwieldy to have both "barred" and "hat" quantities together.

The nodal forces in the local system are defined by:

$$\hat{f}_{iI} = \bar{N}_{ij} G_{jI} \qquad (0.88)$$

where the element uniform stress resultant tensor, $\bar{N}_{ij}$, is defined in the local coordinate system and represents the integration of the material point stresses through the thickness of the membrane element (see below). Note: we have dropped the hat "^" designation for notational simplicity. The body operator, B_I, for this element is identical to the one for the 2D quadrilateral element.

$$B_I = \int_{V_e} \Phi_I dV = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} \Phi_I \, det(J) d\xi d\eta$$

7.3.5 Hourglass Control

The hourglass operator for the element (in the local coordinate system) is computed exactly as for the plane stress quadrilateral:

$$\dot{\hat{q}}_i = \frac{1}{2}\dot{\hat{u}}_{iI}\gamma_I \qquad (0.89)$$

$$\hat{f}_{iI}^{HG} = \frac{1}{2} Q_i \gamma_I \qquad (0.90)$$

The hourglass generalized forces Q_i represent element variables. There are two kinds of hourglass resistance algorithms that you can use, stiffness (K) or damping (V). Or, you can use a combination of the two. We express the combination of the two by:

$$Q_i = Q_i^K + Q_i^V \qquad (0.91)$$

In terms of adjustable stiffness (κ) and viscosity (ε) factors, these resistances are given by:

$$\dot{Q}_i^K = \frac{\kappa}{2} 2\mu \frac{G_{kl} G_{kl}}{A} \cdot h \cdot \dot{q}_i \qquad (0.92)$$

$$Q_i^V = \varepsilon \sqrt{min(0, 2\mu) M} \cdot h \cdot \dot{q}_i \qquad (0.93)$$

where h is the total thickness of the membrane (the sum of the thickness over all layers).

Table 7-7. 3D Membrane Element Modes.

Mode Description	Number of Modes
Rigid Body Translation	3
Rigid Body Rotation	3
Strain Modes	3
In Plane Hourglass Modes	2
Butterfly Hourglass Mode	1

The 4-Node quadrilateral in three dimensions has 12 degrees of freedom and hence must have 12 modes represented. Up to this point in the discussion of the element, we have accounted for 11 of the modes in the element (see Table 7-7). There is still one un-resisted mode in the element. This is the so-called "butterfly" mode of the element shown in Figure 7-4.

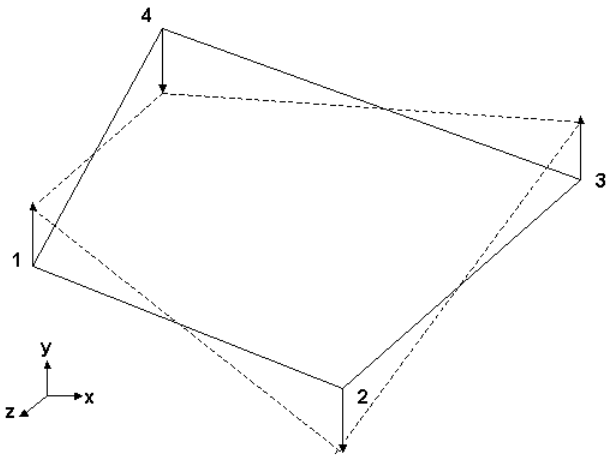


Figure 7-4. Butterfly Mode of the Membrane Element.

The butterfly mode is a zero energy mode just like the element hourglass modes. Fortunately, we have a means to measure the amount of the displacement field of the element represented by this mode. When we applied the hourglass operator to the displacement field (in the local coordinate system) using Equation (0.89), we compute the 3rd component representing the out of plane hourglassing (the butterfly mode). We simply compute hourglass resistances and forces using Equations (0.90) through (0.93) for the butterfly mode just like the other two hourglass modes.

7.3.6 Integration Through the Membrane Thickness

The membrane element has some finite thickness that is not accounted for in the 2D plane stress element formulation in the projected and transformed element. In general, we can allow a layered section so that the model can accommodate a lay-up of different materials through the thickness as shown in Figure 7-5 for the case of 3 distinct layers. NOTE: at the present time, Nastran and Autodesk Explicit do not provide input for layered sections for membranes.

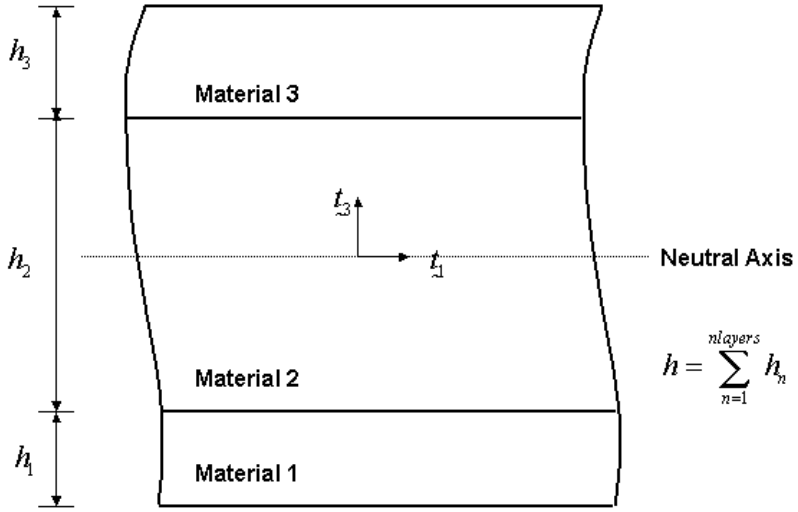


Figure 7-5. Layered Section for Three Layers with Different Thickness and Materials in Each Layer.

The membrane force resultants in the material configuration follow by integrating the material point stresses through the thickness.

$$\overline{N}_{\alpha\beta} = \int_h \sigma_{\alpha\beta} d\hat{x}_3 \qquad (0.94)$$

where the Greek indices indicate that the quantities are in the material configuration (and have a range of one to two). The thinning strain in each layer, ε_{33}, is computed by the plane stress material model and applied to the thickness to account for thinning effects in finite deformations.

The stress resultant in the element local configuration by transforming from the material frame to the element local frame of reference:

$$N_{ij} = R_{\alpha i}^{t+\Delta t} \overline{N}_{\alpha\beta} R_{\beta j}^{\ t+\Delta t} \qquad (0.95)$$

7.4 3D Uniform Gradient 4-Node Quadrilateral Shell Element with Hourglass Control

This element makes use of the 4-node quadrilateral membrane element formulation for many of the element operators. It augments that element formulation with bending terms to capture the moment-curvature aspects of shell behavior.

7.4.1 Element Degrees of Freedom

The shell element has 3 spatial displacement degrees of freedom at each node representing the displacement vector:

$$u_I = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \tag{0.96}$$

In the dynamics formulation, we have spatial accelerations, mean velocities, state velocities and incremental displacements for each node, which represent the time derivatives of the displacement degrees of freedom at each node.

We also have rotational degrees of freedom at each node connected to a shell element. The rotation at each node represents the total rotation at the node:

$$\theta_I = \begin{Bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} \tag{0.97}$$

In an incremental dynamics algorithm, the total rotations are in fact a vector but we cannot simply integrate them by adding up the incremental rotations. We must perform a quaternion update to integrate them. Therefore, the total rotation is stored in a quaternion:

$$\{\theta\} = \begin{Bmatrix} sin\frac{\theta}{2} \cdot \hat{n} \\ cos\frac{\theta}{2} \end{Bmatrix} \tag{0.98}$$

where the usual quaternion conventions represent the quaternion as a vector part and a scalar part. Equation (0.98) defines a rotation θ about a direction defined by the unit normal $\hat{n}$.

The angular accelerations, velocities and incremental rotations are all vectors (i.e. not quaternions). The explicit dynamics Physics procedure computes the following rotational kinematics:

$$\ddot{\theta}_I = {M_I}/{I_I} \text{ (no sum on I)} \tag{0.99}$$

$$\dot{\theta}_I^{t+\Delta t/2} = \dot{\theta}_I^{t-\Delta t/2} + \frac{\left(\Delta t_k + \Delta t_{k+1}\right)}{2}\ddot{\theta}_I \tag{0.100}$$

$$\dot{\theta}_I^{t+\Delta t} = 2\dot{\theta}_I^{t+\Delta t/2} - \dot{\theta}_I^{t} \tag{0.101}$$

$$\Delta\theta_I = \Delta t_{k+1} \cdot \dot{\theta}_I^{t+\Delta t/2} \tag{0.102}$$

$$\{\theta\}_I^{t+\Delta t} = \Delta\theta_I \{\theta\}_I^{t} \tag{0.103}$$

where I_I represents the moments of inertia at each node and we note that Equation (0.103) represents a quaternion update for the total rotation at the node.

In general, the finite element analyst would be completely baffled by providing quaternions as output for the nodal rotations. They simply want us to output the rotation vector defined at each node by the quaternion represented by Equation (0.103). Recovering this rotation from quaternion is quite straightforward albeit somewhat computationally expensive (because it requires the computation of an arctangent). We recover the vector components of rotation for output purposes only.

7.4.2 Coordinate Systems

A local orthogonal coordinate system is defined for the shell element and assumed to be uniform over the element. The base vectors in the local system are denoted by $\underset{\sim}{t}_i$. A compact indicial notation is used to define the local system: t_{ij}. The matrix t_{ij} gives the direction cosine matrix for the transformation from the global 3D Cartesian coordinate system to the local 2D element Cartesian coordinate system. The second index of t_{ij} indicates the base vector number and the first index indicates the component of the base vector in the global 3D coordinate system (i.e. the columns of the matrix are the base vectors). The third element base vector, t_{i3}, is always the outward normal for the element defined by the counter-clockwise numbering of the four nodes. Figure 7-6 shows a typical membrane element and the construction of the local coordinate system from the mid-chords of the element.

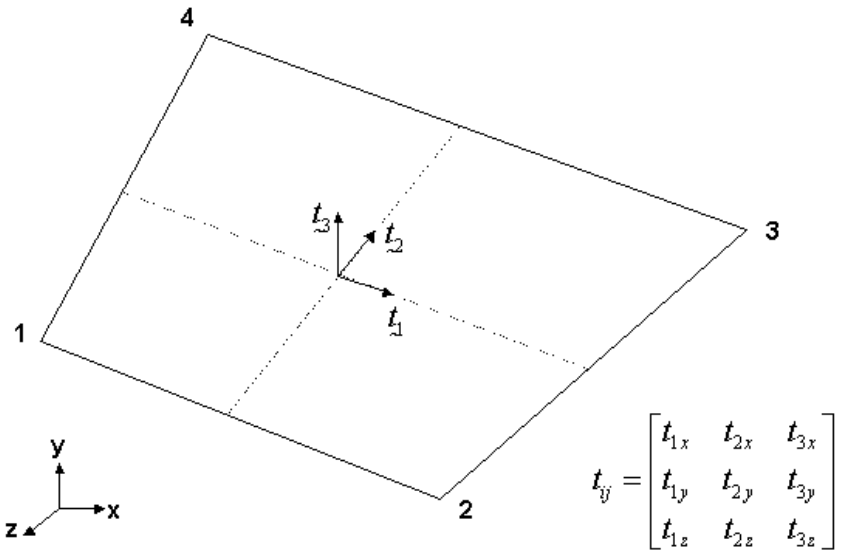


Figure 7-6. Membrane Local Coordinate System.

The first operation performed on the element in its initialization is to compute the local coordinate system of the element, t_{ij} based upon the mid-chords shown in Figure 7-6. We compute the initial coordinate transformation based purely upon the element geometry. This direction cosine matrix is used to compute the initial state quaternion, $\{q\}^{old}$ using the framework mathematical method to convert a direction cosine to a quaternion.

7.4.3 Projected Element

The shell element is written as a uniform strain element. In general, the 4-Node, 3D, shell element will have a warped geometry. The warped geometry vastly complicates integrals over the volume of the element. The fundamental simplifying assumption of this element is that we can project the warped three-dimensional element onto the plane defined by the uniform normal to the element, t_{i3}. The element domain is defined as the projection of the warped element geometry onto the plane defined by the mean normal.

The element formulation depends upon keeping track of the local coordinate system in the projected element. This coordinate system is stored in a state quaternion. In the initialization of the element we computed the initial orientation of the projected coordinate system based purely upon the element geometry. As we march through time and the element deforms, we update the coordinate system using the following algorithm.

1. Compute the current normal to the deformed element by crossing the mid-chords.
2. Recover the old direction cosine matrix t_{ij}^{old} from the old state quaternion $\{q\}^{old}$ using the framework mathematical method to compute a direction cosine matrix from a quaternion.
3. Calculate the cross product of the old normal (the 3rd column of t_{ij}^{old}) with the new normal computed in step 1 above. This represents the rotation vector for the local coordinate system over this time increment.
4. Update the state quaternion to obtain the new values $\{q\}^{new}$ using the framework mathematical method for quaternion updates.
5. Recover the new, update direction cosine matrix from the new quaternion.

Once we have computed the new local coordinate system of the element, t_{ij}, we rotate the gathered element nodal values (coordinates, spatial displacement increments, rotation increments) into the local system:

$$\hat{x}_{iI} = t_{ij} x_{jI} \qquad (0.104)$$

$$\Delta\hat{u}_{iI} = t_{ij} \Delta u_{jI} \qquad (0.105)$$

$$\Delta\hat{\theta}_{iI} = t_{ij} \Delta\theta_{jI} \qquad (0.106)$$

Once the local kinematic variables are computed using Equations (0.83) and (0.84), the element formulation proceeds using the two dimensional, uniform gradient element formulation to compute the internal force contribution for the element. These element internal forces are in the local element coordinate system and must be transformed to the global, 3D, Cartesian coordinate system for assembly:

$$f_{iI} = t_{ji} \hat{f}_{jI} \qquad (0.107)$$

7.4.4 Element Kinematics and Interpolation

The element formulation follows exactly the description provided in the two dimensional uniform gradient formulation. The same set of nodal shape functions are used to define a position of a point within an element:

$$\Phi_I = \frac{1}{4}\Sigma_I + \frac{1}{2}\xi\Lambda_{1I} + \frac{1}{2}\eta\Lambda_{2I} + \xi\eta\Gamma_I \qquad (0.108)$$

7.4.5 Gradient and Body Operator

We define the discrete gradient operator as in the two dimensional element (except all computations are performed in the local element coordinate system):

$$G_{iI} = \int_A \Phi_{I,i}\, dA \qquad (0.109)$$

The mean velocity gradient is given by:

$$\dot{\bar{u}}_{i,j} = \frac{1}{A}\dot{\hat{u}}_{iI}G_{jI} \qquad (0.110)$$

Note, the mean velocities are in the local coordinate system and we have dropped the hat "^" because the notation becomes too unwieldy to have both "barred" and "hat" quantities together.

