

INGRID

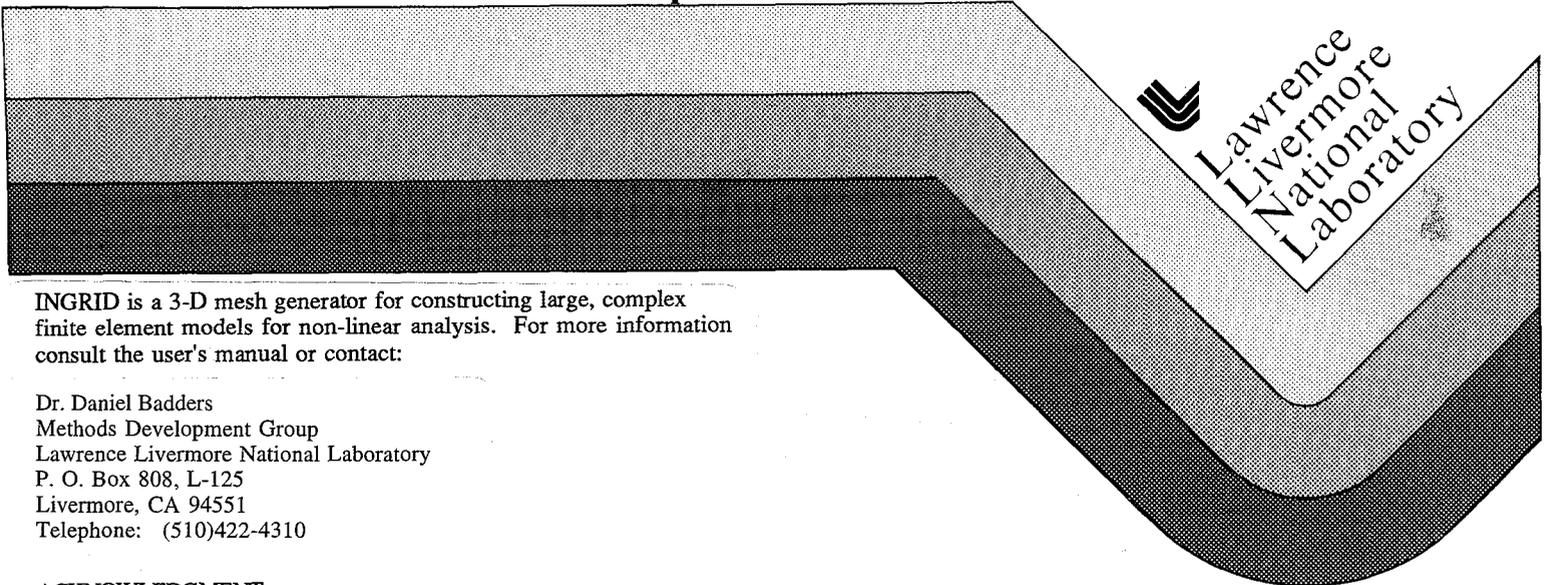
A 3-D MESH GENERATOR FOR MODELING NONLINEAR SYSTEMS

User Manual

Mark A. Christon
Donald Dovey
Methods Development Group
Mechanical Engineering

Originated by:
D. W. Stillman
J. O. Hallquist
R. B. Rainsberger

September 1992



INGRID is a 3-D mesh generator for constructing large, complex finite element models for non-linear analysis. For more information consult the user's manual or contact:

Dr. Daniel Badders
Methods Development Group
Lawrence Livermore National Laboratory
P. O. Box 808, L-125
Livermore, CA 94551
Telephone: (510)422-4310

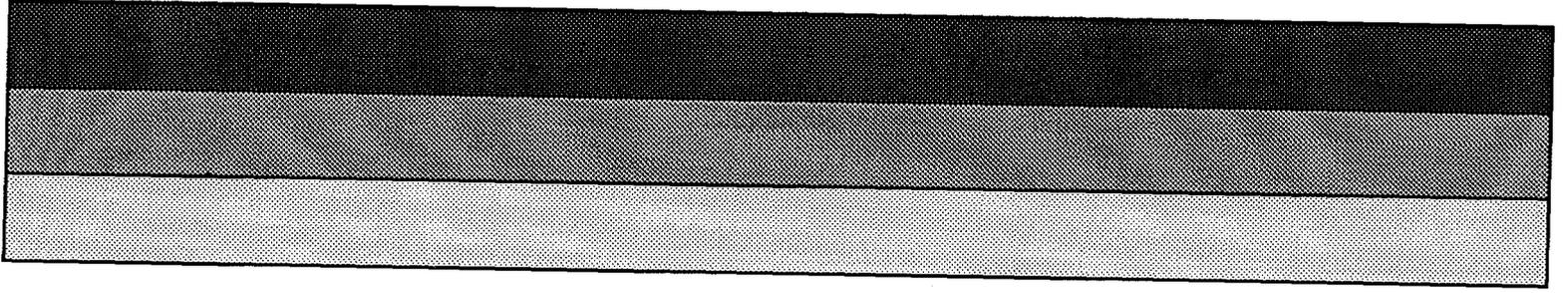
ACKNOWLEDGMENT:

Acknowledgment is given to the prior INGRID developers.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

***Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551***



INGRID

A 3-D MESH GENERATOR FOR MODELING NONLINEAR SYSTEMS

User Manual

**Mark A. Christon
Donald Dovey
Methods Development Group
Mechanical Engineering**

**Originated by:
D. W. Stillman
J. O. Hallquist
R. B. Rainsberger**

Draft - September 1992

Table of Contents

PREFACE	7
ABSTRACT.....	8
INTRODUCTION	9
MANUAL ORGANIZATION	10
OVERVIEW	11
NOMENCLATURE AND CONVENTIONS	12
UNITS	13
ELEMENT TYPES	14
PART TYPES	18
MESH GENERATION STRATEGIES	19
COORDINATE SYSTEMS	19
ANALYSIS OPTIONS	19
MATERIAL MODELS	21
INTERACTIVE FEATURES	21
BASIC CONCEPTS	23
Index Progression	23
Mesh Handles	24
Initial Coordinates	24
Regions	25
Advanced Index Notation	28
Circular and Spherical Parts	31
MESH & MODEL GENERATION COMMANDS - I	35
Command Index:	35
INGRID Input File Format	45
Global Control Commands	47
Analysis Options	47
DYNA3D Analysis Options	50
NIKE3D Analysis Options	54
TOPAZ3D Analysis Options	56
MONT3D Analysis Options	59
Fourier Series Definitions	61
General Control Commands	62
Geometric Transformations	63
Geometrical Part Control	66
Beam Section Definitions	66
Joint Definitions	67
Spring - Mass - Damper Systems	69

Table of Contents (cont.)

Global Initial & Boundary Conditions	70
Rigid Body Motion	71
Thermal Conditions	71
Load Curve Definitions	72
Line Definitions	73
Material Specification	77
Global Material Control Commands	77
Shell Materials	78
Establishing Local T-axis Orientation	78
Modifying Density and Viscosity	78
Material Detonation Points	79
Global Material Specification	79
DYNA3D Materials	83
DYNA3D Material Types	84
DYNA3D Material Type 1: Elastic	86
DYNA3D Material Type 2: Orthotropic Elastic	87
DYNA3D Material Type 3: Kinematic / Isotropic Elastic-Plastic	88
DYNA3D Material Type 4: Thermo-Elastic-Plastic	89
DYNA3D Material Type 5: Soil and Crushable Foam	90
DYNA3D Material Type 6: Viscoelastic	91
DYNA3D Material Type 7: Blatz-Ko Rubber	91
DYNA3D Material Type 8: High Explosive Burn	92
DYNA3D Material Type 9: Null Hydrodynamics	92
DYNA3D Material Type 10: Isotropic-Elastic-Plastic-Hydrodynamic	93
DYNA3D Material Type 11: Temperature Dependent, Elastoplastic	93
DYNA3D Material Type 12: Isotropic-Elastic-Plastic	95
DYNA3D Material Type 13: Elastic-Plastic with Failure Model	95
DYNA3D Material Type 14: Soil and Crushable Foam with Failure Model ..	96
DYNA3D Material Type 15: Johnson/Cook Strain and Temperature Sensitive	97
DYNA3D Material Type 16: Pseudo TENSOR Concrete/Geological Model	98
DYNA3D Material Type 17: Isotropic Elastic-Plastic Oriented Crack Model	99
DYNA3D Material Type 18: Power Law Isotropic Plasticity	99
DYNA3D Material Type 19: Strain Rate Dependent Isotropic Plasticity	100
DYNA3D Material Type 20: Rigid	101
DYNA3D Material Type 21: Thermal Orthotropic Elastic	101
DYNA3D Material Type 22: Composite Damage Model	103
DYNA3D Material Type 23: Thermal Orthotropic Elastic with 12 Curves	104
DYNA3D Material Type 24: Piecewise Linear Isotropic Plasticity	106
DYNA3D Material Type 25: Inviscid Two Invariant Geologic Cap Model ...	107
DYNA3D Material Type 26: Metallic Honeycomb	107

Table of Contents (cont.)

DYNA3D Material Type 27: Compressible Mooney-Rivlin Rubber	109
DYNA3D Material Type 28: Resultant Plasticity	109
DYNA3D Material Type 30: Closed-Form Update Shell Plasticity	110
DYNA3D Material Type 31: Frazer-Nash Rubber Model	111
DYNA3D Equation of State Types	112
DYNA3D Equation-of-State Form 1: Linear Polynomial	113
DYNA3D Equation-of-State Form 2: JWL	114
DYNA3D Equation-of-State Form 3: Sack "Tuesday"	114
DYNA3D Equation-of-State Form 4: Gruneisen	115
DYNA3D Equation-of-State Form 5: Ratio of Polynomials	115
DYNA3D Equation-of-State Form 6: Linear Polynomial with Energy Leak .	117
DYNA3D Equation-of-State Form 7: Ignition and Growth of Reaction in HE	118
DYNA3D Equation-of-State Form 8: Tabulated Compaction	119
DYNA3D Equation-of-State Form 9: Tabulated	119
DYNA3D Equation-of-State Form 11: TENSOR Pore Collapse	120
NIKE3D Materials	121
NIKE3D Material Type 1: Elastic	121
NIKE3D Material Type 2: Orthotropic Elastic	122
NIKE3D Material Type 3: Elastic-Plastic	123
NIKE3D Material Type 4: Thermo-Elastic-Plastic	123
NIKE3D Material Type 5: Soil and Crushable Foam	125
NIKE3D Material Type 6: Viscoelastic	126
NIKE3D Material Type 7: Thermo-Orthotropic Elastic	126
NIKE3D Material Type 8: Thermo-Elastic-Creep	128
TOPAZ3D Materials	129
TOPAZ3D Material Type 1: Isotropic	129
TOPAZ3D Material Type 2: Orthotropic	130
TOPAZ3D Material Type 3: Isotropic Temperature Dependent	130
TOPAZ3D Material Type 4: Orthotropic Temperature Dependent	131
TOPAZ3D Electrostatic Material Type 1: Isotropic	131
ADINA Materials	132
ADINA Material Type 1: Isotropic Linear Elastic	132
ADINA Material Type 9: Elastic Plastic	132
ADINA Material Type 10: Thermo-Elastic Plastic	133
Verbatim Materials	134
Part Tolerancing	135
Slide Surface Specification	136
Surface Definitions	138
Discrete 3-D Surface Definitions	138

Table of Contents (cont.)

Analytic 3-D Surface Definitions	140
Volume Definitions	144
MESH & MODEL GENERATION COMMANDS - II	145
Command Index:	145
Standard Part	161
Index Partition Progressions	163
Standard Part Control	164
General Control Commands	165
Geometric Transformations - Projections	167
Surface Projections	171
Multiple Surface Projections	174
Geometric Transformations - Point/Line Modifications	175
Geometric Transformations - Part Replication	178
Geometrical Part Control	179
Standard Part Boundary Conditions	181
Electromagnetic Loads and Boundary Conditions	181
Electrostatic Loads and Boundary Conditions	183
Sliding Interface Commands	183
Solid Mechanics Boundary Conditions	186
Thermal Loads and Boundary Conditions	189
Beam Element Generation	192
Edit and the In-line Calculator	193
MAZE Parts	196
MAZE Cross Section Definition	198
Electromagnetic Loads and Boundary Conditions	202
Sliding Interface Definitions	203
Solid Mechanics Boundary Conditions	204
Thermal Loads and Boundary Conditions	205
Beam Parts	208
Coordinate Specification	210
Element Generation Commands	210
Beam Part Options	211
Beam Element Attributes	211
Joint Definition	211
Slide Surface Specification	212
Importing Existing Mesh Parts	213
Import Part Control Commands	218
Interactive Commands	219

Table of Contents (cont.)

Display Commands	219
Rotation Commands	221
Translational Commands	221
Zooming and Panning Commands	222
Magnification Commands	222
Scaling Commands	222
Part and Materials Display Commands	223
Graphics Diagnostic Commands	224
Sliding Interface Display Commands	224
Symmetry Plane, Stone Wall and Non-Reflecting Boundary Identification ...	224
Node and Element Identification	225
Identification of Nodes in Regions Subjected to Load Curves	225
Identification of Regions Subject to Prescribed Boundary Conditions	226
Part Tolerancing	226
General Line Display	227
Load Curve Display	227
Non-Graphical Diagnostics Commands	228
Output Commands	228
LLNL Octopus Specific Commands	229
INGRID USER GUIDE	231
ABSTRACT	231
INTERACTIVE COMMANDS	231
Executing INGRID	234
Coordinate Systems	235
Perspective	237
Display Commands	239
Rotation Commands	242
Panning and Zooming Commands	244
Scaling	247
Displaying Part and Material Subsets	249
Restoring the Display	253
Viewing Line information	254
Part Tolerancing	256
Graphical Diagnostic Information Display - the di commands	260
Non-Graphical Diagnostic Information Display	266
Output Generation	269
Interactive Input Errors	270
INGRID EXAMPLES & PART LIBRARY	271
DYNA3D Examples	272

Table of Contents (cont.)

NIKE3D Examples 299
TOPAZ3D Examples 302
MONT3D Examples 307
INGRID Part Library 311
Advanced Modelling Examples 387
ACKNOWLEDGEMENTS 409
Bibliography 411
Index 414

PREFACE

INGRID has served as the primary 3-D mesh generator for the finite element analysis codes DYNA3D, NIKE3D and TOPAZ3D at the Lawrence Livermore National Laboratory for the past 10 years. In that time, INGRID has been generalized and adapted to the task of generating large, complex meshes for non-linear finite element analysis. The initial implementation of INGRID had as its roots the mesh generator, INGEN, from the Los Alamos National Laboratory. Since 1981, INGRID has seen extensive improvements in the areas of geometry modelling and interactive graphics due to the efforts and ingenuity of Doug W. Stillman, John O. Hallquist and Robert Rainsberger. Indeed, most of the current interactive graphics capability in INGRID is owed to the efforts of Robert Rainsberger. It is with respect for the previous INGRID code developers that the authors pledge to continue to enhance and extend INGRID's capabilities.

ABSTRACT

This document is intended to serve as the User Manual for the 1992 version of INGRID. INGRID is a 3-D mesh generator for constructing large, complex finite element models for non-linear analysis. The INGRID User Manual is divided into two parts: a reference section, and a user's guide. The reference section is intended to provide detailed information about mesh generation options, interactive commands, material models, and analysis code output options. The user's guide consists of examples which produce complete finite element models for DYNA3D, NIKE3D, TOPAZ3D and MONT3D. In addition, the user's guide provides examples of commonly used interactive commands, and a part library which illustrates how to construct a variety of mesh parts.

1.0 INTRODUCTION

INGRID is a generalized 3-D finite element mesh generator for modeling nonlinear systems. INGRID provides the capability to generate complex geometrical models using beam, shell and hexahedral elements. Similarly, boundary conditions, initial conditions and material properties can be specified for complex regions with a minimum of user input. Interactive graphics in INGRID offer the ability to probe and interrogate mesh structure, boundary conditions, slide surfaces, and radiation enclosures so that valid models can be easily constructed and rapidly verified.

As the primary mesh generator for DYNA3D and NIKE3D, INGRID has seen widespread use at the Lawrence Livermore National Laboratory (LLNL), and in industry over the past 10 years. As the demands for ease and speed of mesh generation increase, INGRID continues to evolve to meet the current expectations for a 3-D mesh generator in the engineering community. Versions of INGRID are available for a wide variety of computing platforms including CRAY UNICOS, Vax VMS, Vax Ultrix, Sun, Silicon Graphics and IBM RS/6000 machines. Because all development is done with a single source code and conditional compiling, all of the most recent code developments are immediately available in all supported versions of the code.

This manual is an attempt to pull together the information contained in the 5 earlier INGRID documents into a single, comprehensive user manual. It is hoped that by condensing the previous manuals, that the information contained in the current document will aid new INGRID users while still providing a complete command reference for the experienced user.

INGRID is part of a set of public domain codes developed in the Methods Development Group at LLNL. Other analysis codes include NIKE3D, DYNA2D, TOPAZ2D and TOPAZ3D, as well as PALM2D, MONT3D, MAZE, ORION and TAURUS. Currently all of the graphics capability in INGRID is supported using the DIGLIB public domain graphics library developed by Hal Brand at LLNL.

As a public domain code, INGRID has benefited greatly by the useful comments and suggestions of collaborators outside LLNL. It is hoped that the INGRID user community will continue to expand and provide feedback which is crucial to the continued evolution of INGRID.

1.1 MANUAL ORGANIZATION

The user manual is divided into two primary parts: a reference section and a user guide. The user guide is intended to be tutorial in nature and is aimed at the inexperienced INGRID user, but it is by no means a complete INGRID tutorial. The reference section provides the following:

- A description of INGRID's capabilities
- Basic INGRID Concepts
- How to run INGRID
- INGRID Mesh Generation Commands
- INGRID Material Models
- INGRID Interactive Commands

The user guide includes:

- Interactive Command Examples
- INGRID Examples for DYNA3D, NIKE3D, TOPAZ3D and MONT3D
- INGRID Part Library

2.0 OVERVIEW

INGRID capabilities include the rapid generation of large detailed meshes with complete boundary conditions, initial conditions and material specifications with a minimum of user input. INGRID provides the ability to generate meshes using the following element types in Cartesian, cylindrical or spherical coordinate systems:

Element Types:

- 2-node beams with variable section properties
- 4-node shell elements
- 8-node thick shell elements
- 8-node brick elements

Individual parts may be comprised of:

- beams
- shell and brick elements
- brick elements generated by extruding or spinning 2-D parts
- plane 2-D shell elements for 2-D analyses

Because INGRID can handle multiple parts, global commands are provided to:

- specify rigid body velocity and rotation
- initialize temperature in a given part
- specify materials by part
- set the total mass of a given material
- perform coordinate transformations
- specify slide surfaces
- initialize load curves

Part Duplication and Coordinate Transformation:

- Local and global coordinate transformations dynamically defined by ordered combinations of translations, rotations, reflections and scaling factors.
- Part duplication with coordinate transformation

Other INGRID features include:.

- Symbolic parameter definition.
- Interactive mesh diagnostics.
- Automated generation/application of boundary conditions such as nodal constraints, forces, pressures, etc.
- Stone walls, slide surfaces, joints and symmetry planes.
- Global and local control of part merging.
- Global and local definition of surfaces such as surfaces of revolution, conics, planes, cylinders, spheres, toroids, and generalized surfaces.
- Material and equation-of-state definition.
- Section properties for shells and beams.
- Nodal and element time history blocks.
- General algebraic operation on the coordinates using the existing coordinates, nodal indices, temporary variables, coordinate systems, filters, +, -, *, /, FORTRAN intrinsic functions, and uniform and normal random number generators.
- In-line algebraic expression evaluation.
- 3-D graphics with hidden line, perspective, rotations, translations, part/material selection, scaling, zooming, display of special features, initial conditions, node and element numbers.
- Interactive help package.

2.1 NOMENCLATURE AND CONVENTIONS

The following typeface conventions are followed throughout this document in order to allow the reader to easily distinguish between keywords commands and input variables. The input descriptions distinguish commands and keywords from parameters by using **boldface** text for keywords and commands, and *italic* text for parameters, e.g., **command** $p_1 p_2 \dots p_n$.

Commands and keywords are immediately followed by required parameters, and then by optional parameters. Optional parameters are always clearly delineated as optional in the reference manual. This convention applies to interactive keyboard input as well as commands and their associated parameters in the INGRID input file. A bold vertical line in the right margin is used to identify groups of parameters associated with a single INGRID command. By default, commands in each section are alphabetized except when a logical grouping of the commands makes more sense.

Only the first four characters of any and all commands are significant. The remainder of the text in a command may be included to improve the readability of the INGRID commands. Frequently commands have several alternate spellings such as `cy` and `cyli`. This is done for a variety of reasons, but primarily, this is done to protect the users. However, this also provides compatibility with older INGRID input files. Computer generated text such as error messages, or input prompts are presented in a `typewriter` font.

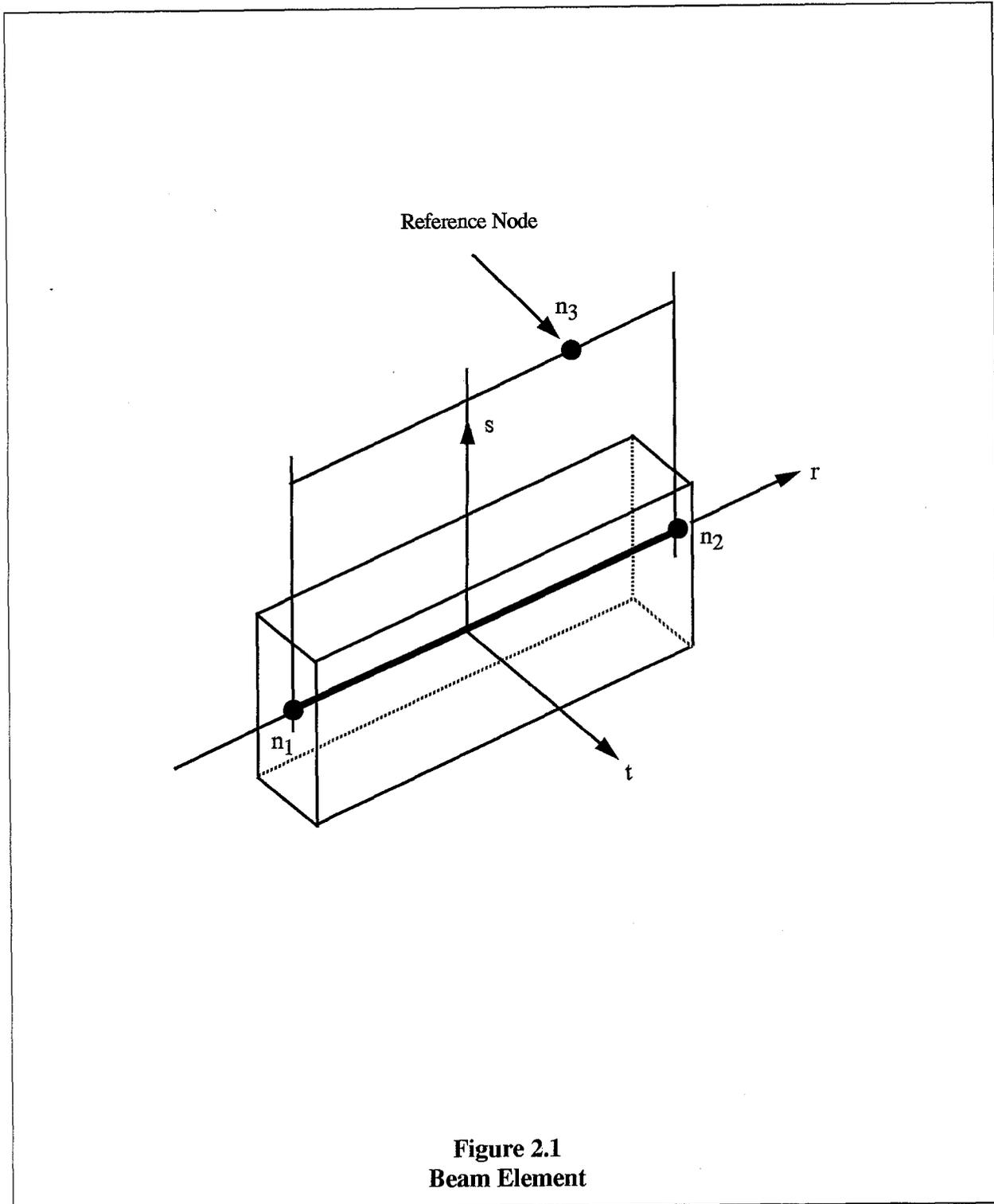
Many INGRID commands require the specification of points or regions in the mesh. Because the syntax is always the same, the following conventions for points, regions and index progressions are followed. The presence of `<point>` as a command parameter indicates that the command is expecting a sequence of three integer index values as: $i\ j\ k$. `<region>` as a command parameter requires that six integer reduced index values be entered as: $i_{min}\ j_{min}\ k_{min}\ i_{max}\ j_{max}\ k_{max}$. `<r-index>` indicates that a reduced index progression is required as: $i_1\ i_2\ \dots\ i_n;\ j_1\ j_2\ \dots\ j_n;\ k_1\ k_2\ \dots\ k_n$.

2.2 UNITS

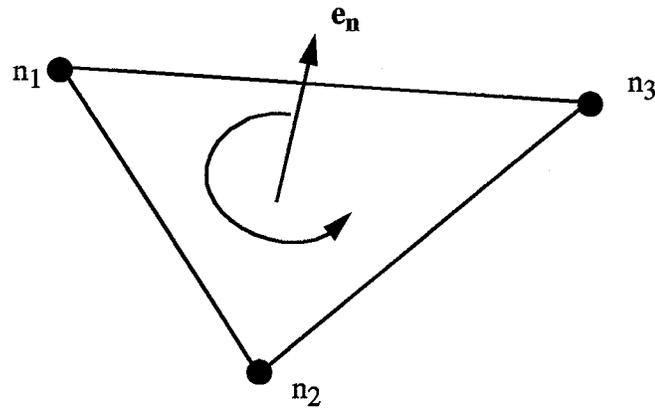
INGRID makes no assumptions about the presence of units in the input file. It is expected that all input to `ingrid` will be in a consistent unit system most convenient to the user. The ultimate selection of a unit system is specified by the user in the selection of an analysis output option. However, INGRID makes no attempt to resolve input variables which have inconsistent units when the analysis mesh file is created.

2.3 ELEMENT TYPES

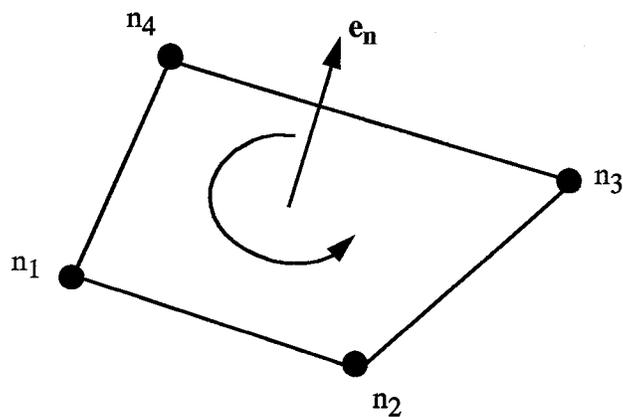
There are four basic element types currently supported in INGRID. Figure 2.1 illustrates the beam element and its nodes.



The shell element shown in Figure 2.2 requires 4 nodes, and is used for both 2-D and 3-D meshes. In the case of 2-D meshes, the shell elements lie in the x-z plane.



Degenerate Triangular Shell Element



Quadrilateral Shell Element

Figure 2.2
Shell Element

Figure 2.3 depicts the “thick” shell element and its local coordinate system.

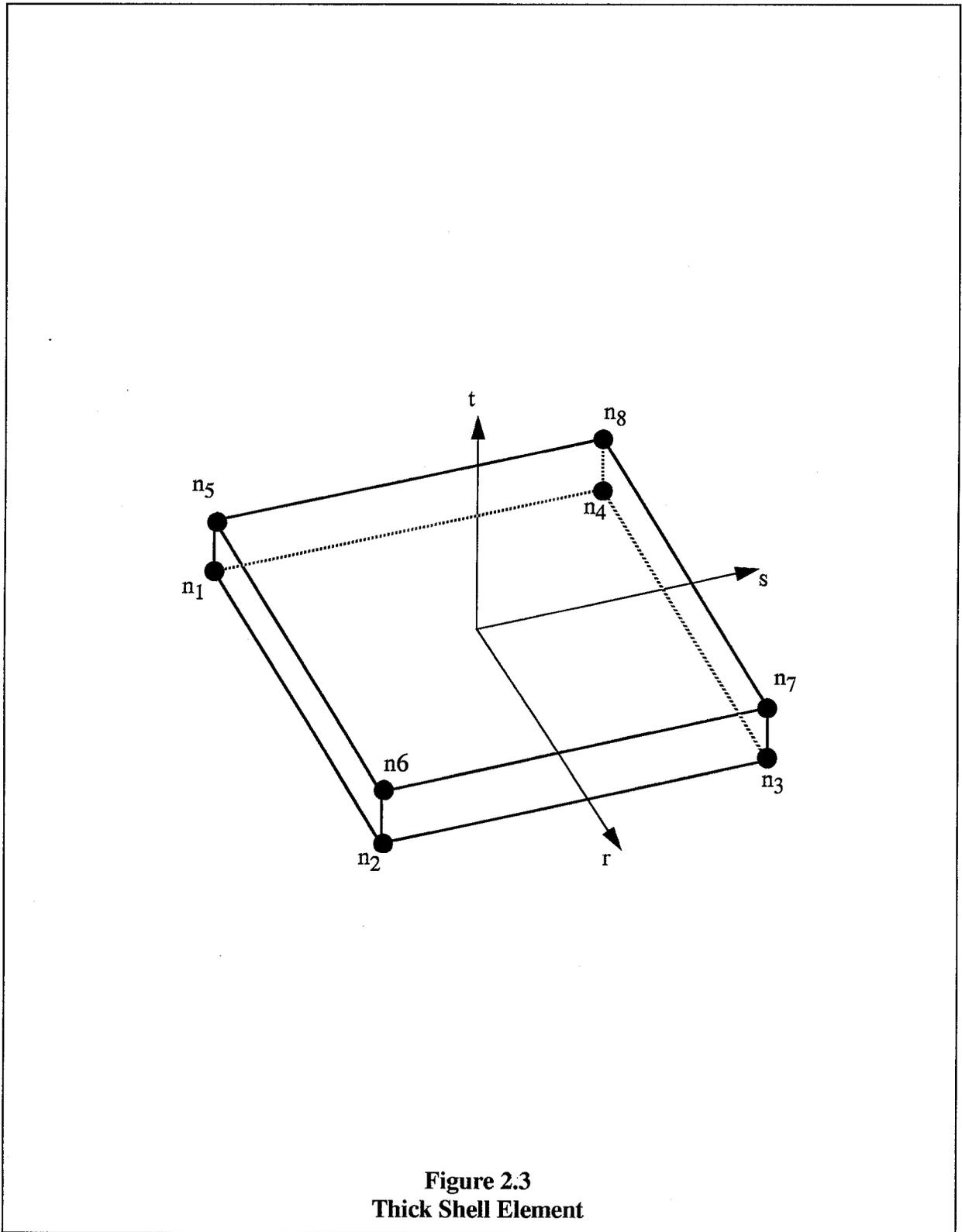
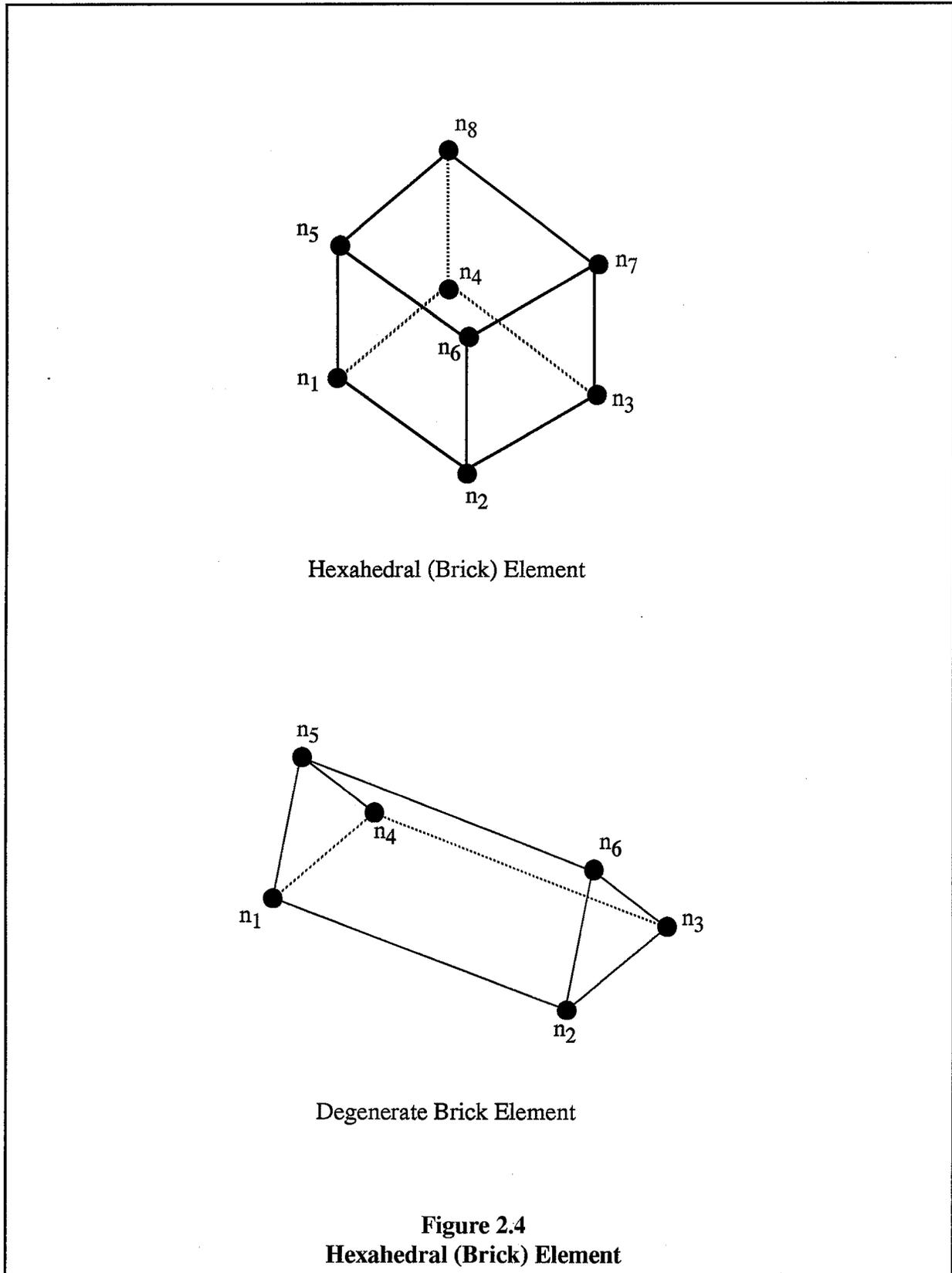


Figure 2.4 shows the configuration of the 8-node hexahedral element.



Degenerate 3-D elements are automatically generated in INGRID when necessary. Currently, there is no explicit control over the generation of the degenerate elements. One of the unique features of INGRID is its ability to generate meshes which contain beam, shell, thick shell and hexahedral elements complete with specified boundary conditions and initial conditions

2.4 PART TYPES

Currently, INGRID supports three basic part types, the beam part, the MAZE part and the standard part. The MAZE part can produce both shell and brick elements, while the standard part allows the generation of shell, thick shell and brick elements. The MAZE part allows the user to produce 3-D meshes either by extrusion or by spinning a given 2-D geometry. The standard part provides a rich set of surface mapping capabilities which enables complex 3-D geometries to be easily modelled and discretized.

Beam parts provide the following features:

- Generation in Cartesian, cylindrical or spherical coordinates
- Specification of displacement boundary conditions
- Part replication in either global or local coordinate systems
- Specification of initial translational and rotational velocities
- 1-D slide line support for NIKE3D and DYNA3D

Currently supported MAZE part features include:

- Support for MAZE line definitions
- Generation of 3-D elements by extrusion or “spinning”
- Part replication in either global or local coordinate systems
- 3-D slide surface support
- Specification of initial and boundary conditions

Features included in the standard part are:

- Generation of “in-line” beam elements simultaneously with brick elements
- Generation of both thin and thick shells
- A rich set of surface mapping functions for complex 3-D regions

- Generation of radiation enclosures
- Complete specification of initial and boundary conditions for both thermal and structural models

2.5 MESH GENERATION STRATEGIES

Because INGRID has the capability to incorporate multiple element types in a given mesh, and provides multiple part definition types, there are many possible strategies which can successfully be used in obtaining a 3-D mesh. The MAZE part is quite acceptable for the generation of 3-D objects which can be thought of as an outline which can be either extruded or spun. For example, a prismatic beam with a complicated cross section is a good example of when to use a MAZE part. In general, it is advisable to use the standard part when complicated surfaces are required or irregular cross sections are necessary.

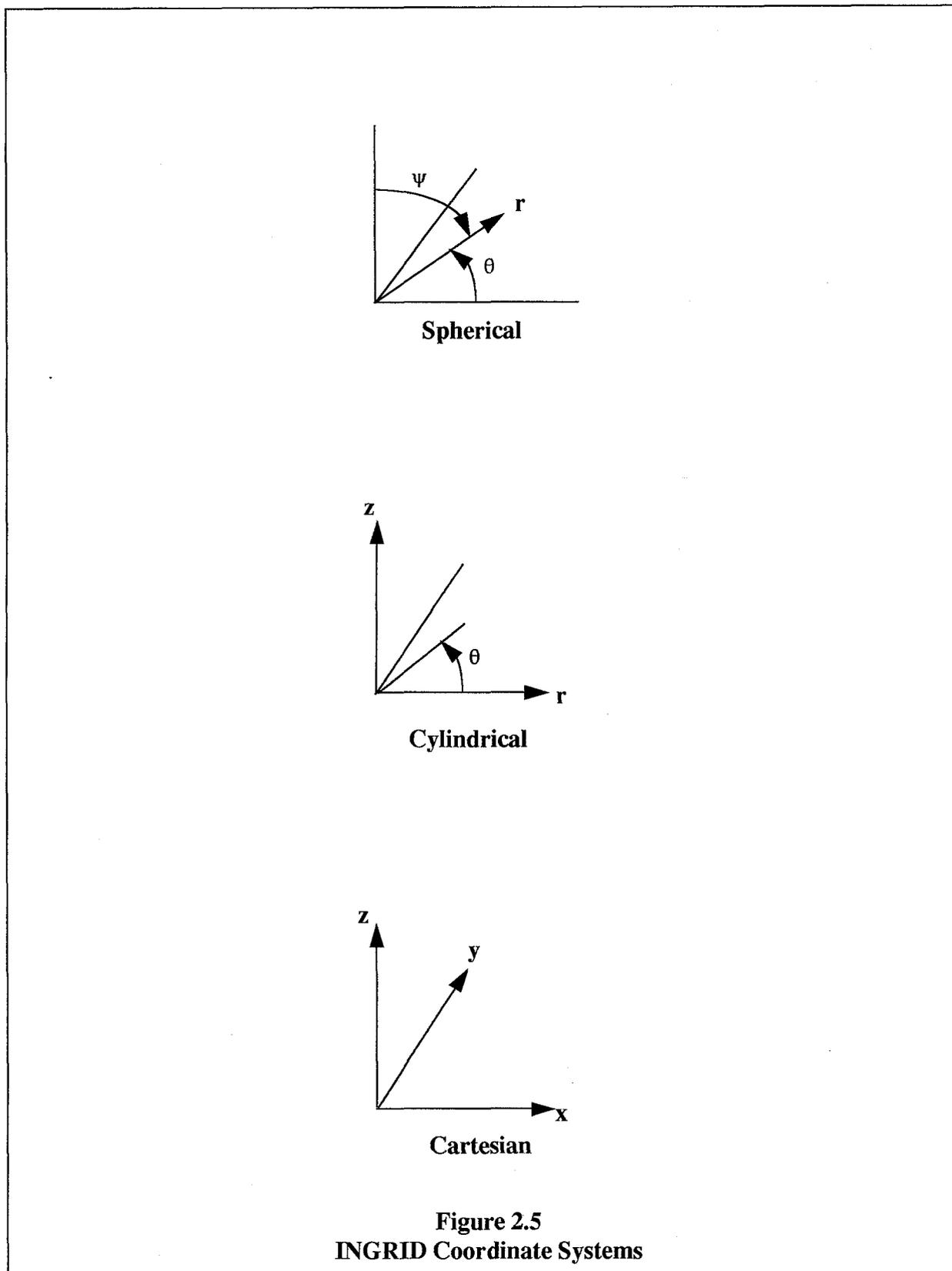
2.6 COORDINATE SYSTEMS

By default, INGRID uses a global Cartesian coordinate. However, INGRID provides the capability to use either cylindrical or spherical global coordinate systems. In addition, the beam, MAZE and standard parts can use local coordinate systems which may be Cartesian, cylindrical or spherical. Both local and global coordinate transformation can be defined by ordered combinations of translations, rotations, reflections and scaling operations Figure 2.5 illustrates the relationship between Cartesian, cylindrical and spherical coordinate systems.

2.7 ANALYSIS OPTIONS

As mentioned previously, INGRID is the primary mesh generator for the Methods Development Group suite of codes. However, INGRID provides basic support for many other analysis codes as well. A partial list of analysis code options include:

- DYNA2D
- NIKE2D
- DYNA3D
- NIKE3D
- TOPAZ2D
- TOPAZ3D
- MONT3D



2.8 MATERIAL MODELS

While generating meshes for complex 3-D geometries is a principal focus for INGRID, the specification of physical material models is also necessary for finite element analysis. Individual beam, MAZE and standard parts in INGRID may be associated with specific material behavior in the finite element model. While INGRID provides many of the commonly encountered material models in solid mechanics, it has not been possible to keep pace with the rate of development of material models. For this reason, a “verbatim” material model has been included in INGRID to enable the generation of meshes with experimental material models for the purposes of rapid prototype and test.

2.9 INTERACTIVE FEATURES

Graphical interrogation and inspection of a mesh is a crucial step in the assembly of a large, complex 3-D finite element analysis. INGRID attempts to provide some graphical diagnostics by making use of “glyphs” to highlight certain mesh features. Some of the typical uses for the diagnostics are:

- Validation of boundary conditions
- Inspection of slide surfaces
- Display of symmetry planes
- Inspection of internal/external mesh features
- Validation of radiation enclosures
- Exploded views of geometries with multiple parts

In addition, INGRID provides 4 levels of mesh display with interactive rotation, translation and scaling operations. Nodal point numbers and element numbers may viewed in order to graphically identify key regions in the mesh.

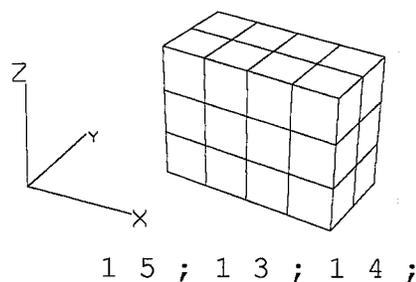
3.0 BASIC CONCEPTS

To build an INGRID model, an input file must be created. This file begins with a title line, followed by a keyword identifying the target analysis code. Next, global control commands may be included, and one or more *parts* may be defined. The input file is concluded with the `end` command.

Generally, an INGRID mesh will contain one separate part definition for each physical object in the finite element model. Objects with complex geometries are often more easily described using several INGRID parts, which are later merged together. This section describes the concepts and notation used to create parts with INGRID. An INGRID part definition begins with a keyword, followed by an *index progression*, a set of *initial coordinates*, and an optional number of *part control commands*. Each part definition is terminated with an `end` command. A detailed description of all part commands may be found in Chapter 4.

3.1 Index Progression

INGRID's fundamental mesh descriptor is the index progression, a compact, three dimensional notation used to describe the mesh within a part. The index progression naturally has three sets of numbers, one set for each spatial direction, separated by semicolons. For example, the index progression



describes a rectangular mesh with five nodes in the x-direction, three nodes in the y-direction, and four nodes in the z-direction.

Remark: The term “direction” has been used loosely here. More precisely, the index progression defines the index space or computational domain of the mesh. This domain is commonly referenced by (i,j,k) basis vectors, and is regular (rectangular) in nature. The index values then define

the ranges for “do-loop” indices used in operations on the computational domain. The mapping from the computational domain to the physical domain involves a choice of coordinate system (i.e. cartesian, cylindrical, or spherical), initial coordinates, transformations, and projections. For simplicity, in this section cartesian coordinates will be assumed, where the i- and x-directions coincide, the j- and y-directions coincide, etc.

3.2 Mesh Handles

The simple example above defined a five by three by four mesh, for a total of sixty node points. Consider now a similar index progression:

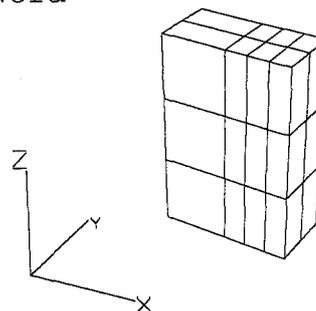
```
1 2 5 ; 1 3 ; 1 4 ;
```

This defines a mesh which still has sixty node points, but now an additional index, or *handle*, has been added in the x-direction. Handles provide partitions within the mesh. These can be used to apply surface projections, boundary conditions, and many other constraints which operate on a sub-region of the part. The simplest index progression contains two handles in each coordinate direction. Additional handles may be added as needed to build physical details into a part.

3.3 Initial Coordinates

Since handles are used to partition a mesh, each must be assigned an initial coordinate in space. The combination of an index progression and a set of initial coordinates completely defines a simple INGRID part. As an example, consider the input file and resulting part shown below:

```
simple INGRID part      c title
nk3d                   c analysis code keyword
start                  c begin part
1 2 5 ; 1 3 ; 1 4 ;   c index progression
1.0 2.0 3.0           c x-coordinates
1.0 2.0               c y-coordinates
0.0 3.0               c z-coordinates
end                    c end part
end                    c end input deck
```



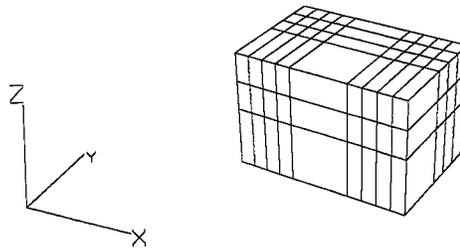
Notice the appearance of three x-coordinates, one for each x-direction handle in the index progression; two y-coordinates for their corresponding y-direction handles, etc. These coordinates define the initial location of the part in space. Further, since the index progression defines the number of node points between each handle, in each direction, then the initial coordinates define initial nodal spacing. By default, INGRID spaces node points uniformly between handles.

Notice in this example that the title, code keyword, and end statements are included, making this a complete INGRID input deck. The comment character 'c', followed by a blank space, has been used to allow comments within the input.

3.4 Regions

The parts constructed so far have been simple rectangular shapes. But the power of INGRID lies in its ability to convert these regular shapes into complex three-dimensional meshes. Here the utility of handles becomes clear: handles allow local *regions* of the mesh to be identified for modification by the many part commands. Regions can describe volumes, used for example to assign material type; areas, for operations such as surface projections; or lines, which might be used to apply boundary conditions along an edge.

Regions are described by referencing the handles which enclose them. For example, consider the part definition and resulting object:



```

start
1 5 6 10 ; 1 2 5 ; 1 2 4 ;           c index progression
2. 3. 4. 5.                          c x-coords (4 handles)
2. 3. 4.                              c y-coords (3 handles)
0. 1. 2.                              c z-coords (3 handles)
end

```

In the x-direction, the appearance of four handles in the index progression partitions the mesh into three regions. Similarly, two regions exist in each of the y- and z-directions. The upper right front corner of this part is identified using *<region>* notation as:

$$3 \ 1 \ 2 \quad 4 \ 2 \ 3$$

The *<region>* notation describes a region using two sets of three numbers. The numbers refer to handles in the index progression (not their values). Thus the triplet 3 1 2 identifies the third x-direction handle, the first y-direction handle, and the second z-direction handle. In this manner, each triplet defines a point in the mesh, where these handles intersect. By giving two triplets, two points are specified. These correspond to corners of the region. This notation therefore describes a region by identifying two opposing corners.

Remark: A region is more precisely defined as any regular group or block of nodes in index space. The *<region>* notation identifies a pair of ordered triplets $(i_{\min}, j_{\min}, k_{\min})$ and $(i_{\max}, j_{\max}, k_{\max})$ which give the minimum and maximum ranges of indices in computational space. Thus the region is defined by two particular “opposing corners”: those with minimum and maximum index values. For a solid region, all eight corners may be represented by combinations of these indices. The initial coordinates, or control points, associated with these indices then define a mapping of the region from computational space to physical space.

Notice that in the above *<region>* notation, the first set of numbers gives the beginning range of the region, the second gives the end of the region. Taken together then, the two sets of numbers define the range of handles over which the region spans, in each coordinate direction. For example, in the x-direction the region extends from the third to the fourth handle. In the y-direction, the region spans from the first to the second handle, and in the z-direction from the second to the third handle.

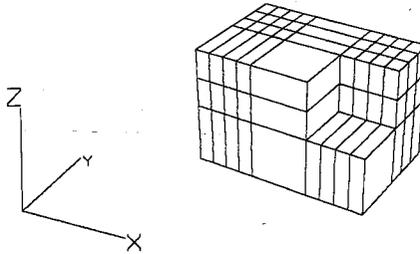
Describing a region in terms of the range of handles it spans motivates another notation, the *reduced index* or *<r-index>* notation. Here the region is described with three sets of numbers, giving the range of handles in each direction. Thus an equivalent representation using reduced index notation is:

$$3 \ 1 \ 2 \quad 4 \ 2 \ 3 \quad \langle \Rightarrow \quad 3 \ 4 \ ; \ 1 \ 2 \ ; \ 2 \ 3 \ ;$$

$$\langle \textit{region} \rangle \quad \langle \textit{r-index} \rangle$$

These two types of notation for describing a region are used frequently, but not interchangeably, throughout INGRID. Later in the manual, they are distinguished by the keywords *<region>* and *<r-index>* as shown above.

To illustrate the use of regions, consider the following examples where the delete command is used to remove sections of a part. The delete command requires a single argument: the region to be deleted.

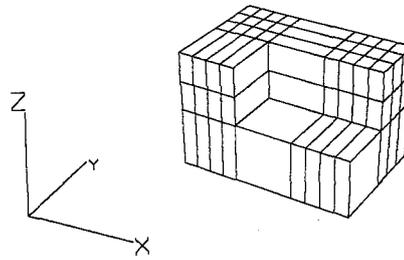


```

start
1 5 6 10 ; 1 2 5 ; 1 2 4 ;           c index progression
2. 3. 4. 5.                          c x-coords (4 handles)
2. 3. 4.                              c y-coords (3 handles)
0. 1. 2.                              c z-coords (3 handles)
d 3 1 2 4 2 3                         c delete (<region>)
end

```

Example: Right front upper corner of part removed using
d *<region>* command.



```

start
1 5 6 10 ; 1 2 5 ; 1 2 4 ;      c index progression
2. 3. 4. 5.                    c x-coords (4 handles)
2. 3. 4.                        c y-coords (3 handles)
0. 1. 2.                        c z-coords (3 handles)
di 2 4 ; 1 2 ; 2 3 ;           c delete (<r-index>)
end

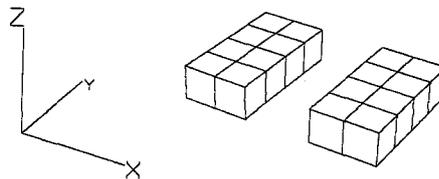
```

Example: Middle front upper and right front upper regions of part removed using **di** *<r-index>* command.

3.5 Advanced Index Notation

INGRID's notation described above accepts three additional features to allow shorthand descriptions of parts and regions. These are *zero*, *omitted*, and *negative* handles and entries.

Zero entries may be included in an index progression or *<r-index>* to skip a portion of the mesh. For example, including a zero in an index progression can cause disjoint parts to be generated:



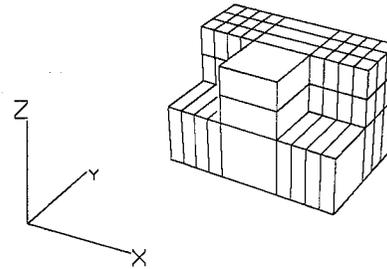
```
1 3 0 4 6 ; 1 5 ; 1 2
```

Similarly, in an *<r-index>* the appearance of a zero causes a region to be skipped:

```
di 1 2 ; 1 2 ; 2 3 ;
di 3 4 ; 1 2 ; 2 3 ;
```

- or -

```
di 1 2 0 3 4 ; 1 2 ; 2 3 ;
```

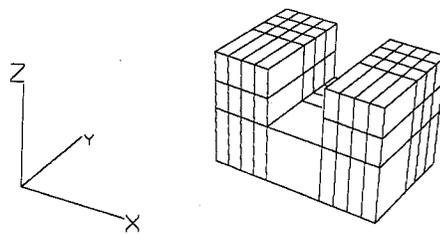


```
1 5 6 10 ; 1 2 5 ; 1 2 4 ;
```

Example: The *<r-index>* zero handle notation allows two **di** commands to be combined.

A zero entry in a *<region>* has an entirely different meaning. In a *<region>*, zeros signify that the entire range of handles in a particular direction are to be considered. The same effect can be obtained in the *<r-index>* notation by omitting a set of handles. For example, equivalent methods are given below to describe a *<region>* or *<r-index>* which includes all of the handles in the y-direction:

```
1 5 6 10 ; 1 2 5 ; 1 2 4 ;
```



```
d 2 1 2 3 3 3
```

- or -

```
d 2 0 2 3 0 3
```

<region>

```
di 2 3 ; 1 3 ; 2 3 ;
```

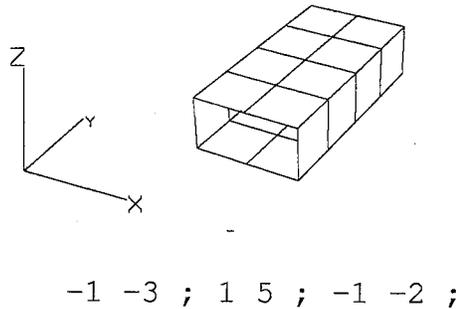
- or -

```
di 2 3 ; ; 2 3 ;
```

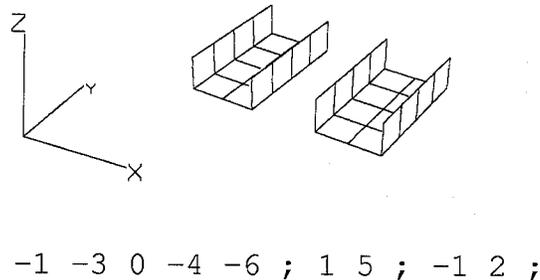
<r-index>

Notice that for the zero and omitted entry notations described above, equivalent methods exist which use only basic notation. In these cases, the advanced notation merely allows convenient and compact description of a mesh. The negative index notation, however, provides a unique capability for describing four-noded or shell element meshes.

A negative handle may appear in either an index progression or an $\langle r\text{-index} \rangle$. This identifies that the plane normal to that direction is to be considered. For example, consider the following index progression and associated part:



The appearance of two negative handles in the x-direction (-1 -3) generates two planes normal to the x-direction. The values of these handles indicate that there be three node points in the x-direction. Likewise, the negative handles in the z-direction (-1 -2) identify planes normal to the z-axis. The resulting part therefore contains only planes normal to the x- and z- directions, and is five nodes deep in the y-direction. This simple notation allows solid parts to be transformed into hollow parts with very few modifications to the INGRID part description. A further example illustrates how several advanced notational features may be combined to generate two shell parts with a single index progression:

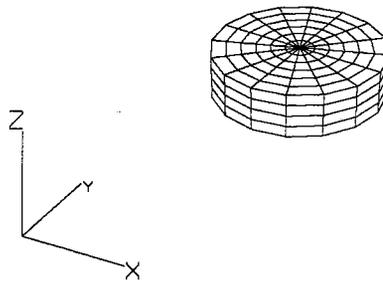


3.6 Circular and Spherical Parts

The last basic concept to be presented is the unique strategy used in INGRID to create circular and spherical parts. To simplify the presentation, only circular parts will be covered. The extension to spherical geometry is straightforward, and many examples of the technique are included at the end of the manual.

The most obvious method for constructing a solid right circular cylinder is to employ cylindrical coordinates. This is done using the part control command `cyli` (see chapter 4). In this case the index progression takes a new interpretation: rather than the (x,y,z) directions, the three sets of handles and associated initial coordinates correspond to the (r,θ,z) directions. Coordinates in the θ -direction are given in units of degrees. Consider the following example, where a “hockey puck” of radius 2 and height 1 is constructed:

```
Hockey Puck #1 - Cylindrical Coordinates
nk3d
start
1 7 ; 1 17 ; 1 5 ;
0.0 2.0
0.0 360.0
-0.5 0.5
cyli
end
end
```



At this point, a postulate must be introduced: that prismatic or “wedge” elements are to be avoided. This is done purely in the interest of the finite element analysis code, where such elements often generate inaccuracies. From the point of view of INGRID, the above mesh is perfectly valid.

Ideally then, the hockey puck mesh must be modified to contain rectangular elements at its center. Conceptually, this is accomplished in INGRID by starting with a rectangular mesh, and deleting the four corner regions. Next the newly exposed edges are folded together, closing the corner voids. Finally, the outer perimeter is projected radially to the desired cylindrical surface. This process is illustrated by the input file below:

Hockey Puck #2 - rectangular coordinates

nk3d

start

1 4 7 10 13 ; 1 4 7 10 13 ; 1 5 ;

-1. -1. 0. 1. 1.

-1. -1. 0. 1. 1.

-.5 .5

c == delete 4 corners ===

di 1 2 0 4 5 ; 1 2 0 4 5 ; ;

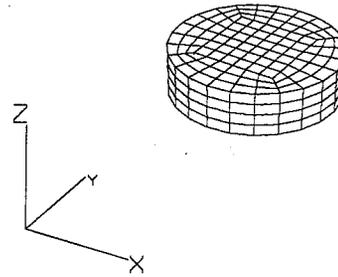
c == project outer surface onto cylinder ==

sfi 2 4 ; -1 -5 ; ; cy 0. 0. 0. 0. 0. 1. 2.0

sfi -1 -5 ; 2 4 ; ; cy 0. 0. 0. 0. 0. 1. 2.0

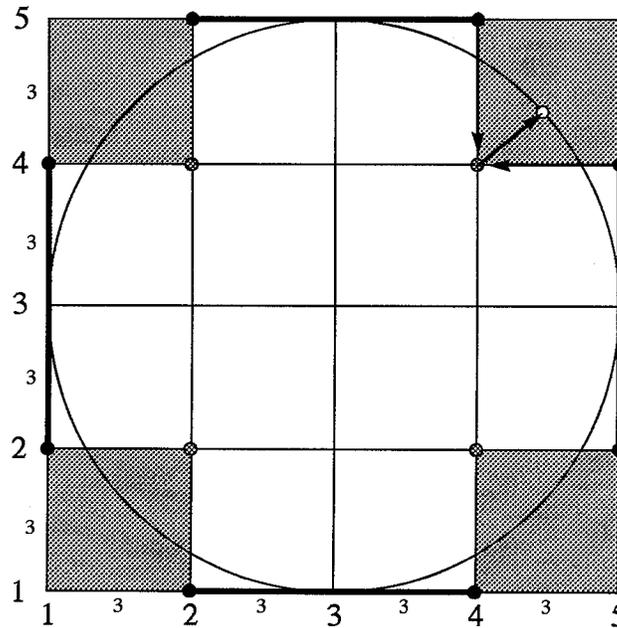
end

end



Remark: The concept of deleting a region, and then folding the newly exposed surfaces together, is unique to INGRID. Notice that in order to avoid mesh discontinuity in the resulting part, the newly exposed surfaces must have identical mesh densities, such that when folded together, the surface meshes are perfectly aligned. INGRID then performs a nodal merging process, wherein duplicate nodes are removed.

The example above clearly demonstrates that a diagramming technique is essential to INGRID mesh building. The diagram below represents the index progression and projection strategy used to build the hockey puck part.



Example: Diagram for hockey puck part shows schematic view of regions and handles in x-y plane, allows *<r-index>* values to be easily selected.

Since the part is uniform in the z-direction, a top view is sufficient to represent the operations used to form the cylinder. The index progression contains five handles in each of the x- and y-directions, represented schematically by light grid lines, with corresponding (large) numbers across the lower and left edges of the diagram. The light grid lines delineate 16 regions which these handles form. The small numbers along the edges of the diagram indicate the number of finite elements contained in each region. This representation allows *<r-index>* values to be easily selected as part commands are formed.

The first part command uses compact notation to delete the four corner regions, highlighted in grey in the diagram. The x-direction *<r-index>* component 1 2 0 4 5 “catches” nodes between handles 1 and 2, skips those from 2 to 4, and catches nodes between handles 4 and 5. The same *<r-index>* components are used in the y-direction.

Following the delete command, a pair of `sfi` commands perform a projection of the outer surface of the mesh to a cylindrical surface. In the x-y plane of the diagram, these surfaces are represented by the four heavy black lines. The cylindrical target surface is represented by the circle. Again utilizing compact *<r-index>* notation, the first `sfi` command identifies the upper and lower heavy line segments for projection onto the cylinder, and the second operates on the left and right segments.

INGRID's projection algorithm moves nodes in a direction normal to the target surface. Thus considering the schematic, projecting the heavy line segments directly onto the dashed circle would result in gaps between the segments, since the segment endpoints would project to different locations on the circle. This difficulty is circumvented by assigning equivalent initial coordinates to these endpoints.

Notice in the above input deck that the initial coordinates are identical for handles 1 and 2, and also for handles 4 and 5, in both the x- and y-directions. This results in an initial geometry where the outermost regions are collapsed. Most importantly, the endpoints of the heavy line segments in the schematic lie initially at the same location. Thus they project to the same final location on the cylindrical surface (arrows in schematic diagram). This basic concept, projection of initially collapsed regions, provides the last prerequisite for browsing the examples at the end of the manual.

4.0 MESH & MODEL GENERATION COMMANDS - I

This is the first of two chapters which present the INGRID commands used to describe the physical geometry, degree of discretization, material data, boundary conditions and analysis options. This chapter describes the form of the INGRID input files, global control commands, analysis options, line, surface and volume definitions. Chapter 5.0 describes the part level mesh and model generation commands, and includes a discussion of interactive commands.

Command Index:

Analysis Output Options	47
adina ADINA mesh format	48
ansys ANSYS mesh input format	48
ape Ohio State's apE graphics format	48
bdiv B-Division's input format	48
cube Tavlove's EM code format	48
dn2d DYNA2D mesh input format	48
dn3d DYNA3D mesh input format	48
dn3d88 1988 DYNA3D mesh input format	48
dn3d88s 1988 DYNA3D small input format	48
es3d TOPAZ3D electrostatic input format	48
face	48
gem3	48
gem3d GEM3D mesh input format	48
gemini GEMINI mesh input format	48
maxwell3	48
monte MONT3D radiation enclosure format	48
nast NASTRAN mesh format	48
nk2d NIKE2D mesh input format	48
nk3d NIKE3D mesh input format	48

Command Index: (cont.)

nk3d84	1984 NIKE3D mesh input format.....	48
nk3d89	1989 NIKE3D mesh input format.....	48
pafe	48
poly3d	Output a simple ASCII surface polygon file	49
rigid	Generate a mesh for DYNA3D with rigid bodies.....	49
sap4	Generate a mesh which can be used with SAP4.....	49
tz2d	TOPAZ2D mesh input format	49
tz3d	TOPAZ3D mesh input format	49
DYNA3D	Analysis Options	50
brode	Define a Brode function	50
coef	Specify the coefficient values for a modified Brode function	50
gfun	Specify the gfun values for a modified Brode function.....	50
rang	Specify the rang values for a modified Brode function	50
facrx	Set the dynamic relation factor	50
grav	Specify the gravity acceleration vector.....	50
hydt	Activate the flag for dumping hydrodynamic variables	50
itrx	Specify the number of iterations between convergence checks	50
itss	Specify the initial time step size	51
lcmax	Specify the load curve number that limits the maximum time step size	51
nrest	Specify the number of time steps between dumps of the restart file.....	51
plastic	Set the plane stress plasticity option	51
plti	Specify the time interval for dumping data to the graphics database	51
prtflg	Set the printout flag for the element time step size.....	51
prti	Specify the time interval for dumping data to the output file for printing.....	51
rgbm	Rigid body merge.....	51
scftrx	Set the scale factor for the dynamic relaxation time step	51

Command Index: (cont.)

sfor	Specify the shell theory options to use	51
snrs	Specify the Hughes-Liu shell normal update option	51
ssdm	Activate flag for dumping shell strain tensors to the TAURUS database	51
stat	Specify the number of time steps between status reports	51
stss	Set the shell step size option	51
stup	Specify shell thickness change option	51
teo	Activate the thermal effects option	52
term	Terminate the simulation at time t	52
tolrx	Set the convergence tolerance for dynamic relaxation	52
tsmin	Set the reduction factor, s , for the minimum time step size.....	52
tssf	Specify the time step size scale factor	52
\$n	Enter a comment which will be passed to the DYNA3D input file.....	53
NIKE3D Analysis Options		54
anal	Specify the type of analysis	54
bwmo	Bandwidth minimization in NIKE3D	54
dctol	Specify the displacement convergence tolerance	54
delt	Specify the simulation time step size.....	54
ectol	Specify the energy convergence tolerance.....	54
grav	Specify the gravity acceleration vector.....	54
itss	Specify the initial time step size	54
iplt	Specify the time interval for dumping data to the graphics database	54
iprt	Specify the time interval for dumping data to the output file for printing.....	54
msrf	Specify the maximum number of stiffness reformations per time step	54
nbei	Specifies that NIKE3D take n time steps between equilibrium iteration	54
nbsr	Specify that NIKE3D take n time steps between the reformation	54
neig	Specify the extraction of n Eigenvectors	54

Command Index: (cont.)

nibsr	Specify the maximum number of equilibrium iterations	55
nsmd	<i>n</i> specifies the type of nonlinear solution algorithm employed in NIKE3D	55
nstep	Specify the desired number of time steps in the simulation	55
sbrf	Specify the number of time steps between the dumps to the restart file	55
teo	Activate the thermal effects option	55
term	Terminate the simulation at time <i>t</i>	55
TOPAZ3D Analysis Options		56
bwmo	Bandwidth minimization will be performed in TOPAZ3D	56
dcmx	Set the desired maximum temperature change in each time step	56
dctol	Specify the convergence tolerance	56
delt	Specify the simulation time step size	56
dtmax	Set the maximum time step size	56
dtmin	Set the minimum time step size	56
flux	Nodal heat flux computations are performed in TOPAZ3D	56
iunit	Specifies the temperature units	56
iplt	Specify the number of time steps between dumps to the graphics database	56
iprt	Specify the number of time steps between dumps to the output file	56
linear	The conduction problem is linear	56
mfts	Modification factor for increasing or decreasing the time step	56
mrdi	Set the maximum number of radiosity iterations	56
msrf	Maximum number of conductivity matrix reformations per time step	56
nbei	Specifies that TOPAZ3D take <i>n</i> time steps between equilibrium iteration	57
nbsr	Specify that TOPAZ3D take <i>n</i> time steps between the reformations	57
nibsr	Specify the maximum number of equilibrium iterations	57
nip1	Specify the first Newmark integration parameter	57
nonlinear	The conduction problem is nonlinear	57

Command Index: (cont.)

nssd	Specify the number of surface subdivisions for radiation	57
phase	Phase change computations are performed in TOPAZ3D	57
radiation	Specifies the type of radiation calculations to be made	57
rband	Set the radiation bands	57
retol	Specify the radiosity convergence tolerance to	57
relax	Specify the relaxation parameter	57
sbc	Set the Stefan-Boltzmann constant	57
sbrf	Specify the number of time steps between the dumps to the restart file	57
steady	Specifies a steady state computation	57
step	Specifies the type of time integration to use.....	57
term	Terminate the simulation at time t	58
timin	Set the initial analysis time to t	58
transient	Specifies a transient computation	58
MONT3D Analysis Options		59
delt	Set the cone angle increment for material properties	59
ibug	Activates debugging	59
idata	Forces data checking only to be executed	59
iprnt	Forces the exchange factor matrix to printed in the output	59
nbands	Specify the number of radiation wavelength bands.....	59
ndivx	Set the number of emission points in the local x-direction.....	59
ndivy	Set the number of emission points in the local y-direction.....	59
nlost	Set the maximum number of lost photons	59
nphton	Specify the number of photons emitted per surface	59
nrefs	Specify the maximum number of reflections per photon	59
nshade	Specify the type of shading algorithm to use.....	59
numcat	Specify the number of concatenated surfaces in the radiation enclosure	59

Command Index: (cont.)

xscale	Set the scale factor for the x-dimension.....	60
xshift	Specify the amount to shift the enclosure geometry in the x-direction	60
yscale	Set the scale factor for the y-dimension.....	60
yshift	Specify the amount to shift the enclosure geometry in the y-direction	60
zscale	Set the scale factor for the z-dimension.....	60
zshift	Specify the amount to shift the enclosure geometry in the z-direction	60
Fourier Series Definitions		61
fsd	Define Fourier series.....	61
rfsd	Define Fourier series with random phase shifts.....	61
General Control Commands		62
box	Batch command for specifying the Octopus box number.....	62
noplot	Do not perform any of the interactive graphics operations	62
parameter	Set parameter values	62
pause	Causes INGRID to pause for approximately n seconds	62
tmds	Specify the TMDS number to be used (see also tv)	62
Geometric Transformations		63
cscs	Scale all nodal coordinates by s	63
cyli	Nodal point coordinates are converted from cylindrical to Cartesian	63
exch	Performs a permutation of the coordinate axes	63
gct	Perform a sequence of n global coordinate transformations.....	63
grep	Performs global part replication	64
gexc	Perform a permutation of the coordinate axes	64
gmi	Perform global material number increment	64
lev	Define part transformation sequence	64
loof	Plate elements do not lie in a plane.....	64
pslv	Begin performing part transformation sequence	65

Command Index: (cont.)

pplv	Terminate the part transformation stack operations	65
sph	Nodal point coordinates are converted from spherical to Cartesian	65
xoff	Global x-coordinate offset	65
yoff	Global y-coordinate offset	65
zoff	Global z-coordinate offset	65
xzca	Scale all x-coordinates	65
ysca	Scale all y-coordinates	65
zsca	Scale all z-coordinates	65
Geometrical Part Control (cont.)		66
bsd	Define beam section.....	67
jd	Begin the definition of joint.....	68
spd	Identify spring, mass, damper combination.....	69
Initial & Boundary Conditions		69
arri	Generate arrival times for pressure.....	70
plane	Input plane definitions	71
rotation	Assign an initial rigid body rotation	71
velocity	Assign a rigid body velocity	71
temp	Set initial temperature of T	71
Load Curve Definitions		72
flcd	Force load curve.....	72
lcd	Define load curve.....	72
prcd	Define variable pressure curve.....	72
thcd	Define variable thickness curve.....	73
vecd	Define variable velocity curve.....	73
Line Definitions		73
edgefile	Specify that line definitions are to be read from <i>file</i>	73

Command Index: (cont.)

lcc	Define lines consisting of circular arcs	73
lrl	Define lines consisting of radial lines	73
ld	Line definition.....	73
rlns	Read line definitions from the edge file.....	76
rln	Read a single line definition from the edge file.....	76
rseg	Read the next line segment definition from the edge file	76
Global Material Commands		77
detp	Define detonation points	79
mate	The default material number is set to <i>m</i> for all of the parts	79
mbc	Generate electromagnetic material interface conditions	79
rho	Specify that material <i>n</i> have density ρ	79
thic	Shell elements have the a thickness of <i>t</i>	79
tmm	The total mass of material <i>n</i> is <i>t</i>	79
mat	Define material model	80
DYNA3D Materials		83
DYNA3D Equation-of-State Types.....		112
NIKE3D Material		121
TOPAZ3D Materials.....		129
ADINA Materials		132
Verbatim Materials		134
Part Tolerancing.....		135
bptol	Specify a between-part tolerance	135
ptol	Specify a tolerance.....	135
stol	Specify a tolerance to be used in surface tolerancing	135
stp	Specify a tolerance to be used in surface tolerancing	135
tol	Specify a tolerance to be used in volumetric tolerancing	135

Command Index: (cont.)

tp	Specify a tolerance to be used in volumetric tolerancing	135
ztol	Specify a minimum absolute value for coordinates	135
Slide Surface Specification		136
si	Input slide surface control data	136
Surface Definitions		138
ds	Define a general 3-D surface for input to INGRID	138
mkds	Make a binary database of the discrete 3-D surfaces	139
rdds	Read a binary database of the digitized 3-D surfaces	139
sd	Surface definition number	140
Volume Definitions		144
vd	Volume definition	144

4.1 INGRID Input File Format

The following conventions are used for all INGRID input files. All INGRID commands are format free, must be blank delimited, and are case insensitive. Comments may be included in the input file by using a “c” anywhere in the input followed by a blank space and the comment. (A comment at the end of a command must include a “c” preceded by a blank.) Regions of the input file which are enclosed between any pair of braces “{ }” will be ignored.

INGRID mesh generation commands are not order dependent with the following exceptions:

- The first line of an INGRID input file must contain an 80 character title or be blank.
- The definition of an INGRID entity must occur before it is referenced in a command or a part. For example, a line definition must occur before it is used to generate a Maze part.
- Material data and code execution options cannot be interpreted until an analysis code output option has been specified.
- Some commands are order dependent, e.g., rotating local coordinate systems about different axes.
- Any commands following the final **end** command in an INGRID input file will be interpreted as interactive commands. For example, tolerancing commands, viewing options, etc. may be placed after the final **end** command.
- The typical form of an INGRID input file is shown below:

INGRID Input File

```
Title Card (80 characters or less)
Control Commands
Part Definitions
Control Commands
Part Definitions
...
end
Interactive Commands
```

The “end” statement terminates all input processing in INGRID, and any part or material definitions and control commands will be interpreted as interactive commands if they occur after the last end statement.

The command descriptions in this chapter contain keywords and variables. The input descriptions distinguish keywords from variables by using **boldface** text for the keyword and *italic* text for variables, e.g.:

keyword *variable₁ variable₂ ... variable_n*

Options for a command immediately follow the command in the documentation and in the INGRID input file. Only the first four characters of any and all commands are significant. The remainder of the text in a command may be included to improve the readability of INGRID commands. Frequently commands have several alternate spellings such as cy and cyli. This is done for a variety of reasons, but primarily, this is done to protect the users. However, this also provides backward compatibility with older INGRID input files. Compatibility with MAZE, ORION and TAURUS is maintained by using multiple spellings for a command. Note that not all the possible spelling permutations of all commands are presented here.

This chapter of the manual is organized according to command type beginning with global control commands, and is intended to be used as a strictly as reference. Following the global control commands is a definition of the beam, MAZE, and standard part syntax. Finally, interactive commands for graphical model interrogation are presented

4.2 Global Control Commands

Control commands may be thought of as global commands which control the general features of the finite element model being generated. The control commands may appear in any order in the INGRID input file. In this section, the syntax and scope for all global control commands are detailed in the following order:

- Analysis Options
- Fourier Series Definitions
- General Control Commands
- Geometric Transformations
- Geometrical Part Control
- Initial & Boundary Conditions
- Load Curve Definitions
- Material Specification
- Part Tolerancing
- Slide Surface Specification
- Line Definitions
- Surface Definitions
- Volume Definitions

4.2.1 Analysis Options

INGRID analysis options may be specified at any point in the INGRID input file. The following commands are used to select the type of mesh file to be generated, and to control specific features of the analysis codes. Note that some INGRID commands are not functional until the analysis option is selected, making it advisable to specify the analysis output option before any analysis code specific commands in the INGRID input file. In this section, the analysis output options are described, then analysis options specific to DYNA3D, NIKE3D, TOPAZ3D and MONT3D are presented.

All of the 2-D output options (**dn2d**, **nk2d**, **tz2d**, etc.) below require that the mesh be generated in the x-z plane (i-k index space for Cartesian coordinates, or i-j index space for cylindrical coordinates) coordinates.

Analysis Output Options

<u>Command</u>	<u>Description</u>
adina*	ADINA mesh format.
ansys*	ANSYS mesh input format.
ape	Ohio State's apE graphics format.
bdiv*	B-Division's input format.
cube*	Tavlove's EM code format.
dn2d*	DYNA2D mesh input format.
dn3d	DYNA3D mesh input format. This option generates the DYNA3D input file format consistent with the most current version of DYNA3D.
dn3d88	1988 DYNA3D mesh input format.
dn3d88s	1988 DYNA3D <i>small</i> input format. This enables generation of the small 1988 DYNA3D input file format using an E20.0 format for nodal point coordinates.
es3d*	TOPAZ3D electrostatic input format.
face*	
gem3*	
gem3d*	GEM3D mesh input format.
gemini*	GEMINI mesh input format.
maxwell3*	
monte	MONT3D radiation enclosure format.
nast*	NASTRAN mesh format.
nk2d*	NIKE2D mesh input format.
nk3d	NIKE3D mesh input format. This options generates the NIKE3D input file format consistent with the most current version of NIKE3D.
nk3d84	1984 NIKE3D mesh input format.
nk3d89	1989 NIKE3D mesh input format.
pafe*	

Analysis Output Options (cont.)

<u>Command</u>	<u>Description</u>
poly3d	Output a simple ASCII surface polygon file.
rigid*	Generate a mesh which can be used with the version of DYNA3D for rigid bodies.
sap4*	Generate a mesh which can be used with SAP4.
tz2d*	TOPAZ2D mesh input format.
tz3d	TOPAZ3D mesh input format. A FACET file is also generated if necessary.

* - Indicates that the analysis output option may be either untested or incomplete at this time.

Analysis Output Options (cont.)

DYNA3D Analysis Options

Command	Description
brode options ;	Define a Brode function with the following optional parameters. Note that if the rang , coef or gfun commands are specified, then a modified Brode function will be used in DYNA3D. For the modified Brode function, load curve # 1 is used.
<u>brode - Options:</u>	
cl l	Use <i>l</i> to convert from feet to DYNA3D length units.
cp p	Use <i>p</i> to convert from psi to DYNA3D pressure units.
ct t	Use <i>t</i> to convert from time in to DYNA3D time units.
height h	Height of burst.
tbo t	time origin of Brode function.
xbo x	x-coordinate of Brode origin in INGRID units.
ybo y	y-coordinate of Brode origin in INGRID units.
zbo z	z-coordinate of Brode origin in INGRID units.
yld σ	Yield (kilotons).
coef $c_1 \dots c_8$	Specify the coefficient values for a modified Brode function.
gfun $g_1 \dots g_7$	Specify the gfun values for a modified Brode function.
rang $r_1 \dots r_5$	Specify the rang values for a modified Brode function.
facrx x	Set the dynamic relation factor.
grav $g_x g_y g_z$	Specify the gravity acceleration vector. This option makes use of load curve number 1.
hydt	Activate the flag for dumping hydrodynamic variables for shells to the TAURUS database.
itrn n	Specify the number of iterations between convergence checks for dynamic relaxation.

Analysis Output Options (cont.)

DYNA3D Analysis Options (cont.)

Command	Description
itss Δt_o	Specify the initial time step size. This is an optional input for DYNA3D because if Δt_o is zero, DYNA3D will automatically pick the time step size.
lcmx n	Specify the load curve number that limits the maximum time step size.
nrest n	Specify the number of time steps, n , between dumps of the restart file.
plastic n	Set the plane stress plasticity option.
plti Δt	Specify the time interval for dumping nodal and element data to the graphics database.
prtflg n	Set the printout flag for the element time step size.
prti Δt	Specify the time interval for dumping nodal and element data to the output file for printing.
rgbm $m_1 m_2$;	Rigid body merge of materials m_1 and m_2 .
scftrx s	Set the scale factor for the dynamic relaxation time step.
sfor <i>options</i>	Specify the shell theory options to use.
<u>sfor - Options:</u>	
hugh	Specifies the Hughes-Liu shell theory.
bt	Specifies the Belytschko - Tsay Theory.
snrs n	Specify the Hughes-Liu shell normal update option, n .
ssdm	Activate flag for dumping shell strain tensors to the TAURUS database.
stat n	Specify the number of time steps between status reports
stss n	Set the shell step size option.
stup	Specify shell thickness change option.

Analysis Output Options (cont.)

DYNA3D Analysis Options (cont.)

<u>Command</u>	<u>Description</u>
teo <i>n</i>	Activate the thermal effects option using load curve <i>n</i> to initialize nodal temperatures. Here <i>n</i> specifies the load curve number. <i>n</i> = 0 no thermal effects are simulated. <i>n</i> = -1 a new temperature state is to be read from a disk file at each time step. The time word at the beginning of each temperature state is ignored. <i>n</i> = -2 a temperature state is interpolated from the temperature states in a disk file. The time words at the beginning of each temperature state is used in the calculation. <i>n</i> = -3 only one temperature state is contained in the disk file, and the initial temperature state is assumed to be zero.
term <i>t</i>	Terminate the simulation at time <i>t</i> .
tolrx ϵ	Set the convergence tolerance for dynamic relaxation.
tsmin <i>s</i>	Set the reduction factor, <i>s</i> , for the minimum time step size.
tssf <i>s</i>	Specify the time step size scale factor.

Analysis Output Options (cont.)

DYNA3D Analysis Options (cont.)

Command	Description
$\$n$ <i>comment</i>	Enter a comment which will be passed to the DYNA3D input file. Here, n be left blank, in which case the comment is output at the beginning of the input file, otherwise, n may take on one of the following values:
<u>n - Values:</u>	
mt m	Put the comment with material number m .
np	Nodal point comments.
he	Solid element comments.
be	Beam element comments.
se	Shell element comments.
ts	Thick shell element comments.
sp m	Put the comment with slide plane m .
nh	Nodal time history block comments.
hh	Solid element time history block comments.
bh	Beam element time history block comments.
sh	Shell element time history block comments.
th	Thick shell element time history block comments.
br	Brode function comments.
lc m	Put the comment with load curve m .
nl	Nodal load comments.
pr	Pressure load comments.
ve	Velocity comments.
sw m	Put the comment with stonewall m .
nc	Nodal constraint comments.
in	Initial condition comments.
si m	Put the comment with slide surface m .
ti	Tied node comments.
ex	Extra node comments for rigid bodies.
jt m	Put the comment with joint definition m .
xa	X-acceleration comments.
ya	Y-acceleration comments.
za	Z-acceleration comments.
xv	X-velocity comments.
yv	Y-velocity comments.
zv	Z-velocity comments.
dt m	Put the comment with detonator m .
sd	Spring-mass-damper comments.
nr	Non-reflecting boundary comments.

Analysis Output Options (cont.)

NIKE3D Analysis Options

Command	Description
anal <i>type</i>	Specify the type of analysis, where: <i>type = stat</i> specifies a static analysis. This is the default analysis <i>type</i> . <i>type = dyn</i> specifies a direct time integration. <i>type = dyns</i> specifies a direct time integration with static initialization. <i>type = eige</i> specifies Eigenvalue extraction.
bwmo <i>n</i>	If <i>n = on</i> , then bandwidth minimization will be performed in NIKE3D. If <i>n = off</i> , no bandwidth minimization will be performed. The default in NIKE3D is <i>n = on</i> .
dctol ϵ	Specify the displacement convergence tolerance. NIKE3D defaults to $\epsilon = 0.001$.
delt Δt	Specify the simulation time step size.
ectol ϵ	Specify the energy convergence tolerance. NIKE3D defaults to $\epsilon = 0.01$.
grav $g_x g_y g_z$	Specify the gravity acceleration vector. This option makes use of load curve number 1.
itss Δt_o	Specify the initial time step size. This is an optional input for NIKE3D because if Δt_o is zero, NIKE3D will automatically pick the time step size.
iplt Δt	Specify the time interval for dumping nodal and element data to the graphics database.
iprt Δt	Specify the time interval for dumping nodal and element data to the output file for printing.
msrf <i>n</i>	Specify the maximum number of stiffness reformations per time step. NIKE3D defaults to a recommended value of 15.
nbei <i>n</i>	Specifies that NIKE3D take <i>n</i> time steps between equilibrium iteration.
nbsr <i>n</i>	Specify that NIKE3D take <i>n</i> time steps between the reformation of the stiffness matrix.
neig <i>n</i>	Specify the extraction of <i>n</i> Eigenvectors.

Analysis Output Options (cont.)

NIKE3D Analysis Options (cont.)

Command	Description
nibsr <i>n</i>	Specify the maximum number of equilibrium iterations permitted between stiffness matrix reformations. NIKE3D defaults to a recommended value of 10.
nsmd <i>n</i>	<p><i>n</i> specifies the type of nonlinear solution algorithm employed in NIKE3D.</p> <p><i>n</i> = bfgs specifies the BFGS algorithm which is the default.</p> <p><i>n</i> = broy uses the Broyden algorithm</p> <p><i>n</i> = modn uses a modified Newton iteration.</p>
nstep <i>n</i>	Specify the desired number of time steps in the simulation.
sbrf <i>n</i>	Specify the number of time steps to be taken between the dumps to the restart file. If <i>n</i> is zero, then NIKE3D only writes a restart file as it terminates.
teo <i>n</i>	<p>Activate the thermal effects option using load curve <i>n</i> to initialize nodal temperatures. Here <i>n</i> specifies the load curve number.</p> <p><i>n</i> = 0 no thermal effects are simulated.</p> <p><i>n</i> = -1 a new temperature state is to be read from a disk file at each time step. The time word at the beginning of each temperature state is ignored.</p> <p><i>n</i> = -2 a temperature state is interpolated from the temperature states in a disk file. The time words at the beginning of each temperature state is used in the calculation.</p> <p><i>n</i> = -3 only one temperature state is contained in the disk file, and the initial temperature state is assumed to be zero.</p>
term <i>t</i>	Terminate the simulation at time <i>t</i> . The dynamic time step size will be computed if this command is used instead of the <i>delt</i> command.