The nodal forces in the local system are defined by:

$$\hat{f}_{iI} = \bar{N}_{ij}G_{jI} \qquad (0.111)$$

where, the element uniform stress resultant tensor, $\bar{N}_{ij}$, is defined in the local coordinate system and represents the integration of the material point stresses through the thickness of the membrane element (see below). Note: we have dropped the hat "^" designation for notational simplicity.

The body operator, B_I, for this element is identical to the one for the 2D quadrilateral element.

$$B_I = \int_{V_e} \Phi_I dV = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} \Phi_I \, det(J) d\xi d\eta$$

7.4.6 Hourglass Control

The hourglass operator for the element (in the local coordinate system) is computed exactly as for the plane stress quadrilateral:

$$\dot{\hat{q}}_i = \frac{1}{2}\dot{\hat{u}}_{iI}\gamma_I \qquad (0.112)$$

$$\hat{f}_{iI}^{HG} = \frac{1}{2}Q_i\gamma_I \qquad (0.113)$$

The hourglass generalized forces Q_i represent element variables. There are two kinds of hourglass resistance algorithms that you can use, stiffness (K) or damping (V). Or, you can use a combination of the two. We express the combination of the two by:

$$Q_i = Q_i^K + Q_i^V \tag{0.114}$$

In terms of adjustable stiffness (κ) and viscosity (ε) factors, these resistances are given by:

$$\dot{Q}_i^K = \frac{\kappa}{2} 2\mu \frac{G_{kl} G_{kl}}{A} \cdot h \cdot \dot{q}_i \tag{0.115}$$

$$Q_i^V = \varepsilon \sqrt{min(0, 2\mu) M} \cdot h \cdot \dot{q}_i \tag{0.116}$$

where h is the total thickness of the membrane (the sum of the thickness over all layers).

The hourglass pattern can appear in both the displacement degrees of freedom and the rotational degrees of freedom. Hence we apply the hourglass control algorithm to the in-plane (projected) displacements and rotations. Therefore, we constrain two in-plane displacement hourglass modes and two in-plane rotational hourglass modes.

7.4.7 Transverse Shear Constraints

We apply the transverse shear constraints at each edge element by penalizing the transverse shear strain at the mid-edge. At each edge of the element, we introduce a local coordinate system t_{ij}^e similar to the local system defined for the projected element. The first index (i) indicates the global Cartesian direct and the second index (j) indicates the unit vector number. The local system is constructed so that the 1st basis vector points along the edge, the 2nd basis vector points out of the element, and the 3rd basis vector points in the direction of the projected normal of the element.

The increment in transverse shear strain at the mid-edge is given by:

$$\Delta\gamma_e = \Delta\phi_e - \Delta\theta_e$$

where $\Delta\phi_e$ is the rotation about the t_{2j}^e axis defined by averaging the nodal rotations of the two nodes attached to the edge:

$$\Delta\phi_e = \frac{1}{2} t_{2j}^e \Delta\phi_{jl}$$

and $\Delta\theta_e$ is the rigid body rotation of the edge about the t_{2j}^e axis defined by the spatial displacements of the two nodes attached to the edge:

$$\Delta\theta_e = \frac{1}{L_e} t_{i3}^e \left(\Delta u_{i1} - \Delta u_{i2}\right)$$

Note that this definition of $\Delta\theta_e$ assumes small incremental rigid body rotations.

For each edge, we integrate the total transverse shear as:

$$\gamma_e^{t+\Delta t} = \gamma_e^t + \Delta\gamma_e$$

The penalty transverse shear force is calculated in terms of a tunable stiffness factor κ_{TS} as:

$$f^{TS} = \kappa_{TS} K_{TS} \gamma_e^{t+\Delta t}$$

where the determination of the overall stiffness term, K_{TS}, is discussed below. The virtual work contribution is $\kappa_{TS} K_{TS} \gamma_e \delta\gamma_e$ from which it follows that the nodal force and moment contributions to the two edge nodes due to the edge transverse shear are:

$$f_{il}^{TS} = f^{TS} t_{3l}^3 A_l^3 \qquad m_{il}^{TS} = m^{TS} t_{2i}^e$$

where A_l^3 is a simple operator used to ensure the forces are applied in the right directions.

$$A_1^e = -1, \quad A_2^e = +1$$

The transverse shear stiffness K^{TS} is chosen to reasonably approximate "linear curvature" stiffness for thin elements and "pure shear" stiffness for thick elements. The formulation should transition smoothly between these two regimes. This is accomplished by the stiffness as follows:

$$K^{TS} = min\left(\mu h W, \frac{E h^3 W}{L_e^3}\right)$$

This can be recast into terms that are more convenient in the form:

$$K^{TS} = 2\mu h W \; min\left(\frac{1}{2}, \left(\frac{h}{L_e}\right)^2\right)$$

7.4.8 Drilling Degrees of Freedom

The rotational degrees of freedom normal to the shell elements have so far not appeared anywhere in our projected element formulation. Therefore, in general they are completely free of any stiffness associated with them especially in a flat geometry. We refer to these as the "drilling" degrees of freedom. In reality, this causes no real problem since they have no effect on the solution. However, if the user requests output of the rotation vectors, they should not simply have arbitrary values for the out-of-plane components. Therefore, we define four rotational moments (one for each node of the element) about the drilling direction. These moments are calculated by penalizing the difference between the incremental drilling rotation and the material rotation increment defined in the polar decomposition of the membrane deformation gradient. The four drilling moment resistances are given by:

$$m_l^{t+\Delta t} = m_l^t + \kappa_D 2\mu \cdot L_{el}\left(\Delta_{3l} - \Delta\Omega\right)$$

where κ_D is a small tunable parameter to define the penalty stiffness.

7.4.9 Degree of Freedom Count

Table 7-8 provides an accounting of all the degrees of freedom found in the four-node quadrilateral and three-node triangular shell elements described in this chapter. Table 7-9 summarizes the constraints applied to the shell elements.

Table 7-8. 3D Shell Element Modes and Constraints.

Mode Description	Quadrilateral Number of Modes	Triangle Number of Modes
Rigid Body Translation	3	3
Rigid Body Rotation	3	3
Membrane Strain Modes	3	3
Bending Strain Modes	3	3
In Plane Hourglass Modes	2	0
Bending Hourglass Modes	2	0
Drilling Rotations	4	3
Edge Transverse Shear	4	3
Total Number DOFS =	24	18

Table 7-9. Constraint Stiffness Factors.

Constraint	Default Scale Factor	Stiffness Type	Stiffness Factor
Displacement HG	.05	hypoelastic	$(2\hat{\mu}h)\frac{1}{g}$
Bending HG	.05	hypoelastic	$(2\hat{\mu}h)\left(\frac{h}{g}\right)^2$
Drilling Rotation	.0005	constant	$(2\hat{\mu}h)(Ag)$
Transverse Shear	5/6	constant	$(2\hat{\mu}h)W \cdot min\left(\frac{1}{2}, \left(\frac{h}{L}\right)^2\right)$

7.4.10 The Layered Section and Section Resultants

The shell element has some finite thickness that is not accounted for in the 2D plane stress element formulation in the projected and transformed element. We allow a layered section so that the model can accommodate a lay-up of different materials through the thickness as shown in Figure 7-7 for the case of 3 distinct layers. Such a three layer composite section would be defined using the PCOMP property option for the element.

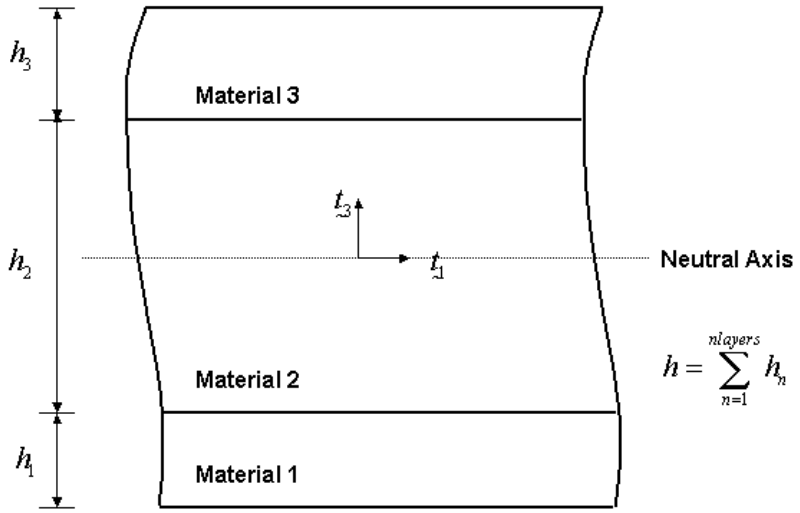


Figure 7-7. Layered Section for Three Layers with Different Thickness and Materials in Each Layer.

The shell force resultants in the material configuration follow by integrating the material point stresses through the thickness.

$$\overline{N}_{\alpha\beta} = \int_h \sigma_{\alpha\beta} d\hat{x}_3 \qquad (0.117)$$

where the Greek indices indicate that the quantities are in the material configuration (and have a range of one to two). The thinning strain in each layer, ε_{33}, is computed by the plane stress material model and applied to the thickness to account for thinning effects in finite deformations.

The stress resultant in the element local configuration is determined by transforming from the material frame to the element local frame of reference:

$$N_{ij} = R_{\alpha i}^{t+\Delta t} \overline{N}_{\alpha\beta} R_{\beta j}^{t+\Delta t} \qquad (0.118)$$

The shell moment resultants in the material configuration follow by integrating the material point stresses times the distance from the neutral axis of the material point through the thickness.

$$\overline{M}_{\alpha\beta} = \int_h \sigma_{\alpha\beta} x_3 d\hat{x}_3 \qquad (0.119)$$

where the Greek indices indicate that the quantities are in the material configuration (and have a range of one to two). The thinning strain in each layer, ε_{33}, is computed by the plane stress material model and applied to the thickness to account for thinning effects in finite deformations.

The moment resultant in the element local configuration is determined by transforming from the material frame to the element local frame of reference:

$$M_{ij} = R_{\alpha i}^{t+\Delta t} \overline{N}_{\alpha\beta} R_{\beta j}^{t+\Delta t} \qquad (0.120)$$

7.4.11 Interlaminar Shear Stresses

When a shell section consists of more than one layer, we can determine the interlaminar shear stresses between the plies from element transverse shear forces computed for the shell section.

Consider a simple beam with a uniform depth (out of the plane) with multiple layers, each of which has different material properties. Hence the Young's Modulus, E, varies throughout the layers. The origin of the z coordinate direction is assumed to be at the neutral axis of the beam. For a beam with N layers, the z coordinate of the top of each layer is defined by z_i. Therefore the location of the bottom free surface is denoted z_0 and the top free surface is denoted z_N.

We denote the Young's Modulus in the individual layers by E_i and define the he mean value of Young's Modulus, $\bar{E}$, as:

$$\bar{E} = \frac{\sum_{i=1}^{N} E_i t_i}{\sum_{i=1}^{N} t_i} \qquad (0.121)$$

where the thickness of each layer is defined by t_i and the total thickness of the layered beam, t, is given by:

$$t = \sum_{i=1}^{N} t_i \qquad (0.122)$$

The area moment of inertia of the beam cross section is given by:

$$I = \frac{t^3}{12} \qquad (0.123)$$

where we have assumed a unit width (out of the plane) for the beam.

The governing equilibrium equations are:

$$\frac{\partial \tau_{xz}}{\partial z} + \frac{\partial \sigma_x}{\partial x} = 0 \qquad (0.124)$$

$$V + \frac{\partial M}{\partial x} = 0 \qquad (0.125)$$

and for pure bending, σ_x, may be expressed as:

$$\sigma_x = -\frac{E\left(\bar{z} - z\right)}{\bar{E}I} M \qquad (0.126)$$

where $\bar{z}$ represents the location of the bending center of the cross-section and is defined by:

$$\bar{z} = \frac{\int Ezdz}{\int Edz} = \frac{\sum_{i=1}^{N} E_i t_i \left(z_i - \frac{t_i}{2} \right)}{\sum_{i=1}^{N} E_i t_i} \qquad (0.127)$$

and $t_i = z_i - z_{i-1}$ represents the thickness of layer i.

Differentiating Equation (0.126) with respect to x and combining with Equation (0.124):

$$\frac{\partial \tau_{xz}}{\partial z} = -\frac{E\left(\bar{z} - z\right)}{\bar{E}I} \frac{\partial M}{\partial x} \qquad (0.128)$$

Using Equation (0.125) in Equation (0.128):

$$\frac{\partial \tau_{xz}}{\partial z} = \frac{E\left(\bar{z} - z\right)}{\bar{E}I} V \qquad (0.129)$$

Integrating Equation (0.129) with respect to z gives:

$$\tau_{xz} = \frac{V}{\bar{E}I} \int E\left(\bar{z} - z\right) dz \qquad (0.130)$$

Now, in Equation (0.130) we cannot bring E outside the integral because if depends upon z as each layer has a different value for Young's modulus. However, in any individual layer E is constant and we can perform the integration. Therefore, in any ply:

$$\tau_{xz} = C_i + \frac{VE_i}{\bar{E}I} \left(\bar{z}z - \frac{1}{2} z^2 \right) \qquad z_{i-1} \le z \le z_i \qquad (0.131)$$

Equation (0.131) can be used to determine the constants C_i for each of the layers. Consider a layered beam with N layers. At the bottom surface $i = 1$, $z = z_0$ and $\tau_{xz} = 0$ and using Equation (0.131) evaluated in the interval for $i = 1$:

$$C_1 = -\frac{VE_1}{\bar{E}I} \left(\bar{z}z_0 - \frac{1}{2} z_0^2 \right) \qquad (0.132)$$

At the top of the first ply, at the interface between $i = 1$ and $i = 2$, we have $z = z_1$ and Equation (0.131) can be evaluated again in the interval for $i = 1$ along with the C_1 constant given in Equation (0.132) to obtain:

$$\tau_{xz}\Big|_{z_1}^{1} = \frac{VE_1}{\bar{E}I} \left[\bar{z}\left(z_1 - z_0\right) - \frac{1}{2}\left(z_1^2 - z_0^2\right) \right] \qquad (0.133)$$

Now, at the interface between the first and second plies, we can use Equation (0.131) evaluated in the interval for $i=2$ and $z=z_1$ to evaluate the C_2 constant:

$$\tau_{xz}\Big|_{z_1}^{2} = C_2 + \frac{VE_2}{\bar{E}I}\left(\bar{z}z_1 - \frac{1}{2}z_1^2\right) \qquad (0.134)$$

Equations (0.133) and (0.134) are two independent expressions for the interlaminar shear stress at the top of ply 1 and we can determine the value of C_2 as:

$$C_2 = \frac{VE_1}{\bar{E}I}\left[\bar{z}\left(z_1 - z_0\right) - \frac{1}{2}\left(z_1^2 - z_0^2\right)\right] - \frac{VE_2}{\bar{E}I}\left(\bar{z}z_1 - \frac{1}{2}z_1^2\right) \qquad (0.135)$$

Repeating this process for the next ply, at the top of the second ply, at the interface between $i=2$ and $i=3$, we have $z=z_2$ and Equation (0.131) can be used evaluated again in the interval for $i=2$ along with the C_2 constant given in Equation (0.135) to obtain:

$$\tau_{xz}\Big|_{x_2}^{2} = \frac{VE_1}{\bar{E}I}\left[\bar{z}\left(z_1 - z_0\right) - \frac{1}{2}\left(z_1^2 - z_0^2\right)\right] + \frac{VE_2}{\bar{E}I}\left[\bar{z}\left(z_2 - z_1\right) - \frac{1}{2}\left(z_2^2 - z_1^2\right)\right] \qquad (0.136)$$

At the interface between the second and third plies, we can use Equation (0.131) evaluated in the interval for $i=3$ and $z=z_2$ to evaluate the C_3 constant:

$$\tau_{xz}\Big|_{x_2}^{3} = C_3 + \frac{VE_3}{\bar{E}I}\left(\bar{z}z_2 - \frac{1}{2}z_2^2\right) \qquad (0.137)$$

Again Equations (0.135) and (0.136) are two independent expressions for the interlaminar shear stress at the top of ply 2 and we can determine the value of C_3 as:

$$\begin{aligned} C_3 = {} & \frac{VE_1}{\bar{E}I}\left[\bar{z}\left(z_1 - z_0\right) - \frac{1}{2}\left(z_1^2 - z_0^2\right)\right] \\ & + \frac{VE_2}{\bar{E}I}\left[\bar{z}\left(z_2 - z_1\right) - \frac{1}{2}\left(z_2^2 - z_1^2\right)\right] \\ & - \frac{VE_3}{\bar{E}I}\left(\bar{z}z_2 - \frac{1}{2}z_2^2\right) \end{aligned} \qquad (0.138)$$

Repeating this process for the next ply, at the top of the third ply, at the interface between $i=3$ and $i=4$, we have $z=z_3$ and Equation (0.131) can be used evaluated again in the interval for $i=3$ along with the C_3 constant given in Equation (0.138) to obtain:

$$\tau_{xz}\Big|_{x_3}^{3} = \frac{VE_1}{\bar{E}I}\left[\bar{z}\left(z_1 - z_0\right) - \frac{1}{2}\left(z_1^2 - z_0^2\right)\right] + \frac{VE_2}{\bar{E}I}\left[\bar{z}\left(z_2 - z_1\right) - \frac{1}{2}\left(z_2^2 - z_1^2\right)\right] + \frac{VE_3}{\bar{E}I}\left[\bar{z}\left(z_3 - z_2\right) - \frac{1}{2}\left(z_3^2 - z_2^2\right)\right] \quad (0.139)$$

A clear pattern then emerges that allows us to write a concise formula to evaluate the interlaminar shear at the top of each ply:

$$\tau_{xz}\Big|_i = \frac{V}{\bar{E}I}\sum_{j=1}^{i} E_j\left[\bar{z}\left(z_j - z_{j-1}\right) - \frac{1}{2}\left(z_j^2 - z_{j-1}^2\right)\right] = \frac{V}{\bar{E}I}\sum_{j=1}^{i} E_j t_j\left[\bar{z} - \frac{1}{2}\left(z_j + z_{j-1}\right)\right] \quad (0.140)$$

7.4.11.1 Case 1 - Uniform Values of Young's Modulus

For the case of all uniform thickness layers and uniform values of Young's Modulus, E, our formula reproduces the classic quadratic shape of the shear distribution in a beam. Consider the case of 10 layers, each of thickness .1, a constant Young's Modulus of 1000, and a shear force of 50. Figure 7-8 shows the quadratic distribution of shear stress through the thickness that is recovered using Equation (0.140).

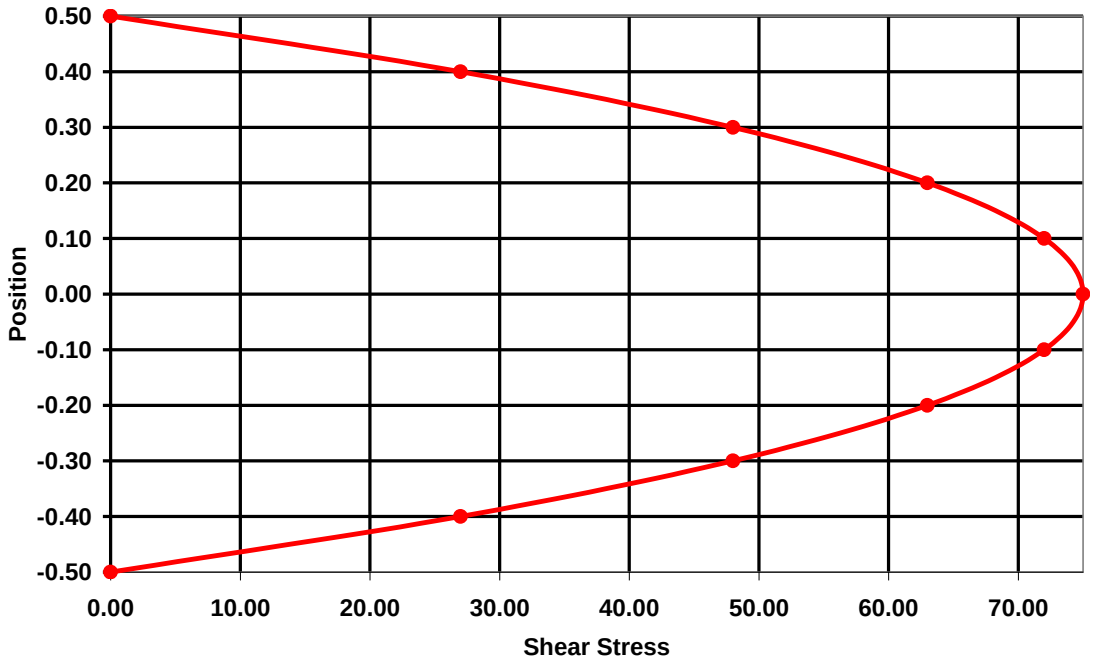


Figure 7-8. Shear Distribution for Uniform Thickness Layers, Young's Modulus and Shear Force.

7.4.11.2 Case 2 - Non-Uniform Young's Modulus, Stiff Outer Layers, Soft Inner Layers

Now, consider the case where there are again 10 layers of uniform 0.1 thickness. The first and last layer has Young's modulus of 1000. Layers 2 through 9 have a Young's modulus of 250. The shear force value is 50. Figure 7-9 shows the distribution of shear stress through the thickness that is recovered using Equation (0.140) for this case. We verify that the algorithm derived here correctly predicts zero shear stress on the free surface at the top layer of the cross-section.

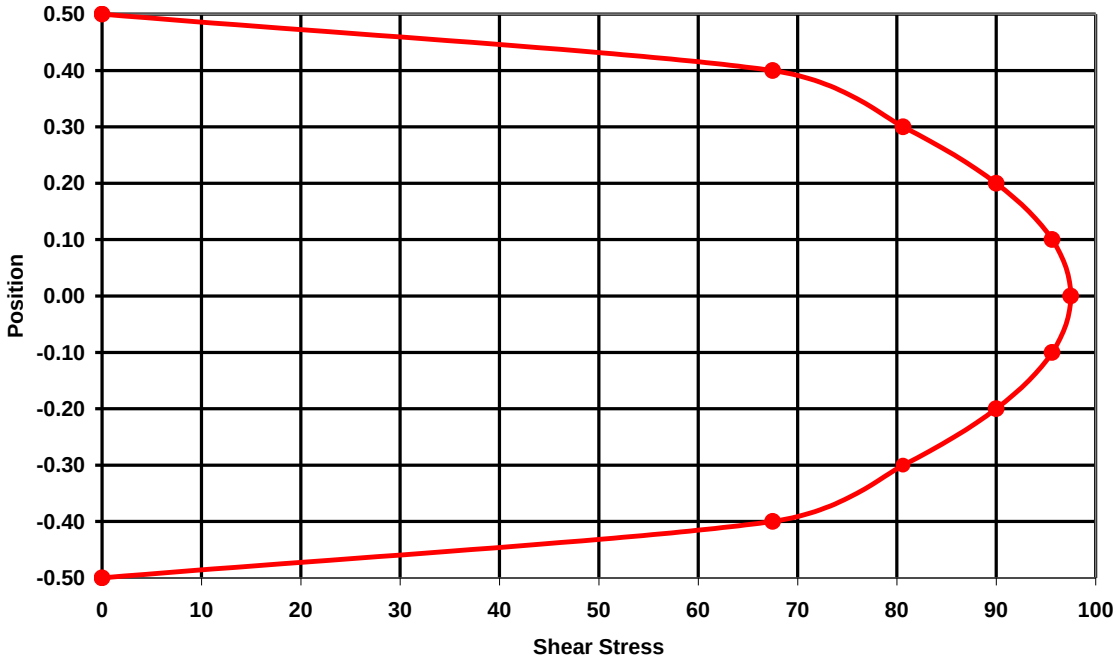


Figure 7-9. Shear Distribution for Non-Uniform Young's Modulus, Stiff Outer Layers and Soft Inner Layers.

7.4.11.3 Case 3 - Non-Uniform Young's Modulus, Soft Outer Layers, Stiff Inner Layers

Now, consider the case where there are again 10 layers of uniform 0.1 thickness. The first and last layer has Young's modulus of 250. Layers 2 through 9 have a Young's modulus of 1000. This is just the reverse of Case 2 above. The shear force value is 50. Figure 7-10 shows the distribution of shear stress through the thickness that is recovered using Equation (0.140) for this case. Again, we verify that the algorithm derived here correctly predicts zero shear stress on the free surface at the top layer of the cross-section.

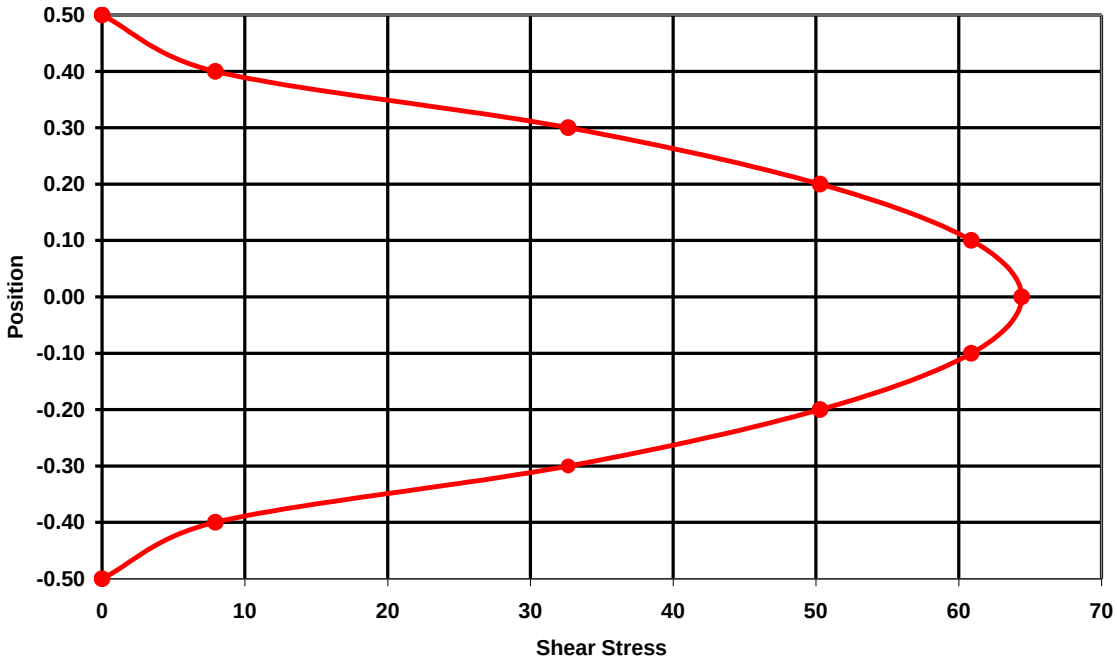


Figure 7-10. Shear Distribution for Non-Uniform Young's Modulus, Soft Outer Layers and Stiff Inner Layers.

7.4.11.4 Case 4 - Non-Uniform Young's Modulus, Five Stiff Layers Topped by Five Soft Layers

Now, consider the case where there are again 10 layers all with 0.1 thickness. The first five have Young's modulus of 1000. The last five layers have Young's modulus of 250. The shear force value is 50. Figure 7-11 shows the distribution of shear stress through the thickness that is recovered using Equation (0.140) for this case. Again, we verify that the algorithm derived here correctly predicts zero shear stress on the free surface at the top layer of the cross-section.

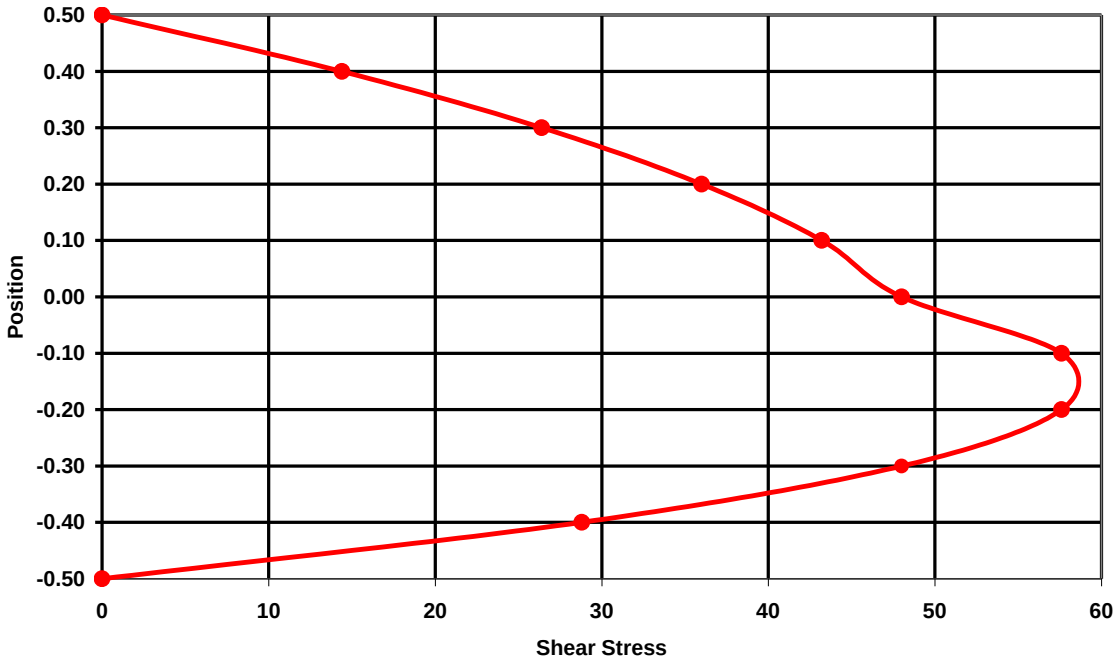


Figure 7-11. Shear Distribution for Five Stiff Layers Topped by Five Soft Layers.