Analysis Output Options (cont.)

TOPAZ3D Analysis Options

Command	Description
bwmo <i>n</i>	If <i>n</i> = on , then bandwidth minimization will be performed in TOPAZ3D. If <i>n</i> = off , no bandwidth minimization will be performed. The default in TOPAZ3D is <i>n</i> = on .
dcmx ΔT	Set the desired maximum temperature change in each time step.
dctol ϵ	Specify the convergence tolerance. TOPAZ3D defaults to $\epsilon = 0.0001$.
delt Δt	Specify the simulation time step size. For the variable time step integration, this specifies the initial time step used.
dtmax Δt	Set the maximum time step size.
dtmin Δt	Set the minimum time step size.
flux <i>n</i>	If <i>n</i> = on , then nodal heat flux computations are performed in TOPAZ3D. If <i>n</i> = off , no nodal heat fluxes are not computed. The default in TOPAZ3D is <i>n</i> = off .
iunit <i>n</i>	Specifies the temperature units. <i>n</i> may take on one of the following values: <i>n</i> = dime for dimensionless <i>n</i> = cent for Centigrade <i>n</i> = fahr for Fahrenheit <i>n</i> = kelv for Kelvin <i>n</i> = rank for Rankine
iplt <i>n</i>	Specify the number of time steps between dumps of nodal and element data to the graphics database.
iprt <i>n</i>	Specify the number of time steps between dumps of nodal and element data to the output file for printing.
linear	The conduction problem is linear.
mfts <i>t</i>	Modification factor for increasing or decreasing the time step.
mrdi <i>n</i>	Set the maximum number of radiosity iterations.
msrf <i>n</i>	Specify the maximum number of conductivity matrix reformations per time step. TOPAZ3D defaults to a recommended value of 10.

Analysis Output Options (cont.)

TOPAZ3D Analysis Options (cont.)

Command	Description
nbei <i>n</i>	Specifies that TOPAZ3D take <i>n</i> time steps between equilibrium iteration. The default is <i>n</i> = 1.
nbsr <i>n</i>	Specify that TOPAZ3D take <i>n</i> time steps between the reformation of the stiffness matrix. The default is <i>n</i> = 1.
nibsr <i>n</i>	Specify the maximum number of equilibrium iterations permitted between conductivity matrix reformations. TOPAZ3D defaults to a recommended value of <i>n</i> = 1.
nip1 <i>x</i>	Specify the first Newmark integration parameter as <i>x</i> . The default is <i>x</i> = 0.5.
nonlinear	The conduction problem is nonlinear.
nssd <i>n</i>	Specify the number of surface subdivisions for radiation view factor calculations. The default is <i>n</i> = 5.
phase <i>n</i>	If <i>n</i> = on , then phase change computations are performed in TOPAZ3D. If <i>n</i> = off , no phase change calculations are mad. The default in TOPAZ3D is <i>n</i> = off .
radiation <i>type</i>	Specifies the type of radiation calculations to be made. <i>type</i> = view specifies the use of view factors, while <i>type</i> = exch specifies that exchange fractions should be used.
rband <i>m n</i> $\lambda_1 \lambda_2 \dots \lambda_m$ $\epsilon_1 \epsilon_2 \dots \epsilon_m$	Set the radiation bands to account for <i>m</i> wavelength break points with <i>n</i> curves. The wavelength break points. n families of emissivity curves.
rtol ϵ	Specify the radiosity convergence tolerance to ϵ .
relax ω	Specify the relaxation parameter ω . The default is ω = 1.
sbc σ	Set the Stefan-Boltzmann constant.
sbrf <i>n</i>	Specify the number of time steps to be taken between the dumps to the restart file.
steady	Specifies a steady state computation.
step <i>type</i>	Specifies the type of time integration to use. If <i>type</i> = fixe , then a fixed time step is used. If <i>type</i> = vari , then variable time step sizes are used.

Analysis Output Options (cont.)**TOPAZ3D Analysis Options (cont.)**

<u>Command</u>	<u>Description</u>
term t	Terminate the simulation at time t .
timin t	Set the initial analysis time to t .
transient	Specifies a transient computation.

Analysis Output Options (cont.)

MONT3D Analysis Options

Command	Description
delt $\Delta\theta$	Set the cone angle increment for material properties (default: $\Delta\theta = 0.01$).
ibug n	Setting $n = 1$ activates debugging during the input phase, $n = -1$ activates debugging for the entire code, while $n = 0$ deactivates debugging (default: $n = 0$).
idata n	Setting $n = 1$ forces data checking only to be executed (default $n = 0$).
iprnt n	Setting $n = 0$ forces the exchange factor matrix to printed in the output file, while $n = 1$ suppresses printing (default: $n = 0$).
nbands n	Specify the number of radiation wavelength bands (default: $n = 1$).
ndivx n_x	Set the number of emission points in the local x-direction on each enclosure surface (default: $n_x = 10$).
ndivy n_y	Set the number of emission points in the local y-direction on each enclosure surface (default: $n_y = 10$).
nlost n	Set the maximum number of lost photons (default: $n = 100$).
nphton n	Specify the number of photons emitted per surface in the radiation enclosure (default: $n = 10$).
nrefs n	Specify the maximum number of reflections per photon before a warning message in MONT3D is issued (default: $n = 50$).
nshade n	Specify the type of shading algorithm to use. $n = -1$ activates the Margolies grid shading algorithm, $n = 0$ specifies that there are no shaded surfaces in the enclosure, $n = 1$ activates the distance shading algorithm (default: $n = 1$).
ngx n_x	Set the number of grid spaces in the x-direction for the Margolies Grid shading algorithm.
ngy n_y	Set the number of grid spaces in the y-direction for the Margolies Grid shading algorithm.
ngz n_z	Set the number of grid spaces in the z-direction for the Margolies Grid shading algorithm.
numcat n	Specify the number of concatenated surfaces in the radiation enclosure (default: $n = 0$).

Analysis Output Options (cont.)

MONT3D Analysis Options (cont.)

<u>Command</u>	<u>Description</u>
xscale <i>s</i>	Set the scale factor for the x-dimension (default: $s = 1.0$).
xshift Δx	Specify the amount to shift the enclosure geometry in the x-direction (default: $\Delta x = 0.0$).
yscale <i>s</i>	Set the scale factor for the y-dimension (default: $s = 1.0$).
yshift Δy	Specify the amount to shift the enclosure geometry in the y-direction (default: $\Delta y = 0.0$).
zscale <i>s</i>	Set the scale factor for the z-dimension (default: $s = 1.0$).
zshift Δz	Specify the amount to shift the enclosure geometry in the z-direction (default: $\Delta z = 0.0$).

4.2.2 Fourier Series Definitions

Fourier series may be defined for use in line definitions to simulate imperfections in geometry. Multiple series may be defined each with its own set of unique coefficients using either a deterministic series or a series with random phase shifts.

Fourier Series Definitions

Command	Description
fsd <i>n m a₀ a₁ φ₁ ... a_m φ_m</i>	Define Fourier series <i>n</i> . The number of terms in the series is <i>m</i> . The coefficients are used in computing the series as:

$$f(\theta) = a_0 + \sum_{i=1}^m a_i \cos(i\theta + \phi_i)$$

rfsd <i>n a x m</i>	Define Fourier series <i>n</i> with random phase shifts.
----------------------------	--

$$f(\theta) = \sum_{i=2}^m \frac{a}{(i^x)} \cos(i\theta + \phi)$$

where ϕ is a random phase shift uniformly distributed between 0 and 2π .

4.2.3 General Control Commands

This section describes the commands which are used to control graphical and interactive features of INGRID. Some features are site specific for LLNL, and are identified as such.

General Control Commands

Command	Description
box <i>n</i>	Batch command for specifying the Octopus box number. A box number specified in the "user.ing" file will always supersede the number specified in the input file. LLNL site specific.
noplot	Do not perform any of the interactive graphics operations. Because the graphics system is never initialized, this command forces INGRID to operate in strictly a batch mode.
parameter <i>name value ;</i>	<p>The parameter command assigns real numbers to named variables. The values are used in the INGRID input file everywhere the macro name occurs as: <i>%name</i>. It is impossible to define "c" as a parameter name because "c" is predefined as the character delimiter for comments. Any parameter names which are used before they are defined take on a value of 0 by default, however, it is recommended that parameter values always be defined before they are used. Parameters are always associated with their most recent definition. Currently, INGRID can handle a maximum of 300 parameters, and may be used with the in-line calculator (Section 5.1.3) When arithmetic operations are to be performed, the parameters may be enclosed in square brackets as: <i>[-%name]</i>.</p> <p>As an example, a parameter <i>xmin</i> can be defined and used as:</p> <pre>C xmin and xmax are parameters param xmin 5.0; param xmax 10.0; start 1 2; 1 2; 1 ; %xmin %xmax %ymin %ymax %zmin %zmax end</pre>
pause <i>n</i>	Causes INGRID to pause for approximately <i>n</i> seconds.
tmds <i>n</i>	Specify the TMDS number to be used (see also tv). LLNL site specific.

4.2.4 Geometric Transformations

This section details the INGRID commands used to perform geometric transformations. Geometric transformations include operations such as scaling nodal point coordinates, performing conversions between coordinates systems, and permuting coordinate axes.

Geometric Transformations

Command	Description
cscs <i>s</i>	Scale all nodal coordinates by <i>s</i> .
cyli	Nodal point coordinates are converted from cylindrical to Cartesian coordinates as: $x = r \cos(\theta)$, $y = r \sin(\theta)$.
exch $i_x i_y i_z$	This command performs a permutation of the coordinate axes. This operation is performed <i>before</i> any other coordinate transformations. i_x , i_y and i_z are flags which determine what happens to the generated x,y and z coordinates. If a flag takes on a value of 1, then it becomes the x-coordinate. If a flag takes on a value of 2, then the coordinate is mapped to the y-coordinate. If a flag takes on a value of 3, then the coordinate is mapped to the z-coordinate.
gct $n ; t_1 ; t_2 ; \dots ; t_n ;$	Define a sequence of n global coordinate transformations $t_1 \dots t_n$. Acceptable transformations used to define $t_1 \dots t_n$ are listed below, and are required. This command must precede grep for global part replication. gct and grep are intended to be used as companion commands for performing coordinate transformations and part replications. gct is also used with the pplv and pslv commands
<u>gct - Required data for transformations:</u>	
cscs <i>s</i>	Scale all nodal coordinates by <i>s</i> .
mx Δx	Translate by Δx .
my Δy	Translate by Δy .
mz Δz	Translate by Δz .
v $\Delta x \Delta y \Delta z$	Translate by Δx , Δy , Δz .
repe <i>m</i>	Repeat the global coordinate transformation <i>m</i> times.
rx θ_x	Rotate by θ_x^0 about the x-axis.
ry θ_y	Rotate by θ_y^0 about the y-axis.
rz θ_z	Rotate by θ_z^0 about the z-axis.
rx <i>y</i>	Reflect the part about the x-y plane.
ry <i>z</i>	Reflect the part about the y-x plane.
rz <i>x</i>	Reflect the part about the z-x plane.
save <i>m</i>	The sequence of coordinate transformations is generated starting from coordinate system <i>m</i> when using the repe command.
xcscs <i>s</i>	Scale all nodal x-coordinates by <i>s</i> .
ycscs <i>s</i>	Scale all nodal y-coordinates by <i>s</i> .
zcscs <i>s</i>	Scale all nodal z-coordinates by <i>s</i> .

Geometric Transformations (cont.)

Command	Description
grep <i>0 1 ... n ;</i>	The grep command performs global part replication, and is intended to be used with the gct command. Here, the <i>0</i> indicates that the original instance of the part should be kept in the finite element mode. The sequence of <i>1</i> to <i>n</i> applies the gct coordinate transformations to the part keeping each instance of the transformed part. For example, grep 0 1 2; would keep the original part, and apply the global transformations 1 and 2 from the gct command: gct 2; mx Δx; my Δy;
gexc <i>i_x i_y i_z</i>	Perform a permutation of the coordinate axes <i>after</i> any other coordinate transformations. <i>i_x</i> , <i>i_y</i> , <i>i_z</i> are flags which determine what happens to the generated <i>x</i> , <i>y</i> , <i>z</i> coordinate. If a flag is 1, then it becomes the <i>x</i> -coordinate. If it is 2, then the coordinate is mapped to the <i>y</i> -coordinate. If it is 3, then the coordinate is mapped to the <i>z</i> -coordinate.
gmi <i>n</i>	Increment the global material number by <i>n</i> .
lev <i>n options ;</i>	Define part transformation sequence number <i>n</i> . This defines a series of operations which can be performed on groups of parts. The following are valid options for the lev command.
<u>lev - Options:</u>	
add <i>m</i>	Add the list of transformations in sequence number <i>m</i> to the current sequence
coor <i>n data</i>	Add <i>n</i> coordinate transformations to the current sequence. Here, <i>data</i> is indicates the possible optional coordinate transformation options.
cyli	Perform a cylindrical coordinate transformation.
prod <i>i j</i>	Form the product of sequence <i>i</i> with sequence <i>j</i> . If sequence <i>i</i> has <i>l</i> transformations and <i>j</i> has <i>m</i> transformations, then this option produces <i>l*m</i> transformations and adds them to the current sequence.
repe <i>l₁ l₂ ... l_n</i>	Copy parts in the global coordinate systems <i>l₁</i> , <i>l₂</i> , ... <i>l_n</i> .
sphe	Perform a spherical coordinate transformation.
loof	Plate elements do not lie in a plane (they lie out-of plane). If this command is not used, mid-side nodes will be moved to the plane of the plate or shell element.

Geometric Transformations (cont.)

<u>Command</u>	<u>Description</u>
pslv <i>n</i>	Begin performing part transformation sequence <i>n</i> on all the following parts. This command remains active until a pplv command is given. pslv adds a transformation sequence and pplv eliminates the first sequence.
pplv	Terminate the part transformation operations. This command eliminates the top part transformation from the list of transformations.
sphe	Nodal point coordinates are converted from spherical to Cartesian coordinates as: $x = r \cos(\theta) \sin(\psi)$, $y = r \sin(\theta) \sin(\psi)$, $z = r \cos(\psi)$.
xoff Δx	Global x-coordinate offset by Δx .
yoff Δy	Global y-coordinate offset by Δy .
zoff Δz	Global z-coordinate offset by Δz .
xzca <i>s</i>	Scale all x-coordinates by <i>s</i> .
ysca <i>s</i>	Scale all y-coordinates by <i>s</i> .
zsca <i>s</i>	Scale all z-coordinates by <i>s</i> .

4.2.5 Geometrical Part Control

This section describes the commands used to control parts which are constructed using joints and spring - mass - damper systems. Joint definitions may be used to construct meshes using pre-defined joint geometries. Figure 4.1 below shows the types of joints currently supported in INGRID. Spring - mass - damper systems are intended to be used with the **spdp** command in the standard part options.

Beam Section Definitions

Command	Description
bsd <i>n options</i> ;	Define beam section <i>n</i> with the following options. Please refer to Figure 2.1.
<u>bsd - Options:</u>	
sloc <i>i</i>	Set the beam reference surface location in the s-direction. <i>i</i> may take on the following values: <i>i</i> = -1 : Sets the reference to s = -1. <i>i</i> = 0 : Sets the reference to s = 0. <i>i</i> = +1 : Sets the reference to s = +1.
sthi <i>t</i>	Set the beam thickness in the s-direction.
sthi1 <i>t</i>	Set the beam thickness in the s-direction at node #1.
sthi2 <i>t</i>	Set the beam thickness in the s-direction at node #2.
tlloc <i>i</i>	Set the beam reference surface location in the t-direction. <i>i</i> may take on the following values: <i>i</i> = -1 : Sets the reference to t = -1. <i>i</i> = 0 : Sets the reference to t = 0. <i>i</i> = +1 : Sets the reference to t = +1.
tthi <i>t</i>	Set the beam thickness in the t-direction.
tthi1 <i>t</i>	Set the beam thickness in the t-direction at node #1.
tthi2 <i>t</i>	Set the beam thickness in the t-direction at node #2.

Geometrical Part Control (cont.)

Joint Definitions

Command	Description
jd <i>n options ;</i>	Begin the definition of joint <i>n</i> . Diagrams of the types of permissible joints are shown in Figure 4.1. Nodes are assigned to joint definitions within parts.
<u>jd - Options:</u>	
sj	Specify a spherical joint. (Figure 4.1)
rj	Specify a revolute joint. (Figure 4.1)
cj	Specify a cylindrical joint. (Figure 4.1)
pj	Specify a planar joint (Figure 4.1)
uj	Specify a universal joint (Figure 4.1)
tj	Specify a translational joint (Figure 4.1)
pnlt λ	Specify a joint penalty λ .
nc <i>i</i>	Specify the nodal constraint type. <i>i</i> may have values ranging from 1 to 7. The common degrees of freedom associated with <i>i</i> are: <i>i</i> = 1: x <i>i</i> = 2: y <i>i</i> = 3: z <i>i</i> = 4: x and y <i>i</i> = 5: y and z <i>i</i> = 6: x and z <i>i</i> = 7: x, y and z
repe <i>m</i>	Repeat the current joint definition <i>m</i> times for a total of <i>m</i> joints.

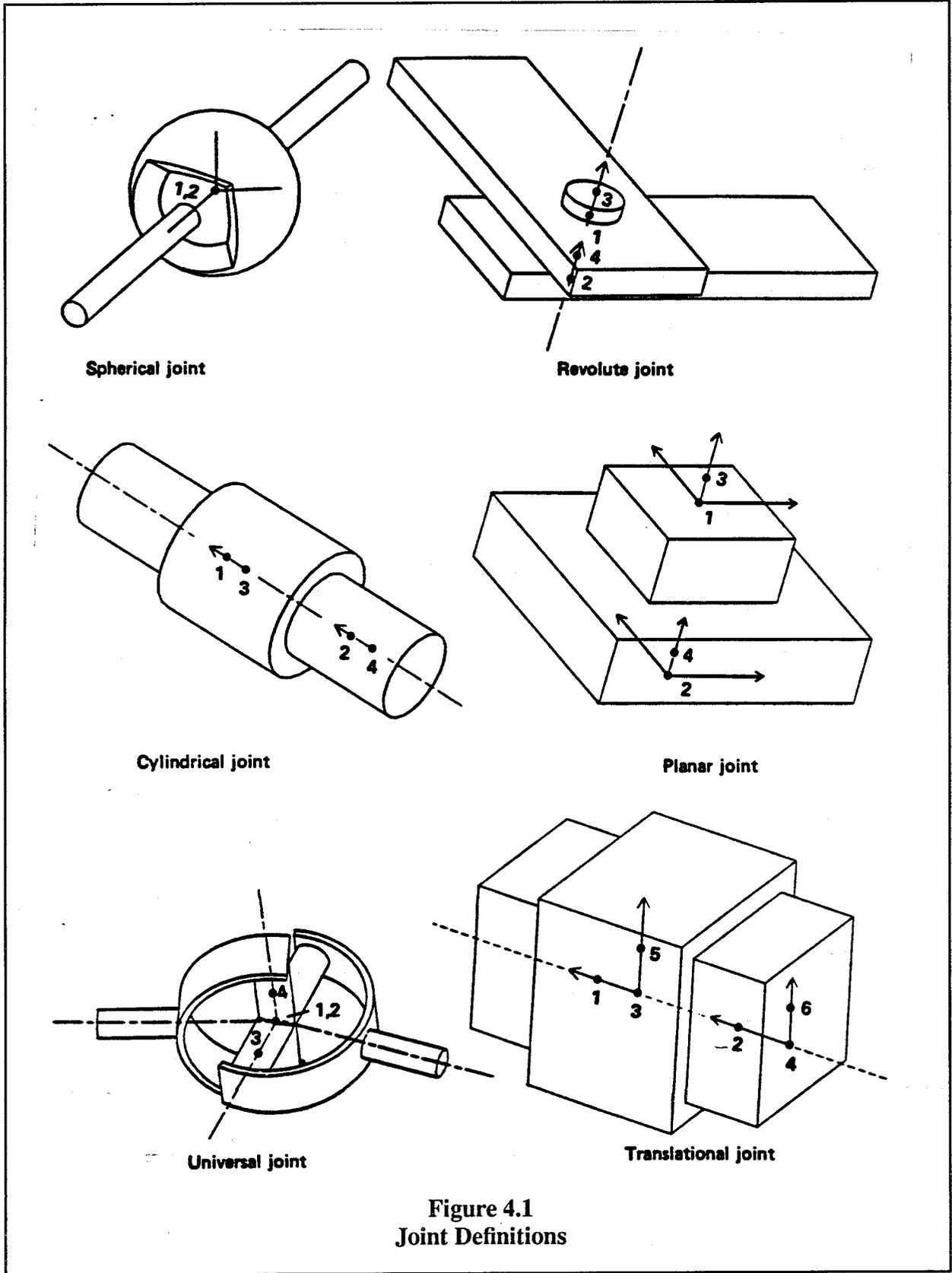


Figure 4.1
Joint Definitions

Geometrical Part Control (cont.)

Spring - Mass - Damper Systems

Command	Description
spd <i>n options</i> ;	Identify spring, mass, damper combination <i>n</i> for use with the spdp part command.
<u>spd - Options:</u>	
le <i>k</i>	Specify linear-elastic spring with stiffness <i>k</i> .
lv ξ	Specify linear viscous damping, damping coefficient ξ .
iep <i>E F_t F_y</i>	Specify isotropic elastoplastic where <i>E</i> is the elastic stiffness (force per unit displacement), <i>F_t</i> is the tangent stiffness (force per unit displacement), <i>F_y</i> is the yield force.
ne <i>n</i>	Specifies a nonlinear elastic spring using load curve <i>n</i> .
nv <i>n</i>	Specifies a nonlinear viscous damping where <i>n</i> is a load curve.
gn <i>n₁ n₂ b F_t F_c</i>	Specify a general nonlinear spring, where <i>n₁</i> the nonlinear elastic load curve, <i>n₂</i> is the nonlinear viscous damping curve, <i>b</i> is a hardening parameter, <i>F_t</i> is the tensile yield force, <i>F_c</i> is the compression yield force.

4.2.6 Global Initial & Boundary Conditions

The global control commands for the specification of both initial conditions and boundary conditions are described in this section.

Global Initial & Boundary Conditions

Command	Description
arri n options;	Generate arrival times for pressure surfaces associated with load curve n. Arrival times are generated by assuming that the loads are caused by a wave. This wave starts from a 3-D surface and travels with a given velocity. The arrival time is the time required for the wave to travel from the source to an individual pressure segment (a mesh surface element).
<u>arri - Options:</u>	
velo v	The wave travels with a speed v.
toff Δt	Add Δt to the arrival time.
point p_x p_y p_z	The source for the pressure wave is a point located at: (p _x , p _y , p _z).
line p_x p_y p_z v_x v_y v_z	The source is a line. p _x , p _y , p _z is a point on the plane, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector along the line.
plane p_x p_y p_z v_x v_y v_z	The source is a plane. (p _x , p _y , p _z) is a point in the plane, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector normal to the plane.
cosine	The pressure varies as a function of the angle between the pressure segments (mesh surface elements) normal and the direction of the wave from the source.
cg P_{max}	The pressures cannot exceed P _{max} .
cl P_{min}	The pressures cannot be less than P _{min} . (This option is selected with P=0.0 when the cosine option is executed.)
decay d type	The pressure wave decays as a function of the distance from the source. The distance at which the scale factor for the input pressure equals 1.0 is d. The type of decay may be r, r2, r3 or constant.
<u>decay - Type:</u>	
r	The pressure decays via a 1/r relationship.
r2	The pressure decays via a 1/r ² relationship.
r3	The pressure decays via a 1/r ³ relationship.
constant	The pressure is constant - no decay.

Global Initial & Boundary Conditions (cont.)

Plane Definitions

plane n data

Input n plane definitions. These planes are used for implementing boundary condition specifications. A point which lies in the plane, normal vector, tolerance and type are required for each plane definition. This data must immediately follow the plane command, and there must be n occurrences of this data (one for each plane).

Required Data:

p_x p_y p_z
 n_x n_y n_z
options

Global coordinates of any point on the plane.

Any vector normal to the plane.

options can be:

symm - for a symmetry plane,

asym - for an asymmetry plane,

ston - for a stonewall

stick i ston - for a sticking stonewall with stonewall option i .

dicv n ston - for a stonewall with velocity load curve number n .

penalty λ ston - for a stonewall with penalty factor λ .

move m v i ston - for a moving stonewall with mass m , velocity v , and option i .

limit x_h y_h z_h x_l y_l i ston - for a stonewall with head (x_h, y_h, z_h) , length x_l and y_l , and option i .

Rigid Body Motion

Command	Description
rotation p_x p_y p_z ω_x ω_y ω_z	Assign an initial rigid body rotation to all parts defined after this command. (p_x, p_y, p_z) is any point on the axis of rotation and $(\omega_x, \omega_y, \omega_z)$ is the angular velocity vector in radians per second.
velocity v_x v_y v_z	Assign the rigid body velocity $\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ to all parts defined after this command.

Thermal Conditions

Command	Description
temp T	All parts defined after this command in the mesh have an initial temperature of T .

4.2.7 Load Curve Definitions

Load curves are global INGRID entities which serve a multitude of uses. In general, load curves may be used to describe a function which varies with time for use in the description of boundary conditions. Force load curves are simply load curves specialized for the purpose of handling force boundary conditions. Where load curve numbers are referenced in the description of a command, it is assumed that a global load curve for that quantity has been defined in the INGRID input file.

Load Curve Definitions

Command	Description
flcd <i>n options ;</i>	Force load curve <i>n</i> is defined by one of the following options.
<u>flcd - Options:</u>	
lp <i>m t₁f₁ ... t_mf_m</i>	Specify <i>m</i> time-force data points for load curve <i>n</i> .
sine <i>m cps t</i>	Specify a sine wave consisting of <i>m</i> data points. <i>cps</i> defines the wave frequency in cycles per second, and the curve continues for a period of time <i>t</i> . The sinusoid begins at the end of the last time value defined.
cosine <i>m cps t</i>	Specify a cosine wave consisting of <i>m</i> data points. <i>cps</i> defines the wave frequency in cycles per second, and the curve continues for a period of time <i>t</i> . The sinusoid begins at the end of the last time value defined.
phase θ	Use phase angle θ for the cosine or sine wave. θ is in degrees.
scale <i>s</i>	Scale the load curve values by <i>s</i> .
offset Δf	Offset the load curve values by Δf . The combination of scaling plus offset is computed as: $f' = sf + \Delta f$.
to <i>f₀</i>	Reset the initial amplitude at time zero to <i>f₀</i> .
tsca <i>s</i>	Scale the time values by <i>s</i> .
toff Δt	Offset the time values by Δt . This is computed as: $t' = t + \Delta t$.
lcd <i>n m $\theta_1 f_1$... $\theta_m f_m$</i>	Define load curve, <i>n</i> , with <i>m</i> pairs of time - function points.
prcd <i>n m $\theta_1 p_1$... $\theta_m p_m$</i>	Define variable pressure curve, <i>n</i> , with <i>m</i> pairs of pressure - θ function points. This curve is a piecewise linear curve where θ is the independent variable. In this curve, $\theta_i < \theta_{i-1}$, and <i>p</i> is assumed to be constant at <i>p₁</i> for all $\theta < \theta_1$, and at <i>p_m</i> all $\theta > \theta_m$. This command is used also to define the thickness variation for thin shells using the thth command.

Load Curve Definitions (cont.)

Command	Description
thcd $n m \theta_1 t_1 \dots \theta_m t_m$	Define variable thickness curve, n , with m pairs of thickness - θ function points. This curve is a piecewise linear curve where θ is the independent variable. In this curve, $\theta_i < \theta_{i-1}$, and t is assumed to be constant at t_1 for all $\theta < \theta_1$, and at t_m all $\theta > \theta_m$. This command is also used to define the thickness variation for thin shells using the thth command.
vecd $n m \theta_1 v_1 \dots \theta_m v_m$	Define variable velocity curve, n , with m pairs of velocity - θ function points. This curve is a piecewise linear curve where θ is the independent variable. In this curve, $\theta_i < \theta_{i-1}$, and v is assumed to be constant at v_1 for all $\theta < \theta_1$, and at v_m all $\theta > \theta_m$. This command is used to define the velocity variation for thin shells using the veth command.

4.2.8 Line Definitions

Line definitions are global INGRID entities which are used in the description of part geometry. Line definitions are used both with MAZE parts and with the surface projection algorithms in INGRID. This section details the commands available to define lines within INGRID.

Line Definitions

Command	Description
edgefile <i>file</i>	Specify that line definitions are to be read from <i>file</i> . The <i>file</i> must be a local file, existing in the same directory as the INGRID input file. The file name, <i>file</i> , is limited to a total of eight characters.
lcc $n r_c z_c \theta_1 \theta_2 r_1 \dots r_n$	Define n lines consisting of circular arcs centered at the point (r_c, z_c) that sweep from the angle θ_1 to the angle θ_2 . $r_1 \dots r_n$ are the radii of the n lines. With this command, line numbers are automatically assigned to avoid conflicts with previously defined lines.
lrl $n r_c z_c R \theta_1 \dots \theta_2$	Define n lines consisting of radial lines of length R originating at the point (r_c, z_c) and oriented at angles $\theta_1 \dots \theta_2$. Positive angles are measured counter-clockwise from the r -axis.

Line Definitions (cont.)

Command	Description
ld n options	Line definition number n. If line n has been previously defined, this command has the effect of destroying the old definition. <i>options</i> consists of one of the following sets of line definition options.
<u>ld - Options:</u>	
ladv n Δr Δz	Add the vector (Δr,Δz) to line n.
lap r₁ z₁ r_c z_c	Define a circular arc by specifying points. The arc is assumed to begin at the last point defined, and to end at (r ₁ ,z ₁). The point (r _c ,z _c) lies at the center of arc.
lar r₁ z₁ R	Define a circular arc by specifying the radius. An arc of radius R is assumed to begin at the last point defined, and to end at (r ₁ ,z ₁). If R is positive, then the arc is swept out in the r-z plane in a counterclockwise sense. If R is negative, then the arc is swept out in a clockwise sense.
lat r₁ z₁ r₂ z₂	Define a circular arc of radius R tangent to the last line segment and a line segment connecting the point (r ₁ ,z ₁) to (r ₂ ,z ₂). These line segments will be automatically extended or truncated to the point of tangency.
lad r_c z_c θ	Define a circular arc centered at the point (r _c ,z _c) beginning at the last point defined and sweeping through θ degrees. Positive values of θ indicate a counterclockwise sweep, while negative θ indicates a clockwise sweep.
lcfi m r₀ θ_i θ_f n	The current line definition is a cylinder with imperfections defined by Fourier series m. The equation for the cylinder is: $r = r_0 + \sum_{i=1}^n a_i \cos(i\theta + b_i)$ <p>Where n, a_i, b_i are defined in Fourier series m (see Fourier Series Definitions above). r₀ is the nominal cylinder radius, and θ_i, θ_f are the initial and final angles respectively over which the cylinder is defined. n digitized points will be used.</p>
lep a b r_c z_c θ₁ θ₂ φ	Define an elliptical arc by the semi-major and semi-minor axes, a and b respectively, the center point (r _c ,z _c), and a sweep from θ ₁ to θ ₂ as measured from the major axis. φ is the angle between the major axis and the r-axis. A circular arc is generated by setting a = b. Positive angles represent counterclockwise rotations.

Line Definitions (cont.)

Command	Description
<u>ld - Options (cont.):</u>	
lp $m r_1 z_1 \dots r_m z_m$	Define m points to be used in line definition n .
lint $l_1 l_2 s$	The current line definition is formed by interpolation between lines l_1 and l_2 . The interpolation equation is: $l = s l_1 + (l_0 - s) l_2$
lo $m r_1 z_1 r_2 z_2$	Define a line segment for line n by offsetting the line m such that the new line segment starts at (r_1, z_1) and ends at (r_2, z_2) .
lod $m \delta$	Define a line segment for line n by applying an offset of δ to the entire line definition m . When oriented in the original direction of line m , a positive value of δ causes an offset to the left in a relative sense. Similarly, a negative δ causes a relative offset to the right.
lpil $n_1 n_2$	Specify that INGRID is to compute a point of intersection between lines n_1 and n_2 to be used in line n .
lpt $r_1 z_1 r_2 z_2 R$	Define a circular arc of radius R which begins at the last point defined, and is tangent to a line segment joining the point (r_1, z_1) and (r_2, z_2) . This line segment will be extended or truncated to the point of tangency.
lpta $r_c z_c R$	Define a line segment beginning at the last point defined, and terminating at its tangency point on an arc of radius R centered at (r_c, z_c) . The first tangency point encountered as the arc is generated by a counterclockwise rotation from the r-axis will become the end point. If R is a negative number, a clockwise rotation from the r-axis will determine the first tangency point.
lrot $n \theta$	Rotate line n about the r-z origin θ degrees.
lsca $n S$	Scale line n by S .
lscr $n S$	Scale the r-coordinates of line n by S .
lscz $n S$	Scale the z-coordinates of line n by S .
lstl $m \Delta r \Delta z$	Define a line segment for line n by translating line m by an amount of $(\Delta r, \Delta z)$.
lt $n \Delta r \Delta z$	Translate line n by $(\Delta r, \Delta z)$.
ltbo $m_1 d_1 \dots m_m d_m$	Define a line segment by offsetting the last line defined with ltbc or ltbo . The radii of the first m_1 points are offset by d_1 , the following m_2 points by d_2 , etc. Note that $m = m_1 + m_2 + \dots + m_m$ where m is the number of points specified on the ltbc command.

Line Definitions (cont.)

Command	Description
<u>ld - Options (cont.):</u>	
ltas $r_1 z_1 \omega r_2 z_2 R$	Define a line segment tangent to a circular arc centered at (r_1, z_1) beginning at the last point defined and sweeping counterclockwise if $\omega = 1$, and clockwise if $\omega = -1$. The line segment terminates at its tangency point on a second arc of radius R centered at (r_2, z_2) . The first tangency point encountered as the second arc is generated by a counterclockwise rotation from the r-axis will become the end point. If R is a negative number, a clockwise rotation from the r-axis will determine the first tangency point.
ltbc $m \theta \Delta\theta S R_1 \dots R_2$	Define a line segment with tab cell data. Tab cell data is often used in drafting programs, and consists of m radii (R_i) each $\Delta\theta$ degrees apart starting at an angle θ . Each radius is scaled by S . Positive angles indicate counterclockwise rotations.
ltp $r z R$	Define a circular arc of radius R which is tangent to the last line segment and terminates at the point (r, z) . The last line segment will be automatically extended or truncated to the point of tangency.
lvc $r z \theta M$	Define a line segment in a vector direction by specifying a magnitude M , with orientation θ (θ is measured counterclockwise from the r-axis). If this is the first command in the definition of a new line, then the origin (r, z) must be specified. A negative value for M indicates that the second point is being defined, and the vector points towards the first point which has already been defined.
rlns	Read line definitions from the edge file (see edgefile above). INGRID assigns unique line numbers sequentially.
rln	Read a single line definitions from the edge file and assigns a unique line number.
rseg	Read the next line segment definition from the edge file.

4.2.9 Material Specification

Considerable effort has been invested in making INGRID capable of modelling complex geometry. That effort is reflected in the ease by which analysts have been able to generate complex meshes. While generating complex geometry is a principal focus of INGRID, defining material properties for various material models is equally important for generating complete descriptions of finite element models.

Entering material information within INGRID is accomplished through material model definitions. Each material model is bounded by two keywords which signify the beginning and end of the definition - **mat** and **endmat**. Input between these two keywords consists of keyword-parameter pairs in which control settings for the specific material model are defined. Material commands which influence the behavior of models in a global context may be entered in much the same way that control commands are entered for parts.

Global Material Control Commands

There are two distinct phases in defining materials in INGRID. First, each material must be defined by a material model. Each material model is further categorized by the definition of a material type (e.g., elastic, elastic-plastic, viscoelastic, etc.). The material type dictates the information required to completely define the constitutive law for the material model. Second, each generated part to be used in the analysis must be linked to a global material definition using the **mate** command either as a global control command, or in the part definition. This section details the global INGRID material commands, and the analysis specific material definitions currently available in INGRID.

In addition to using the material definition to specify the material behavior, there are also some general control commands which provide additional information needed by the analysis codes for certain models to behave correctly. The control commands affect such things as gravity loading, beam and shell formulation options, establishing density, controlling material mass, and indexing material numbers. Commands are also available to influence the generation and behavior of shell materials, establish local t-axis orientations, and globally set density and viscosity values.

Whenever part duplication commands are used, both within a part definition (local) or outside a part definition (global), a default material model is used for the duplicated section. It is possible to change the default material number for each of those copies using the **lmi** or **gmi** commands. The **lmi** command increments the default material model number for each local copy of a part whereas the **gmi** command increments that number for global copies.

Typically, material density is input into INGRID and the total mass of a specific material is determined by the sum of the volume occupied by each of the elements made up of that material. Entering that density is done through the **rho** command. It is occasionally necessary to do the converse - specify the total mass and let the density be determined by the volume. The **tmm** command is used for such a situation. When using **tmm**, the total mass of a specific material is input and the density is determined from the volume of the part.

Since INGRID is capable of generating meshes which can be later used for electromagnetic studies, it was necessary for INGRID to be capable of establishing material interface conditions between two materials. The **mbc** command is used to control those conditions. Three remaining materials commands apply specifically to DYNA3D material options. The **gravity** command is used to flag material initialization for gravity loading. The **elfm** command is used to indicate the beam or shell formulation option. Lastly, the **shear** command may be used to input the necessary shear factor within the material definition.

Shell Materials

The generation of shell elements is largely a function of part definitions. While shells are usually generated using index progressions and global control commands, it is sometimes necessary to make a distinction between the generation of thin and thick shell elements. There are two general material commands which assert that distinction. Whenever the **shell** command is used in a material model definition, all brick elements made up of that particular material model are, instead, constructed of thin shell elements. Likewise, whenever the **tshell** command is used, all brick elements made up of a particular material model are constructed of thick shell elements. The default thickness for all thin shell elements is 0.0.

Establishing Local T-axis Orientation

Some material models require a specific orientation of the local t-axis (e.g. composite materials). The behavior of the model is influenced by that orientation. In order to establish one of two configurations, two commands are available in INGRID. If it is necessary to set the orientation of the local t-axis so that it follows the shortest direction through each brick element, the **ltmin** command should be used. If the orientation should be along the longest direction of each brick element, the **ltmax** command should be entered instead.

Modifying Density and Viscosity

Six commands are available within INGRID which alter the general material model characteristics of density and viscosity. The **ro** command may be used within a material model definition to specify the material model density. Material density input is required for all DYNA3D material

models. The bulk viscosity type may be set using the **bqt** command. Two bulk viscosity coefficients - the linear and quadratic coefficient may be set using the **bql** and **bqq** commands, respectively. The hourglass coefficient type can be established using the **hgqt** command and the corresponding hourglass viscosity coefficient can be set with **hgq**.

Global Material Commands

Material Detonation Points

Command	Description
detp <i>n options</i> ;	Define detonation points for material <i>n</i> . If <i>n</i> is zero, then all materials are detonated.
<u>detp - Options:</u>	
time <i>t</i>	Lighting time for detonation point.
point $p_x p_y p_z$	Detonate the point (p_x, p_y, p_z) .
lnpt $p_x p_y p_z n_x n_y n_z n$	Generate <i>n</i> equally spaced detonation points on the line from (p_x, p_y, p_z) to (q_x, q_y, q_z) .

Global Material Specification

Command	Description
mate <i>m</i>	The default material number is set to <i>m</i> for all of the parts following this command.
mbc <i>m n</i>	Generate electromagnetic material interface conditions between materials <i>m</i> and <i>n</i> .
rho <i>n</i> ρ	Specify that material <i>n</i> have density ρ .
thic <i>t</i>	Shell elements have the a thickness of <i>t</i> .
tmm <i>n t</i>	The total mass of material <i>n</i> is <i>t</i> . The density of the material is determined by $\rho = t/V$ where <i>V</i> is the total volume.

Global Material Commands (cont.)Global Material Specification (cont.)

Command	Description
mat <i>n m options endmat</i>	Define material model <i>n</i> to type <i>m</i> .
<u>mat - Options:</u>	
<i>General material commands...</i>	
head	The next line within the input file will contain a title for the material currently being specified.
lmi <i>m</i>	Forces the default material model number to increment by <i>m</i> for each local copy of a part.
gmi <i>m</i>	Forces the default material model number to be increment by <i>m</i> for each global copy of a part.
tmm <i>n t</i>	The total mass of material <i>n</i> is <i>t</i> .
mbc <i>m n</i>	Generate electromagnetic material interface conditions between materials <i>m</i> and <i>n</i> .
shear τ	Establishes the shear factor as τ .
rho ρ	Sets the material density to ρ .
gravity	Flags the material initialization for gravity loading.
elfm <i>m</i>	Sets the beam or shell formulation option to <i>m</i> .
<i>Beam materials ...</i>	
beam	Forces all elements made with using current material to be constructed of thin shells.
sloc <i>i</i>	Set the beam reference surface location in the s-direction. <i>i</i> may take on the following values: <i>i</i> = -1 : Sets the reference to s = -1. <i>i</i> = 0 : Sets the reference to s = 0. <i>i</i> = +1 : Sets the reference to s = +1.
sthi <i>t</i>	Set the beam thickness in the s-direction.
sthi1 <i>t</i>	Set the beam thickness in the s-direction at node #1.
sthi2 <i>t</i>	Set the beam thickness in the s-direction at node #2.

Global Material Commands (cont.)

Global Material Specification (cont.)

Command	Description
<u>mat - Options:</u>	
tloc <i>i</i>	Set the beam reference surface location in the t-direction. <i>i</i> may take on the following values: <i>i</i> = -1 : Sets the reference to t = -1. <i>i</i> = 0 : Sets the reference to t = 0. <i>i</i> = +1 : Sets the reference to t = +1.
tthi <i>t</i>	Set the beam thickness in the t-direction.
tthi1 <i>t</i>	Set the beam thickness in the t-direction at node #1.
tthi2 <i>t</i>	Set the beam thickness in the t-direction at node #2.
<i>Shell materials...</i>	
quad <i>n</i>	Specify that <i>n</i> of quadrature points be used through the shell thickness for DYNA3D.
shell	Forces all elements made with using current material to be constructed of thin shells.
shloc <i>i</i>	Set the shell reference surface location. <i>i</i> may take on the following values: <i>i</i> = -1 : Sets the reference to s = -1. <i>i</i> = 0 : Sets the reference to s = 0. <i>i</i> = +1 : Sets the reference to s = +1.
shthi <i>t</i>	Set the shell thickness.
shthi1 <i>t</i>	Set the shell thickness at node #1.
shthi2 <i>t</i>	Set the beam thickness at node #2.
shthi3 <i>t</i>	Set the beam thickness at node #3.
shthi4 <i>t</i>	Set the beam thickness at node #4.
stsi <i>n</i>	For DYNA3D, define the number of integration points, <i>n</i> , to use through the shell thickness. This option is only available when the shell option is selected for a material definition.
propt	Select the DYNA3D print output option. This option is only available when the shell option is selected for a material definition.

Global Material Commands (cont.)

Global Material Specification (cont.)

Command	Description
<u>mat - Options:</u>	
tshell	Forces all brick elements made with the current material to be constructed of thick shells. The tshell command can also be used in the eos - endeos pair for equations-of-state.
<i>Establishing the local t-axis orientation...</i>	
ltmin	The local t-axis for brick elements of the current material is the shortest direction through the brick.
ltmax	The local t-axis for brick elements of the current material is the longest direction through the brick.
<i>Modifying density and viscosity...</i>	
ro ρ	Set the current density to ρ .
bqt i	Use bulk viscosity type i .
bql Q_i	Linear bulk viscosity coefficient.
bqq Q_q	Quadratic bulk viscosity coefficient.
hgqt i	Set the hourglass viscosity type to i .
hgq Q_h	Hourglass viscosity coefficient.
<i>Entering DYNA3D equation-of-state information...</i>	
eos n $eost$	Begin an equation-of-state definition using eos form $eost$ and assign it to material model n .
endeos	Terminate this equation-of-state definition.
head	The next line within the input file will contain a title for the current equation-of-state being specified.
<i>DYNA3D viscosities...</i>	
ihg i	Set the hourglass viscosity to type i .
qh Q_h	Hourglass viscosity coefficient.
ibq i	Use bulk viscosity type i .

Global Material Commands (cont.)

Global Material Specification (cont.)

Command	Description
<u>mat - Options (cont.):</u>	
q1 Q_1	Quadratic viscosity coefficient.
q2 Q_2	Linear viscosity coefficient.
<i>Altering the default values for DYNA viscosities...</i>	
dbqt i	Set the default bulk viscosity type to i .
dql Q_i	Set the default linear bulk viscosity coefficient.
dqq Q_q	Set the default quadratic bulk viscosity coefficient.
dhqt i	Set the default hourglass viscosity type to i .
dhgq Q_h	Set the default hourglass viscosity.

DYNA3D Materials

While INGRID supports a considerable number of analysis programs, material type support is currently limited to DYNA3D, NIKE3D, TOPAZ, and ADINA. This section details INGRID commands which only affect material controls for DYNA3D. Material types for NIKE3D, TOPAZ3D, and ADINA are detailed in subsequent sections.

There are several commands which affect the viscosity of materials input to DYNA3D. They fall into two main categories - establishing viscosity settings and changing default values. The commands **ihg**, **qh**, **ibq**, **q1**, and **q2** establish viscosity values. The commands **dbqt**, **dql**, **dqq**, **dhgq**, and **dhqt** affect the default values. The hourglass viscosity type and hourglass coefficient are controlled using **ihg** and **qh**, respectively. The hourglass viscosity type can be selected from several options - standard DYNA3D, Flanagan-Belytschko, Flanagan-Belytschko with exact volume integration, and two stiffness form types. Input to the command is an index which selects the proper type. The hourglass coefficient for each type has been set to a default value of 0.10. That coefficient, however, may be modified using the **qh** command. Note: changing the hourglass coefficient to a value greater than 0.15 may cause instabilities in the analysis. The bulk viscosity type can be

controlled using the **ibq** command. It is also possible to adjust the default values of the quadratic and linear viscosity coefficients using the **q1** and **q2** commands. By default, the quadratic viscosity coefficient is set to 1.5 and the linear viscosity coefficient is established at 0.06.

When specifying a large number of material models, it is sometimes more convenient to simply use default values for many of the viscosity variables. The problem with using such values is that the default values established in DYNA3D are not always appropriate. Because of that, several commands are available within INGRID to establish default values for the DYNA3D viscosity variables - **dbqt**, **dql**, **dqq**, **dhgq**, and **dhqt**. The bulk viscosity type may be set using the **dbqt** command. The default linear bulk viscosity may be set using the **dql** command. The default quadratic bulk viscosity may be established using the **dqq** command. Two hourglass viscosity variables may be set using the **dhgq** and **dhqt** commands. Those commands change the default hourglass viscosity and the hourglass viscosity type, respectively.

DYNA3D Material Types

There are currently 31 different DYNA materials which INGRID supports. The complete list of material types is shown in Table 4.1. Since DYNA3D is capable of performing analysis on materials with complex pressure-volume relationships which requires equations-of-state for some material types. Data input for equations-of-state is much like that for a material model. The format in which equation-of-state information is specified is very much like the material models. Equation-of-state definitions are bounded by the input cards **eos** and **endeos**. The specific parameters for each equation-of-state are then entered in keyword-parameter style. Specific equation-of-state models are applied to particular material models by the parameters supplied with the **eos** command.

The input parameters for each of the DYNA3D material types are detailed on the following pages. For a full description of the parameters, the reader should refer to the material models section of the DYNA3D manual.

Table 4.1: DYNA3D Material Types

Type	Description	EOS Required	Element Types
1	Elastic	No	beam, shell, thick shell, hexahedron
2	Orthotropic Elastic	No	shell, thick shell, hexahedron
3	Kinematic / Isotropic Elastic-Plastic	No	beam, shell, thick shell, hexahedron
4	Thermo - Elastic-Plastic	No	beam, hexahedron
5	Soil & Crushable Foam	No	hexahedron
6	Viscoelastic	No	hexahedron
7	Blatz-Ko Rubber	No	hexahedron
8	High Explosive Burn	Yes	hexahedron
9	Null Hydrodynamics	Yes	hexahedron
10	Isotropic Elastic-Plastic Hydrodynamic	Yes	hexahedron
11	Temperature Dependent Elastoplastic Hydrodynamic	Yes	hexahedron
12	Isotropic Elastic-Plastic	No	shell, hexahedron
13	Elastic-Plastic with Failure	No	hexahedron
14	Soil and Crushable Foam with Failure	No	hexahedron
15	Johnson/Cook Strain and Temperature Sensitive Plasticity	Yes	shell, hexahedron
16	Pseudo Tensor Concrete/Geological	Yes	hexahedron
17	Isotropic Elastic-Plastic Oriented Crack	Yes	hexahedron
18	Power Law Isotropic Plasticity	No	hexahedron
19	Strain Rate Dependent Isotropic Plasticity	No	shell, hexahedron
20	Rigid Body	No	beam, shell, thick shell, hexahedron
21	Thermal Orthotropic Elastic	No	shell, hexahedron

Table 4.1: DYNA3D Material Types

Type	Description	EOS Required	Element Types
22	Composite Damage Model	No	shell, hexahedron
23	Thermal Orthotropic Elastic with 12 Curves	No	shell, hexahedron
24	Piecewise Linear Isotropic Plasticity	No	shell, hexahedron
25	Inviscid, Two Invariant Geological Cap	No	hexahedron
26	Metallic Honeycomb	No	hexahedron
27	Compressible Mooney-Rivlin Rubber	No	hexahedron
28	Resultant Plasticity	No	beam, shell
30	Closed-Form Update Shell Plasticity	No	shell
31	Frazer-Nash Rubber	No	hexahedron

DYNA3D Material Type 1: Elastic

DYNA material type 1 simulates an elastic material. In order to use it, the value for both the Young's modulus and the for Poisson's ratio must be specified. Both are required input to this material model specification.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus, E
pr v	Poisson's ratio, ν

DYNA3D Material Type 2: Orthotropic Elastic

The second material type available in DYNA3D models orthotropic elastic. This material type requires considerably more input than material type 1 because of the directional nature of orthotropic materials. Not only must the elastic properties be specified, but their respective direction needs to be input as well. To specify the elastic properties, values must be provided which specify the material law which relates stresses to strains.

Command	Description
<u>mat - Options (cont.):</u>	
ea E_a	Elastic modulus, E_a
eb E_b	Elastic modulus, E_b
ec E_c	Elastic modulus, E_c
prba ν_{ba}	Poisson's ratio, ν_{ba}
prca ν_{ca}	Poisson's ratio, ν_{ca}
prcb ν_{cb}	Poisson's ratio, ν_{cb}
gab G_{ab}	Shear modulus, G_{ab}
gbc G_{bc}	Shear modulus, G_{bc}
gca G_{ca}	Shear modulus, G_{ca}
aopt a_{opt}	Specifies the material axis option, a_{opt} , where $a_{opt} = 0.0$: The material axis is locally orthotropic with material axes determined by the element nodes. $a_{opt} = 1.0$: The material axis is locally orthotropic with material axes determined by the element center and a point in space. $a_{opt} = 2.0$: The material axis is globally orthotropic with material axes determined by two input vectors. $a_{opt} = 3.0$: SHELL ELEMENTS ONLY. The material axis is locally orthotropic with material axes determined by a vector in the plane of the shell and the shell normal.
<i>The following information must be supplied when aopt is set to 1.0:</i>	
xp x_p	The x-coordinate of the point in space.
yp y_p	The y-coordinate of the point in space.

DYNA3D Material Type 2: Orthotropic Elastic (cont.)

Command	Description
<u>mat - Options (cont.):</u>	
zp z_p	The z-coordinate of the point in space.
<i>The following information must be supplied when aopt is set to 2.0:</i>	
a1 a_x	The x-coordinate of the a vector.
a2 a_y	The y-coordinate of the a vector.
a3 a_z	The z-coordinate of the a vector.
d1 d_x	The x-coordinate of the d vector.
d2 d_y	The y-coordinate of the d vector.
d3 d_z	The z-coordinate of the d vector.

Material Type 3: Kinematic / Isotropic Elastic-Plastic

DYNA material type 3 is a kinematic / isotropic elastic-plastic model. Input to the material type consists of a value for Young's modulus, Poisson's ratio, and the yield stress. In addition, there are two hardening parameters - the hardening modulus and the hardening parameter, β . Isotropic, kinematic, or a linear combination of kinematic and isotropic hardening may be specified by varying the β parameter between 1 and 0. For β equal to 0 and 1, respectively, kinematic and isotropic hardening are obtained. For isotropic hardening ($\beta = 1.0$), it is recommended that material type 12 be used instead of material type 3. Material type 12 requires less storage and is more efficient.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
etan E_t	Tangent modulus
beta β	Hardening parameter

DYNA3D Material Type 4: Thermo-Elastic-Plastic

DYNA3D material type 4 simulates thermo-elastic-plastic behavior. The behavior of the material type is influenced by temperature. Because of that, parameters entered into the material type consist of not just single values for Young's modulus, Poisson's ratio, and yield stress, but rather arrays of values for each of those constants at specific temperatures. In addition, coefficients of thermal expansion need to be entered. The number of values for which parameters and temperatures are specified is controlled by the **npts** parameter.

At least two temperatures and their corresponding material properties must be entered for this material type. If the material temperature exceeds the range defined within the input, the analysis will terminate. It is also important to note that the coefficient of thermal expansion is defined with respect to the reference temperature at the beginning of the calculation for the material.

Command	Description
<u>mat - Options (cont.):</u>	
npts n	The number of temperature values for which material constants are defined.
temp $T_1 T_2 T_3 \dots T_n$	Temperatures
e $E_1 E_2 E_3 \dots E_n$	Young's modulus for each of the temperatures
pr $\nu_1 \nu_2 \nu_3 \dots \nu_n$	Poisson's ratio for each of the temperatures
alpha $\alpha_1 \alpha_2 \alpha_3 \dots \alpha_n$	Coefficients of thermal expansion
sigy $\sigma_{y1} \sigma_{y2} \sigma_{y3} \dots \sigma_{yn}$	Yield stresses for each of the temperatures
etan $E_{t1} E_{t2} E_{t3} \dots E_{tn}$	Tangent modulus for each of the temperatures

DYNA3D Material Type 5: Soil and Crushable Foam

Material type 5 simulates soil and crushable foam. Input parameters to the material type include a shear and bulk unloading modulus, three yield function constants, and a pressure cutoff for tensile fracture. In addition to the constants noted below, a series of values must be included which define the volumetric strain vs. pressure curve for the model. Input to that curve consists of up to 10 pairs of coefficients (pressure values are positive and volumetric strain values are negative in compression - the volumetric strain is given by the natural log of the relative volume). The pairs of coefficients should be tabulated in order of increasing compression. If the pressure drops below the cutoff value supplied in the pressure cutoff constant, it is reset to that constant.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
ku K_u	Bulk unloading modulus
a0 a_0	Yield function constant, a_0
a1 a_1	Yield function constant, a_1
a2 a_2	Yield function constant, a_2
pc p_c	Pressure cutoff for tensile fracture
npts n	The number of points in the volumetric strain vs. pressure curve (n must be less than or equal to 10)
vs ϵ_{v1} ϵ_{v2} ϵ_{v3} ... ϵ_{vn}	Volumetric strain values
p p_1 p_2 p_3 ... p_n	Corresponding pressure values for each of the volumetric strain constants

DYNA3D Material Type 6: Viscoelastic

Input to DYNA3D material type 6 includes a bulk modulus, two parameters for the shear modulus, and a decay constant.

Command	Description
<u>mat - Options (cont.):</u>	
k K	Elastic bulk modulus
g0 G_0	Short-time shear modulus
gi $G_{infinity}$	Long-time shear modulus
beta β	Decay constant

DYNA3D Material Type 7: Blatz-Ko Rubber

Material type 7 is the first of the two rubber types available in DYNA3D. In this case, the Blatz-Ko form of the stored energy function is used. Input to the material type consists of a single shear modulus.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus

DYNA3D Material Type 8: High Explosive Burn

High explosive burns may be modeled in DYNA3D using material type 8. Input to the material type consists of two constants, the detonation velocity and the Chapman-Jouget pressure. In addition to the constants supplied in the material type parameters, an equation-of-state must be included due to the complicated pressure/volume behavior exhibited by this material type.

Command	Description
<u>mat - Options (cont.):</u>	
d	Detonation velocity
pcj P_{CJ}	Chapman-Jouget pressure

DYNA3D Material Type 9: Null Hydrodynamics

Material type 9 is the first of the three DYNA3D material types which model hydrodynamics. Input consists of a pressure cutoff and a viscosity coefficient. The pressure cutoff is negative when in tension. An equation-of-state is required for this material type.

Command	Description
<u>mat - Options (cont.):</u>	
pc P_{cut}	Pressure cutoff
mu μ	Viscosity coefficient

DYNA3D Material Type 10: Isotropic-Elastic-Plastic-Hydrodynamic

Material type 10 is the second of three DYNA3D material types which model hydrodynamics. Input for this material type differs from the previous material type in that a shear modulus, yield strength, a hardening modulus, and a yield stress-effective plastic strain curve or yield stress-pressure curve is required in addition to the pressure cutoff. Yield stress may be defined as a function of plastic strain or pressure but not both. An equation-of-state is required for this material type.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
sigy σ_y	Yield strength
eh E_p	Plastic modulus
pc p_{cut}	Pressure cutoff
npts n	The number of points in the yield stress-effective plastic strain curve or yield stress-pressure curve.
es $\sigma_1 \sigma_2 \sigma_3 \dots \sigma_n$	Yield stresses
eps $\epsilon_{p1} \epsilon_{p2} \epsilon_{p3} \dots \epsilon_{pn}$	Effective plastic strains

DYNA3D Material Type 11: Temperature Dependent, Elastoplastic, Hydrodynamic

The third hydrodynamic material type available in DYNA3D is material type 11. The theoretical basis for this material type is the paper by Steinberg and Guinan. Interested users should consult the DYNA3D manual for more information and for references.

Command	Description
<u>mat - Options (cont.):</u>	
g0 G_0	Shear modulus constant
sig0 σ_0	Yield stress constant
beta β	Strain hardening law constant
n n	Strain hardening exponent
gama γ_i	Initial plastic strain
sigm σ_m	Yield stress work hardening limit
b b	Shear modulus pressure constant
bp b'	Yield stress pressure constant
h h	Energy coefficient
f f	Energy exponential coefficient
a A	Atomic weight
t0 T_{mo}	Melting temperature constant
gam0 γ_0	Thermodynamic gamma
pc P_{cut}	Pressure cutoff
ec0 EC_0	EC_0
ec1 EC_1	EC_1
ec2 EC_2	EC_2
ec3 EC_3	EC_3
ec4 EC_4	EC_4
ec5 EC_5	EC_5
ec6 EC_6	EC_6
ec7 EC_7	EC_7
ec8 EC_8	EC_8
ec9 EC_9	EC_9
spall $spall$	Specifies the Spall type (0.0, 1.0, 2.0 or 3.0)

DYNA3D Material Type 12: Isotropic-Elastic-Plastic

DYNA3D material type 12 models isotropic-elastic plastic. Input to the material type consists of a shear modulus, a yield strength, and a hardening and bulk modulus.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
sigy σ_y	Yield strength
eh E_h	Plastic hardening modulus
k K	Bulk modulus

DYNA3D Material Type 13: Elastic-Plastic with Failure Model

DYNA3D material type 13 is the first of several material types which simulate failure. When the effective plastic strain reaches the failure strain, or when the pressure reaches the failure pressure, elements lose their ability to carry tension. When that happens, the deviatoric stresses are set to zero and the material behaves like a fluid.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
sigy σ_y	Yield strength
eh E_h	Plastic hardening modulus
k K	Bulk modulus
fs ϵ_f	Failure strain
fp p_f	Failure pressure

DYNA3D Material Type 14: Soil and Crushable Foam with Failure Model

The input to DYNA3D material type 14 is identical to DYNA3D material type 5. The behavior of the material type is the same with the exception that when the pressure reaches the failure pressure, elements lose their ability to carry tension.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
ku K_u	Bulk unloading modulus
a0 a_0	Yield function constant, a_0
a1 a_1	Yield function constant, a_1
a2 a_2	Yield function constant, a_2
pc p_c	Pressure cutoff for tensile fracture
npts n	The number of points in the volumetric strain vs. pressure curve (n must be less than or equal to 10.)
vs ϵ_{v1} ϵ_{v2} ϵ_{v3} ... ϵ_{vn}	Volumetric strain values
p p_1 p_2 p_3 ... p_n	Corresponding pressure values for each of the volumetric strain constants.

DYNA3D Material Type 15: Johnson/Cook Strain and Temperature Sensitive Plasticity

DYNA3D material type 15 models temperature sensitive plasticity. It is the result of a model presented by G.R. Johnson and W.H. Cook. Refer to the DYNA3D manual for more information.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
a A	Yield stress coefficient
b B	Strain hardening coefficient
n n	Strain hardening exponent
c C	Strain rate dependence coefficient
m m	Temperature dependence coefficient
tm T_m	Melt temperature
tr T_r	Room temperature
x0 ϵ_0	Reference strain rate
sh c_v	Specific heat
pmin p_{min}	Pressure cutoff, p_{min} , or failure stress, σ_m
spall $spall$	Spall type (0.0, 1.0, 2.0 or 3.0)
iterate $flag$	Plastic strain iteration flag (0.0 or 1.0)
d1 D_1	D_1
d2 D_2	D_2
d3 D_3	D_3
d4 D_4	D_4
d5 D_5	D_5

DYNA3D Material Type 16: Pseudo TENSOR Concrete/Geological Model

DYNA3D material type 16 is a pseudo tensor concrete/geological model. If the yield stress - plastic strain curve is not defined, and if α_1, α_2 are zero, the bilinear stress-strain relationship is obtained with $\beta = 1.0$. Note that this material type requires an equation-of-state.

Command	Description
<u>mat - Options (cont.):</u>	
g G	Shear modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
eh E_h	Plastic hardening modulus
pc p_c	Pressure cutoff
a1 a_1	Yield function constant, a_1
a2 a_2	Yield function constant, a_2
rsc c	Rate sensitive coefficient
rsa a	Rate sensitive coefficient
rse e	Rate sensitive exponent
ssm b	Strain softening modulus
pf f	Post-fracture coefficient of internal friction
fs ϵ_f	Failure strain
npts n	The number of points in the yield stress vs. effective plastic strain curve or yield stress vs. pressure curve.
es $\sigma_1 \sigma_2 \sigma_3 \dots \sigma_n$	Yield stress values
eps $\epsilon_{p1} \epsilon_{p2} \epsilon_{p3} \dots \epsilon_{pn}$	Effective plastic strain values
p $p_1 p_2 p_3 \dots p_n$	Corresponding pressure values for each of the yield stress or effective plastic strain values.

DYNA3D Material Type 17: Isotropic Elastic-Plastic Oriented Crack Model

DYNA3D material type 17 is another isotropic elastic plastic material type. In this material type, a fracture stress is used. When the maximum principal stress exceeds the fracture stress, the element fails on a plane perpendicular to the direction of the maximum principal stress. In tension, the element will not carry any stresses on the fracture plane, but, in compression, it will carry both normal and shear stresses. If the fracture stress is exceeded in another direction, the element fails losing its ability to carry tension, the deviatoric stresses are set to zero, and the material behaves as a fluid. Input to material type 17 is much like that of material type 3 where Young's modulus, Poisson's ratio, the yield stress, hardening modulus, and pressure cutoff are specified. In addition, a fracture strength is included to control the crack behavior.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
eh E_h	Hardening modulus
fs f_s	Fracture strength
pc p_c	Pressure cutoff

DYNA3D Material Type 18: Power Law Isotropic Plasticity

DYNA3D material type 18 simulates isotropic plasticity using a power law. In it, elastoplastic behavior with isotropic hardening is specified in the material type.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
k k	Strength coefficient
n n	Hardening exponent

DYNA3D Material Type 19: Strain Rate Dependent Isotropic Plasticity

DYNA3D material type 19 models strain rate dependent plasticity. Input to the material type consists of Young's modulus, Poisson's ratio, and a hardening modulus. In addition to the above parameters, a time step size needs to be supplied in order to control automatic element deletion.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
lcs0 lc_0	Load curve number defining Young's modulus as a function of strain rate
et E_t	Hardening modulus
tss tss	Time step size for automatic element deletion
lce lc_e	Load curve number defining Young's modulus as a function of strain rate (optional)
lct lc_t	Load curve number defining the tangent modulus as a function of strain rate (optional)
lcs lc_s	Load curve number defining Von Mises stress at failure as a function of strain rate (optional)

DYNA3D Material Type 20: Rigid

DYNA3D material type 20 may be used for rigid body analysis. Input to the material type consists of values for Young's modulus and Poisson's ratio. Because INGRID is capable of setting the appropriate velocity specification cards for rigid bodies, a velocity may be specified for the material using the **velo** parameter. The velocity load curve is specified using the **fv** command.

Command	Description
<u>mat - Options (cont.):</u>	
E <i>e</i>	Specify a value of <i>e</i> for Young's modulus
pr <i>v</i>	Specify a value of <i>v</i> for Poisson's ratio
velo <i>l_c A n v_x v_y v_z</i>	Specify a prescribed rigid body velocity or rotation using load curve <i>lc</i> with amplitude <i>A</i> . ($v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$) indicates the direction for a translational velocity or a rotational velocity. <i>n</i> specifies the applicable degrees of freedom, and may take on the following values: <i>n</i> = 1 : x-translational degree-of-freedom <i>n</i> = 2 : y-translational degree-of-freedom <i>n</i> = 3 : z-translational degree-of-freedom <i>n</i> = 4 : translational velocity in direction of $v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ <i>n</i> = 5 : x-rotational degree-of-freedom <i>n</i> = 6 : y-rotational degree-of-freedom <i>n</i> = 7 : z-rotational degree-of-freedom <i>n</i> = 8 : rotational velocity in direction of $v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ <i>n</i> = 9 : tangential component for rotation about global x-axis <i>n</i> = 10 : tangential component for rotation about global y-axis <i>n</i> = 11 : tangential component for rotation about global z-axis

DYNA3D Material Type 21: Thermal Orthotropic Elastic

DYNA3D material type 21 simulates orthotropic elastic material behavior much like material type 2, but with the added feature of computing thermal stress. For information regarding the parameters and a discussion on the constitutive matrix and the orientation of the local material axis, refer to the discussion on material type 2 in the DYNA3D manual.

Command	Description
<u>mat - Options (cont.):</u>	
ea E_a	Elastic modulus, E_a
eb E_b	Elastic modulus, E_b
ec E_c	Elastic modulus, E_c
prba ν_{ba}	Poisson's ratio, ν_{ba}
prca ν_{ca}	Poisson's ratio, ν_{ca}
prcb ν_{cb}	Poisson's ratio, ν_{cb}
gab G_{ab}	Shear modulus, G_{ab}
gbc G_{bc}	Shear modulus, G_{bc}
gca G_{ca}	Shear modulus, G_{ca}
alaa α_{aa}	Coefficient of thermal expansion, α_{aa}
alab α_{ab}	Coefficient of thermal expansion, α_{ab}
alac α_{ac}	Coefficient of thermal expansion, α_{ac}
aopt a_{opt}	Specifies the material axis option, a_{opt} (0.0, 1.0, 2.0 or 3.0)
<i>The following information must be supplied when aopt is set to 1.0:</i>	
xp x_p	The x-coordinate of the point in space.
yp y_p	The y-coordinate of the point in space.
zp z_p	The z-coordinate of the point in space.
<i>The following information must be supplied when aopt is set to 2.0:</i>	
a1 a_x	The x-coordinate of the a vector.
a2 a_y	The y-coordinate of the a vector.
a3 a_z	The z-coordinate of the a vector.
d1 d_x	The x-coordinate of the d vector.
d2 d_y	The y-coordinate of the d vector.
d3 d_z	The z-coordinate of the d vector.

DYNA3D Material Type 22: Composite Damage Model

DYNA3D material type 22 is a composite damage model. For information regarding the parameters and a discussion on the constitutive matrix and the orientation of the local material axis, refer to the discussion on material type 2 in the DYNA3D manual.

Command	Description
<u>mat - Options (cont.):</u>	
ea E_a	Elastic modulus, E_a
eb E_b	Elastic modulus, E_b
ec E_c	Elastic modulus, E_c
prba ν_{ba}	Poisson's ratio, ν_{ba}
prca ν_{ca}	Poisson's ratio, ν_{ca}
prcb ν_{cb}	Poisson's ratio, ν_{cb}
gab G_{ab}	Shear modulus, G_{ab}
gbc G_{bc}	Shear modulus, G_{bc}
gca G_{ca}	Shear modulus, G_{ca}
k K_f	Bulk modulus of failed material
sc S_c	Shear strength, ab-plane
xt x_t	Longitudinal tensile strength, a-axis
yt y_t	Longitudinal tensile strength, b-axis
yc y_c	Transverse compressive strength
nssp s	Nonlinear shear stress
nip $n \beta_1 \beta_2 \beta_3 \dots \beta_n$	Input n material angles (β_1 – β_n) at the integration points
aopt a_{opt}	Specifies the material axis option, a_{opt} (0.0, 1.0, 2.0 or 3.0)
<i>The following information must be supplied when a_{opt} is set to 1.0:</i>	
xp x_p	The x-coordinate of the point in space.
yp y_p	The y-coordinate of the point in space.
zp z_p	The z-coordinate of the point in space.

Command	Description
---------	-------------

mat - Options (cont.):

The following information must be supplied when aopt is set to 2.0:

a1 a_x	The x-coordinate of the a vector.
a2 a_y	The y-coordinate of the a vector.
a3 a_z	The z-coordinate of the a vector.
d1 d_x	The x-coordinate of the d vector.
d2 d_y	The y-coordinate of the d vector.
d3 d_z	The z-coordinate of the d vector.

DYNA3D Material Type 23: Thermal Orthotropic Elastic with 12 Curves

DYNA3D material type 23 simulates orthotropic elastic material behavior much like material type 2, but with the added feature of computing thermal stress. For information regarding the parameters and a discussion on the constitutive matrix and the orientation of the local material axis, refer to the discussion on material type 2 in the DYNA3D manual.

Command	Description
<u>mat - Options (cont.):</u>	
npts n	The number of points in the curves
ea $E_{a1} E_{a2} E_{a3} \dots E_{an}$	Elastic modulus, E_a
eb $E_{b1} E_{b2} E_{b3} \dots E_{bn}$	Elastic modulus, E_b
ec $E_{c1} E_{c2} E_{c3} \dots E_{cn}$	Elastic modulus, E_c
prba $\nu_{ba1} \nu_{ba2} \dots \nu_{ban}$	Poisson's ratio, ν_{ba}
prca $\nu_{ca1} \nu_{ca2} \dots \nu_{can}$	Poisson's ratio, ν_{ca}
prcb $\nu_{cb1} \nu_{cb2} \dots \nu_{cbn}$	Poisson's ratio, ν_{cb}
gab $G_{ab1} G_{ab2} \dots G_{abn}$	Shear modulus, G_{ab}
gbc $G_{bc1} G_{bc2} \dots G_{bcn}$	Shear modulus, G_{bc}
gca $G_{ca1} G_{ca2} \dots G_{can}$	Shear modulus, G_{ca}
t $t_1 t_2 t_3 \dots t_n$	Temperature values
nip $n \beta_1 \beta_2 \beta_3 \dots \beta_n$	Input n material angles (β_1 – β_n) at the integration points
aopt a_{opt}	Specifies the material axis option, a_{opt} (0.0, 1.0, 2.0 or 3.0)
<i>The following information must be supplied when aopt is set to 1.0:</i>	
xp x_p	The x-coordinate of the point in space.
yp y_p	The y-coordinate of the point in space.
zp z_p	The z-coordinate of the point in space.
<i>The following information must be supplied when aopt is set to 2.0:</i>	
a1 a_x	The x-coordinate of the a vector.
a2 a_y	The y-coordinate of the a vector.
a3 a_z	The z-coordinate of the a vector.
d1 d_x	The x-coordinate of the d vector.
d2 d_y	The y-coordinate of the d vector.
d3 d_z	The z-coordinate of the d vector.

DYNA3D Material Type 24: Piecewise Linear Isotropic Plasticity

DYNA3D material type 24 models piecewise linear isotropic plasticity. Input to the material type consists of Young's modulus, Poisson's ratio, a yield stress, and a tangent modulus. If a stress-strain curve is defined, the tangent modulus input will be ignored. Strain rate effects are accounted for by scaling the yield stress using a load curve number. Because of that, a load curve number must be supplied for this material type. In addition to the above parameters, a time step size needs to be supplied in order to control automatic element deletion.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
lc lc_{num}	The load curve number in which to scale the yield stress to account for strain rate effects
psf psf	Plastic strain at failure
tss tss	Time step size for automatic element deletion
tm $tmod$	Tangent modulus (ignored if stress-strain curve is defined)
npts n	The number of points in the yield stress vs. effective plastic strain curve or yield stress vs. pressure curve.
es $\sigma_1 \sigma_2 \sigma_3 \dots \sigma_n$	Yield stress values
eps $\epsilon_{p1} \epsilon_{p2} \epsilon_{p3} \dots \epsilon_{pn}$	Effective plastic strain values

DYNA3D Material Type 25: Inviscid Two Invariant Geologic Cap Model

DYNA3D material type 25 simulates an inviscid two invariant geological cap model. See the DYNA3D manual for more information and for references.

Command	Description
<u>mat - Options (cont.):</u>	
k K	Bulk modulus
g G	Shear modulus
t t	Tension cutoff
ltype $type$	$type$ (1 = soil/concrete, 2 = rock)
alpha α	Alpha
theta θ	Theta
gamma γ	Gamma
beta β	Beta
r r	R
d d	D
w w	W
x0 x_0	X_0
nplot $type$	Specify the TAURUS iplot parameter for this model (1.0 to 9.0)

DYNA3D Material Type 26: Metallic Honeycomb

DYNA3D material type 26 models a metallic honeycomb. Input to the material type consists of Young's modulus, Poisson's ratio, a yield stress, several load curves, and several elastic moduli. The behavior for the fully compacted honeycomb material is treated as elastic - perfectly plastic. The behavior before compaction is orthotropic where the components of the stress tensor are uncoupled, i.e., an x component of strain will generate resistance in the x direction with no coupling to the y and z directions. The elastic moduli vary from their initial values to the fully compacted values linearly with the relative volume.

The load curves define the magnitude of the average stress as the material changes density (relative volume). By properly specifying these curves, one dimensional crushing can be achieved. Each curve related to this material type must have the same number of points and the same abscissa values. The first value in the curve should correspond to a value of relative volume slightly less than the fully compacted value.

The orthotropic material axis are defined one way, but can be generalized if needed. The local z-direction is found by computing the cross product of two vectors corresponding to the diagonals of the lower surface of the brick element. One vector beginning at node 4 and terminating at node 2 is crossed with the vector from node 1 to node 3. The local x direction is determined by side 1-2, and the local y-direction is found by crossing the z vector with the x vector. The vector from node 1 to node 2 (side 1-2) is modified so that it is orthogonal to the normal vector.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress for fully compacted honeycomb
sigxx i	Load curve number i relates σ_{xx} to the relative volume
sigyy i	Load curve number i relates σ_{yy} to the relative volume
sigzz i	Load curve number i relates σ_{zz} to the relative volume
ssrv i	Load curve number i relates shear stress to relative volume
crv x	Relative volume at which the honeycomb is fully compacted
exx E_{xx}	Elastic modulus E_{xx} in uncompressed configuration
eyy E_{yy}	Elastic modulus E_{yy} in uncompressed configuration
ezz E_{zz}	Elastic modulus E_{zz} in uncompressed configuration
gxy G_{xy}	Elastic shear modulus G_{xy} in uncompressed configuration
gyz G_{yz}	Elastic shear modulus G_{yz} in uncompressed configuration
gzx G_{zx}	Elastic shear modulus G_{zx} in uncompressed configuration

DYNA3D Material Type 27: Compressible Mooney-Rivlin Rubber

DYNA3D material type 27 provides an alternative to the Blatz-Ko rubber material type. Input to the material type consists of two constants, A and B , and Poisson's ratio.

Command	Description
<u>mat - Options (cont.):</u>	
a A	A
b B	B
pr ν	Poisson's ratio

DYNA3D Material Type 28: Resultant Plasticity

DYNA3D material type 28 models resultant plasticity. It is available only for the Belytschko-Schwer beam element and the Belytschko-Tsay shell. For beams, the treatment is elastic-perfectly plastic. For shell elements, isotropic hardening is approximately modeled. Input to the material type consists of values for Young's modulus, Poisson's ratio, the yield stress, and a hardening modulus.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
etan E_t	Hardening modulus (shells only)

DYNA3D Material Type 30: Closed-Form Update Shell Plasticity

Material type 30 implements a closed form solution for the plane stress constitutive update under conditions of perfect plasticity or kinematic hardening for a bilinear Von Mises model. Input to the material type consists of values for Young's modulus, Poisson's ratio, the yield stress, and a hardening modulus.

Command	Description
<u>mat - Options (cont.):</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
etan E_t	Hardening modulus

DYNA3D Material Type 31: Frazer-Nash Rubber Model

DYNA3D material type 31 is a Frazer-Nash Rubber model developed by C.J. Kenchington. It models a hyperelastic constitutive law. The material type is similar to the Blatz-Ko rubber model, but differs in that it can be used for general non-linear elastic behavior at relatively low strain levels. Input to the material type consists of several constants which make up the terms of the stress-strain relationships in the strain energy density function. Input also consists of establishing minimum and maximum strain limits and specifying whether the analysis should continue if those limits are exceeded.

Command	Description
<u>mat - Options (cont.):</u>	
g001 G_{001}	See the DYNA3D manual for a description of these constants
g010 G_{010}	
g020 G_{020}	
g100 G_{100}	
g101 G_{101}	
g110 G_{110}	
g200 G_{200}	
g210 G_{210}	
g300 G_{300}	
g400 G_{400}	
stmx ϵ_{max}	Maximum strain limit
stmn ϵ_{min}	Minimum strain limit
exit <i>value</i>	Establish the <i>exit</i> or <i>continue</i> option (0.0 or 1.0)

DYNA3D Equation of State Types

Because DYNA3D is capable of performing analysis with material models where complex pressure-volume relationships exist, some of the material type definitions require an equation-of-state. The specific DYNA3D material types which require equation-of-state information are type numbers 8 through 11 and 15 through 17. A complete list of the equation-of-state forms are presented in Table 4.2.

Providing equation-of-state information is accomplished much like setting up a material model. Each equation-of-state definition is bounded by two cards - **eos** and **endeos** - which signify the beginning and end to an equation-of-state definition. Placed between those two commands are the necessary parameters which define the behavior of a specific equation-of-state. Assignment of particular equations-of-state to specific material models is accomplished through parameters supplied in the **eos** command.

Table 4.2: DYNA3D Equation-of-State Forms

Form #	Description
1	Linear Polynomial
2	JWL
3	Sack "Tuesday"
4	Gruneisen
5	Ratio of Polynomials
6	Linear Polynomial with Energy Leak
7	Ignition and Growth of Reaction in HE
8	Tabulated Compaction
9	Tabulated
11	Tensor Pore Collapse

Equation-of-State Definitions

Command	Description
eos <i>n options</i> endeos	Equation-of-state models for DYNA3D can be input with this command.

DYNA3D Equation-of-State Form 1: Linear Polynomial

The first equation-of-state type available in DYNA3D is a linear polynomial. Input consists of several coefficients which make up the polynomial, initial internal energy, and initial relative volume.

Command	Description
<u>eos - Options (cont.):</u>	
c0 C_0	C_0 - C_6 and E_0 are constants used in the pressure calculation
c1 C_1	
c2 C_2	
c3 C_3	
c4 C_4	
c5 C_5	
c6 C_6	
e0 E_0	Initial internal energy
v0 V_0	Initial relative volume

DYNA3D Equation-of-State Form 2: JWL

The input for the JWL High Explosive equation-of-state consists of coefficients for an equation which defines the pressure, the initial internal energy, and the initial relative volume.

Command	Description
<u>eos - Options (cont.):</u>	
a A	The constants A through OMEGA are used in the pressure calculation for the JWL equation-of-state
b B	
r1 R₁	
r2 R₂	
omega ω	
e0 E₀	Initial internal energy
v0 V₀	Initial relative volume

DYNA3D Equation-of-State Form 3: Sack "Tuesday"

An alternative to the JWL High Explosive is the Sack "Tuesday" equation-of-state form. The Sack Tuesday" equation-of-state is typically used for the detonation products of high explosives.

Command	Description
<u>eos - Options (cont.):</u>	
a1 A₁	The constants A ₁ through B ₂ are used in the pressure calculation for the Sack equation-of-state
a2 A₂	
b1 B₁	
b2 B₂	
e0 E₀	Initial internal energy
v0 V₀	Initial relative volume

DYNA3D Equation-of-State Form 4: Gruneisen

The Gruneisen equation-of-state with cubic shock velocity-particle velocity.

Command	Description
<u>eos - Options (cont.):</u>	
sp C	Velocity curve intercept
s1 S_1	First slope coefficient
s2 S_2	Second slope coefficient
s3 S_3	Third slope coefficient
gamma γ_0	Gruneisen coefficient
sa a	First order volume correction coefficient, a
e0 E_0	Initial internal energy
v0 V_0	Initial relative volume

DYNA3D Equation-of-State Form 5: Ratio of Polynomials

This equation-of-state was adopted from the KOVEC code.

Command	Description
<u>eos - Options (cont.):</u>	
a10 A_{10}	The constants A_{10} through A_{73} are used to define the equation-of-state pressure
a11 A_{11}	
a12 A_{12}	
a13 A_{13}	
a20 A_{20}	
a21 A_{21}	
a22 A_{22}	

Command	Description
<u>eos - Options (cont.):</u>	
A23 <i>A₂₃</i>	
a30 <i>A₃₀</i>	
a31 <i>A₃₁</i>	
a32 <i>A₃₂</i>	
a33 <i>A₃₃</i>	
a40 <i>A₄₀</i>	
a41 <i>A₄₁</i>	
a42 <i>A₄₂</i>	
a43 <i>A₄₃</i>	
a50 <i>A₅₀</i>	
a51 <i>A₅₁</i>	
a52 <i>A₅₂</i>	
a53 <i>A₅₃</i>	
a60 <i>A₆₀</i>	
a61 <i>A₆₁</i>	
a62 <i>A₆₂</i>	
a63 <i>A₆₃</i>	
a70 <i>A₇₀</i>	
a71 <i>A₇₁</i>	
a72 <i>A₇₂</i>	
a73 <i>A₇₃</i>	
alpha α	
beta β	
a14 <i>A₁₄</i>	
a24 <i>A₂₄</i>	

Command	Description
<u>eos - Options (cont.):</u>	
coef $A_{10} A_{11} \dots \beta A_{14} A_{24}$	The coefficients A_{10} through A_{24} may alternatively be specified using the coef command. All of the above 32 constants must be provided in the same order as listed above.
e0 E_0	Initial internal energy
v0 V_0	Initial relative volume

DYNA3D Equation-of-State Form 6: Linear Polynomial with Energy Leak

DYNA3D equation-of-state type 6 is much like form 1. It is a linear polynomial with an added capability of being able to incorporate an energy leak into the pressure calculation. The energy deposition rate is supplied through a time history curve identified using the **cn** parameter. Input to the equation-of-state definition consists of several coefficients which make up the polynomial, the initial internal energy, and the initial relative volume to be used.

Command	Description
<u>eos - Options (cont.):</u>	
c0 C_0	C_0 - C_6 and E_0 are constants used in the pressure calculation
c1 C_1	
c2 C_2	
c3 C_3	
c4 C_4	
c5 C_5	
c6 C_6	
e0 E_0	Initial internal energy
v0 V_0	Initial relative volume
cn cn	The number of the time history curve which dictates the energy deposition rate

DYNA3D Equation-of-State Form 7: Ignition and Growth of Reaction in HE

DYNA3D equation-of-state form 7 attempts to model the reaction of the high explosive more accurately than equation-of-state form 2.

Command	Description
<u>eos - Options (cont.):</u>	
ap A_p	See the DYNA3D manual for a description of these parameters
bp B_p	
r1p R_{1p}	
r2p R_{2p}	
g G	Second ignition coefficient
wpcp $\omega_p c_p$	
ae A	
be B	
wece $\omega_e c_e$	
r1e R_{1e}	
r2e R_{2e}	
fcrit F_{crit}	Critical fraction reacted
i I	First ignition coefficient
h H	Growth coefficient
z z	Pressure exponent
x x	
y y	
cp c_p	Heat capacity of reaction products
ce c_e	Heat capacity of unreacted HE
m m	Product compression exponent
e0 E_0	Initial energy of HE per unit volume
t0 T_0	Initial temperature

DYNA3D Equation-of-State Form 8: Tabulated Compaction

As many as 10 points and as few as 2 may be used when defining the tabulated compaction functions. DYNA3D will extrapolate to find the pressure if necessary.