7.5 3D 4-Node Tetrahedral Element

7.5.1 Element Node Ordering

The 4-node tetrahedral element has nodes designated by *i, j, k, m* as shown in Figure 7-12. The ordering of the node numbers must follow the "right-hand-rule" such that the rotation defined by traversing the first face (defined by node numbers *i, j, k*) produces a vector that points towards node *m*.

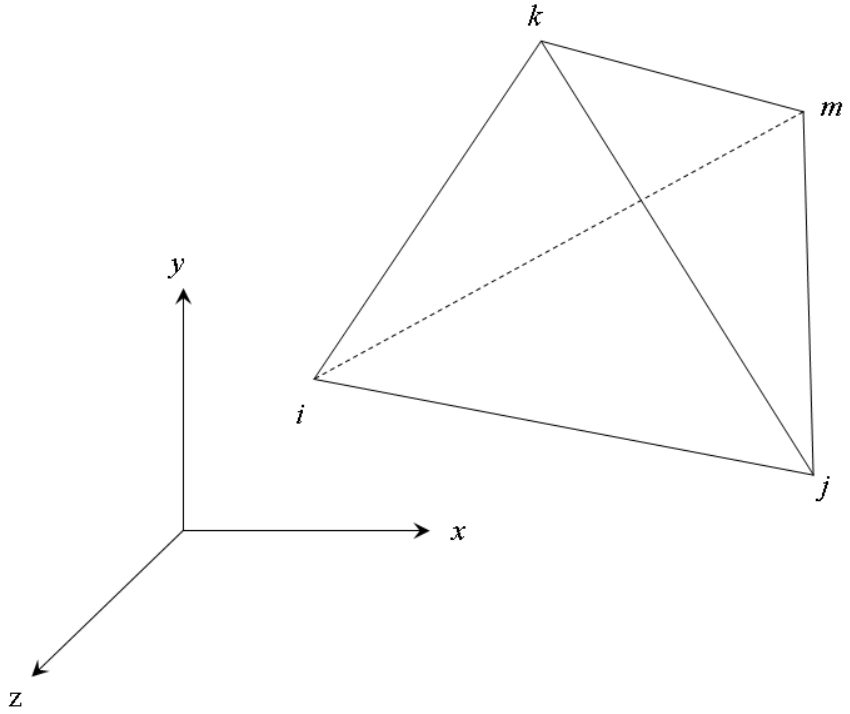


Figure 7-12. A 4-Node tetrahedral Element Defined by Nodes *i, j, k, m*.

7.5.2 Formulation

The displacement at any point within the element in the global coordinate directions x, y, and z is written as:

$$\underset{\sim}{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

The displacement is defined as a linear variation in the coordinates by writing:

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z$$

This equation can be specialized to each of the four nodes of the tetrahedral element to obtain the four equations:

$$u_i = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 z_i$$
$$u_j = \alpha_1 + \alpha_2 x_j + \alpha_3 y_j + \alpha_4 z_j$$
$$u_k = \alpha_1 + \alpha_2 x_k + \alpha_3 y_k + \alpha_4 z_k$$
$$u_m = \alpha_1 + \alpha_2 x_m + \alpha_3 y_m + \alpha_4 z_m$$

which can be solved for the coefficients. It is possible to write this solution in a determinant form as:

$$u = \frac{1}{6V}[(a_i + b_i x + c_i y + d_i z)u_i + (a_j + b_j x + c_j y + d_j z)u_j + (a_k + b_k x + c_k y + d_k z)u_k + (a_m + b_m x + c_m y + d_m z)u_p]$$

with the volume of the tetrahedron defined by the determinant:

$$6V = \det \begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & z_m & y_m & x_m \end{vmatrix}$$

Expanding all the determinants into their cofactors gives:

$$a_i = \det \begin{vmatrix} x_j & y_j & z_j \\ x_k & y_k & z_k \\ x_m & y_m & z_m \end{vmatrix} \qquad b_i = -\det \begin{vmatrix} 1 & y_j & z_j \\ 1 & y_k & z_k \\ 1 & y_m & z_m \end{vmatrix}$$

$$c_i = -\det \begin{vmatrix} x_j & 1 & z_j \\ x_k & 1 & z_k \\ x_m & 1 & z_m \end{vmatrix} \qquad d_i = \det \begin{vmatrix} x_j & y_j & 1 \\ x_k & y_k & 1 \\ x_m & y_m & 1 \end{vmatrix}$$

All the other coefficients can be obtained via a cyclic permutation in the order *i, j, k, m.*

The gradient of the shape functions $\frac{\partial N_i}{\partial x}, \frac{\partial N_i}{\partial y}, \frac{\partial N_i}{\partial z}$ only depend upon the b_i, b_j, b_k, b_m, c_i, c_j, c_k, c_m, and d_i, d_j, d_k, d_m coefficients and they can be written directly in terms of the differences of the coordinates of the nodes. We denote the coordinate difference between two nodes with a double subscript as in the form $x_{ij} = x_i - x_j$, which denotes the difference in the x coordinate between nodes i and j of the tetrahedral element. Then:

$$\frac{\partial N_i}{\partial x} = \frac{1}{6V} b_i = \frac{1}{6V}\left(y_{jm} z_{jk} - y_{jk} z_{jm}\right)$$

$$\frac{\partial N_i}{\partial y} = \frac{1}{6V} c_i = \frac{1}{6V}\left(z_{jm} x_{jk} - z_{jk} x_{jm}\right)$$

$$\frac{\partial N_i}{\partial z} = \frac{1}{6V} d_i = \frac{1}{6V}\left(x_{jm} y_{jk} - x_{jk} y_{jm}\right)$$

$$\frac{\partial N_j}{\partial x} = \frac{1}{6V} b_j = \frac{1}{6V}\left(y_{ik} z_{im} - y_{im} z_{ik} \right)$$
$$\frac{\partial N_j}{\partial y} = \frac{1}{6V} c_j = \frac{1}{6V}\left(z_{ik} x_{im} - z_{im} x_{ik} \right)$$
$$\frac{\partial N_j}{\partial z} = \frac{1}{6V} d_j = \frac{1}{6V}\left(x_{ik} y_{im} - x_{im} y_{ik} \right)$$

$$\frac{\partial N_k}{\partial x} = \frac{1}{6V} b_k = \frac{1}{6V}\left(y_{im} z_{ij} - y_{ij} z_{im} \right)$$
$$\frac{\partial N_k}{\partial y} = \frac{1}{6V} c_k = \frac{1}{6V}\left(z_{im} x_{ij} - z_{ij} x_{im} \right)$$
$$\frac{\partial N_k}{\partial z} = \frac{1}{6V} d_k = \frac{1}{6V}\left(x_{im} y_{ij} - x_{ij} y_{im} \right)$$

$$\frac{\partial N_m}{\partial x} = \frac{1}{6V} b_m = \frac{1}{6V}\left(y_{ij} z_{ik} - y_{ik} z_{ij} \right)$$
$$\frac{\partial N_m}{\partial y} = \frac{1}{6V} c_m = \frac{1}{6V}\left(z_{ij} x_{ik} - z_{ik} x_{ij} \right)$$
$$\frac{\partial N_m}{\partial z} = \frac{1}{6V} d_m = \frac{1}{6V}\left(x_{ij} y_{ik} - x_{ik} y_{ij} \right)$$

where the volume is can be determined from:

$$6V = x_{im}\left(z_{ij} y_{ik} - y_{ij} z_{ik} \right) + y_{im}\left(x_{ij} z_{ik} - z_{ij} x_{ik} \right) + z_{im}\left(y_{ij} x_{ik} - x_{ij} y_{ik} \right)$$

8. MATERIALS

All property options (e.g. PSOLID, PSHELL, etc) must reference valid material IDs.

Every material model must have a density property included within the set of material properties that are assigned to the material model.

8.1 Linear Elasticity

The Linear Elastic Entry type is used to define an isotropic, linear elastic material and is defined using the MAT1 option where you define the elastic properties for the material. The stress increments $\Delta\sigma_{ij}$ are defined in terms of the strain increments $\Delta\varepsilon_{ij}$ by:

$$\Delta\sigma_{ij} = \lambda\Delta\varepsilon_{kk} + 2\mu\Delta\varepsilon_{ij}$$

where λ and 2μ are the usual Lamé material constants.

8.2 Metal Plasticity

This model requires the definition of the elastic properties of the material (Young's modulus and Poisson's ratio) using the MAT1 option as well as a specification of the yielding behavior of the material using the MATS1 option. The yield stress is defined as a function of plastic strain. If the stress strain curve from a uniaxial tension test is provided, the total strain values must be converted to plastic strain values. Figure 8-1 (a) shows the stress strain curve for a uniaxial test as a set of five data points defining a bilinear stress strain curve. In Figure 8-1 (b) the elastic strain ($\bar{\sigma}/E$) at each point is subtracted from the total strain for the point to obtain the plastic strain value.

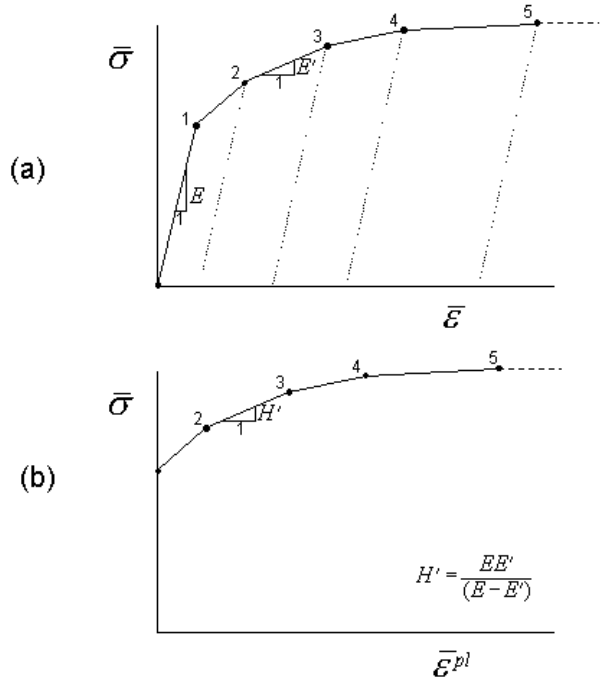


Figure 8-1. Conversion of Uniaxial Tension Test to Yield Stress vs Plastic Strain.

For the stress tensor, σ_{ij}, its stress deviator, S_{ij}, is defined as:

$$S_{ij} = \sigma_{ij} - p\,\delta_{ij}$$

where p is the mean pressure.

The first and second invariants of the stress tensor are:

$$p = \frac{1}{2}\sigma_{kk}$$

$$q = \sqrt{\frac{3}{2}S_{ij}S_{ij}}$$

The Von Mises yield surface is a circle in the deviatoric stress space and is defined as:

$$f(\sigma_{ij}) = \frac{1}{2}R^2 = \kappa^2$$

where R is the radius of the yield surface and is defined by:

$$R = \sqrt{S_{ij}S_{ij}}$$

The Von Mises effective stress is defined as:

$$\bar{\sigma} = \sqrt{\frac{3}{2}S_{ij}S_{ij}}$$

Since R is the magnitude of the deviatoric stress it follows that $R = \sqrt{\frac{3}{2}}\bar{\sigma}$

The normal to the yield surface is defined by:

$$Q_{ij} = \frac{S_{ij}}{R}$$

We assume the strain rate can be decomposed into elastic and plastic parts by an additive decomposition:

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}_{ij}^{el} + \dot{\varepsilon}_{ij}^{pl}$$

and assume that the plastic part of the strain rate is given by a normality condition:

$$\dot{\varepsilon}_{ij}^{pl} = \gamma Q_{ij}$$

with the scalar multiplier, γ, to be determined.

Combining the consistency condition (the state of stress must remain on the yield surface at all times), the additive strain rate decomposition, and the normality condition, and performing some tedious algebra allows us to solve the scalar multiplier, γ, as:

$$\gamma = \frac{1}{\left(1+\frac{H'}{3\mu}\right)} Q_{ij}\varepsilon_{ij}$$

Where H' is the slope of the effective stress versus equivalent plastic strain curve. Once the scalar multiplier, γ, is computed, these equations can be used to update the stress at the end of the increment from all known quantities. Note that Autodesk Explicit requires the input of stress versus strain (not equivalent plastic strain) and the code automatically make the conversion to yield stress versus plastic strain as shown in Figure 8-1.

8.3 Drucker-Prager Plasticity

The Drucker-Prager model plasticity model defines the yield stress as a function of the mean pressure (i.e. volumetric mean stress).

For the stress tensor, σ_{ij}, its stress deviator, S_{ij}, is defined as:

$$S_{ij} = \sigma_{ij} - p\,\delta_{ij}$$

where p is the mean pressure.

The first and second invariants of the stress tensor are:

$$p = \frac{1}{2}\sigma_{kk}$$

$$q = \sqrt{\frac{3}{2}S_{ij}S_{ij}}$$

Generally, this functional dependence is defined by the an angle, β, which defines the slope of the yield stress versus pressure line in the $p-q$ plane as shown in Figure 8-2. Note that pressure is positive in compression and that there is a maximum tensile pressure that can be achieved $p_t = d/\tan\beta$.

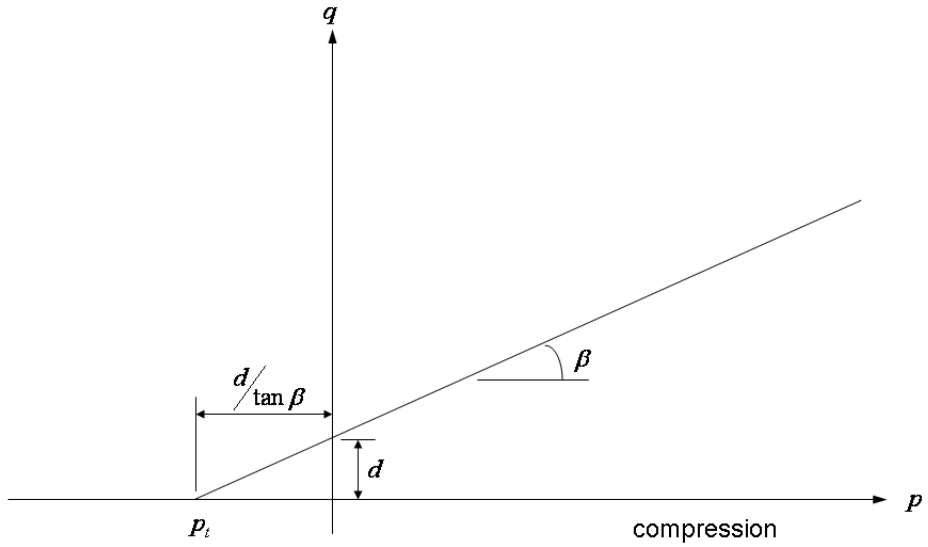


Figure 8-2. Yield Stress as a Function of Pressure.

The yield surface is defined by:

$$F = q - p \cdot \tan\beta - d = 0 \qquad (0.141)$$

where d represents the cohesion of the material and is defined as:

$d = (1 - \frac{1}{3}\tan\beta)\sigma_c^0$ if $\sigma(\bar{\varepsilon}^{pl})_c^0$ is the yield stress obtained from uniaxial compression

$d = (1 + \frac{1}{3}\tan\beta)\sigma_t^0$ if $\sigma(\bar{\varepsilon}^{pl})_t^0$ is the yield stress obtained from uniaxial tension

$d = \tau^0$ if $\tau(\bar{\varepsilon}^{pl})^0$ is the shear yield stress obtained from a shear test

Note that for the case where the yield stress is defined by uniaxial compression the value of $\beta \le \sqrt{3}$ which represents 71.5°.

In the deviatoric plane the yield surface is a circle as shown in Figure 8-3.

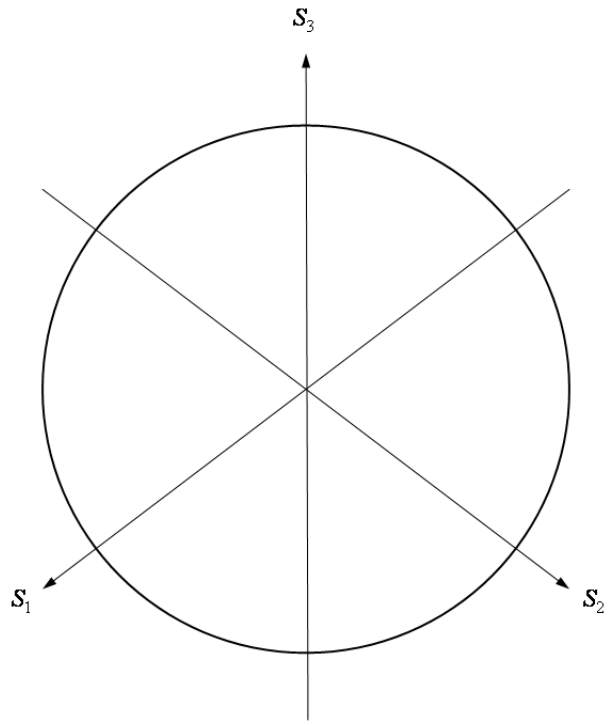


Figure 8-3. Drucker-Prager Yield Surface in the Deviatoric Plane.

We assume a potential flow rule as:

$$d\varepsilon_{ij}^{pl} = \frac{d\bar{\varepsilon}^{pl}}{a}\frac{\partial g}{\partial \sigma_{ij}}$$

The flow potential, g, is chosen as:

$$g = q - p \cdot \tan\psi$$

The angle, ψ, is referred to as the dilation angle for the material. The flow is associated in the deviatoric plane because the yield surface and flow potential both have the same functional dependence on q. However, the dilation angle, ψ, and friction angle, β, may be different in which case the model is not associated in the in the $p-q$ as shown in Figure 8-4. For $\psi = 0$ the flow is non-dilatational. For $\psi = \beta$ the model is fully associated.

The coefficient, a, in the potential flow rule is defined as:

$a = (1 - \frac{1}{3}\tan\psi)\sigma_c^0$ if $\sigma(\bar{\varepsilon}^{pl})_c^0$ is yield stress from uniaxial compression

$a = (1 + \frac{1}{3}\tan\psi)\sigma_t^0$ if $\sigma(\bar{\varepsilon}^{pl})_t^0$ is yield stress from uniaxial tension

$a = \tau^0$ if $\tau(\bar{\varepsilon}^{pl})^0$ is yield stress obtained from a shear test

and:

$d\bar{\varepsilon}^{pl} = \left\| d\bar{\varepsilon}_{11}^{pl} \right\|$ in the uniaxial compression case

$d\bar{\varepsilon}^{pl} = d\bar{\varepsilon}_{11}^{pl}$ in the uniaxial tension case

$d\bar{\varepsilon}^{pl} = d\gamma^{pl} / \sqrt{3}$ in the pure shear case, where γ^{pl} is the engineering shear plastic strain

Note that for the case where the yield stress is defined by uniaxial compression the value of $\psi \le \sqrt{3}$ which represents 71.5°.

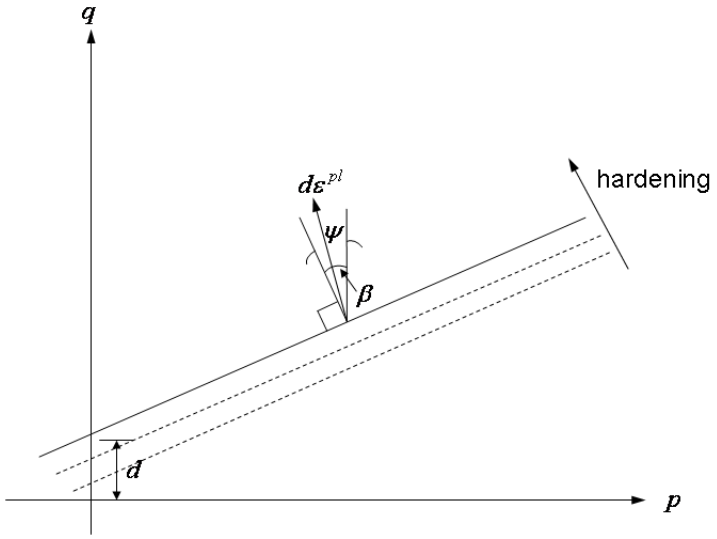


Figure 8-4. Hardening in the $p-q$ Plane.

The increment in plastic strain is then given by:

$$\Delta\varepsilon_{ij}^{pl} = \frac{\Delta\bar{\varepsilon}^{pl}}{a}\left[\frac{3}{2}\frac{S_{ij}}{q} + \frac{\tan\psi}{3}\delta_{ij}\right]$$

Assuming an additive strain rate decomposition and combining the elasticity with the flow rule results in:

$$S_{ij}^{n+1} = 2\mu\left(\Delta\hat{\varepsilon}_{ij} - \frac{\Delta\bar{\varepsilon}^{pl}}{a}\frac{3}{2}\frac{S_{ij}^{n+1}}{q}\right) + S_{ij}^{n}$$

where $\Delta\hat{\varepsilon}_{ij}$ are the deviatoric increment strains and the deviatoric trial stress is defined as:

$$S_{ij}^{trial} = S_{ij}^{n} + 2\mu\,\Delta\hat{\varepsilon}_{ij}$$

Solving for S_{ij}^{n+1}

$$S_{ij}^{n+1} = \frac{S_{ij}^{trial}}{C_S^{n+1}} \qquad (0.142)$$

where:

$$C_S^{n+1} = 1 + \frac{3}{2}\frac{2\mu\,\Delta\bar{\varepsilon}^{\,pl}}{a\,q^{n+1}} \qquad (0.143)$$

Making use of the Mises equivalent stress definition gives:

$$\left(C_S^{n+1} q^{n+1}\right)^2 = \frac{3}{2} S_{ij}^{Trial} S_{ij}^{Trial} = \left(q^{Trial}\right)^2 \qquad (0.144)$$

This may be simplified to:

$$q^{n+1} + 3\mu\frac{\Delta\bar{\varepsilon}^{\,pl}}{a} = q^{Trial} \qquad (0.145)$$

Combining the elastic and flow rule along the hydrostatic axis:

$$p^{n+1} = \Delta\bar{\varepsilon}^{\,pl} - K\left(\Delta\varepsilon_{vol} - \frac{\Delta\bar{\varepsilon}^{\,pl}}{a}\tan\psi\right) \qquad (0.146)$$

This may be simplified to:

$$p^{n+1} - K\frac{\Delta\bar{\varepsilon}^{\,pl}}{a}\tan\psi = p^{trial} \qquad (0.147)$$

Substituting Equations (0.145) and (0.146) in the yield Equation (0.141) and solving for the increment in equivalent plastic strain gives:

$$\Delta\bar{\varepsilon}^{\,pl} = \frac{q^{Trial} - \tan\beta\, p^{Trial} - d^{n}}{\dfrac{\partial d}{d\bar{\varepsilon}^{\,pl}} + \dfrac{3\mu}{a} + K\dfrac{\tan\beta\tan\psi}{a}} \qquad (0.148)$$

Under hydrostatic tension, we can reach the apex of the yield surface in the $p-q$ plane as shown in Figure 8-2. At this point $p = -d/\tan\beta$ and $q = 0$. Equation (0.142) gives:

$$\Delta\bar{\varepsilon}^{\,pl} = \frac{q^{Trial} a}{3\mu} \qquad (0.149)$$

Once we have determined the increment in equivalent plastic strain, $\Delta\bar{\varepsilon}^{pl}$, we can substitute this back into Equations (0.142) and (0.143) to recover the new deviatoric stress values and into Equation (0.146) to recover the new mean stress value.

Note that if $\beta = \psi = 0$ the Drucker-Prager model has no pressure dependence and the equations degenerate to the standard expressions for the conventional Mises plasticity model.

8.4 Crushable Foam

The crushable foam model in Autodesk Explicit specifies a material where the volumetric and deviatoric response can be specified with independent plasticity models. T his model may only be used with elements that use the PSOLID property type.

The crushable foam model is a phenomenological model designed to model crushable foams (e.g. Styrofoam). These types of foams tend to have very weak tensile strength. Furthermore, under compression they tend to "bulk up" as the voids in the foam are compressed and closed. The model is also suitable for soils which behave in a similar fashion. A typical volumetric strain versus pressure curve for such a foam is shown in Figure 8-5. For this model, you can specify the volumetric behavior in a piece-wise linear form as shown. The volumetric response can be purely elastic and this case is defined by simply giving two points on the curve (the zero pressure value is required). As shown in the figure, Autodesk Explicit will extrapolate the pressure in compression using the last value of the slope of the curve. As shown, the tensile response is elastic until the tensile fracture pressure is reached. The tensile fracture pressure is determined from the definition of deviatoric yield stress versus pressure described below.

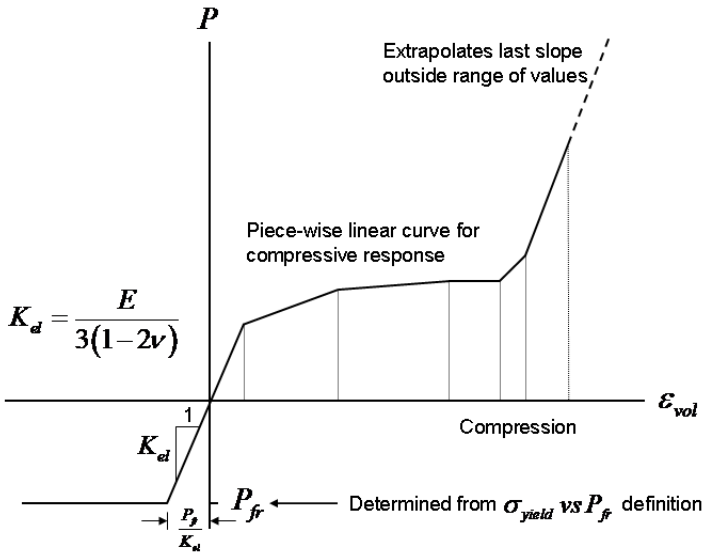


Figure 8-5. Pressure Versus Volumetric Response Curve.

The deviatoric plasticity is pressure dependent and may be defined with up to a quadratic dependence on pressure. Figure 8-6 shows the quadratic form of the dependence of yield stress on pressure. The input parameters are quite simple in that you need only provide the yield stress at zero pressure and the pressure value at which the dependence becomes linear with a constant slope. Autodesk Explicit automatically fits the quadratic curve to these parameters so that the curve passes through the zero pressure point specified and matches the slope at the point specified. Note that the tensile fracture pressure is automatically computed from this fit.

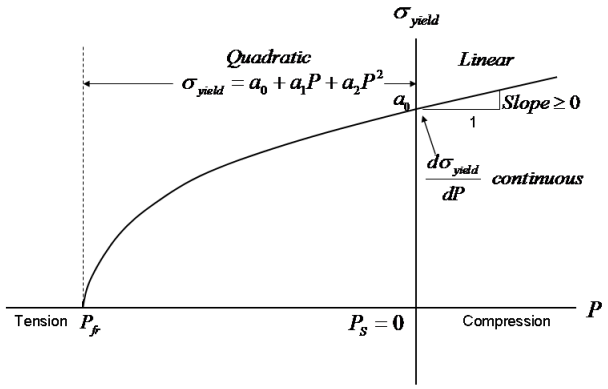


Figure 8-6. Quadratic Dependence of Yield Stress on Pressure.

When a linear dependence is used this model it is similar to a classical Drucker-Prager material model. This is shown in Figure 8-7. In this case you need only define the yield stress at zero pressure and the slope of the curve. Again, the tensile fracture pressure is automatically computed from this fit. If the volumetric response is purely elastic, there is very little difference between the classical Drucker-Prager model and the foam model.

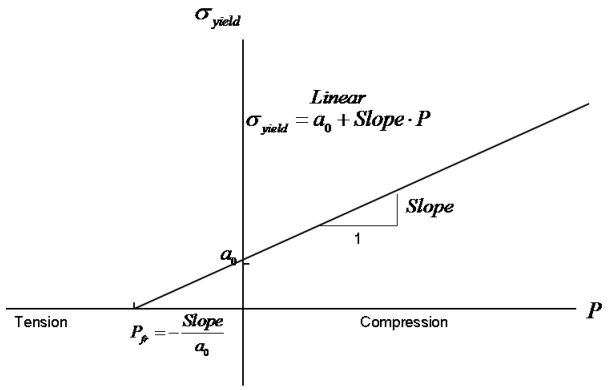


Figure 8-7. Linear Dependence of Yield Stress on Pressure.

A constant yield stress can be specified in this model as shown in Figure 8-8. In this case, if the pressure versus volumetric strain is purely linear (no piece-wise linear tabular dependence) the classic Mises plasticity model with no hardening is obtained. Note that for this case the tensile fracture pressure is set to an infinite value and there is no tensile failure.