Command	Description
<u>eos - Options (cont.):</u>	
npts n	The number of points in the tabulated curves
inv $\epsilon_{v1} \epsilon_{v2} \dots \epsilon_{vn}$	Volumetric strain points
pc $c_1 c_2 \dots c_n$	Points on the curve for c
pt $t_1 t_2 \dots t_n$	Points on the curve for t
ku $k_1 k_2 \dots k_n$	Points on the curve for the unloading bulk modulus
gamma γ	Gamma
e0 E_0	Initial internal energy
v0 V_0	Initial relative volume

DYNA3D Equation-of-State Form 9: Tabulated

As many as 10 points and as few as 2 may be used when defining tabulated functions. DYNA3D will extrapolate to find the pressure if necessary.

Command	Description
<u>eos - Options (cont.):</u>	
npts n	The number of points in the tabulated curves
inv $\epsilon_{v1} \epsilon_{v2} \dots \epsilon_{vn}$	Volumetric strain points
pc $c_1 c_2 \dots c_n$	Points on the curve for c
pt $t_1 t_2 \dots t_n$	Points on the curve for t
gamma γ	Gamma
e0 E_0	Initial internal energy
v0 V_0	Initial relative volume

DYNA3D Equation-of-State Form 11: TENSOR Pore Collapse

The virgin loading and completely crushed curves are modeled with monotonic cubic-splines. An optimized vector interpolation scheme is then used to evaluate the cubic-splines.

Command	Description
<u>eos - Options (cont.):</u>	
mu1 μ_1	Excess compression required before any pores collapse
mu2 μ_2	Excess compression point where the Virgin Loading Curve and Completely Crushed Curve intersect
ie E_0	Initial internal energy
iec iec	Initial excess compression
nld n_{ld}	The number of virgin loading curve points
virgin $c_1 p_1 \dots c_{nld} p_{nld}$	Specifies points on the virgin load curve. The c_i values represent excess compression and the p_i values are corresponding pressures.
ncr n_{cr}	The number of completely crushed curve points
crushed $c_1 p_1 \dots c_{ncr} p_{ncr}$	Specifies points on the completely crushed curve. The c_i values represent excess compression and the p_i values are corresponding pressures.

NIKE3D Materials

This section details INGRID commands which only affect material definitions for NIKE3D. There are currently 8 NIKE3D material types supported by INGRID. The complete list of material types is shown in Table 4.3. Except for material types 7 and 8, input to these material types is very similar to those input for their DYNA3D counterparts. The detailed information for these material types can be found in the NIKE3D user's manual.

Table 4.3: NIKE3D Material Types

Type	Description	Element Types
1	Elastic	beam, shell, thick shell, hexahedron
2	Orthotropic Elastic	hexahedron
3	Elastic-Plastic	beam, shell, thick shell, hexahedron
4	Thermo Elastic-Plastic	hexahedron
5	Soil and Crushable Foam	hexahedron
6	Linear Viscoelastic	hexahedron
7	Thermo Orthotropic Elastic	hexahedron
8	Thermo Elastic-Creep	hexahedron

NIKE3D Material Type 1: Elastic

This material type models isotropic, elastic material behavior. The only input required to use this material type is Young's modulus and Poisson's ratio.

Command	Description
mat n options endmat	Material models for a specific code can be input with this command.
<u>mat - Options:</u>	
e E	Young's modulus
pr v	Poisson's ratio

NIKE3D Material Type 2: Orthotropic Elastic

This material type requires considerably more input than the first material type because of the nature of orthotropic materials. Not only must the elastic properties be specified, but their principal directions need to be input as well.

Command	Description
<u>mat - Options:</u>	
ea E_a	Elastic modulus, E_a
eb E_b	Elastic modulus, E_b
ec E_c	Elastic modulus, E_c
prba ν_{ba}	Poisson's ratio, ν_{ba}
prca ν_{ca}	Poisson's ratio, ν_{ca}
prcb ν_{cb}	Poisson's ratio, ν_{cb}
gab G_{ab}	Shear modulus, G_{ab}
gbc G_{bc}	Shear modulus, G_{bc}
gca G_{ca}	Shear modulus, G_{ca}
aopt a_{opt}	Specifies the material axis option, a_{opt} (0.0, 1.0, 2.0 or 3.0)
<i>The following information must be supplied when aopt is set to 1.0:</i>	
xp x_p	The x-coordinate of the point in space.
yp y_p	The y-coordinate of the point in space.
zp z_p	The z-coordinate of the point in space.
<i>The following information must be supplied when aopt is set to 2.0:</i>	
a1 a_x	The x-coordinate of the a vector.
a2 a_y	The y-coordinate of the a vector.
a3 a_z	The z-coordinate of the a vector.
d1 d_x	The x-coordinate of the d vector.
d2 d_y	The y-coordinate of the d vector.
d3 d_z	The z-coordinate of the d vector.

NIKE3D Material Type 3: Elastic-Plastic

NIKE3D material type 3 models piecewise linear hardening elastic-plastic behavior. Input consists of values for Young's modulus, Poisson's ratio, and the yield stress. Piecewise linear hardening is accomplished through the **es** and **eps** parameters. If the number of points in the yield stress-effective plastic strain curve or the yield stress-pressure curve (determined by the **npts** parameter) is non-zero, then piecewise linear hardening is performed. If **npts** is zero, then bilinear hardening is performed using the yield stress and hardening modulus supplied through the **sigy** and **etan** parameters. Isotropic, kinematic, or a combination of the kinematic hardening may be specified by varying the β parameter between 1 and 0.

Command	Description
<u>mat - Options:</u>	
e E	Young's modulus
pr ν	Poisson's ratio
sigy σ_y	Yield stress
etan E_t	Hardening modulus
beta β	Hardening parameter
npts n	The number of points in the yield stress-effective plastic strain curve or yield stress-pressure curve.
es $\sigma_1 \sigma_2 \sigma_3 \dots \sigma_n$	Yield stresses
eps $\epsilon_{p1} \epsilon_{p2} \epsilon_{p3} \dots \epsilon_{pn}$	Effective plastic strains

NIKE3D Material Type 4: Thermo-Elastic-Plastic

NIKE3D material type 4 simulates thermo-elastic-plastic behavior. The behavior of the material type is influenced by temperature. The material properties may change as a function of temperature as the analysis evolves. Therefore, an array of values for E , ν , σ_y , E_t and β at various temperatures is defined. In addition, a coefficient of thermal expansion, α , at each temperature level must be specified. The number of points in which all of these quantities are specified is given by **npts**.

It is important to note that at least two temperatures and their corresponding material properties be specified in the input to the material type. If the material temperature exceeds the range defined within the input or if less than two temperatures are specified, the analysis will be terminated by NIKE3D. It is also important to note that the coefficient of thermal expansion is defined with respect to the reference temperature at the beginning of the calculation for that material.

Command	Description
<u>mat - Options:</u>	
npts n	The number of temperature values for which material constants are defined.
temp $T_1 T_2 T_3 \dots T_n$	Temperatures
e $E_1 E_2 E_3 \dots E_n$	Young's modulus for each of the temperatures
pr $\nu_1 \nu_2 \nu_3 \dots \nu_n$	Poisson's ratio for each of the temperatures
alpha $\alpha_1 \alpha_2 \alpha_3 \dots \alpha_n$	Coefficients of thermal expansion
sigy $\sigma_{y1} \sigma_{y2} \sigma_{y3} \dots \sigma_{yn}$	Yield stresses for each of the temperatures
etan $E_{t1} E_{t2} E_{t3} \dots E_{tn}$	Tangent modulus for each of the temperatures

NIKE3D Material Type 5: Soil and Crushable Foam

Input to the soil and crushable foam material type consists of a shear and bulk unloading modulus, three yield function constants, and a pressure cutoff (**pc**) for tensile fracture. Volumetric crushing may be controlled using the unloading option.

In addition to the constants noted above, a series of values must be included which define the volumetric strain vs. pressure curve for the model. Input to that curve consists of not more than 10 pairs of coefficients where pressure is positive in compression and volumetric strain is negative in compression. Volumetric strain is given by the natural log of the relative volume. The pairs of coefficients should be tabulated in order of increasing compression. In the analysis, if the pressure drops below the cutoff value, it is reset to the pressure cutoff value.

Command	Description
<u>mat - Options:</u>	
g G	Shear modulus
ku K_u	Bulk unloading modulus
a0 a_0	Yield function constant, a_0
a1 a_1	Yield function constant, a_1
a2 a_2	Yield function constant, a_2
pc p_c	Pressure cutoff for tensile fracture
ul u_{opt}	Unloading option (0.0 or 1.0)
npts n	The number of points in the volumetric strain vs. pressure curve (n must be less than or equal to 10)
vs $\epsilon_{v1} \epsilon_{v2} \epsilon_{v3} \dots \epsilon_{vn}$	Volumetric strain values
p $p_1 p_2 p_3 \dots p_n$	Corresponding pressure values for each of the volumetric strain constants

NIKE3D Material Type 6: Viscoelastic

Input for this material includes the elastic bulk modulus, two shear moduli, and a decay constant.

Command	Description
<u>mat - Options:</u>	
k K	Elastic bulk modulus
g0 G_0	Short-time shear modulus
gi $G_{infinity}$	Long-time shear modulus
beta β	Decay constant

NIKE3D Material Type 7: Thermo-Orthotropic Elastic

Input is similar to that of NIKE3D material type 2. For further information on the parameters for the constitutive matrix and orienting the local material axis using the **aopt** parameter, consult the discussion on material type 2.

Command	Description
<u>mat - Options:</u>	
ea E_a	Elastic modulus, E_a
eb E_b	Elastic modulus, E_b
ec E_c	Elastic modulus, E_c
prba ν_{ba}	Poisson's ratio, ν_{ba}
prca ν_{ca}	Poisson's ratio, ν_{ca}
prcb ν_{cb}	Poisson's ratio, ν_{cb}
gab G_{ab}	Shear modulus, G_{ab}
gbc G_{bc}	Shear modulus, G_{bc}
gca G_{ca}	Shear modulus, G_{ca}
alpa α_a	Coefficient of thermal expansion along material axis a

Command	Description
<u>mat - Options:</u>	
alpb α_b	Coefficient of thermal expansion along material axis b
alpc α_c	Coefficient of thermal expansion along material axis c
aopt a_{opt}	Specifies the material axis option, a_{opt} (0.0, 1.0, 2.0 or 3.0)
<i>The following information must be supplied when aopt is set to 1.0:</i>	
xp x_p	The x-coordinate of the point in space.
yp y_p	The y-coordinate of the point in space.
zp z_p	The z-coordinate of the point in space.
<i>The following information must be supplied when aopt is set to 2.0:</i>	
a1 a_x	The x-coordinate of the a vector.
a2 a_y	The y-coordinate of the a vector.
a3 a_z	The z-coordinate of the a vector.
d1 d_x	The x-coordinate of the d vector.
d2 d_y	The y-coordinate of the d vector.
d3 d_z	The z-coordinate of the d vector.

NIKE3D Material Type 8: Thermo-Elastic-Creep

NIKE3D material type 8 simulates thermo-elastic-creep.

Command	Description
<u>mat - Options:</u>	
npts n	The number of temperature values for which material constants are defined.
temp $T_1 T_2 T_3 \dots T_n$	Temperatures
g $G_1 G_2 G_3 \dots G_n$	Young's modulus for each of the temperatures
k $\nu_1 \nu_2 \nu_3 \dots \nu_n$	Poisson's ratio for each of the temperatures
alpha $\alpha_1 \alpha_2 \alpha_3 \dots \alpha_n$	Coefficients of thermal expansion
a $a_1 a_2 a_3 \dots a_n$	Creep parameters
b $b_1 b_2 b_3 \dots b_n$	Creep parameters

TOPAZ3D Materials

This section details INGRID commands which only affect material controls for both the TOPAZ3D materials and the TOPAZ3D electrostatic materials. Further information on these material types can be found by referring to the TOPAZ3D user's guide.

TOPAZ3D Material Type 1: Isotropic

Command	Description
mat n options endmat	Material models for a specific code can be input with this command.
<u>mat - Options:</u>	
ro ρ	Density
tlha t	Temperature at which latent heat is absorbed or released
lh h	Latent heat
tgc lc	Thermal generation rate curve number
tgm r	Thermal generation rate multiplier
cp c	Heat capacity
k k	Thermal conductivity

TOPAZ3D Material Type 2: Orthotropic

Command	Description
<u>mat - Options:</u>	
ro ρ	Density
tlha t	Temperature at which latent heat is absorbed or released
lh h	Latent heat
tgc lc	Thermal generation rate curve number
tgm r	Thermal generation rate multiplier
cp c	Heat capacity
k1 k_1	Thermal conductivity in local 1 direction
k2 k_2	Thermal conductivity in local 2 direction
k3 k_3	Thermal conductivity in local 3 direction

TOPAZ3D Material Type 3: Isotropic Temperature Dependent

Command	Description
<u>mat - Options:</u>	
ro ρ	Density
tlha t	Temperature at which latent heat is absorbed or released
lh h	Latent heat
tgc lc	Thermal generation rate curve number
tgm r	Thermal generation rate multiplier
npts n	Number of temperature points
t $T_1 T_2 T_3 \dots T_n$	Temperatures
cp $c_1 c_2 c_3 \dots c_n$	Heat capacity at each temperature point
k $k_1 k_2 k_3 \dots k_n$	Thermal conductivity at each temperature point

TOPAZ3D Material Type 4: Orthotropic Temperature Dependent

Command	Description
<u>mat - Options:</u>	
ro ρ	Density
tlha t	Temperature at which latent heat is absorbed or released
lh h	Latent heat
tgc lc	Thermal generation rate curve number
tgm r	Thermal generation rate multiplier
npts n	Number of temperature points
t $T_1 T_2 T_3 \dots T_n$	Temperatures
cp $c_1 c_2 c_3 \dots c_n$	Heat capacity at each temperature point
k1 $k_{11} k_{12} k_{13} \dots k_{1n}$	Thermal conductivities in the local 1 direction
k2 $k_{21} k_{22} k_{23} \dots k_{2n}$	Thermal conductivities in the local 2 direction
k3 $k_{31} k_{32} k_{33} \dots k_{3n}$	Thermal conductivities in the local 3 direction

TOPAZ3D Electrostatic Material Type 1: Isotropic

Command	Description
<u>mat - Options:</u>	
e e	Electric permittivity
ro ρ	Volume charge density

ADINA Materials

This section details the material specification commands for ADINA.

ADINA Material Type 1: Isotropic Linear Elastic

Command	Description
mat n options endmat	Material models for a specific code can be input with this command.
<u>mat - Options:</u>	
e E	Young's modulus
pr v	Poisson's ratio

ADINA Material Type 9: Elastic Plastic

Command	Description
<u>mat - Options:</u>	
e E	Young's modulus
pr v	Poisson's ratio
sigy σ_y	Yield stress
etan E_t	Hardening modulus

ADINA Material Type 10: Thermo-Elastic Plastic

Command	Description
<u>mat - Options:</u>	
npts n	Number of temperature points
temp $T_1 T_2 T_3 \dots T_n$	Temperatures
e $E_1 E_2 E_3 \dots E_n$	Young's modulus at each temperature point
pr $\nu_1 \nu_2 \nu_3 \dots \nu_n$	Poisson's ratio at each of temperature point
alpha $\alpha_1 \alpha_2 \alpha_3 \dots \alpha_n$	Coefficients of thermal expansion
sigy $\sigma_{y1} \sigma_{y2} \sigma_{y3} \dots \sigma_{yn}$	Yield stresses at each temperature point
etan $E_{t1} E_{t2} E_{t3} \dots E_{tn}$	Hardening modulus at each temperature point
creep $C_1 C_2 C_3 \dots C_n$	Creep law constants for each temperature
tref T_{ref}	Reference temperature
xkcrp C_{law}	Creep law
tint t	Integration parameter
xisubm SS_{max}	Maximum number of step subdivisions
xnite IS_{max}	Maximum number of iterations for each subdivision
xnalg A_{num}	Algorithm number
tolil σ_{tol}	Stress convergence tolerance
tolpc IS_{tol}	Inelastic tolerance

Verbatim Materials

The “verbatim” material model enables the use of new materials not currently supported in INGRID. A material type that is greater than or equal to 90000 signals that the accompanying material specification is for a verbatim material. With this material model, the global material commands can be used. These commands should be followed by the command: **verb *n***

verb must be followed by *n* lines of text. In the output file, INGRID writes the first two lines of the material definition in the normal way, followed by the *n* lines of text that were provided after the **verb** command. The example below demonstrates the use of the verbatim material model.

```
mat 1 90000
ro 7.1e-4 c Set the density of the material
verb 6
1.170E+00 0.000E+00 0.000E+00 . . . 0.000E+00 9.000E+00
3.300E-01 0.000E+00 0.000E+00 . . . 0.000E+00 7.000E+00
4.000E-03 0.000E+00 0.000E+00 . . . 0.000E+00 5.000E+00
1.000E-03 0.000E+00 0.000E+00 . . . 0.000E+00 3.000E+00
1.000E+00 0.000E+00 0.000E+00 . . . 0.000E+00 7.000E+00
0.000E+00 0.000E+00 0.000E+00 . . . 0.000E+00 1.000E+00
endmat
```

For material numbers greater 90000, the difference between the material type and 90000 will be used as actual material number in the INGRID output file. For example, if 90003 is used as the verbatim material type, then the material type written to the INGRID output file will be 3.

4.2.10 Part Tolerancing

Part tolerancing in INGRID consists of the process whereby multiple nodal points sharing common spatial coordinates are reduced to single nodes. INGRID provides commands for both surface and volumetric part tolerancing, with associated commands for controlling the tolerancing process on a part by part basis. Nodes which are identified for the merging process are compared to all other nodes whose coordinates fall within the specified tolerance before the actual nodal merging process is performed. The **stp**, **tp** and **tol** commands may be placed in the INGRID input file to automatically invoke part tolerancing when INGRID enters its interactive phase. Similarly, the part tolerancing commands, **stol**, **stp**, **tol**, **tp** and **ztol** may be used in the interactive phase. Currently there is a limit of 300 parts for tolerancing, with a maximum of 30 parts using special tolerancing (**bptol** or **ptol**).

Part Tolerancing

Command	Description
bptol <i>n m</i> ϵ	Specify a between-part tolerance of ϵ to be used in redundant node suppression between parts <i>n</i> and <i>m</i> .
ptol <i>n</i> ϵ	Specify a tolerance of ϵ to be used in part # <i>n</i> .
stol ϵ	Specify a tolerance of ϵ to be used in surface tolerancing. With this command, only nodes on the surfaces of parts are considered candidates for the nodal merging process. stol produces a summary of merged nodes.
stp ϵ	Specify a tolerance of ϵ to be used in surface tolerancing. With this command, only nodes on the surfaces of parts are considered candidates for the nodal merging process. stp is identical to stol , but does its work silently.
tol ϵ	Specify a tolerance of ϵ to be used in volumetric tolerancing. With this command, all nodes of parts are considered candidates for the nodal merging process. tol produces a summary of merged nodes.
tp ϵ	Specify a tolerance of ϵ to be used in volumetric tolerancing. With this command, all nodes of parts are considered candidates for the nodal merging process. tp is identical to tol , but does its work silently.
ztol ϵ	Specify a minimum absolute value for coordinates.

4.2.11 Slide Surface Specification

Slide surfaces are INGRID entities which must be defined both globally and on the part level to completely define sliding interfaces in the finite element model. This section describes the commands for defining global slide surfaces in INGRID.

Slide Surface Specification

Command	Description
si <i>n options</i> ;	Input slide surface control data for slide surface number <i>n</i> with <i>options</i> .
<u>Options - si:</u>	
sl	Sliding surface only.
sv	Sliding surface with voids (default option).
tied	Tied slide surface.
single	Single sided slide surface.
dni	Discrete nodes impacting a surface.
dni	Discrete nodes tied.
sets	Shell edge tied to shell.
nsw	Nodes spot welded.
break	Tiebreak interface.
ld σ_y	1-D slide line for beam part. σ_y specifies the axial yield stress. This slide surface option is for use with NIKE3D and beam parts only.
pnlt λ	Change penalty value to λ (default = 0).
pnltm λ	Change master penalty value to λ (default = 0).
pnlts λ	Change slave penalty value to λ (default = 0).
fric μ	Change friction coefficient to μ (default = 0).
kfric μ	Change kinetic friction coefficient to μ (default = 0).
delay K	Change exponential decay coefficient to K (default = 0).

Slide Surface Specification (cont.)

Command	Description
<u>Options - si:</u>	
decay K	Change exponential decay coefficient to K (default = 0).
tcrs R	Set thermal contact resistance to R .
dummy	Dummy slide surface. This option can be used to give separate degrees-of-freedom to coincident nodes.
merge	Coincident nodes on opposite sides of slide surfaces are merged.
nomerge	Coincident nodes on opposite sides of slide surfaces are not merged.
mfif	Include master side data in printed interface force file.
sfif	Include slave side data in printed interface force file.
pen	Flag small penetration.
sand data	Specifies a SAND (Slide-surface with Adaptive New Definitions) sliding interface. The following <i>data</i> is required to completely specify the SAND sliding interface.
<u>sand - Required Data:</u>	
svm $n m_1 \dots m_n$	Specifies that the current slave slide surface contains n materials from m_1 to m_n .
srm $n m_1 \dots m_n$	Specifies that the current master slide surface contains n materials from m_1 to m_n .

4.2.12 Surface Definitions

INGRID provides the capability to define surfaces (both globally and on the part level) which can be used with INGRID's projection algorithms to form complicated parts. The commands for surface definitions include both discrete and analytic surfaces. In this section, the commands for both discrete and analytic surfaces are presented.

Surface Definitions

Discrete 3-D Surface Definitions

Command	Description
ds <i>n options</i> ;	Define a general 3-D surface for input to INGRID.
<u>ds - Options:</u>	
grid <i>n m data</i>	The surface is defined by a topologically regular set of points in three dimensions. $n*m$ points must be input in the following order immediately following the grid option.
<u>grid - Required data:</u>	
<i>x₁₁ y₁₁ z₁₁</i>	
<i>x₂₁ y₂₁ z₂₁</i>	
<i>x₃₁ y₃₁ z₃₁</i>	
...	
<i>x_{nm} y_{nm} z_{nm}</i>	

Surface Definitions (cont.)

Discrete 3-D Surface Definitions (cont.)

Command	Description
cont nc	The surface is defined by nc contours that each have an arbitrary number of points. The following options may immediately follow the cont command.
<u>cont - Options:</u>	
xle x	Specify x-coordinate of leading edge.
yle y	Specify y-coordinate of leading edge.
zle z	Specify z-coordinate of leading edge.
chord s	Scale factor for the chord length.
fnu n_u fnl n_l	The contour is defined by n_u points on the upper surface followed by n_l points on the lower surface.
xf x	All contour points have the value x.
yf y	All contour points have the value y.
zf z	All contour points have the value z.
n	Number of points on the contour. This is a required variable for the cont option. The discrete contour data should be input as follows:
<u>cont - Required data:</u>	
$x_{11} y_{11} z_{11}$	
$x_{21} y_{21} z_{21}$	
...	
$x_{nn} y_{nn} z_{nn}$	
	If xf was specified, then only the y and z coordinates must be input. Similarly if yf was specified, then only the x and z coordinates must be input, etc.
mkds	Make a binary database of the discrete 3-D surfaces.
rdds	Read a binary database of the digitized 3-D surfaces.

Surface Definitions (cont.)

Analytic 3-D Surface Definitions

Command	Description
sd n options	Surface definition number n. If surface n has been previously defined, this command has the effect of destroying the old definitions.
<u>sd - Options:</u>	
ac $p_x p_y p_z q_x q_y q_z n a_0 a_1 \dots a_n$	Defines a surface by rotating a cosine polynomial about an axis. (p_x, p_y, p_z) is a point defining the location of the curve, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of rotation, n is the number of terms in the polynomial, and a_0, a_1, \dots, a_n are the n coefficients for the cosine series. The surface radius is computed as: $r = a_0 + a_1 \cos^2(\theta) + \dots + a_n \cos^n(\theta)$. For this surface type, n must be less than or equal to 12.
cn2p $p_x p_y p_z q_x q_y q_z r_1 z_1 r_2 z_2$	Define a conical surface using two points. (p_x, p_y, p_z) is point on the axis of the cone, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the cone. (r_1, z_1) and (r_2, z_2) are radial and axial coordinates of two points on the conical surface.
cone $p_x p_y p_z q_x q_y q_z r \theta$	Define a conical surface. (p_x, p_y, p_z) is point on the axis of the cone, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the cone, r is the radius of the cone at the point (p_x, p_y, p_z) . θ is the angle of the cone relative to the vector (q_x, q_y, q_z) , and is specified in degrees.
cp trans n	Form an infinite surface from line definition n. Initially, the line is defined in the r-z plane. After projecting to a standard part, the part will have the same profile as the curve in the x-y plane along the z-axis. Transformations, <i>trans</i> , may be applied to the surface. These transformation are described in section 5.1.1 Geometric Transformations - Part Replication.
cr $p_x p_y p_z q_x q_y q_z n$	Define a surface by rotating a discrete line about an axis. (p_x, p_y, p_z) is point defining the location of the line, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of rotation, n is the number of a line definition which has been defined with the ld command.
crx n	Generate a surface by rotating a line defined in the x-z (or r-z) plane about the x-axis using line definition n .
cry n	Generate a surface by rotating a line defined in the x-z (or r-z) plane about the y-axis using line definition n .
crz n	Generate a surface by rotating a line defined in the x-z (or r-z) plane about the z-axis using line definition n .

Surface Definitions (cont.)

Analytic 3-D Surface Definitions (cont.)

Command	Description
<u>sd - Options (cont.):</u>	
cy $p_x p_y p_z q_x q_y q_z r$	Define a cylindrical surface with radius r . (p_x, p_y, p_z) is point defining the origin of the axis of rotation, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of rotation.
er $p_x p_y p_z q_x q_y q_z r_1 r_2$	Defines an surface generated by rotating an ellipse with center (p_x, p_y, p_z) where (q_x, q_y, q_z) is a vector oriented normal to the axis of revolution. r_1 is the maximum radius in the plane normal to the axis of rotation, and r_2 is the distance from the center point along the axis of rotation to the ellipse.
gs n	Use discrete surface n , where the surface has been defined using the ds command described above.
gsl n	Use the lower (under) side of discrete surface n , where the surface has been defined using the ds command described above.
gsu n	Use the upper side of discrete surface n , where the surface has been defined using the ds command described above.
pl3 $f_1 x_1 y_1 z_1 f_2 x_2 y_2 z_2 f_3 x_3 y_3 z_3 \delta$	Define a plane using three points. The flags f_1, f_2, f_3 specify the type of coordinate system associated with each point. f_1, f_2 and f_3 may take on values of rt for Cartesian, cy for cylindrical or sp for spherical coordinates. δ specifies the offset for the plane along the positive outward surface normal. The sense of the outward normal is defined by the cross product of the vector from point 1 to point 3 with the vector from point 1 to point 2.
plan $p_x p_y p_z q_x q_y q_z$	Define a surface which is a plane. (p_x, p_y, p_z) is point on the plane, and (q_x, q_y, q_z) is a vector normal to the plane.
poly $p_x p_y p_z q_x q_y q_z n a_0 \dots a_n$	Define a surface by revolving a polynomial of degree n about an axis. (p_x, p_y, p_z) is a point defining the origin of the axis of rotation, and (q_x, q_y, q_z) is the vector direction of the axis of rotation. n is the degree of the polynomial. $n+1$ coefficients ($a_0 \dots a_n$) are required.

Surface Definitions (cont.)

Analytic 3-D Surface Definitions (cont.)

Command	Description
---------	-------------

sd - Options (cont.):

pr $p_x p_y p_z n_x n_y n_z r_1 t_1 r_2 t_2 r_3 t_3$

Defines a parabolic surface of revolution where (p_x, p_y, p_z) is a point defining the location of the parabola, (q_x, q_y, q_z) is a vector oriented parallel to the t-axis (Figure 4.2). (r_1, t_1) is the first point on the parabola. If $t_1=0$, then the pint lies on a plane normal to the t-axis at point (p_x, p_y, p_z) . The positive direction on the t-axis is the same as the direction of (q_x, q_y, q_z) . (r_2, t_2) and (r_3, t_3) are the 2nd and 3rd points on the parabola respectively.

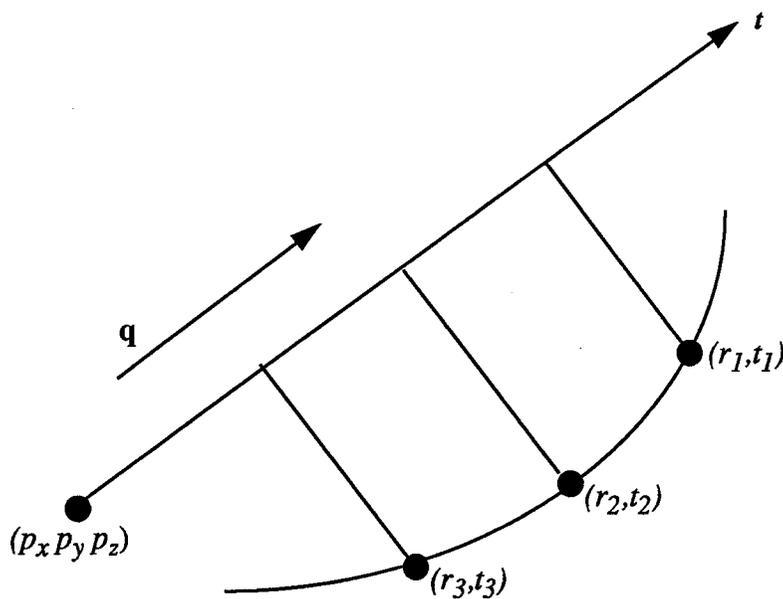


Figure 4.2
pr - Parabola revolved about an axis.

Surface Definitions (cont.)

Analytic 3-D Surface Definitions (cont.)

Command	Description
<u>sd - Options (cont.):</u>	
sp $p_x p_y p_z r$	Defines a spherical surface where (p_x, p_y, p_z) is the center of a sphere with radius r .
ts $p_x p_y p_z q_x q_y q_z r t R$	Define a toroidal surface. (p_x, p_y, p_z) is point on the axis of the torus, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the torus.
ts2p $p_x p_y p_z q_x q_y q_z r_1 z_1 r_2 z_2 R$	Define a toroidal surface using two points. (p_x, p_y, p_z) is point on the axis of the torus, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the torus. (r_1, z_1) and (r_2, z_2) are radial and axial coordinates of two points on the toroidal surface. $ R $ is the radius of the surface from point (p_x, p_y, p_z) . If $R > 0$, then (r_2, z_2) lies to the left as one moves from (r_1, z_1) and (r_2, z_2) , otherwise, (p_x, p_y, p_z) lies to the right.

4.2.13 Volume Definitions

This section documents the primitives available for modelling solid objects. Solid primitives are global INGRID entities, and are generated using the **vd** command. The five primitives are: spheres, cylinders, rectangles, wedges and solids formed from curves rotated about a user defined axis.

Volume Definitions

Command	Description
vd n options	Volume definition number n. If volume n has been previously defined, this command has the effect of destroying the old definition.
<u>vd - Options:</u>	
sphe $p_x p_y p_z R$	Define a spherical volumetric primitive with center at (p_x, p_y, p_z) and radius R .
cyli $p_x p_y p_z q_x q_y q_z R$	Define a cylinder where (p_x, p_y, p_z) is a point on the axis of the cylinder, (q_x, q_y, q_z) is a vector oriented parallel to the axis of the cylinder. The cylinder radius is R .
rect $n x_{min} x_{max} y_{min} y_{max} z_{min} z_{max}$	Define a solid rectangular primitive using global coordinate transformation n to orient the solid. The rectangular solid is defined in terms of the local dimensions by $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$.
tria $n x_1 y_1 x_2 y_2 x_3 y_3 z_{min} z_{max}$	Define a solid wedge shaped primitive using global coordinate transformation n to orient the solid. The solid is defined in terms of a triangular section in the x-y plane. The section is defined by the three points (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . The local dimensions in the z-direction are set z_{min} and z_{max} .
cr $p_x p_y p_z q_x q_y q_z n$	Devine a volumetric primitive generated by a curve rotated about an axis. (p_x, p_y, p_z) is a point on the axis of the solid, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the solid. n refers to a global line definition.

5.0 MESH & MODEL GENERATION COMMANDS - II

This chapter presents the INGRID commands used to describe the physical geometry, degree of discretization, material data, boundary conditions and analysis options beginning with the standard part commands. MAZE, beam and import part commands are then described. The final section of this chapter details interactive commands used to graphically interrogate and validate a model.

Command Index:

Standard Part Definition	161
start define a new standard part	161
end Required end statement for part-end definition of standard part	161
Stab Part Definition	162
stab Define a new stab part	162
end Required end statement for part-end definition of stab part	162
Standard Part: General Control Commands.....	165
cyli Convert from cylindrical to Cartesian coordinates	165
d Delete <region> from the index space and the part.....	165
di Delete <r-index> from the index space and the part.	165
epb Setup an element print block	165
po Use the element offset	165
ro Use the block of elements in <region> offset from <point>	165
mate Specify material number.....	165
mt Specify material number.....	165
mti Specify material number.....	165
mtv All elements contained in the volume m are assigned a material number n....	165
n+ Specify the positive outward shell normals	165
n- Specify the positive outward shell normals	165
npb Setup a nodal print block	165
or Specify the orientation of the local axes.....	166
print Specify that node numbers be printed in the INGRID save file	166
sphe Nodal point coordinates are converted from spherical to Cartesian	166
temp Specify an initial temperature.....	166

Command Index: (cont.)

thic	Specify shell element thickness	166
th	Specify a shell thickness	166
thi	Specify a shell thickness	166
thth	Specify the thickness variation of a thin shell	166
tn	Specify the shell thickness number.....	166
tni	Specify a shell thickness number.....	166
ve	Activate the edge visibility for outline and phantom edge plotting.....	166
Standard Part: Geometric Transformations - Projections		167
a	Project onto an arc	168
a	Project onto a cylindrical surface.....	169
ac	Project onto a cylindrical surface.....	170
ace	Project onto a cylindrical surface.....	170
ae	Project onto a cylindrical surface.....	170
ae	Project onto an arc	170
sf	Project onto a surface.....	171
sfi	Project onto a surface.....	171
Standard Part: Surface Projections		171
ac	Defines a surface by rotating a cosine polynomial about an axis	171
cn2p	Define a conical surface using two points	171
cone	Define a conical surface.....	171
cp	Form an infinite surface from line definition n	172
cr	Define a surface by rotating a discrete line about an axis.....	172
crx	Generate a surface by rotating a line defined in the x-z (or r-z) plane.....	172
cry	Generate a surface by rotating a line defined in the x-z (or r-z) plane.....	172
crz	Generate a surface by rotating a line defined in the x-z (or r-z) plane	172
cy	Define a cylindrical surface with radius	172
er	Defines a surface generated by rotating an ellipse	172
gs	Use discrete surface n	172
gsl	Use the lower (under) side of discrete surface n.....	172
gsu	Use the upper side of discrete surface n	172
pl3	Define a plane using three points	173

Command Index: (cont.)

plan	Define a surface which is a plane.....	173
poly	Define a surface by revolving a polynomial of degree n about an axis.....	173
pr	Define a parabolic surface of revolution.....	173
sp	Define a spherical surface.....	173
ts	Define a toroidal surface.....	173
ts2p	Define a toroidal surface using two points.....	173
Standard Part: Multiple Surface Projections.....		174
ms	Apply multiple surface projections.....	174
Standard Part: Geometric Transformations - Point/Line Modifications.....		175
cpl	Center points along a line.....	175
i	Coordinate modifying function i-index.....	175
j	Coordinate modifying function j-index.....	175
k	Coordinate modifying function k-index.....	176
int	Moves nodes to the nearest point on the intersection of two surfaces.....	176
ma	Modify 1, 2, or 3 coordinates in index space.....	176
mb	Modify 1, 2, or 3 coordinates in index space.....	176
pa	Modify 1, 2, or 3 coordinates in index space.....	177
pb	Modify 1, 2, or 3 coordinates in index space.....	177
res	Rezone elements.....	177
rr	Apply coordinate transformations.....	177
lct	Define a sequence of coordinate transformations.....	178
lrep	Perform local part replication.....	178
repe	Makes copies of the part in each of the global coordinate systems.....	178
jt	Generate a joint.....	179
p	The local joint node, n, is at the point (px, py, pz).....	179
inc	Increment the joint number.....	179
b	The local joint node has boundary constraint n.....	179
Standard Part: Electromagnetic Loads and Boundary Conditions.....		181
em	Specify boundary conditions for MAXWELL3.....	181
emi	Specify boundary conditions for MAXWELL3.....	181

Command Index: (cont.)

fbc	Specify an electromagnetic boundary condition	181
fbci	Specify an electromagnetic boundary condition	182
per	Specify periodic boundary conditions for MAXWELL3	182
spc	Specify a current	182
spci	Specify a current	182
spf	Specify a field	182
spfi	Specify a field	182
eff	Specify an electrostatic flux	182
effi	Specify an electrostatic flux	182
v	Specify an electrostatic potential	183
vi	Specify an electrostatic potential	183
Standard Part: Sliding Interface Commands.....		183
si	Specify that <region> should be a slide surface	183
si+	Specify that <region> should be a slide surface	183
si-	Specify that <region> should be a slide surface	184
sii	Specify that <r-index> should be a slide surface	184
sii+	Specify that <r-index> should be a slide surface	185
sii-	Specify that <r-index> should be a slide surface	185
Standard Part: Solid Mechanics Boundary Conditions.....		186
acc	Specify acceleration	186
acc	Specify acceleration	186
accs	Specify acceleration	186
acci	Specify acceleration	186
b	Specify displacement boundary conditions	186
fc	Specify fixed forces	186
fcc	Specify fixed forces in cylindrical coordinates.....	186
fcs	Specify fixed forces in spherical coordinates	186
fd	Specify fixed displacements.....	186
fdc	Specify fixed displacements in cylindrical coordinates.....	187
fds	Specify fixed displacements in spherical coordinates.....	187
fv	Specify a fixed velocity.....	187

Command Index: (cont.)

fvc	Specify a fixed velocity in cylindrical coordinates.....	187
fvs	Specify a fixed velocity in spherical coordinates.....	187
ndl	Specify a distributed load	187
nr	Specify a non-reflecting boundary	187
nri	Specify a non-reflecting boundary	187
pm	Specify a region with discrete masses	187
pr	Specify pressure boundary conditions	187
pri	Specify application of a pressure load	188
prth	Specify a thin shell pressure variation	188
rotation	Assign an initial rigid body rotation	188
spdb	Identify nodes within a standard part as belonging to a spring-damper system	188
sw	Identifies a stone wall	188
syf	Specify a symmetry plane with failure.	188
syfi	Define a symmetry plane with failure.....	188
te	Specify that <region> have temperature T	188
tei	Specify that <r-index> have temperature T	188
velocity	Assign the rigid body velocity	188
veth	Specify a thin shell velocity variation.....	188
Standard Part: Thermal Loads and Boundary Conditions		189
cv	Specify convection boundary conditions on <region>	189
cv+	Specify convection boundary conditions on <region>	189
cv-	Specify convection boundary conditions on <region>	189
cvi	Specify convection boundary conditions on <r-index>	189
fl	Specify a heat flux boundary condition on <region>.....	189
fli	Specify a heat flux boundary condition on <r-index>	189
ft	Specify a fixed temperature boundary condition on <region>	189
fti	Specify a fixed temperature boundary condition on <r-index>	189
Radiation Boundary Conditions.....		190
rb	Specify radiation boundary conditions on <region>.....	190
rb+	Specify radiation boundary conditions on <region>.....	190
rb-	Specify radiation boundary conditions on <region>.....	190

Command Index: (cont.)

rbi	Specify radiation boundary conditions on <r-index>	190
rbc	Specify radiation boundary conditions on <region>.....	190
rbc	Specify radiation boundary conditions on <r-index>	190
Radiation Enclosures		190
re	Identify <region> as part of a radiation enclosure	190
re+	Identify <region> as part of a radiation enclosure	191
re-	Identify <region> as part of a radiation enclosure	191
rei	Identify <r-index> as part of a radiation enclosure.....	191
tm	Specify an initial temperature condition (T) on <region>	191
vhg	Specify volumetric heat generating elements in <region>	191
vhgi	Specify volumetric heat generating elements in <r-index>	191
Standard Part: Beam Element Generation		192
ibm	Generate beam elements <region> in the j-k plane	192
ibmi	Generate beam elements in <r-index> in the j-k plane.....	192
jbm	Generate beam elements in <region> in the i-k plane	192
jbmi	Generate beam elements in <r-index> in the i-k plane.....	192
kbm	Generate beam elements in <region> in the i-j plane	192
kbmi	Generate beam elements in < index> in the i-j plane	192
Edit and the In-line Calculator.....		193
edit	Begin the edit phase.....	194
cart	Convert the current part coordinates to Cartesian coordinates	194
cyli	Convert the current part coordinates to cylindrical coordinates	194
sphe	Convert the current part coordinates to spherical coordinates.....	194
xcyli	Convert to cylindrical coordinates, where the x-axis is an axis of symmetry ..	194
ycyli	Convert to cylindrical coordinates, where the y-axis is an axis of symmetry ..	194
zcyli	Convert to cylindrical coordinates, where the z-axis is an axis of symmetry ..	194
xsphe	Convert to spherical coordinates, where the x-axis is an axis of symmetry	194
ysphe	Convert to spherical coordinates, where the y-axis is an axis of symmetry	194
zsphe	Convert to spherical coordinates, where the z-axis is an axis of symmetry	194
fir	Specify that editing should occur on the full index	194
phr	Restrict the scope of any following expressions	194

Command Index: (cont.)

rir	Specify that editing should occur on a reduced index space	194
rand	Produces random perturbations to the mesh	195
norm	Cause normal perturbations to the mesh.....	195
MAZE Part Definition		196
part	Define a new MAZE part.....	197
end	Required end statement for part-end definition of MAZE part	197
MAZE Cross Section Definition.....		198
MAZE Part Control		199
drag	Perform a dragging operation to generate solid elements.....	199
move	Form n layers of solid elements	199
res	The ratio of one element length to the next adjacent element length is r	199
rota	Form n layers of solid elements by rotating the original shell elements	199
epb	Setup an element print block	199
po	Use the element offset from <point> by (hick).....	199
ro	Use the block of elements in <region> offset from <point>	199
lct	Define a sequence of n local coordinate transformations	200
lrep	Performs local part replication.....	200
mate	The part has material number n	200
repe	Make copies of the part in each of the global coordinate	200
rotation	Assign an initial rigid body rotation to the part.....	200
sphe	Nodal point coordinates are converted from spherical to Cartesian.....	200
spin	Generate 3-D solid elements by performing a spin operation	201
stack	Generate 3-D solid elements by performing a stacking operation.....	201
thick	Shells have a thickness of t for this part	201
temp	The initial temperature is T.....	201
velocity	Assign a rigid body velocity	201
t12	Element transition is achieved with quadrilateral elements.....	201
t13	Element transition is achieved with quadrilateral elements.....	201
MAZE Part: Electromagnetic Loads and Boundary Conditions		202
fbc	Specify an electromagnetic boundary condition.....	202

Command Index: (cont.)

spc	Specify a current on <region>	202
spf	Specify a field on <region>.....	202
efl	Specify an electrostatic flux, f, on <region>.....	203
v	Specify an electrostatic potential, f, on <region>	203
MAZE Part: Sliding Interface Definitions		203
si	Specify that <region> should be a slide surface	203
si+	Specify that <region> should be a slide surface	203
si-	Specify that <region> should be a slide surface	204
MAZE Part: Solid Mechanics Boundary Conditions		204
acc	Specify acceleration on <region>	204
accc	Specify acceleration on <region> in cylindrical coordinates.....	204
accs	Specify acceleration on <region> in spherical coordinates	204
b	Specify displacement boundary conditions on <region>.....	204
fc	Specify fixed forces on <region>.....	205
fd	Specify fixed displacements on <region>	205
fv	Specify a fixed velocity on <region>	205
pr	Specify pressure boundary conditions on <region>	205
sw	Identifies <region> as a stone wall	205
MAZE Part: Thermal Loads and Boundary Conditions		205
cv	Specify convection boundary conditions on <region>	205
cv+	Specify convection boundary conditions on <region>	205
cv-	Specify convection boundary conditions on <region>	206
fl	Specify a heat flux boundary condition on <region>.....	206
ft	Specify a fixed temperature boundary condition on <region>	206
rb	Specify radiation boundary conditions on <region>.....	206
rb+	Specify radiation boundary conditions on <region>.....	206
rb-	Specify radiation boundary conditions on <region>.....	206
rbc	Specify radiation boundary conditions on <region>.....	206
re	Identify <region> as part of a radiation enclosure.....	206
re+	Identify <region> as part of a radiation enclosure.....	207

Command Index: (cont.)

re-	Identify <region> as part of a radiation enclosure.....	207
tm	Specify an initial temperature condition (T) on <region>.....	207
Beam Part Definition		208
beam	Define a new beam part	208
end	Required end statement for beam-end definition of beam part.....	208
Beam Part: Coordinate Specification.....		210
ct	Acceptable values for ct are rt, cy, or sp.....	210
Beam Part: Element Generation Commands		210
n1	First beam index to be used in the current beam part.....	210
n2	Last beam index to be used in the current beam part.....	210
N	Number of beam elements to generate	210
material	Material number for the beam elements	210
section	Section material number for the beam elements.....	210
n3	Reference node for defining the orientation of the beam elements	210
Beam Part: Options.....		211
cyli	Nodes are converted from cylindrical to Cartesian	211
rotation	Assign an initial rigid body rotation	211
sphe	Nodes are converted from spherical to Cartesian coordinates.....	211
temp	The initial temperature is T.....	211
velocity	Assign the rigid body velocity	211
Beam Part: Element Attributes		211
bsd	Specify a beam section definition number n.....	211
Beam Part: Joint Definition.....		211
jt	Joint definition	211
lrep	Makes copies of the part in the local coordinate system	211
repe	Makes copies of the part in the global coordinate system	211
Beam Part: Slide Surface Specification		212
si	Specify that the beam elements are to be associated with slide line n.....	212

Command Index: (cont.)

Import Part Definition	213
import Define a new import part	213
end Required end statement for import-end definition of import part.....	213
Import Part Commands	213
nodes nodal points are to be read in	214
form Specify that nodal points are to be read with a user defined format.....	214
number Node numbers are to be read in	214
k Node increments are to be read.....	214
bcnd NIKE3D displacement boundary codes are to be read.....	214
bcnr NIKE3D rotational boundary codes are to be read.....	214
bcsp SAP boundary codes are to be read	214
x X-coordinate is to be read.....	214
y Y-coordinate is to be read	214
z Z-coordinate is to be read	214
t Temperature is to be read.....	214
dummy Read, but ignore a given numeric field.....	214
beams beam elements are to be read	215
form Specify that beam elements are to be read with a user defined format.....	215
number Element numbers are to be read.....	215
k Element increment values are to be read	215
mate Material numbers are to be read	215
section Section material numbers are to be read.....	215
nodes Three node numbers which define the local coordinates (r,s,t) are to be read .	215
n1 Read node n1	215
n2 Read node n2	215
n3 Read node n3	215
dummy Read, but ignore a given numeric field.....	215
shells shell elements are to be read	216
form Specify that shell elements are to be read with a user defined format.....	216
number Element numbers are to be read.....	216
k Element increment values are to be read	216

Command Index: (cont.)

mate	Material numbers are to be read	216
thickness	Shell element thicknesses are to be read.....	216
nodes	Four node numbers which define the shell orientation are to be read	216
n1	Read node n1	216
n2	Read node n2	216
n3	Read node n3	216
n4	Read node n4	216
dummy	Read, but ignore a given numeric field.....	216
bricks	8-node brick elements are to be read	217
form	Specify that solid elements are to be read with a user defined format.....	217
number	Element numbers are to be read.....	217
k	Element increment values are to be read	217
mate	Material numbers are to be read	217
nodes	Eight node numbers which define the brick orientation are to be read.....	217
n1	Read node n1	217
n2	Read node n2	217
n3	Read node n3	217
n4	Read node n4	217
n5	Read node n5	217
n6	Read node n6	217
n7	Read node n7	217
n8	Read node n8	217
dummy	Read, but ignore a given numeric field.....	217
Import Part: Control Commands.....		218
cyli	Nodal point coordinates are converted from cylindrical to Cartesian	218
lrep	Makes copies of the part in each of the local coordinate system.....	218
mate	The imported part has material number n.....	218
repe	Makes copies of the part in each of the global coordinate system	218
rotation	Assign an initial rigid body rotation to the part.....	218
sphe	Nodal point coordinates are converted from spherical to Cartesian.....	218
thic	Shells have a thickness of t for this part	218

Command Index: (cont.)

temp	The initial temperature is T.....	218
velocity	Assign a rigid body velocity	218
Interactive Commands: Display Commands		219
dioff	Turn off the diagnostic node highlighting.....	219
disize	Scale the glyphs used to identify slide surface, normals, etc.....	219
disp	Generates an image of the mesh using the best z-buffer algorithm.....	219
draw	Displays the mesh with all element edges drawn	219
grid	Activate a reference grid which is displayed with the mesh.....	219
gridoff	De-activate the reference grid.....	219
help	This command prints out terse help information	219
pause	Pause for approximately n seconds.....	219
poor	Displays the mesh using back face culling on a part-by-part basis	220
reso	Specify the z-buffer resolution for the disp command.....	220
set	Set the defaults for the following commands	220
tv	display, draw, none, poor or view	220
grid	Set the grid color.....	220
marker	Set the mesh marker color	220
mesh	Set the mesh color.....	220
rax	Set the color used to display the r-axis	220
sax	Set the color used to display the s-axis.....	220
tax	Set the color used to display the t-axis	220
tv	Specify the graphics device driver to be used.....	220
view	Generates an image of the mesh using a crude z-buffer algorithm.....	221
Interactive Commands: Rotation Commands		221
rx	The picture is rotated about the x-axis.....	221
ry	The picture is rotated about the y-axis.....	221
rz	The picture is rotated about the z-axis	221
fix	Fixes the current center of rotation	221
unfix	Releases the fixed center of rotation established by fix	221

Command Index: (cont.)

Interactive Commands: Translation Commands.....	221
u The picture is shifted up.....	221
d The picture is shifted down.....	221
l The picture is shifted left	221
r The picture is shifted right	221
Interactive Commands: Zooming and Panning Commands	222
restore Reset all rotations and magnifications	222
trans The center of the current picture is set to coordinates (x,y,z)	222
Interactive Commands: Magnification Commands	222
zf The magnification level is increased by a factor of s.....	222
zb The magnification level is decreased by a factor of s.....	222
Interactive Commands: Scaling Commands.....	222
xscale Apply scale factor sx to all of the x coordinates.....	222
yscale Apply scale factor sy to all of the y coordinates.....	222
zscale Apply scale factor sz to all of the z coordinates	222
scale Apply scale factor s to all coordinates	222
Interactive Commands: Part and Materials Display Commands	223
dap All of the parts are placed in the display list.....	223
dps A subset of the display list is selected	223
p Only part number n is placed in the display list	223
ap Part number n is added to the display list	223
rp Part number n is removed from the display list	223
parts Similar in function to the dap command.....	223
dms A subset of the materials display list is selected.....	223
dam All of the materials are placed in the materials display list	223
m Only material number n is placed in the materials display list.....	223
am Material number n is added to the materials display list	223
rm Material number n is removed from the materials display list	223

Command Index: (cont.)

Interactive Commands: Graphics Diagnostic Commands	224
di dx Highlight nodes with constrained x-axis translation.....	224
di dy Highlight nodes with constrained y-axis translation.....	224
di dz Highlight nodes with constrained z-axis translation.....	224
di rx Highlight nodes with constrained x-axis rotation	224
di ry Highlight nodes with constrained y-axis rotation	224
di rz Highlight nodes with constrained z-axis rotation	224
Interactive Commands: Sliding Interface Display Commands.....	224
di si Highlight sliding interface number n	224
Interactive Commands: Symmetry Plane, Stone Wall Commands.....	224
di nrb Highlight nodes on non-reflecting boundaries.....	224
di sy Highlight nodes on symmetry plane number n.....	224
di sw Highlight nodes on stone wall number n	224
Interactive Commands: Node and Element Identification Commands.....	225
di npb Display nodal print blocks	225
di epb Display element print blocks	225
di nodes Label visible nodes on the display where able	225
di 3d Label visible three dimensional brick elements on the display where able.....	225
di 2d Label visible two dimensional shell elements on the display where able	225
di 1d Label visible one dimensional beam elements on the display where able.....	225
Interactive Commands: Load Curve Display Commands	225
di d Display the nodes subject to the forced displacement of load curve n.....	225
di f Display nodes subject to the forces of load curve n	225
di pr Display nodes subject to the pressures of load curve n	225
Interactive Commands: Prescribed Boundary Condition Display Commands.....	226
di cv Locate the nodes subject to a convection boundary condition	226
di flux Highlight nodes subject to a flux boundary condition	226
di rb Highlight nodes subject to a radiation boundary condition	226
di re Highlight the facets of a radiation enclosure	226

Command Index: (cont.)

di resn	Display the radiation enclosure surface numbers	226
di tb	Highlight the nodes subject to a temperature boundary condition	226
di ti	Highlight the nodes subject to an initial temperature condition	226
di vb	Highlight nodes subject to the velocity boundary conditions of load curve n..	226
Interactive Commands: Part Tolerancing Commands		226
stol	Specify a tolerance of ϵ to be used in surface tolerancing	226
stp	Specify a tolerance of ϵ to be used in surface tolerancing	226
tol	Specify a tolerance of ϵ to be used in volumetric tolerancing	226
tp	Specify a tolerance of ϵ to be used in volumetric tolerancing	227
ztol	Specify a minimum absolute value for coordinates	227
Interactive Commands: General Line Display Commands		227
lv	Display all lines	227
lvi	Display a list of n lines with line numbers l1 to lm	227
ckl	Display line number	227
lvs	Display a sequence of lines, n through m, inclusive	227
Interactive Commands: Load Curve Display Commands		227
lcv	Display load	227
Interactive Commands: Non-Graphical Display Commands		228
ajnp	Prints node number of closest node to point (x,y,z)	228
center	Establishes the centroid of the model	228
centroid	Produces centroid information	228
ckv	Check element volumes, and print a summary	228
info	Provides general element information about the mesh	228
mass	Generates a mass table containing the volume, mass and center of mass	228
pinfo	Provides general element information	228
pmass	Generates mass information for the parts in the active display list	228
reference	Establishes point (x,y z) as a reference for moments of inertia	228
size	Determines and prints the current physical X, Y, and Z extent of the model ...	228
tmass	Generates mass information similar to pmass	228

Command Index: (cont.)

Interactive Commands: Output Commands.....		228
cont	Causes output file to be generated and terminates execution of INGRID.....	228
end	Graceful termination of INGRID. No output file is generated	228
stop	Causes an immediate and ungraceful termination of INGRID.....	228
Interactive Commands: LLNL Specific Output Commands.....		229
dli	Generate a dli file on the CRAY	229
rjet	Dispatch hardcopy output to RJET number n.....	229
plots	Create an FR80 plotfile	229
fr80	Send film output to the camera's.....	229
tv	Send displays to TMDS monitor number n	229

5.1 Standard Part

INGRID's standard part is used to generate 4-node shell elements, 8-node thick shell elements, and 8-node hexahedral elements. For an overview of INGRID's mesh generation capabilities, coordinate systems and analysis options, please refer to Chapter 2. Chapter 3 provides an introduction to basic concepts in mesh generation. By default, standard parts are generated in INGRID's Cartesian coordinate system using material number 1 (even if material 1 is undefined). Shell elements are generated with zero thickness unless otherwise specified. Standard parts are defined with the following syntax:

```
start
Index Progression
Part Control Commands
Functions
end
```

Standard Part Definition

<u>Command</u>	<u>Description</u>
start	Define a new standard part.
end	Required end statement for part-end definition of standard part.

In the standard part, generation of shell elements vs. brick elements is determined by the index progression automatically. If only hexahedral elements are to be generated, then the following simplified syntax can be used:

```
stab  
Index Partition Progression  
Part Control Commands  
Functions  
end
```

Stab Part Definition

<u>Command</u>	<u>Description</u>
stab	Define a new stab part.
end	Required end statement for part-end definition of stab part.

Index Partition Progressions

While the index progression provides a means for describing the region boundaries in the computational domain in a compact form, the index partition progressions make it possible to describe the computational domain directly in terms of the number of elements between region boundaries. This requires the use of the **stab** part. The index partition progression requires one less index in each of the coordinate directions (i,j,k) in index space than the standard part does. As an example, the following produces the same mesh as shown in section 3.3 for initial coordinates..

```
stab
1 3; 2; 3;
1.0 2.0 3.0
1.0 2.0
0.0 3.0
end
```

While the number of indices is reduced for the **stab** part, all of the control points must be specified for the region boundaries just as in the standard part syntax. All of the part control commands and functions for the standard part (**start - end**) apply to the stab part (**stab - end**). For details regarding index and index partition progressions, refer to chapter 3 on Basic Concepts.

5.1.1 Standard Part Control

This section defines the commands used to perform coordinate transformations, specify materials, initial conditions, etc. in INGRID's standard part. The control commands are presented in the following order:

- General Control Commands
- Geometric Transformations - Projections
- Geometric Transformations - Point/Line Modifications
- Geometric Transformations - Part Replication
- Boundary Condition and Sliding Interface Commands
- In-part Beam Generation
- Geometrical Part Control
- In-line editor/calculator

In the description of the standard part control commands and functions, *<index>* indicates that an index progression must be input by the user, while *<r-index>* indicates that a reduced index progression is required. *<region>* denotes a region of index space using reduced index values. *<point>* requires that the user input a single value of i, j, and k to identify a single point in index space. For more information regarding index progressions and regions, refer to Chapter 3 on Basic Concepts.

By default, INGRID's standard part produces either 4-node shells or 8-node bricks in Cartesian coordinates. The default material number associated with all standard parts is 1, and the default shell thickness is 0.0.

Standard Part Control

General Control Commands

Command	Description
cyli	Nodal point coordinates are converted from cylindrical to Cartesian coordinates as: $x = r \cos(\theta)$, $y = r \sin(\theta)$. Please refer to Figure 2.5.
d <region>	Delete <region> from the index space and the part.
di <r-index>	Delete <r-index> from the index space and the part.
epb <point> options;	Setup an element print block.
<u>epb - options:</u>	
po <i>i j k</i>	Use the element offset from <point> by (<i>i,j,k</i>) in reduced index space in the element print block.
ro <region>	<i>Use the block of elements in <region> offset from <point> in the element print block.</i>
mate <i>m</i>	Specify material number <i>m</i> for the current standard part.
mt <region> <i>n</i>	Specifies that material number <i>n</i> should be used in <region>.
mti <r-index> <i>n</i>	Specifies that material number <i>n</i> should be used in <r-index>.
mtv <i>m n</i>	All elements contained in the volume <i>m</i> are assigned a material number <i>n</i> . Here, <i>m</i> is a global volume definition (Section 4.2.13).
n+ <region> $p_x p_y p_z$	Specify that the positive outward shell normals face toward the point (p_x, p_y, p_z) in the local coordinate system.
n- <region> $p_x p_y p_z$	Specify that the positive outward shell normals face away from the point (p_x, p_y, p_z) in the local coordinate system.
npb <point> options;	Setup a nodal print block.
<u>npb - options:</u>	
po $\Delta i \Delta j \Delta k$	Use the node offset from <point> by $\Delta i, \Delta j, \Delta k$ for the nodal print block.
ro <region>	Use the block of nodes in <region> offset from <point>.

Standard Part Control (cont.)

General Control Commands (cont.)

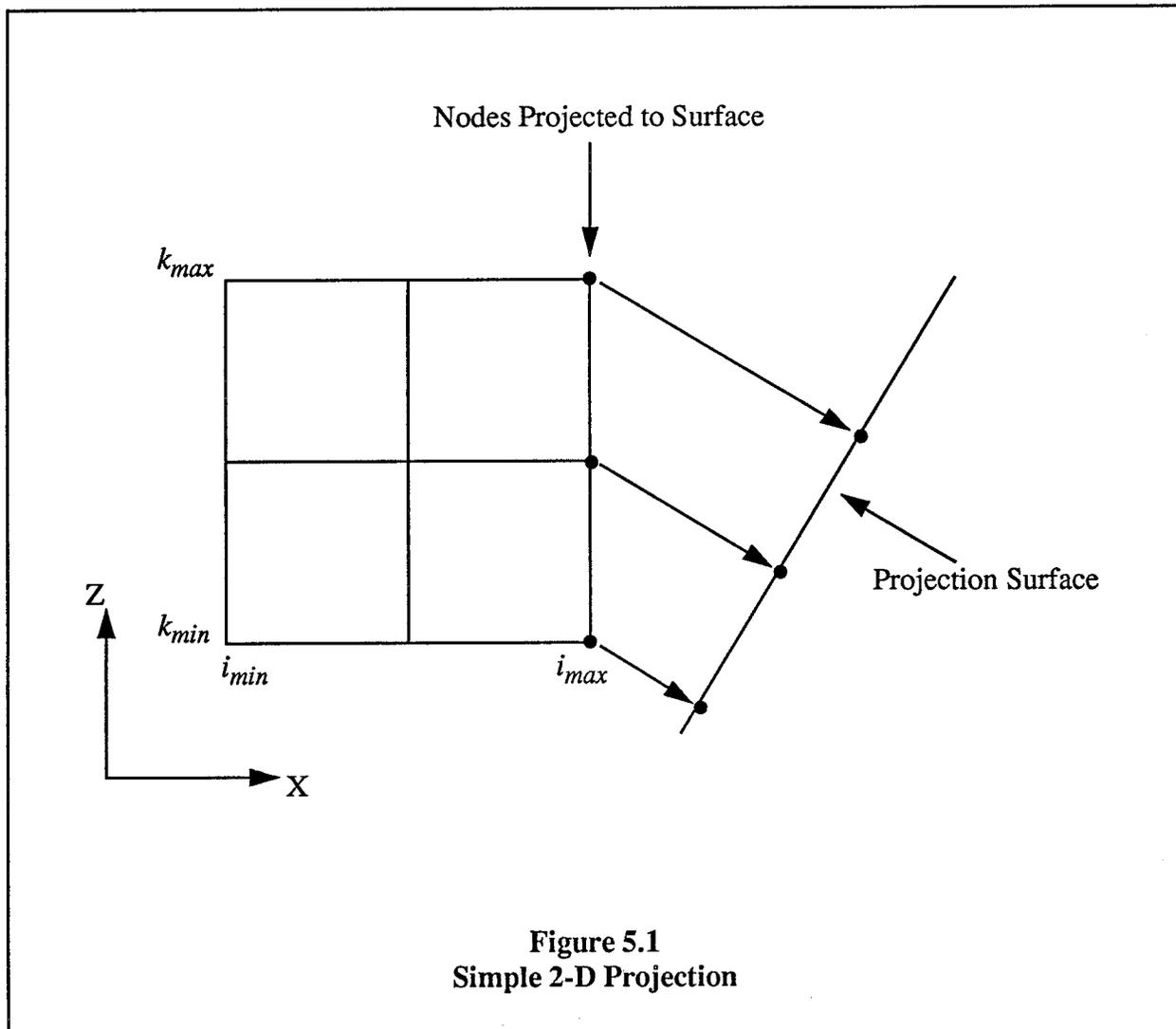
Command	Description
or <i><region></i> l_1 l_2	Specify the orientation of the local axes where l_1 and l_2 may be one of the following: i : specifies i-direction j : specifies j-direction k : specifies k-direction Note that this command is not available for the stab part.
print nodes <i><region></i> n	Specify that node numbers be printed in the INGRID save file for <i><region></i> . The block of node numbers is labeled according to the value of n , and all of the node number blocks are sorted according to n .
sphe	Nodal point coordinates are converted from spherical to Cartesian coordinates as: $x = r \cos(\theta) \sin(\psi)$, $y = r \sin(\theta) \sin(\psi)$, $z = r \cos(\psi)$. Please refer to Figure 2.5.
temp T	All parts defined after this command in the mesh have an initial temperature of T .
thic t	Shell elements have thickness t .
th <i><region></i> t	Specifies that <i><region></i> should have a shell thickness t .
thi <i><r-index></i> t	Specifies that <i><r-index></i> should have a shell thickness t .
thth n v_x v_y v_z	Specify the thickness variation of a thin shell using thcd curve n . The thickness is determined from the angle between the vector $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ and the outward normal for the shell at each nodal point. This thickness variation applies to the entire standard part.
tn <i><region></i> n	Specifies that <i><region></i> should have a shell thickness number n .
tni <i><r-index></i> n	Specifies that <i><r-index></i> should have a shell thickness number n .
ve <i><region></i>	Activate the edge visibility for outline and phantom edge plotting. <i><region></i> must identify a line in the reduced index space.

Standard Part Control (cont.)

Geometric Transformations - Projections

The projection commands operate by moving nodes from an initial location based upon the index progression and control coordinates to the closest point on the surface. Figure 5.1 illustrates a very simple example of this type of projection in 2-D. Intersections of surfaces in the index space are automatically detected and calculated. Since intersections are rarely unique, the user must define initial coordinates which are near the final part configuration in order for INGRID to generate the desired geometry (please refer to Chapter 3 for more details).

This section begins with the commands for performing simple projections to arcs and cylindrical regions. The general surface projection commands are then presented, followed by the commands for performing multiple surface projections.



Standard Part Control (cont.)

Geometric Transformations - Projections (cont.)

Command	Description
a <region> type x y z r	Project <region> onto an arc of type. x,y,z specifies a point at the center of the arc, or through which the arc must pass (Figure 5.2). r specifies the arc radius, and type may have the following values
a - type:	
1	A parabolic arc passing through point p ₁ - see Figure 5.2 below.
2	A circular arc passing through point p ₁ .
3	A circular arc with center at p ₂ .
	If r is non-zero for a circular arc with center at p ₂ , then nodes n ₁ and n ₂ are moved radially from p ₂ until they are a distance r from p ₂ . A circular arc is then formed which passes through n ₁ and n ₂ .

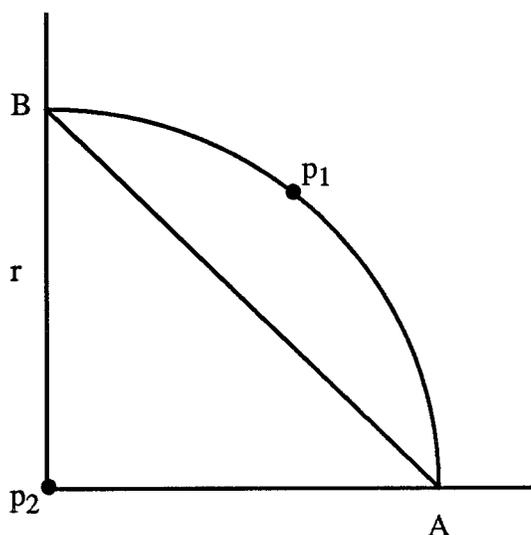
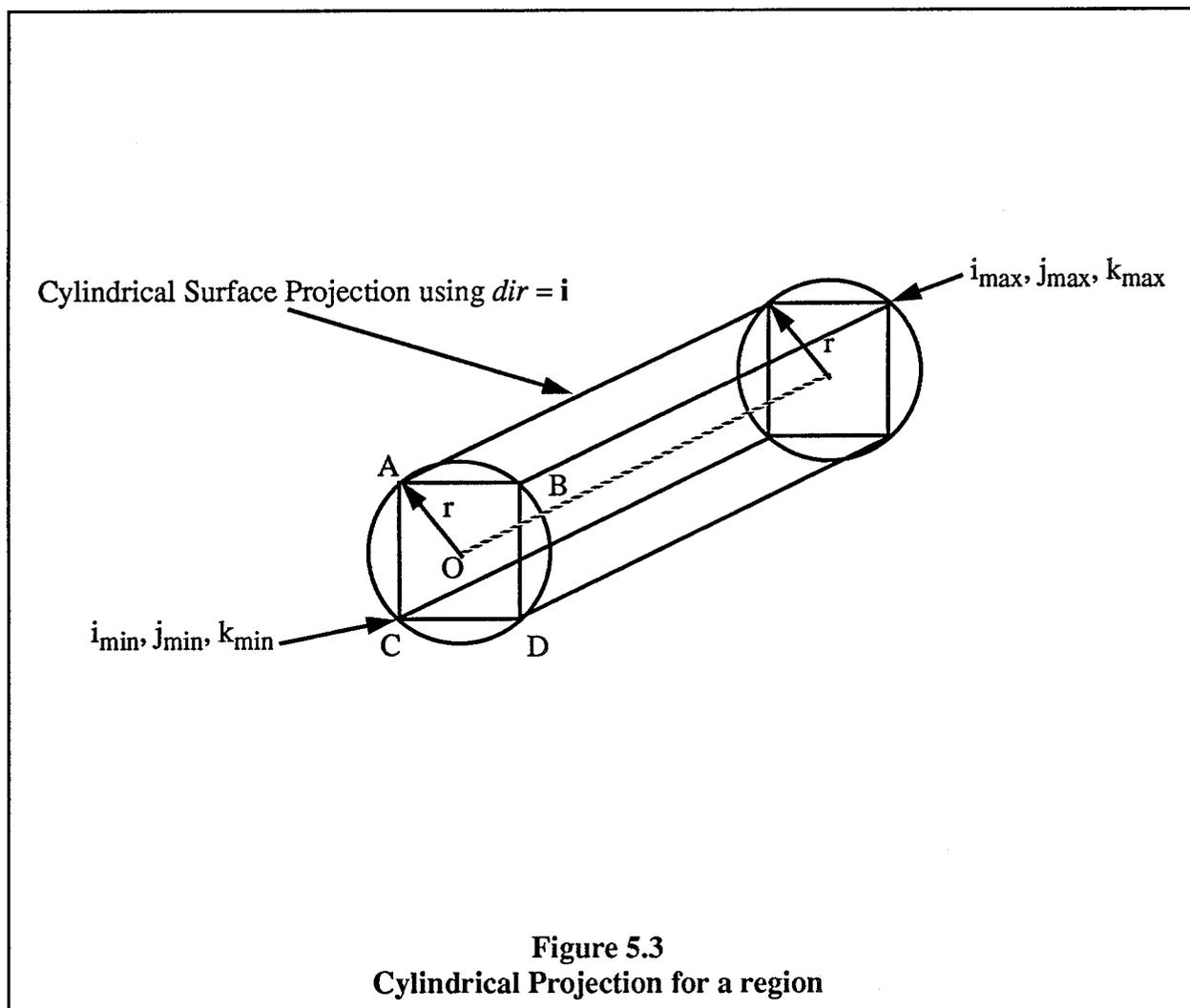


Figure 5.2
Curved Boundaries using the "a" function.

Standard Part Control (cont.)

Geometric Transformations - Projections (cont.)

Command	Description
a <region> <i>dir</i> <i>r</i>	<p>Project a plane or a solid in the reduced index space onto a cylindrical surface with radius <i>r</i>.</p> <p><i>dir</i> = i : Uses the i-axis for the axis of rotation. <i>dir</i> = j : Uses the j-axis for the axis of rotation. <i>dir</i> = k : Uses the k-axis for the axis of rotation.</p> <p>For any plane normal to the axis of rotation, such as ABCD in Figure 5.3, a point O on the axis of rotation is located in the center of the plane. If the radius of the cylinder is not zero, then the points A,B,C,D are moved radially from O until they are a distance <i>r</i> from O. Curved boundaries are then formed for the segments AB, BD, AC and CD using O as the center. This is done for each plane normal to the axis of rotation in the reduced index space.</p>



Standard Part Control (cont.)

Geometric Transformations - Projections (cont.)

Command	Description
ac <i><region></i> <i>dir</i> $p_x p_y p_z r v$	Project a surface (<i><region></i>) to a cylindrical surface with radius r . (p_x, p_y, p_z) is any point on the axis of the cylinder, and $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is vector parallel to the axis of the cylinder. <i>dir</i> specifies the direction of the axis of rotation. <i>dir</i> = \mathbf{i} : Uses the i-axis for the axis of rotation. <i>dir</i> = \mathbf{j} : Uses the j-axis for the axis of rotation. <i>dir</i> = \mathbf{k} : Uses the k-axis for the axis of rotation.
ace <i><region></i> <i>dir</i> $p_x p_y p_z r v$	Project a surface (<i><region></i>) to a cylindrical surface with radius r . (p_x, p_y, p_z) is any point on the axis of the cylinder, and $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is vector parallel to the axis of the cylinder. ace forces nodal points to be uniformly distributed on the arcs. <i>dir</i> specifies the direction of the axis of rotation. <i>dir</i> = \mathbf{i} : Uses the i-axis for the axis of rotation. <i>dir</i> = \mathbf{j} : Uses the j-axis for the axis of rotation. <i>dir</i> = \mathbf{k} : Uses the k-axis for the axis of rotation.
ae <i><region></i> <i>dir</i> r	Project a plane or a solid in the reduced index space onto a cylindrical surface with radius r . ae forces nodal points to be uniformly distributed on the arcs. <i>dir</i> = \mathbf{i} : Uses the i-axis for the axis of rotation. <i>dir</i> = \mathbf{j} : Uses the j-axis for the axis of rotation. <i>dir</i> = \mathbf{k} : Uses the k-axis for the axis of rotation.
ae <i><region></i> <i>type</i> $x y z r$	Project <i><region></i> onto an arc of <i>type</i> . x, y, z specifies a point at the center of the arc, or through which the arc must pass (Figure 5.2). ae forces nodal points to be uniformly distributed on the arcs. r specifies the arc radius, and <i>type</i> may have the following values
ae - type:	
1	A parabolic arc passing through point p_1 .
2	A circular arc passing through point p_1 .
3	A circular arc with center at p_2 . If r is non-zero for a circular arc with center at p_2 , then nodes n_1 and n_2 are moved radially from p_2 until they are a distance r from p_2 . A circular arc is then formed which passes through n_1 and n_2 .

Standard Part Control (cont.)

Geometric Transformations - Projections (cont.)

The following commands are used specifically for surface projections and allow for the exact specification of equations for 3-D surfaces. If a part is generated in cylindrical coordinates, the surfaces are still assumed to be in Cartesian coordinates. This permits non-axisymmetric surfaces to be generated on primarily axisymmetric parts.

Surface Projections

Command	Description
sf <region> type <u>sf - type:</u> sd n	Project <region> to a surface specified by type. Use the globally defined surface definition n. If type is not sd, then a surface definition may be input "in-line" according to section 4.2.12. The available sd commands have been listed below, and may be included in-line in the sf command.
sfi <r-index> type <u>sf - type:</u> sd n	Project <r-index> to a surface specified by type. Use the globally defined surface definition n. If type is not sd, then a surface definition may be input "in-line" according to section 4.2.12. The available sd commands have been listed below, and may be included in-line in the sfi command.

sd - Surface Definitions (Section 4.2.12):

ac $p_x p_y p_z q_x q_y q_z n a_0 a_1 \dots a_n$

Defines a surface by rotating a cosine polynomial about an axis. (p_x, p_y, p_z) is a point defining the location of the curve, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of rotation, n is the number of terms in the polynomial, and a_0, a_1, \dots, a_n are the n coefficients for the cosine series. The surface radius is computed as: $r = a_0 + a_1 \cos^2(\theta) + \dots + a_n \cos^n(\theta)$. For this surface type, n must be less than or equal to 12.

cn2p $p_x p_y p_z q_x q_y q_z r_1 z_1 r_2 z_2$

Define a conical surface using two points. (p_x, p_y, p_z) is point on the axis of the cone, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the cone. (r_1, z_1) and (r_2, z_2) are radial and axial coordinates of two points on the conical surface.

cone $p_x p_y p_z q_x q_y q_z r \theta$

Define a conical surface. (p_x, p_y, p_z) is point on the axis of the cone, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the cone, r is the radius of the cone at the point (p_x, p_y, p_z) . θ is the angle of the cone relative to the vector (q_x, q_y, q_z) , and is specified in degrees.

Standard Part Control (cont.)

Surface Projections (cont.)

Command	Description
<u>sd - Surface Definitions (Section 4.2.12):</u>	
cp <i>trans n</i>	Form an infinite surface from line definition <i>n</i> . Initially, the line is defined in the r-z plane. After projecting to a standard part, the part will have the same profile as the curve in the x-y plane along the z-axis. Transformations, <i>trans</i> , may be applied to the surface. These transformation are described in section 5.1.1 Geometric Transformations - Part Replication.
cr $p_x p_y p_z q_x q_y q_z n$	Define a surface by rotating a discrete line about an axis. (p_x, p_y, p_z) is point defining the location of the line, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of rotation, <i>n</i> is the number of a line definition which has been defined with the ld command.
crx <i>n</i>	Generate a surface by rotating a line defined in the x-z (or r-z) plane about the x-axis using line definition <i>n</i> .
cry <i>n</i>	Generate a surface by rotating a line defined in the x-z (or r-z) plane about the x-axis using line definition <i>n</i> .
crz <i>n</i>	Generate a surface by rotating a line defined in the x-z (or r-z) plane about the x-axis using line definition <i>n</i> .
cyl $p_x p_y p_z q_x q_y q_z r$	Define a cylindrical surface with radius <i>r</i> . (p_x, p_y, p_z) is point defining the origin of the axis of rotation, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of rotation.
er $p_x p_y p_z q_x q_y q_z r_1 r_2$	Defines an surface generated by rotating an ellipse with center (p_x, p_y, p_z) where (q_x, q_y, q_z) is a vector oriented normal to the axis of revolution. r_1 is the maximum radius in the plane normal to the axis of rotation, and r_2 is the distance from the center point along the axis of rotation to the ellipse.
gs <i>n</i>	Use discrete surface <i>n</i> , where the surface has been defined using the ds command described above.
gsl <i>n</i>	Use the lower (under) side of discrete surface <i>n</i> , where the surface has been defined using the ds command described above.
gsu <i>n</i>	Use the upper side of discrete surface <i>n</i> , where the surface has been defined using the ds command described above.

Standard Part Control (cont.)

Surface Projections (cont.)

Command	Description
<u>sd - Surface Definitions (Section 4.2.12):</u>	
pl3 $f_1 x_1 y_1 z_1 f_2 x_2 y_2 z_2 f_3 x_3 y_3 z_3 \delta$	Define a plane using three points. The flags f_1, f_2, f_3 specify the type of coordinate system associated with each point. f_1, f_2 and f_3 may take on values of rt for Cartesian, cy for cylindrical or sp for spherical coordinates. δ specifies the offset for the plane along the positive outward surface normal. The sense of the outward normal is defined by the cross product of the vector from point 1 to point 3 with the vector from point 1 to point 2.
plan $p_x p_y p_z q_x q_y q_z$	Define a surface which is a plane. (p_x, p_y, p_z) is point on the plane, and (q_x, q_y, q_z) is a vector normal to the plane.
poly $p_x p_y p_z q_x q_y q_z n a_0 \dots a_n$	Define a surface by revolving a polynomial of degree n about an axis. (p_x, p_y, p_z) is a point defining the origin of the axis of rotation, and (q_x, q_y, q_z) is the vector direction of the axis of rotation. n is the degree of the polynomial. $n+1$ coefficients ($a_0 \dots a_n$) are required.
pr $p_x p_y p_z n_x n_y n_z r_1 t_1 r_2 t_2 r_3 t_3$	Defines a parabolic surface of revolution where (p_x, p_y, p_z) is a point defining the location of the parabola, (q_x, q_y, q_z) is a vector oriented parallel to the t-axis (Figure 4.2). (r_1, t_1) is the first point on the parabola. If $t_1=0$, then the pint lies on a plane normal to the t-axis at point (p_x, p_y, p_z) . The positive direction on the t-axis is the same as the direction of (q_x, q_y, q_z) . (r_2, t_2) and (r_3, t_3) are the 2nd and 3rd points on the parabola respectively.
sp $p_x p_y p_z r$	Defines a spherical surface where (p_x, p_y, p_z) is the center of a sphere with radius r .
ts $p_x p_y p_z q_x q_y q_z r t R$	Define a toroidal surface. (p_x, p_y, p_z) is point on the axis of the torus, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the torus.
ts2p $p_x p_y p_z q_x q_y q_z r_1 z_1 r_2 z_2 R$	Define a toroidal surface using two points. (p_x, p_y, p_z) is point on the axis of the torus, and (q_x, q_y, q_z) is a vector oriented parallel to the axis of the torus. (r_1, z_1) and (r_2, z_2) are radial and axial coordinates of two points on the toroidal surface. $ R $ is the radius of the surface from point (p_x, p_y, p_z) . If $R > 0$, then (r_2, z_2) lies to the left as one moves from (r_1, z_1) and (r_2, z_2) , otherwise, (p_x, p_y, p_z) lies to the right.

Standard Part Control (cont.)

Multiple Surface Projections

The following commands permit the identification of parallel index planes for the purpose of performing multiple surface projections. The function of the commands is similar to the `sf` command. However, the multiple surface definition commands can considerably reduce the amount of necessary user input for many common situations.

Command	Description
<code>ms <region> dir options</code>	Apply multiple surface projections to <code><region></code> . The <code><region></code> of index space is divided into a series of planes normal to the axis in the index space specified by <code>dir</code> . Next, one surface definition must be input for each of the index planes in <code><region></code> . <code>dir</code> may take on the following values: <code>dir = i</code> : Equal spacing in the i-direction <code>dir = j</code> : Equal spacing in the j-direction <code>dir = k</code> : Equal spacing in the k-direction
<u>ms - option #1:</u>	List of surface equations
<code>sd₁ ... sd_n</code>	A list of n surface definitions corresponding to the n-planes of index space in <code><region></code> . Surface definitions are detailed in section 4.2.12..
<u>ms - option #2:</u>	Parallel planes normal to a coordinate axis
<code>ppx</code>	Specifies plane normal to the x-axis.
<code>ppy</code>	Specifies a plane normal to the y-axis.
<code>ppz</code>	Specifies a plane normal to the z-axis.
<code>p₁ p₂ ... p_n</code>	The point along the specified axis where the planes intercept. One value must be input for each plane.
<u>ms - option #3:</u>	Concentric spheres
<code>cnsf p_x p_y p_z r₁ r₂ ... r_n</code>	(p _x , p _y , p _z) specifies the center of the spheres, and r ₁ , r ₂ , ... r _n are the sphere radii for the spheres from 1 to n respectively.
<u>ms - option #4:</u>	Concentric Cylinders
<code>ency x y z v_x v_y v_z r₁ r₂ ... r_n</code>	(x,y,z) is any point on the axis of the cylinder, $v = v_x i + v_y j + v_z k$ is vector parallel to the axis of the cylinder, and r ₁ , r ₂ , ... r _n are the cylinder radii for the cylinders from 1 to n respectively.

Standard Part Control (cont.)

Multiple Surface Projections (cont.)

Command	Description
<u>ms - option #5:</u>	Planes with offsets.
pon, pox, poy, poz	Planes offset normal or in the x, y, or z directions respectively.
<u>Required Data:</u>	
$P_x P_y P_z$	Any point on the plane.
$v_x v_y v_z$	Any vector normal to the plane
$\Delta_1 \Delta_2 \dots \Delta_n$	Offsets in the requested direction.

Geometric Transformations - Point/Line Modifications

Command	Description
cpl <region> dir	Center points along a line. If <region> is a line in index space, then this command has the effect of forcing equal spaced subdivisions along the line. If <region> is a surface or a volume, then <i>dir</i> specifies the direction in which equally spaced subdivisions should be generated. <i>dir</i> can take on the following values: <i>dir</i> = i : Equal spacing in the i-direction <i>dir</i> = j : Equal spacing in the j-direction <i>dir</i> = k : Equal spacing in the k-direction
i <point> dir c₁ c₂ c₃ ...	Coordinate modifying function which specifies that the coordinates vary as a function of the i-index. <i>dir</i> = x specifies that the x-coordinates will be modified. <i>dir</i> = y specifies that the y-coordinates will be modified. <i>dir</i> = z specifies that the z-coordinates will be modified. A coordinate progression must follow <i>dir</i> with the number of coordinates corresponding to the original index progression.
j <point> dir c₁ c₂ c₃ ...	Coordinate modifying function which specifies that the coordinates vary as a function of the j-index. <i>dir</i> = x specifies that the x-coordinates will be modified. <i>dir</i> = y specifies that the y-coordinates will be modified. <i>dir</i> = z specifies that the z-coordinates will be modified. A coordinate progression must follow <i>dir</i> with the number of coordinates corresponding to the original index progression.

Standard Part Control (cont.)