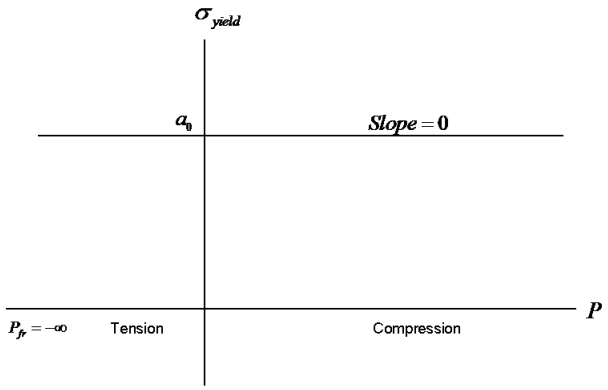


Figure 8-8. No Pressure Dependence of Yield Stress.

8.5 Brittle Failure

The brittle failure model in Autodesk Explicit is a tensile cracking model which allows cracks to form in the principal strain directions of the material. The cracks are always orthogonal to one another. Cracks only form in tension and once they are formed they do not heal. That is, the material remembers that it has cracked in a particular direction and will never carry tensile stress in that direction again. Furthermore, subsequent cracks that form will be orthogonal to that direction. Crack can close and carry compressive stress. The tensile behavior of the material is shown in Figure 8-9. The tensile strength, TS, is specified by the user. The strain softening behavior shown in the figure is used to ramp the stresses down to zero over a handful of time steps rather than simple set the stress immediately to zero. This is required to gracefully dissipate the violent energy that is release when a crack forms. The brittle material is implemented so that you may use the element deletion option (DELETION) to delete elements from the mesh after they have cracked. You may specify the multiple of the tensile strain at fracture that want to use for deleting the element. In the case shown in Figure 8-9 a tensile failure multiplier of 4 would delete the element when the stresses reach the value shown on the curve.

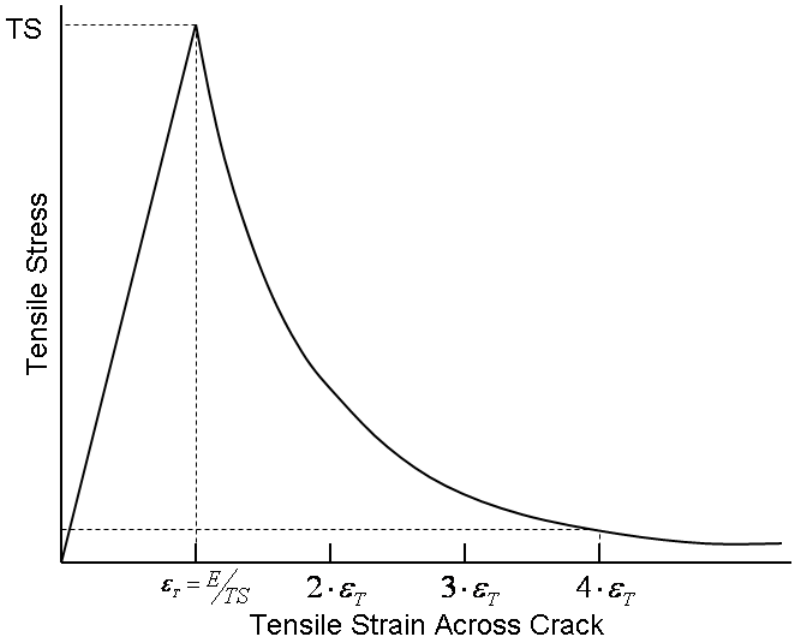


Figure 8-9. Tensile Stress/Strain Curve.

This model can be used with solid (PSOLID) or shell (PSHELL) elements. For the plane stress case only two cracks can form (since the stress is zero in the out-of-plane direction. For the 3D strain case of solids, up to 3 orthogonal cracks can form.

The tensile behavior is purely elastic up to the point of fracture in one of the principal directions. The compressive behavior the material can be specified to be purely elastic, elastic/perfectly plastic, or elastic/plastic strain softening. The elastic/perfectly plastic behavior is shown in Figure 8-10. The elastic/plastic strain softening case is shown in Figure 8-11. As with the tensile behavior, the compressive response is integrated with the deletion options (DELETION) so that you can have the element delete when the compressive strain reaches some multiple of the initial yield strain value.

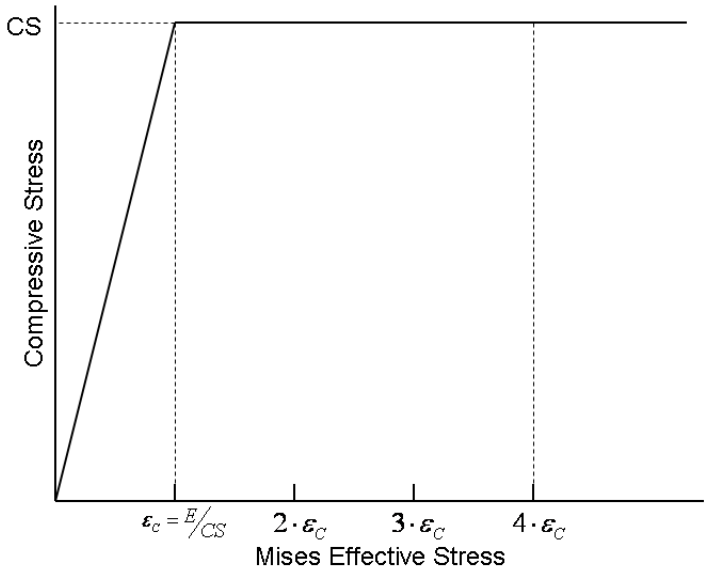


Figure 8-10. Elastic/Perfectly Plastic Compressive Model.

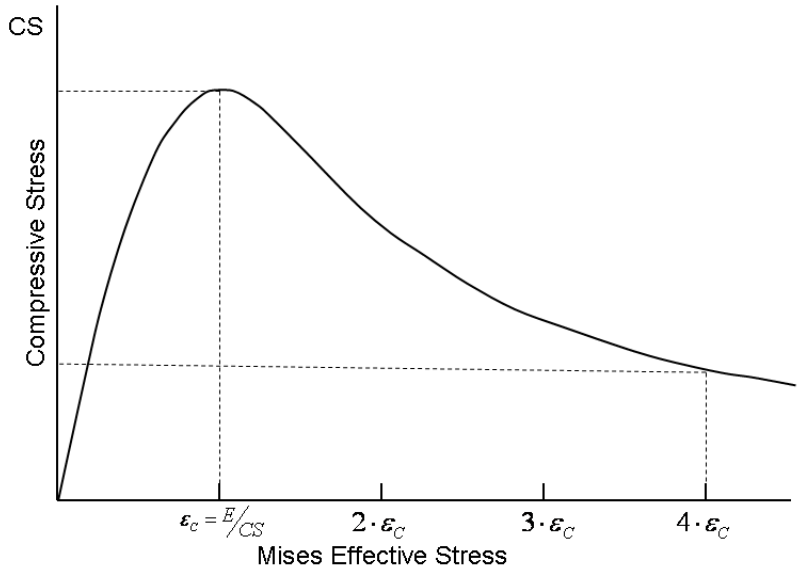


Figure 8-11. Elastic/Perfectly Plastic Compressive Model – Strain Softening.

8.6 Hyperelastic Polynomial Material (MATHP)

Most elastomers (solid, rubber-like materials) behave in a nearly incompressible manner. The hyperelastic polynomial formulation offered in Autodesk Explicit provides an isotropic, large-deformation formulation for elastomeric materials where the stresses are determined in terms of a "strain energy potential for the material, $U(\varepsilon)$, written in terms of the strain invariants.

The deformation gradient at a point in space is defined as

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

Where x_i are the current coordinates at the point and X_j are the original coordinates of the point. Then the total volume change at the point is given by the determinant of the deformation gradient.

$$J = \det(F)$$

It is convenient to define

$$\bar{F}_{ij} = \frac{1}{J^{1/3}} F_{ij}$$

The left Cauchy-Green strain tensor is defined as

$$\bar{B}_{ij} = \bar{F}_{ik}\bar{F}_{kj}$$

which is a symmetric tensor.

The first invariant of $\bar{B}$ is simply the trace of the tensor

$$I_1 = \bar{B}_{kk}$$

And the second invariant of $\bar{B}$ is defined as

$$I_2 = \bar{B}_{ij}\bar{B}_{ij} = \frac{1}{2}\left(I_1^2 - trace(\bar{B}_{ik}\bar{B}_{kj})\right)$$

The strain energy, $U(\varepsilon)$, at a point is written as a function of I_1, I_2, and J

$$U = \sum_{i+j=1}^{N} C_{ij}\left(I_1 - 3\right)^i \left(I_2 - 3\right)^j + \frac{K}{2}\left(J - 1\right)^2$$

The stresses are determined by taking the partial derivatives of the strain energy potential with respect to I_1, I_2, and J.

The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2(C_{10} + C_{01}) \qquad K_0 = 2D_1$$

If only a single term is used in the polynomial ($N = 1$), the Mooney-Rivlin law is obtained:

$$U = C_{10}\left(I_1 - 3\right) + C_{01}\left(I_2 - 2\right) + D_1\left(J - 1\right)^2$$

When $C_{01} = 0$ in the Mooney-Rivlin law, the neo-Hookean law is obtained.

The hyperelastic material with polynomial form (MATHP) provides a test data input option where Autodesk Explicit will fit the coefficients for the strain energy potential form user-supplied test data. The various deformation modes allowed are shown in Figure 8-12.

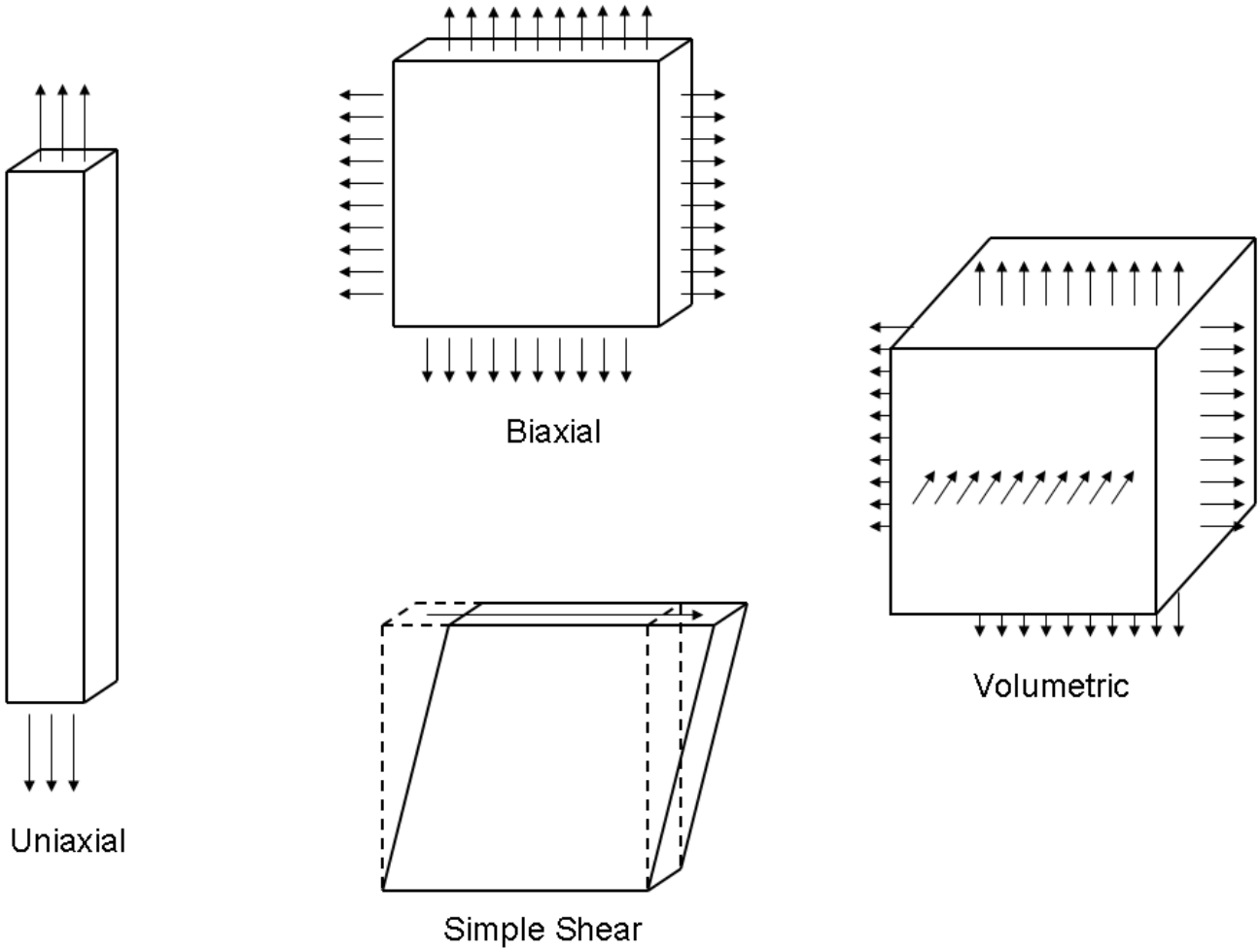


Figure 8-12. Deformations modes for test data input for hyperelastic and hyperfoam materials.

8.7 Hyperelastic Ogden Material (MATHP1)

Most elastomers (solid, rubber-like materials) behave in a nearly incompressible manner. The hyperelastic Ogden formulation offered in Autodesk Explicit provides an isotropic, large-deformation formulation for elastomeric materials where the stresses are determined in terms of a "strain energy potential for the material, $U(\varepsilon)$, written in terms of the principal strain values.

The deformation gradient at a point in space is defined as

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

Where x_i are the current coordinates at the point and X_j are the original coordinates of the point. Then the total volume change at the point is given by the determinant of the deformation gradient.

$$J = \det(F)$$

It is convenient to define

$$\bar{F}_{ij} = \frac{1}{J^{1/3}} F_{ij}$$

The left Cauchy-Green strain tensor is defined as

$$\bar{B}_{ij} = \bar{F}_{ik} \bar{F}_{kj}$$

which is a symmetric tensor.

The strain energy, $U(\varepsilon)$, at a point is written as a function of the principal stretches (λ_i) and J as

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3 \right) + \sum_{i=1}^{N} D_i \left(J - 1 \right)^2$$

$$\bar{\lambda}_i = \frac{1}{J^{1/2}} \lambda_i \rightarrow \bar{\lambda}_1 \bar{\lambda}_2 \bar{\lambda}_3 = 1$$

The stresses are determined by taking the partial derivatives of the strain energy potential with respect to the λ_i and J.