Geometric Transformations - Point/Line Modifications

Command	Description
k <point> dir c ₁ c ₂ c ₃ ...	Coordinate modifying function which specifies that the coordinates vary as a function of the k-index. <i>dir = x</i> specifies that the x-coordinates will be modified. <i>dir = y</i> specifies that the y-coordinates will be modified. <i>dir = z</i> specifies that the z-coordinates will be modified. A coordinate progression must follow dir with the number of coordinates corresponding to the original index progression.
int <region> n ₁ n ₂	Moves nodes to the nearest point on the intersection of two surfaces (or curves). Surfaces must be defined using the global control command, <i>sd</i> , for definition of surfaces. n ₁ and n ₂ are the surface numbers for the two intersecting surfaces. Note that this command will automatically be generated if two <i>sf</i> commands result in intersecting surfaces.
ma <point> type Δx Δy Δz	Modify 1, 2, or 3 coordinates at a <point> in index space. The Δx, Δy, Δz values are added to the current x,y,z values at <point>, and are optional based upon the type of point manipulation. For example, only the Δx value is required when <i>type = x</i> . <i>type</i> may have the following values:
<u>ma - type:</u> x y z xy yz xyz	x-coordinate is modified (Δx required). y-coordinate is modified (Δy required). z-coordinate is modified (Δz required). x and y-coordinates are modified (Δx,Δy required). y and z-coordinate is modified (Δy,Δz required). x, y and z-coordinate is modified (Δx,Δy,Δz required).
mb <region> type Δx Δy Δz	Modify 1, 2, or 3 coordinates over a <region> in index space. The Δx, Δy, Δz values are added to the current x,y,z values over the <region>, and are optional based upon the type of manipulation. For example, only the Δx value is required when <i>type = x</i> . <i>type</i> may have the following values:
<u>mb - type:</u> x y z xy yz xyz	x-coordinate is modified (Δx required). y-coordinate is modified (Δy required). z-coordinate is modified (Δz required). x and y-coordinates are modified (Δx,Δy required). y and z-coordinate is modified (Δy,Δz required). x, y and z-coordinate is modified (Δx,Δy,Δz required).

Standard Part Control (cont.)

Geometric Transformations - Point/Line Modifications (cont.)

Command	Description
pa <i><point> type x y z</i>	Modify 1, 2, or 3 coordinates at a <i><point></i> in index space. The <i>x</i> , <i>y</i> , <i>z</i> values are optional based upon the type of point manipulation, and are substituted for the current <i>x,y,z</i> values at <i><point></i> . For example, only <i>x</i> is required when <i>type = x</i> . <i>type</i> may have the following values: pa - type: <i>x</i> x-coordinate is substituted (<i>x</i> required). <i>y</i> y-coordinate is substituted (<i>y</i> required). <i>z</i> y-coordinate is substituted (<i>z</i> required). <i>xy</i> x and y-coordinates are substituted (<i>x,y</i> required). <i>yz</i> y and z-coordinate is substituted (<i>y,z</i> required). <i>xyz</i> x, y and z-coordinate is substituted (<i>x,y,z</i> required).
pb <i><region> type x y z</i>	Modify 1, 2, or 3 coordinates over a <i><region></i> in index space. The <i>x</i> , <i>y</i> , <i>z</i> values are optional based upon the type of point manipulation, and are substituted for the current <i>x,y,z</i> values over the <i><region></i> . For example, only <i>x</i> is required when <i>type = x</i> . <i>type</i> may have the following values: pa - type: <i>x</i> x-coordinate is substituted (<i>x</i> required). <i>y</i> y-coordinate is substituted (<i>y</i> required). <i>z</i> y-coordinate is substituted (<i>z</i> required). <i>xy</i> x and y-coordinates are substituted (<i>x,y</i> required). <i>yz</i> y and z-coordinate is substituted (<i>y,z</i> required). <i>xyz</i> x, y and z-coordinate is substituted (<i>x,y,z</i> required).
res <i><region> dir r</i>	Rezone elements along the side of <i><region></i> . <i>dir</i> specifies the direction to be operated on in <i><region></i> , and may take on one of the following values: <i>ll = I</i> : specifies i-direction <i>ll = J</i> : specifies j-direction <i>ll = K</i> : specifies k-direction <i>r</i> is the ratio of the length of one element side to the next adjacent element side as the index increases in the specified index direction.
rr <i><region> trans</i>	Apply one of the coordinate transformations detailed in Geometric Transformations - Part Replication below to the specified region. For example, to rotate a given region about the x-axis by 10 degrees: rr <region> rx 10;

Standard Part Control (cont.)

Geometric Transformations - Part Replication

Command	Description
lct $n; t_1; t_2; \dots; t_n;$	Define a sequence of n local coordinate transformations $t_1 \dots t_n$. Acceptable transformations used to define $t_1 \dots t_n$ are listed below, and are required. This command must precede lrep for local part replication. lct and lrep are intended to be used as companion commands for performing coordinate transformations and part replications.
<u>lct - Required data for transformations:</u>	
cscs s	Scale all nodal coordinates by s .
mx Δx	Translate by Δx .
my Δy	Translate by Δy .
mz Δz	Translate by Δz .
v $\Delta x \Delta y \Delta z$	Translate by $\Delta x, \Delta y, \Delta z$.
repe m	Repeat the local coordinate transformation m times.
rx θ_x^0	Rotate by θ_x^0 about the x-axis.
ry θ_y^0	Rotate by θ_y^0 about the y-axis.
rz θ_z^0	Rotate by θ_z^0 about the z-axis.
rxv	Reflect the part about the x-y plane.
ryz	Reflect the part about the y-x plane.
rzx	Reflect the part about the z-x plane.
save m	The sequence of coordinate transformations is generated starting from coordinate system m when using the repe command.
xcsca s	Scale all nodal x-coordinates by s .
ycsca s	Scale all nodal y-coordinates by s .
zcsca s	Scale all nodal z-coordinates by s .
lrep $0 \ 1 \dots n;$	The lrep command performs local part replication, and is intended to be used with the lct command. Here, the 0 indicates that the original instance of the part should be kept in the finite element mode. The sequence of 1 to n applies the lct coordinate transformations to the part keeping each instance of the transformed part. For example, lrep 0 1 2; would keep the original part, and apply the local transformations 1 and 2 from the lct command: lct 2; mx Δx; my Δy;
repe $c_1 \ c_2 \dots c_n;$	Repeat command. This command makes copies of the part in each of the global coordinate systems $c_1 \dots c_n$. If c_i is zero, then the part is replicated without transformation.

Standard Part Control (cont.)Geometrical Part Control

Command	Description
jt <i>n m options;</i>	Generate a joint using joint definition <i>n</i> and local node number <i>m</i> (Figure 5.4).
<u>jt - options:</u>	
n <i><point></i>	The local joint node, <i>m</i> , is defined by <i><point></i> in the index space.
p <i>p_x p_y p_z rb</i>	The local joint node, <i>n</i> , is at the point (<i>p_x, p_y, p_z</i>) in the local coordinate system. <i>rb</i> is the rigid body number which is attached to the node.
inc <i>i</i>	Increment the joint number <i>n</i> by <i>i</i> for each copy of the part. The default number of copies is 1.
b <i>n</i>	The local joint node has boundary constraint <i>n</i> , where <i>n</i> is a six digit number specifying the constrained degree of freedom. <i>n</i> = 100000 fixes the x-displacement <i>n</i> = 010000 fixes the y-displacement <i>n</i> = 001000 fixes the z-displacement <i>n</i> = 000100 fixes the x-rotation <i>n</i> = 000010 fixes the y-rotation <i>n</i> = 000001 fixes the z-rotation

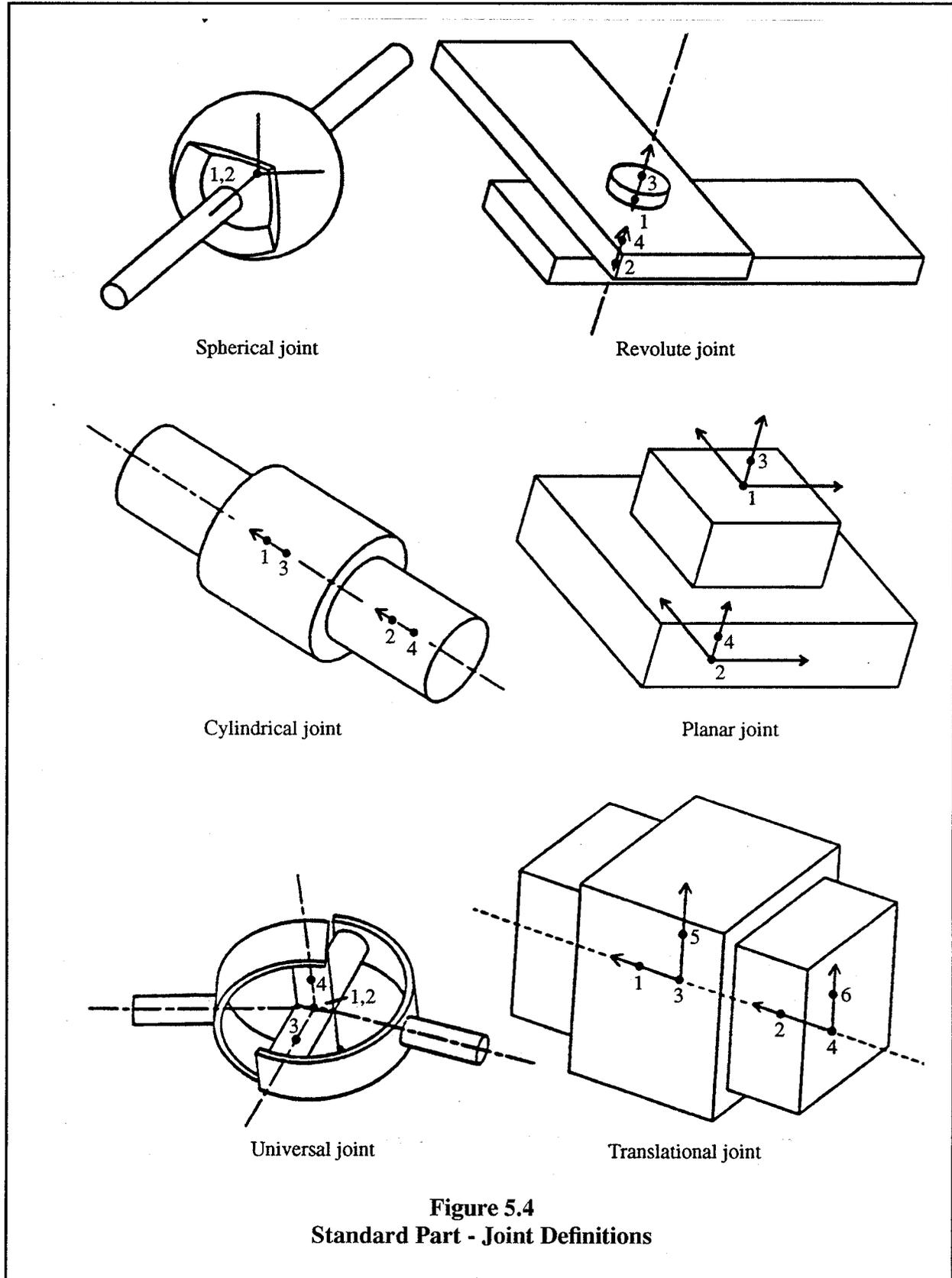


Figure 5.4
Standard Part - Joint Definitions

5.1.2 Standard Part Boundary Conditions

This section details the commands used to specify boundary conditions in standard parts.

Electromagnetic Loads and Boundary Conditions

Command	Description
em <region> <i>id options</i> $v_x v_y v_z$	Specify boundary conditions for MAXWELL3. Here, <i>id</i> is an integer identifier, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the normal direction in the local coordinate system.
em - Options:	
inc	Incident fields are specified by function <i>id</i> .
scf	Scattered fields are specified by function <i>id</i> .
out	Output surface with part <i>id</i> for post processing
cxv	Specify coaxial termination with voltage output <i>id</i> .
rbc	Radiation boundary condition.
ebc	E-field tangential to surface is zero.
hbc	H-field tangential to surface is zero.
e=0	The total E-field at the surface is zero.
mat	Specifies a material interface at the surface.
emi <r-index> <i>id options</i> $v_x v_y v_z$	Specify boundary conditions for MAXWELL3. Here, <i>id</i> is an integer identifier, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the normal direction in the local coordinate system.
emi - Options:	
inc	Incident fields are specified by function <i>id</i> .
scf	Scattered fields are specified by function <i>id</i> .
out	Output surface with part <i>id</i> for post processing
cxv	Specify coaxial termination with voltage output <i>id</i> .
rbc	Radiation boundary condition.
ebc	E-field tangential to surface is zero.
hbc	H-field tangential to surface is zero.
e=0	The total E-field at the surface is zero.
mat	Specifies a material interface at the surface.
fbc <region> <i>type</i>	Specify an electromagnetic boundary condition on <region> where <i>type</i> may be one of the following.

Standard Part Boundary Conditions (cont.)

Electromagnetic Loads and Boundary Conditions (cont.)

Command	Description
fbci < <i>r-index</i> > <i>type</i>	Specify an electromagnetic boundary condition on < <i>r-index</i> > where <i>type</i> may be one of the following.
<u>fbci/fbci - types:</u>	
metal	metal electric conductor.
ametal	magnetic conductor.
zero	zero out fields.
00	E and H fields are zero - for use when specifying total fields.
01	The normal component of H is set to zero, and E is set to zero.
02	The tangential component of H is set to zero, and E is set to zero.
03	E is set to zero.
04	The normal component of E is set to zero, and H is set to zero.
05	The normal components of E and H are set to zero.
06	The tangential components of E and H are set to zero.
07	The normal component of E is set to zero.
10	The tangential component of E is set to zero, and H is set to zero.
11	The tangential component of E and the normal component of H is set to zero.
12	The tangential components of E and H are set to zero.
13	The tangential component of E is set to zero.
14	H is set to zero.
15	The normal component of H is set to zero.
16	The tangential component of H is set to zero.
17	No fields are set to zero.
per < <i>region</i> > <i>n type</i>	Specify periodic boundary conditions for MAXWELL3. <i>n</i> is the boundary number of the region, and <i>type</i> can be either m for master, or s for slave. For each boundary of the region, an ordered list of master and slave nodes is generated. Here, if the minimum index values in the region exceed the maximum values, then the node ordering is reversed.
spc < <i>region</i> > <i>n</i>	Specify a current on < <i>region</i> > using load curve <i>n</i> .
spci < <i>r-index</i> > <i>n</i>	Specify a current on < <i>r-index</i> > using load curve <i>n</i> .
spf < <i>region</i> > <i>n</i>	Specify a field on < <i>region</i> > using load curve <i>n</i> .
spfi < <i>r-index</i> > <i>n</i>	Specify a field on < <i>r-index</i> > using load curve <i>n</i> .
efl < <i>region</i> > <i>f</i>	Specify an electrostatic flux, <i>f</i> , on < <i>region</i> >.
efli < <i>r-index</i> > <i>f</i>	Specify an electrostatic flux, <i>f</i> , on < <i>r-index</i> >.

Standard Part Boundary Conditions (cont.)

Electrostatic Loads and Boundary Conditions

Command	Description
$v <region> \phi$	Specify an electrostatic potential, ϕ , on $<region>$.
$vi <r-index> \phi$	Specify an electrostatic potential, ϕ , on $<r-index>$.

Sliding Interface Commands

Command	Description
$si <region> n \text{ type options}$	Specifies that $<region>$ should be a slide surface of <i>type</i> using the global slide surface definition n .
<u>si - type:</u>	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
<u>si - Options for slave surface only:</u>	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the global slide surface <i>type</i> is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.
$si+ <region> n \text{ type } p_x p_y p_z$	Specifies that $<region>$ should be a slide surface of <i>type</i> using the global slide surface definition n . (p_x, p_y, p_z) specifies a point in the local coordinate system towards which the sliding interface faces.
<u>si+ - type:</u>	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
<u>si+ - Options for slave surface only:</u>	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the slide surface <i>type</i> is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.

Standard Part Boundary Conditions (cont.)

Sliding Interfaces Commands (cont.)

Command	Description
si- <i><region> n type p_x p_y p_z</i>	Specifies that <i><region></i> should be a slide surface of <i>type</i> using the global slide surface definition <i>n</i> . (<i>p_xp_yp_z</i>) specifies a point in the local coordinate system which the sliding interface faces away from.
si- - type:	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
si- - Options for slave surface only:	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the global slide surface type is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.
sii <i><r-index> n type</i>	Specifies that <i><r-index></i> should be a slide surface of <i>type</i> using the global slide surface definition <i>n</i> .
sii - type:	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
sii - Options for slave surface only:	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the global slide surface <i>type</i> is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.

Standard Part Boundary Conditions (cont.)

Sliding Interfaces Commands (cont.)

Command	Description
sii+ < <i>r-index</i> > <i>n type p_x p_y p_z</i>	Specifies that < <i>r-index</i> > should be a slide surface of <i>type</i> using the global slide surface definition <i>n</i> . (<i>p_xp_yp_z</i>) specifies a point in the local coordinate system toward which the sliding interface faces.
<u>sii+ - type:</u>	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
<u>sii+ - Options for slave surface only:</u>	
<i>f_nf_s α β</i>	If the global slide surface <i>type</i> is nsw , then the normal force at failure, <i>f_n</i> , the shear force at failure, <i>f_s</i> , and the exponents for the normal and shear forces (<i>α</i> and <i>β</i> respectively) must be specified for the slave slide surface.
<i>σ_n σ_s</i>	If the global slide surface <i>type</i> is break , then the normal failure stress, <i>σ_n</i> , and the shear failure stress, <i>σ_s</i> , must be specified for the slave slide surface.
sii- < <i>r-index</i> > <i>n type p_x p_y p_z</i>	Specifies that < <i>r-index</i> > should be a slide surface of <i>type</i> using the global slide surface definition <i>n</i> . (<i>p_xp_yp_z</i>) specifies a point in the local coordinate system which the sliding interface faces away from.
<u>sii- - type:</u>	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
<u>sii- - Options for slave surface only:</u>	
<i>f_nf_s α β</i>	If the global slide surface <i>type</i> is nsw , then the normal force at failure, <i>f_n</i> , the shear force at failure, <i>f_s</i> , and the exponents for the normal and shear forces (<i>α</i> and <i>β</i> respectively) must be specified for the slave slide surface.
<i>σ_n σ_s</i>	If the global slide surface <i>type</i> is break , then the normal failure stress, <i>σ_n</i> , and the shear failure stress, <i>σ_s</i> , must be specified for the slave slide surface.

Standard Part Boundary Conditions (cont.)

Solid Mechanics Boundary Conditions

Command	Description												
acc <region> <i>n</i> <i>A</i> $a_x a_y a_z$	Specify acceleration on <region>. <i>n</i> identifies the load curve number, <i>A</i> is the load curve scale factor, and $\mathbf{a} = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
acc <region> <i>n</i> <i>A</i> $a_x a_y a_z$	Specify acceleration on <region> in cylindrical coordinates. <i>n</i> identifies the load curve number, <i>A</i> is the load curve scale factor, and $\mathbf{a} = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
accs <region> <i>n</i> <i>A</i> $a_x a_y a_z$	Specify acceleration on <region> in spherical coordinates. <i>n</i> identifies the load curve number, <i>A</i> is the load curve scale factor, and $\mathbf{a} = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
acci < <i>r-index</i> > <i>n</i> <i>A</i> $a_x a_y a_z$	Specify acceleration on < <i>r-index</i> >. <i>n</i> identifies the load curve number, <i>A</i> is the load curve scale factor, and $\mathbf{a} = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
b <region> <i>n</i>	Specifies displacement boundary conditions on <region> where <i>n</i> is a six digit number specifying the constrained degree of freedom. <table style="margin-left: 2em; border: none;"> <tr> <td><i>n</i> = 100000</td> <td>fixes the x-displacement</td> </tr> <tr> <td><i>n</i> = 010000</td> <td>fixes the y-displacement</td> </tr> <tr> <td><i>n</i> = 001000</td> <td>fixes the z-displacement</td> </tr> <tr> <td><i>n</i> = 000100</td> <td>fixes the x-rotation</td> </tr> <tr> <td><i>n</i> = 000010</td> <td>fixes the y-rotation</td> </tr> <tr> <td><i>n</i> = 000001</td> <td>fixes the z-rotation</td> </tr> </table>	<i>n</i> = 100000	fixes the x-displacement	<i>n</i> = 010000	fixes the y-displacement	<i>n</i> = 001000	fixes the z-displacement	<i>n</i> = 000100	fixes the x-rotation	<i>n</i> = 000010	fixes the y-rotation	<i>n</i> = 000001	fixes the z-rotation
<i>n</i> = 100000	fixes the x-displacement												
<i>n</i> = 010000	fixes the y-displacement												
<i>n</i> = 001000	fixes the z-displacement												
<i>n</i> = 000100	fixes the x-rotation												
<i>n</i> = 000010	fixes the y-rotation												
<i>n</i> = 000001	fixes the z-rotation												
fc <region> <i>n</i> <i>F</i> $v_x v_y v_z$	Specify fixed forces on <region>. <i>n</i> identifies the load curve number, <i>F</i> is the load curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the force.												
fcc <region> <i>n</i> <i>F</i> $v_x v_y v_z$	Specify fixed forces on <region> in cylindrical coordinates. <i>n</i> identifies the load curve number, <i>F</i> is the load curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the force.												
fcs <region> <i>n</i> <i>F</i> $v_x v_y v_z$	Specify fixed forces on <region> in spherical coordinates. <i>n</i> identifies the load curve number, <i>F</i> is the load curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the force.												
fd <region> <i>n</i> δ $v_x v_y v_z$	Specify fixed displacements on <region>. <i>n</i> identifies the displacement curve number, δ is the displacement curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the displacement.												

Standard Part Boundary Conditions (cont.)

Solid Mechanics Boundary Conditions (cont.)

Command	Description
fdc <i><region></i> <i>n</i> δ $v_x v_y v_z$	Specify fixed displacements on <i><region></i> in cylindrical coordinates. <i>n</i> identifies the displacement curve number, δ is the displacement curve scale factor, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the direction of the displacement.
fds <i><region></i> <i>n</i> δ $v_x v_y v_z$	Specify fixed displacements on <i><region></i> in spherical coordinates. <i>n</i> identifies the displacement curve number, δ is the displacement curve scale factor, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the direction of the displacement.
fv <i><region></i> <i>n</i> V $v_x v_y v_z$	Specify a fixed velocity on <i><region></i> . <i>n</i> identifies the velocity curve number, V is the velocity curve scale factor, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the direction of the velocity.
fvC <i><region></i> <i>n</i> V $v_x v_y v_z$	Specify a fixed velocity on <i><region></i> in cylindrical coordinates. <i>n</i> identifies the velocity curve number, V is the velocity curve scale factor, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the direction of the velocity.
fvS <i><region></i> <i>n</i> V $v_x v_y v_z$	Specify a fixed velocity on <i><region></i> in spherical coordinates. <i>n</i> identifies the velocity curve number, V is the velocity curve scale factor, and $v = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is a vector defining the direction of the velocity.
ndl <i><region></i> A $p_x p_y p_z$	Specify a distributed load on <i><region></i> . The distributed load data is then used to determine nodal forces on <i><region></i> . This command provides the only mechanism available for generating loads for GEMINI at this time. The ndl command automatically creates load cards using load curve 2 for DYNA3D. The coordinate, (p_x, p_y, p_z) , specifies a point in the local coordinate system towards which the distributed load acts.
nr <i><region></i>	Specify that <i><region></i> be a non-reflecting boundary. This option is for DYNA3D only.
nri <i><r-index></i>	Specify that <i><r-index></i> be a non-reflecting boundary. This option is for DYNA3D only.
pm <i><region></i> <i>n</i>	Specify a region with discrete masses. Here, <i>n</i> is the mass for each node in the <i><region></i> .
pr <i><region></i> <i>n</i> P $p_x p_y p_z$	Specify pressure boundary conditions on <i><region></i> . <i>n</i> identifies the pressure load curve number, P is the pressure curve scale factor, and $p = p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k}$ is a point in the local coordinate system towards which the pressure acts.

Standard Part Boundary Conditions (cont.)

Solid Mechanics Boundary Conditions (cont.)

Command	Description
pri <i><r-index> n P p_x p_y p_z</i>	Specifies application of a pressure load over <i><r-index></i> . <i>n</i> identifies the pressure load curve number, <i>P</i> is the pressure curve scale factor, and $p = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k}$ is a point in the local coordinate system towards which the pressure acts.
prth <i>n v_x v_y v_z</i>	Specify a thin shell pressure variation using the pressure curve <i>n</i> defined by the prcd command. The pressures are based upon the angle between the vector $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ and the normal to the shell surface at each nodal point.
rotation <i>p_x p_y p_z ω_x ω_y ω_z</i>	Assign an initial rigid body rotation to all parts defined after this command. (<i>p_x, p_y, p_z</i>) is any point on the axis of rotation and (<i>ω_x, ω_y, ω_z</i>) is the angular velocity vector in radians per second.
spdb <i><region> n type S</i>	Identify nodes within a standard part as belonging to a spring-damper system. Here, <i>n</i> is the spring-damper number defined using the spd command in the global control section. Each spring-damper must have a slave and master node. <i>S</i> is a scale factor, and <i>type</i> may be either s or m for slave or master respectively.
sw <i><region> n</i>	Identifies <i><region></i> as a stone wall associated with stonewall number <i>n</i> .
syf <i><region> n σ_f</i>	Specify that <i><region></i> should be a symmetry plane with failure. Here, <i>n</i> is the number of the symmetry plane, and <i>σ_f</i> is the failure stress.
syfi <i><r-index> n σ_f</i>	Specify that <i><r-index></i> should be a symmetry plane with failure. Here, <i>n</i> is the number of the symmetry plane, and <i>σ_f</i> is the failure stress.
te <i><region> T</i>	Specify that <i><region></i> have temperature <i>T</i> .
tei <i><r-index> T</i>	Specify that <i><r-index></i> have temperature <i>T</i> .
velocity <i>v_x v_y v_z</i>	Assign the rigid body velocity $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ to all parts defined after this command.
veth <i>n v_x v_y v_z</i>	Specify a thin shell velocity variation using the velocity curve <i>n</i> defined by the vecd command. The velocities are based upon the angle between the vector $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ and the normal to the shell surface at each nodal point.

Standard Part Boundary Conditions (cont.)

Thermal Loads and Boundary Conditions

Command	Description
cv <i><region> n₁ h n₂ T α</i>	Specify convection boundary conditions on <i><region></i> . <i>n₁</i> is the load curve identifier for the heat transfer convection coefficient, <i>h</i> is the convection coefficient load curve scale factor, <i>n₂</i> is the load curve identifier for the ambient temperature, <i>T</i> is the temperature load curve scale factor, and <i>α</i> is the free convection exponent.
cv+ <i><region> n₁ h n₂ T α p_x p_y p_z</i>	Specify convection boundary conditions on <i><region></i> . <i>n₁</i> is the load curve identifier for the heat transfer convection coefficient, <i>h</i> is the convection coefficient load curve scale factor, <i>n₂</i> is the load curve identifier for the ambient temperature, <i>T</i> is the temperature load curve scale factor, and <i>α</i> is the free convection exponent. (<i>p_x,p_y,p_z</i>) is a point in the local coordinate system towards which the convection surfaces face.
cv- <i><region> n₁ h n₂ T α p_x p_y p_z</i>	Specify convection boundary conditions on <i><region></i> . <i>n₁</i> is the load curve identifier for the heat transfer convection coefficient, <i>h</i> is the convection coefficient load curve scale factor, <i>n₂</i> is the load curve identifier for the ambient temperature, <i>T</i> is the temperature load curve scale factor, and <i>α</i> is the free convection exponent. (<i>p_x,p_y,p_z</i>) is a point in the local coordinate system which the convection surfaces face away from.
cvi <i><r-index> n₁ h n₂ T α</i>	Specify convection boundary conditions on <i><r-index></i> . <i>n₁</i> is the load curve identifier for the heat transfer convection coefficient, <i>h</i> is the convection coefficient load curve scale factor, <i>n₂</i> is the load curve identifier for the ambient temperature, <i>T</i> is the temperature load curve scale factor, and <i>α</i> is the free convection exponent.
fl <i><region> n q</i>	Specify a heat flux boundary condition on <i><region></i> . <i>n</i> identifies the heat flux load curve number, and <i>q</i> is the load curve scale factor.
fli <i><r-index> n q</i>	Specify a heat flux boundary condition on <i><r-index></i> . <i>n</i> identifies the heat flux load curve number, and <i>q</i> is the load curve scale factor.
ft <i><region> n T</i>	Specify a fixed temperature boundary condition on <i><region></i> . <i>n</i> identifies the temperature load curve number, and <i>T</i> is the load curve scale factor.
fli <i><r-index> n T</i>	Specify a fixed temperature boundary condition on <i><r-index></i> . <i>n</i> identifies the temperature load curve number, and <i>T</i> is the load curve scale factor.

Standard Part Boundary Conditions (cont.)

Thermal Loads and Boundary Conditions (cont.)

Command	Description
<i>Radiation Boundary Conditions ...</i>	
rb <region> $n_1 f n_2 T$	Specifies radiation boundary conditions on <region> using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T .
rb+ <region> $n_1 f n_2 T p_x p_y p_z$	Specifies radiation boundary conditions on <region> using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T . (p_x, p_y, p_z) is a point in the local coordinate system towards which the radiation surfaces face.
rb- <region> $n_1 f n_2 T p_x p_y p_z$	Specifies radiation boundary conditions on <region> using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T . (p_x, p_y, p_z) is a point in the local coordinate system which the convection surfaces face away from.
rbi <r-index> $n_1 f n_2 T$	Specifies radiation boundary conditions on <r-index> using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T .
rbc <region>	Specify radiation boundary conditions on <region>.
rbc <r-index>	Specify radiation boundary conditions on <r-index>.
<i>Radiation Enclosures ...</i>	
re <region> $n T opt$	Identify <region> as part of a radiation enclosure using n as the emissivity load curve. T is the temperature of the region, and must be input only if n is less than or equal to zero. opt may be either <i>yes</i> to include the surfaces in obstructing surface calculations, or <i>no</i> to omit the surfaces from the obstruction calculations. <i>Note that this command assumes that the surface of the radiation enclosure faces away from any adjacent conduction elements. An error occurs if radiation segments defined by this command are not adjacent to a conduction element because the outward normal for the surface segment becomes indeterminate.</i>

Standard Part Boundary Conditions (cont.)

Thermal Loads and Boundary Conditions (cont.)

Command	Description
re+ <i><region> n T opt p_x p_y p_z</i>	Identify <i><region></i> as part of a radiation enclosure using <i>n</i> as the emissivity load curve. <i>T</i> is the temperature of the region, and must be input only if <i>n</i> is less than or equal to zero. <i>opt</i> may be either yes to include the surfaces in obstructing surface calculations, or no to omit the surfaces from the obstruction calculations. (<i>p_x, p_y, p_z</i>) is a point in the local coordinate system toward which the radiation surfaces face.
re- <i><region> n T opt p_x p_y p_z</i>	Identify <i><region></i> as part of a radiation enclosure using <i>n</i> as the emissivity load curve. <i>T</i> is the temperature of the region, and must be input only if <i>n</i> is less than or equal to zero. <i>opt</i> may be either yes to include the surfaces in obstructing surface calculations, or no to omit the surfaces from the obstruction calculations. (<i>p_x, p_y, p_z</i>) is a point in the local coordinate system which the radiation surfaces faces away from.
rei <i><r-index> n T opt</i>	Identify <i><r-index></i> as part of a radiation enclosure using <i>n</i> as the emissivity load curve. <i>T</i> is the temperature of the region, and must be input only if <i>n</i> is less than or equal to zero. <i>opt</i> may be either yes to include the surfaces in obstructing surface calculations, or no to omit the surfaces from the obstruction calculations. <i>Note that this command assumes that the surface of the radiation enclosure faces away from any adjacent conduction elements. An error occurs if radiation segments defined by this command are not adjacent to a conduction element because the outward normal for the surface segment becomes indeterminant.</i>
tm <i><region> T</i>	Specify an initial temperature condition (<i>T</i>) on <i><region></i> .
vhg <i><region> n Q</i>	Specify volumetric heat generating elements in <i><region></i> . <i>n</i> specifies the load curve for the volumetric heat generation, and <i>Q</i> is the amplitude of the load curve.
vhgi <i><r-index> n Q</i>	Specify volumetric heat generating elements in <i><r-index></i> . <i>n</i> specifies the load curve for the volumetric heat generation, and <i>Q</i> is the amplitude of the load curve.

5.1.3 Beam Element Generation

The beam element generation in the standard part provides the ability to generate beam elements which coincide with the edges of brick or shell elements. The use of this type of beam element is primarily for simulation of rebar pullout in concrete. There are three primary beam generation commands, one for each of the index directions.

Command	Description
ibm <i><region></i> $n_j n_k m dir s$	Generate beam elements <i><region></i> in the j-k plane.
ibmi <i><r-index></i> $n_j n_k m dir s$	Generate beam elements in <i><r-index></i> in the j-k plane.
jbm <i><region></i> $n_i n_k m dir s$	Generate beam elements in <i><region></i> in the i-k plane.
jbmi <i><r-index></i> $n_i n_k m dir s$	Generate beam elements in <i><r-index></i> in the i-k plane.
kbm <i><region></i> $n_i n_j m dir s$	Generate beam elements in <i><region></i> in the i-j plane.
kbmi <i><r-index></i> $n_i n_j m dir s$	Generate beam elements in <i><index></i> in the i-j plane.

Parameter Meanings:

n_i	number of beams in the i-direction.
n_j	number of beams in the j-direction.
n_k	number of beams in the k-direction.
m	material number for rebar elements.
dir	orientation of rebar. j or k for ibm/ibmi. i or k for jbm/jbmi. i or j for kbm/kbmi.
s	beam section property number.

5.1.4 Edit and the In-line Calculator

The in-line calculator provides the capability to use FORTRAN like equations as arguments in commands. The **edit** command allows FORTRAN like equations involving the coordinates x , y , z and the indices i , j and k as symbolic quantities. Pressures can be edited in the edit mode, just like the coordinates using the variable p for the pressure amplitudes.

Coordinates or pressures are modified as:

$$\text{coordinate} = \text{expression}(x, y, z, i, j, k)$$

$$p = \text{expression}(x, y, z, i, j, k)$$

where *expression* is any standard FORTRAN expression, coordinate is x , y or z .

The edit command provides three temporary variables: u , v , and w , and allows the following binary operators: $+$, $-$, $*$, $/$, $**$, $^$. Paratheses may be used as delimiters for the symbolic expression. The following intrinsic functions are currently permitted: *int*, *nint*, *abs*, *mod*, *sign*, *max*, *min*, *sqrt*, *exp*, *log*, *log10*, *sin*, *cos*, *tan*, *asin*, *acos*, *atan*, *atan2*, *sinh*, *cosh*, *tanh*, *norm* and *rand*.

With the **edit** command, all calculations are performed in floating point, and all of the trigonometric functions accept arguments in degrees rather than in radians. All symbolic expressions are limited to 80 characters. If it is necessary to continue an equation to the next line, a space followed by **&** is required. In-line comments must be preceded by a space, **\$**, and a space. Parameters defined using the **parameter** command (Section 4.2.3) may be used in the FORTRAN statements when enclosed in square brackets (see Section 7.6 on Advanced Modelling Examples).

The edit command must follow all of the standard part commands, but precede the **end** statement in the **start - end** standard part definition. In the editing phase, the **fir** (or **rir**) command defines the region of the part whose coordinates will be modified. If there is no **fir** or **rir** command, then the entire part is modified according to the supplied FORTRAN expression.

The following commands may be used with the edit command as options.

Edit and the In-line Calculator (cont.)

Command	Description
edit	Begin the edit phase using the following commands or an expression.
<u>edit - Options:</u>	
cart	Convert the current part coordinates to Cartesian coordinates.
cyli	Convert the current part coordinates to cylindrical coordinates.
sphe	Convert the current part coordinates to spherical coordinates.
xcyli	Convert to cylindrical coordinates, where the x-axis is an axis of symmetry.
ycyli	Convert to cylindrical coordinates, where the y-axis is an axis of symmetry.
zcyli	Convert to cylindrical coordinates, where the z-axis is an axis of symmetry.
xsphe	Convert to spherical coordinates, where the x-axis is an axis of symmetry.
ysphe	Convert to spherical coordinates, where the y-axis is an axis of symmetry.
zsphe	Convert to spherical coordinates, where the z-axis is an axis of symmetry.
fir <region>	Specify that editing should occur on the full index space region identified by <region>. If <i>fir</i> is not used, the default for editing is the entire index space of the current standard part.
phr $x_{min} y_{min} z_{min} x_{max} y_{max} z_{max}$	Restrict the scope of any following expressions to those nodes whose physical coordinates fall within the range specified by the minimum and maximum values of x, y and z.
rir <region>	Specify that editing should occur on the reduced index space region identified by <region> for a MAZE. Here, the meaning of <region> is that of a region for the MAZE part syntax. If <i>rir</i> is not used, the default for editing is the entire index space of the current standard part.

Edit and the In-line Calculator (cont.)

Command	Description
<u>edit - Options:</u>	
rand	This command produces random perturbations to the part of the mesh identified by the fir command. It is possible to seed the random number generator by replacing rand with rand(s) where seed is a floating point number. It is also possible to specify both a seed and a mean as : rand(s, a) where <i>s</i> is the seed, and <i>a</i> is the average. By default, the seed and mean are set to 0.0.
norm	This command causes a normal perturbations to the part of the mesh identified by the fir command. Like the rand command, a seed and mean can be specified as : norm(s) , or norm(s, a) . In addition, the standard deviation can be specified as : norm(s, a, σ) . This is a pseudo-random number generator which averages 20 uniform random numbers. By default $\sigma = 1.0$, and the seed and mean are set to 0.0.

5.2 MAZE Parts

MAZE parts provide a simple means for generation of 2-D cross sections. In INGRID, these 2-D cross sections may be used for the generation of shell elements, or in the generation of 3-D solid elements. MAZE part definitions rely completely upon the line definitions detailed in section 4.2.8. The general INGRID syntax for MAZE parts are:

```
part
Cross Section Definition
Part Control
Optional Commands
end
```

Here, the keywords `part` and `end` are required for the MAZE part, and the cross section, part control and optional commands are detailed below.

Because MAZE parts begin as 2-D entities, it is necessary to define the coordinate system for the 2-D parts relative to INGRID's 3-D cartesian coordinate system. In the definition of a MAZE part, the coordinate system is referred to as (r,z) with $r = x$, and $z = z$ in INGRID's cartesian space. Figure 5.5 illustrates this coordinate system relative to INGRID's Cartesian coordinate system.

In the following description of MAZE functions, $\langle region \rangle$ will be taken to indicate a pseudo-reduced index space for the MAZE part. In such a region the first and last indices (i.e., i and k) can take on values of 1 or 2 corresponding to the endpoints of the various edges making up the 2-D MAZE part. However, the j -index may take on values consistent with the number of planes of 2-D elements generated with using the `spin` or `stack` command.

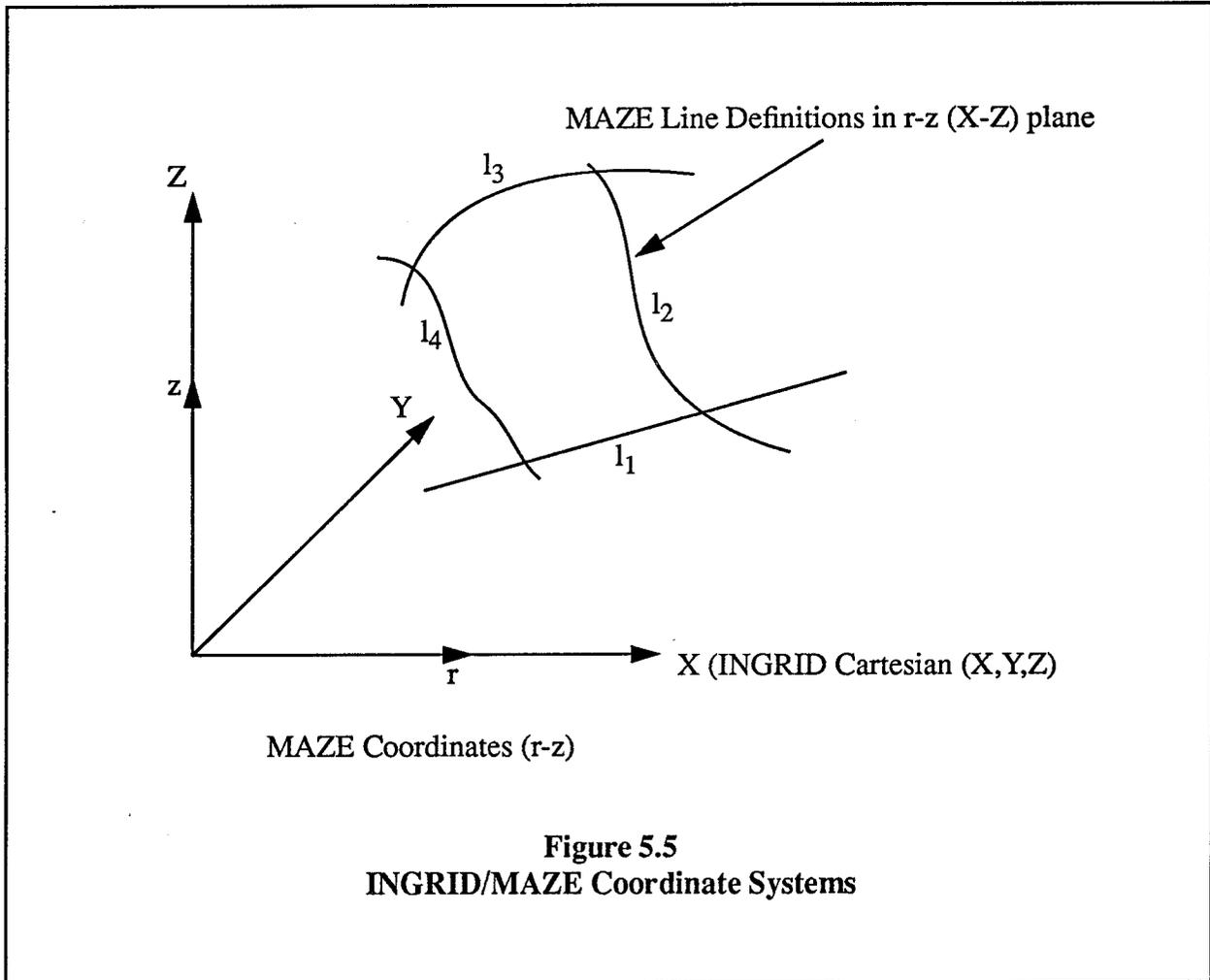


Figure 5.5
INGRID/MAZE Coordinate Systems

MAZE Part Definition

Command	Description
part	Define a new MAZE part.
end	Required end statement for part-end definition of MAZE part.

5.2.1 MAZE Cross Section Definition

When defining a four sided MAZE part, the first line (l_1 in Figure 5.2) defines the side where $k = 1$, and the third line (l_3) defines the MAZE side where $k = 2$ in index space. The second line (l_2 in Figure 5.2) forms the side of the MAZE part where $i=2$, and the fourth line (l_4) defines the MAZE side where $i = 1$. The original r-z (or X-Z) plane forms the surface where $j = 1$. When defining a three sided MAZE part, the fourth side degenerates to a point. For more details regarding regions, refer to Chapter 3 on Basic Concepts.

MAZE Cross Section Definition

Command	Description
$l_1 l_2 l_3 l_4 mat n m$	Define a 4 sided region with edges consisting of the intersecting lines l_1, l_2, l_3 , and l_4 . This region will have a material number of mat , and will have n subdivisions along l_1 and l_3 , with m subdivisions along l_2 and l_4 . Line definitions must be referenced in a counter-clockwise order. If n is zero, then the number of subdivision along l_1 (l_3) is assumed to be one less than the number of points used to define l_1 . If m is zero, then the number of subdivisions along l_2 (l_4) is assumed to be one less than the number of points used to define l_2 . nIf n or m are zero, the user can effectively force points used in the line definition to be used as nodal points at the boundary of the MAZE region.
$l_1 l_2 l_3 l_3 mat n m$	Define a 3 sided region with edges consisting of the intersecting lines l_1, l_2 , and l_3 . This region will have a material number of mat , and be subdivided into $m \times (2n+m)$ elements with $(n+m)$ elements along edges l_1 and l_2 , and $2m$ elements along l_3 .
$l_1 l_2 l_3 0 mat n m$	Define a 3 sided region with edges consisting of the intersecting lines l_1, l_2 , and l_3 . This region will have a material number of mat , and be subdivided into $m \times (2n+m)$ elements with $(n+m)$ elements along edges l_1 and l_2 , and $2m$ elements along l_3 .
$l_1 l_2 l_3 l_4 mat -n -m r_1 r_2$	Define a 4 sided region with edges consisting of the intersecting lines l_1, l_2, l_3 , and l_4 . This region will have a material number of mat , and will have n subdivisions along l_1 and l_3 , with m subdivisions along l_2 and l_4 . r_1 and r_2 are the ratios of the first subdivision to the last subdivision on the l_1 (l_3) and l_2 (l_4) sides respectively.
$l_1 l_2 l_3 l_4 mat -n m r_1$	Define a 4 sided region with edges consisting of the intersecting lines l_1, l_2, l_3 , and l_4 . This region will have a material number of mat , and will have n subdivisions along l_1 and l_3 , with m subdivisions along l_2 and l_4 . r_1 is the ratio of the first subdivision to the last subdivision on the l_1 (l_3) sides.

MAZE Cross Section Definition (cont.)

Command	Description
$l_1 l_2 l_3 l_4 mat n -m r_2$	Define a 4 sided region with edges consisting of the intersecting lines $l_1, l_2, l_3,$ and l_4 . This region will have a material number of mat , and will have n subdivisions along l_1 and l_3 , with m subdivisions along l_2 and l_4 . r_2 is the ratio of the first subdivision to the last subdivision on the l_2 (l_4) sides.
$l_1 l_2 l_3 l_4 -mat n m r_1 r_2 r_3 r_4$	Define a 4 sided region with edges consisting of the intersecting lines $l_1, l_2, l_3,$ and l_4 . This region will have a material number of mat , and will have n subdivisions along l_1 and l_3 , with m subdivisions along l_2 and l_4 . r_1, r_2, r_3, r_4 are the ratios of the first subdivision to the last subdivision on the $l_1, l_2, l_3,$ and l_4 sides respectively.

MAZE Part Control

Command	Description
drag options ;	Perform a dragging operation to generate solid elements from shell elements.
<u>drag - Options:</u>	
move n data	Form n layers of solid elements by moving the original shell elements to the location specified by <i>data</i> .
res r	The ratio of one element length to the next adjacent element length is r . This applied only to the previous drag operation.
rota $p_x p_y p_z \omega_x \omega_y \omega_z \theta$	Form n layers of solid elements by rotating the original shell elements about an axis. (p_x, p_y, p_z) is any point on the axis of rotation and $(\omega_x, \omega_y, \omega_z)$ is a vector parallel to the axis of rotation. The angle of rotation (θ) is in degrees.
epb <point> option;	Setup an element print block.
<u>epb - options:</u>	
po i j k	Use the element offset from <point> by (i,j,k) in reduced index space in the element print block.
ro <region>	Use the block of elements in <region> offset from <point> in the element print block.

MAZE Part Control (cont.)

Command	Description
lct $n ; t_1 ; t_2 ; \dots ; t_n ;$	Perform a sequence of n local coordinate transformations $t_1 \dots t_n$. Acceptable transformations used to define $t_1 \dots t_n$ are listed below, and are required. This command must precede lrep for local part replication. lct and lrep are intended to be used as companion commands for performing coordinate transformations and part replications.
<u>lct - Required data for transformations:</u>	
cscs s	Scale all nodal coordinates by s .
mx Δx	Translate by Δx .
my Δy	Translate by Δy .
mz Δz	Translate by Δz .
v $\Delta x \Delta y \Delta z$	Translate by $\Delta x, \Delta y, \Delta z$.
repe m	Repeat the local coordinate transformation m times.
rx θ_x	Rotate by θ_x^0 about the x-axis.
ry θ_y	Rotate by θ_y^0 about the y-axis.
rz θ_z	Rotate by θ_z^0 about the z-axis.
rx y	Reflect the part about the x-y plane.
ry z	Reflect the part about the y-z plane.
rz x	Reflect the part about the z-x plane.
save m	The sequence of coordinate transformations is generated starting from coordinate system m when using the repe command.
xcscs s	Scale all nodal x-coordinates by s .
ycscs s	Scale all nodal y-coordinates by s .
zcscs s	Scale all nodal z-coordinates by s .
lrep $0 1 \dots n ;$	The lrep command performs local part replication, and is intended to be used with the lct command. Here, the 0 indicates that the original instance of the part should be kept in the finite element mode. The sequence of 1 to n applies the lct coordinate transformations to the part keeping each instance of the transformed part. For example, lrep 0 1 2; would keep the original part, and apply the local transformations 1 and 2 from the lct command: lct 2; mx Δx; my Δy;
mate n	The imported part has material number n .
repe $c_1 c_2 \dots c_n ;$	Repeat command. This command makes copies of the part in each of the global coordinate systems $c_1 \dots c_n$. If c_i is zero, then the part is replicated without transformation.
rotation $p_x p_y p_z \omega_x \omega_y \omega_z$	Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and $(\omega_x, \omega_y, \omega_z)$ is the angular velocity vector in radians per second.
sphe	Nodal point coordinates are converted from spherical to Cartesian coordinates as: $x = r \cos(\theta) \sin(\psi)$, $y = r \sin(\theta) \sin(\psi)$, $z = r \cos(\psi)$. Please refer to Figure 2.5.

MAZE Part Control (cont.)

Command	Description
spin $n \theta$	Generate 3-D solid elements by performing a spin operation about the z-axis. The total number of nodes is n , and the total rotation about the z-axis is θ .
stack $n d$	Generate 3-D solid elements by performing a stacking operation in the Y-direction. The number of layers of nodes is n , and the total depth of the part is d .
thick t	Shells have a thickness of t for this part.
temp T	The initial temperature is T .
velocity $v_x v_y v_z$	Assign the rigid body velocity $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ to all parts defined after this command.
t12	This command must occur just before the part command, and has the effect that the third side of the MAZE part (l_3) will have exactly twice as many elements as side l_1 . The element transition is achieved with quadrilateral elements. This command does not apply to triangular parts.
t13	This command must occur just before the part command, and has the effect that the third side of the MAZE part (l_3) will have exactly three times as many elements as side l_1 . The element transition is achieved with quadrilateral elements. This command does not apply to triangular parts.

MAZE Part Boundary Conditions

Electromagnetic Loads and Boundary Conditions

Command	Description
fbc <i><region> type</i>	Specify an electromagnetic boundary condition on <i><region></i> where <i>type</i> may be one of the following.
<u>fbc - types:</u>	
metal	metal electric conductor.
amet	magnetic conductor.
zero	zero out fields.
00	E and H fields are zero - for use when specifying total fields.
01	The normal component of H is set to zero, and E is set to zero.
02	The tangential component of H is set to zero, and E is set to zero.
03	E is set to zero.
04	The normal component of E is set to zero, and H is set to zero.
05	The normal components of E and H are set to zero.
06	The tangential components of E and H are set to zero.
07	The normal component of E is set to zero.
10	The tangential component of E is set to zero, and H is set to zero.
11	The tangential component of E and the normal component of H is set to zero.
12	The tangential components of E and H are set to zero.
13	The tangential component of E is set to zero.
14	H is set to zero.
15	The normal component of H is set to zero.
16	The tangential component of H is set to zero.
17	No fields are set to zero.
spc <i><region> n</i>	Specify a current on <i><region></i> using load curve <i>n</i> .
spf <i><region> n</i>	Specify a field on <i><region></i> using load curve <i>n</i> .

MAZE Part Boundary Conditions (cont.)

Electrostatic Loads and Boundary Conditions (cont.)

Command	Description
$\text{efl } \langle \text{region} \rangle f$	Specify an electrostatic flux, f , on $\langle \text{region} \rangle$.
$\text{v } \langle \text{region} \rangle \phi$	Specify an electrostatic potential, ϕ , on $\langle \text{region} \rangle$.

Sliding Interface Definitions

Command	Description
$\text{si } \langle \text{region} \rangle n \text{ type options}$	Specifies that $\langle \text{region} \rangle$ should be a slide surface of <i>type</i> using the global slide surface definition n .
<u>si - type:</u>	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
<u>si - Options for slave surface only:</u>	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the global slide surface <i>type</i> is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.
$\text{si+ } \langle \text{region} \rangle n \text{ type } p_x p_y p_z$	Specifies that $\langle \text{region} \rangle$ should be a slide surface of <i>type</i> using the global slide surface definition n . (p_x, p_y, p_z) specifies a point in the local coordinate system towards which the sliding interface faces.
<u>si+ - type:</u>	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
<u>si+ - Options for slave surface only:</u>	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the slide surface <i>type</i> is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.

MAZE Part Boundary Conditions (cont.)

Sliding Interface Definitions (cont.)

Command	Description
si- $\langle region \rangle$ n type $p_x p_y p_z$	Specifies that $\langle region \rangle$ should be a slide surface of <i>type</i> using the global slide surface definition n . (p_x, p_y, p_z) specifies a point in the local coordinate system which the sliding interface faces away from.
si- - type:	
m	Designates the slide surface as a master.
s	Designates the slide surface as a slave.
si- - Options for slave surface only:	
$f_n f_s \alpha \beta$	If the global slide surface type is nsw , then the normal force at failure, f_n , the shear force at failure, f_s , and the exponents for the normal and shear forces (α and β respectively) must be specified for the slave slide surface.
$\sigma_n \sigma_s$	If the global slide surface type is break , then the normal failure stress, σ_n , and the shear failure stress, σ_s , must be specified for the slave slide surface.

Solid Mechanics Boundary Conditions

Command	Description												
acc $\langle region \rangle$ n A $a_x a_y a_z$	Specify acceleration on $\langle region \rangle$. n identifies the load curve number, A is the load curve scale factor, and $a = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
acc $\langle region \rangle$ n A $a_x a_y a_z$	Specify acceleration on $\langle region \rangle$ in cylindrical coordinates. n identifies the load curve number, A is the load curve scale factor, and $a = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
accs $\langle region \rangle$ n A $a_x a_y a_z$	Specify acceleration on $\langle region \rangle$ in spherical coordinates. n identifies the load curve number, A is the load curve scale factor, and $a = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$ is a vector defining the direction of the acceleration.												
b $\langle region \rangle$ n	Specifies displacement boundary conditions on $\langle region \rangle$ where n is a six digit number specifying the constrained degree of freedom. <table style="margin-left: 40px;"> <tr> <td>$n = 100000$</td> <td>fixes the x-displacement</td> </tr> <tr> <td>$n = 010000$</td> <td>fixes the y-displacement</td> </tr> <tr> <td>$n = 001000$</td> <td>fixes the z-displacement</td> </tr> <tr> <td>$n = 000100$</td> <td>fixes the x-rotation</td> </tr> <tr> <td>$n = 000010$</td> <td>fixes the y-rotation</td> </tr> <tr> <td>$n = 000001$</td> <td>fixes the z-rotation</td> </tr> </table>	$n = 100000$	fixes the x-displacement	$n = 010000$	fixes the y-displacement	$n = 001000$	fixes the z-displacement	$n = 000100$	fixes the x-rotation	$n = 000010$	fixes the y-rotation	$n = 000001$	fixes the z-rotation
$n = 100000$	fixes the x-displacement												
$n = 010000$	fixes the y-displacement												
$n = 001000$	fixes the z-displacement												
$n = 000100$	fixes the x-rotation												
$n = 000010$	fixes the y-rotation												
$n = 000001$	fixes the z-rotation												

MAZE Part Boundary Conditions (cont.)

Solid Mechanics Boundary Conditions (cont.)

Command	Description
fc <i><region></i> <i>n</i> <i>F</i> $v_x v_y v_z$	Specify fixed forces on <i><region></i> . <i>n</i> identifies the load curve number, <i>F</i> is the load curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the force.
fd <i><region></i> <i>n</i> δ $v_x v_y v_z$	Specify fixed displacements on <i><region></i> . <i>n</i> identifies the displacement curve number, δ is the displacement curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the displacement.
fv <i><region></i> <i>n</i> <i>V</i> $v_x v_y v_z$	Specify a fixed velocity on <i><region></i> . <i>n</i> identifies the velocity curve number, <i>V</i> is the velocity curve scale factor, and $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ is a vector defining the direction of the velocity.
pr <i><region></i> <i>n</i> <i>P</i> $p_x p_y p_z$	Specify pressure boundary conditions on <i><region></i> . <i>n</i> identifies the pressure curve number, <i>P</i> is the pressure curve scale factor, and (p_x, p_y, p_z) is a point in the local coordinate system towards which the pressure acts.
sw <i><region></i> <i>n</i>	Identifies <i><region></i> as a stone wall associated with stonewall number <i>n</i> .

Thermal Loads and Boundary Conditions

Command	Description
cv <i><region></i> n_1 <i>h</i> n_2 <i>T</i> α	Specify convection boundary conditions on <i><region></i> . n_1 is the load curve identifier for the heat transfer convection coefficient, <i>h</i> is the convection coefficient load curve scale factor, n_2 is the load curve identifier for the ambient temperature, <i>T</i> is the temperature load curve scale factor, and α is the free convection exponent.
cv+ <i><region></i> n_1 <i>h</i> n_2 <i>T</i> α $p_x p_y p_z$	Specify convection boundary conditions on <i><region></i> . n_1 is the load curve identifier for the heat transfer convection coefficient, <i>h</i> is the convection coefficient load curve scale factor, n_2 is the load curve identifier for the ambient temperature, <i>T</i> is the temperature load curve scale factor, and α is the free convection exponent. (p_x, p_y, p_z) is a point in the local coordinate system towards which the convection surfaces face.

MAZE Part Boundary Conditions (cont.)

Thermal Loads and Boundary Conditions (cont.)

Command	Description
cv- $\langle region \rangle n_1 h n_2 T \alpha p_x p_y p_z$	Specify convection boundary conditions on $\langle region \rangle$. n_1 is the load curve identifier for the heat transfer convection coefficient, h is the convection coefficient load curve scale factor, n_2 is the load curve identifier for the ambient temperature, T is the temperature load curve scale factor, and α is the free convection exponent. (p_x, p_y, p_z) is a point in the local coordinate system which the convection surfaces face away from.
fl $\langle region \rangle n q$	Specify a heat flux boundary condition on $\langle region \rangle$. n identifies the heat flux load curve number, and q is the load curve scale factor.
ft $\langle region \rangle n T$	Specify a fixed temperature boundary condition on $\langle region \rangle$. n identifies the temperature load curve number, and T is the load curve scale factor.
rb $\langle region \rangle n_1 f n_2 T$	Specifies radiation boundary conditions on $\langle region \rangle$ using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T .
rb+ $\langle region \rangle n_1 f n_2 T p_x p_y p_z$	Specifies radiation boundary conditions on $\langle region \rangle$ using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T . (p_x, p_y, p_z) is a point in the local coordinate system towards which the radiation surfaces face.
rb- $\langle region \rangle n_1 f n_2 T p_x p_y p_z$	Specifies radiation boundary conditions on $\langle region \rangle$ using load curve n_1 with scale factor f , and ambient temperature load curve n_2 with scale factor T . (p_x, p_y, p_z) is a point in the local coordinate system which the convection surfaces face away from.
rbc $\langle region \rangle$	Specify radiation boundary conditions on $\langle region \rangle$.
re $\langle region \rangle n T opt$	Identify $\langle region \rangle$ as part of a radiation enclosure using n as the emissivity load curve. T is the temperature of the region, and must be input only if n is less than or equal to zero. opt may be either yes to include the surfaces in obstructing surface calculations, or no to omit the surfaces from the obstruction calculations. Note that this command assumes that the surface of the radiation enclosure faces away from any adjacent conduction elements. An error occurs if radiation segments defined by this command are not adjacent to a conduction element because the outward normal for the surface segment becomes indeterminate.

MAZE Part Boundary Conditions (cont.)

Thermal Loads and Boundary Conditions (cont.)

Command	Description
re+ $\langle region \rangle n T opt p_x p_y p_z$	Identify $\langle region \rangle$ as part of a radiation enclosure using n as the emissivity load curve. T is the temperature of the region, and must be input only if n is less than or equal to zero. opt may be either yes to include the surfaces in obstructing surface calculations, or no to omit the surfaces from the obstruction calculations. $p = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k}$ is a point in the local coordinate system toward which the radiation surfaces face.
re- $\langle region \rangle n T opt p_x p_y p_z$	Identify $\langle region \rangle$ as part of a radiation enclosure using n as the emissivity load curve. T is the temperature of the region, and must be input only if n is less than or equal to zero. opt may be either yes to include the surfaces in obstructing surface calculations, or no to omit the surfaces from the obstruction calculations. $p = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k}$ is a point in the local coordinate system which the radiation surfaces face away from.
tm $\langle region \rangle T$	Specify an initial temperature condition (T) on $\langle region \rangle$.

5.3 Beam Parts

The INGRID beam part provides a primitive mechanism for constructing beam elements in an INGRID mesh. This part requires the specification of three spatial locations to sufficiently describe the beam segment to be discretized. Figure 5.6 below illustrates the beam part in local coordinates (r,s,t). A beginning and ending node (n_1 and n_2) along with a reference node must be specified in order to uniquely describe the beam element generation process. The general syntax for the beam part is as follows:

```

beam
Coordinate Specification
0
Element Generation Commands
0
Options
end

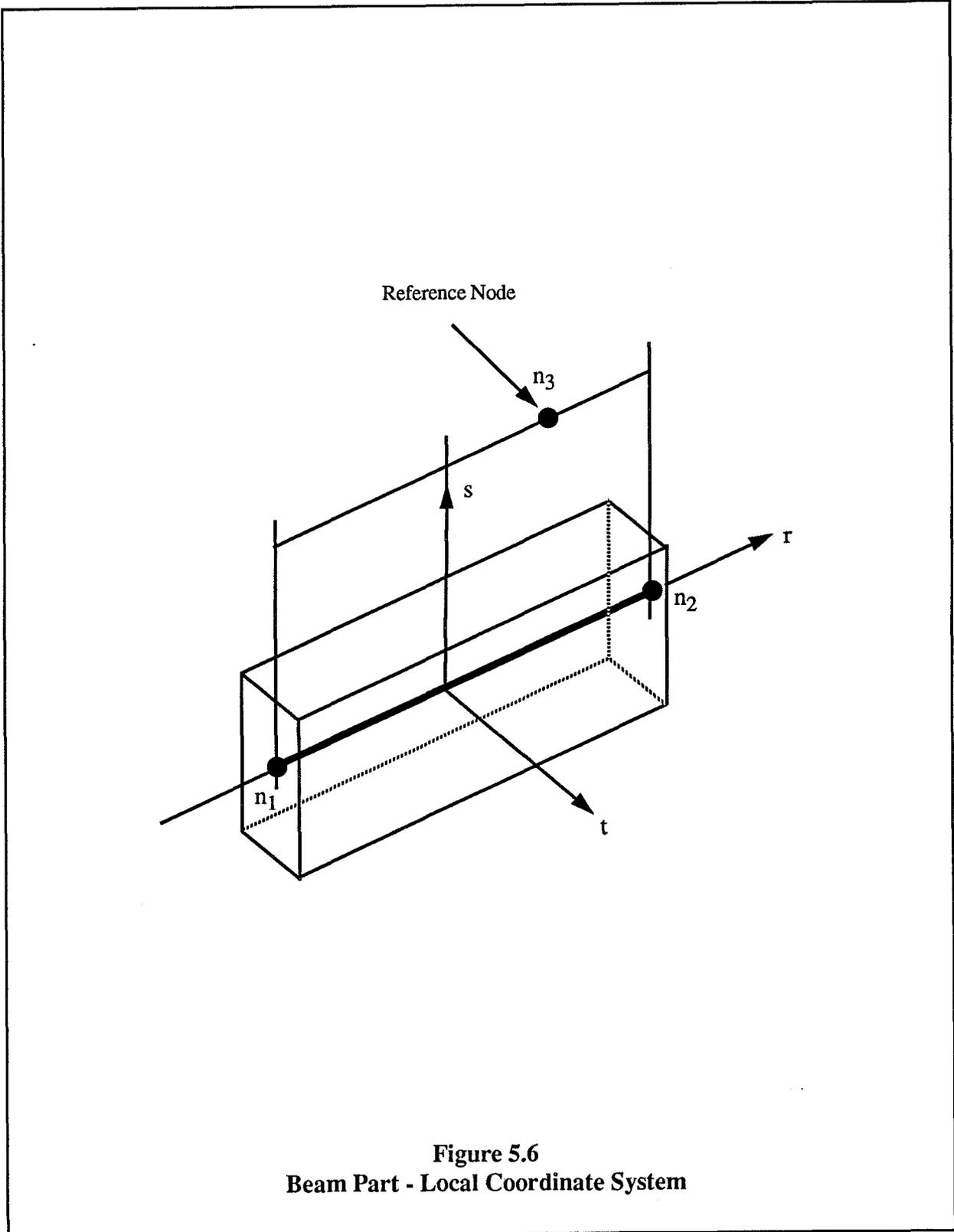
```

The necessary spatial coordinates and boundary conditions are entered in the *Coordinate Specification* section of the beam part. All coordinates in this section are assigned numeric indices beginning with 1 for the first input coordinate. The following sections describe the INGRID syntax for the *Coordinate Specification*, *Element Generation Commands*, and *Options* sections of the beam part.

Beam Part Definition

Command	Description
beam	Define a new beam part.
end	Required end statement for beam-end definition of beam part.

Beam Parts (cont.)



Beam Parts (cont.)Coordinate Specification

Command	Description
<i>ct n x y z</i>	<p>Acceptable values for <i>ct</i> are rt for a Cartesian coordinate transformation, cy for a cylindrical transformation, or sp for a spherical transformation. point coordinates. <i>n</i> is a six digit number which specifies constrained degrees of freedom (0 for free, 1 for fixed).</p> <p><i>n</i> = 100000 fixes the x-displacement <i>n</i> = 010000 fixes the y-displacement <i>n</i> = 001000 fixes the z-displacement <i>n</i> = 000100 fixes the x-rotation <i>n</i> = 000010 fixes the y-rotation <i>n</i> = 000001 fixes the z-rotation</p> <p>x,y,z values for the spatial coordinate must also be input for the beam part.</p>

Element Generation Commands

Command	Description
<i>n₁</i>	First beam index to be used in the current beam part. This corresponds to the starting point of the beam (Figure 5.6).
<i>n₂</i>	Last beam index to be used in the current beam part. This corresponds to the ending point of the beam (Figure 5.6).
<i>N</i>	Number of beam elements to generate.
<i>material</i>	Material number for the beam elements.
<i>section</i>	Section material number for the beam elements.
<i>n₃</i>	Reference mpde for defining the orientation of the beam elements. Note that this point can be moved using the <i>repe</i> command, and is not necessarily in global coordinates. The spatial coordinates for this node must be distinct from those for <i>n₁</i> and <i>n₂</i> (Figure 5.6).

Beam Parts (cont.)

Beam Part Options

Command	Description
cyli	Nodes are converted from cylindrical to Cartesian coordinates.
rotation $p_x p_y p_z \omega_x \omega_y \omega_z$	Assign an initial rigid body rotation to all parts defined after this command. (p_x, p_y, p_z) is any point on the axis of rotation and $(\omega_x, \omega_y, \omega_z)$ is the angular velocity vector in radians per second.
sphe	Nodes are converted from spherical to Cartesian coordinates.
temp T	The initial temperature is T .
velocity $v_x v_y v_z$	Assign the rigid body velocity $\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ to all parts defined

Beam Element Attributes

Command	Description
bsd n	Specify a beam section definition number n .

Joint Definition

Command	Description
jt $jd jn bn j_i$	Joint definition jd using joint local node number jn for joint beam number bn with increment j_i .
lrep $c_1 c_2 \dots c_n$;	Local repeat command. This command makes copies of the part in each of the local coordinate systems $c_1 \dots c_n$. If the c_i is zero, then the part is replicated without transformation.
repe $c_1 c_2 \dots c_n$;	Repeat command. This command makes copies of the part in each of the global coordinate systems $c_1 \dots c_n$. If c_i is zero, then the part is replicated without transformation.

Beam Parts (cont.)**Slide Surface Specification**

Command	Description
si n 1d	Specifies that the beam elements are to be associated with slide line <i>n</i> . 1d indicates 1-D slide lines are in effect for use with NIKE3D.
si n dd $F_n F_t e_n e_t$	Specifies that the beam elements are to be associated with slide line <i>n</i> . Normal and shear failure forces (F_n and F_t) with exponents (e_n and e_t) are to be used for the 1-D slide line with DYNA3D.

5.4 Importing Existing Mesh Parts

This part permits existing meshes to be imported into INGRID. The element types in the imported mesh part must be of a type consistent with INGRID's internal generation capabilities. That is, meshes containing beams, shells and hexahedral elements may be imported. If the material specified for the imported part has not been defined in the INGRID input file, then it will be impossible to graphically display the imported mesh part. The general syntax for importing existing mesh parts is:

```
import
commands
...
options
end
```

All data for the imported mesh part must exist in the INGRID input file, and must begin with the **import** command, and terminate with an end command. The following sections detail the INGRID syntax for importing existing mesh parts.

Import Part Definition

<u>Command</u>	<u>Description</u>
import	Define a new import part.
end	Required end statement for import-end definition of import part.

Import Part Commands

<u>Command</u>	<u>Description</u>
nodes <i>n options</i> ;	<i>n</i> nodal points are to be read in.
<u>nodes - Options:</u>	
form <i>format</i>	Specifies that nodal points are to be read with a user defined format. If this option is omitted, then all nodal point data is read in using the FORTRAN format free specification. <i>format</i> may contain the following key words to identify the order in which nodal point data is to be read.
number	Node numbers are to be read in. If this option is omitted, node numbers are assigned sequentially.
k	Node increments are to be read.
bcnd	NIKE3D displacement boundary codes are to be read.
bcnr	NIKE3D rotational boundary codes are to be read.
bcsp	SAP boundary codes are to be read.
x	X-coordinate is to be read.
y	Y-coordinate is to be read.
z	Z-coordinate is to be read.
t	Temperature is to be read.
dummy	Read, but ignore a given numeric field. This option is useful for skipping certain columns of the input nodal point data.
<u>Required Data:</u>	
<i>Nodal point data</i>	Nodal point data formatted according to the <i>format</i> specification above must immediately follow the nodes command.

Import Part Commands (cont.)

Command	Description
beams <i>n options</i> ;	<i>n</i> beam elements are to be read.
<u>beams - Options:</u>	
form <i>format</i>	Specifies that beam elements are to be read with a user defined format. If this option is omitted, then all beam element data is read in using the FORTRAN format free specification. <i>format</i> may contain the following key words to identify the order in which beam element data is to be read.
number	Element numbers are to be read. If this option is not used, then element numbers are assigned sequentially.
k	Element increment values are to be read.
mate	Material numbers are to be read.
section	Section material numbers are to be read.
nodes	Three node numbers which define the local coordinates (<i>r,s,t</i>) are to be read in sequential order (<i>n</i> ₁ , <i>n</i> ₂ , <i>n</i> ₃).
n₁	Read node <i>n</i> ₁ .
n₂	Read node <i>n</i> ₂ .
n₃	Read node <i>n</i> ₃ .
dummy	Read, but ignore a given numeric field. This option is useful for skipping certain columns of the input beam element data.
<u>Required Data:</u>	
<i>Beam element data</i>	Beam element data formatted according to the <i>format</i> specification must immediately follow the beams command.

Import Part Commands (cont.)

Command	Description
shells <i>n options</i> ;	<i>n</i> shell elements are to be read.
<u>shells - Options:</u>	
form <i>format</i>	Specifies that shell elements are to be read with a user defined format. If this option is omitted, then all shell element data is read in using the FORTRAN format free specification. <i>format</i> may contain the following key words to identify the order in which shell element data is to be read.
number	Element numbers are to be read. If this option is not used, then element numbers are assigned sequentially.
k	Element increment values are to be read.
mate	Material numbers are to be read.
thickness	Shell element thicknesses are to be read.
nodes	Four node numbers which define the shell orientation are to be read in sequential order (<i>n</i> ₁ , <i>n</i> ₂ , <i>n</i> ₃ , <i>n</i> ₄).
n ₁	Read node n ₁ .
n ₂	Read node n ₂ .
n ₃	Read node n ₃ .
n ₄	Read node n ₄ .
dummy	Read, but ignore a given numeric field. This option is useful for skipping certain columns of the input shell element data.
<u>Required Data:</u>	
<i>Shell element data</i>	Shell element data formatted according to the <i>format</i> specification must immediately follow the shells command.

Import Part Commands (cont.)

Command	Description
bricks <i>n options</i> ;	<i>n</i> 8-node brick elements are to be read.
<u>bricks - Options:</u>	
form <i>format</i>	Specifies that solid elements are to be read with a user defined format. If this option is omitted, then all solid element data is read in using the FORTRAN format free specification. <i>format</i> may contain the following key words to identify the order in which solid element data is to be read.
number	Element numbers are to be read. If this option is not used, then element numbers are assigned sequentially.
k	Element increment values are to be read.
mate	Material numbers are to be read.
nodes	Eight node numbers which define the brick orientation are to be read in sequential order ($n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8$).
n₁	Read node n₁ .
n₂	Read node n₂ .
n₃	Read node n₃ .
n₄	Read node n₄ .
n₅	Read node n₅ .
n₆	Read node n₆ .
n₇	Read node n₇ .
n₈	Read node n₈ .
dummy	Read, but ignore a given numeric field. This option is useful for skipping certain columns of the input solid element data.
<u>Required Data:</u>	
<i>Solid element data</i>	Solid element data formatted according to the format specification must immediately follow the bricks command.

Import Part Commands (cont.)

Import Part Control Commands

Command	Description
cyl	Nodal point coordinates are converted from cylindrical to Cartesian coordinates as: $x = r \cos(\theta)$, $y = r \sin(\theta)$. Please refer to Figure 2.5
lrep $l_1 l_2 \dots l_n$;	Local repeat command. This command makes copies of the part in each of the local coordinate systems $l_1 \dots l_n$. If the l_i is zero, then the part is replicated without transformation.
mate n	The imported part has material number n .
repe $l_1 l_2 \dots l_n$;	Repeat command. This command makes copies of the part in each of the global coordinate systems $l_1 \dots l_n$. If l_i is zero, then the part is replicated without transformation.
rotation $p_x p_y p_z \omega_x \omega_y \omega_z$	Assign an initial rigid body rotation to the part. (p_x, p_y, p_z) is any point on the axis of rotation and $(\omega_x, \omega_y, \omega_z)$ is the angular velocity vector in radians per second.
sphe	Nodal point coordinates are converted from spherical to Cartesian coordinates as: $x = r \cos(\theta) \sin(\psi)$, $y = r \sin(\theta) \sin(\psi)$, $z = r \cos(\psi)$. Please refer to Figure 2.5
thic t	Shells have a thickness of t for this part.
temp T	The initial temperature is T .
velocity $v_x v_y v_z$	Assign the rigid body velocity $v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ to all parts defined after this command.

5.5 Interactive Commands

The following pages present a concise summary of all of the interactive INGRID commands. The **di** commands require the use of the **disp** command to generate an image which accurately displays the glyphs associated with a specific model feature. If the information presented in this summary is not adequate, refer to the more elaborate descriptions in the Interactive Commands section of the INGRID User Guide.

Display Commands

Command	Description
di off	Turn off the diagnostic node highlighting.
disize <i>s</i>	Scale the glyphs used to identify slide surface, normals, etc. by the scale factor <i>s</i> .
disp	Generates an image of the mesh using the best and most expensive z-buffer algorithm. This command must be used to display any of the di options.
draw	Displays the mesh with all element edges drawn.
grid	Activate a reference grid which is displayed with the mesh.
gridoff	De-activate the reference grid.
help <i>options</i>	This command prints out terse help information regarding the interactive commands. When the help option is a specific command, then the command syntax is displayed on the screen. Otherwise, the following help categories may be specified.
<u>help - Options:</u> conditions diagnostics graphics lines list materials merging output parts transforms	Information about viewing initial or boundary conditions. Information about the model characteristics. Display options. Displaying lines and load curves. List all the interactive commands available. Material manipulation commands. Part tolerancing commands. Options for producing INGRID output. Part manipulation commands. Image manipulation commands.
pause <i>n</i>	Pause for approximately <i>n</i> seconds.

Interactive Commands (cont.)Display Commands (cont.)

Command	Description
poor	Displays the mesh using back face culling on a part-by-part basis.
reso <i>n</i>	Specify the z-buffer resolution for the disp command. <i>n</i> must be an integer multiply of 512. The default z-buffer resolution is 512x512. If the details of a mesh are not clearly resolved, increasing the z-buffer resolution can in many instances produce a image where the mesh details are resolved.
set <i>command options</i>	Set the defaults for certain commands. The allowable commands and options are:
tv <i>options</i>	Here, <i>options</i> can be set to : display, draw, none, poor or view . This command is not to be confused with tv below.
grid <i>color</i>	Here, the grid color may be set to one of : white, red, green, blue, cyan, magenta, or yellow . This applies only to color display surfaces.
marker <i>color</i>	Here, the mesh marker color may be set to one of : white, red, green, blue, cyan, magenta, or yellow . This applies only to color display surfaces.
mesh <i>color</i>	Here, the mesh color may be set to one of : white, red, green, blue, cyan, magenta, or yellow . This applies only to color display surfaces.
rax <i>color</i>	Here, the color used to display the r-axis may be set to one of : white, red, green, blue, cyan, magenta, or yellow . This applies only to color display surfaces.
sax <i>color</i>	Here, the color used to display the s-axis may be set to one of : white, red, green, blue, cyan, magenta, or yellow . This applies only to color display surfaces.
tax <i>color</i>	Here, the color used to display the t-axis may be set to one of : white, red, green, blue, cyan, magenta, or yellow . This applies only to color display surfaces.
tv <i>n</i>	Specify the graphics device driver to be used. In the case of the standard workstation implementation, DIGLIB provides the following drivers. <i>n</i> = 1 : X window system <i>n</i> = 2 : Portrait PostScript <i>n</i> = 3 : Landscape PostScript <i>n</i> = 4 : Portrait Color PostScript <i>n</i> = 5 : Landscape Color PostScript

Interactive Commands (cont.)

Display Commands (cont.)

Command	Description
view	Generates an image of the mesh using a crude z-buffer algorithm.

Rotation Commands

Command	Description
rx θ	The picture is rotated about the x-axis through an angle of θ degrees relative to the current picture orientation.
ry θ	The picture is rotated about the y-axis through an angle of θ degrees relative to the current picture orientation.
rz θ	The picture is rotated about the z-axis through an angle of θ degrees relative to the current picture orientation.
fix	Fixes the current center of rotation.
unfix	Releases the fixed center of rotation established by fix .

Translational Commands

Command	Description
u y	The picture is shifted up by y , where y represents a fraction of the screen width.
d y	The picture is shifted down by y , where y represents a fraction of the screen width.
l x	The picture is shifted left by x , where x represents a fraction of the screen width.
r x	The picture is shifted right by x , where x represents a fraction of the screen width.
trans $x y z$	Set the center of the picture to the coordinates (x,y,z) .

Interactive Commands (cont.)

Zooming and Panning Commands

<u>Command</u>	<u>Description</u>
restore	Reset all rotations and magnifications to zero and normalize the display to the parts in the active display list.
trans $x\ y\ z$	The center of the current picture is set to coordinates (x,y,z) of the model.

Magnification Commands

<u>Command</u>	<u>Description</u>
zf s	The magnification level is increased by a factor of s .
zb s	The magnification level is decreased by a factor of s .

Scaling Commands

<u>Command</u>	<u>Description</u>
xscale s_x	Apply scale factor s_x to all of the x coordinates.
yscale s_y	Apply scale factor s_y to all of the y coordinates.
zscale s_z	Apply scale factor s_z to all of the z coordinates.
scale s	Apply scale factor s to all coordinates.

Interactive Commands (cont.)

Part and Materials Display Commands

<u>Command</u>	<u>Description</u>
dap	All of the parts are placed in the display list.
dps <i>n</i>	A subset of the display list is selected. <i>n</i> parts, are placed in the display list.
p <i>n</i>	Only part number <i>n</i> is placed in the display list.
ap <i>n</i>	Part number <i>n</i> is added to the display list.
rp <i>n</i>	Part number <i>n</i> is removed from the display list.
parts <i>n</i>	Similar in function to the dap command. A subset of the part display list is selected and placed in the display list. If <i>n</i> is 0, parts functions exactly like dap and all of the parts are displayed.
dms <i>n</i>	A subset of the materials display list is selected.
dam	All of the materials are placed in the materials display list.
m <i>n</i>	Only material number <i>n</i> is placed in the materials display list.
am <i>n</i>	Material number <i>n</i> is added to the materials display list.
rm <i>n</i>	Material number <i>n</i> is removed from the materials display list.

Interactive Commands (cont.)

Graphics Diagnostic Commands

Command	Description
di dx	Highlight nodes with constrained x-axis translation.
di dy	Highlight nodes with constrained y-axis translation.
di dz	Highlight nodes with constrained z-axis translation.
di rx	Highlight nodes with constrained x-axis rotation.
di ry	Highlight nodes with constrained y-axis rotation.
di rz	Highlight nodes with constrained z-axis rotation.

Sliding Interface Display Commands

Command	Description
di si <i>n</i> m	Highlight the master nodes of sliding interface number <i>n</i> .
di si <i>n</i> s	Highlight the slave nodes of sliding interface number <i>n</i> .
di si <i>n</i> b	Highlight both the master and slave nodes of sliding interface number <i>n</i> .

Symmetry Plane, Stone Wall and Non-Reflecting Boundary Identification

Command	Description
di nrb	Highlight nodes on non-reflecting boundaries.
di sy <i>n</i>	Highlight nodes on symmetry plane number <i>n</i> .
di sw <i>n</i>	Highlight nodes on stone wall number <i>n</i> .

Interactive Commands (cont.)

Node and Element Identification

<u>Command</u>	<u>Description</u>
di npb	Display nodal print blocks.
di epb	Display element print blocks.
di nodes	Label visible nodes on the display where able.
di 3d	Label visible three dimensional brick elements on the display where able.
di 2d	Label visible two dimensional shell elements on the display where able.
di 1d	Label visible one dimensional beam elements on the display where able.

Identification of Nodes in Regions Subjected to Load Curves

<u>Command</u>	<u>Description</u>
di d <i>n</i>	Display the nodes subject to the forced displacement of load curve <i>n</i> .
di f <i>n</i>	Display nodes subject to the forces of load curve <i>n</i> .
di pr <i>n</i>	Display nodes subject to the pressures of load curve <i>n</i> .

Interactive Commands (cont.)

Identification of Regions Subject to Prescribed Boundary Conditions

Command	Description
di cv	Locate the nodes subject to a convection boundary condition.
di flux	Highlight nodes subject to a flux boundary condition.
di rb	Highlight nodes subject to a radiation boundary condition.
di re	Highlight the facets of a radiation enclosure.
di resn	Display the radiation enclosure surface numbers.
di tb	Highlight the nodes subject to a temperature boundary condition.
di ti	Highlight the nodes subject to an initial temperature condition.
di vb <i>n</i>	Highlight nodes subject to the velocity boundary conditions of load curve <i>n</i> .

Part Tolerancing

For convenience, the interactive part tolerancing commands have been reproduced from Section 4.2.10. All of these commands may also be included in the INGRID input file, or entered at the keyboard.

Part Tolerancing Commands

Command	Description
stol ϵ	Specify a tolerance of ϵ to be used in surface tolerancing. With this command, only nodes on the surfaces of parts are considered candidates for the nodal merging process. stol outputs the total count of merged nodes.
stp ϵ	Specify a tolerance of ϵ to be used in surface tolerancing. With this command, only nodes on the surfaces of parts are considered candidates for the nodal merging process. stp is identical to stol , but produces a summary of the merged nodes.
tol ϵ	Specify a tolerance of ϵ to be used in volumetric tolerancing. With this command, all nodes of parts are considered candidates for the nodal merging process.

Interactive Commands (cont.)

Part Tolerancing Commands (cont.)

Command	Description
tp ϵ	Specify a tolerance of ϵ to be used in volumetric tolerancing. With this command, all nodes of parts are considered candidates for the nodal merging process. tp is identical to tol , but outputs a summary of the merged nodes.
ztol ϵ	Specify a minimum absolute value for coordinates.

General Line Display

Command	Description
lv	Display all lines.
lvi n l_1 ... l_m	Display a list of n lines with line numbers l_1 to l_m .
ckl n	Display line number n .
lvs n m	Display a sequence of lines, n through m , inclusive.

Load Curve Display

Command	Description
lcv n	Display load curve number n .

Interactive Commands (cont.)Non-Graphical Diagnostics Commands

Command	Description
ajnp <i>x y z</i>	Prints node number of closest node to point (x,y,z) .
center	Establishes the centroid of the model as a reference for moments and products of inertia calculations.
centroid	Produces centroid information.
ckv	Check element volumes, and print a summary of the minimum and maximum volumes.
info	Provides general element information about the mesh.
mass	Generates a mass table containing the volume, mass and center of mass for all parts and materials in the model.
pinfo	Provides general element information about the parts in the active display list.
pmass	Generates mass information for the parts in the active display list.
reference <i>x y z</i>	Establishes point $(x,y z)$ as a reference for moments and products of inertia calculations.
size	Determines and prints the current physical X, Y, and Z extent of the model.
tmass	Generates mass information similar to pmass except for the entire model, irrespective of the active display list.

Output Commands.

Command	Description
cont	Causes output file to be generated and terminates execution of IN-GRID.
end	Graceful termination of INGRID. No output file is generated.
stop	Causes an immediate and ungraceful termination of INGRID.

Interactive Commands (cont.)