The initial shear modulus and bulk modulus are given by

$$\mu_0 = \sum_{i=1}^{N} \mu_i \qquad K_0 = \sum_{i=1}^{N} 2D_i$$

The Mooney-Rivilin law is obtained when $N = 1$, $\alpha_1 = 2$, $\alpha_2 = -2$, $\mu_1 = 2C_{10}$ and $\mu_2 = 2C_{01}$.

The neo-Hookean form is obtained when $N = 1$, $\alpha_1 = 2$, and $\mu_1 = 2C_{10}$.

8.8 Hyperfoam Materials

Hyperfoam materials represent elastometric foams that undergo large volumetric deformations as well as large shears and recover their shape upon unloading. Typical applications are seat cushions and packaging materials. The stresses are determined in terms of a "strain energy potential" for the material, $U(\varepsilon)$, written in terms of strain invariants or principal strain values.

The deformation gradient at a point in space is defined as

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

Where x_i are the current coordinates at the point and X_j are the original coordinates of the point. Then the total volume change at the point is given by the determinant of the deformation gradient.

$$J = \det(F)$$

The left Cauchy-Green strain tensor is defined as

$$B_{ij} = F_{ik} F_{kj}$$

which is a symmetric tensor.

We define λ_i as the "principal stretches" which are the three eigenvalues of the left Cauchy-Green strain tensor, B_{ij}.

The strain energy potential is written as

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}\left[\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3 + \frac{1}{\beta_i}\left(J^{-\alpha_i/\beta_i} - 1\right)\right]$$

The initial shear modulus of the material is given by

$$\mu_0 = \sum_{i=1}^{N} \mu_i$$

And the initial bulk modulus is given by

$$K_0 = \sum_{i=1}^{N} 2\mu_i\left(\beta_i + \frac{1}{3}\right)$$

The degree of compressibility in each term of the series is defined by the β_i coefficients and these are related to the effective Poisson's ratio in each term of the series by

$$\beta_i = \frac{\nu_i}{1-2\nu_i} \qquad \nu_i = \frac{\beta_i}{1+2\beta_i}$$

The material input for the hyperfoam material is defined in terms of the α_i, β_i and ν_i coefficients. The equations above show impose numerical restrictions that $-1 < \nu_i < .5$. For "real-life" materials negative values of Poisson's ratio are not reasonable but are allowed and may occur for specific fits of the coefficients to experimental data.

The hyperfoam material (MATHPF) provides a test data input option where Autodesk Explicit will fit the coefficients for the strain energy potential form user-supplied test data. The various deformation modes allowed are shown in Figure 8-12.

8.9 Viscoelasticity

8.9.1 Isotropic Linear Elastic

The elasticity relation for small strain isotropic, linear elastic materials is written as

$$\sigma_{ij} = \frac{E}{(1+\nu)}\varepsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)}\varepsilon_{kk}\delta_{ij} \qquad (1)$$

If we define the deviatoric portion of the strains, ε'_{ij}, as

$$\varepsilon'_{ij} = \varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}\delta_{ij}$$

and note that $G = \frac{E}{2(1+\nu)}$ and $K = \frac{E}{1(1-2\nu)}$, then the stress/strain law can be re-written in terms of the deviatoric strains as

$$\sigma_{ij} = 2G\varepsilon'_{ij} + K\varepsilon_{ll}\delta_{ij} \qquad (2)$$

There are two convenient ways that the response of viscoelastic materials can be modeled: (1) with relaxation modulus $E(t)$ and a constant Poisson's ratio ν; or (2) with a relaxation shear modulus $G(t)$ and relaxation bulk modulus $K(t)$.

Thus

$$\sigma_{ij}(t) = \int_0^t \left[\frac{E(t-\tau)}{(1+\nu)} \frac{\partial \varepsilon_{ij}}{\partial \tau} + \frac{\nu E(t-\tau)}{(1+\nu)(1-2\nu)} \frac{\partial \varepsilon_{kk}}{\partial \tau} \delta_{ij} \right] d\tau \tag{3}$$

or

$$\sigma_{ij}(t) = \int_0^t \left[2G(t-\tau) \frac{\partial \varepsilon'_{ij}}{\partial \tau} + 2K(t-\tau) \frac{\partial \varepsilon_{kk}}{\partial \tau} \delta_{ij} \right] d\tau \tag{4}$$

If (3) is used, then the mean stress/volumetric strain will relax or creep, just line the uniaxial stress relaxation or creep behavior. If (4) is used, the mean stress/volumetric behavior follows a separate creep/relaxation behavior.

VE behavior is also associated with a time-temperature shift function. This behavior is referred to as TRS or Theromorheologically Simple behavior (obviously, an oxymoron), where

$$\xi(t) = \int_0^t \varphi(T(\tau)) d\tau \tag{5}$$

where $\varphi(T)$ is the time-temperature shift function.

In order to simplify the formulation of the viscoelastic materials, we limit modeling of TRS materials with either a ***constant Poisson's ratio*** or a ***constant bulk modulus***. Thus,

$$\sigma_{ij}(t) = \int_0^t \left[\frac{E(\xi(t)-\xi(\tau))}{(1+\nu)} \frac{\partial \varepsilon_{ij}}{\partial \tau} + \frac{\nu E(\xi(t)-\xi(\tau))}{(1+\nu)(1-2\nu)} \frac{\partial \varepsilon_{kk}}{\partial \tau} \delta_{ij} \right] d\tau \tag{6}$$

or

$$\sigma_{ij}(t) = \int_0^t \left[2G(\xi(t)-\xi(\tau)) \frac{\partial \varepsilon'_{ij}}{\partial \tau} + K \frac{\partial \varepsilon_{kk}}{\partial \tau} \delta_{ij} \right] d\tau \tag{7}$$

The only practical way to numerically implement either viscoelastic formulation is using Prony series where

$$E(t) = \sum_{k=1}^{K} E_K e^{-t/\tau} + E_\infty \tag{8}$$

or

$$G(t) = \sum_{k=1}^{K} G_K e^{-t/\tau} + G_\infty \tag{9}$$

E_∞ and G_∞ are the long-term (static) moduli which we will not retain in the following because $E_\infty \to E_K$ as $\tau_K \to \infty$. Note that generally $\tau_1 < \tau_2 \cdots < \tau_K$.

Prony series are used to model viscoelastic relaxation because they provide the only method of updating the integral of the stress rate/stress history. Note that the strain rate $\dot{\varepsilon}_{ij}$ is constant over each time increment, thus, a typical Prony series term integrates as

$$\sigma_k\left(t_2\right) = \int_0^{t_1} E_k e^{-\frac{t_2-\tau}{\tau_1}} \dot{\varepsilon} d\tau = \int_0^{t_1} E_k e^{-\frac{t_2-\tau}{\tau_1}} \dot{\varepsilon} d\tau + \int_{t_1}^{t_2} E_k e^{-\frac{t_2-\tau}{\tau_1}} \dot{\varepsilon} d\tau \tag{10a}$$

and

$$\sigma_k\left(t_2\right) = \int_0^{t_1} E_k e^{-\frac{t_2-t}{\tau_1}} e^{-\frac{t_1-\tau}{\tau_1}} \varepsilon d\tau + \tau_1 E_k \frac{\Delta\varepsilon}{\Delta t}\left(1-e^{-\frac{\Delta t}{\tau_1}}\right) \tag{10b}$$

Where

$$\Delta t = t_2 - t_1 \tag{10c}$$

Thus

$$\sigma_k\left(t_2\right) = e^{-\frac{\Delta t}{\tau_1}} \int_0^{t_1} E_k e^{-\frac{t_1-\tau}{\tau_1}} \varepsilon d\tau + E_k \Delta\varepsilon\left(1-e^{-\frac{\Delta t}{\tau_1}}\right) \tag{10d}$$

or

$$\sigma_k\left(t_2\right) = e^{-\frac{\Delta t}{\tau_1}} \sigma\left(t_1\right) + \Delta\sigma_k \tag{10e}$$

where

$$\Delta\sigma_k = E_k \Delta\varepsilon\left(\frac{\tau_1}{\Delta t}\right)\left(1-e^{-\frac{\Delta t}{t_1}}\right) \tag{10f}$$

Introducing the TRS shift function, define

$$\Delta\xi = \xi\left(t_2\right) - \xi\left(t_1\right) \tag{11a}$$

$$\xi\left(t_2\right) = \int_0^{t_2} \varphi\left(T\left(\tau\right)\right) d\tau = \xi\left(t_1\right) + \int_{t_1}^{t_2} \varphi\left(T\left(\tau\right)\right) d\tau \tag{11b}$$

And let

$$\bar{\varphi} = \varphi\left(\frac{T\left(t_1\right) + T\left(t_2\right)}{2}\right) \tag{11c}$$

Then

$$\xi(t_2) = \xi(t_1) + \bar{\varphi}\Delta t = \xi(t_1) + \Delta\xi \tag{11d}$$

Define

$$\lambda_1 = \frac{\bar{\varphi}\Delta t}{\tau_1} \tag{11e}$$

Then

$$\sigma_k(t_2) = e^{-\frac{\Delta\xi}{\tau_1}}\sigma(t_1) + \Delta\sigma_k = e^{-\lambda_1}\sigma(t_1) + \Delta\sigma_k \tag{12a}$$

where

$$\Delta\sigma_k = E_k\Delta\varepsilon\frac{(1-e^{-\lambda_1})}{\lambda_1} \tag{12b}$$

Viscoelasticity with Prony-Series Relaxation Modulus $E(t)$

From Equation (6),

$$\sigma_{ij}(t) = \int_0^t \left[\frac{E\left(\xi(t)-\xi(\tau)\right)}{(1+\nu)}\frac{\partial\varepsilon_{ij}}{\partial\tau} + \frac{\nu E\left(\xi(t)-\xi(\tau)\right)}{(1+\nu)(1-2\nu)}\frac{\partial\varepsilon_{kk}}{\partial\tau}\delta_{ij} \right] d\tau \tag{13a}$$

where

$$E(t) = \sum_{k=1}^{K} E_K e^{-t/\tau} + E_\infty \tag{13b}$$

Using equations (12a) and (12b)

$$\sigma_{ij}(t_{n+1}) = \sum_{k=1}^{K}\left[e^{-\lambda_k}\ \sigma_{ij}(t_n) + \Delta\sigma_{ij}^k \right] \tag{13c}$$

and

$$\Delta\sigma_{ij}^k = \frac{E_k}{(1+\nu)}\Delta\varepsilon_{ij} + \frac{\nu E_k}{(1+\nu)(1-2\nu)}\Delta\varepsilon_{ll}\delta_{ij} \tag{13d}$$

Thus at every time step we must update the kth stress state-variable as

$$\sigma_{ij}(t_{n+1}) = e^{-\lambda}\ \sigma_{ij}(t_n) + \frac{E_k}{(1+\nu)}\left[\Delta\varepsilon_{ij} + \frac{\nu}{(1-2\nu)}\Delta\varepsilon_{ll}\delta_{ij} \right]\frac{\left(1-e^{-\lambda_k}\right)}{\lambda_k} \tag{13e}$$

Viscoelasticity with Prony Series Relaxation Modulus $G(t)$

From Equation (7),

$$\sigma_{ij}(t) = \int_0^t \left[2G\left(\xi(t) - \xi(\tau)\right) \frac{\partial \varepsilon'_{ij}}{\partial \tau} + K \frac{\partial \varepsilon_{kk}}{\partial \tau} \delta_{ij} \right] d\tau \qquad (14a)$$

Where

$$G(t) = \sum_{k=1}^{K} G_K e^{-t/\tau} + G_\infty \qquad (14b)$$

Using Equations (12a) and (12b)

$$\sigma_{ij}(t_{n+1}) = \sum_{k=1}^{K} [\ ]\ \sigma_{ij}(t_{n+1}) = \sum_{k=1}^{K} \left[e^{\frac{\Delta\xi}{\tau_k}}\ \sigma_{ij}^k(t_n) + \Delta\sigma_{ij}^k \right] + K\varepsilon_{ll}\delta_{ij} \qquad (14b)$$

where

$$\Delta\sigma_{ij}^k = 2G_k \Delta\varepsilon_{ij} \qquad (14c)$$

8.610 Rigid Materials

Autodesk Explicit contains a very powerful technique for creating rigid bodies in the model. You simply add a MATR1 option to the Nastran model that references the property ID of a part that you wish to be rigid. All elements in the element block that was created from that property ID will be bound together as a single rigid body for the analysis. All material properties for the elements will simply be ignored except for the density value. You can just leave them in the model.

When an element block is made rigid, it no longer performs any element calculations (e.g. no material stresses or strains are computed). Moreover, the elements of the rigid body do not contribute to the Courant stability limit considerations of the analysis. This can vastly increase the performance of some modes and reduce the total computer time for the analysis. Consider the case where you want to crush or form some relatively soft material between two steel dies. The shape of the dies can be create using CQUAD4 elements to form a shell or membrane element block. There could be a large number of elements required to capture complex geometric details. Furthermore, the wave speed in the steel may be many orders of magnitude higher than that of the soft material you are crushing. If the dies are not rigid they could very well be the critical elements that control the stability limit of the model. By making them rigid, you only perform calculations in the crushed material and you obtain your stability limit from the softer material as well.

Autodesk Explicit will compute the center of mass and the mass properties of the rigid body automatically for you and compute the rigid body motions of the element block due to any forces acting upon it. You can still assign point loads and pressures to the body. Furthermore, the element block can participate in contact just as if it was deformable.

There are two special boundary conditions, SPCR and SPCRD, that have been added to Autodesk Explicit so that you can apply boundary conditions to the center of mass of the body. Autodesk Explicit creates an internal node for the center of mass of the rigid body with 6 degrees of freedom – 3 displacement degrees of freedom and 3 rotational degrees of freedom. The SPCR and SPCRD options behave exactly like the SPC and SPCD option for regular grid points in the model. The only difference is that instead of referencing a grid ID as on the SPC and SPCD options, you reference a property ID for the rigid body on the SPCR and SPCRD options.

Autodesk Explicit is not a "rigid body dynamics application" and you should not try to make every part in your model rigid.

9. LOADS, BOUNDARY CONDITIONS AND INITIAL CONDITIONS

9.1 Boundary Conditions

Autodesk Explicit contains options for defining boundary conditions on nodes in the mesh. These boundary conditions are applied using the SPC, SPC1, SPCD and SPCR options.

The SPC and SPC1 options allow you to prescribe zero value (fixed) boundary conditions that do not vary with time.

The SPCD and SPCR options allow you to prescribe boundary conditions that vary with time using the TABLED1 and TABLED2 options.

If the grid point specified in one of the boundary condition options was defined in a local coordinate system, the SPC option is applied in that coordinate system as well.