LLNL Octopus Specific Commands

Command	Description
dli	Generate a dli file on the CRAY. This option produces dli graphics files which can be manipulated with dltv. The dli files are named : d0ingaXX. Here, <i>a</i> is the family file number, and <i>XX</i> is a file number beginning with 00.
rjet <i>n size form</i>	Dispatch hardcopy output to RJET number <i>n</i> . The <i>size</i> of the output can either be set to narrow (default) or wide and the format, <i>form</i> , can be set to any of the following values: <i>form</i> = 1 : produces 5" plots. <i>form</i> = 2 : produces 8" plots. <i>form</i> = 3 : produces 10.5" plots. <i>form</i> = 4 : produces the largest possible plots for a specific RJET printer. <i>form</i> = 5 : produces 8.5" by 11" landscape plots. If the values supplied for <i>form</i> are positive, portrait plots are created. If the values supplied are negative, landscape plots are created (this has no effect on option 5).
plots	Create an FR80 plotfile containing a record of the displays produced on the TMDS monitor.
fr80 options	Send film output to the camera's. Option can be set to any of the values below to produce the corresponding types of output:
<i>fr80 - options:</i>	
fiche48	48x microfiche, comic format.
fiche48d	48x microfiche, document format.
fiche24	24x microfiche, comic format.
fiche24d	24x microfiche, document format.
color35	sprocketed 35mm color film.
dico35	sprocketed 35mm color film on DICOMED.
p16mm	sprocketed 16mm black-and-white film.
color16	sprocketed 16mm color film.
dico16	sprocketed 16mm color film on DICOMED.
cslide35	full frame 35mm slides on DICOMED.
hardcopy	high quality 11" film.
report	high quality 8.5" by 11" film.
vugraph	high quality 11" transparency.
vugraph11	high quality 8.5" by 11" transparency.
tv <i>n</i>	Send displays to TMDS monitor number <i>n</i> .
tv -r g b	Send displays to color TMDS. <i>r</i> displays red, monitor number <i>g</i> displays green, monitor number <i>b</i> displays blue. Note : in order to distinguish between normal TMDS and color TMDS with the TV command, the negative first argument is necessary.

INGRID USER GUIDE

ABSTRACT

The INGRID User Guide is the second part of the INGRID User Manual, and includes examples for the interactive commands, and model examples for DYNA3D, NIKE3D, TOPAZ3D and MONT3D. A part library is included as the last section of the User Guide, and is intended to serve as a resource for the generation of new or complex finite element models with INGRID. This part of the INGRID User Manual serves as a complement to the INGRID Reference Manual which details the commands used for describing geometry, boundary conditions, initial conditions and material specifications.

6.0 INTERACTIVE COMMANDS

This chapter of the manual presents the INGRID interactive commands used to control the graphical interrogation of mesh structure, boundary conditions, slide surfaces, radiation enclosures, etc. INGRID provides a rich collection of commands to allow the many details of a finite element model to be graphically interrogated and verified. Both graphical and non-graphical diagnostics are provided to facilitate the process of model verification. This chapter of the INGRID Manual also details the execution of INGRID on various computing platforms. Please refer to Chapter 5 for a complete list of interactive commands.

Input of all interactive commands is entirely format free, and usually occurs as keyword - parameter pairs. Commands are presented in a **bold** font, with the associated parameters in *italics*. All interactive commands are case insensitive, and keyword parameters may be input as either integer or floating point values without any adverse effects. INGRID parses interactive commands from the command line, and up to 80 characters may be input on a single command line. After the model description within the input file is processed, INGRID enters the interactive phase. Entry into the interactive phase is automatic unless the **noplots** command is encountered in the INGRID input file. INGRID may be terminated by using any of the following interactive commands: **stop**, **end**, **quit** or **cont**.

In order to simplify the discussion of how many of the commands will affect the display, a unit cube is used to illustrate many of the graphics commands. The cube, shown in Figure 6.1 is used throughout this chapter to emphasize INGRID's graphics capabilities.

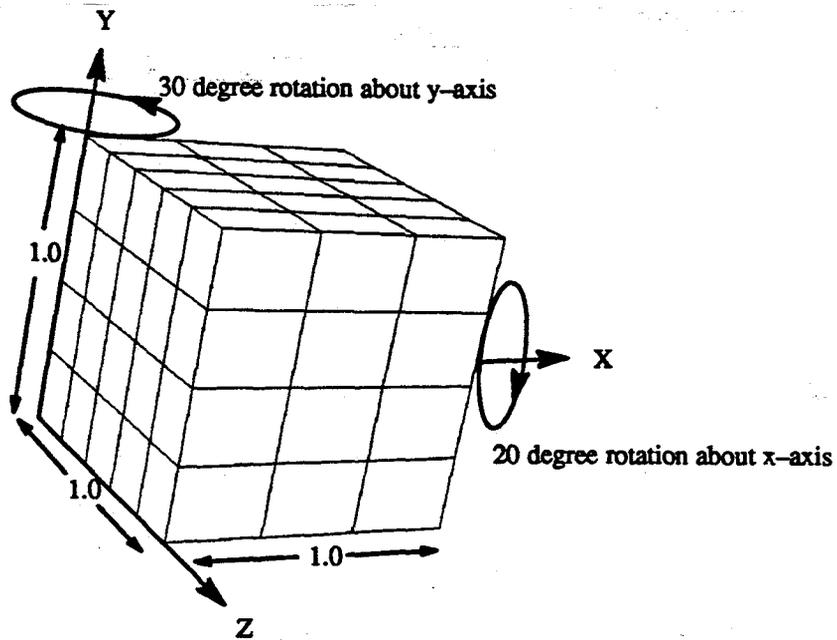


Figure 6.1
Simple Unit Cube Mesh

When possible, the cube is displayed using a 20° rotation about the x-axis and a 30° rotation about the y-axis to illustrate specific INGRID interactive commands. The unit cube is made up of 60 solid elements and occupies a physical space of $1.0 \times 1.0 \times 1.0$. It is discretized with three elements along the x-axis, four elements along the y-axis, and five elements along the z-axis. The INGRID input file used to create the simple cube mesh is presented in Figure 6.2

```
A 3x4x5 mesh on a unit cube
dn3d
start
1 4; 1 5; 1 6;      C Index Progression
0.0 1.0              C X-Coordinates
0.0 1.0              C Y-Coordinates
0.0 1.0              C Z-Coordinates
end
end
```

Figure 6.2
INGRID Input file for 3x4x5 Mesh

6.1 Executing INGRID

The UNIX (BSD, UNICOS, ULTRIX, etc.) version of INGRID may be executed from the shell (C, Bourne, Korn, etc.) and requires at most 3 command line arguments. The following command line arguments are acceptable:

```
ingrid [-i input] [-o output] [-s save]  
ingrid [i=input] [o=output] [s=save]
```

where *input* is the name of the file containing the INGRID input, *output* is the name of the file which INGRID will write to, and *save* is the name of the file for recording the command history. The command line keys may be upper or lower case, may be mixed and are not order dependent (e.g., “**ingrid -i input s=save O=output**” is acceptable). On all UNIX machines, /dev/null is a valid file specification for either the output or save files. Further, file names which include directory paths or UNIX environment variables may be used. For example:

```
ingrid -i ../Input/mesh.1 -o /user/people/bob/dynmsh -s /dev/null
```

Note that the VMS version of INGRID does support command line parsing of arguments, but INGRID must be defined in the user’s LOGIN.COM file as a foreign command. For example:

```
$ INGRID == “$disk$user:[username.path]ingrid.exe”
```

in the LOGIN.COM file makes INGRID a VMS foreign command.

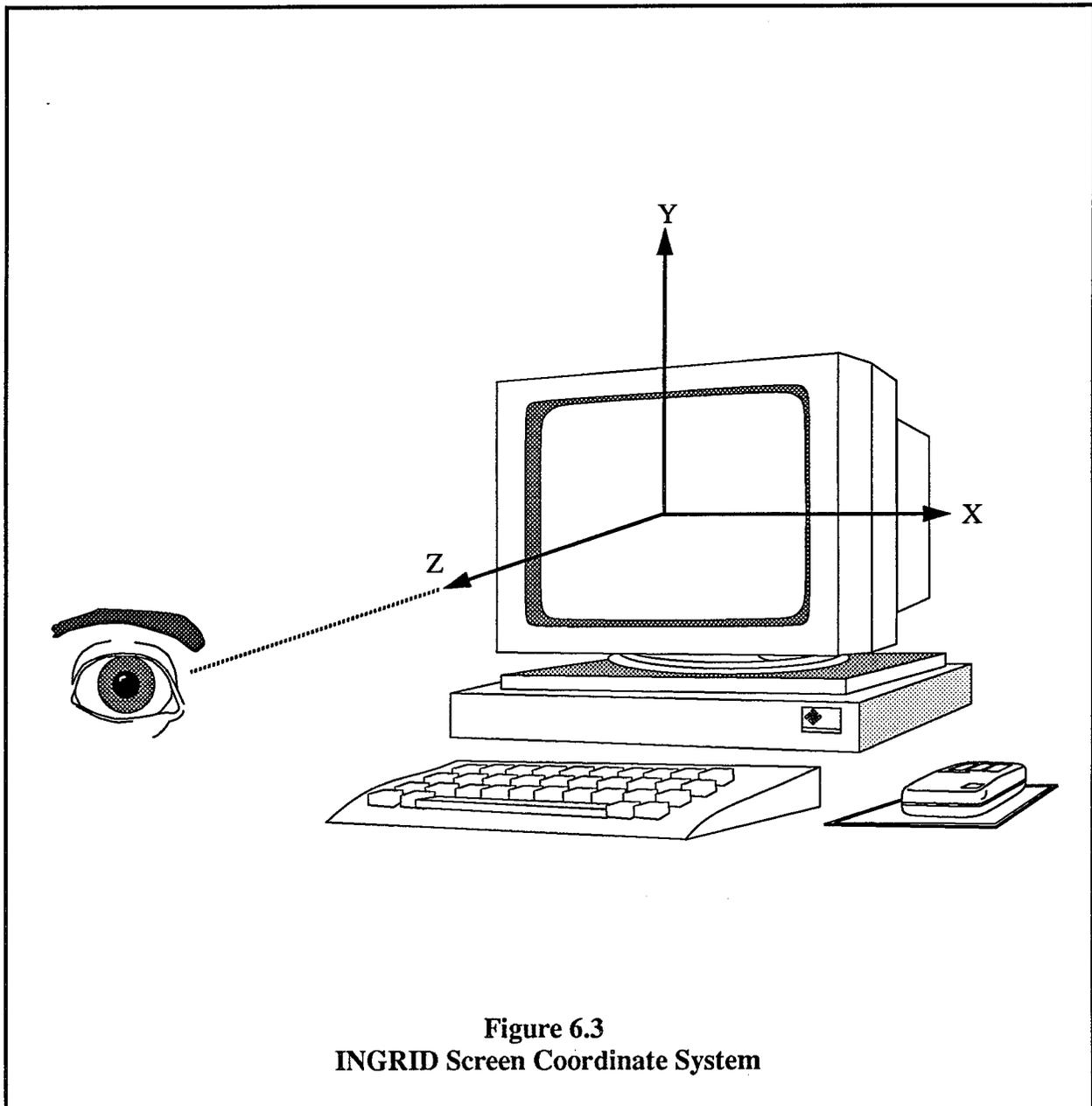
On any computing platform, if no command line arguments are used, INGRID will prompt for the input, output and save file names as shown below.

```
Optional : [i=input] [o=output] [s=save] [quit]  
           or  
           [-i input] [-o output] [-s save] [quit]  
>
```

At this point, **quit** may be entered instead of command line keys to exit INGRID.

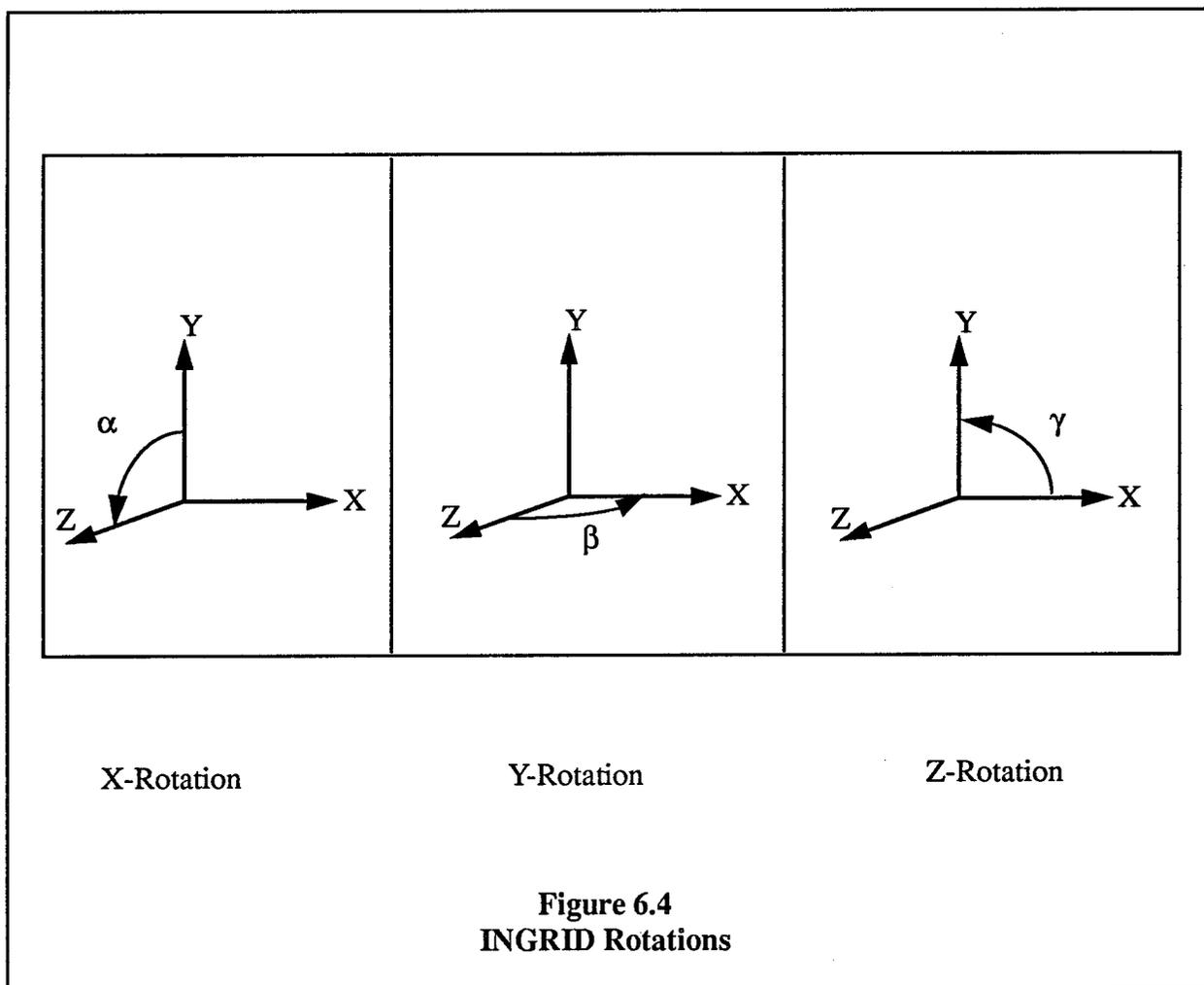
6.2 Coordinate Systems

There are two principal coordinate systems associated with INGRID's interactive graphics which the user must understand in order to effectively manipulate the INGRID display algorithms. The first coordinate system is a fixed, right-handed system, with its origin at the center of the display surface as shown in Figure 6.3. This is the so-called "screen" coordinate system. The x and y-axes of screen coordinates are placed in the plane of the screen with the positive x-axis directed to the right and the positive y-axis directed up. The positive z-axis is normal to the screen and directed out toward the viewer.



The second coordinate system for INGRID's graphics is based on the location of the center of rotation of the finite element mesh. Initially, this coordinate system has the same orientation as the screen coordinates, but translated a distance of ΔZ into the screen coordinate system where the model resides. Unlike the screen coordinate system, the world coordinate system is allowed to move about the screen, even off the screen, as the center of rotation moves. The location of the center of rotation for the world coordinate system is important when using the rotation commands.

Rotations are specified in degrees about a particular axis. The direction of a rotation is specified by either a positive or negative angle. Rotations are performed according to a right hand rule, whereby positive angles of rotation are counterclockwise and negative angles are clockwise when looking down the axis of rotation. The positive angles of rotation for INGRID's right handed coordinate system are illustrated in Figure 6.4.

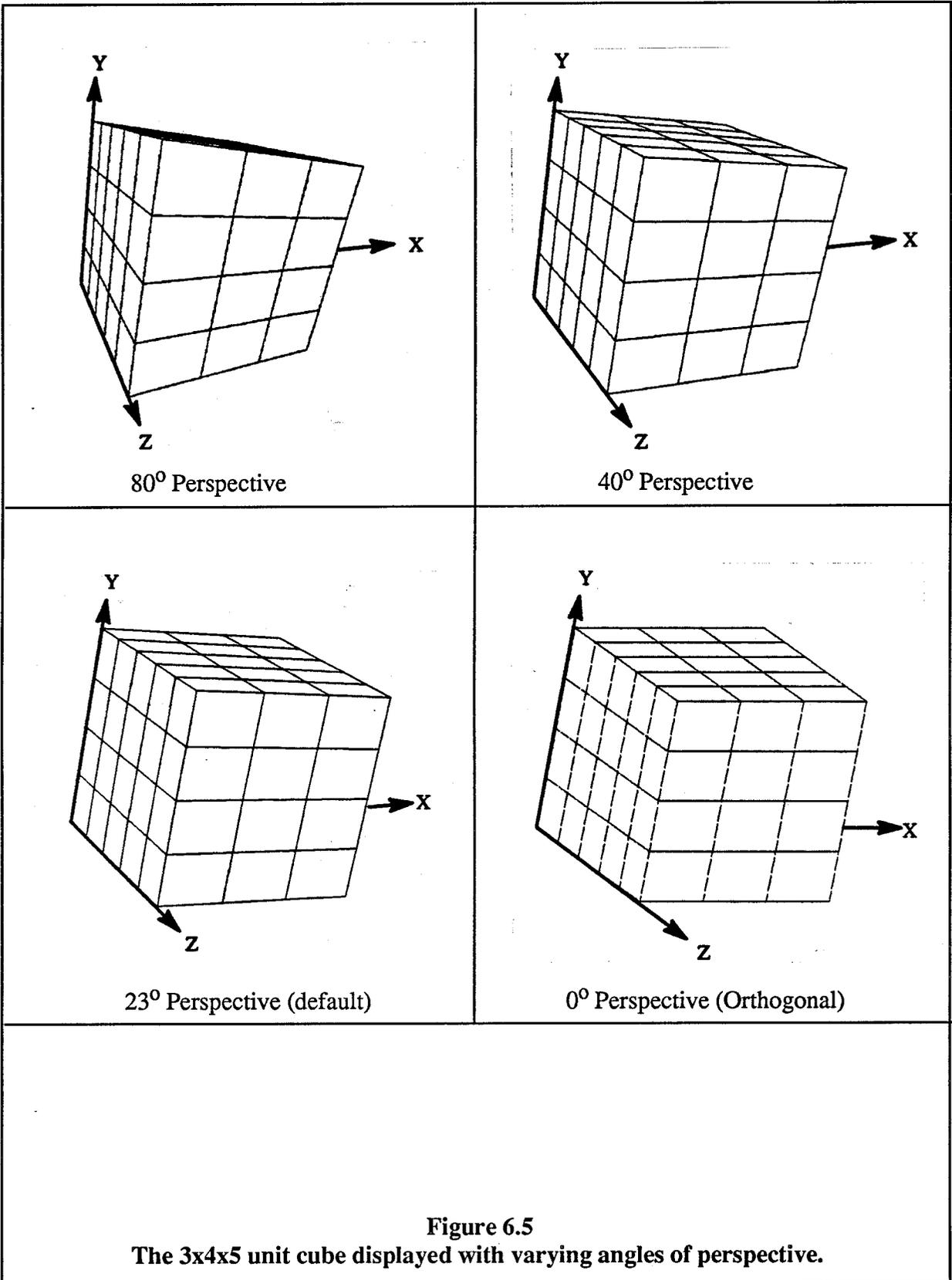


6.3 Perspective

Perspective has been included in the graphics display algorithms to enhance the visual appearance of depth. Objects closer to the viewer will appear larger while those farther away will appear smaller. Perspective is computed using a viewing volume (frustum). The viewer is placed at the apex of the viewing frustum (or the center of projection). The angle of the perspective is determined by the size of the projection plane and its distance from the viewer.

Unfortunately, while perspective improves overall image quality, too much perspective can cause the objects within the display to appear artificially distorted. Controlling the degree of the apparent distortion can be accomplished by controlling the amount of perspective applied to the perspective projection algorithm. There are two common methods typically employed. One method forces the viewer to specify a number of controls such as the positions of reference points, view planes, and center of projection. A second method which requires considerably less input is employed by INGRID. INGRID provides viewer perspective control through the **angle** command. With this command, the user specifies the angle of perspective in degrees. By default, the perspective angle has been set to 23° . An angle of perspective of 0° would imply that the center of projection has been placed at infinity, and as such, an orthogonal projection is generated. Figures 6.5a through 6.5d demonstrate the effect of progressively decreasing the angle of perspective from 80° to an 0° for an orthogonal view.

It is important to note that the apparent distortion caused by perspective will not be affected by the current zoom factor. Magnifying and shrinking the picture using **zf** and **zb** commands will not change the perspective. Scaling, however, will change the amount of apparent distortion caused by perspective. The reason for this is that points farther away from the center of the projection become increasingly influenced by the projection algorithm. Since scaling has the effect of moving points closer or farther from the center line, the amount of perspective and apparent distortion will change when a scale factor is applied.



6.4 Display Commands

Typically, hidden line algorithms are resource intensive procedures. To minimize the impact on interactivity, INGRID has several display commands - **display**, **view**, **poor**, and **draw** which use decreasing amounts of computing resources with an associated decrease in image quality. While each of the commands will produce a picture, the speed at which they execute and the utility of the final image varies significantly.

The highest quality picture is produced by the **display** command. It utilizes a very accurate, modified z-buffer algorithm to calculate edge visibility, surface, and edge intersections. Unfortunately, the price for the high quality rendering is speed (or a lack thereof). Even so, the time required to generate a display is not prohibitive so the command remains usable under most circumstances.

It is sometimes necessary to produce a high quality image. In order to calculate the intersection of edges accurately, the **display** command uses a number which specifies how close two edges can be before they are considered to be intersecting. That intersection resolution is controlled indirectly using the **reso** command. The argument to the **reso** command specifies, in multiples of 512, the z-buffer resolution in pixels. By default, INGRID establishes a resolution of 512x512 pixels. By increasing the resolution, more accurate edge intersections can be calculated and displayed. Unfortunately, increasing the screen resolution also increases the time required to produce the image. Note also that the **reso** command only influences edge intersections and does not imply a line width.

A good alternative to the **display** command which produces a usable picture with hidden lines removed is the **view** command. The **view** command, like **display**, utilizes a z-buffer to display the visible lines of the mesh. The main difference between the algorithm used by **view** is that edge and surface intersections are subjected to a somewhat coarser intersection criteria. The result is that **view** performs somewhat quicker than **display** with a corresponding reduction in image quality. In addition, the **view** command *does not display beam elements*.

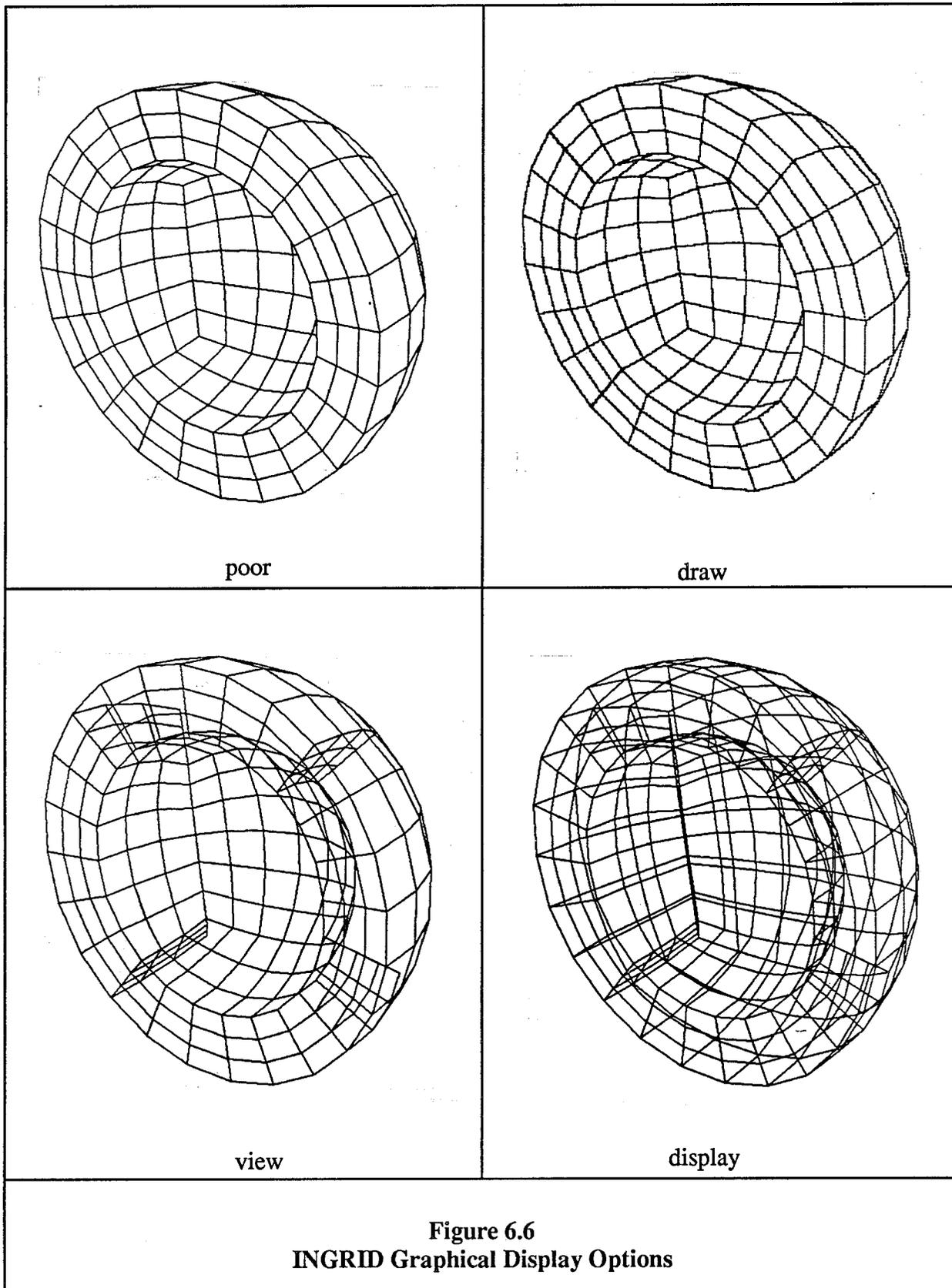
Another command available for generating displays is the **poor** command. It displays all of the exterior polygons in the model which face the viewer. No edge or surface intersections are calculated, instead, a normal for each polygon is calculated and used in a primitive backface culling operation on a part-by-part basis. The result is that a usable picture is generated quite rapidly. The problem with the **poor** technique is that some polygons which would not normally be visible,

including some internal to the model, are displayed. Even so, **poor** will render a useful display in a fraction of the time **display** or **view** would require. **poor** is especially useful when experimenting with rotations, translations, and scales to find the optimal view of a large mesh.

Of course, if hidden line removal is not desired, the **draw** command can be used. **draw** will display all of the polygons in the display list without testing to see if the edges are visible. Since no hidden line criteria are enforced, the rendering is the fastest available. One problem, though, is that if the mesh to be viewed is even moderately complicated, the image generated by **draw** may be confused by the presence of a large number of lines. When complicated meshes are viewed, typically the **poor** command is most useful for quick looks, and the **display** command is used to examine the details.

In order to compare the differences between all of the algorithms, a side-by-side comparison of the renderings can be seen in Figure 6.6. In this figure, a hollowed half-sphere has been oriented such that some of the differences between the rendering methods can be discerned. Note the increased granularity of the image generated with the **view** command when comparing it to the **display** image. Also note the internal polygons not included in the back-face culling operation which are subsequently included in the image generated with the **poor** display.

One last note concerning the display commands is that whenever a command is issued which directly affects the orientation, size, position of the picture on the screen, or the number of parts or materials displayed, INGRID does not automatically issues any command to generate an image. Automatic execution of one of the hidden line display options can be set with the **set tv** command sequence. By default, INGRID starts up as if the command '**set tv none**' had been issued.

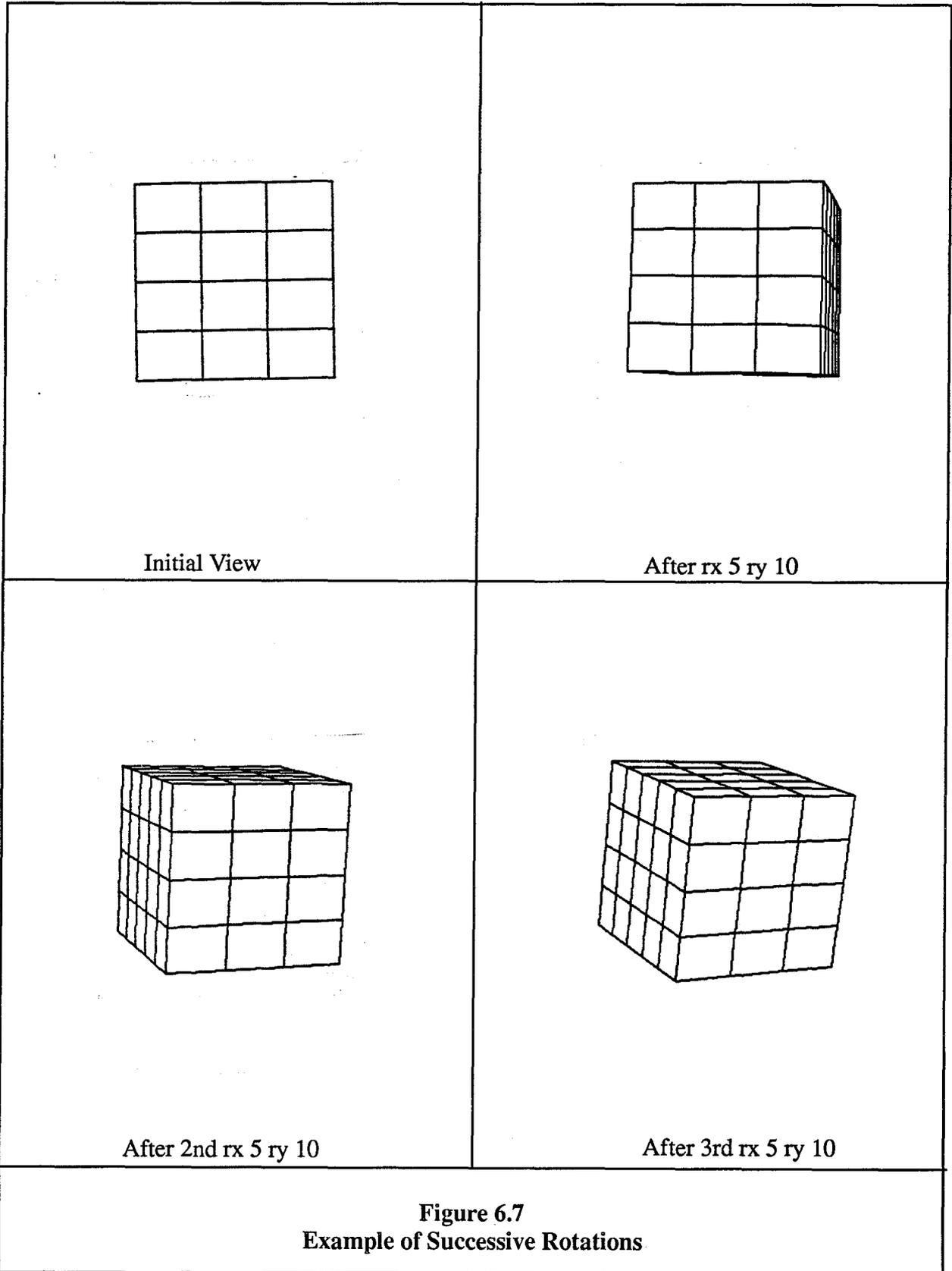


6.5 Rotation Commands

The orientation of the mesh displayed on the screen can be controlled using **rx**, **ry**, and **rz** commands, where the rotation angle is specified in degrees as an argument. Rotations may be either positive or negative angles which correspond to counterclockwise or clockwise rotations about the specified axis. All rotations are applied to the current orientation of the mesh which implies that the **rx**, **ry**, and **rz** commands have *relative* rather than absolute application.

A sequence of images have been produced to illustrate the effect that the rotation commands have upon the unit cube. Figure 6.7 shows how the cube appears initially without any rotations applied, and how the mesh would appear after successively executing commands to rotate about the x-axis by 5° and the y-axis by 10° . The final orientation in Figure 6.7 illustrates the unit cube after it has been rotated about the x-axis by 15° and about the y-axis by 30° .

Initially, INGRID displays the mesh without any rotation. This sets the initial view to be looking directly down the z-axis into the screen. The center of rotation is established using the bounding box for the current part display list. If this center of rotation is not acceptable, it is possible to alter it by issuing **fix**, **unfix**, **trans**, any of the panning commands: **u**, **d**, **r**, or **l**, or the **restore** command. Once specified, the center of rotation can be fixed and unfixed using the **fix** and **unfix** command. Once fixed, all of the rotational transformations are applied using the fixed center of rotation. The effect that **fix** has on the display can be quite subtle.



6.6 Panning and Zooming Commands

In addition to being able to control the orientation of the model, INGRID also provides magnification and translation commands for panning. The size of the displayed mesh is controlled using the zoom commands: **zf** and **zb**. Translation is controlled using the pan commands: **u**, **d**, **l**, and **r**, and alternatively, the **trans** command.

Panning up, down, left, and right is done using the **u**, **d**, **l**, and **r** commands. The distance to translate is expressed in terms of a normalized fraction of the window size. For example, the following commands:

```
>> u 0.5 r 0.25 poor
```

moves the mesh one half the screen width up and one quarter the screen width to the right. Whenever a translation command is executed, a new center of rotation is calculated. That calculation can be inhibited by fixing the center of rotation using the **fix** command. The new center is found by determining which portion of the picture is at the center of the window and then calculating the midpoint z-coordinate of the portion which is visible in the window.

An alternative way to translate the model about the screen is to use the **trans** command. The **trans** command has three arguments: an x, y, and z-coordinate in the physical dimensions. When **trans** is executed, these coordinates become the center of rotation. An example of the use of the **trans** command can be seen in Figure 6.8 where the unit cube is shown both before and after executing a **trans 1.0 1.0 1.0** command.

Altering the size of the displayed mesh can be accomplished using the zooming commands: **zf** and **zb**. Magnifying the picture is done by moving into the picture or zooming forward: **zf**. Reducing the size of the picture is accomplished by moving away from it, or zooming backward: **zb**. The amount of magnification or reduction is controlled by a single factor. For instance,

```
>> zf 3.2 poor
```

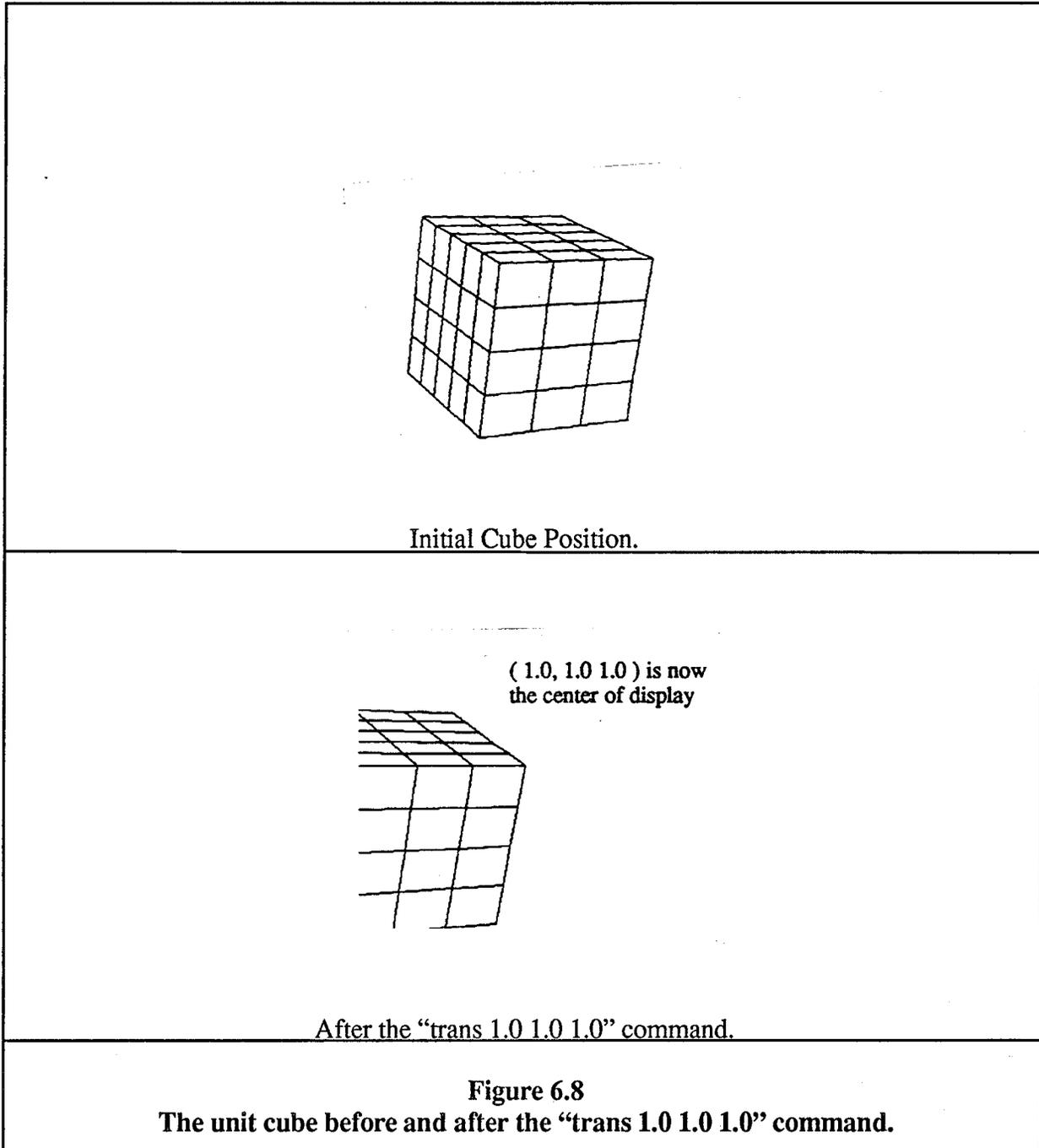
would cause the picture to be magnified by a factor of 3.2, while

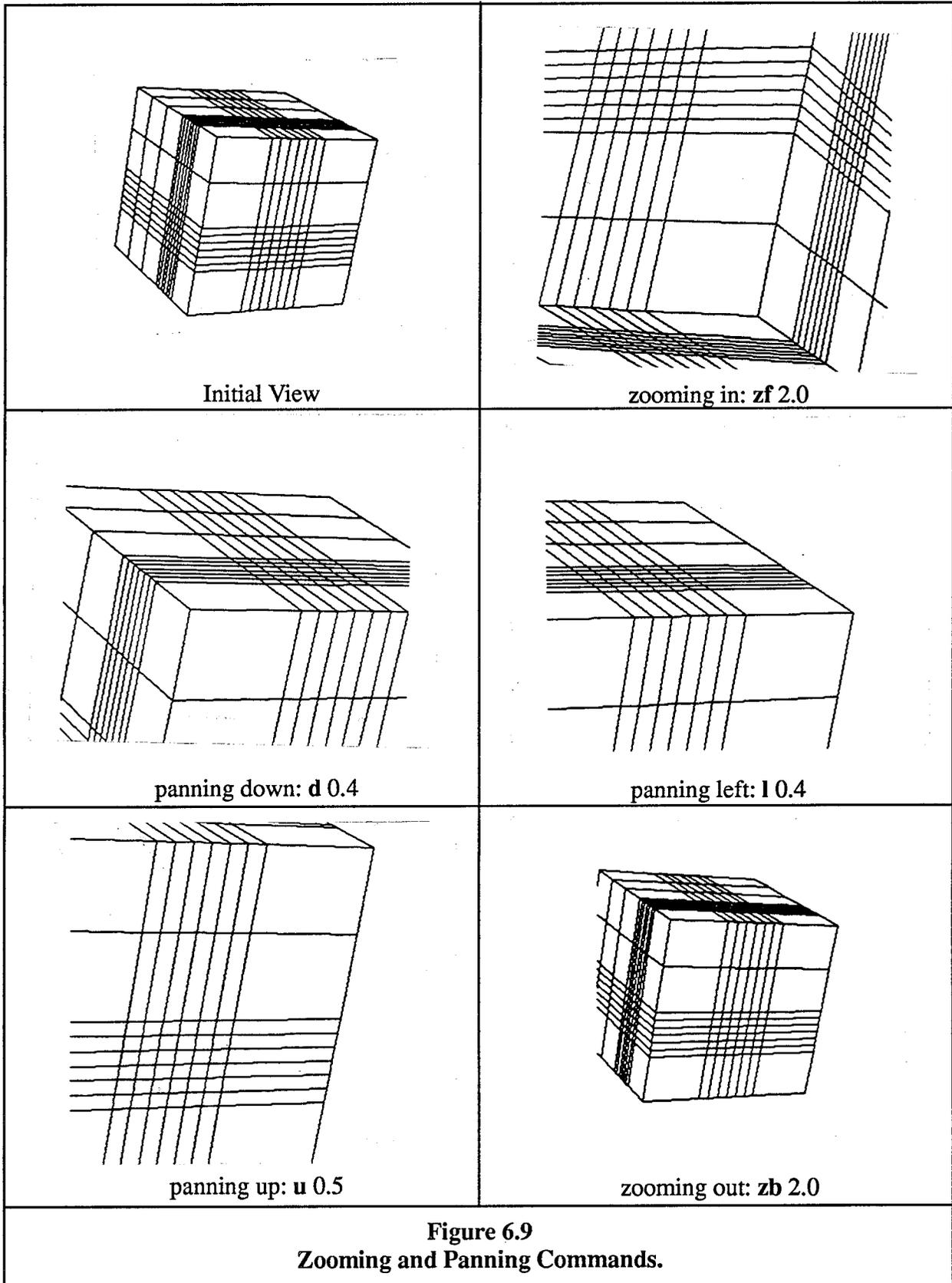
```
>> zb 2.0 poor
```

would reduce the picture by one half.

The magnification level is, by default, established at 1.0. The effect of applying multiple zoom factors is cumulative. The current magnification level can be determined by dividing the product of the **zf** factors by the product of the **zb** factors. Note: altering the zoom factor will have no effect on the perspective of the picture or the center of rotation.

In order to show how the panning and zooming commands can be used to examine a mesh, sections of the unit cube have been refined. In Figure 6.9 the mesh is shown after application of the **zf**, **d**, **l**, **u**, **zb** commands.





6.7 Scaling

Occasionally, a model will have one or more dominant axes. In such a case, the size of the model will be much larger along one or more axes. Figure 6.10 shows an I-beam before and after scaling in the x-direction. In the initial state, the horizontal length of the beam is much larger than both the width and height. When displayed, it is difficult to view the cross-section of the beam. While it is possible to zoom in and pan to view the end of the beam, it may require several panning commands to get positioned correctly. In addition, the rotational commands will be quite sensitive to adjustments along the x-axis. If, instead, the entire model was scaled along the x-axis using the `xscale` command, then it would be possible to scale, zoom, and view the entire model at once.

There are four scaling commands available in INGRID: `xscale`, `yscale`, `zscale`, and `scale`. Each command requires a single argument which is a factor to be applied to the corresponding x, y, or z-coordinates of the entire model. In the last scaling command, `scale`, the factor will be applied to all three spatial coordinates. In some senses, it may seem that scaling and zooming have identical effects. The principal difference between the scaling and zooming commands is that the scaling commands can be applied to a single axis while the zooming commands cannot. Furthermore, scaling commands will affect the distortion generated by the perspective projection algorithms, while the zooming commands do not affect the perspective.

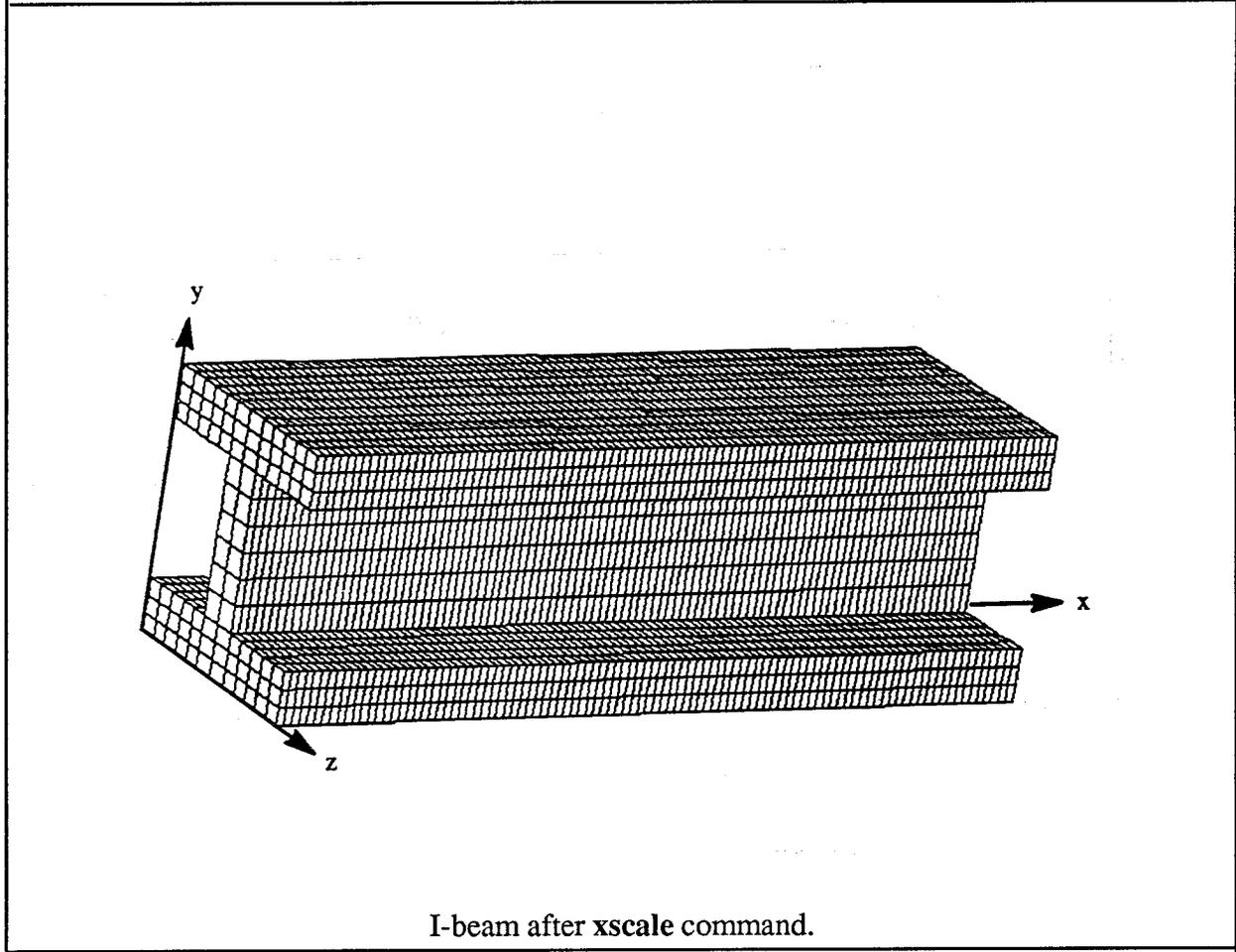
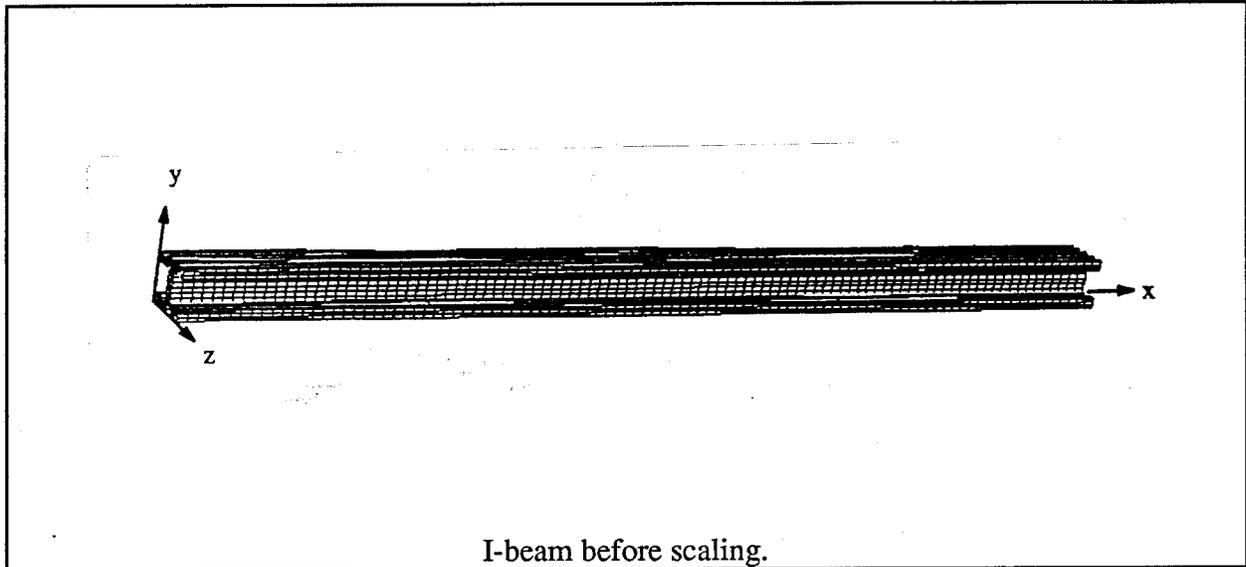


Figure 6.10
I-beam before and after scaling the x-dimension.

6.8 Displaying Part and Material Subsets

When dealing with large scale finite element analyses, the final model often becomes too complicated to permit the mesh to be interrogated in a meaningful way. Typically, INGRID meshes require more than one part using a variety of materials. In order to gracefully manipulate large models, INGRID can selectively display portions of the model through part and material subset commands.

Part numbers are assigned to the individual mesh components of the model in the order they are encountered in the INGRID input file. This list is maintained for conditional display options within INGRID. Whenever a display command is issued, INGRID checks the display list to see which parts should be included in the subsequent image. Only those part numbers which are in the display list are included in the z-buffer operations. By default, all of the parts of the model are included in the display list. The part display list can be modified by using any of the commands: **p**, **dps**, **ap**, **rp**, **dap**, or **parts**. The **p** command is used to view a single part. A subset of parts can be selected using the **dps** (display part subset) command. Single parts can be added or removed from the display list by issuing either an **ap** (add part) or **rp** (remove part) command, respectively. Of course, it is always possible to restore the original display list by entering a **dap** (display all parts) command. In addition, the **parts** command is available which can either restore the original parts list or specify a sublist, depending on the input given to it.

To illustrate how parts and material subsets can be displayed, the unit cube was dissected into four parts and unevenly discretized so that the parts could be easily distinguished when the model was assembled. The first two parts mate using a male/female interface. The last two parts are pins which go through the entire assembly. An exploded view of the four parts can be seen in Figure 6.11. The two pins are driven into two horizontal voids that have been removed from the first two parts.

When that model is displayed with all four parts placed in their proper position (Figure 6.12), it is difficult to see if the pins are actually penetrating the first two parts correctly. In order to check the installation of those pins, two part subsets were displayed. The first display was created by displaying parts 1, 3, and 4 (Figure 6.13). Figure 6.13 was created by executing a **dps 3 1 3 4** command and then plotting the mesh with **display**. Figure 6.14 was created by displaying parts 2, 3, and 4 using the **dps 3 2 3 4** command and then plotting the mesh with **display**.

A similar display list of part numbers is kept within a display list for materials. All or part of the materials in a model can be viewed by changing the entries in that list. A suite of commands, similar in function to the part commands, exist to modify the materials display list: **m**, **dam**, **dms**,

am, and **rm**. The **m** command, like the **p** command, will display only a single material. **dam** displays all of the materials in the list., and **dms** produces a subset. Individual material numbers can be added or deleted from the display list using the **am** and **rm** commands.

Whenever both part and material subsets are selected, the display will be that of the intersection of the two sets. For instance, suppose the unit cube in figure been made up of several different materials. If both a material and part subset was selected, only those parts within the part subset that contained the specific materials in the material subset would be shown. Furthermore, if a single part consisted of several different materials, only that portion of the part which contains the selected materials would be shown.

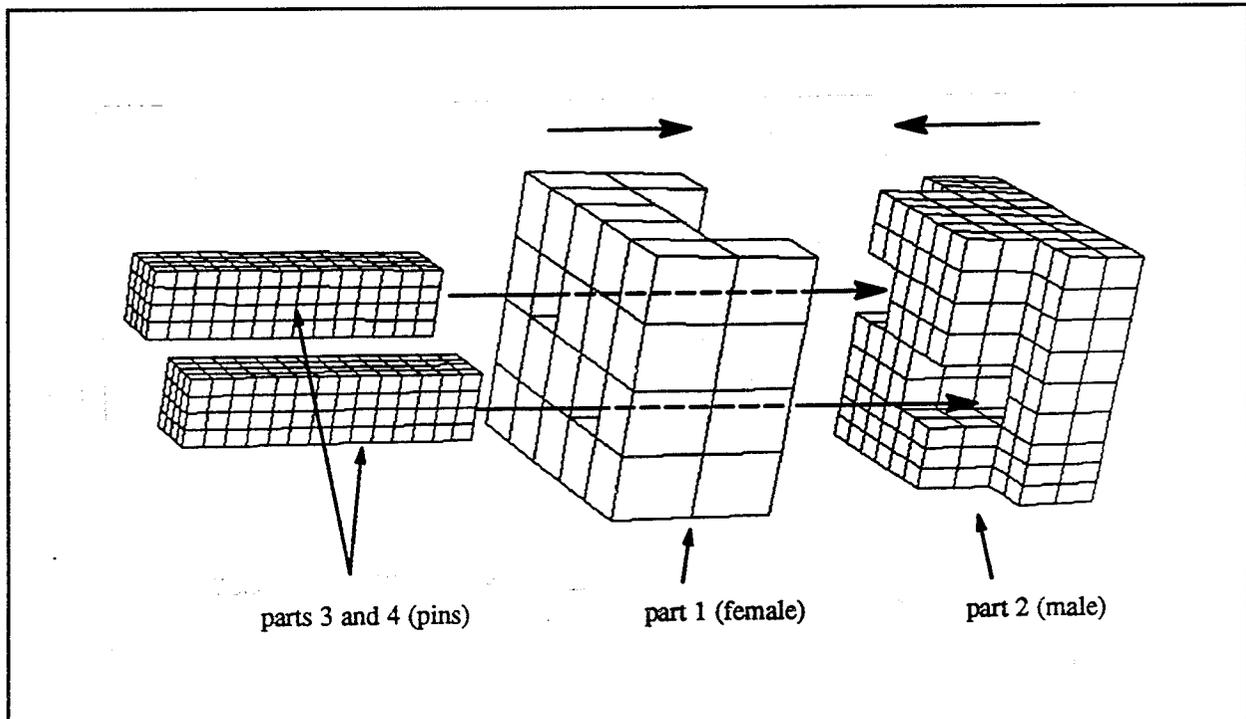


Figure 6.11
Exploded view of unit cube with male/female pins.

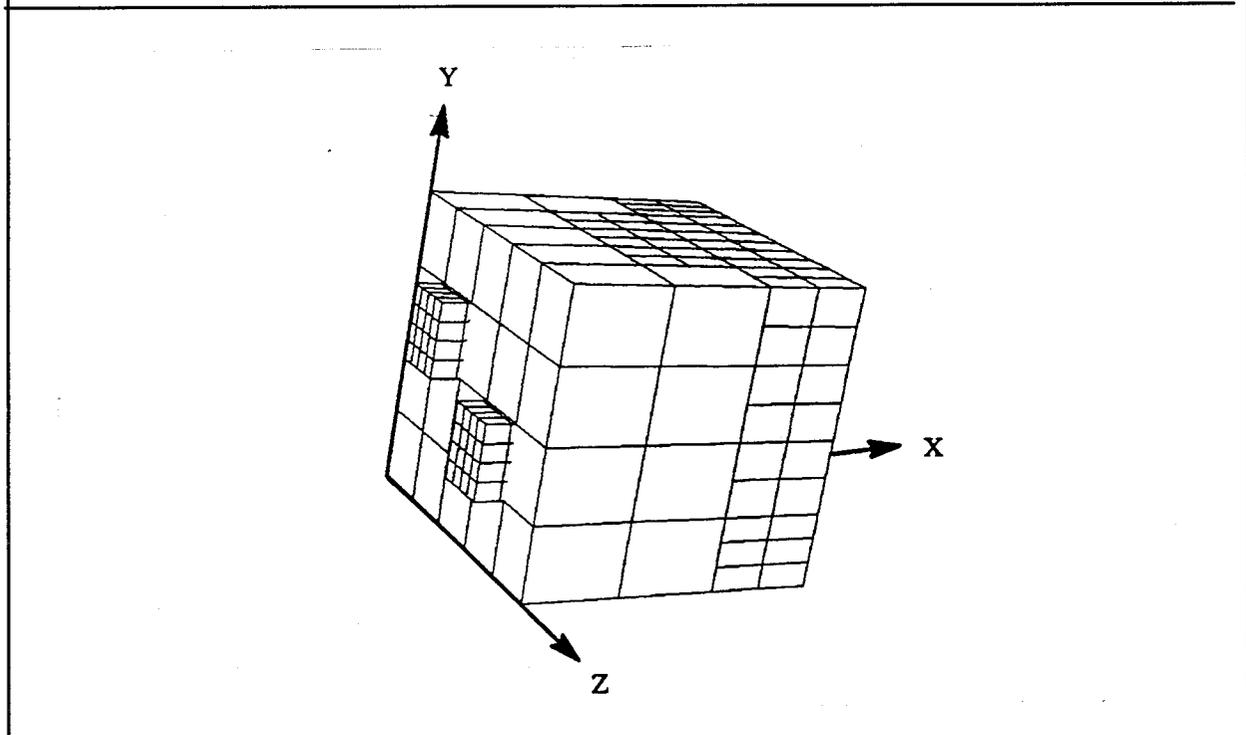
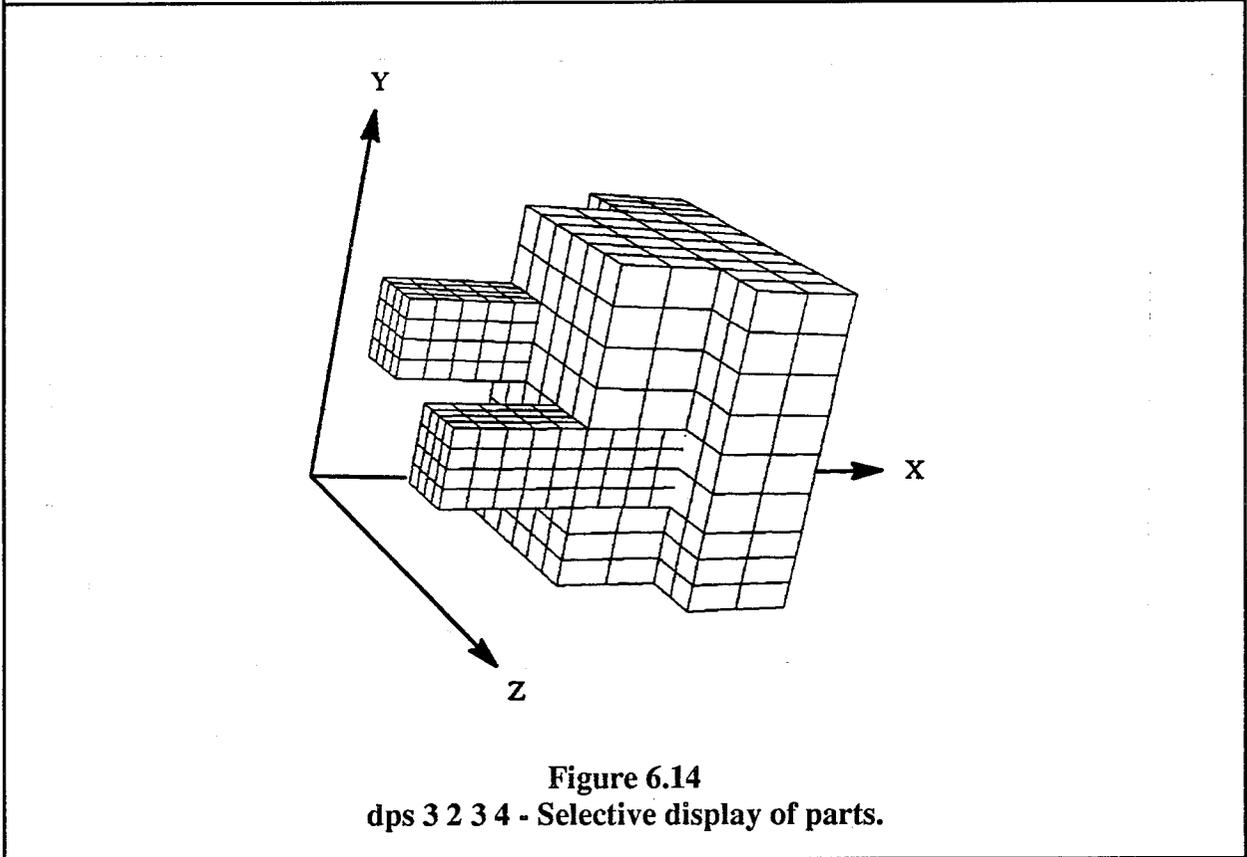
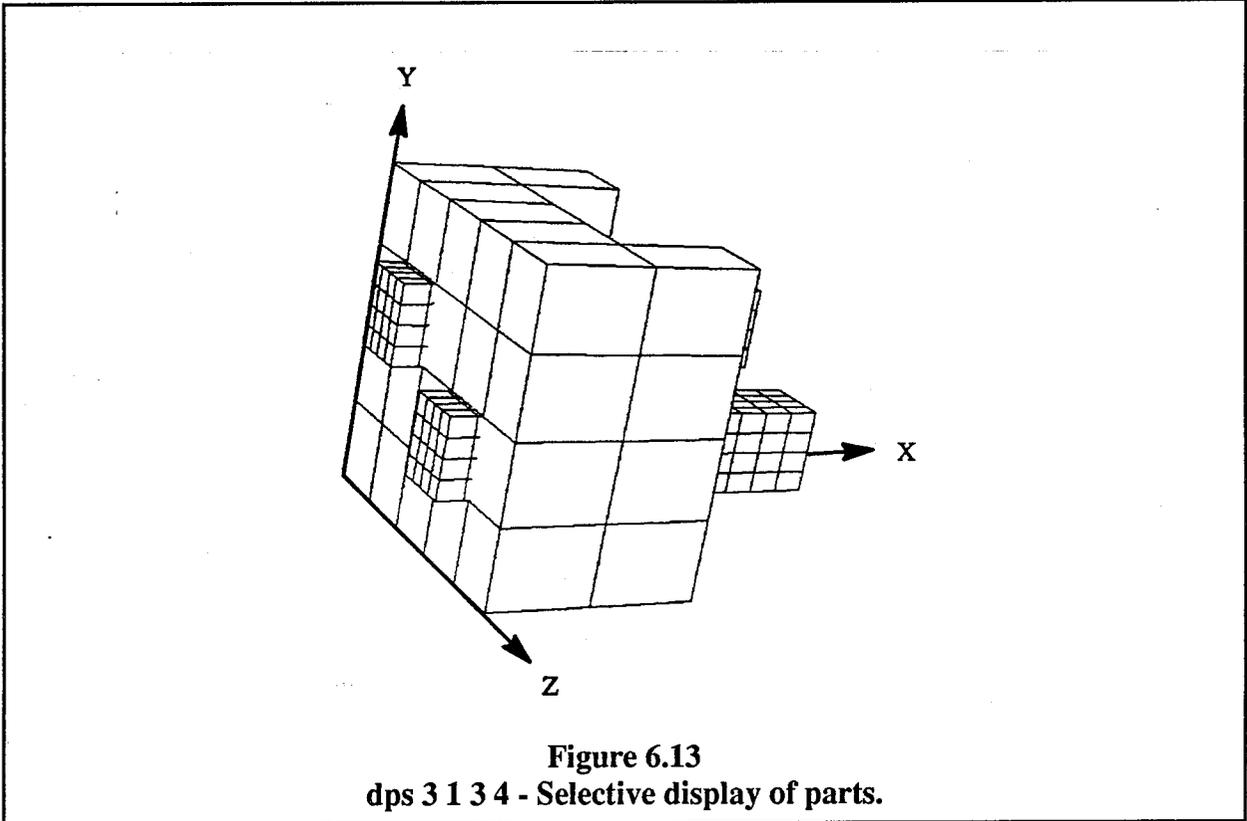


Figure 6.12
Mated parts for unit cube with male/female pins.



6.9 Restoring the Display

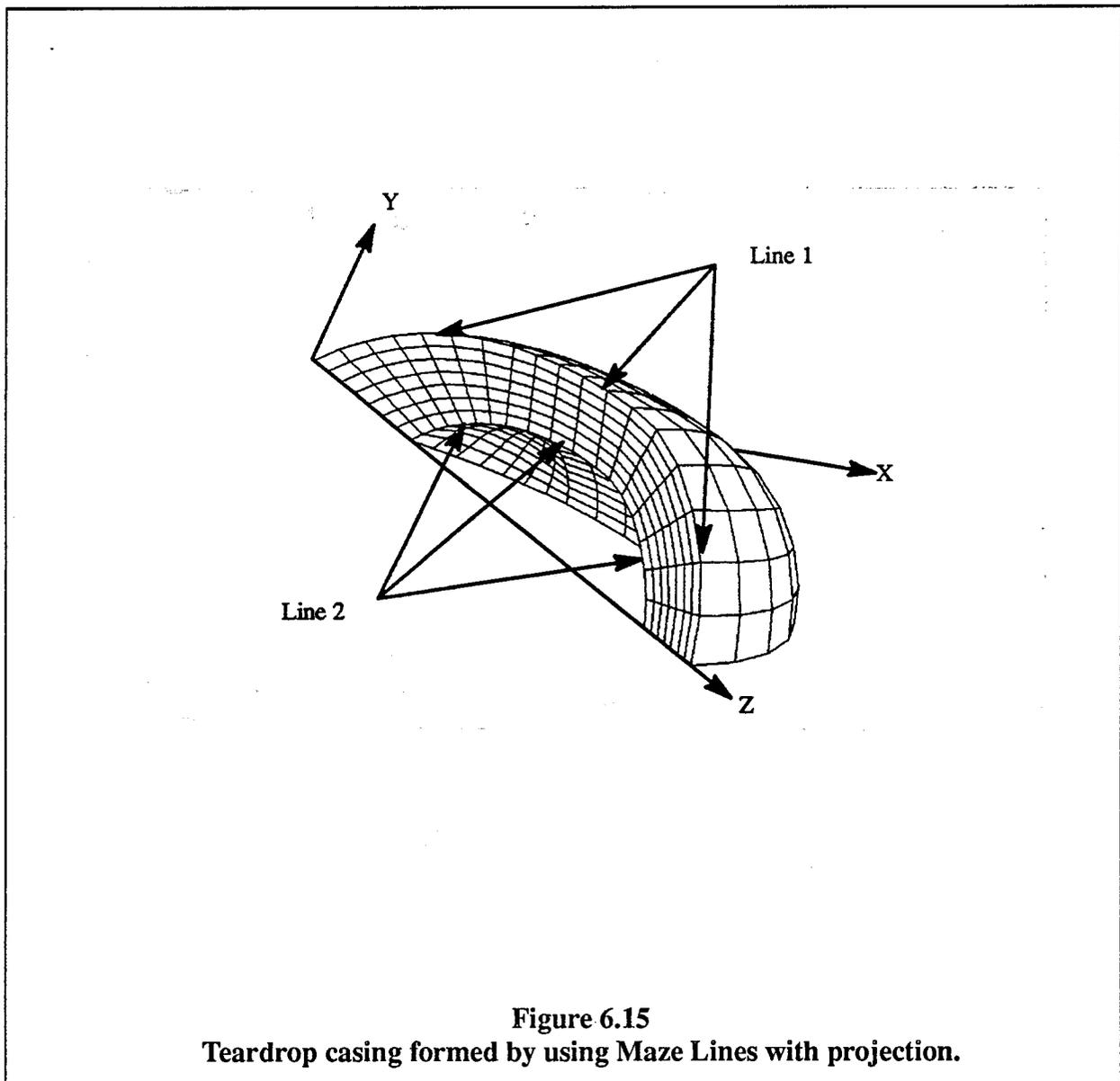
When INGRID begins an interactive session, all the rotations are set to zero and the initial view is looking down the z-axis into the screen. The size of the picture is determined by fitting the smallest prism containing all of the parts into a view which maximizes the use of the display surface. In a sense, INGRID sizes the display so that it fits entirely on the terminal screen. Sometimes, as an interactive session progresses, it is necessary to reset the display. This can be accomplished using the **restore** command.

When **restore** is executed, all of the rotations and zoom factors are reset to 0.0 and 1.0, respectively. While rotations and zoom factors are reset with **restore**, scaling factors are not reset. Executing a **restore** command will have no effect on the factors established using the **scale** commands. The net effect is that all of the parts in the current display list are presented in a display which has been fit to the screen using no rotation or zoom factors.

In most cases, resetting to the original orientation is desirable. There are some cases, however, when resetting to some other position is necessary. Those exceptions can be handled using the **fix** command. When the center of rotation has been fixed, the **restore** command will reset the model to the orientation established when the **fix** command was executed. The size of the normalizing prism will remain the same.

6.10 Viewing Line information

There are a number of reasons why line information is used in INGRID. One simple use of lines is to generate contours which are then 'spun' about an axis to create a surface. INGRID has a suite of commands: `ckl`, `lcv`, `lv`, `lvi`, and `lvs`, which can be used to verify specific line information. Figure 6.15 illustrates a teardrop shaped casing which was generated by specifying two contour lines in the y-z plane, rotating those lines about the z-axis, and then projecting solid elements to the surfaces which were produced (see Chapter 3 for more information on projections). The casing has been oriented so that the two lines which form the inner and outer contours of the casing are easy to see. If, however, it was not readily possible to examine the two lines which, when rotated about the z-axis, form the surface definitions, it would still be possible to verify them using the `lv` command. Figure 6.16 illustrates the resulting display when the `lv` command is executed.



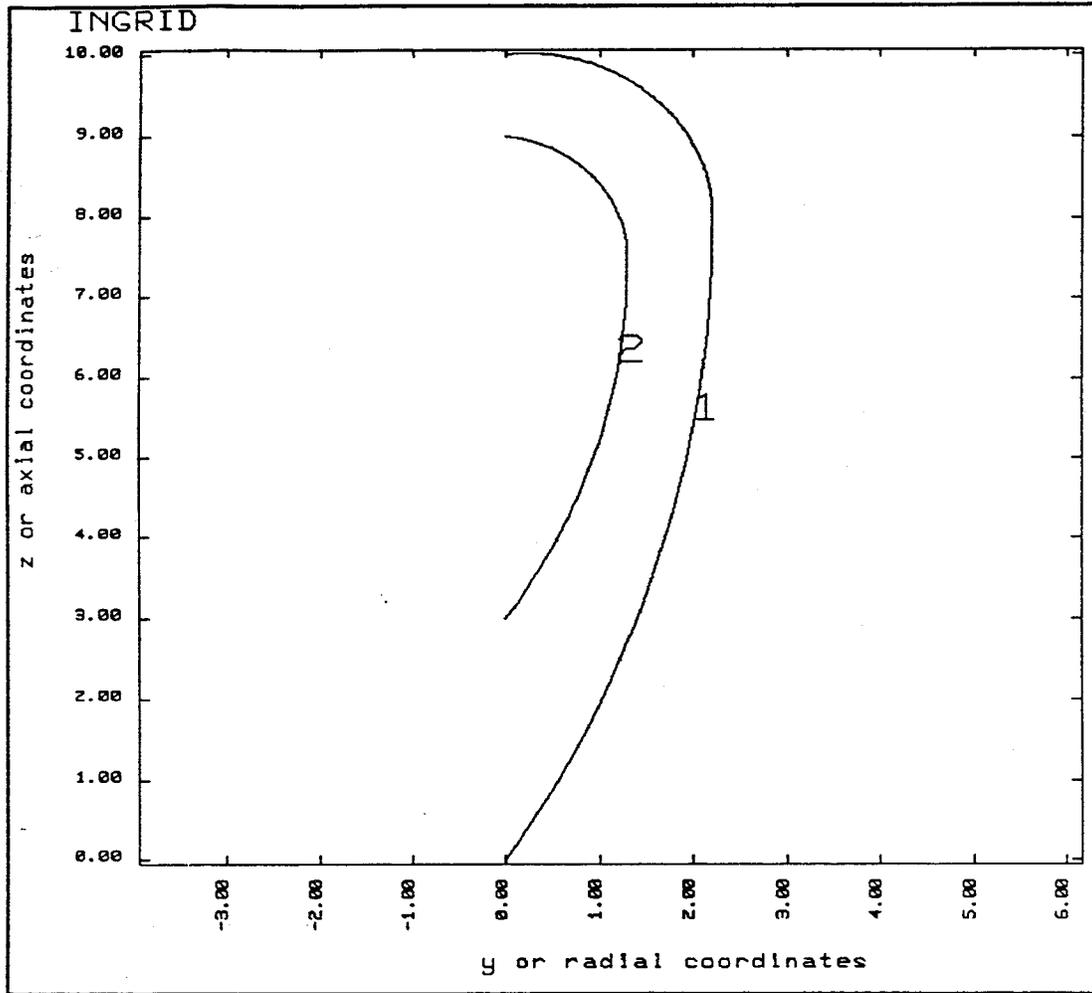


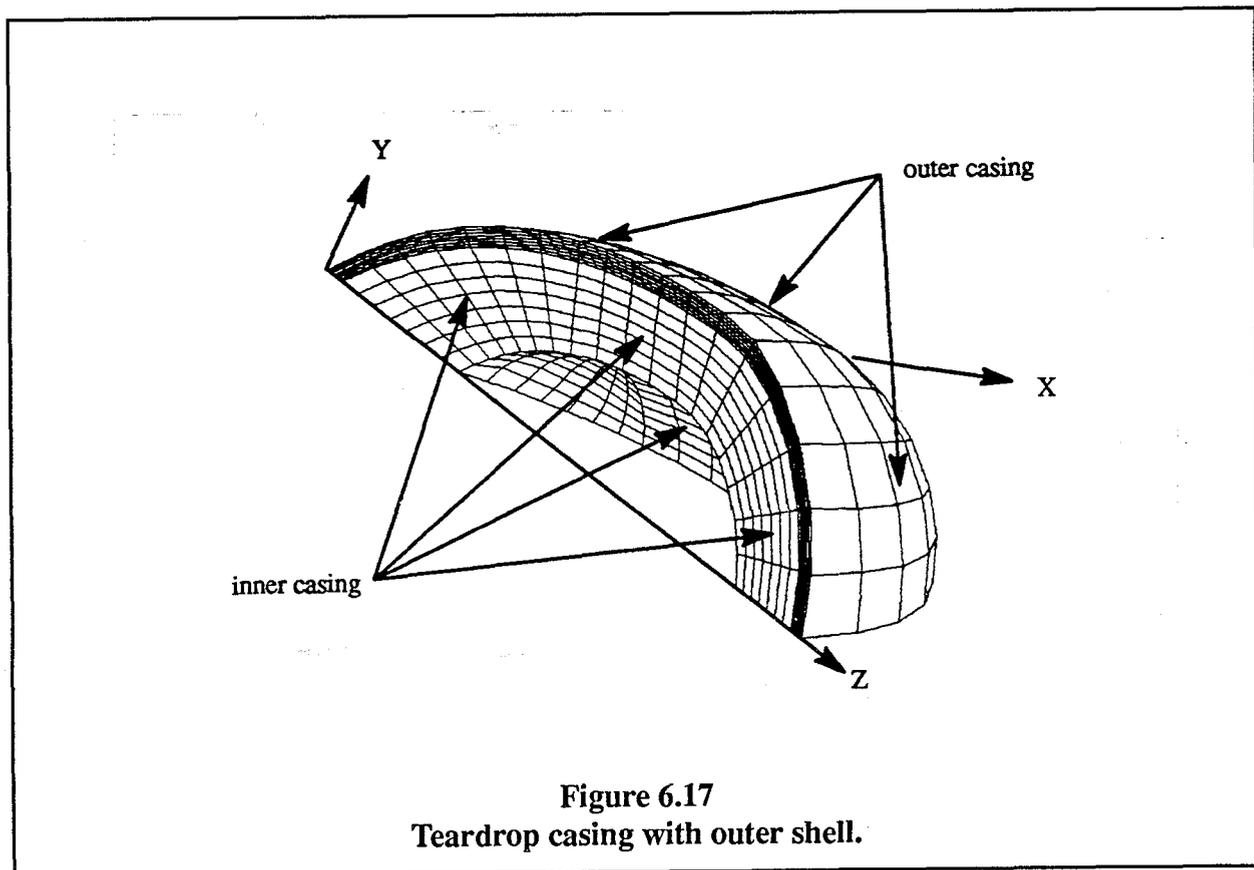
Figure 6.16
Line verification using the lv command.

6.11 Part Tolerancing

Tolerancing, in INGRID, is the process of identifying multiple nodes sharing the same spatial location in the mesh, and then merging the nodes into a single node. There are three types of tolerancing which occur in INGRID - internal, external, and global. Internal tolerancing occurs between nodes which are within the same part. External tolerancing occurs between nodes which occur on different parts. Global tolerancing is done irrespective of whether two nodes belong to the same part or not. The merging process is controlled through the commands **stol**, **stp**, **tol**, **tp**, **bptol**, and **ptol**.

Global tolerancing is accomplished using the **stol**, **stp**, **tol**, and **tp** commands. Each of the commands have a single argument which specifies the tolerance used to determine whether or not nodes are sufficiently close to be merged. The **stol** and **stp** commands are used to merge only those nodes which occur on surfaces of the reduced index space.

In order to demonstrate the tolerancing process, the casing in Figure 6.15 is to be merged with an additional shell on the outside of the case. This is illustrated in Figure 6.17. Each of the shared surfaces has exactly 173 nodes for the purposes of illustration. Each of the nodes on the outer



surface of the inner casing has a corresponding node on the inner surface of the outer casing. One set of these nodes must be removed, or merged, in order to create a valid finite element mesh. The most direct approach to merging the redundant nodes which share the same spatial coordinates is with the **stp** command for this particular mesh.

```
>> stp 0.0001
```

After issuing the **stp** command, INGRID will respond by merging the nodes and producing the following output:

```
245 nodes merged between parts 1 and 1
173 nodes merged between parts 1 and 2
245 nodes merged between parts 2 and 2
663 nodes were deleted by tolerancing
```

Had **stol** been used instead of **stp**, INGRID would have only printed the total number of merged nodes as follows:

```
>> stol 0.0001
```

```
663 nodes were deleted by tolerancing
```

Occasionally it is necessary that all of the nodes within a specified tolerance are merged, not just the surface nodes. Two commands are available for global tolerancing of such nodes: **tol** and **tp**. The **tol** command, like **stol**, will pass through the nodal database and remove those nodes which are within a specified tolerance of each other. The difference between **tol** and **stol** is that **tol** will apply the tolerance to all of the nodes rather than just surface nodes. The **tp** command, like **stp**, in addition to providing the same type of tolerancing as **tol**, will also print some explanatory diagnostic messages.

While global tolerancing will usually merge redundant nodes, having the same tolerance applied to the entire model is not always desirable. Examine again the teardrop case in Figure 6.17. The nodes in the outer shell are much more densely spaced than those in the inner core. A global tolerance which might be appropriate for the inner shell might not work so well for the outer shell, or vice-versa. Two commands, **bptol** and **ptol**, are available to resolve this problem. They can be used to augment the global tolerances when they are applied. **bptol** controls the external merging between two parts, while **ptol** controls the internal merging of nodes within a specific part. In the case of the teardrop case, using **bptol** and **ptol** produces the following:

```
>> bptol 1 2 0.00001
>> ptol 2 0.00001
>> tp 0.001
```

The printed diagnostics caused by using the **tp** command are:

```
245 nodes merged between parts 1 and 1
173 nodes merged between parts 1 and 2
245 nodes merged between parts 2 and 2
663 nodes were deleted by tolerancing
```

The tolerancing process is actually quite straightforward. Its effects, however, can become very complicated very quickly whenever the model becomes even modestly complicated. Internally, INGRID keeps a database of nodal positions. Whenever two nodes are within a specified distance, or tolerance, one is removed and the other is used for both cases. Whenever the commands **bptol** or **ptol** are used, an entry is placed in a list of tolerancing exceptions. Whenever two nodes satisfy a particular exception, the exception tolerance is used instead of the global tolerance.

It is important to realize that the actual tolerancing occurs when one of the four commands **stol**, **tol**, **stp**, **tol**, or **tp** are used. The **bptol** and **ptol** commands only modify the entries in an exception list which is traversed whenever the tolerancing commands (**tol**, **stp**, **tol**, or **tp**) are executed. It is just as important to realize that there is no way to control the order in which nodes are merged. If three nodes occur near each other in one part and three nodes occur close together in an adjacent part, there is no hierarchy which forces all the nodes within a particular part to be merged before the common nodes between two adjacent parts are merged. That notion influences the numbers which are printed as diagnostics when the **stp** or **tp** commands are processed.

One last tolerancing command is useful to dampen the numerical errors which frequently occur during the projection process. When nodal positions are altered, particularly by cylindrical or spherical coordinate transformations, discretization errors introduce a small level of noise into the model. Those errors become most apparent when browsing through the output file. Numbers within machine epsilon are printed as very small numbers rather than simply zero. Because of that, an additional command is available to make the output file more readable: **ztol**. By using **ztol**, it is possible to round values which are near zero, to be exactly zero. Examine the output excerpt below:

```
*----- NODE DEFINITIONS -----*
*
1  0. 0.000000000000e+00 0.000000000000e+00 1.000000000000e+00 7.
2  0. 0.000000000000e+00 0.000000000000e+00 1.4285714600000e+00 7.
3  0. 0.000000000000e+00 0.514000000000e-14 1.8571428100000e+00 7.
4  0. 0.429400000000e-14 0.000000000000e+00 2.2857141500000e+00 7.
5  0. 0.000000000000e+00 0.000000000000e+00 2.7142856100000e+00 7.
6  0. 0.000000000000e+00 0.000000000000e+00 3.1428573100000e+00 7.
7  0. 0.000000000000e+00 0.392100000000e-14 3.5714285400000e+00 7.
8  0. 0.612701000000e-14 0.642013000000e-14 4.000000000000e+00 7.
9  0. 0.367200000000e-15 3.9510405100000e-01 1.6861050100000e+00 7.
10 0. 0.143110000000e-14 3.6566308100000e-01 2.0602057000000e+00 7.
```

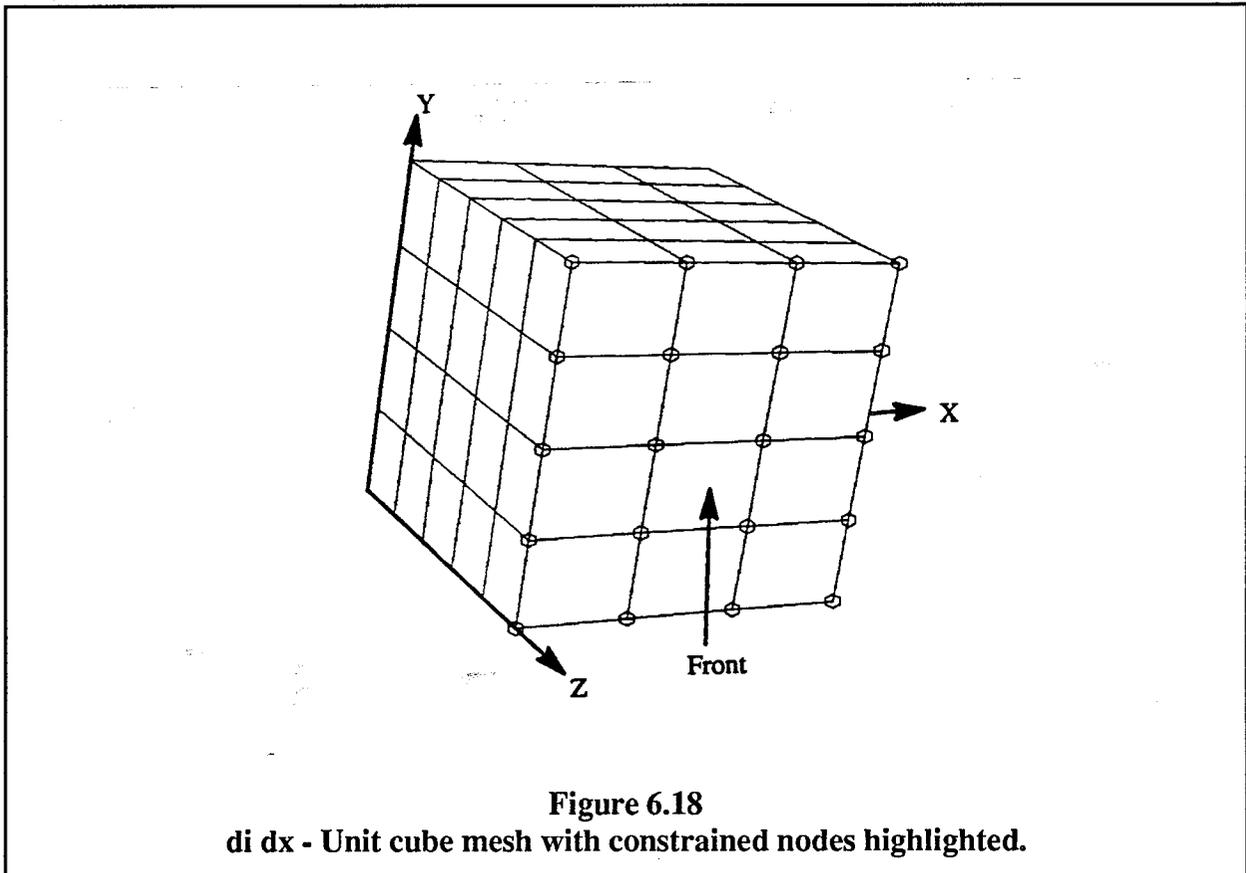
There are a number of entries with magnitudes on the order of 1.0e-15 where they should be 0.0 instead. `ztol 1.0e-12` would clean up the output to look like:

```
*----- NODE DEFINITIONS -----*
*
1  0. 0.000000000000e+00 0.000000000000e+00 1.000000000000e+00 7.
2  0. 0.000000000000e+00 0.000000000000e+00 1.4285714600000e+00 7.
3  0. 0.000000000000e+00 0.000000000000e+00 1.8571428100000e+00 7.
4  0. 0.000000000000e+00 0.000000000000e+00 2.2857141500000e+00 7.
5  0. 0.000000000000e+00 0.000000000000e+00 2.7142856100000e+00 7.
6  0. 0.000000000000e+00 0.000000000000e+00 3.1428573100000e+00 7.
7  0. 0.000000000000e+00 0.000000000000e+00 3.5714285400000e+00 7.
8  0. 0.000000000000e+00 0.000000000000e+00 4.000000000000e+00 7.
9  0. 0.000000000000e+00 3.9510405100000e-01 1.6861050100000e+00 7.
10 0. 0.000000000000e+00 3.6566308100000e-01 2.0602057000000e+00 7.
```

6.12 Graphical Diagnostic Information Display - the di commands

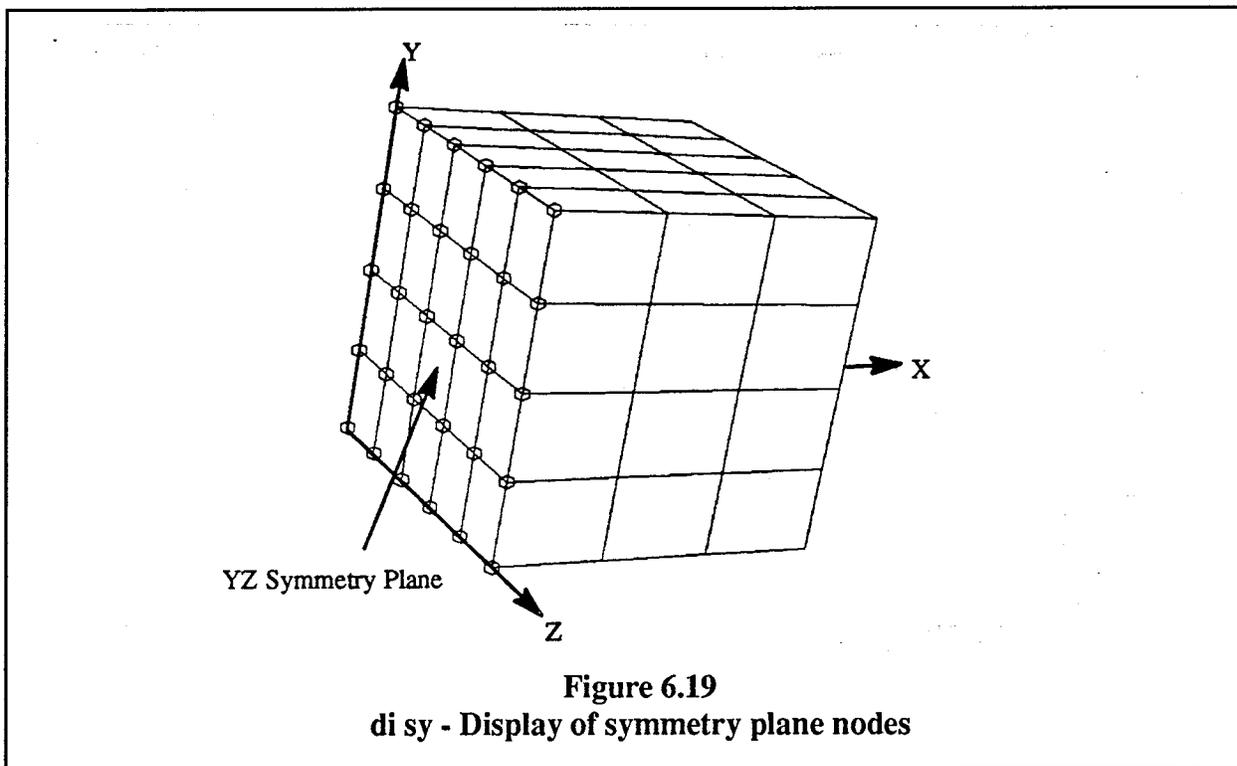
Establishing the initial geometry is only a part of the overall process of creating a finite element model with INGRID. Boundary conditions such as constrained motion, sliding interfaces, load curves, symmetry planes, and non-reflecting boundaries all contribute to a complete model. Verifying the correctness of such model properties is just as important as establishing the geometry. In INGRID, model validation is achieved using the diagnostic display commands in conjunction with the **display** command. Specific diagnostics are enabled using **di** and disabled using **dioff**. Once enabled, objects in the mesh which meet the specific condition are highlighted the next time a **display** command is issued. Specific conditions are enabled by the particular input supplied as arguments to the **di** command.

If it was necessary to constrain the motion of the nodes on the front face of the 3x4x5 unit cube such that they were allowed to expand or contract only in the z-direction, boundary conditions which inhibited translation in the x or y-direction would be required. Entering either a **di dx** or a **di dy** with the **display** command would highlight those nodes which are constrained. Highlighting is done by circling the affected nodes (see Figure 6.18) with small hexagonal outlines.

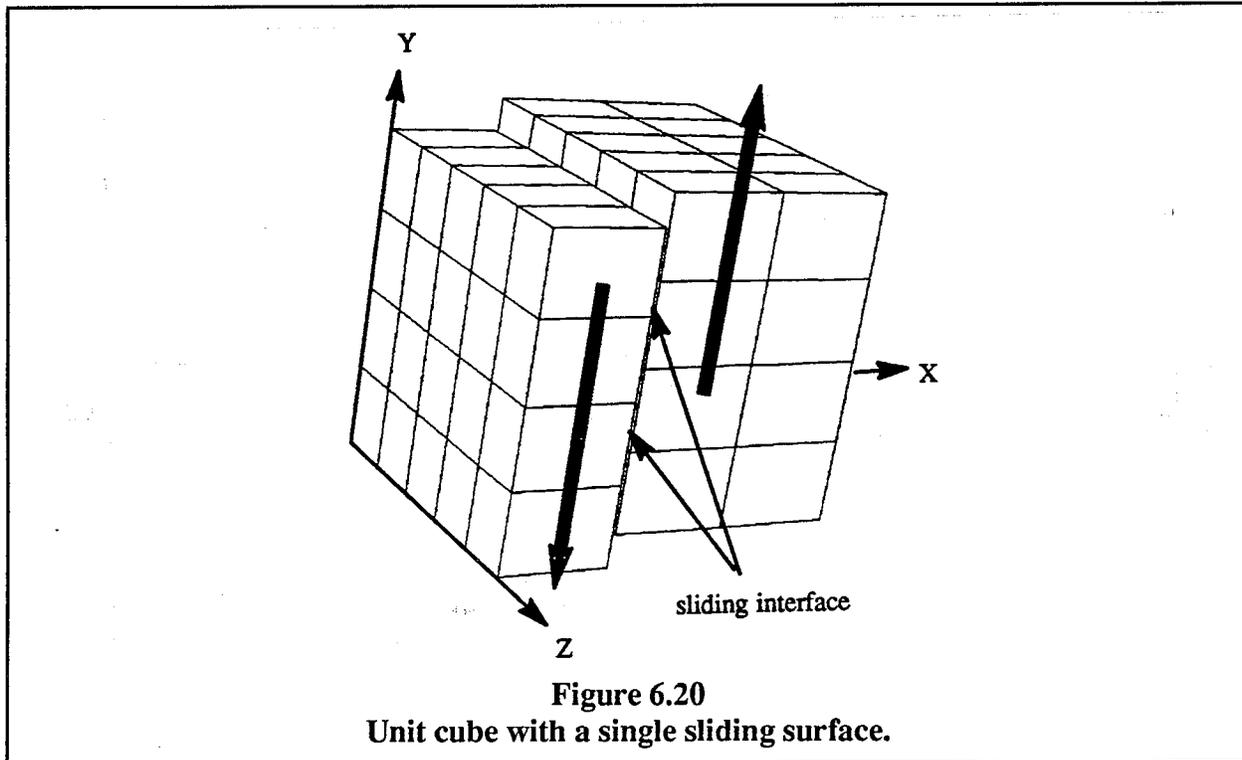


Each node in a DYNA3D or NIKE3D model has motion described with up to six degrees of freedom - translation in the x, y, and z direction and rotation about the z, y, and z-axis. The motion in any one or all three coordinate directions can be constrained in INGRID. Verifying that the correct nodes are constrained can be achieved using the appropriate **di** command. **di dx**, **di dy**, and **di dz** will encircle the respective nodes with constrained translation in x, y, and z. **di rx**, **di ry**, and **di rz** will encircle nodes the respective nodes with constrained rotation in x, y, and z. Note that rotational constraints have no effect on solid elements in both DYNA3D and NIKE3D.

Nodes on a symmetry plane may be examined in much the same way regions of nodes can be inspected for constraints. This similarity is due to the fact that a symmetry plane constrains nodal motion. In Figure 6.19, a symmetry plane was established in the YZ plane.



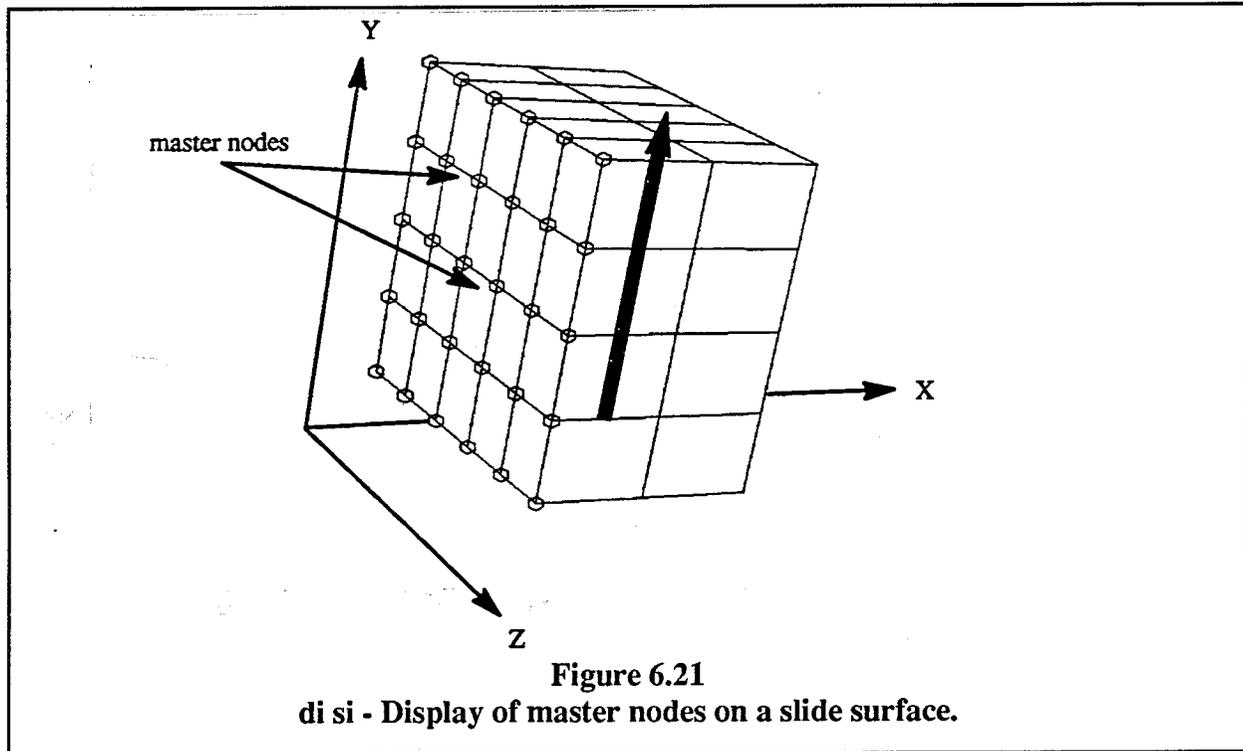
Entering the command **di sy 1** (display symmetry plane 1) resulted in highlighting the nodes associated with plane #1. The **di** commands are also available to highlight nodes along sliding surfaces. To illustrate this feature, the unit cube was cut along one of the x axis partitions and a slide surface was defined between the two parts. A shearing deformation, such as that depicted in Figure 6.20, could then be accommodated with this new model.



It is often necessary to verify the master/slave relationship between the nodes on a sliding interface is correct. For this reason, the **di si** has two arguments - the number of the sliding interface and an indication of which set of nodes are to be displayed: master, slave, or both. The **di si 1 m** command was used to generate Figure 6.21 which shows the master side of slide surface #1.

If the **poor** command is used with **di si**, then nodes on the selected slide surface are encircled. If the **disp** command is used with **di si**, then the slide surface normal vector is displayed at the center of each slide surface segment (or element face). This normal vector may be shown as a projecting arrow which directly indicates the outward normal direction. Alternatively, the normal direction may be indicated by a circular arc with an arrow indicating the direction according to the right-hand-rule. The proper orientation of slide surface normals is discussed at length in the DYNA3D and NIKE3D.

It is sometimes important to be able to locate specific nodes using node numbers. Incorporated into the set of diagnostics are **di** commands which display node numbers and element numbers for one-dimensional beam elements, two-dimensional shell elements, and three-dimensional solid elements. The **di** commands to display those items are, **di nodes**, **di 1d**, **di 2d**, and **di 3d**, respectively.



The unit cube has been modified again in order to show specific INGRID capabilities for the display of node and element numbers. Of particular interest are the three following modifications. First, an additional narrow partition in the y-z plane has been created for reasons described below. Next, a section of solid elements have been removed from the front of the cube and replaced by a layer of shell elements connected to the first partition by two beam elements. The resulting cube can be seen in Figure 6.22 with node numbers displayed. Figure 6.23 shows a close-up view of the cube mesh with nodes numbered.

It is important to realize that INGRID will not label every node. It will only label those nodes which are visible in the current orientation of the model. Even then, some nodes will remain unlabeled if INGRID determines that they are too close together.

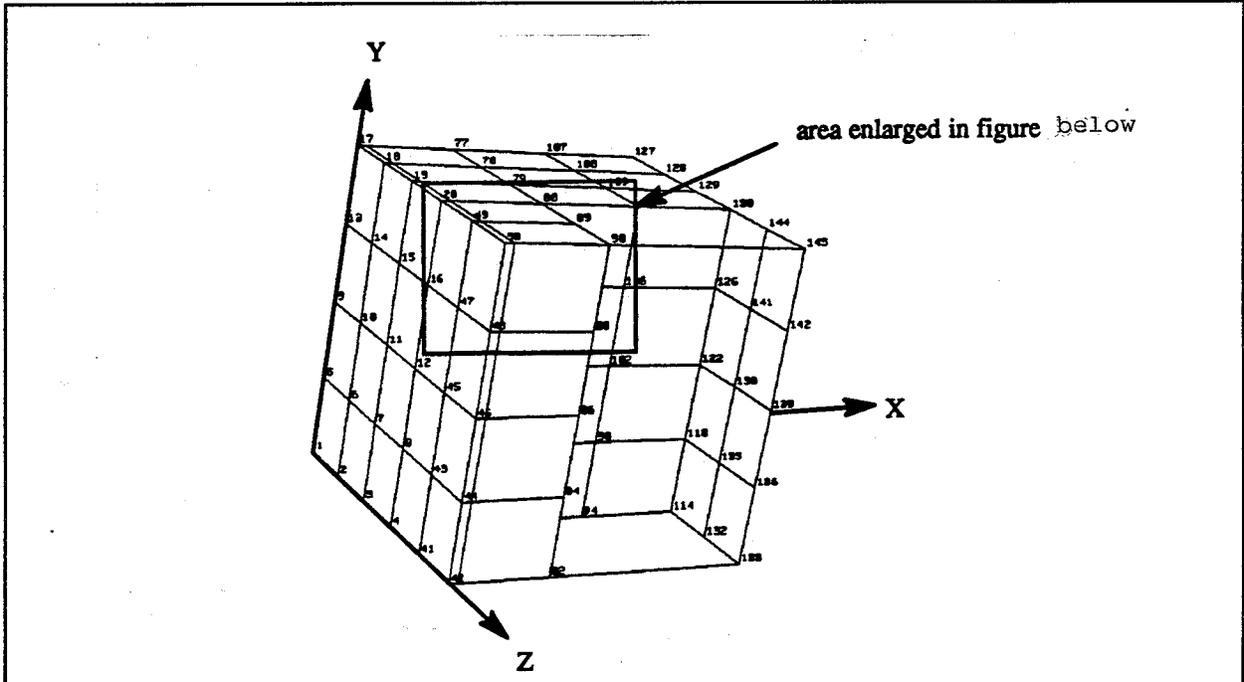


Figure 6.22
di nodes - Modified unit cube with node numbers displayed.

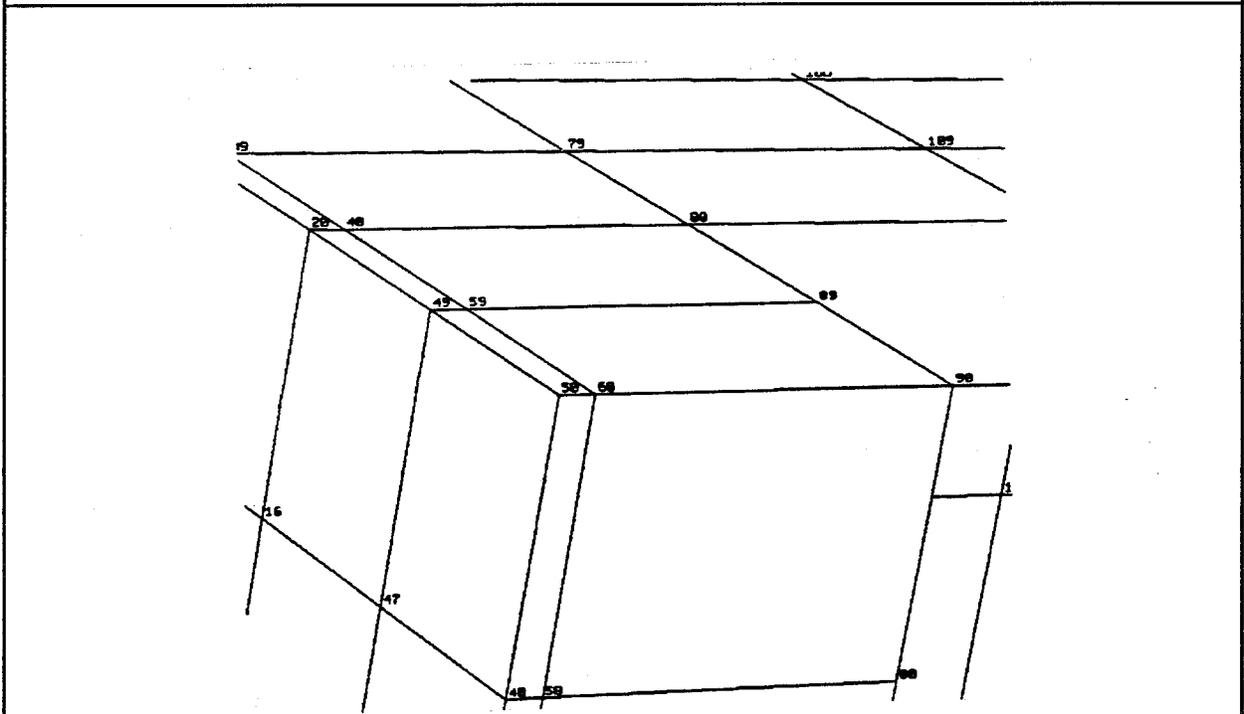


Figure 6.23
di nodes - Close of modified unit cube with node numbers displayed.

Element numbers can be displayed in the much same way as the node numbers. Since there are three types of elements, there are three **di** commands which enable the display of element numbers: **di 3d**, **di 2d**, and **di 1d**. These commands enable the element labeling for solid, shell, and beam elements, respectively. An algorithm similar to that for the node numbering is used to determine the density of element numbers on the display. If there is not enough room to place the element number, an element will remain unlabeled. If it is necessary to determine a specific element number which is not displayed, simply the change current orientation, zoom, pan, or do all three to free up enough display space for the labels to be rendered.

Often surfaces or regions of nodes are given boundary conditions such as forces, fixed displacements, velocities, or pressures. It is possible to verify that the boundary conditions have been prescribed at the correct nodes, and that the correct load curves have been used with the **di f**, **di d**, **di vb**, and **di pr** commands. When used with a display command, **di f** will highlight those nodes subjected to concentrated force loads. **di d** will highlight prescribed displacements. **di vb** will highlight nodes subject to a velocity boundary condition, and **di pr** will identify those nodes subjected to a pressure load. In each case, a load curve number must be specified in addition to the type of boundary condition which is to be displayed.

It is possible to designate nodes subjected to convection boundary conditions as well. Temperature initial conditions and boundary conditions can also be established with INGRID. In order to highlight those nodes subject to a convection boundary condition, the **di cv** command can be used. Nodes subject to an initial temperature condition or a prescribed temperature boundary condition can be highlighted by using the **di ti** or **di tb** commands, respectively.

6.13 Non-Graphical Diagnostic Information Display

There are a number of interactive commands which provide insight into the non-graphical characteristics of the finite element model. In order to illustrate them, the teardrop casing example has been further refined by giving it material attributes and an initial velocity. The outer shell was given the material type of elastic steel. The inner shell was made out of elastic aluminum.

It is possible to determine the spatial extent of the model using the **size** command. When executed, INGRID will scan through the nodal database to determine the minimum and maximum values of all of the coordinates along each axis. Using that information, it is possible to determine a rectangular bounding box in which the entire model will fit. For the shell casing, execution of the **size** command would result in the following output:

```
>> size
0.00000e+00 << x << 2.40792e+00
0.00000e+00 << y << 2.40792e+00
6.00000e-01 << z << 1.12000e+01
```

Information about the number of various types of elements can be determined using the **info** command. When entered, INGRID calculates the number of each element type and prints a small summary after echoing the title from the INGRID input file. Executing the **info** command for the casing mesh shown in Figure 6.15 would result in the following output:

```
>> info
c --- example two shell casing
# of parts = 2
# of nodes = 3258
# of beams = 0
# of plates = 0
# of solids = 2156
```

While **info** provides information general to the entire model, the **pinfo** command provides specific information regarding the number of nodes, beam, plate, and solid elements within a particular part. The particular part, or parts, in which the information is generated for is dictated by the active display list. Had the active display list included both parts in the teardrop model, the following output would have been generated:

```
>> pinfo
part number = 1
nodes = 1629 first = 1 last = 1629
beams = 0 first = 1 last = 0
plates = 0 first = 1 last = 0
bricks = 1078 first = 1 last = 1078
part number = 2
nodes = 1629 first = 1630 last = 3258
beams = 0 first = 1 last = 0
plates = 0 first = 1 last = 0
bricks = 1078 first = 1079 last = 2156
```

There are three commands which can be used to obtain information related to the mass properties of the model - **mass**, **pmass**, and **tmass**. The **mass** command will tabulate each of the volumes, masses, and centers of mass for each of the parts in addition to a total. For the teardrop casing, the output would appear as follows:

```
>> mass
part mat volume mass x-center y-center z-center
1 1 1.73966e+01 4.69702e+01 9.00365e-01 9.00362e-01 7.14219e+00
2 2 6.08880e+00 4.81014e+01 1.17154e+00 1.17153e+00 6.95401e+00
all all 2.34854e+01 9.50716e+01 1.03756e+00 1.03756e+00 7.04698e+00
```

The **pmass** command generates more detailed information related to the mass of the model. In addition to providing mass and volume information, the **pmass** command will print the cumulative centroid, initial kinetic energy, initial momentum, moments or inertia, and products of inertia for the parts in the current active parts list. If the entire model is currently being displayed, the derived values will correspond to the sum of all of the parts. For the teardrop casing **pmass** produces the following output if both parts were in the active display list:

```
>> pmass
partial system mass = 9.50714e+01
volume = 2.34854e+01
centroid: x = 1.03756e+00 y = 1.03756e+00 z = 7.04697e+00
kinetic energy = 4.92564e+00
momentum: xm = 0.00000e+00 ym = 0.00000e+00 zm = -3.06035e+01
moments and products of inertia about centroid:
ixx = 5.97662e+02 iyy = 5.97662e+02 izz = 7.71861e+01 ixy = 1.38378e+01
ixz = -2.87041e+01 iyz = -2.87041e+01
volume = 2.34854e+01
```

It is also possible to obtain the mass characteristics of the entire model using the **tmass** command. **tmass** displays the same information which **pmass** provides, with the exception that the command is not influenced by the current active parts display list. A sample of the output generated by the **tmass** command on the model in figure is as follows:

```
>> tmass
total system mass = 9.50714e+01
volume = 2.34854e+01
centroid: x = 1.03756e+00 y = 1.03756e+00 z = 7.04697e+00
kinetic energy = 4.92564e+00
momentum: xm = 0.00000e+00 ym = 0.00000e+00 zm = -3.06035e+01
moments and products of inertia about centroid:
ixx = 5.97662e+02 iyy = 5.97662e+02 izz = 7.71861e+01 ixy = 1.38378e+01
ixz = -2.87041e+01 iyz = -2.87041e+01
volume = 2.34854e+01
```

An additional command is available in INGRID which provides centroid information about the model. The **centroid** command calculates much of the same information which the **mass**, **pmass**, and **tmass** commands calculate.

The **mass**, **pmass**, **tmass**, and **centroid** commands calculate the moments, products of inertia and the model centroid. It is possible, however, to calculate the moments and products of inertia about an arbitrary point by using the **reference** command. When **reference** is used to alter the point about which the moments and products of inertia are calculated from the centroid of the model to the point (0.0, 0.0, 3.5) for the teardrop casing, the output of those commands will subtly change from:

```
moments and products of inertia about centroid:
ixx = 5.97662e+02 iyy = 5.97662e+02 izz = 7.71861e+01
ixy = 1.38378e+01 ixz = -2.87041e+01 iyz = -2.87041e+01
```

to,

```
moments and products of inertia about a point:
x = 0.00000e+00 y = 0.00000e+00 z = 3.50000e+00
ixx = 1.89610e+03 iyy = 1.89610e+03 izz = 2.81882e+02
ixy = -8.85103e+01 ixz = -3.78586e+02 iyz = -3.78586e+02
```

Of course, it is possible to restore the calculations of the moments and products of inertia back to the centroid of the model using the **center** command.

The **ajnp** command locates specific nodes in physical space. This command accepts the x, y and z-coordinates of a point in physical space, and returns the node number and coordinates of the closest node. For example, in the teardrop casing example, the following **ajnp** command was used to locate the nearest node to the coordinates (0.0, 1.0, 7.0):

```
>> ajnp 0.0 1.0 7.0
nearest node is      1025   with coordinates
( 0.00000e+00,  1.14552e+00,  6.80445e+00)
```

6.14 Output Generation

Once it is verified that all of the properties of the mesh are correct and that the proper nodes have been merged, it is necessary to exit the interactive phase of INGRID and generate the necessary output file. The INGRID output file then becomes the input file for the analysis code for which it was generated. Several types of output can be created. By default, INGRID will generate formatted output information targeted to the specific code for which the model was created. That target code is specified by the analysis output option detailed in Chapter 4. To create the formatted analysis input file, the **cont** command should be used.

If, instead, it is determined that there are errors in the mesh, it will be necessary to exit INGRID and correct those errors before producing an input file for the analysis code. Two commands exist which terminate the execution of the INGRID software: **stop** and **end**. The **stop** command causes to terminate immediately. A more graceful termination of INGRID will occur when the **end** command is executed.

6.15 Interactive Input Errors

Input during the interactive phase is entirely format free. INGRID parses the input information, one command at a time, applying the desired change to the mesh or display until the end of the 80 character input line is reached. If an error does occur, INGRID will issue an error message and then continue processing the rest of the commands entered on the input line. For example, if the following line was used to rotate and magnify the mesh:

```
>> rx 20 ry 40 rz10 zf 3 poor
```

INGRID would generate the message:

```
illegal graphics command: rz10
```

because there was no space between the z-rotation command and its argument. After generating the error message, INGRID would proceed to zoom in on the mesh and display it using the poor man's algorithm. No z-axis rotation would be applied in this case.

7.0 INGRID EXAMPLES & PART LIBRARY

This chapter presents example INGRID input files for generating models for DYNA3D, NIKE3D, TOPAZ3D and MONT3D. (The DYNA3D examples included in this chapter correspond to the examples presented in “DYNA3D Example Problem Manual”, Steve C. Lovejoy and Robert G. Whirley, UCRL-MA-105259, 1990.) The examples in this chapter are intended to illustrate the commands necessary to generate complete input files for the MDG suite of analysis codes. The second part of this chapter is devoted to a part library which can serve as a reference for users generating new meshes. All of the INGRID input files in this chapter include the interactive commands used to generate the images presented in the text.

7.1 DYNA3D Examples

bar impact problem (gm cm microsec) INGDY1.dat
dn3d

```
c Set some DYNA analysis options.
c Terminate dynamic time integration at time 80.0.
term 80.0
c Node and element data dump interval for TAURUS post-processing.
plti 1.0
c Node and element data dump interval for high speed printer.
prti 81.0
```

```
c Assign an initial rigid-body velocity to all parts following
c this command.
velocity 0 0 -0.0227
```

```
c Define three symmetry planes.
plane 3 0 0 0 0 -1 0 .001 symm
0 0 0 -1 0 0 .001 symm
0 0 0 0 0 1 .001 symm
```

```
c Define bar. For the new user it is advised to run each command
c interactively in INGRID so each step is fully understood.
```

```
c Begin part definition.
start
```

```
c Define index space.
1 4 7 10 13; 1 4 7 10 13; 1 37;
c Define coordinates of index space.
-.16 -.16 0 .16 .16 -.16 -.16 0 .16 .16 0 3.24
```

```
c First the mesh is shaped into a cylindrical bar and then
c three-quarters of the cylinder is deleted.
```

```
c Delete corners for the cylindrical mapping.
c The empty list of indices in k specifies that all indices in
c the k direction are included in the command.
di 1 2 0 4 5; 1 2 0 4 5;;
```

```
c Capture the surfaces to be mapped and then project them to the
c cylindrical surface.
sfi -1 -5; -1 -5;;
cy 0 0 0 0 0 1 .32
```

```
c Delete all but one-quarter of the cylinder.
c The zeroes in the i, j or k indices specify that all indices in
c those directions are included in the command.
```

```
d 0 1 0 0 3 0
d 1 0 0 3 0 0
```

```
c Specify the material of the part.
mate 1
```

```
c End part definition.
end
```

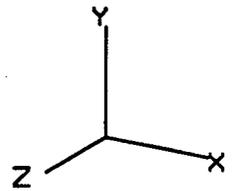
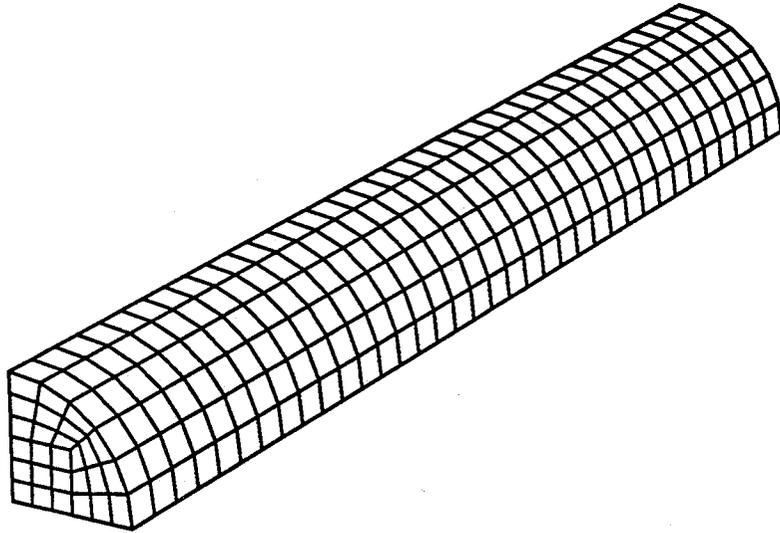
```
c Define material properties for material number 1.
mat 1 3
ro 8.93
sigy 0.004
e 1.17
etan 0.001
beta 1.0
pr 0.33
endmat
```

```
c End of INGRID input.
end
```

```
c Use stp 0.001 interactively in INGRID to remove duplicate nodes
c that are within a distance 0.001 of each other.
c Interactive commands can be put at the end of the input file
c so that INGRID will execute them automatically when it reads in
c the file.
stp 0.001
```

```
ry -30
rx 20
set tv display
```

bar impact problem (gm cm microsec) INGDY1.dat
INGRID display z-rotation condition



Cylinder drop calculation (in,sec,lb s²/in) INGDY2.dat
dn3d

c Set some DYNA analysis options.

term 0.01

plti 0.0001

prti 1.0

c Define two symmetry planes and a stonewall.

plane 3

0 0 0 0 0 1 0.001 symm

0 0 0 -1 0 0 0.001 symm

0 -4.5 0 0 1 0 .001 ston

c Define a sliding interface.

si 1 dummy;

c Define rail.

start

c Define index space and corresponding coordinates.

1 2; 1 2; 1 2;

0 5 -7.5 -4.5 0 -.75

c Define master side of slide surface. The rail has fewer nodes in

c the slide surface definition, therefore it should be the master

c side.

sii 1 2; 2 2; 1 2; 1 m

c Note that no material has been assigned to this part.

end

c Assign an initial rigid-body velocity to the following parts.

velocity 0 -660 0

c Define cylinder.

start

c Define index space and corresponding coordinates. This space is

c in cylindrical coordinates

1 4; 1 13 40; 1 5 9 27;

4.25 4.5 -90 -34.6153846 90 0 -.75 -1.5 -6

cyli

c Define slave nodes of stonewall

sw 2 1 1 2 2 2 3

c Define slave nodes of slide surface

sii 2 2; 1 2; 1 2; 1 s

c Assign a material.

mate 1

end

```
c Define support ring.
start
c Define index space and corresponding coordinates.
1 5; 1 13 40; 1 5;
4.25 3.25 -90 -34.6153846 90 -5 -6
cyli
c Assign a material.
mate 2
end

c Define material properties.
c Cylinder.
mat 1 12
c steel
ro 0.7346e-03
g 11.33e+06
sigy 1.90e+05
eh 0.0
k 24e+06
endmat

c Support ring.
mat 2 1
ro 0.147333e-01
e 30e+06
c Shear modulus equals 22.66e+06.
pr 0.3
endmat

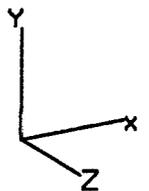
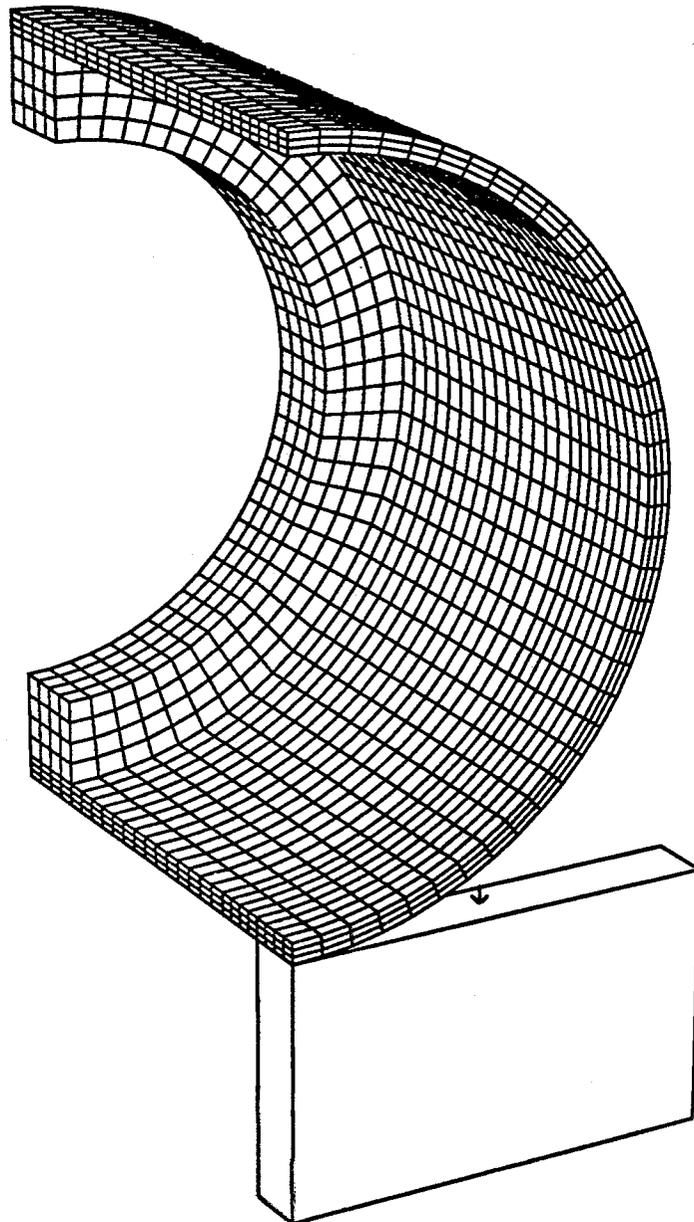
end

c use tp 0.001 interactively in Ingrid to remove duplicate nodes.
tp 0.001

ry 30
rx 20
di si 1 b

set tv display
```

Cylinder drop calculation (in,sec,lb s²/in) INGDY2.dat
INGRID display slide surface 1



Bar impacting bar (non-dimensional) INGDY3.dat
dn3d

```
c Set some DYNA analysis options.
term 1.5
plti .01
prti 2.0

c Define a sliding interface.
si 1 sv;
c Define index space and corresponding coordinates for right half.
start
52 102; 1 2; 1 2;
50.0 100 0 1 0 1
c Fix the y and z translation by setting a boundary condition.
b 1 1 1 2 2 1 011000
b 1 1 2 2 2 2 011000
c Define slave side of slide surface.
si 1 1 1 1 2 2 1 s
c Assign a material.
mate 1
end

c Assign an initial rigid-body velocity for the following parts.
velocity 0.1 0 0

c Define index space and corresponding coordinates for left half.
start
1 51; 1 2; 1 2;
0 50 0 1 0 1
c Fix the y and z translation by setting a boundary condition.
b 1 1 1 2 2 1 011000
b 1 1 2 2 2 2 011000
c Define master side of slide surface.
si 2 1 1 2 2 2 1 m
c Assign a material.
mate 1
end
c Define material properties.
mat 1 1
ro 0.01
e 100
pr 0.0
endmat
end

ry 50
rx 20
set tv display
```

Bar impacting bar (non-dimensional) INGDY3.dat
INGRID display

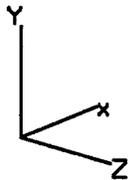
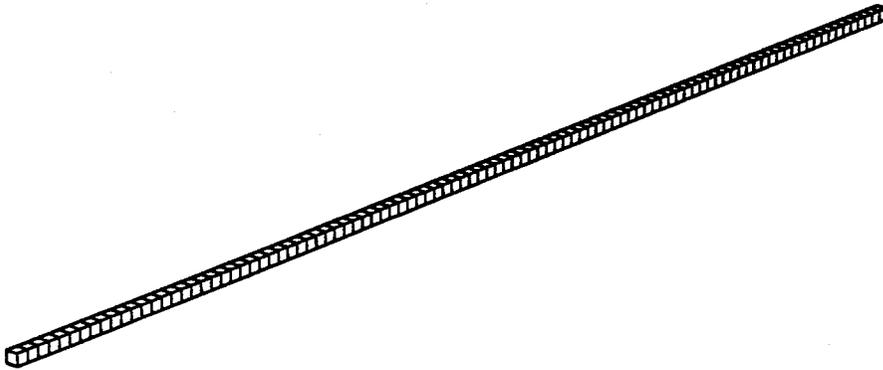


Plate impact by rod (cm, gm, microsec) INGDY4.dat
dn3d

```
c Set some DYNA analysis options.
term 1.0e+04
plti 1.0e+03
prti 1.0e+05

c Define symmetry planes.
plane 2
0 0 0 0 -1 0 .01 symm
0 0 0 -1 0 0 .01 symm

c Define a sliding interface.
si 1 sv ;

c Define main plate.
start
1 9 51 63; 1 9 51 63; -1;
0 4 25 31 0 4 25 31 5
c Specify plate thickness.
th 1 1 1 4 4 1 0.79
c Orient plate slave surface towards impacting rod surface.
c Always check shell surface normal vectors!
si+ 1 1 1 2 2 1 1 s 0 0 10
c Assign a material.
mate 1
end

c Define left support plate.
start
-1; 1 50; 1 2 11;
25 0 24.5 5 4.5 0
c Fix translational DOF on bottom edge of support plate.
b 1 1 3 1 2 3 111000
c Assign a material.
mate 2
end

c Define upper support plate.
start
1 50; -1; 1 2 11;
0 24.5 25 5 4.5 0
c Fix translational DOF on edge of support plate.
b 1 1 3 2 1 3 111000
c Assign a material.
mate 2
end
```

```
c Define rod
velocity 0 0 -1.8e-03
start
1 4 7 10 13;1 4 7 10 13;1 51;
-2 -2 0 2 2 -2 -2 0 2 2 5.005 30
d 0 1 0 0 3 0
d 1 0 0 3 0 0
di 1 2 0 4 5;1 2 0 4 5;;
sfi -1 -5;-1 -5;;
cy 0 0 0 0 0 1 4
c Define bottom surface of rod as the master side of slide surface.
c This surface has fewer nodes than the surface of the plate, i.e.a
c master has many slaves.
si 1 1 1 5 5 1 1 m
c Assign a material.
mate 3
end

c Set tolerance for merge between main plate and supports.
c Use tp 0.0001 interactively before the continue command.
bptol 1 2 0.001
bptol 1 3 0.001

c Define material properties.

c Main plate.
mat 1 3 shell
ro 7.85
e 2.1
etan 0.0124
sigy 0.004
pr 0.3
beta 1.0
endmat

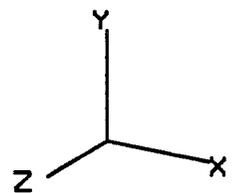
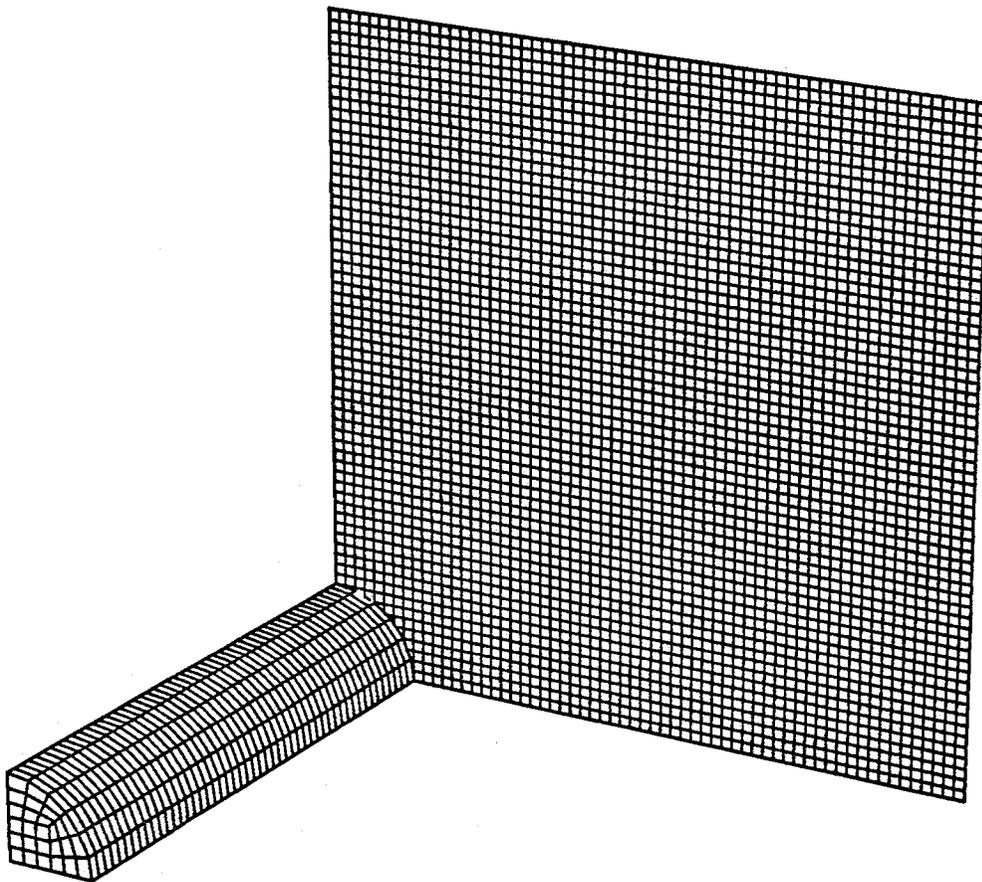
c Support plates.
mat 2 3 shell
ro 7.85
e 2.1
etan 0.0124
sigy 0.00215
pr 0.3
beta 1.0
thick 0.5
endmat

c Rod.
mat 3 20
ro 19.218
```

```
e 2.1
pr 0.0
endmat

end
c Use tp 0.0001 interactively in INGRID!
tp 0.0001
ry -30
rx 20
set tv display
```

Plate impact by rod (cm, gm, microsec) INGDY4.dat
INGRID display



Beam buckle #1 (in,sec,lb s²/in) INGDY5.dat
dn3d

c Set some DYNA analysis options.

term 1.72e-02

plti 1.72e-03

prti 1.0

c Define a load curve.

lcd 1 2 0 1 1 1

c Define two symmetry planes.

plane 2

1.375 0 0 2 0 0 0.01 symm

0 1.375 0 0 2 0 0.01 symm

c Define a sliding interface.

si 1 single;

c Define first side of beam.

start

1 11; -1; 1 74 75 76 91;

0 1.375 0 0 9.7333 9.8667 10 12

c Slave sliding surface.

si 1 0 1 2 0 5 1 s

c Boundary conditions.

b 1 1 1 2 1 1 111111

b 1 1 4 2 1 5 110111

c Velocity boundary condition which uses the load curve

c defined above.

fv 1 1 4 2 1 5 1 273 0 0 -1

c Offset the y coordinate of the nodes in the specified region.

c This is used to put a small crimp in the beam.

mb 1 0 3 2 0 3 2 -.02

c Set the x coordinate of the specified point (so the two sides

c meet correctly at the crimp.)

pa 1 0 3 1 -.02

c Assign a material.

mate 1

end

c Define second side of beam.

start

-1;1 11;1 74 75 76 91;

0 0 1.375 0 9.7333 9.8667 10 12

si 0 1 1 0 2 5 1 s

b 1 1 1 1 2 1 111111

b 1 1 4 1 2 5 110111

fv 1 1 4 1 2 5 1 273 0 0 -1

```
mb 0 1 3 0 2 3 1 -.02
pa 0 1 3 2 -.02
mate 1
end
```

```
c Define material properties.
```

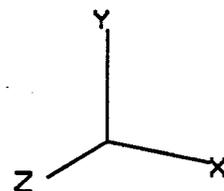
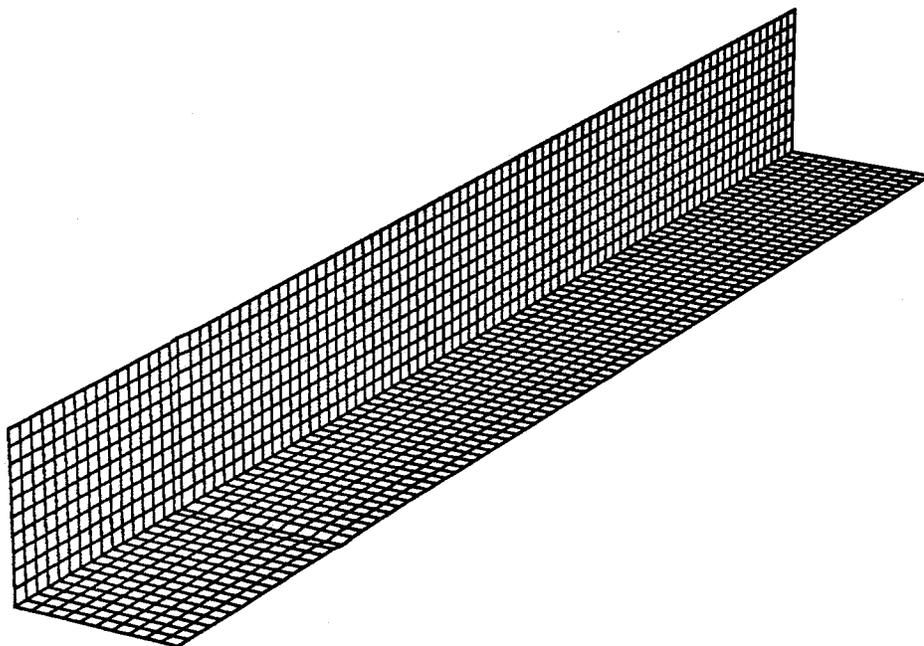
```
mat 1 3 shell
ro 7.1e-04
e 30e+06
etan 60e+03
pr 0.3
sigy 30e+03
beta 1.0
thick 0.06
tsti 3
endmat
```

```
end
```

```
c Use tp 0.001 interactively in INGRID!
```

```
tp 0.001
ry -30
rx 20
scale 1.2
set tv display
```

Beam buckle #1 (in,sec,lb s²/in) INGDY5.dat
INGRID display



Beam element example (in,sec,lb-s²/in)Quarter model INGDY6.dat
dn3d

c Set some DYNA analysis options.

term 1.0e-03

plti 1.0e-05

prti 9.0e-04

c Define symmetry planes.

plane 2

0 0 0 0 -1 0 0.01 symm

0 0 0 -1 0 0 0.01 symm

c Define slide surface type 3.

si 1 sv;

c Begin node definition for beam elements.

c Always define beam elements before other element types.

c Bottom ring nodes.

beam

cy 111111 1 0 0 c Node 1

cy 000000 1 15 10 c Node 2

cy 111111 1 30 0 c Node 3

cy 000000 1 45 10 c Node 4

cy 111111 1 60 0 c Node 5

cy 000000 1 75 10 c Node 6

cy 111111 1 90 0 c Node 7

c Nodes 2,4,6,and 16 are used for aligning cross-sectional
c properties of beam elements.

c Middle nodes.

cy 000110 1.01 0 1 c Node 8

cy 000110 1.01 30 1 c Node 9

cy 000110 1.01 60 1 c Node 10

cy 000110 1.01 90 1 c Node 11

c Top ring nodes.

cy 000000 1 0 2 c Node 12

cy 000000 1 30 2 c Node 13

cy 000000 1 60 2 c Node 14

cy 000000 1 90 2 c Node 15

c Normal vector node for vertical elements.

cy 000000 0 0 1 c Node 16

0

c Begin beam element definition.

c Elements for bottom ring.

1 3 1 1 1 2

3 5 1 1 1 4

5 7 1 1 1 6

c Elements for top ring.

```
12 13 1 1 1 2
13 14 1 1 1 4
14 15 1 1 1 6
c Vertical elements from bottom too middle.
1 8 5 1 1 16
3 9 5 1 1 16
5 10 5 1 1 16
7 11 5 1 1 16
c Vertical elements from top too middle.
12 8 5 1 1 16
13 9 5 1 1 16
14 10 5 1 1 16
15 11 5 1 1 16
0
end

c Define top plate.
start
1 3 5; 1 4; -1;
0.5 1 1.5 0 90 2.0
cyli
c Normal vector of slide surface points towards solid ram part.
si+ 1 1 1 3 2 1 1 s 0 0 3
mate 3
end

c Set a node tolerance between parts 1 and 2.
bptol 1 2 0.001
c Define solid ram.
velocity 0 0 -1000
start
1 3 5;1 4;1 4;
0.5 1 1.5 0 90 2.20 3.0
cyli
si 1 1 1 3 2 1 1 m
c Fix x and y displacement of ram.
b 1 1 1 3 2 2 110000
mate 2
end
c Define material properties.

c Beam properties.
mat 1 3 beam
e 30.e+06
etan 30.e+03
sigy 50000
pr 0.3
ro 2.77e-04
beta 1.0
```

```
sthi 0.25
tthi 0.25
endmat
```

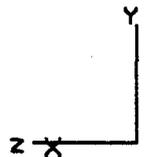
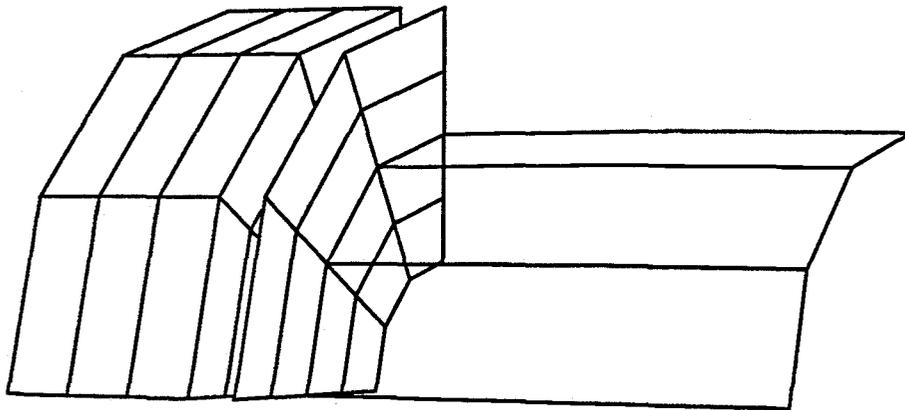
```
c Ram properties.
mat 2 1
e 30.e+07
pr 0.3
ro 2.77e-03
endmat
```

```
c Plate properties.
mat 3 1 shell
e 30.e+06
ro 2.77e-04
pr 0.3
thick 0.125
endmat
```

```
end
```

```
c Use tp 0.0001 interactively in INGRID!
tp 0.0001
ry -120
scale 2
l 0.5
set tv display
```

Beam element example (in,sec,lb-s²/in)Quarter model INGDY6.dat
INGRID display



Impulsively loaded thin beam (in,sec,lb sec²/in) INGDY7.dat
dn3d

c Set some DYNA analysis options.

term 2.0e-03

prti 1.0

plti 1.0e-05

c Define index space and cooresponding coordinates for outer
c portion.

start

3 11; 1 2; -1;

1.0 5.0 0. 0.6 0.

c Set lengthwise edge boundary conditions.

b 1 1 1 2 1 1 010101

b 1 2 1 2 2 1 010101

c Set fixed end boundary condition.

b 2 1 1 2 2 1 111111

mate 1

end

c Define index space and cooresponding coordinates for center
c portion

c that is given initial velocity.

start

1 3; 1 2; -1;

0 1.0 0 .6 0

c Set symmetry boundary conditions on center.

b 1 1 1 1 2 1 110111

c Set lengthwise edge boundary conditions.

b 1 1 1 2 1 1 010101

b 1 2 1 2 2 1 010101

velocity 0. 0. -5000.

mate 1

end

c Define material properties.

mat 1 3 shell

c 6061-T6 AL

ro .261e-03

e 10.4e+06

pr 0.33

sigy 41.4e+03

etan 0.0

beta 1.0

thick 0.125

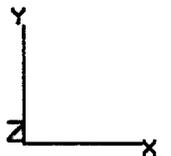
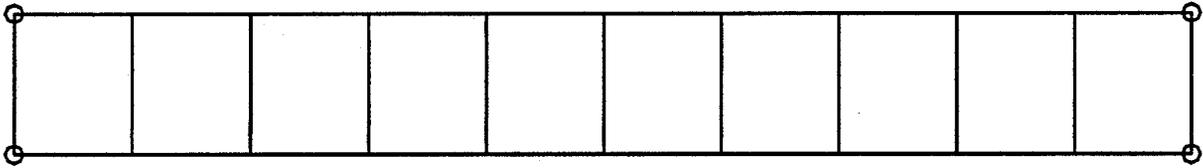
tsti 5

endmat

end

```
c Use tp 0.001 interactively in INGRID!  
tp 0.001  
scale 0.8  
di ry  
set tv display
```

Impulsively loaded thin beam (in,sec,lb sec²/in) INGDY7.dat
INGRID display y-rotation condition



Cylindrical shell (in,sec,lb s²/in) INGDY8.dat
dn3d

c Set some DYNA analysis options.

term 1.0e-03

plti 1.0e-05

prti 1.0

c Define symmetry plane.

plane 1

0 1 0 -1 0 0 0.0001 symm

c Define inner portion of cylinder with initial velocity.

velocity 0.0 -5650 0

start

-1; 1 5; 2 14;

2.938 90 52.5 -0.785 -10.205

cyli

mate 1

end

c Define outer portion of cylinder, note velocities are reset to zero.

c If this were not the case all elements in this part would have the

c above initial velocity.

velocity 0 0 0

start

-1; 1 5 9; 1 2 14 17;

2.938 90 52.5 30 0 -.785 -10.205 -12.56

cyli

c Fix all displacements on bottom edge.

b 1 3 1 1 3 4 111111

c Fix x and y translation on ends.

b 1 1 1 1 3 1 110000

b 1 1 4 1 3 4 110000

c Delete region that corresponds to the first part definition.

d 1 1 2 1 2 3

mate 1

end

c Define material properties

c 6061 t-6 AL

mat 1 3 shell

thick 0.125

ro 2.5e-04

e 10.5e+06

etan 0.0

pr 0.33

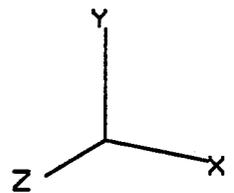
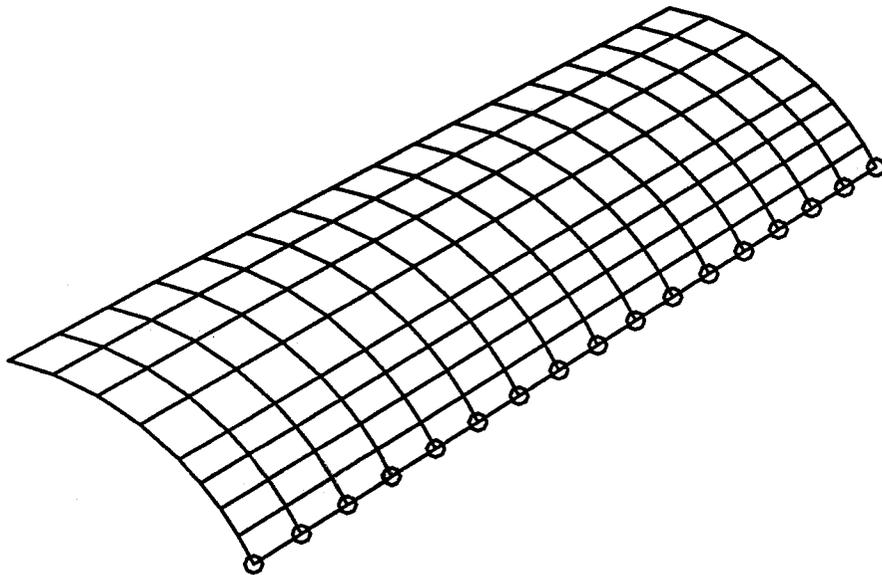
sigy 44.0e+03

```
beta 1.0  
tsti 5  
endmat
```

```
end
```

```
c use tp 0.001 interactively in INGRID!  
tp 0.001  
ry -30  
rx 20  
di dz  
set tv display
```

Cylindrical shell (in,sec,lb s²/in) INGDYB.dat
INGRID display z-displacement condition



Simply supported plate (in,sec,lb s²/in) INGDY9.dat

dn3d

c Set some DYNA analysis options.

term 1.2e-03

plti 2.0e-05

prti 2.0e-04

c Define a load curve.

lcd 1 2 0 1.0 1.0 1.0

c Define symmetry planes.

plane 2

1.0 0.0 0.0 0 -1 0 0.001 symm

0.0 1.0 0.0 -1 0 0 0.001 symm

start

c Define index space and cooresponding coordinates.

1 5; 1 5; -1;

0 5.0 0 5.0 0.0

c Fix y displacement on outside edges.

b 2 1 1 2 2 1 001000

b 1 2 1 2 2 1 001000

c Define applied pressure region.

pr 1 1 1 2 2 1 1 300.0 0 0 -1

c Specify element printout blocks.

epb 0.0 0.0 0.0 0.625 0.625 0;

c Assign material.

mate 1

end

c Specify hourglassing parameters.

qh 0.05

q1 1.2

c Define material.

mat 1 3 shell

c Aluminum.

ro .2588e-03

e 10.0e+06

pr 0.3

sigy 100.0e+03

etan 0.0

beta 1.0

thick 0.5

tsti 5

endmat

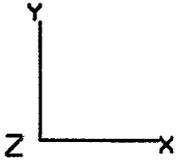
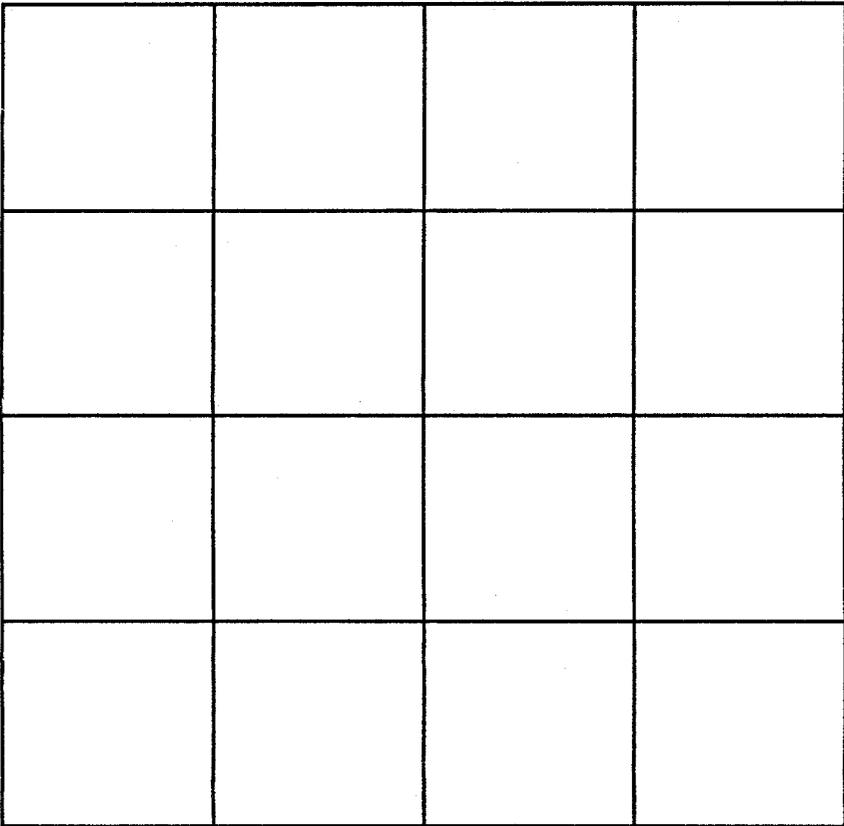
end

scale 0.8

set tv display

Simply supported plate (in,sec,lb s²/in) INGDY9.dat

INGRID display



7.2 NIKE3D Examples

pipe whip problem by R. ferencz

nk3d

c setup the nike3d specific analysis options

anal

dyn

bwmo off

nbsr 1

nbei 1

nibsr 10

msrf 15

dctol .001

ectol .01

iprt 99

iplt 1

nstep 200

term 10e-3

c setup the materials

mat 1 3

shell

ro 7.298e-4

e 3.0e+7

pr .3

sigy 1.0e+5

etan 1.0e+5

mat 2 3

shell

ro 7.298e-4

e 3.0e+7

pr .3

sigy 1.0e+5

etan 1.0e+5

endmat

c setup a symmetry plane

plan 1 0 0 0 1 0 0 .001 symm

coor 2 rt 0 0 0 rt 1 0 0 sp 1 90 -16

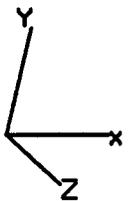
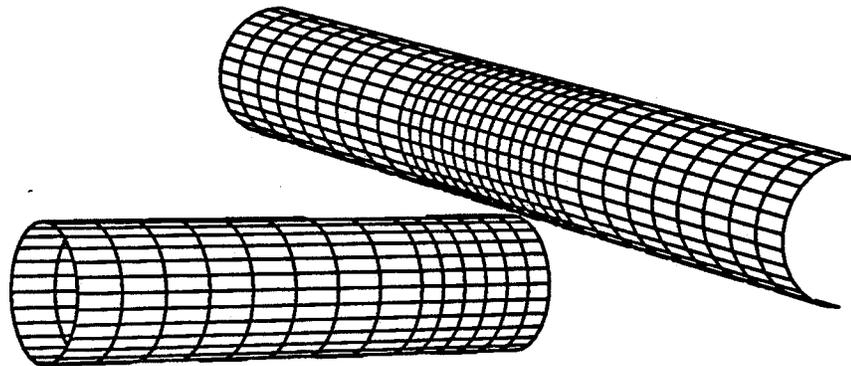
rt 0 0 25 rt 0 0 26 rt 0 1 25

```
c this is the first pipe
start
-1;1 8 15;1 10 24 33;
3.3125 90 180 270 0 18 32 50
cyli thic .432
b 1 1 1 1 1 4 100011
b 1 3 1 1 3 4 100011
b 1 2 1 1 2 1 111000
si- 1 2 2 1 3 3 1 1 0 0 0
rota 1 0 0 75 0 0
repe 1;
end
```

```
c this is the second pipe
start
-1;1 29;1 7 15;
3.3125 0 360 0 7 25
cyli thic .432
n- 1 1 1 1 2 2 20 0 12.5
b 0 0 3 0 0 3 111111
b 0 0 1 0 0 1 100011
si- 1 1 1 1 2 2 1 0 0 0 25
mate 2
repe 2;
end
```

```
end
set tv disp
rx -25
ry 30
```

pipe whip problem by R. ferencz
INGRID display



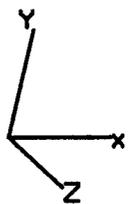
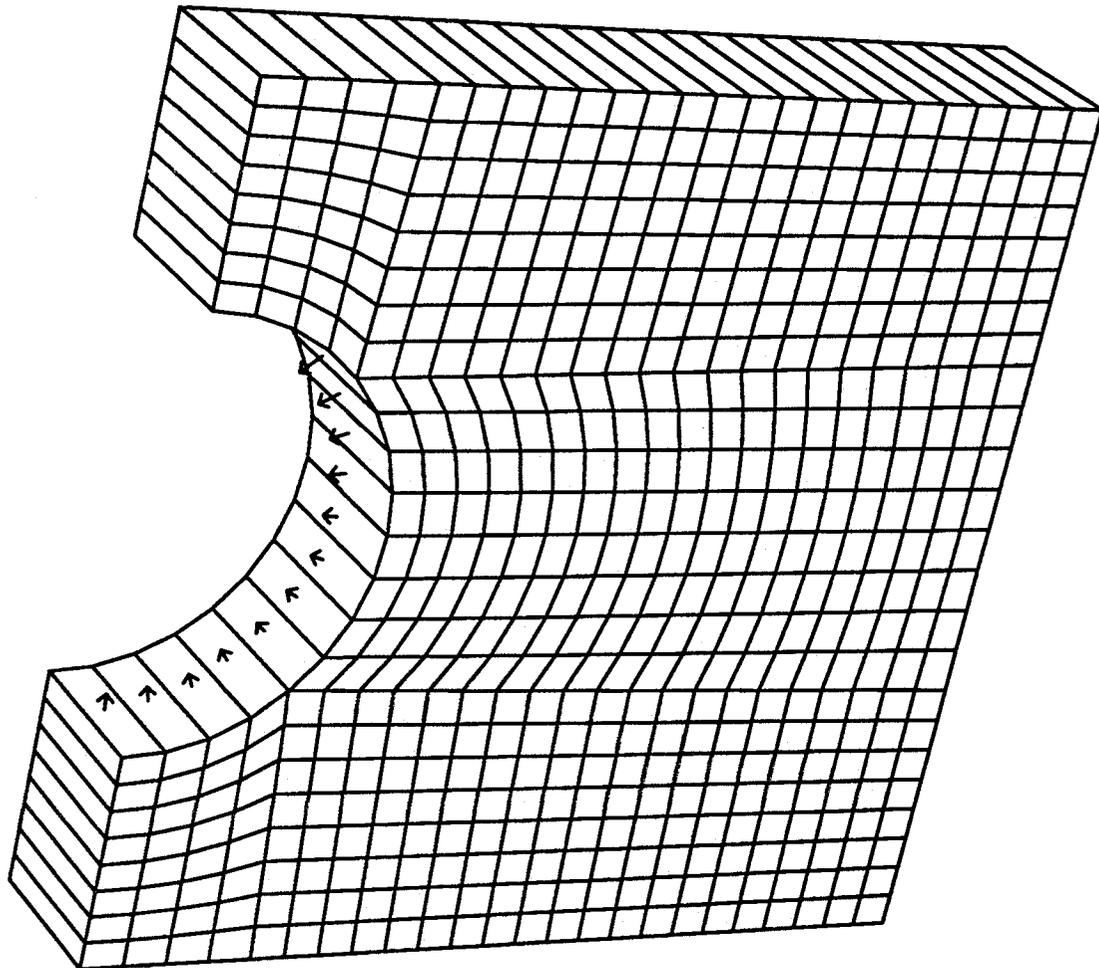
7.3 TOPAZ3D Examples

crucible cooling model, two inch cooling tube spacing

c this input file courtesy of Donald L. Brown, Uranium AVLIS, LLNL

```
tz3d
c
c define problem parameters
iunit kelv
bwmo off
flux on
steady
linear
dctol 0.0001
relax 0.8
c
c load curves
lcd 1 2 0.0 1.0 1.0 1.0
c
c material data
mat 1 1
  head
  copper crucible
  cp 0.43
  k 3.60
endmat
start
1 5 25 ; 1 9 17 25 ; 1 2 ;
0.0 0.50 2.54 0.0 0.76 1.78 2.54 0.0 0.5
d 1 2 0 2 3 0
c circle radius 0.25 centered at x=0, y=0.5 parallel to the z axis
sf 1 2 1 2 2 2 cy 0.0 1.27 0.0 0.0 0.0 1.0 0.635
sf 2 2 1 2 3 2 cy 0.0 1.27 0.0 0.0 0.0 1.0 0.635
sf 1 3 1 2 3 2 cy 0.0 1.27 0.0 0.0 0.0 1.0 0.635
mate 1
c flux on the inside
fli ; -4; ; 1 -100.0
c convection on the inside of the tube
cv 1 2 1 2 2 2 0 4.0 0 20.0 0.0
cv 2 2 1 2 3 2 0 4.0 0 20.0 0.0
cv 1 3 1 2 3 2 0 4.0 0 20.0 0.0
c initial temperature
tm 0 0 0 0 0 0 100.0
end
end
set tv disp
rx 25
ry 30
di cv disp
```

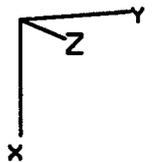
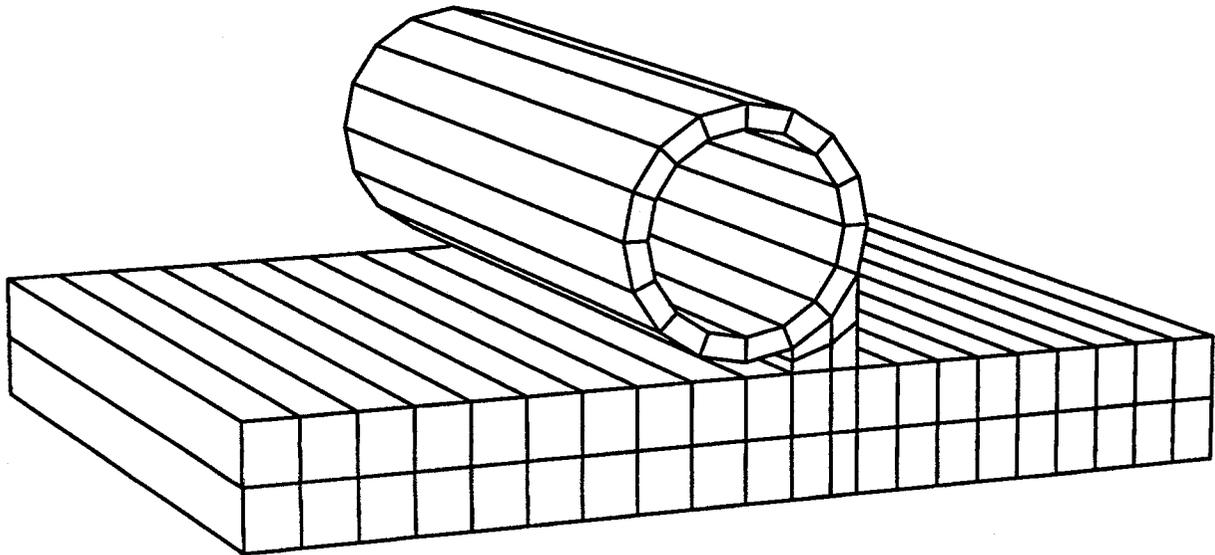
crucible cooling model, two inch cooling tube spacing
INGRID display convection boundary



```
1/4"plt 1/2"x049tube 2"spc 0.030"gap tube/plt 603braze mtl
c ingrid input file ingis1
c This file courtesy of J.Phil Brady, AVLIS Program,
c Thermal-Structural Group
tz3d
tp 0.005
c
c define problem parameters
iunit kelv
bwmo off
flux on
radiation exch
steady
nonlinear
dctol 0.000001
relax 0.8
sbc 5.67e-12
rctol 1.0e-6
c
c material data
c
mat 1 1
  ro 1.73
  head
  copper
  cp 0.43
  k 3.60
mat 2 1
  ro 1.73
  head
  copper
  cp 0.43
  k 3.60
mat 3 1
  ro 1.0
  head
  braze603
  cp 1.0
  k 0.43
endmat
c ingrid part 1 water tube copper .5" od x .049" wall
start
1 2 ; 1 17 ; 1 2 ;
0.511 0.635 0.0 360.0 0.0 4.0
cyli
mate 1
cvi -1 ; ; ; 0 1.62 0 300.0 0.0
tm 0 0 0 0 0 0 350.0
end
```

```
c copper plate 1/4" thick
start
1 3 ; 1 10 11 12 13 22 ; 1 2 ;
.711 1.346 -2.54 0 .243 .449 0.587 2.54 0 4.
mate 2
fli -2 ; ; ; 0 -40.0
tm 0 0 0 0 0 0 350.0
end
c ingrid part 3 braze joint
start
1 3 ; 1 2 3 4 ; 1 2 ;
.635 .711 0 .243 .449 .587 0 4.
pb 1 2 0 1 2 0 x .587
pb 1 3 0 1 3 0 x .449
pb 1 4 0 1 4 0 x .243
d 0 1 0 0 2 0 c delete braze under tube
mate 3
end
end
set tv disp
rz -90
ry 22
rx 10
disp
```

1/4"plt 1/2"x049tube 2"spc 0.030"gap tube/plt 603braze mt1
INGRID display flux



7.4 MONT3D Examples

enclosure radiation problem for mont3d

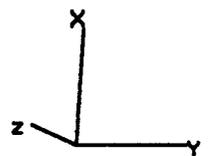
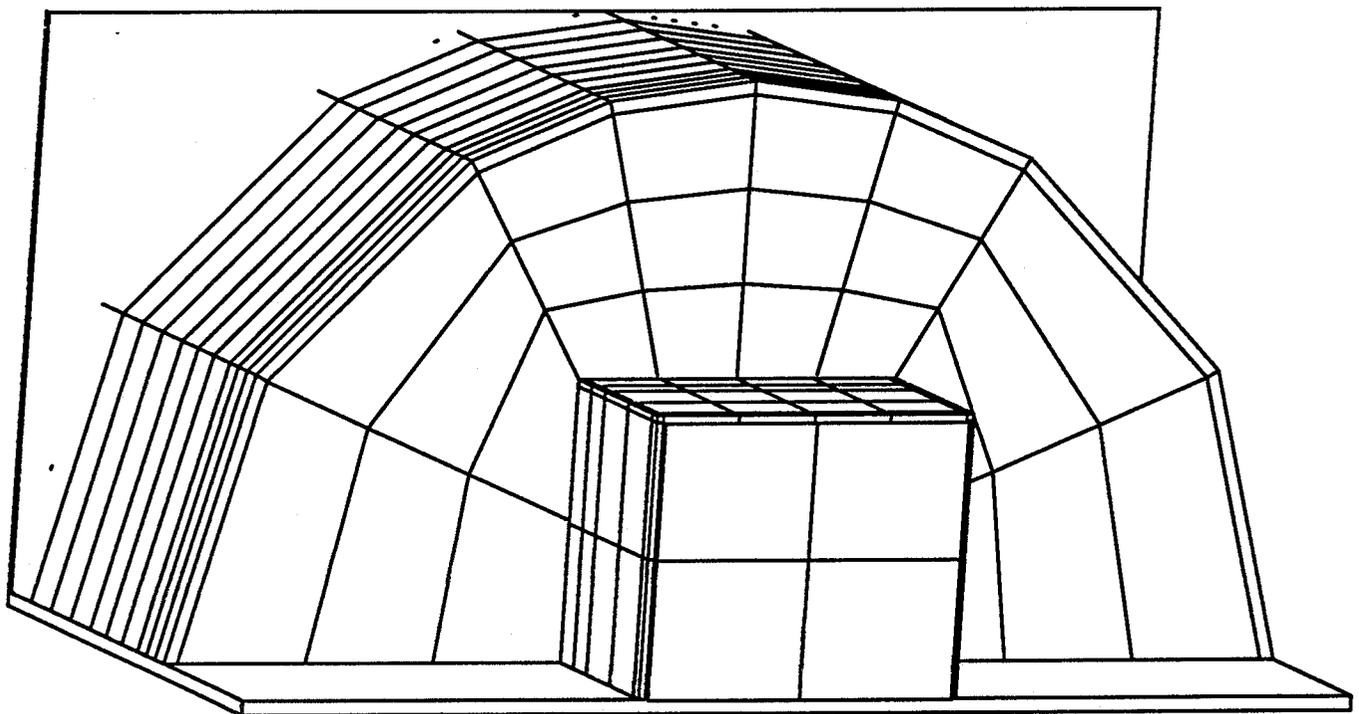
```
c B. Kornblum 4-21-92
c This input generates the mont3d input file, including
c control cards, but ingrid can't make the material properties
c
monte
c
c control card information
delt 0.001
ibug 0
idata 0
c iprint 0
nbands 1
ndivx 20
ndivy 25
nlost 999
nphton 50
nrefs 999
nwarns 888
numcat 0
nshade -1
ngx 2
ngy 3
ngz 4
xscale 1.0
yscale 1.0
zscale 1.0
xshift 0.0
yshift 0.0
zshift 0.0
c
c file has units of inches, convert to centimeters
csca 2.54
c
c create cylinder with endplate with rectangular hole in it
start
1 2 5 7 9 12 13 ;
1 2 5 7 9 12 13 ;
1 4 7 11 ;
-20.5 -20.5 -20.5 0.0 20.5 20.5 20.5
-12.0 -12.0 -12.0 0.0 12.0 12.0 12.0
0.0 5.0 14.0 29.0
c delete corners
di 1 3 0 5 7 ; 1 3 0 5 7 ; 1 4 ;
c create the cylinder using the arc command
a 1 1 0 7 7 0 3 42.0
a 2 2 0 6 6 0 3 41.0
```

```
c delete rectangle in the circular endcap
d 3 3 0 5 5 0
c delete interior of the cylinder
d 2 2 2 6 6 4
c delete half of everything
d 1 0 0 4 0 0
c radiation enclosure
rei -6 ; 2 6 ; 2 4 ; 1 yes
rei 4 6 ; -2 ; 2 4 ; 1 yes
rei 4 6 ; -6 ; 2 4 ; 1 yes
rei 2 6 ; 2 6 ; -2 ; 1 yes
c
mate 1
end
c
c rectangular box
start
1 3 4 ; 1 2 6 7 ; 1 2 4 6 ;
  0.0 20.0 20.5
-12.0 -11.5 11.5 12.0
-15.0 -14.0 -5.0 0.0
d 1 2 1 2 3 4
c radiation enclosure
rei 1 2 ; 2 2 ; 2 3 ; 2 yes
rei 1 2 ; 3 3 ; 2 3 ; 2 yes
rei 2 2 ; 2 3 ; 2 3 ; 2 yes
rei 1 3 ; 1 4 ; 4 4 ; 2 yes
  mate 2
end
c
c
c plate to go on end of rectangular box
start
1 3 ; 1 3 ; 1 2 ;
0.0 20.0
-11.5 11.5
-15.5 -15.0
c radiation enclosure
re 1 1 2 2 2 2 3 yes
  mate 3
end
c
c
c plate to go on the end of the cylinder
c (does not need to fit exactly)
start
1 2 ; 1 2 ; 1 2 ;
0.0 42.0
-42.0 42.0
```

```
29.0 29.5
re 1 1 1 2 2 1 4 no
end
c
c
c symmetry mirror (does not need to fit exactly, just
c cover the area)
start
1 2 ; 1 2 ; 1 2 ;
-1.0 0.0
-42.0 42.0
-15.0 29.5
rei -2 ; 1 2 ; 1 2 ; 5 yes
end
c
end
set tv disp
angle .01
rz -90
rx -170
ry 22
disp
```

enclosure radiation problem for mont3d

INGRID display



7.5 INGRID Part Library

simple cube from a standard part with one element

dn3d

start

c Set up the index space.

1 2; 1 2; 1 2;

c Give the corresponding coordinates for the indices.

0.0 5.0

0.0 5.0

0.0 5.0

end

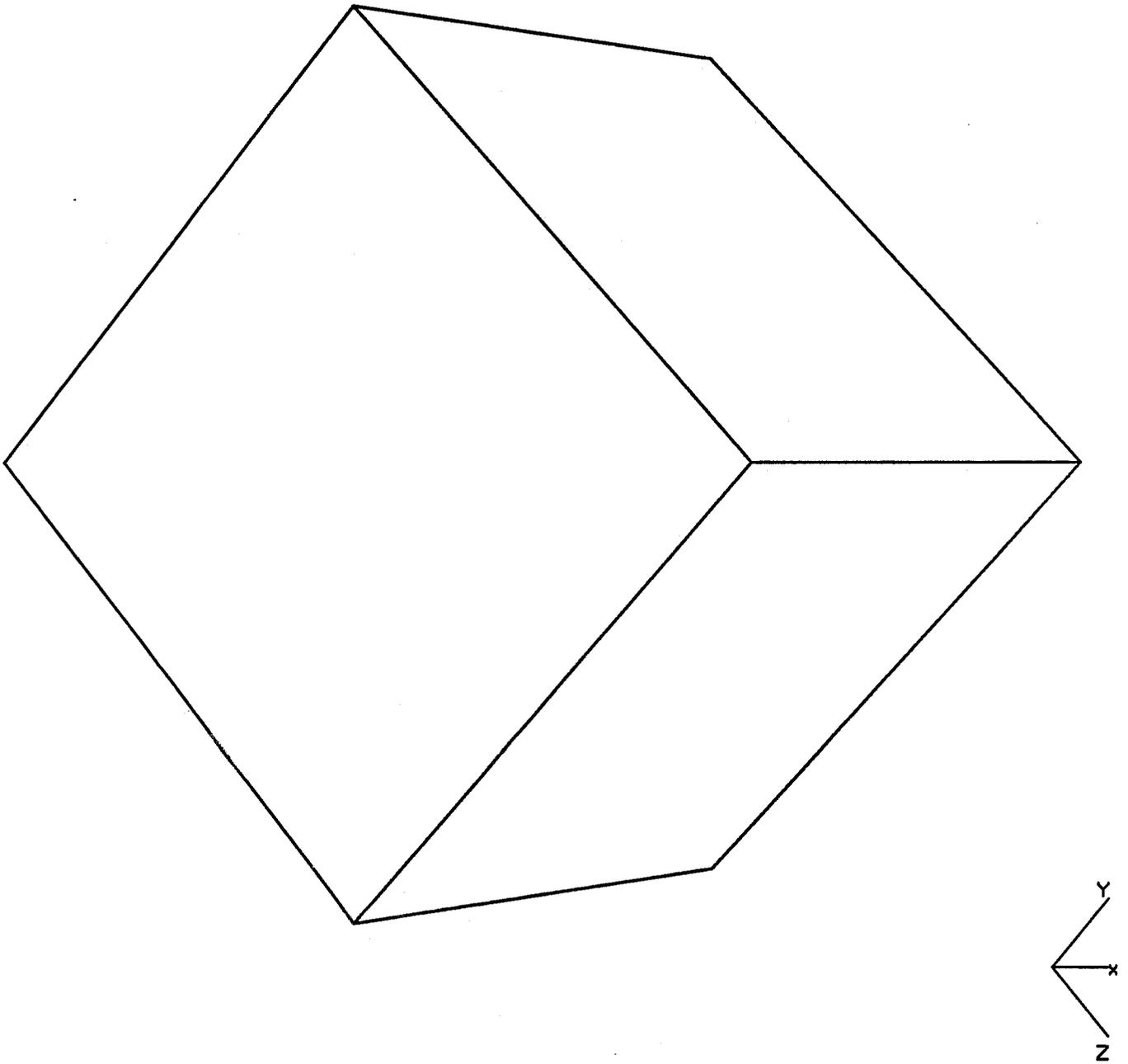
end

rx 45

ry 55

set tv display

simple cube from a standard part with one element
INGRID display



simple cube from a standard part with many elements
dn3d

start

c Set up the index space.