9.2 Loads

Autodesk Explicit contains options for defining mechanical loads on the mesh. These are the FORCE, FORCE1, GRAV, MOMENT, MOMENT1, PLOAD, PLOAD2, and PLOAD4 options in Nastran.

If the grid point specified in one of the FORCE, FORCE1, MOMENT, or MOMENT1 options was defined in a local coordinate system, the SPC option is applied in that coordinate system as well.

9.3 Initial Conditions

Autodesk Explicit defines initial conditions using the conventional Nastran TIC option.

In addition, the INITDIS, INITVEL, and ROTVEL options have been introduced so that you can define initial conditions on a node set or on a part of the model. The INITVEL option is useful to move a part up to another part before beginning an impact analysis so as to avoid performing senseless explicit dynamics integration of rigid body motion to close the gap between the two bodies. The INITVEL and ROTVEL options allow you to specify initial velocities of a part in a single concise option as opposed to the TIC option which requires a command for every grid point with the initial condition. For models with hundreds of thousands of grid points this can significantly reduce the input file size

10. CONTACT

10.1 Mechanical Contact

Autodesk Explicit provides fully automatic capabilities for defining mechanical contact conditions between all the external surfaces of the model. This includes the ability for surfaces to fold upon themselves (self-contact).

We provide options to set parameters for the contact specification (e.g. specify friction coefficients). Surfaces can be specified as "tied" together. Autodesk Explicit will glue the surfaces together if the surfaces are "close enough" to be considered to be touching. The concept of close enough can be controlled through the MAXAD parameter on the BSCONP option.

Autodesk Explicit provides both a pure primary/secondary contact algorithm and a symmetric balanced primary/secondary contact. Parameters on the BSCONP option control these relationships. As a general rule, you should make the contact pair relationship a pure primary/secondary in the following situations:

1. One surface has a much finer mesh than the other. In this case, make the fine mesh the secondary.
2. One surface belongs to an element block that has a very stiff behavior compared to the other. ,For example:
    - A solid body in contact with a shell body where both have approximately the same material behavior. Make the shell the secondary.
    - A shell body in contact with a soft solid body (e.g. steel shell vs. foam solid). Make the solid the secondary.
    - A solid body that is very stiff in contact with a soft solid body (e.g. steel solid in contact with foam solid). Make the foam the secondary.

Autodesk Explicit uses a naming convention for the surfaces so that it is apparent how the surfaces were created. When Autodesk Explicit prints out summary output about the contact conditions, these naming conventions are apparent. All surfaces that are defined using the automatic global surface creation option will represent the skin of the different element blocks. Those surfaces will have a name that has the prefix "eb:" plus the element block name. For example, for the element block in the mesh created from part ID 4 named "EB4", the surface created from the skin of the element block will have the name "eb:EB4". Surfaces that are created from BSSEG definitions that contain only the raw nodal connectivity in Nastran will have a prefix of "f:" plus the BSSEG property ID. For example, if a surface is created from the BSSEG with property ID 88, the surface name will be "fs:BSSEG-88".

10.2 Automatic Surface Creation

Autodesk Explicit contains options for creating surfaces on the parts of the mesh defined by property IDs using the CONTACTGENERATE or BSCREATE options. These options allow you to specify that you want to automatically create surfaces by skinning the free surface of the parts of the model. The CONTACTGENERATE option will skin all the parts of the model and create contact pairs between all possible surfaces (including surface in contact with themselves). If no BSCONP option is used, friction is assumed to be zero between all parts and appropriate default balance factors are chosen based on the element types. All contact is assumed to be general (not welded) contact.

There is a field on the CONTACTGENERATE command to specify that self-contact is not required and therefore not to be generated.

The BSDELETE option can be used to remove any surface pairs that were created by the CONTACTGENERATE option but are not necessary to the analysis. So, if a model contains two parts with PIDs of 88 and 99 and you know they will never contact one another but they will be created by the CONTACTGENERATE option, you can simply use the command:

BSDELETE,88,99

to remove that surface pair.

The BSCREATE option can be used to skin individual parts of the model. Generally this is used when the CONTACTGENERATE option would create more contact surface pairs than necessary. For example:

BSCREATE,99

will find all the elements in the mesh that are assigned property ID 99 and generate a surface with the ID 99 from all the free faces of that part. Care should be taken if you mix CONTACTGENERATE and BSCREATE because this can in fact generate multiple surfaces on the same part.

The surface IDs created using the CONTACTGENERATE or BSCREATE options will be the same as the part ID that was skinned.

The BSCONP option can be used to change friction parameters, balance parameters and the contact model (general surface-to-surface no tension contact or welded contact) for any surface pairs. The surface IDs are all defined in a single ID space. This means that whether you create surfaces using the CONTACTGENERATE, BSCREATE, or BSSEG options care should be taken to avoid conflicts in the surface IDs.

10.3 Two-Sided Surfaces and Surface Orientation

Surfaces that are created on elements that have two sides (shell and membrane elements) will automatically behave as two-sided surfaces in Autodesk Explicit. This means they will inherit the thickness of the parent element and other surfaces can contact either of the two sides of the surface. These surfaces can also fold onto themselves in self contact on both sides of the surface. These surfaces are "oriented" in that the 1st side of the surface all has a positive normal that corresponds to the positive element outward normal defined by the usual clock-wise node numbering on the parent element. The 2nd side of the surface is in the negative normal direction.

Significant computational savings can be achieved if you know that a surface does not have to be two-sided. That is, you know that it can only be contacted from one side. The BSORIENT option allows you to orient a surface to use only one side of the surface. The option allows you to specify whether you want the positive or negative side of the surface to be retained. It also allows you to specify the coordinates of a point in space to orient the surface. This is particularly useful if you have a cylindrical or spherical surface and you would like the surface to be oriented to the inside. In this case, you simply give the coordinates of the center of the cylinder or sphere. If you want the surface to be the outside of the cylinder or sphere then give the coordinates of the center and also specify the "reverse" option.

You can specify the thickness of the surface independent of the parent element thickness using the BSTHICK option. Therefore, if you wan the surface to be off-set from the element definition by more than the shell element thickness, you can increase (or decrease its thickness). Note that the offset is half the thickness specified for shell elements. You can specify a thickness of zero. However, a zero thickness means the surface is no longer two-sided and only the positive side will be used. If you want to use zero thickness with the negative side, you can combine the BSTHICK an BSORIENT options.

10.4 Element Deletion and Contact

Autodesk Explicit contains options for deleting elements from the mesh using deletion criteria defined with the DELETION parameter. The contact logic in Autodesk Explicit has the ability to automatically rebuild the surfaces of the model as elements are delete/eroded from the contact surface. This allows you to model the penetration of one part into another.

This option only applies if the contact surfaces were built using the BSSEG parameter described above. These surfaces "know" which elements they came from and are able to automatically rebuild themselves when elements are deleted. If the surface is defined using a BSSEG option in the conventional Nastran fashion, this option will not be available for that surface.

11. COORDINATE SYSTEMS

Autodesk Explicit provides options for defining transformed coordinate systems. These local systems are particularly useful for defining boundary conditions and loads. Each type of coordinate system available in Autodesk Explicit provides the transformation matrix R_{ij} that transforms a global vector into the local system according to:

$$\hat{x}_i = R_{ij} x_j$$

11.1 Cartesian

A Cartesian local coordinate system defines a transformation that is constant (i.e. the transformation matrix R is identical for all points in space. To define a local Cartesian system, it is necessary to define two points in space as shown in Figure 11-1. The first point, A, defines a point on the local x-axis, where the local x-axis points from the origin of the global system (0,0,0) to point A. The second point B, defines a point on the local y-axis, where the local y-axis points from the origin of the global system to point B. Neither point A nor point B may coincide with the global origin. Also, point A and point B must not be coincident. It is not necessary for the vectors defined by points A and B to be unit vectors. The Cartesian coordinate system is defined using the COORD1R or COORD2R options.

Autodesk Explicit provides fully automatic capabilities for defining mechanical contact conditions between all the external surfaces of the model. This includes the ability for surfaces to fold upon themselves (self-contact).

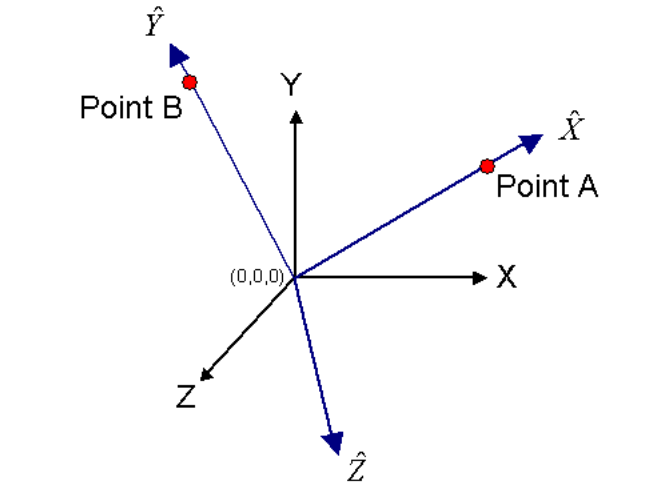


Figure 11-1. Construction of a Local Cartesian Coordinate System.

11.2 Cylindrical

A Cylindrical local coordinate system defines a local system that varies at every point in space. To define a local cylindrical system, it is necessary to define two points in space as shown in Figure 11-2. The first point A defines a point on the axis of the cylindrical system. The second point B defines another point on the axis of the local cylindrical system. The vector that passes from point A through point B defines the axis of the cylinder and therefore the orientation of the local $\hat{Z}$ axis. Point A and point B must not be coincident. It is not necessary for the vector defined by points A and B to be a unit vector. Autodesk Explicit will normalize them internally. The Cylindrical coordinate system is defined using the COORD1C or COORD2C options.

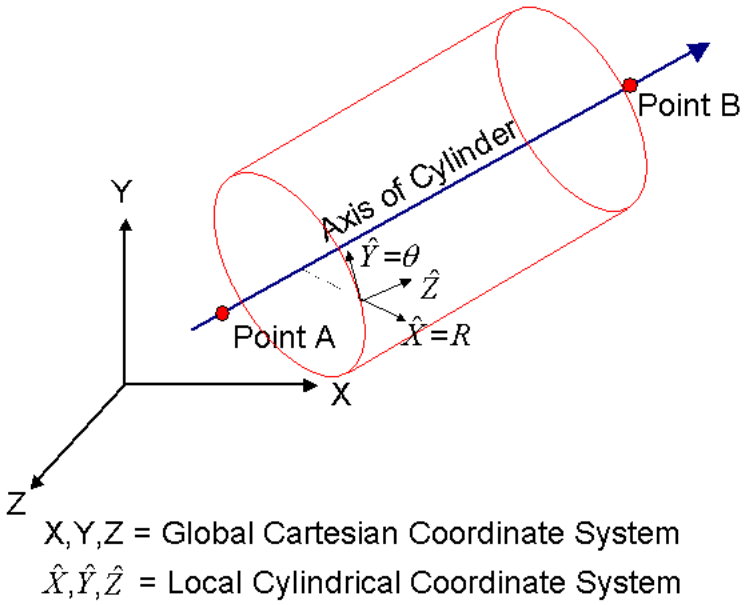
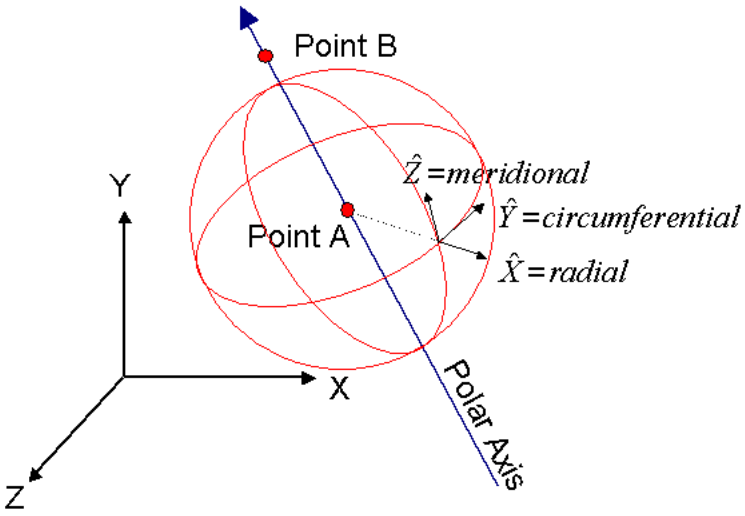


Figure 11-2. Construction of a Local Cylindrical Coordinate System.

11.3 Spherical

A Spherical local coordinate system defines a local system that varies at every point in space. To define a local spherical system, it is necessary to define two points in space as shown in **Error! Reference source not found.**. The first point A defines the center of the spherical system. The second point B defines a point on the polar axis of the local spherical system. The vector that passes from point A through point B defines the polar axis of the sphere. Point A and point B must not be coincident. It is not necessary for the vector defined by points A and B to be a unit vector. Autodesk Explicit will normalize them internally. The Cylindrical coordinate system is defined using the COORD1S or COORD2S options.



X,Y,Z = Global Cartesian Coordinate System

$\hat{X},\hat{Y},\hat{Z}$ = Local Spherical Coordinate System

Figure 11-3. Construction of a Local Spherical Coordinate System.

12. CREATION AND DELETION

Autodesk Explicit provides options for deleting elements from the mesh based upon criteria defined using the DELETION option. This is an advanced option and should be used with care. A proper mesh refinement study should be performed to ensure that the solution is converged.

This option interacts with the contact logic to automatically rebuild the contact surfaces if elements are deleted from the surfaces. The automatic rebuild option only applies to surfaces that were created with the CONTACTGENERATE option or the BSSEG parameter.

The deletion parameter takes the form:

DELETION, <criteria name>,<value>

Element by element deletion criteria available, as shown in Table 12-1.

Table 12-1. Criteria Name and Description.

Criteria Name	Value	Description
NEGVOLUME	None	Delete if an element becomes so distorted that it has negative volume.
EFFSTRAIN	Real > 0 Required	Value of equivalent von Mises strain, $\bar{\varepsilon} = \sqrt{\frac{2}{3}\varepsilon_{ij}\varepsilon_{ij}}$, at which the element is deleted.
EQPS	Real > 0 Required	Value of equivalent plastic strain, at which the element is deleted.
PRINSTRAIN	Real > 0 Required	Value of maximum principal strain (most tensile), at which the element is deleted.
PPFA	None	Deleted when ply failure indeces exceed a value of 1 for all the plies.
BRITTLE	None	Deleted when fully cracked in all directions.

13. REFERENCES

1. Flanagan and Belytschko, “A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control,” Int. Jour. Numer. Meth. Eng, Vol. 15, No. 12, 1981.
2. Flanagan and Belytschko, “EigenValues and Stable Time Steps for the Uniform Strain Hexahedron and Quadrilateral,” Jour. Appl. Mech. 51 (1984).