1 16; 1 16; 1 16;

c Give the corresponding coordinates for the indices.

0.0 5.0

0.0 5.0

0.0 5.0

end

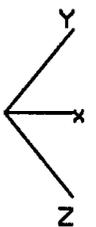
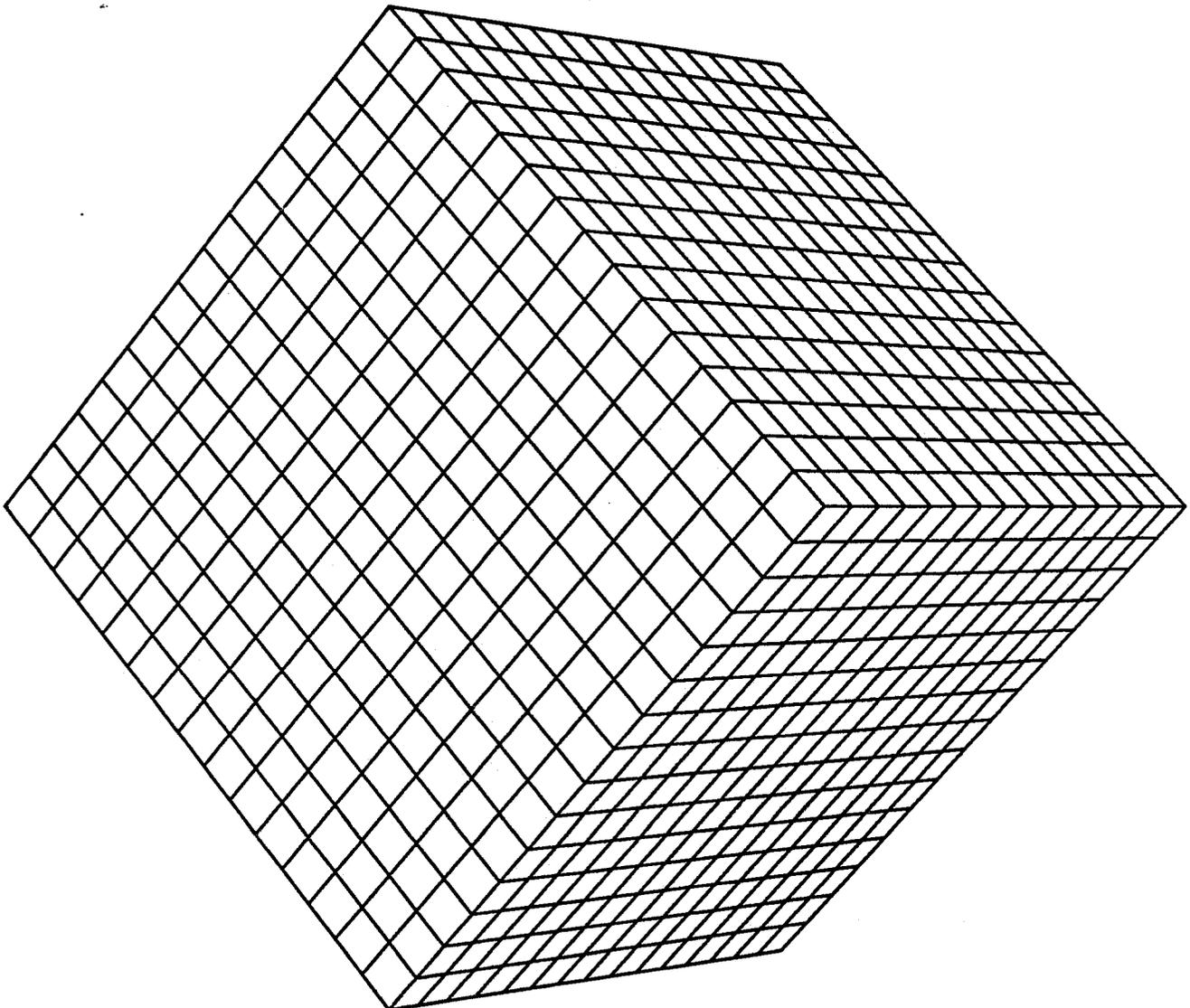
end

rx 45

ry 55

set tv display

simple cube from a standard part with many elements
INGRID display



simple cube from a standard part with varying mesh size

dn3d

start

1 6 11 16 21;

1 6 11 16 21;

1 6 11 16 21;

1.0 2.0 4.0 8.0 16.0

1.0 2.0 4.0 8.0 16.0

1.0 2.0 4.0 8.0 16.0

end

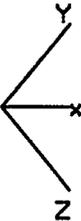
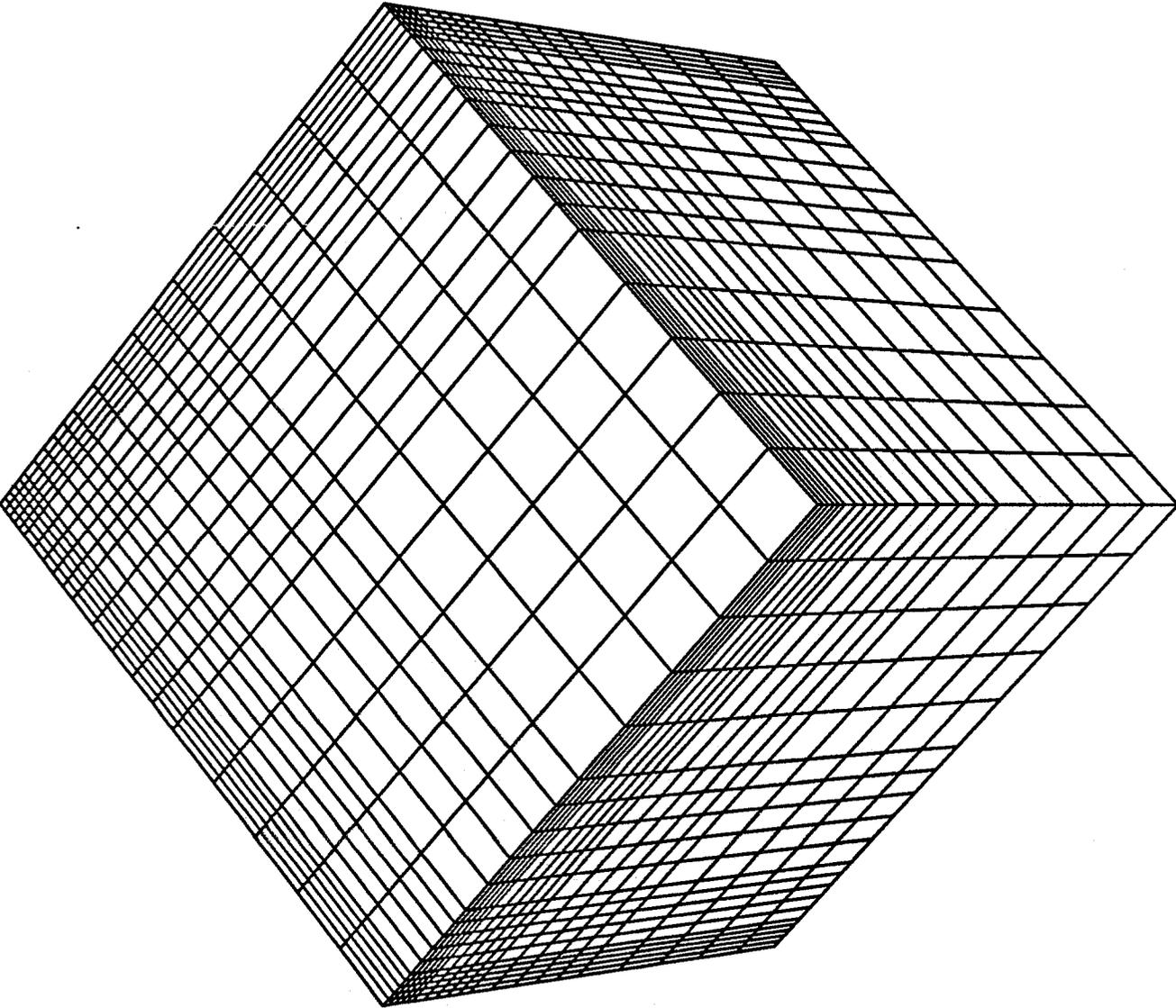
end

rx 45

ry 55

set tv display

simple cube from a standard part with varying mesh size
INGRID display



simple cube from a standard part with sections deleted

dn3d

start

1 6 11 16 21;

1 6 11 16 21;

1 6 11 16 21;

1.0 2.0 4.0 8.0 16.0

1.0 2.0 4.0 8.0 16.0

1.0 2.0 4.0 8.0 16.0

c Delete the eight regions at the corners of the cube.

di 1 2 0 4 5; 1 2 0 4 5; 1 2 0 4 5;

end

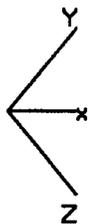
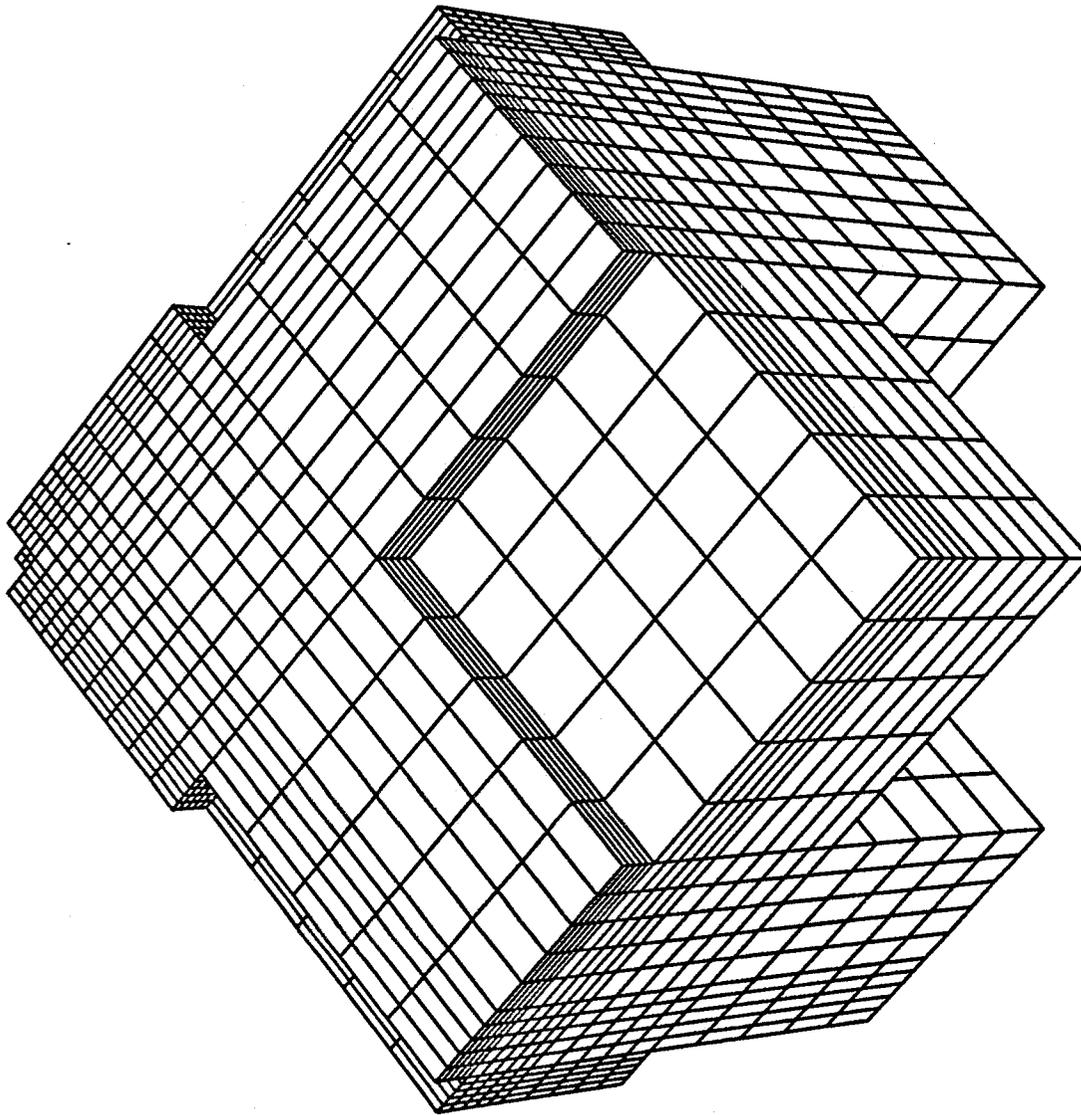
end

rx 45

ry 55

set tv display

simple cube from a standard part with sections deleted
INGRID display



simple example of a solid cylinder

dn3d

start

1 4; 1 37; 1 10;

1 2 0 360 1 14

c Interpret the previous coordinates as cylindrical.

cyli

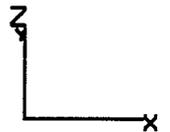
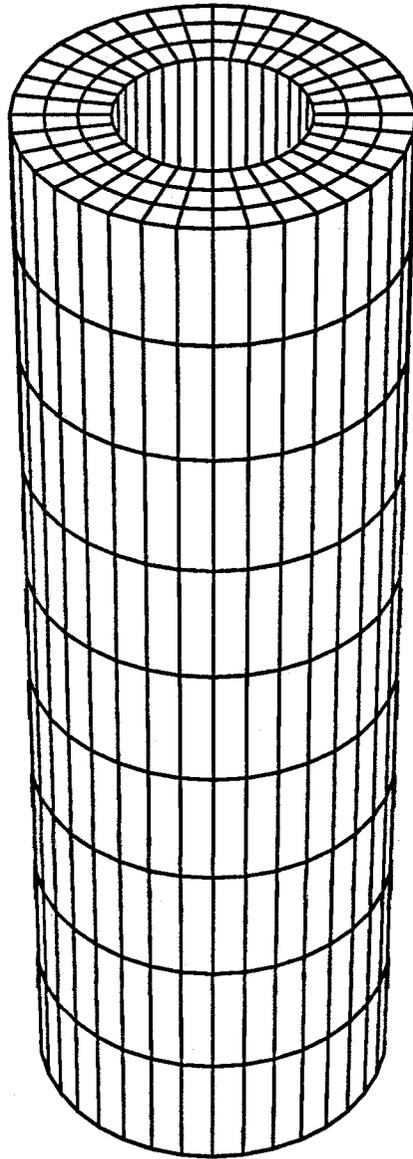
end

end

rx -50

set tv display

simple example of a solid cylinder
INGRID display



telescoping cylinders

dn3d

start

1 2 3 4 5 6; 1 37; 1 4 7 10 13 16;

1 2 3 4 5 6 0 360 1 2 3 4 5 6

c Interpret the previous coordinates as cylindrical.

cyli

c Delete the regions around the four upper rings, working from
c the bottom upward.

c Note that the zeroes in the j index specify that all indices

c in the j direction are included in the region.

d 2 0 5 3 0 6

d 3 0 4 4 0 6

d 4 0 3 5 0 6

d 5 0 2 6 0 6

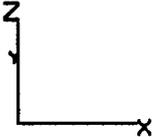
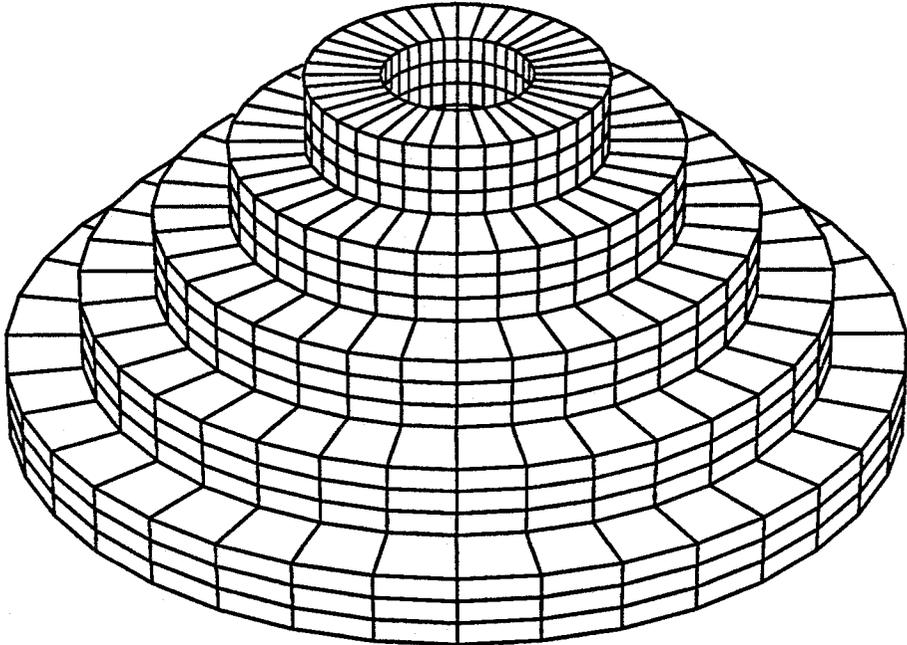
end

end

rx -60

set tv display

telescoping cylinders
INGRID display



simple solid sphere section

dn3d

start

1 4; 1 37; 1 21;

1.8 2 0 180 -80 80

c Interpret the previous coordinates as spherical.

sphe

end

end

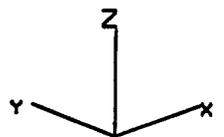
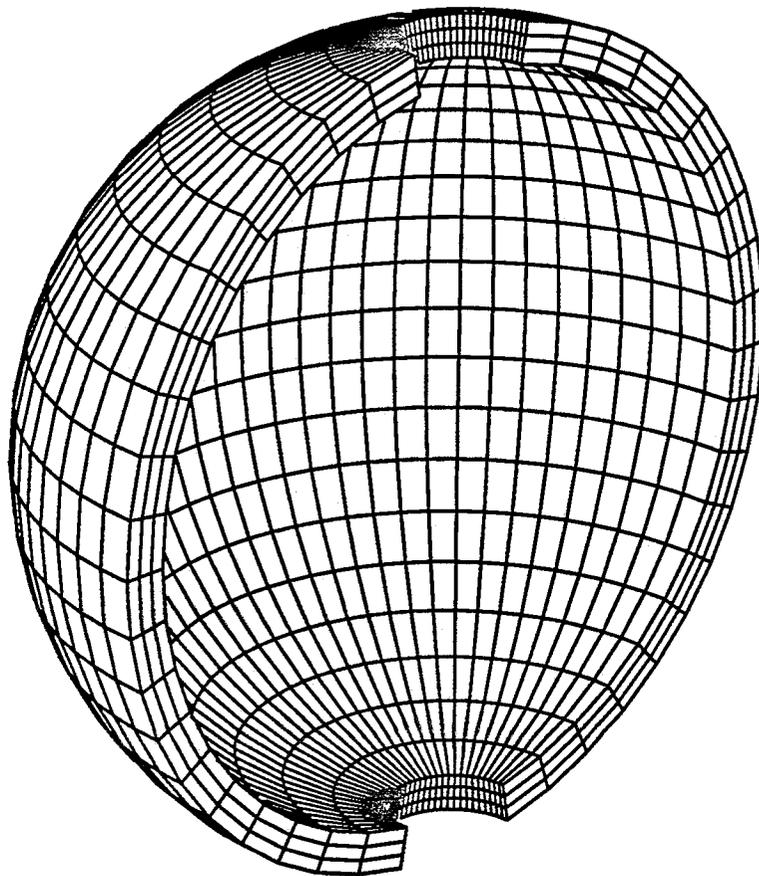
rx -60

ry 40

rz 20

set tv display

simple solid sphere section
INGRID display



simple circular plate

dn3d

c Begin the part definition.

start

c Set up the index space and specify the cooresponding coordinates.

1 5; 1 5; -1;

-1.0 1.0 -1.0 1.0 0.0

c Map the borders of the grid to a circle centered at the origin and

c having a radius of 1.0.

a 1 1 0 2 2 0 3 1.0

c End the part definition.

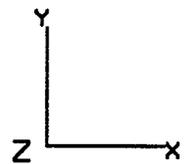
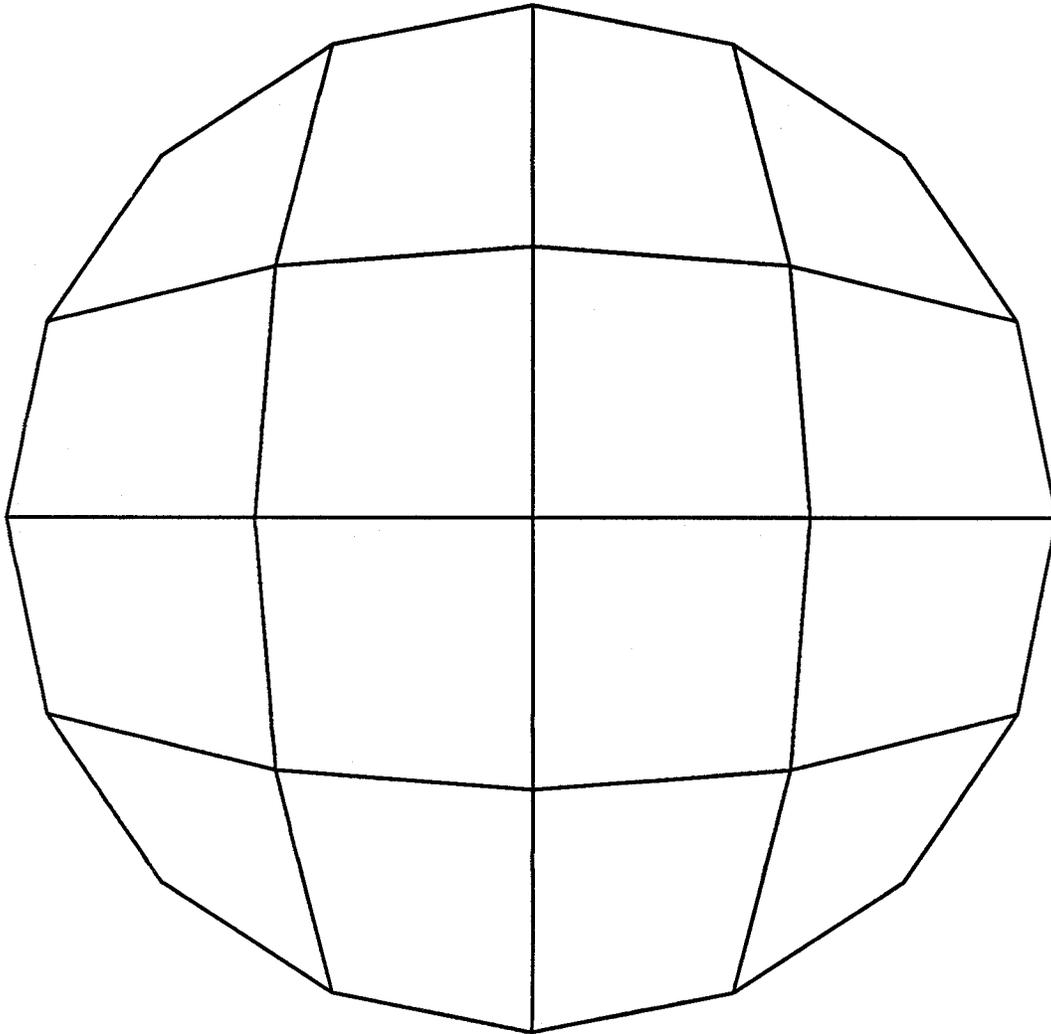
end

c End the input.

end

set tv display

simple circular plate
INGRID display



nicely zoned circular plate - solution #1

```
dn3d
start
c Note that the outer boundary is given the same coordinates as
c the middle boundary. This is so that the projection onto the
c circle will work correctly once the corner regions have been
c deleted.
1 3 7 9; 1 3 7 9; -1;
-.5 -.5 .5 .5
-.5 -.5 .5 .5
0
c Delete the four corner regions of the mesh.
d 1 1 0 2 2 0
d 1 3 0 2 4 0
d 3 1 0 4 2 0
d 3 3 0 4 4 0
c Project the outer boundary out to a circle centered at the origin
c and having a radius of 1.
a 1 1 0 4 4 0 3 1
end
end
set tv display
```

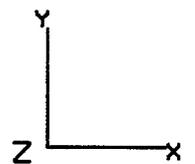
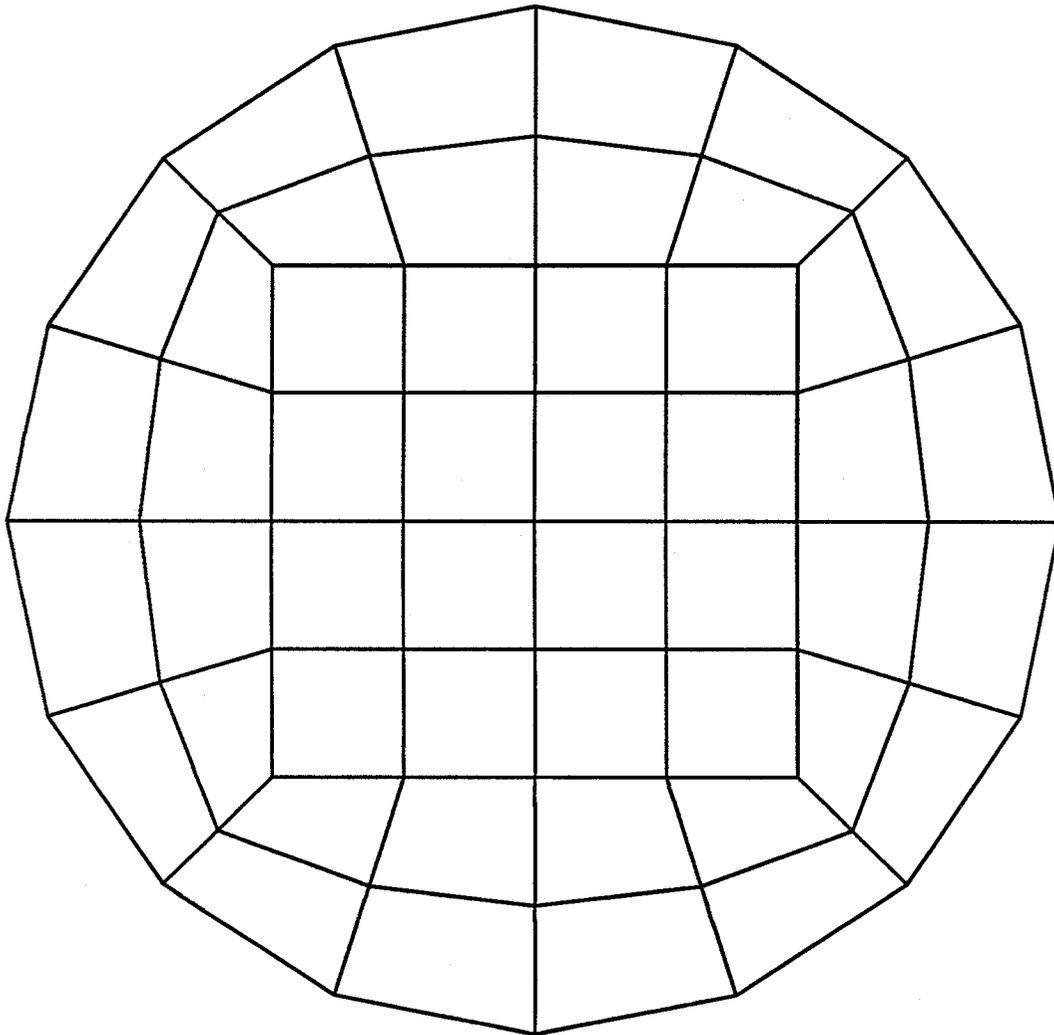
nicely zoned circular plate - solution #2

```
dn3d
start
1 3 7 9; 1 3 7 9; -1;
-.5 -.5 .5 .5
-.5 -.5 .5 .5
0
c Use an index progression to delete the four corner regions
c instead
c of deleting each corner region individually as in solution #1.
di 1 2 0 3 4; 1 2 0 3 4; -1;
a 1 1 0 4 4 0 3 1
end
end
```

nicely zoned circular plate - solution #3

```
dn3d
start
1 3 7 9; 1 3 7 9; -1;
-.5 -.5 .5 .5
-.5 -.5 .5 .5
0
c Another way of doing the same thing.
d 1 1 0 2 2 0 + i 2 + j 2 + i -2
end
end
```

nicely zoned circular plate - solution #1
INGRID display



half circular plate - solution #1

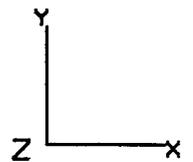
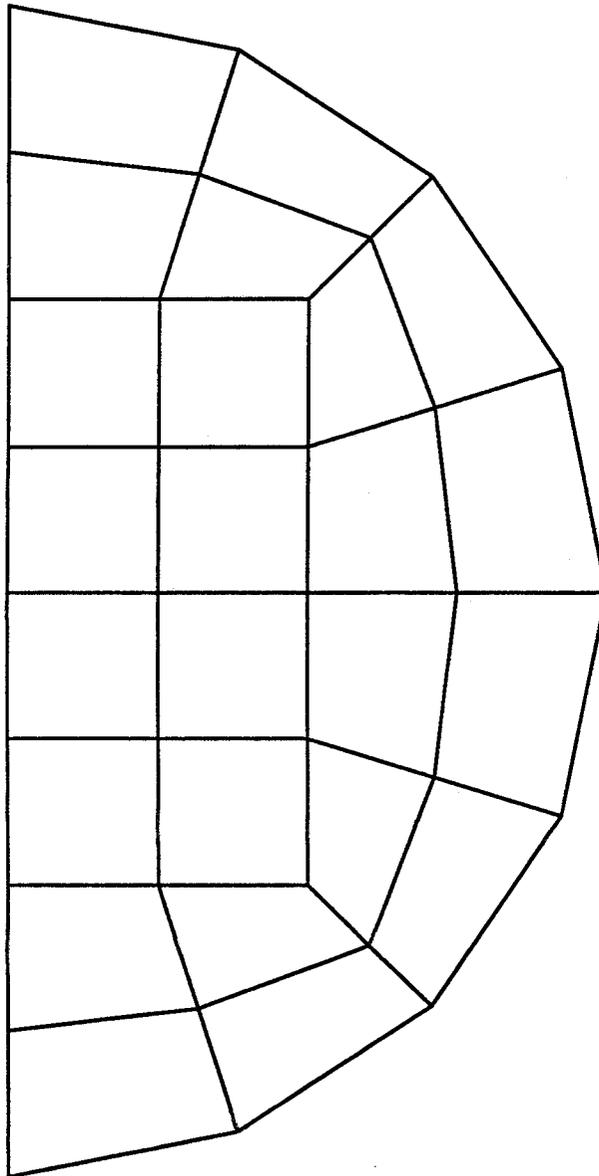
```
dn3d
start
1 3 5 7 9; 1 3 5 7 9; -1;
-.5 -.5 0 .5 .5
-.5 -.5 0 .5 .5
0
c Delete the four corner regions of the grid.
di 1 2 0 4 5; 1 2 0 4 5; -1;
c Project the region boundary out to a circle.
a 1 1 0 5 5 0 3 1
c Delete one half of the circular plate.
d 1 0 0 3 0 0
end
end
```

```
scale 0.9
set tv display
```

half circular plate - solution #2

```
dn3d
start
1 3 5; 1 3 7 9; -1;
0 .5 .5
-.5 -.5 .5 .5
0
c Delete the upper right and lower right corner regions of the grid.
di 2 3; 1 2 0 3 4; -1;
c Project the outer boundary out to a (half) circle.
c Project the right side.
ac 3 0 0 3 0 0 3 0 0 0 1 0 0 1
c Project the bottom.
ac 0 1 0 0 1 0 3 0 0 0 1 0 0 1
c Project the top.
ac 0 4 0 0 4 0 3 0 0 0 1 0 0 1
end
end
```

half circular plate - solution #1
INGRID display



quarter circular plate - solution #1

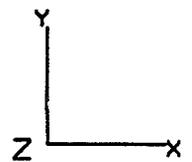
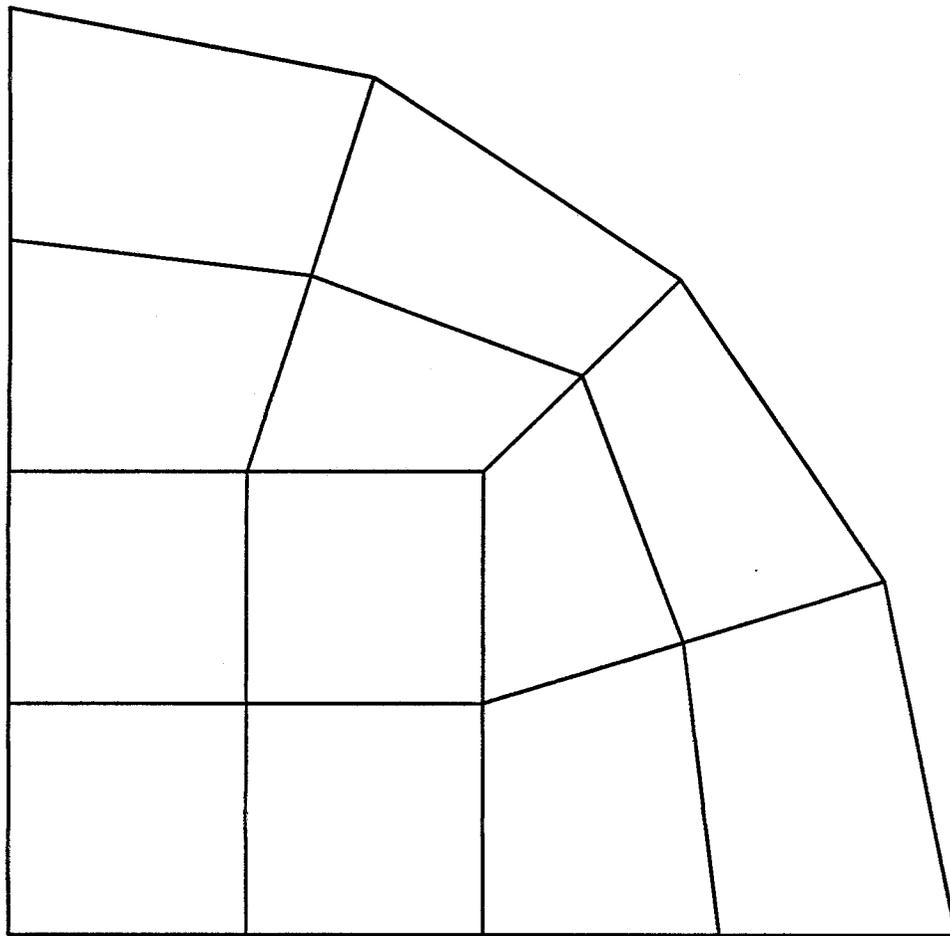
```
dn3d
start
1 3 5 7 9; 1 3 5 7 9; -1;
-.5 -.5 0 .5 .5
-.5 -.5 0 .5 .5
0
c Delete the four corner regions of the grid.
di 1 2 0 4 5; 1 2 0 4 5; -1;
c Project the region boundary out to a circle.
a 1 1 0 5 5 0 3 1
c Delete three-fourths of the circular plate.
d 1 0 0 3 0 0
d 0 1 0 0 3 0
end
end

scale 0.9
set tv display
```

quarter circular plate - solution #2

```
dn3d
start
1 3 5; 1 3 5; -1;
0 .5 .5
0 .5 .5
0
c Delete the corner region.
d 2 2 0 3 3 0
c Project the top and right boundaries out to a circle.
ac 3 0 0 3 0 0 3 0 0 0 1 0 0 1
ac 0 3 0 0 3 0 3 0 0 0 1 0 0 1
end
end
```

quarter circular plate - solution #1
INGRID display



circular plate with outer liner

dn3d

c Define materials.

mat 1 1

e 1.1 pr 1.2

c Set the material heading.

head

material example #1

endmat

mat 2 10

g 1.3 sigy 1.4 eh 1.5 a1 1.6 npts 3

c Set the material heading.

head

material example #2

endmat

start

1 3 5 11 13 15; 1 3 5 11 13 15; -1;

-.5 -.5 -.5 .5 .5 .5

-.5 -.5 -.5 .5 .5 .5

0

c Delete the corners.

di 1 3 0 4 6; 1 3 0 4 6; -1;

c Project the outer ring to the outer radius circle.

a 1 1 0 6 6 0 3 1.1

c Project the inner ring to the inner radius circle.

a 2 2 0 5 5 0 3 1

c Assign a material.

mate 2

c Change inner region to a different material.

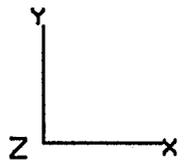
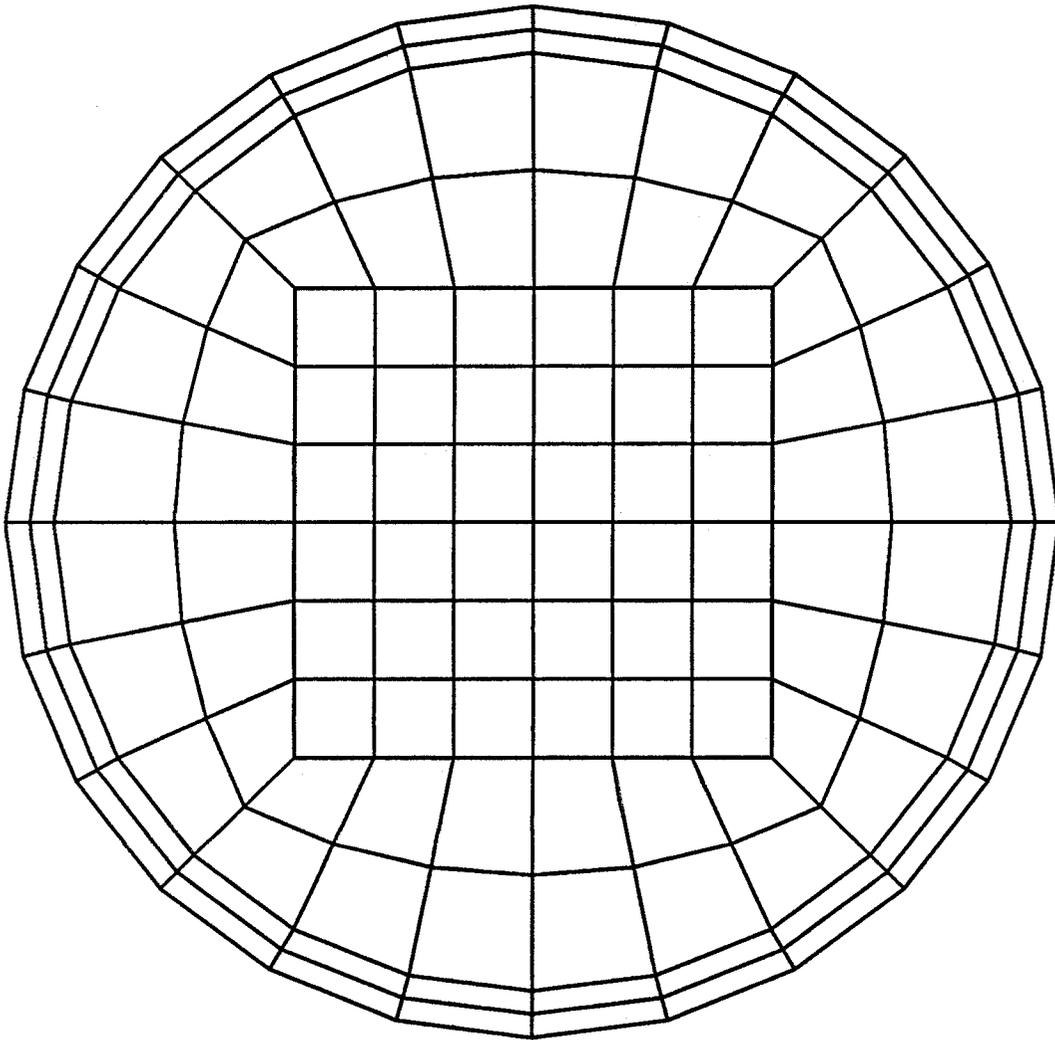
mt 2 2 0 5 5 0 1

end

end

set tv display

circular plate with outer liner
INGRID display



circular plate with hole - solution #1

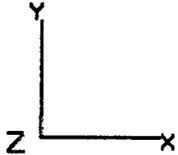
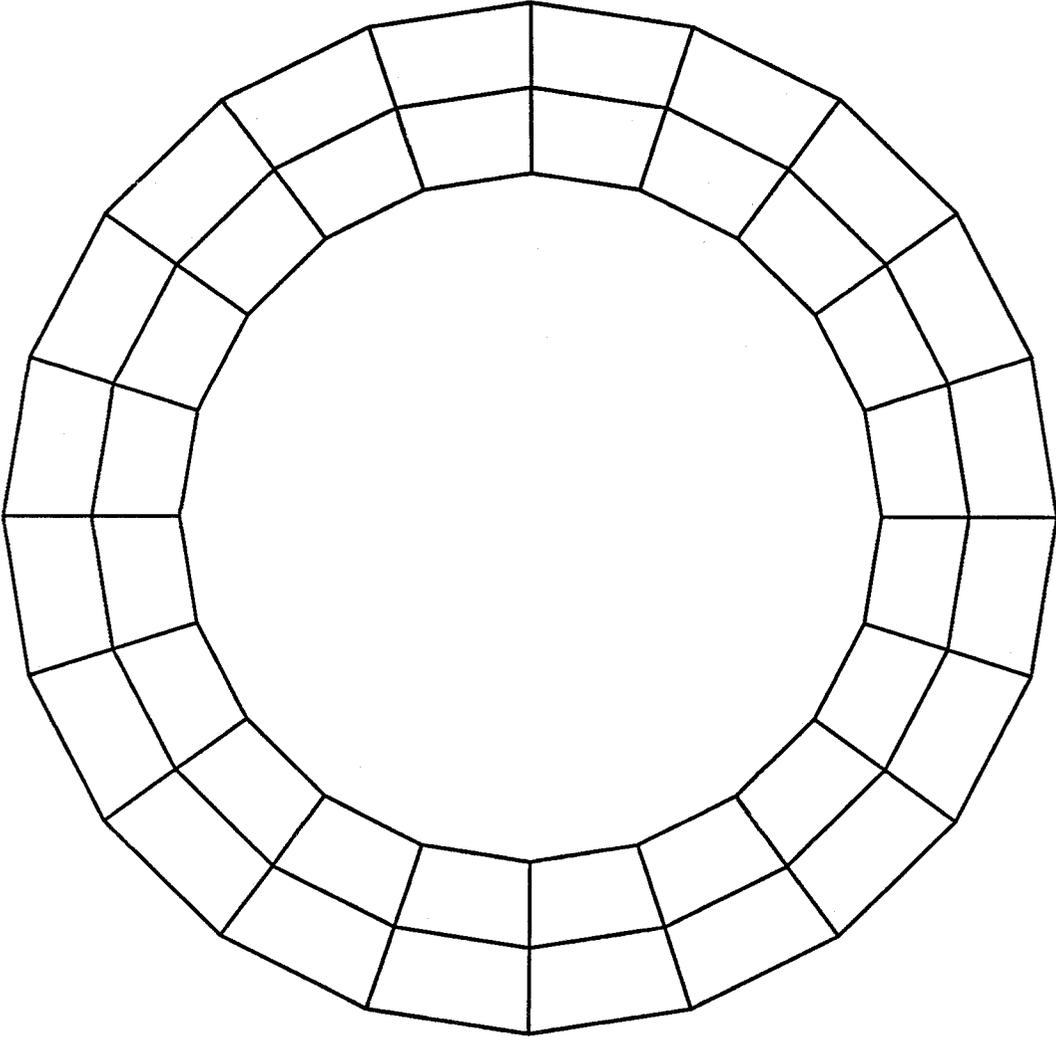
```
dn3d
start
1 3; 1 21; -1;
1 1.5 0 360 0
c Interpret the coordinates as cylindrical.
cyli
end
end

set tv display
```

circular plate with hole - solution #2

```
dn3d
start
1 3 9 11; 1 3 9 11; -1;
-1 -1 1 1
-1 -1 1 1
0
c Delete the corner regions.
di 1 2 0 3 4; 1 2 0 3 4; -1;
c Project the outer boundary to the outer radius circle.
a 1 1 0 4 4 0 3 1.5
c Delete the interior region.
d 2 2 0 3 3 0
c Project the inner boundary to the inner radius circle.
a 2 2 0 3 3 0 3 1
end
end
```

circular plate with hole - solution #1
INGRID display



hollow cylinder - solution #1

```
dn3d
start
-1; 1 19; 1 5;
1.5 0 360 0 2
c Interpret the coordinates as cylindrical.
cyli
end
end

ry -20
rx 35
set tv display
```

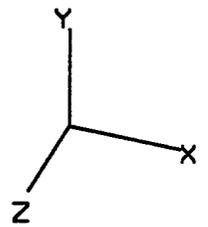
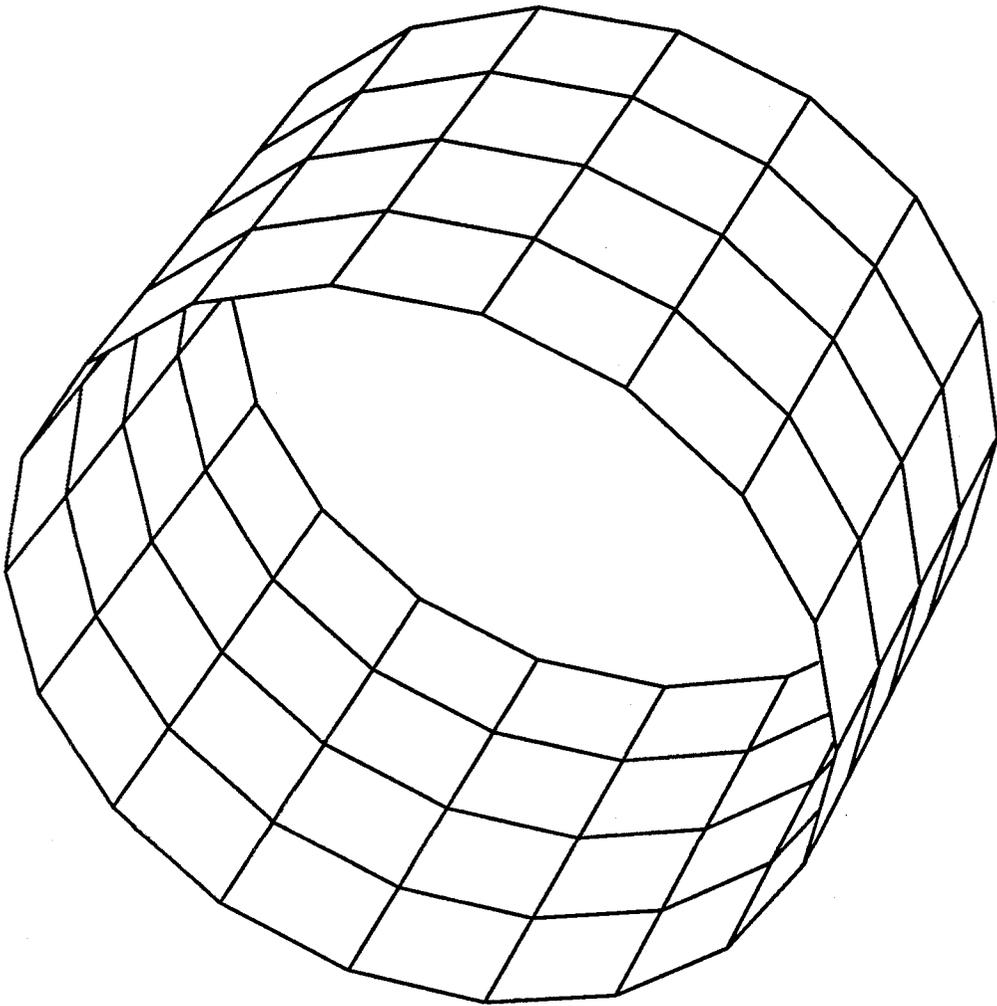
hollow cylinder - solution #2

```
dn3d
start
-1 -5; -1 -5; 1 5;
-1 1 -1 1 0 2
c Project the sides of a square tube out to a cylinder.
a 1 1 0 2 2 0 3 1.5
end
end
```

hollow cylinder - solution #3

```
dn3d
start
-1 -5; -1 -5; 1 5;
-1 1 -1 1 0 2
c Project the sides of a square tube out to a cylinder.
sf 1 1 0 2 2 0 cy 0 0 0 0 0 1 1.5
end
end
```

hollow cylinder - solution #1
INGRID display



cylinder with stiffeners

dn3d

start

1 -3; 1 19; -1 -5;

1.2 1.5 0 360 0 2

c Interpret the coordinates as cylindrical.

cyli

end

end

ry -20

rx 35

set tv display

cylinder with stiffeners
INGRID display

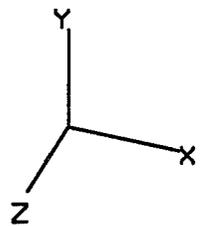
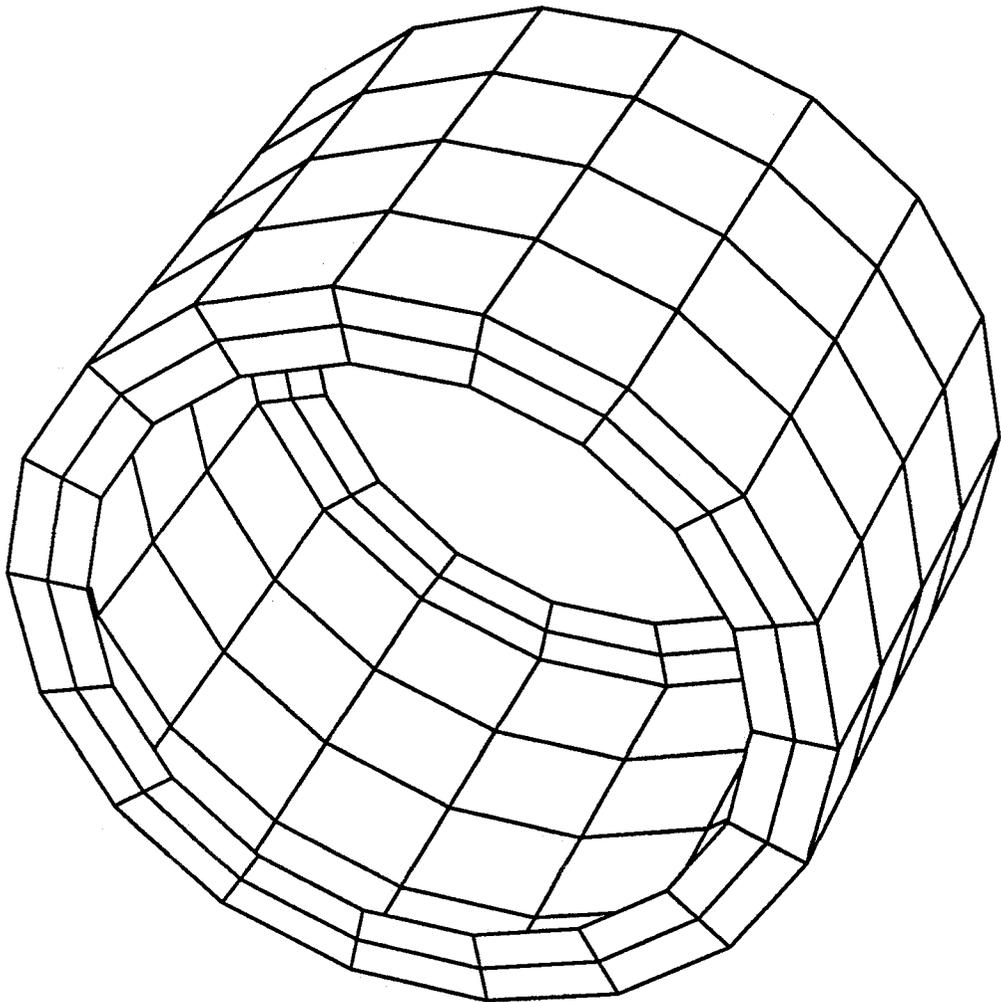
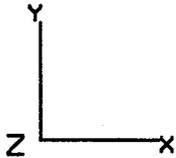
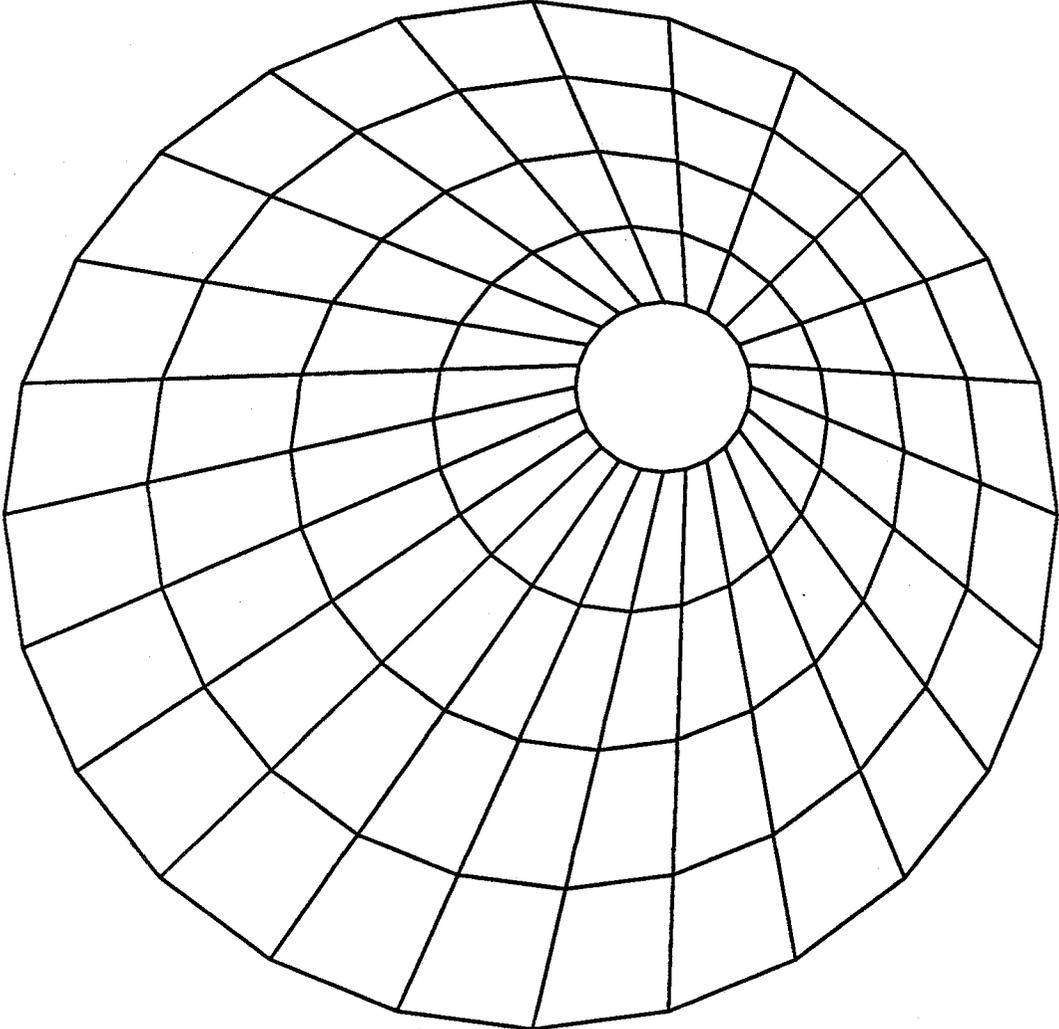


plate with hole in it

```
dn3d
start
1 5 11 15; 1 5 11 15; -1;
-1 -1 1 1
-1 -1 1 1
0
c Delete corner regions.
di 1 2 0 3 4; 1 2 0 3 4; -1;
c Project outer boundary out to circle.
a 1 1 0 4 4 0 3 3
c Delete interior.
d 2 2 0 3 3 0
c Move the nodes on the inner boundary over to where the
c hole will go.
mb 2 2 0 3 3 0 12 .75 .75
c Project inner boundary to inner circle.
a 2 2 0 3 3 0 3 .5
end
end

set tv display
```

plate with hole in it
INGRID display



square plate with hole

dn3d

start

1 5 9 13; 1 5 9 13; -1;

0 1 2 3

0 1 2 3

0

c Delete center region.

d 2 2 0 3 3 0

c Project boundary of center region to circle with radius 0.75.

a 2 2 0 3 3 0 3 .75

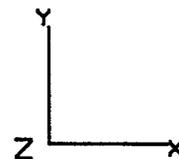
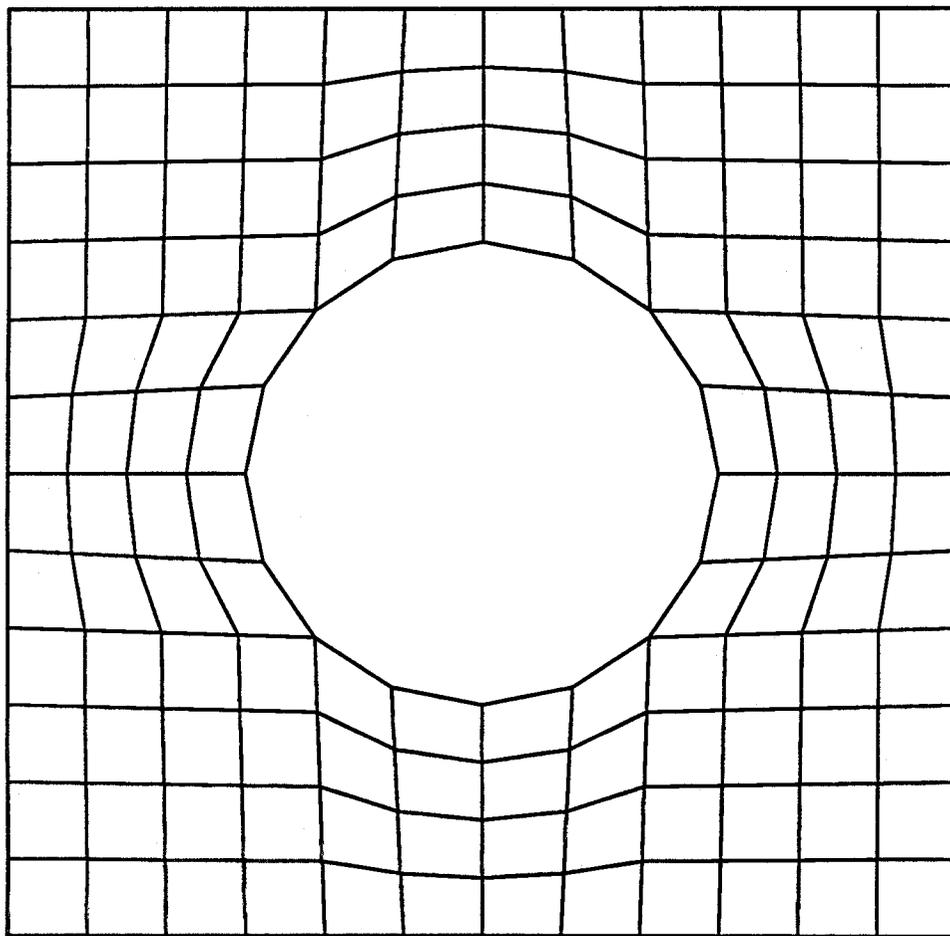
end

end

scale 0.9

set tv display

square plate with hole
INGRID display



square plate with hole

dn3d

start

1 5 9 13; 1 5 9 13; -1;

-2 -2 2 2

-2 -2 2 2

0

c Delete corner regions.

di 1 2 0 3 4; 1 2 0 3 4; -1;

c Delete center region.

d 2 2 0 3 3 0

c Project boundary of center region to circle.

a 2 2 0 3 3 0 3 1.1

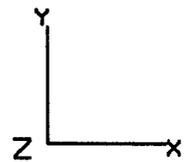
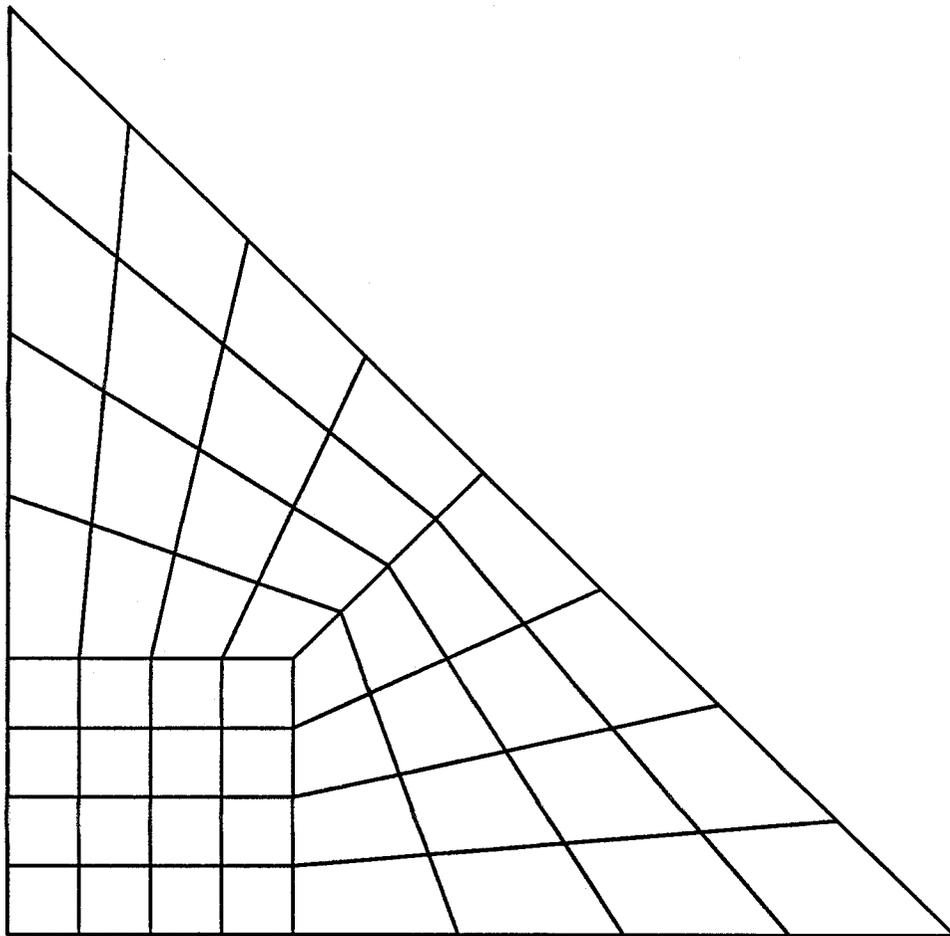
end

end

scale 0.9

set tv display

right triangle
INGRID display



right triangle

dn3d

start

1 5 9; 1 5 9; -1;

0 .3 1 0 .3 1 0

c Delete upper right corner region.

d 2 2 0 3 3 0

c Move the upper right corners of the top and right regions

c to the midpoint of the triangle's base. The rest of the

c nodes in the regions follow accordingly.

pa 2 3 0 12 .5 .5

pa 3 2 0 12 .5 .5

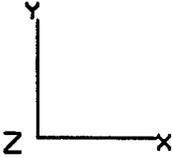
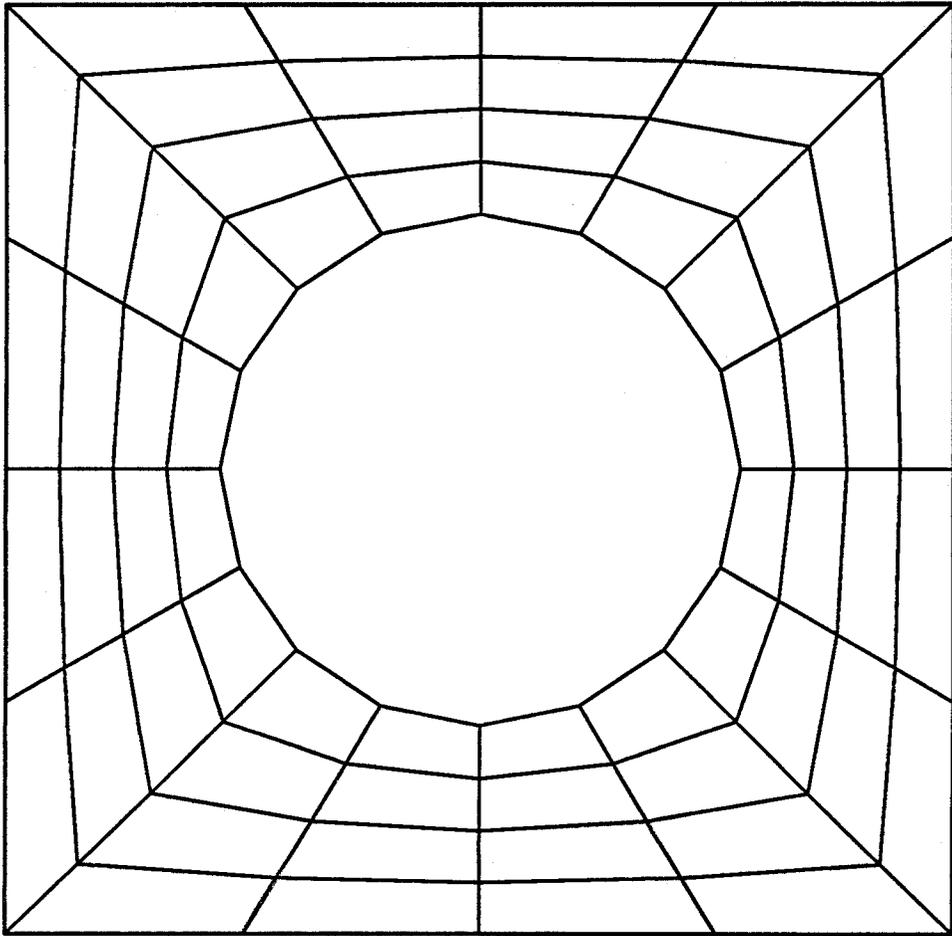
end

end

scale 0.9

set tv display

square plate with hole
INGRID display



solid sphere

dn3d

start

c Set up the indices and give the corresponding coordinates.

1 3 5 7 9; 1 3 5 7 9; 1 3 5 7 9;

-2.5 -2.5 0 2.5 2.5

-2.5 -2.5 0 2.5 2.5

-2.5 -2.5 0 2.5 2.5

c Delete the regions that lie along the edges of the cube.

di 1 2 0 4 5; 1 2 0 4 5; ;

di 1 2 0 4 5; ; 1 2 0 4 5;

di ; 1 2 0 4 5; 1 2 0 4 5;

c Project the boundary to a spherical surface.

sfi -1 -5; -1 -5; -1 -5; sp 0 0 0 5

c Delete half of the sphere.

d 1 0 0 3 0 0

end

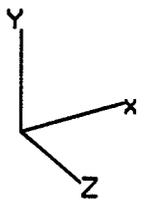
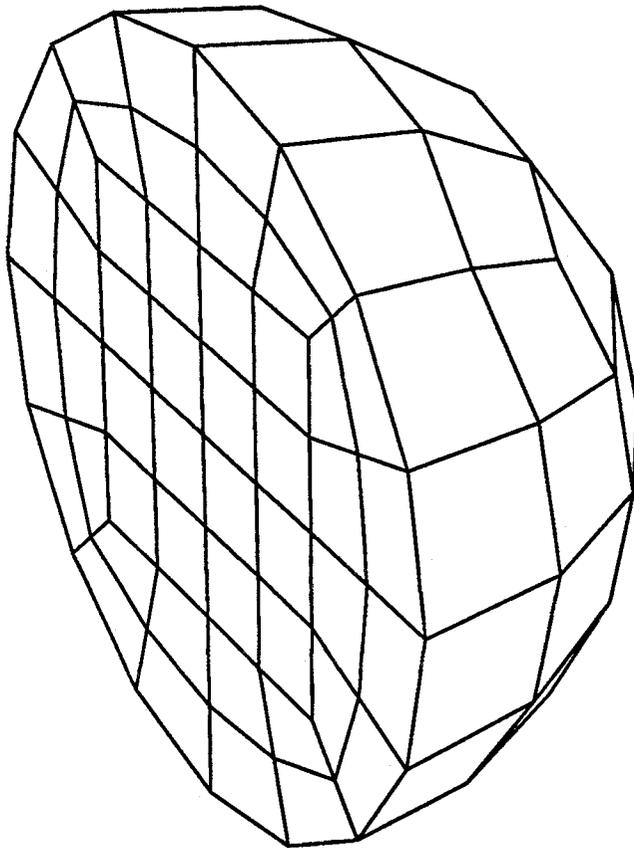
end

ry 30

rx 30

set tv display

solid sphere
INGRID display



spherical shell (radius 5.0) - solution #1

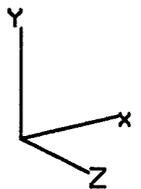
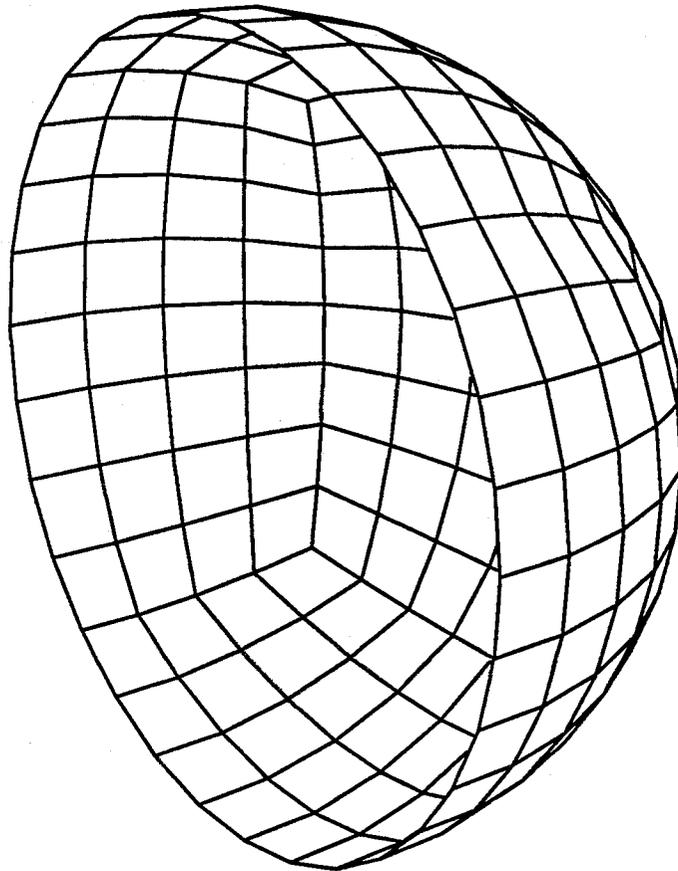
```
dn3d
start
-1 5 -9; -1 5 -9; -1 5 -9;
-1 0 1
-1 0 1
-1 0 1
c Project to a spherical surface.
sfi -1 -3; -1 -3; -1 -3; sp 0 0 0 5
c Delete half of the sphere.
d 1 0 0 2 0 0
end
end

ry 35
rx 20
set tv display
```

spherical shell (radius 5.0) - solution #2

```
dn3d
start
1 5 13 17; 1 5 13 17; -1;
-5 -5 5 5
-5 -5 5 5
0
c Move the nodes on the boundary of the interior region
c in the z direction.
pb 2 2 0 3 3 0 z 5
c Delete the four corners of the sheet.
di 1 2 0 3 4; 1 2 0 3 4; ;
c Project to a spherical surface.
sf 0 0 1 0 0 1 sp 0 0 0 5
end
end
```

spherical shell (radius 5.0) - solution #1
INGRID display



thick spherical shell (r1 = 3.0, r2 = 5.0) - solution #1

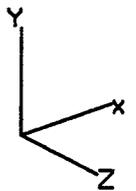
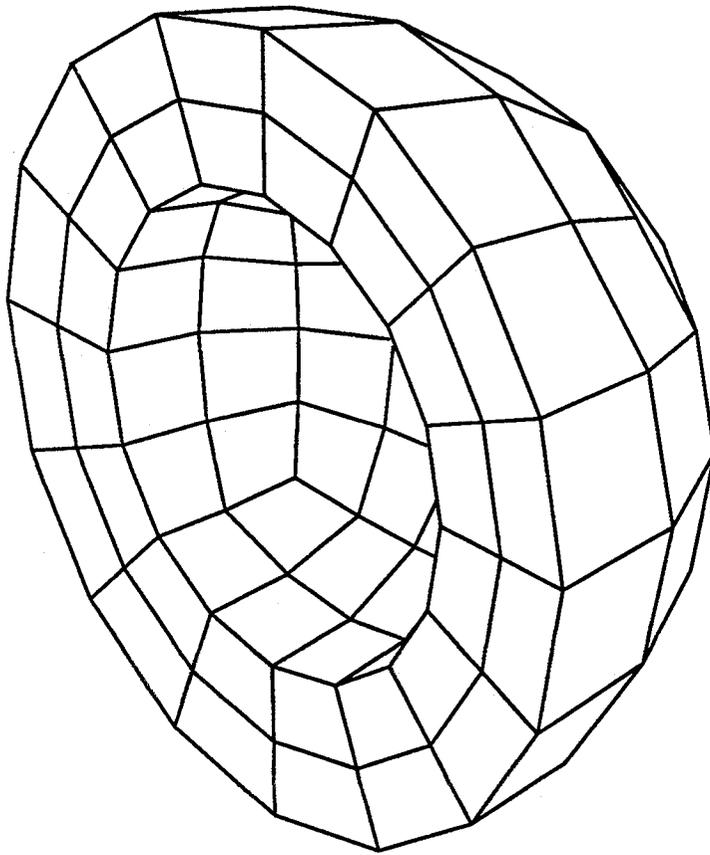
```
dn3d
start
1 3 5 7 9; 1 3 5 7 9; 1 3 5 7 9;
-1 -1 0 1 1
-1 -1 0 1 1
-1 -1 0 1 1
di 1 2 0 4 5; 1 2 0 4 5; ;
di 1 2 0 4 5; ; 1 2 0 4 5;
di ; 1 2 0 4 5; 1 2 0 4 5;
sfi -1 -5; -1 -5; -1 -5; sp 0 0 0 5
sfi -2 -4; -2 -4; -2 -4; sp 0 0 0 3
c Delete interior region (hollow out the sphere).
d 2 2 2 4 4 4
c Delete half of the sphere.
d 1 0 0 3 0 0
end
end

ry 40
rx 25
set tv display
```

thick spherical shell (r1 = 3.0, r2 = 5.0) - solution #2

```
dn3d
start
1 3 7 9; 1 3 7 9; 1 3;
-1 -1 1 1
-1 -1 1 1
0 0
pb 2 2 0 3 3 0 z 1
di 1 2 0 3 4; 1 2 0 3 4; ;
sf 0 0 1 0 0 1 sp 0 0 0 3
sf 0 0 2 0 0 2 sp 0 0 0 5
end
end
```

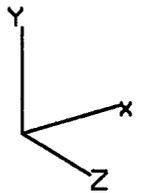
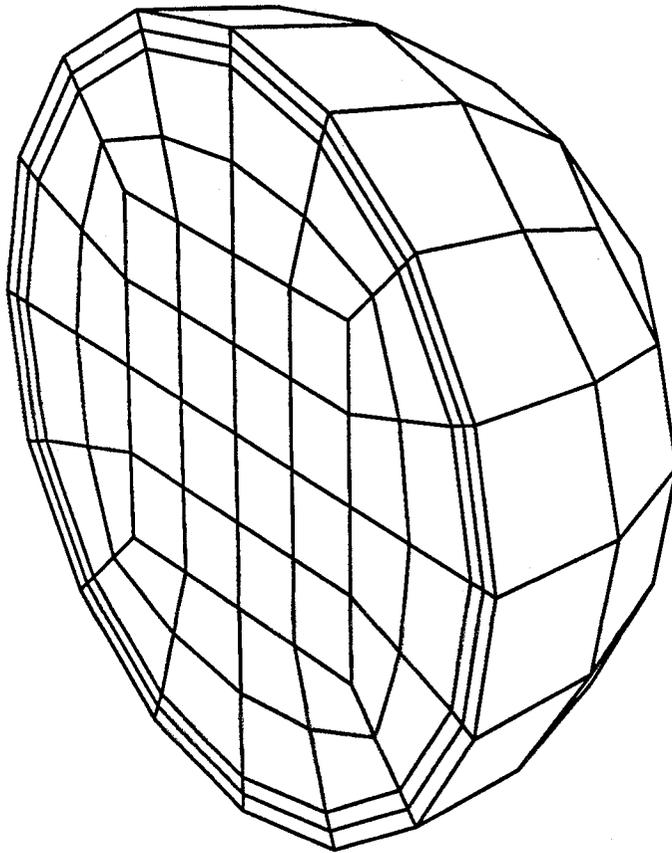
thick spherical shell (r1 = 3.0, r2 = 5.0) - solution #1
INGRID display



```
solid sphere with liner ( r1 = 5.0, r2 = 5.5 )
dn3d
start
1 3 5 7 9 11 13; 1 3 5 7 9 11 13; 1 3 5 7 9 11 13;
-2.5 -2.5 -2.5 0 2.5 2.5 2.5
-2.5 -2.5 -2.5 0 2.5 2.5 2.5
-2.5 -2.5 -2.5 0 2.5 2.5 2.5
di 1 3 0 5 7; 1 3 0 5 7; ;
di 1 3 0 5 7; ; 1 3 0 5 7;
di ; 1 3 0 5 7; 1 3 0 5 7;
sfi -2 -6; -2 -6; -2 -6; sp 0 0 0 5
sfi -1 -7; -1 -7; -1 -7; sp 0 0 0 5.5
d 1 0 0 4 0 0
end
end

ry 35
rx 25
set tv display
```

solid sphere with liner (r1 = 5.0, r2 = 5.5)
INGRID display

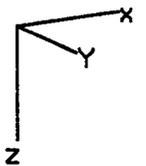
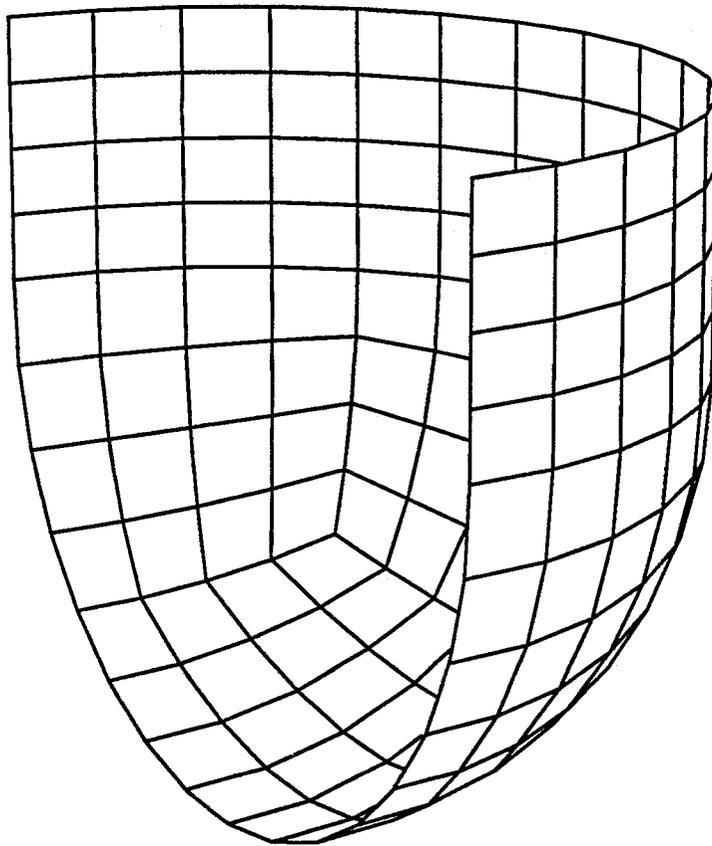


cylinder with spherical cap

```
dn3d
start
-1 5 -9; -1 5 -9; 1 5 -9;
-1 0 1 -1 0 1 -3 0 1
c Project the cylindrical region.
sfi -1 -3; -1 -3; 1 2; cy 0 0 0 0 0 1 5
c Project the spherical region.
sfi -1 -3; -1 -3; 2 -3; sp 0 0 0 5
c Delete one half of the model.
d 1 0 0 2 0 0
end
end

rx 90
ry 30
rx 15
set tv display
```

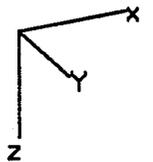
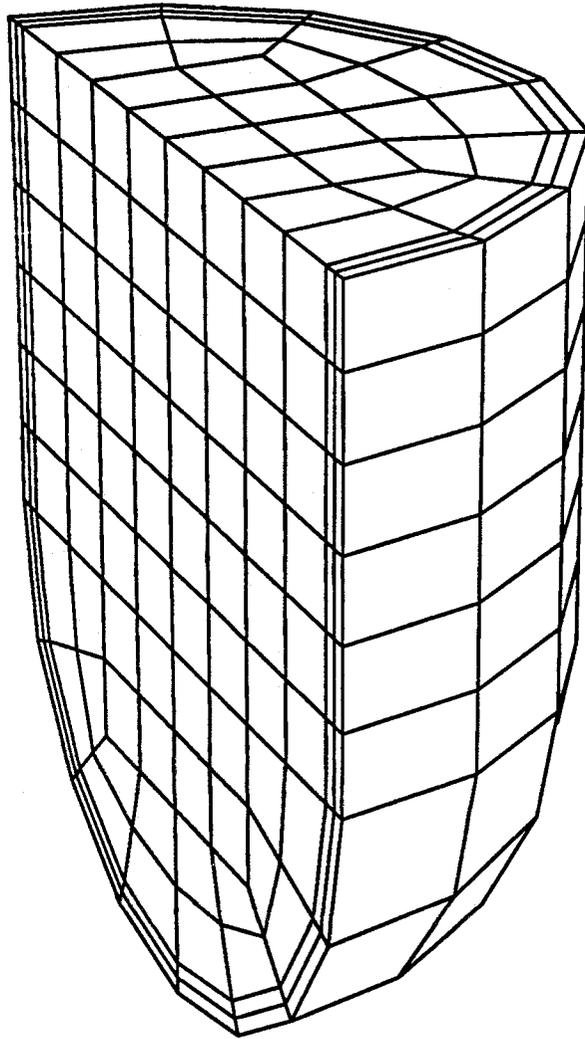
cylinder with spherical cap
INGRID display



```
solid cylinder with spherical end and liner (r1 = 5.0, r2 = 5.5)
dn3d
start
1 3 5 7 9 11 13;
1 3 5 7 9 11 13;
1 7 9 11 13;
-2.5 -2.5 -2.5 0 2.5 2.5 2.5
-2.5 -2.5 -2.5 0 2.5 2.5 2.5
-7.5 0 2.5 2.5 2.5
c Delete the unneeded regions.
di 1 3 0 5 7; 1 3 0 5 7; ;
di 1 3 0 5 7; ; 3 5;
di ; 1 3 0 5 7; 3 5;
c Inner and outer spheres.
sfi -2 -6; -2 -6; 2 -4; sp 0 0 0 5
sfi -1 -7; -1 -7; 2 -5; sp 0 0 0 5.5
c Inner and outer cylinders.
sfi -2 -6; -2 -6; 1 2; cy 0 0 0 0 0 1 5
sfi -1 -7; -1 -7; 1 2; cy 0 0 0 0 0 1 5.5
d 1 0 0 4 0 0
end
end

rx 90
ry 25
rx 25
set tv display
```

solid cylinder with spherical end and liner (r1 = 5.0, r2 = 5.5)
INGRID display

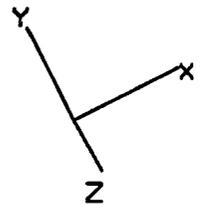
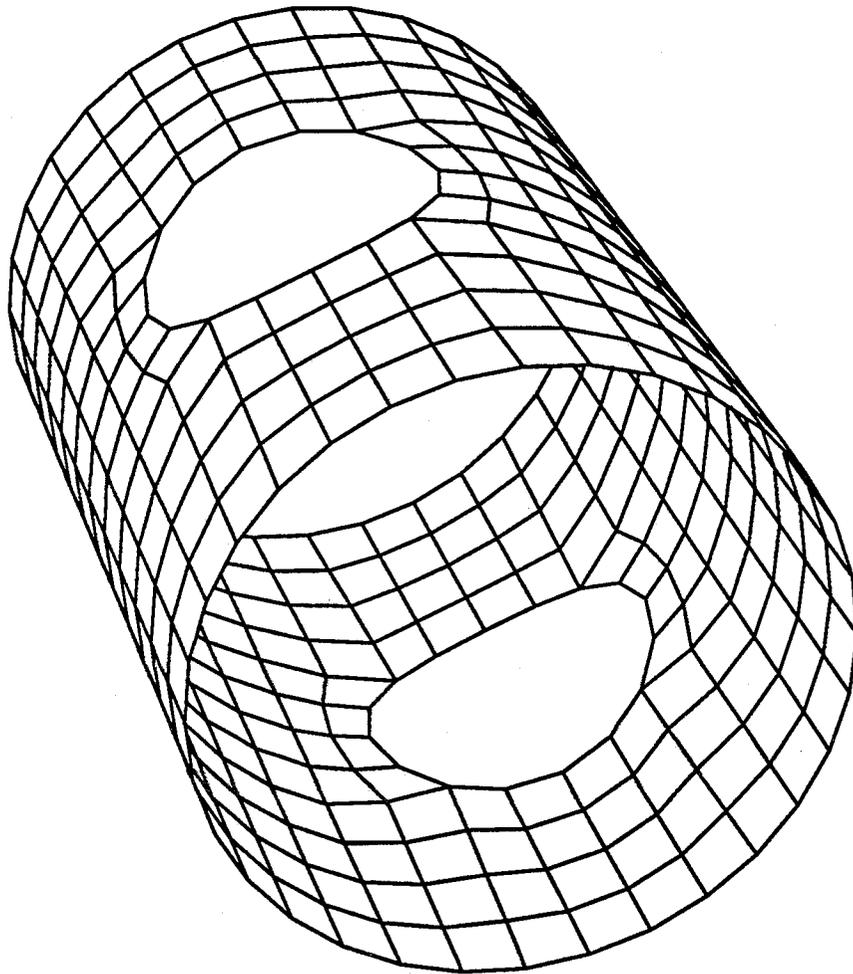


cylindrical shell with hole

```
dn3d
start
-1 3 7 -9; -1 -9; 1 5 9 13;
-2.0 -1.0 1.0 2.0 -2.0 2.0 -3.0 -1.0 1.0 3.0
c Cut out the two holes.
d 2 0 2 3 0 3
c Project the square tube to a cylinder.
sfi -1 -4; -1 -2; ; cy 0 0 0 0 0 1 2.0
c Project the borders of the cut-out areas onto another cylinder.
sfi -2 -3; ; -2 -3; cy 0 0 0 0 1 0 1.0
c Evenly distribute the nodes at the ends of the cylinder to get a
c better mesh.
cpl 1 0 1 4 0 1 1 + k 3 1
end
end

rz 30
rx 25
ry 15
set tv display
```

cylindrical shell with hole
INGRID display



cylindrical shell with hole

dn3d

c Part 1 - cylinder.

start

-1 -7; -1 -7; 1 4 10 13;

-1.0 1.0

-1.0 1.0

-3.0 -2.0 2.0 3.0

di ; -1 -2; 2 3;

sfi -1 -2; -1 -2; ; cy 0 0 0 0 0 1 2.0

end

c Part 2 - transition region.

start

1 7 13 19; -1 -2; 1 7 13 19;

-1.0 -1.0 1.0 1.0

-1.0 1.0

-2.0 -2.0 2.0 2.0

c Delete corners.

di 1 2 0 3 4; -1 -2; 1 2 0 3 4;

c Project onto large cylinder.

sfi ; -1 -2; ; cy 0 0 0 0 0 1 2.0

c Delete hole.

d 2 0 2 3 0 3

c Project hole border onto small cylinder.

sfi -2 -3; ; -2 -3; cy 0 0 0 0 1 0 0.75

end

end

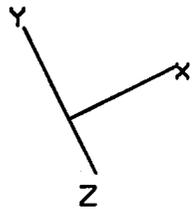
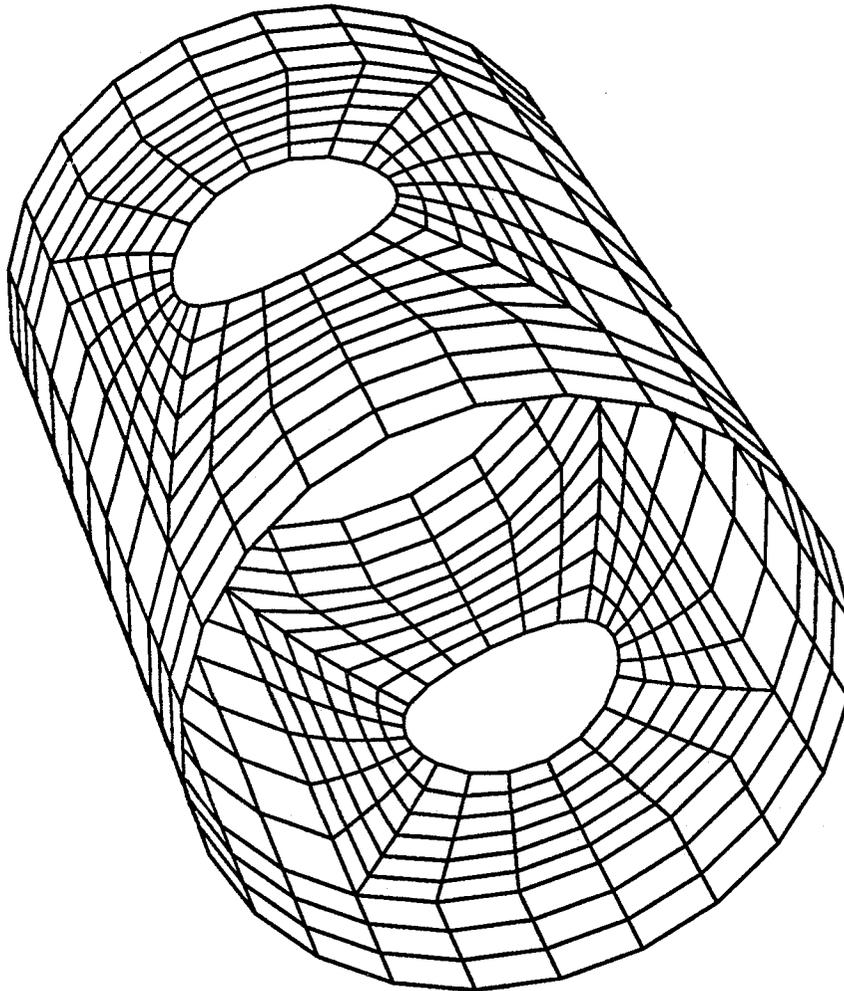
rz 30

rx 27

ry 15

set tv display

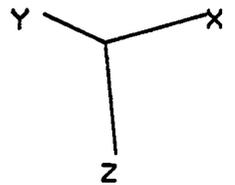
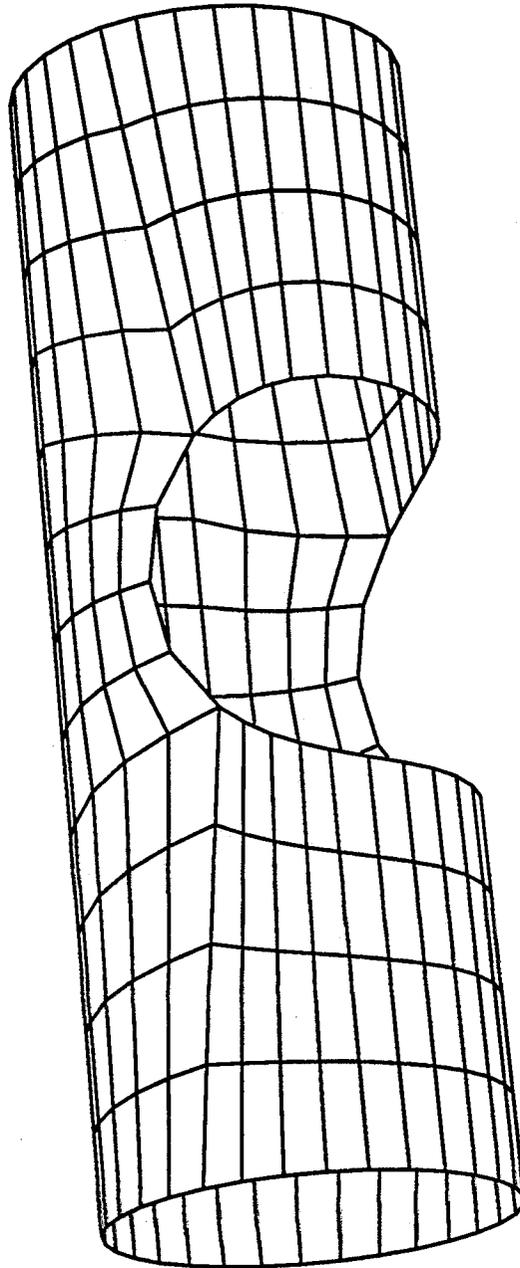
cylindrical shell with hole
INGRID display



```
cylinder with cylindrical cut out
dn3d
start
-1 5 -9; -1 5 -9; 1 5 9 13;
-2.0 0.0 2.0
-2.0 0.0 2.0
-6.0 -2.0 2.0 6.0
d 2 0 2 3 0 3
sfi -1 -3; -1 -3; ; cy 0 0 0 0 0 1 2.0
sfi -2 3; ; -2 -3; cy 2 0 0 0 1 0 2.0
end
end

rz 45
rx 70
ry 15
set tv display
```

cylinder with cylindrical cut out
INGRID display

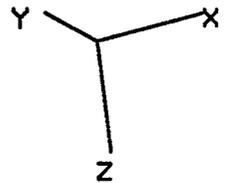
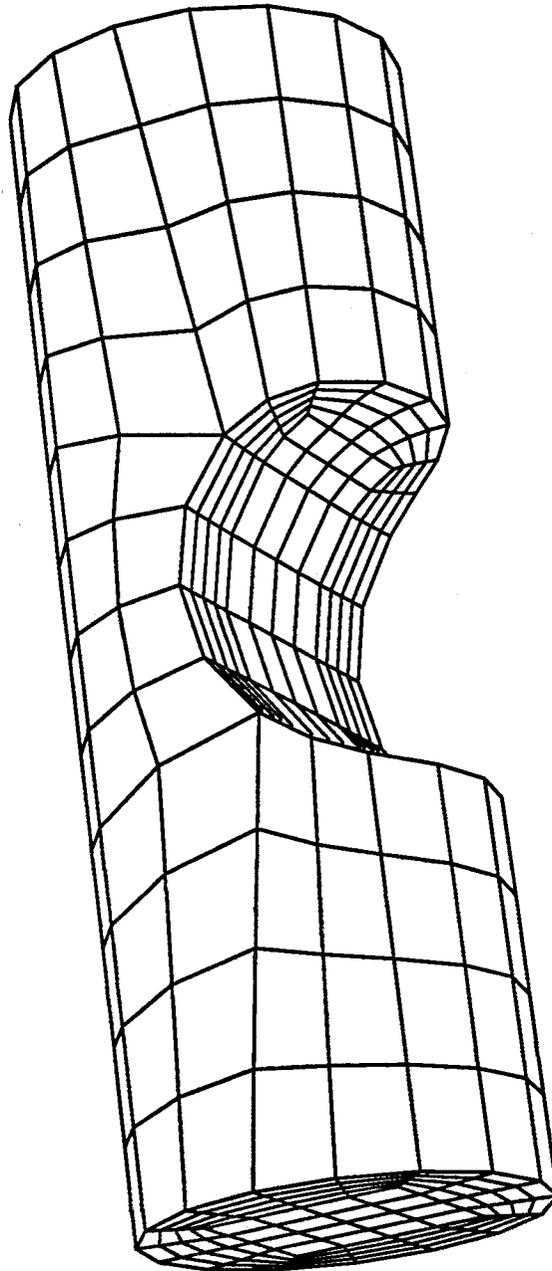


solid cylindrical bar with cylindrical cut out

```
dn3d
start
1 5 7 9 13;
1 5 7 9 13;
1 5 9 13;
-1 -1 0 1 1
-1 -1 0 1 1
-6 -2 2 6
di 1 2 0 4 5; 1 2 0 4 5; ;
d 3 0 2 5 0 3
sfi -1 -5; -1 -5; ; cy 0 0 0 0 0 1 2.0
sfi -3 5; ; -2 -3; cy 2 0 0 0 1 0 2.0
end
end

rz 45
rx 70
ry 20
set tv display
```

solid cylindrical bar with cylindrical cut out
INGRID display

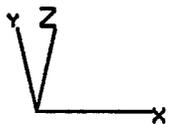
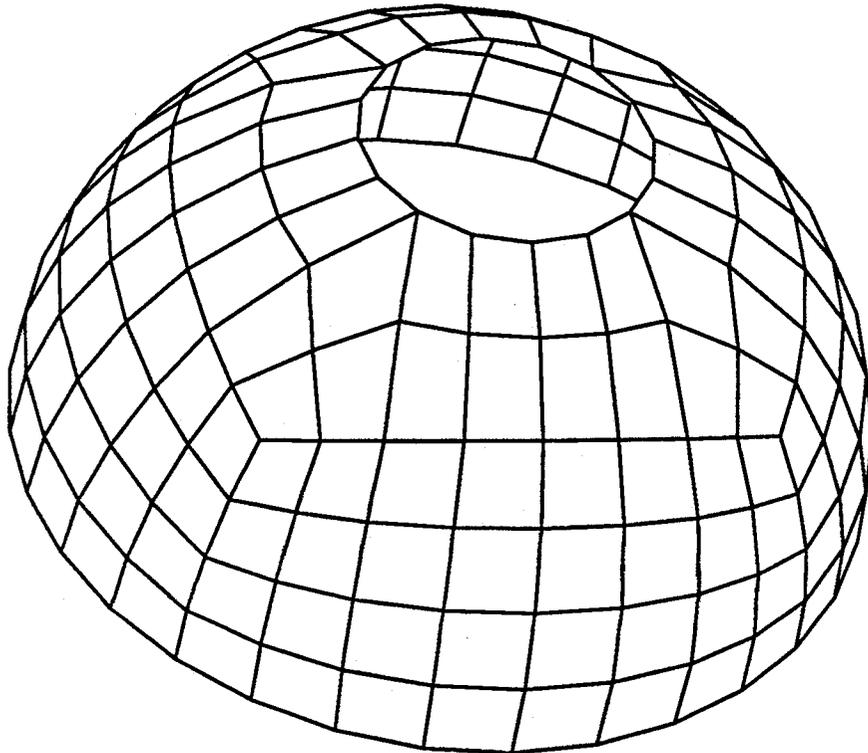


spherical cap with a hole

```
dn3d
start
-1 3 7 -9; -1 3 7 -9; 1 -5;
-2 -1 1 2
-2 -1 1 2
0 2
sfi -1 -4; -1 -4; 1 -2; sp 0 0 0 3.0
d 2 2 0 3 3 0
sfi -2 -3; -2 -3; ; cy 0 0 0 0 0 1 1.0
c Evenly space the nodes around the rim.
cpl 0 0 1 0 0 1 1 + sij j
end
end

rx -45
ry 13
set tv display
```

spherical cap with a hole
INGRID display



spherical cap with hole and transition region

dn3d

start

-1 -7; -1 -7; 1 4;

-2.0 2.0

-2.0 2.0

0.0 2.0

sfi -1 -2; -1 -2; ; sp 0 0 0 2.0

end

start

1 7 13 19; 1 7 13 19; -1;

-2.0 -2.0 2.0 2.0

-2.0 -2.0 2.0 2.0

2.0

di 1 2 0 3 4; 1 2 0 3 4; -1;

sf 0 0 1 0 0 1 sp 0 0 0 2.0

d 2 2 0 3 3 0

sf 2 2 0 3 3 0 cy 0 0 0 0 0 1 1.0

end

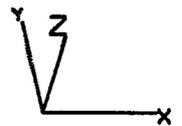
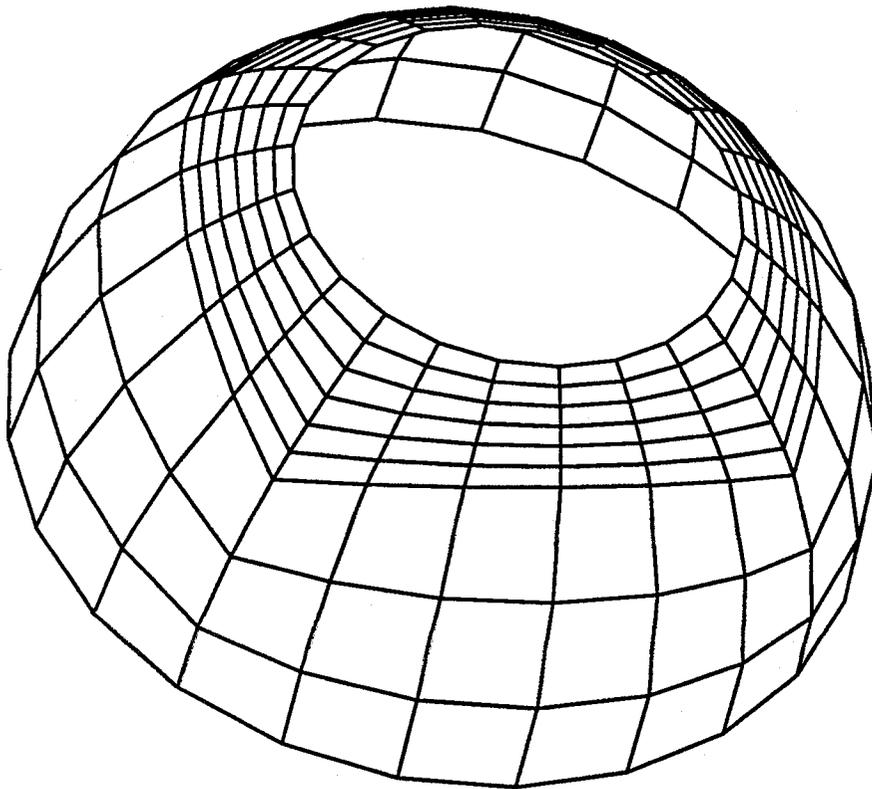
end

rx -40

ry 15

set tv display

spherical cap with hole and transition region
INGRID display

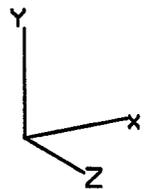
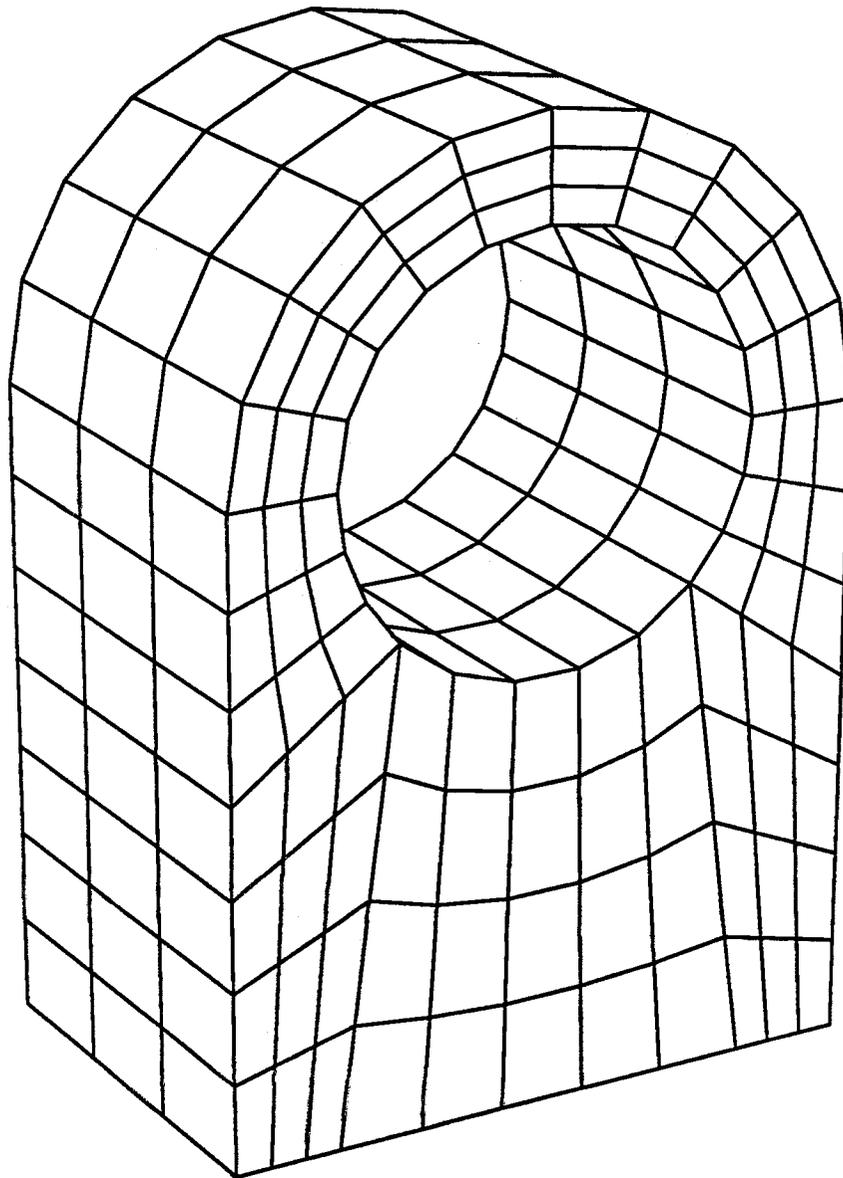


bracket with hole

```
dn3d
start
1 4 9 12;
1 5 8 13;
1 4;
-1.5 -1.0 1.0 1.5
0.0 2.0 3.0 3.0
0.0 2.0
d 2 2 0 3 4 0
pb 0 4 0 0 4 0 xy 0 5
sfi -2 -3; -2 4; ; cy 0 3 0 0 0 1 1.0
sfi -1 -4; 3 4; ; cy 0 3 0 0 0 1 1.5
sf 1 1 0 1 3 0 plan -1.5 0 0 1 0 0
sf 4 1 0 4 3 0 plan 1.5 0 0 1 0 0
cpl 1 1 0 1 3 0 j + i 3 j
end
end

ry 30
rx 20
set tv display
```

bracket with hole
INGRID display



circular flange with hole

dn3d

start

1 5 9 13;

1 3 7 9;

1 4;

-1.0 -1.0 1.0 1.0

10.0 10.0 12.0 12.0

0.0 2.0

d 2 2 0 3 3 0

sfi -2 -3; -2 -3; ; cy 0 11.0 0 0 0 1 1.0

sf 0 1 0 0 1 0 cy 0 0 0 0 0 1 9.0

sf 0 4 0 0 4 0 cy 0 0 0 0 0 1 13.0

sf 1 0 0 1 0 0 pl3 rt 0 0 0 cy 1 110 0 rt 0 0 1 0.0

sf 4 0 0 4 0 0 pl3 rt 0 0 0 cy 1 70 0 rt 0 0 1 0.0

cpl 1 0 0 1 0 0 j + i 3 j

end

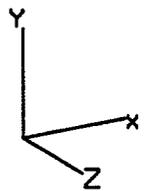
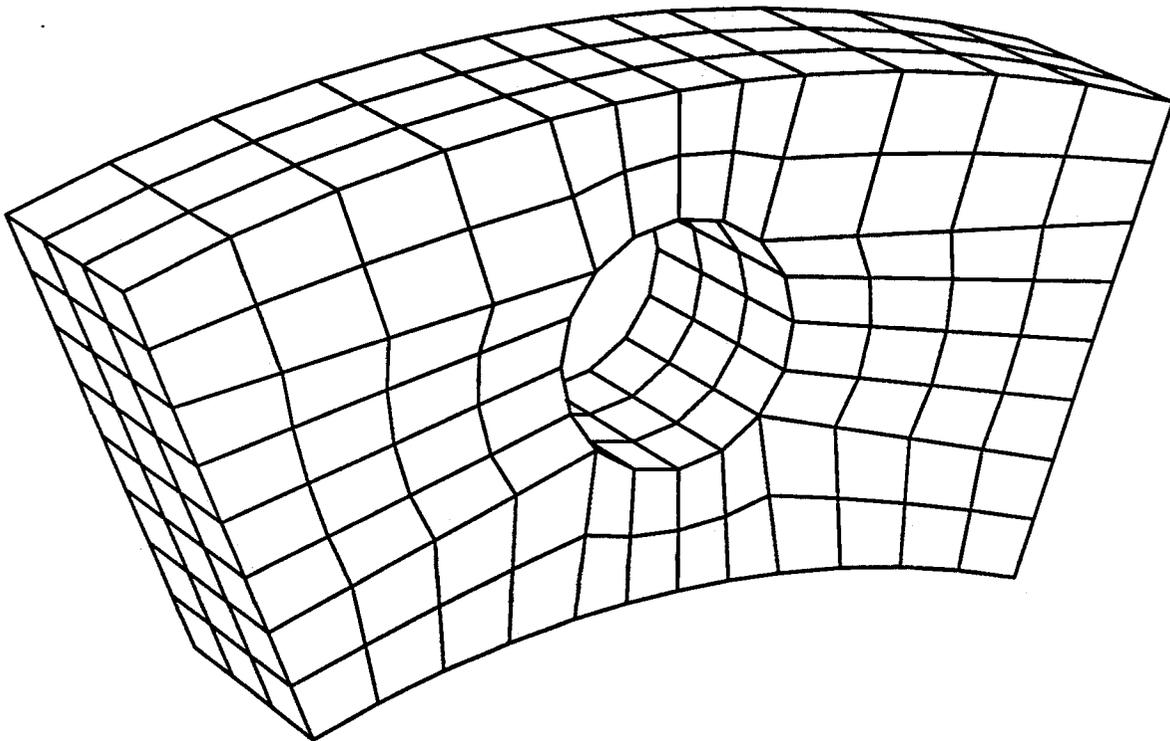
end

ry 30

rx 20

set tv display

circular flange with hole
INGRID display



pipe bend with hole

dn3d

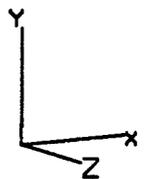
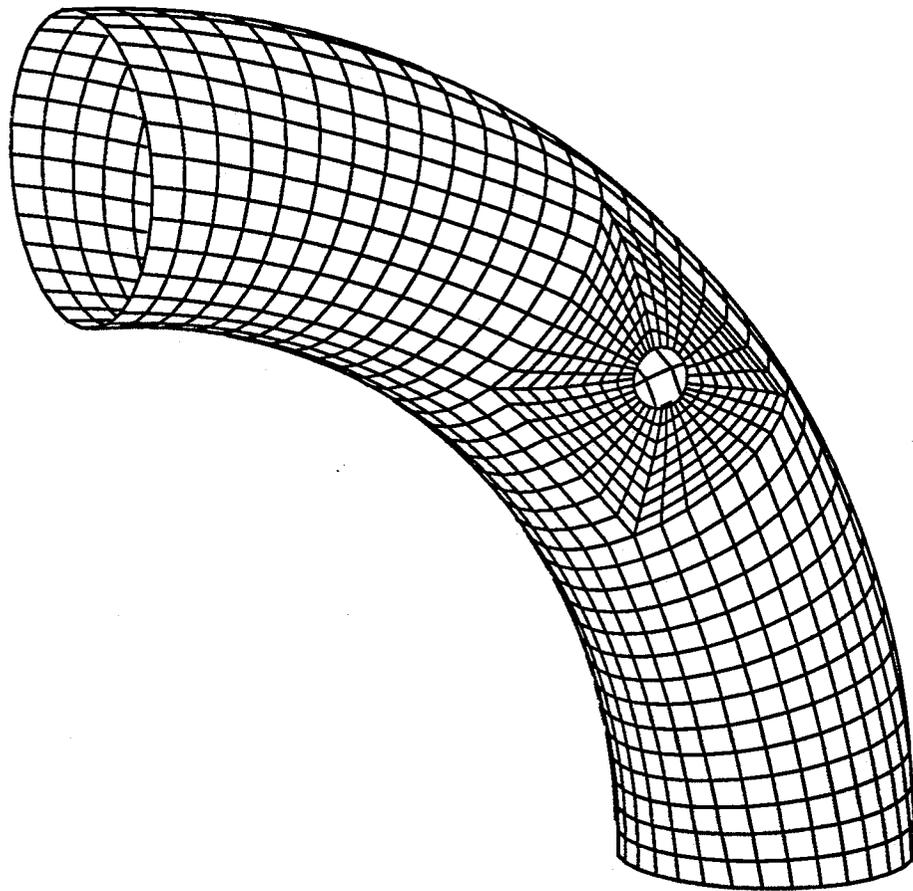
```
start
-1 -9;
1 15 23 37;
-1 -9;
10.0 15.0
0.0 0.0 0.0 0.0
-2.5 2.5
rr 0 2 0 0 2 0 rz 35;
rr 0 3 0 0 3 0 rz 55;
rr 0 4 0 0 4 0 rz 90;
sfi -1 -2; ; -1 -2; ts 0 0 0 0 0 1 12.5 0.0 2.5
d 1 2 2 2 3 2
end
```

```
start
1 9 17 25;
1 9 17 25;
-1;
10.0 10.0 15.0 15.0
0 0 0 0
2.5
mb 2 2 0 2 3 0 x 1.0
mb 3 2 0 3 3 0 x -1.0
rr 0 1 0 0 2 0 rz 35;
rr 0 3 0 0 4 0 rz 55;
di 1 2 0 3 4; 1 2 0 3 4; ;
d 2 2 0 3 3 0
sf 0 0 1 0 0 1 ts 0 0 0 0 0 1 12.5 0.0 2.5
sf 2 2 0 3 3 0 cy 8.8375 8.8375 0 0 0 1 0.5
end
```

end

```
ry 30
rx 10
set tv display
```

pipe bend with hole
INGRID display



two pipes intersecting at 45 degrees

dn3d

c Define two cylindrical surfaces.

sd 1 cy 0 0 0 0 0 1 2.0

sd 2 cy 0 0 0 0 1 1 2.0

start

-1 -10; -1 -10; 1 7 16 22;

-2.0 2.0

-2.0 2.0

-6.0 -2.0 2.0 6.0

d 0 2 2 0 2 3

mb 0 2 2 0 2 3 z 2.0

sfi -1 -2; -1 -2; ; sd 1

sf 0 2 2 0 2 3 sd 2

end

start

-1 -10; 1 15; -1 -10;

-2.0 2.0

2.0 12.0

-2.0 2.0

rr 0 2 0 0 2 0 rx 45;

mb 0 1 0 0 1 0 z 2

sfi -1 -2; ; -1 -2; sd 2

sf 0 1 0 0 1 0 sd 1

end

end

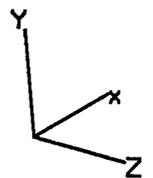
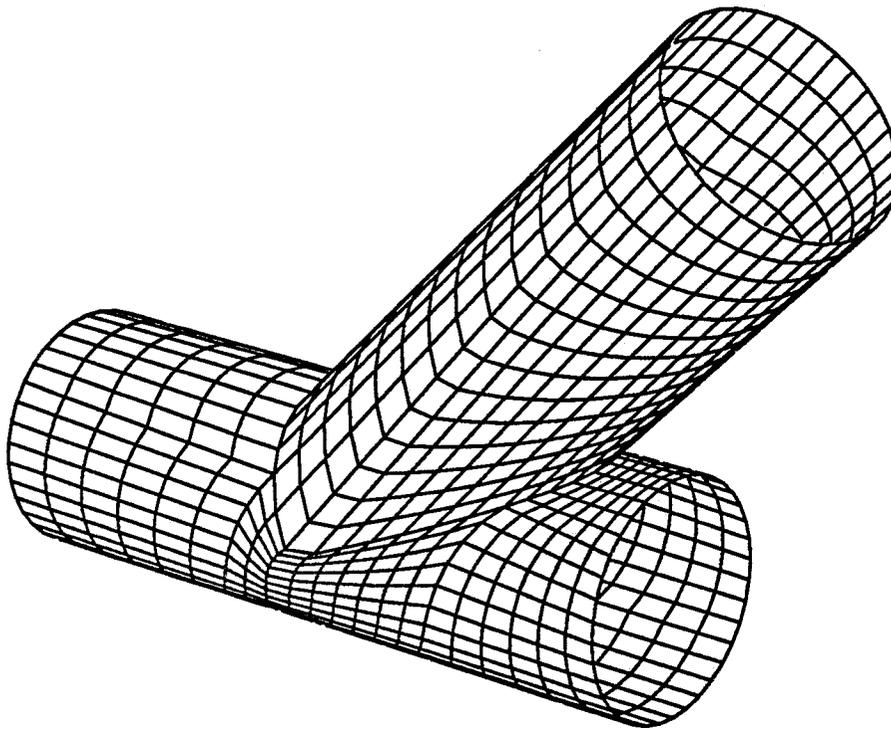
ry 60

rx 25

ry -10

set tv display

two pipes intersecting at 45 degrees
INGRID display



three pipes intersecting

dn3d

c Define three cylindrical surfaces.

```
sd 1 cy 0 0 0 0 1 2.0
sd 2 cy 0 0 0 0 0.866 0.5 2.0
sd 3 cy 0 0 0 0 0.866 -0.5 2.0
```

start

```
-1 -11; -1 -11; 1 6 11 16 21;
-2.0 2.0
-2.0 2.0
-6.0 -2.0 0.0 2.0 6.0
d 0 2 2 0 2 4
pb 0 2 3 0 2 3 y 0
sf 1 2 2 1 2 3 sd 3 + i 1 sd 3
sf 1 2 2 2 2 2 sd 3
sf 1 2 3 1 2 4 sd 2 + i 1 sd 2
sf 1 2 4 2 2 4 sd 2
sfi -1 -2; -1 -2; ; sd 1
end
```

start

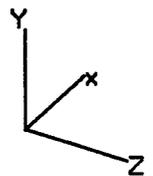
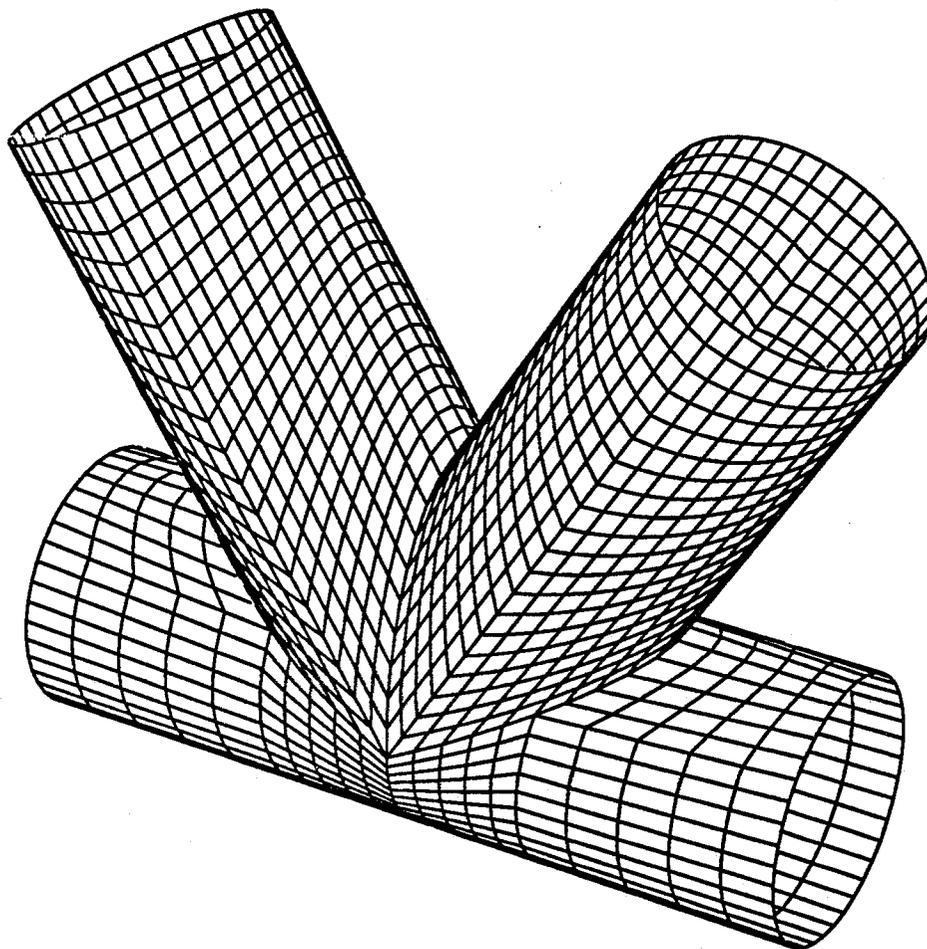
```
-1 -11; 1 21; -1 6 -11;
-2.0 2.0
2.0 10.0
-2.0 0.0 2.0
rr 0 0 0 0 0 0 rx 30;
pb 0 1 2 0 1 2 y 0
sf 0 1 1 0 1 2 sd 3
sf 0 1 2 0 1 3 sd 1
sf 0 0 0 0 0 0 sd 2
sf 0 1 2 0 1 2 plan 0 0 0 0 0 1
coor 1 rxy;
```

```
lrep 0 1;
end
```

end

```
ry 60
rx 32
set tv display
```

three pipes intersecting
INGRID display



bend with fading dent

dn3d

c Line definition, used to describe the dent.

ld 1

lp 2 5.0 0.0 5.0 1.5

lap 4.5 2.0 4.5 1.5

lap 4.0 2.5 4.5 2.5

lap 4.5 3.0 4.5 2.5

lap 5.0 3.5 4.5 3.5

lp 1 5.0 5.0

start

1 31; -1; 1 16 31;

0.0 0.0

5.0

0.0 2.5 5.0

pb 2 0 0 2 0 0 xy 5.0 0.0

sf 0 0 1 0 0 1 cy 0 0 0 0 0 1 5.0

sf 0 0 2 0 0 2 er 0 0 2.5 1 0 0 4.0 5.0

sf 0 0 3 0 0 3 cy 0 0 0 0 0 1 5.0

sf 1 0 0 1 0 0 crz 1

end

end

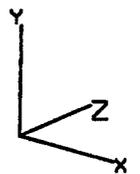
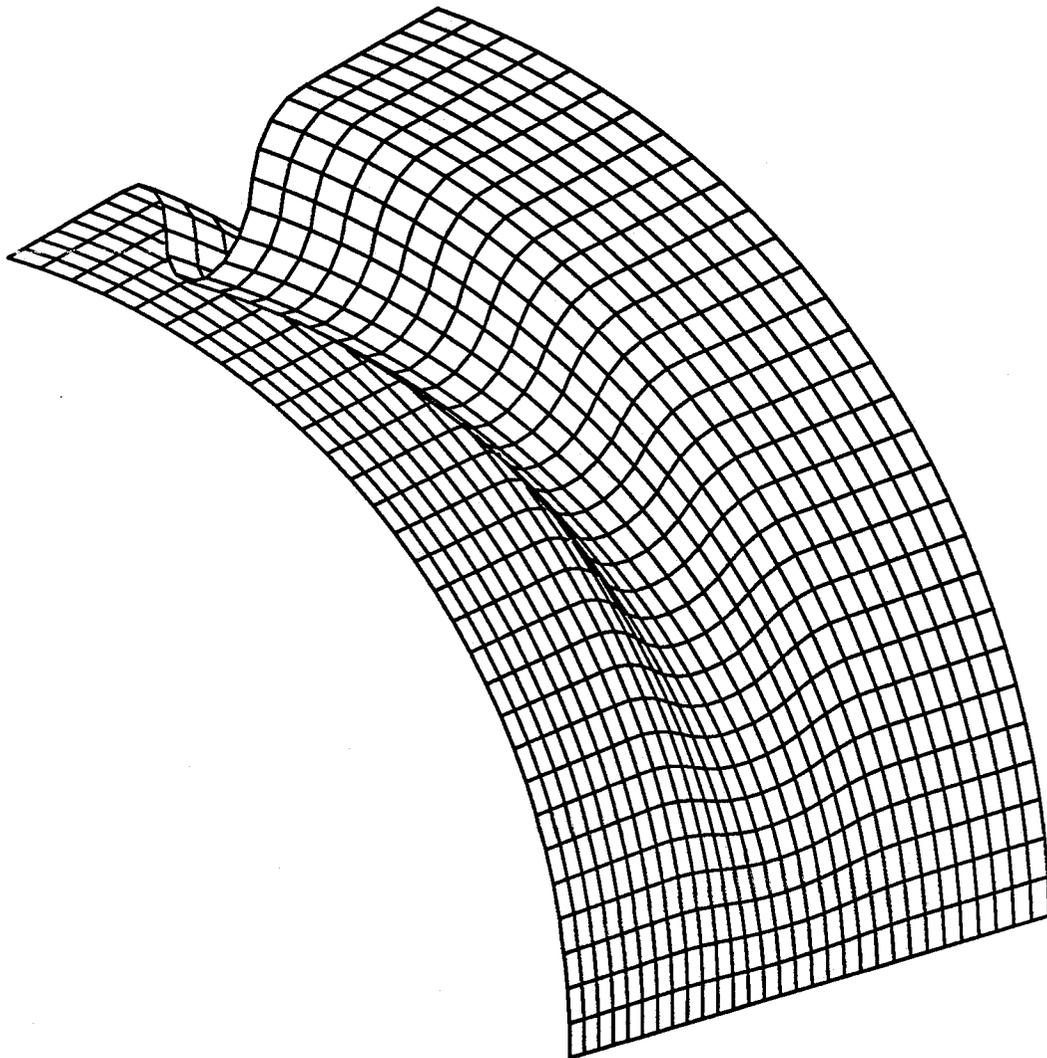
ry 40

rx -20

ry -3

set tv display

bend with fading dent
INGRID display



spherical bubble in block with ratio zoning

dn3d

c Variables.

parameter p1 10

p2 5

rs 1.4;

c Some DYNA analysis options.

term 1

prti 100

plti .01

itss 1.0e-4

c Load curve.

lcd 1 2 0 1 100000 1

c Material definitions.

mat 1 1 e 3.2 pr 17 ro 2.1

mat 2 1 e 3.2 pr 17 ro 2.1

endmat

c Equation of state definitions.

eos 1 1 c0 0 c1 1 c2 2 c3 3

eos 2 1 c0 0 c1 1 c2 2 c3 3

endeos

c Surface definitions.

sd 1 sp 0 0 0 2.0

sd 2 sp 0 0 0 1.5

c Symmetry planes.

plane 2 0 0 0 0 0 -1 .01 symm

0 0 0 -1 0 0 .01 symm

start

1 [1+%p1] [1+%p1+%p2];

1 [1+%p1] [1+%p1+%p2];

1 [1+%p1] [1+%p1+%p2];

0 4 4 0 4 4 0 4 4

d 0 2 2 0 3 3

d 2 0 2 3 0 3

d 2 2 0 3 3 0

d 0 0 0 2 2 2

res 2 1 1 3 2 2 i %rs

res 1 2 1 2 3 2 j %rs

res 1 1 2 2 2 3 k %rs

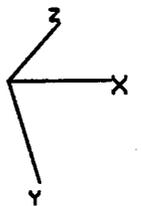
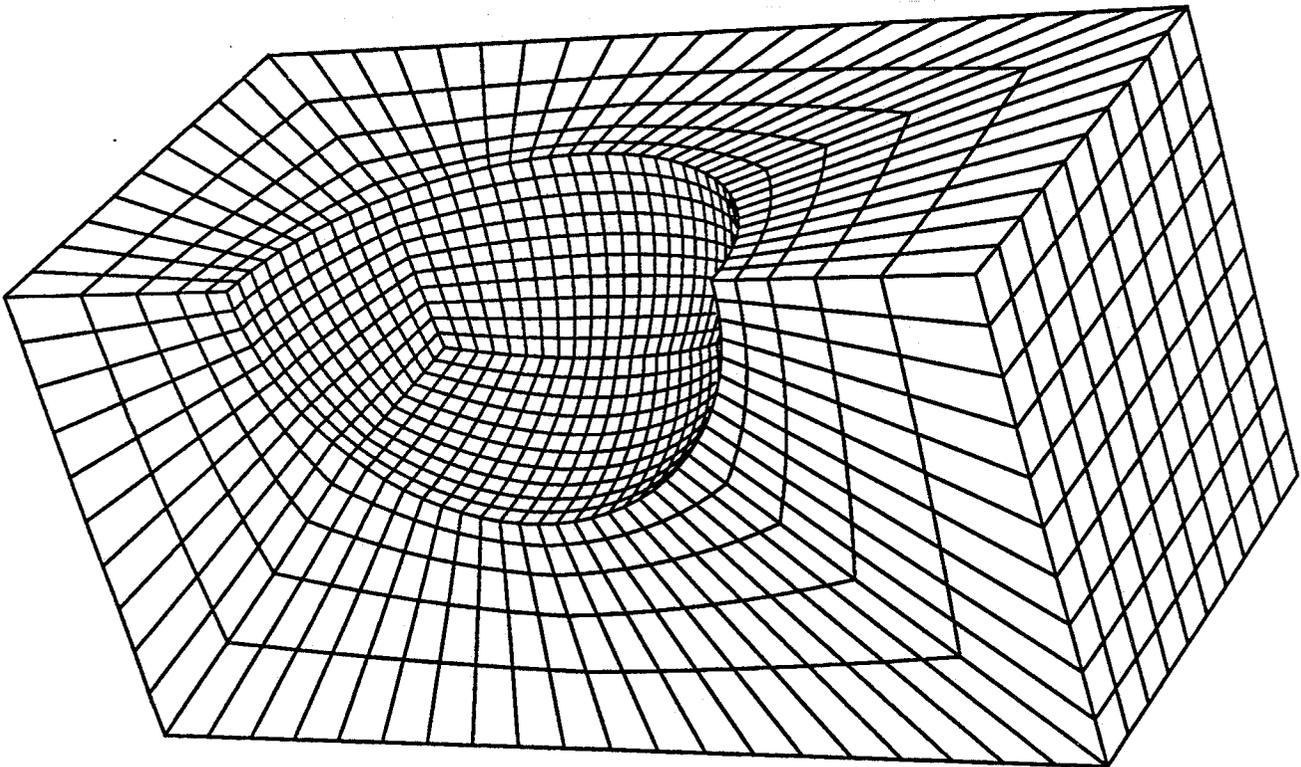
mate 1

pr 1 1 3 2 2 3 1 1 0 0 0

sf 1 1 2 2 2 2 sd 1

```
sf 1 2 1 2 2 2 sd 1
sf 2 1 1 2 2 2 sd 1
lct 1; ryz; lrep 0 1;
end
end
```

```
rx 210
ry -30
set tv display
```



7.6 Advanced Modelling Examples

a-frame by Russ Rosinsky

```

dn3d term 1.5e-01 plti 5e-03 prti 1e+10
c sd 1=bot of lower x-member, sd 2=top of lower x-member
c sd 3=bot of upper x-member, sd 4=top of upper x-member
sd 1 pl3 rt 107.75 62.55501 37.51535
  rt 107.75 67.47247 39.35390
  rt -107.75 29.16460 126.82140 0.
sd 2 pl3 rt 107.75 59.65732 45.26592
  rt 107.75 64.57478 47.10446
  rt -107.75 26.26677 134.57200 0.
sd 3 pl3 rt 107.75 29.16460 126.82140
  rt 107.75 34.08227 128.66010
  rt -107.75 62.55501 37.51535 0.
sd 4 pl3 rt 107.75 26.26677 134.57200
  rt 107.75 31.18444 136.41070
  rt -107.75 59.65732 45.26592 0.
parameter n 2 ;
c -----
start c Middle cross members plus +x A-frame.
  1 [1+5*%n] [1+7*%n] [1+12*%n] [-(1+13*%n)] [1+14*%n] ;
  1 [-(1+5*%n)] [-(1+6*%n)] [-(1+7*%n)] ;
  1 [-(1+4*%n)] [-(1+6*%n)] [-(1+8*%n)] [-(1+10*%n)] [-(1+15*%n)]
    [-(1+17*%n)] [-(1+18*%n)] [-(1+20*%n)] ;
-103.75 -10 10 103.75 107.75 111.75
0 75 79 83
0 38 46 71.25 78.25 127.75 135.50 137.25 144.25
d 0 0 0 4 2 0
d 5 1 1 5 2 2 + k 2 + k 2 + k 2
d 0 3 1 5 3 2 + k 2 + k 2 + k 2
d 0 0 2 0 2 2 + k 1 + k 3 + k 1
d 5 1 2 5 2 3 + k 4
d 4 2 4 5 4 4 + k 1 + k 3 + k 1
d 0 4 0 4 4 0
d 0 0 4 4 0 5 + k 4
d 5 3 0 6 3 0
d 5 2 2 6 4 2 + k 1 + k 1 + k 1 + k 1 + k 1 + k 1 + k 1
d 2 0 0 3 0 3
d 4 3 3 5 3 6
d 4 3 8 5 3 9
sf 1 0 0 1 0 0 plan -103.75 0 0 1 0 0
sf 2 0 0 2 0 3 sd 4
sf 3 0 0 3 0 3 sd 3
sf 2 0 4 2 0 0 sd 1
sf 3 0 4 3 0 0 sd 2
sf 4 0 0 4 0 0 plan 103.75 0 0 1 0 0
sf 5 0 0 5 0 0 plan 107.75 0 0 1 0 0
sf 6 0 0 6 0 0 plan 111.75 0 0 1 0 0

```

```

sf 0 2 0 0 2 0 pl3 rt 103.75 75.01898 1.680362
  rt 111.75 75.01898 1.680362
  rt 103.75 21.71434 144.25 0.
sf 0 3 0 0 3 0 pl3 rt 103.75 [(75.01898+82.46794)/2]
[(1.680362+1.745357)/2]
  rt 111.75 [(75.01898+82.46794)/2] [(1.680362+1.745357)/2]
  rt 103.75 [(21.71434+29.18761)/2] 144.25 0.
sf 0 4 0 0 4 0 pl3 rt 103.75 82.46794 1.745357
  rt 111.75 82.46794 1.745357
  rt 103.75 29.18761 144.25 0.
sf 0 0 1 0 0 1 plan 0 0 0.00 0 0 1
sf 0 0 2 0 0 2 sd 1
sf 0 0 3 0 0 3 sd 2
sf 0 0 4 0 0 4 plan 0 0 71.25 0 0 1
sf 0 0 5 0 0 5 plan 0 0 78.25 0 0 1
sf 0 0 6 0 0 6 sd 3
sf 0 0 7 0 0 7 sd 4
sf 0 0 8 2 0 8 plan 0 0 137.25 0 0 1
sf 0 2 8 0 0 8 pl3 rt 103.75 24.33155 137.25
  rt 111.75 24.33155 137.25
  rt 103.73 29.18761 140.75 0.
sf 0 0 9 0 0 9 plan 0 0 144.25 0 0 1
lct 1 rxz ; lrep 0 1 ;
end
c -----
start c -x A-frame.
  1 [1+5*%n] [1+7*%n] [1+12*%n] [-(1+13*%n)] [1+14*%n] ;
  1 [-(1+5*%n)] [-(1+6*%n)] [-(1+7*%n)] ;
  1 [-(1+4*%n)] [-(1+6*%n)] [-(1+8*%n)] [-(1+10*%n)] [-(1+15*%n)]
  [-(1+17*%n)] [-(1+18*%n)] [-(1+20*%n)] ;
-103.75 -10 10 103.75 107.75 111.75
0 75 79 83
0 38 46 71.25 78.25 127.75 135.50 137.25 144.25
d 0 0 0 4 2 0
d 5 1 1 5 2 2 + k 2 + k 2 + k 2
d 0 3 1 5 3 2 + k 2 + k 2 + k 2
d 0 0 2 0 2 2 + k 1 + k 3 + k 1
d 5 1 2 5 2 3 + k 4
d 4 2 4 5 4 4 + k 1 + k 3 + k 1
d 0 4 0 4 4 0
d 0 0 4 4 0 5 + k 4
d 5 3 0 6 3 0
d 5 2 2 6 4 2 + k 1 + k 1 + k 1 + k 1 + k 1 + k 1 + k 1
d 2 0 0 3 0 3
d 4 3 3 5 3 6
d 4 3 8 5 3 9

d 0 0 0 4 0 0

```

```

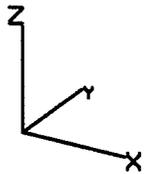
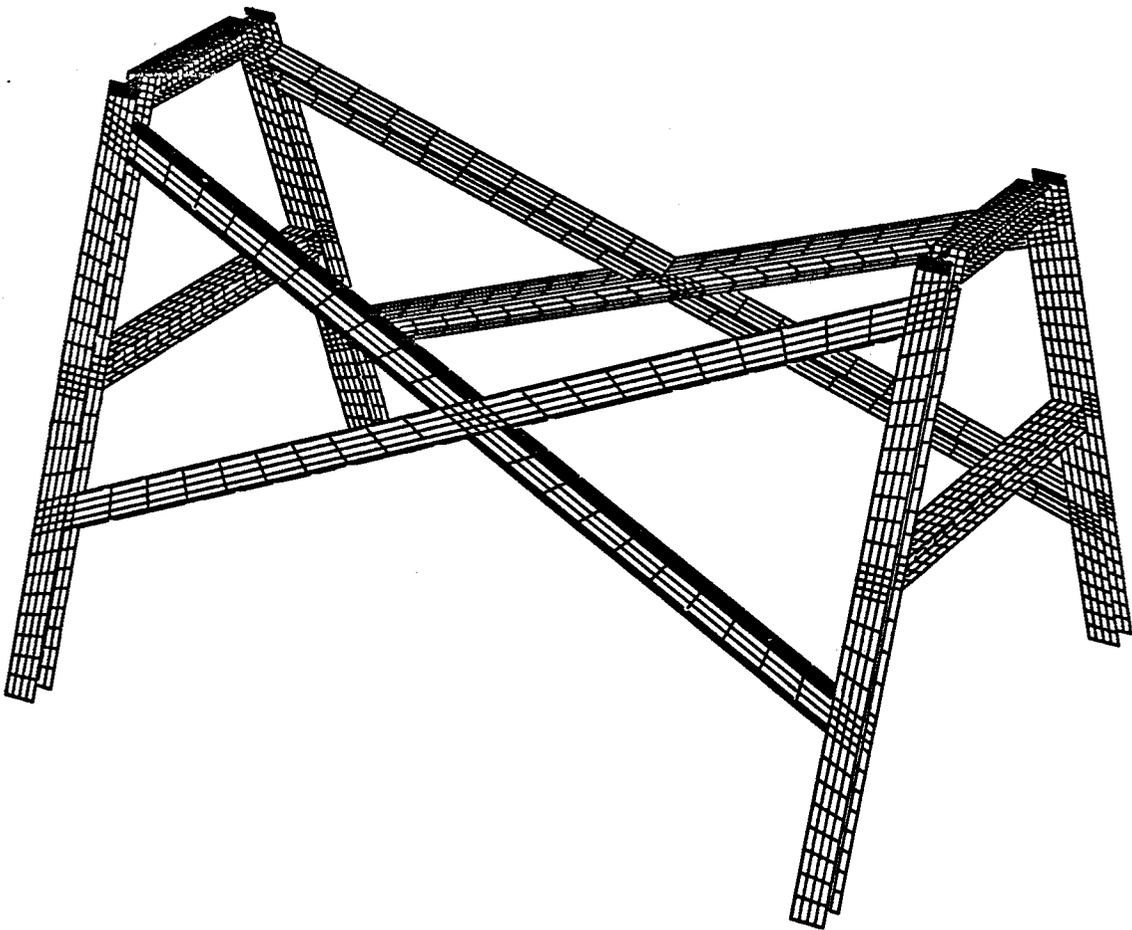
sf 1 0 0 1 0 0 plan -103.75 0 0 1 0 0
sf 2 0 0 2 0 3 sd 4
sf 3 0 0 3 0 3 sd 3
sf 2 0 4 2 0 0 sd 1
sf 3 0 4 3 0 0 sd 2
sf 4 0 0 4 0 0 plan 103.75 0 0 1 0 0
sf 5 0 0 5 0 0 plan 107.75 0 0 1 0 0
sf 6 0 0 6 0 0 plan 111.75 0 0 1 0 0
sf 0 2 0 0 2 0 pl3 rt 103.75 75.01898 1.680362
  rt 111.75 75.01898 1.680362
  rt 103.75 21.71434 144.25 0.
sf 0 3 0 0 3 0 pl3 rt 103.75 [(75.01898+82.46794)/2]
[(1.680362+1.745357)/2]
  rt 111.75 [(75.01898+82.46794)/2] [(1.680362+1.745357)/2]
  rt 103.75 [(21.71434+29.18761)/2] 144.25 0.
sf 0 4 0 0 4 0 pl3 rt 103.75 82.46794 1.745357
  rt 111.75 82.46794 1.745357
  rt 103.75 29.18761 144.25 0.
sf 0 0 1 0 0 1 plan 0 0 0.00 0 0 1
sf 0 0 2 0 0 2 sd 1
sf 0 0 3 0 0 3 sd 2
sf 0 0 4 0 0 4 plan 0 0 71.25 0 0 1
sf 0 0 5 0 0 5 plan 0 0 78.25 0 0 1
sf 0 0 6 0 0 6 sd 3
sf 0 0 7 0 0 7 sd 4
sf 0 0 8 2 0 8 plan 0 0 137.25 0 0 1
sf 0 2 8 0 0 8 pl3 rt 103.75 24.33155 137.25
  rt 111.75 24.33155 137.25
  rt 103.73 29.18761 140.75 0.
sf 0 0 9 0 0 9 plan 0 0 144.25 0 0 1
c lct 1; ryz ; lrep 1 ;
lct 2; ryz ; rxz ryz ; lrep 1 2 ;
end
c
*****
end

rx -90
ry -30
rx 25
set tv display

```

a-frame by Russ Rosinsky

INGRID display



threaded bolt

dn3d

c Head.

start

```
1 4; 1 5 9 13 17 21 25; 1 3;
.25 .45
0 0 0 0 0 0 0
0 .2
rr 0 2 0 0 2 0 rz 60;
rr 0 3 0 0 3 0 rz 120;
rr 0 4 0 0 4 0 rz 180;
rr 0 5 0 0 5 0 rz 240;
rr 0 6 0 0 6 0 rz 300;
rr 0 7 0 0 7 0 rz 360;
sf 1 0 0 1 0 0 cy 0 0 0 0 0 1 .25
end
```

c Shaft.

start

```
1 7; 1 7; 1 3 10;
-1 1
-1 1
0 .2 1.5
sfi -1 -2;-1 -2;; cy 0 0 0 0 0 1 .25
end
```

c Thread.

start

```
1 2 3; 1 501; 1 2 3;
.25 .275 .3
0 9000
.2 .225 .25
d 2 0 2 3 0 3
pa 2 0 1 z .2125
pa 3 0 1 z .225
pa 2 0 2 x .27
pa 2 0 3 xz .275 .2375
pa 3 0 2 xz .275 .2375
ma 0 2 0 z 1.25
cyli
end
```

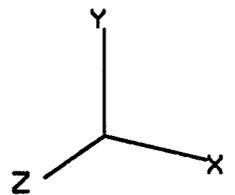
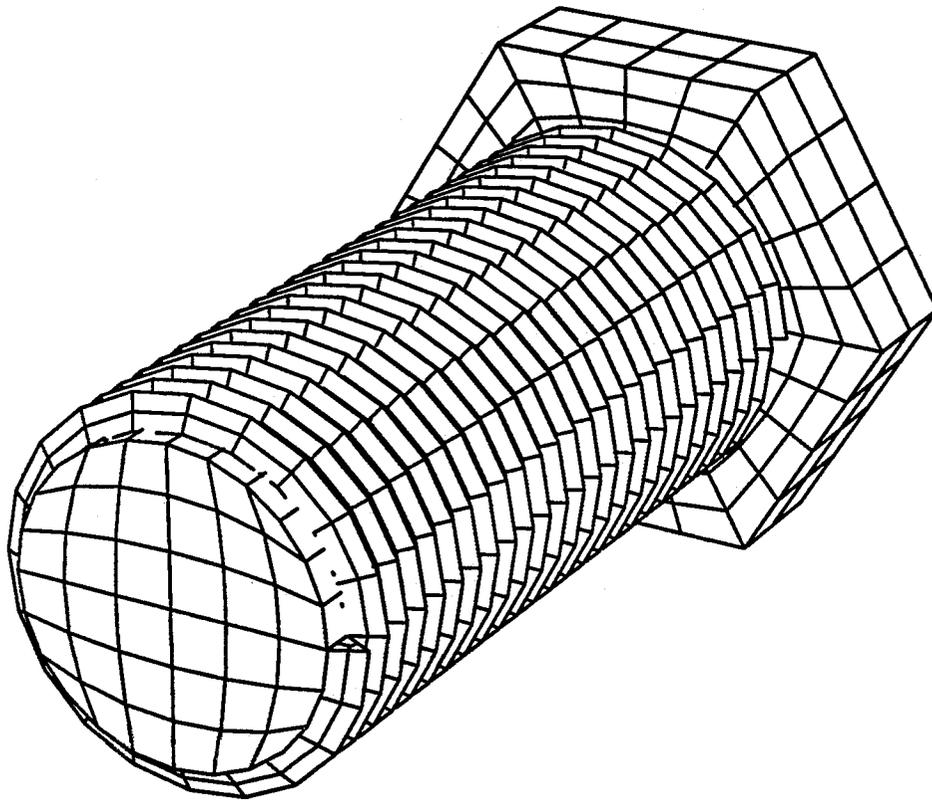
end

ry -30

rx 24

set tv display

threaded bolt
INGRID display

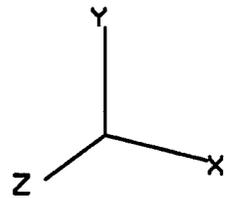
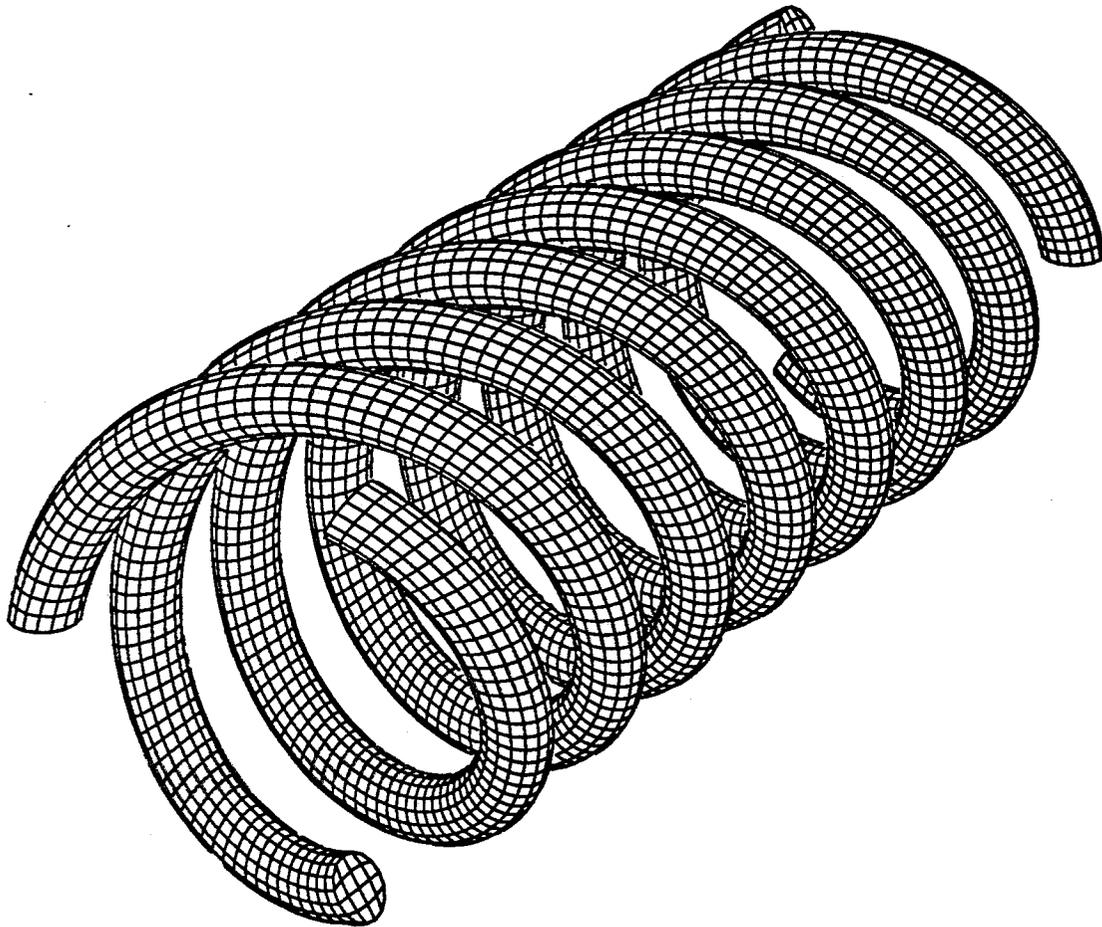


```
helix
dn3d
start
  1 5;
  1 226;
  1 5;
  1.5 2.5 0 900 -.5 .5
cyl1
sfi -1 -2 ; 1 2 ; -1 -2 ; ts 0 0 0 0 0 1 2 0 .3
lct 2;
rz 120;
rz 240;
lrep 0 1 2;
edit
fir 1 1 1 5 226 5
z = z + 0.050*j
end
end

ry -30
rx 25
scale 1.2
set tv display
```

helix

INGRID display



piston

dn3d

```
parameter m 2 n 1 t 1; c course mesh
c parameter m 3 n 2 t 2; c medium mesh
c parameter m 5 n 4 t 3; c fine mesh
```

```
lcd 1 2 0 0 1 1
sd 1 cy 0 0 0 1 0 0 20 sd 2 cy 0 0 0 1 0 0 18
sd 3 cy 21 0 0 0 0 1 8 sd 4 cy 2 0 0 0 1 1 9
sd 5 cy 2 0 0 0 -1 1 9 sd 6 cy 21 0 0 0 0 1 5
sd 7 cy 0 0 0 1 0 0 16
```

start

```
1 [1+%m] [1+2*%m] [1+5*%m] [1+6*%m] [1+6*%m+%n]
[1+6*%m+2*%n] [1+6*%m+3*%n] [1+6*%m+4*%n]
[1+6*%m+5*%n] [1+6*%m+6*%n] [1+7.5*%m+6*%n];
1 [1+%t] [1+2*%t] [1+3*%t+%m] [1+3*%t+4*%m]
[1+4*%t+5*%m] [1+5*%t+5*%m] [1+6*%t+5*%m];
1 [1+%t] [1+2*%t] [1+3*%t+%m] [1+3*%t+4*%m]
[1+4*%t+5*%m] [1+5*%t+5*%m] [1+6*%t+5*%m];
2 11 15 27 31 32 33 34 35 36 37 43
-20 -20 -20 -6.5 6.5 20 20 20
-20 -20 -20 -6.5 6.5 20 20 20
```

```
d 1 2 0 2 7 0
d 5 2 0 6 3 0 + j 4
d 5 3 3 10 6 6
d 3 4 0 4 5 0
d 1 2 2 5 7 7
di ;1 3 0 6 8;1 3 0 6 8;
di 5 7 0 8 9 0 10 11;;1 2 0 7 8;
di 1 2;;1 4 0 5 8;
di 5 7 0 8 9 0 10 11;1 2 0 7 8;;
```

```
a 0 3 3 0 6 6 1 14
a 12 2 2 12 7 7 1 16
pb 3 4 0 3 4 0 x 13
pb 3 5 0 3 5 0 x 13
pb 4 4 0 4 4 0 x 29
pb 4 5 0 4 5 0 x 29
pb 3 4 0 4 4 0 2 -8
pb 3 5 0 4 5 0 2 8
mb 12 3 3 12 6 6 1 2
sf 3 4 0 4 5 0 sd 3
fvc 0 1 0 0 1 0 1 1.0 -1 0 0
fvc 0 8 0 0 8 0 1 1.0 -1 0 0
fvc 0 0 1 0 0 1 1 1.0 -1 0 0
fvc 0 0 8 0 0 8 1 1.0 -1 0 0
```

```
sfi 1 12;-1 -8;-1 -8; sd 1 sfi 1 11;-2 -7;-2 -7; sd 2
sfi 1 -2;7 8;-5 6; sd 4 sfi 1 -2;-5 6;7 8; sd 4
sfi 1 -2;7 8;3 -4; sd 5 sfi 1 -2;3 -4;7 8; sd 5
sfi 1 -2;1 2;-5 6; sd 5 sfi 1 -2;-5 6;1 2; sd 5
sfi 1 -2;1 2;3 -4; sd 4 sfi 1 -2;3 -4;1 2; sd 4
end
```

```
start
```

```
1 [1+%m] [1+4*%m] [1+5*%m];
1 [1+%m] [1+4*%m] [1+5*%m];
1 [1+%t] [1+2*%t] 0 [1+4*%t] [1+5*%t] [1+6*%t];
13 13 29 29
-8 -8 8 8
-20 -20 -14 0 14 20 20
d 2 2 0 3 3 0
di 1 2 0 3 4;1 2 0 3 4;;
sf 2 2 0 3 3 0 sd 6
sfi -1 -4 ; -1 -4 ; ; sd 3
sfi ;;-1 -7; sd 1
sfi ;;-2 -6; sd 2
end
```

```
start
```

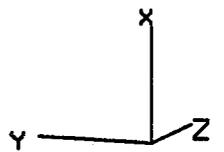
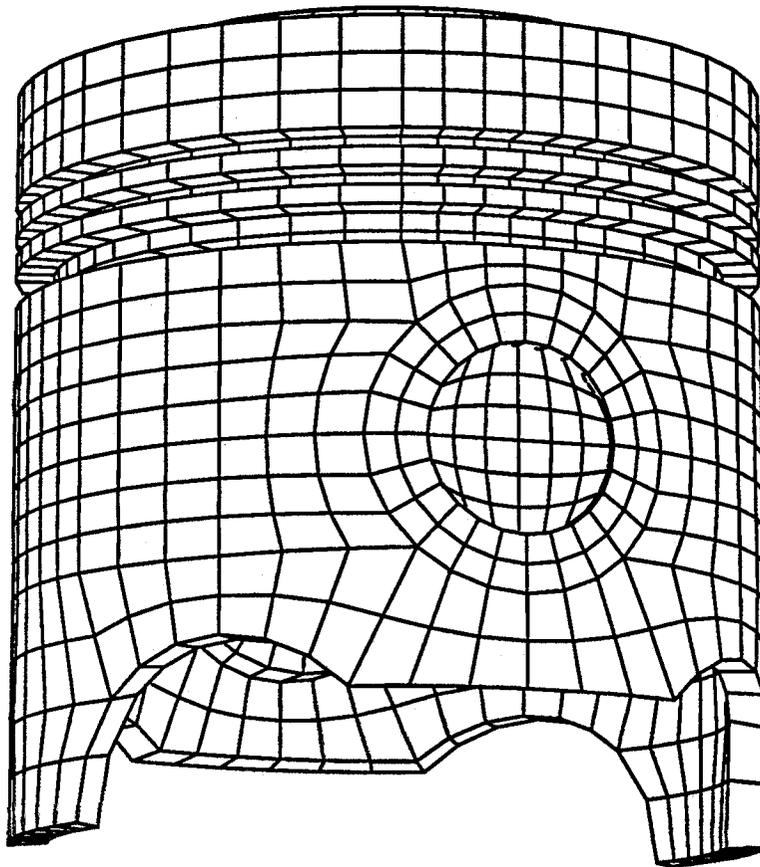
```
1 [1+3*%m];1 [1+3*%m];1 [1+7*%m];
20 22 -1 1 -19 19
a 0 0 0 0 0 0 3 5
end
```

```
end
```

```
rz 90
ry 20
rx -10
set tv display
```

piston by Robert Rainsberger

INGRID display



Concrete test column with explicit rebar definition

dn3d

```
start
  1 2 3 4;
  1 2 3 4;
  1 2 3 4 5 6;
  0. 1. 2. 3.
  0. 1. 2. 3.
  0. 1. 2. 3. 4. 5.
c Generate beams.
ibm 2 2 2 3 3 5 2 1 1 k 1
b 0 0 0 0 0 0 0 000001
c Lateral pressure loading.
pri 4; 1 4; 5 6; 1 1.0 -10. 0. 0.
mate 1
end

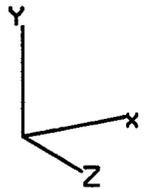
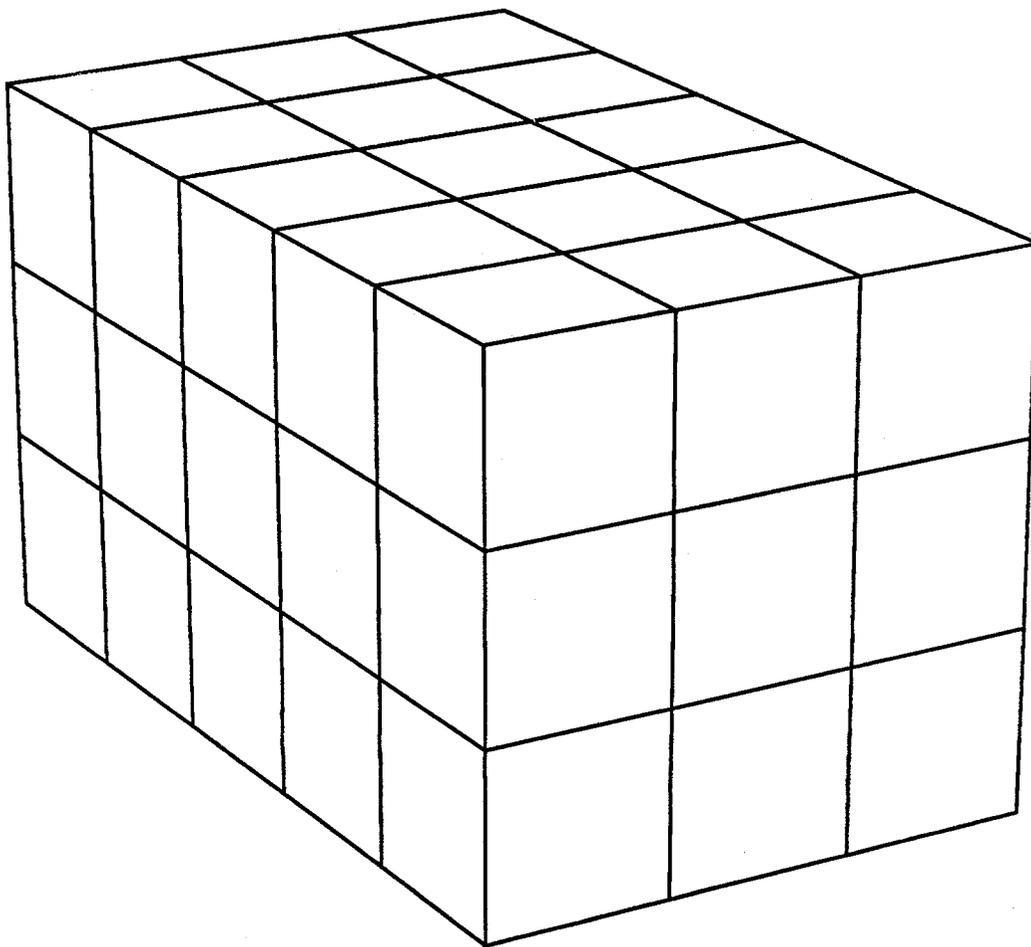
c Material definitions.
mat 1 3
head
Bent concrete - 5,000 psi yield
ro 2.2465e-4
e 4.e6
pr .21
sigy 5000.
etan 1.e3
beta 0.0
endmat

mat 2 3 beam
head
Hinge section central rebar (#10 dia=1.1811 in)
ro 7.324e-4
e 3.e7
pr .3
sigy 46000.
etan 1.1e5
beta 1.0
sloc 0.
tloc 0.
tthi 1.1811
sthi 1.1811
endmat

end
ry 30
rx 20
set tv display
```

Concrete test column with explicit rebar definition

INGRID display



roller

dn3d

```
sd 1 cone 0 0 0 0 0 1 0 35.311
sd 2 cone 0 0 0 0 0 1 5.667 -35.311
sd 3 cone 0 0 0 0 0 1 .75 35.311
sd 4 cone 0 0 0 0 0 1 6.417 -35.311
sd 5 plan 0 0 0 0 0 1
sd 6 plan 0 0 8 0 0 1
```

```
gct 1 ; csca .3866666666 mz 2 ;
lev 1 grep 0 1 ; ;
lev 2 levct 1 ; rxy mz -20 ; grep 0 ; ;
pslv 2
pslv 1
```

start

```
1 2 6 7 8;
1 3 5 6 8 10;
1 3 4 6 7 9;
4.25 5 10 10.5 11
0 8 13.5 16.5 22 30
0 2 3 5 6 8
```

cyli

```
di 2 3 ; 1 3 0 4 6 ; ;
di 2 3 ; ; 1 3 0 4 6 ;
di 3 4 ; 1 2 0 5 6 ; ;
di 3 4 ; 2 5 ; 1 2 0 5 6 ;
di 1 2 ; 1 2 0 5 6 ; ;
```

```
pb 1 2 0 2 2 0 2 0
pb 1 5 0 2 5 0 2 30
mb 2 3 0 2 3 0 2 -1.3 + j 1 2 1.3
sf 1 0 1 1 0 2 sd 2 + k 4 sd 1
sf 2 0 1 2 0 2 sd 4 + k 4 sd 3
sf 0 0 1 0 0 1 sd 5
sf 0 0 6 0 0 6 sd 6
```

```
cpl 1 2 0 1 5 0 j
lct 11 ; rz 30 ; repe 11 ;
lrep 0 1 2 3 4 5 6 7 8 9 10 11 ;
end
```

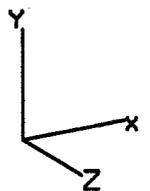
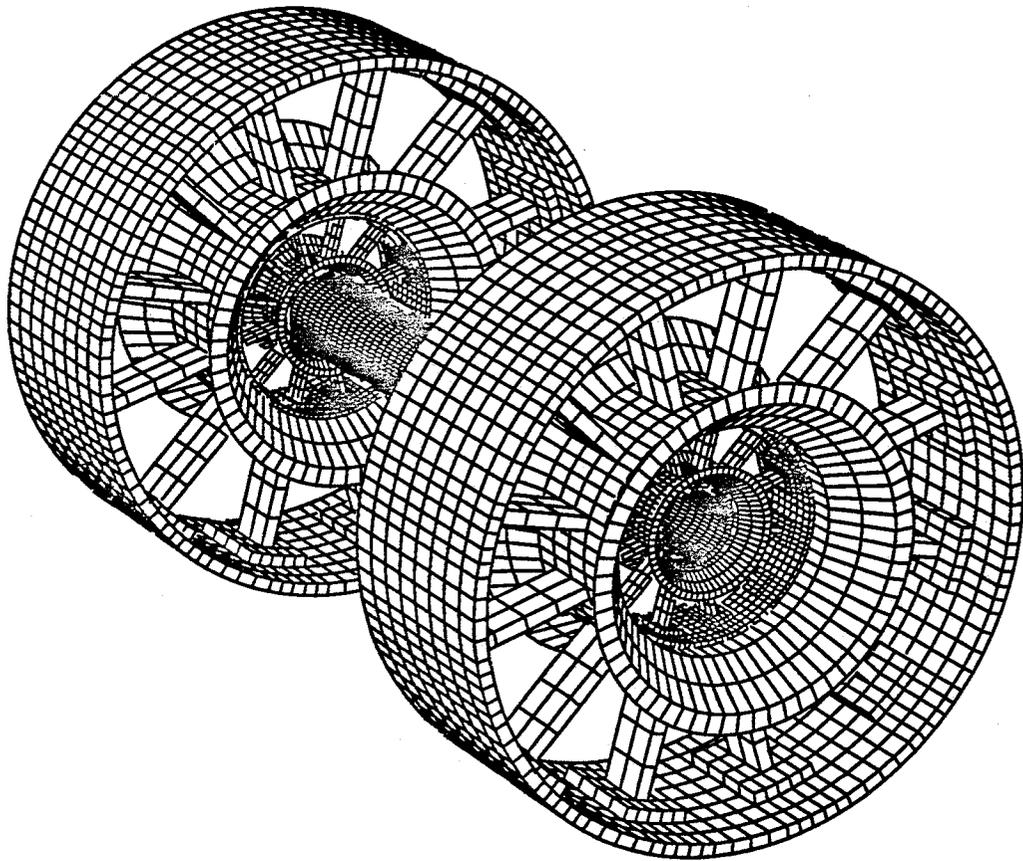
```
pplv
pplv
```

```
start
  1 2 ; 1 61 ; 1 60 ;
  1.4 1.64333333333 0 360 -24 4
cyli
end

end

ry 30
rx 20
set tv display
```

roller
INGRID display



wood screw

dn3d

```
sd 1 sp 0 0 0 2
sd 2 plan 0 0 1 0 0 1
```

start

```
1 5 13 21 25;
1 5 11 13 15 21 25;
1 5 9;
-.7 -.7 0 .7 .7
-.5 -.5 -.3 0 .3 .5 .5
0 .5 .5
di 1 2 0 4 5; 1 7; 2 3;
di 1 5; 1 2 0 6 7; 2 3;
d 2 3 2 4 5 3
di 1 2 0 4 5; 1 2 0 6 7; 1 2;
sfi -1 -5; 2 6; 1 2; sd 1
sfi 2 4; -1 -7; 1 2; sd 1
sfi 2 4; 2 3 0 5 6; -3; sd 1
sf 0 0 2 0 0 2 sd 2
sf 0 0 1 0 0 1 plan 0 0 0 0 0 1
sfi 2 4; -2 -6; 2 3; sd 2
sfi -2 -4; 2 3 0 5 6; 2 3; sd 2
sf 0 3 0 0 3 0 plan 0 -.3 0 0 1 0
sf 0 5 0 0 5 0 plan 0 .3 0 0 1 0
edit
z=-.75*z-.2
end
```

start

```
1 2; 1 37; 1 3;
0 .9 0 360 -.2 0
cyli
edit
fir 1 1 2 2 37 2
z=-.1+.2*(j-1)/36
fir 1 1 3 2 37 3
z=.4*(j-1)/36
end
```

start

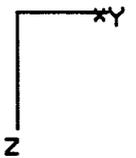
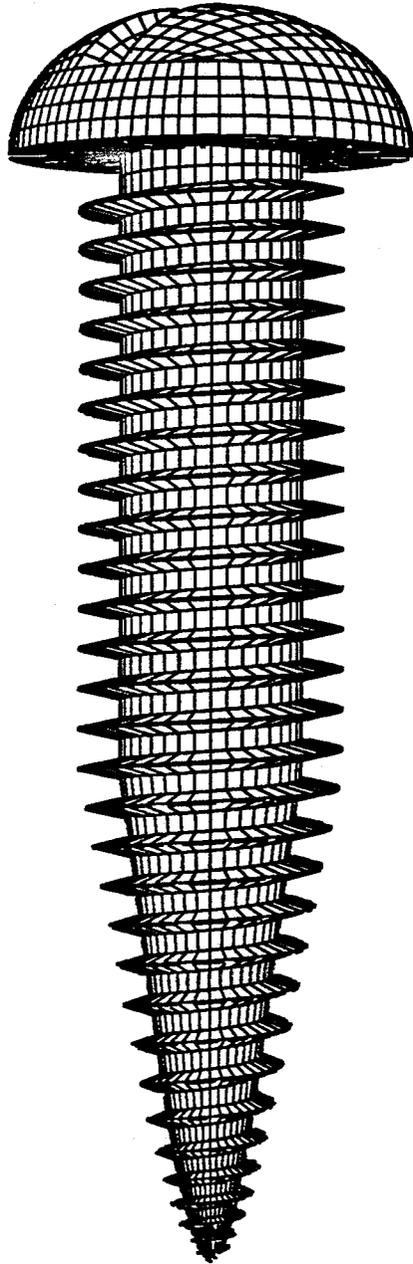
```
1 2 3; 1 1081; 1 2 3;
0 .9 1.3
0 10800
0 .2 .4
d 2 0 2 3 0 3
cyli
edit
```

```
z=z+(j-1)*.4/36
fir 3 1 1 3 1081 1
z=z+.09
fir 3 1 2 3 1081 2
z=z-.09
cyli
fir 1 1 1 3 1081 3
phr 0 -99999 6 6 99999 50
x=x*(1-.15*(z-6))
phr 0 -99999 9 6 99999 50
z=(z-9)*(1-(z-9)/8)+9
end

end

rx 90
ry 50
set tv display
```

wood screw by Robert Rainsberger
INGRID display



wheel - tire, felly, spokes and wheel-nave

```

parameter c choose the next 6 parameters
rad1 300 c center of wheel nave to center of tire
rad2 25 c radius of tire
rad5 [%rad1-%rad2]
sig [%rad2*3/(4*sqrt(2))]
rad4 [%rad5*cos(80)] c outer radius of wheel nave
rad3 20 c inner radius of wheel nave
rad6 10
wnl 40 c half length of wheel nave
rad7 [%wnl*(%rad5-%rad3)/(%rad5-%rad4)];

```

```
dn3d
```

```

term 20.01
prti 100
plti 0.5
lcd 1 2 0 12.5 100 12.5

```

```

mat 1 3 shell e 210 pr .33 sigy .2649 etan .3737 beta 1 ro .785e-5
mat 2 7 g 100 ro .785e-5
mat 3 20 e 210 pr .33 ro .785e-3
mat 4 1 beam e 210 pr .33 ro .785e-5 endmat

```

```

start
-1; 1 289; 1 9; c felly
[%rad1-%sig]
0 360
[-%sig] %sig
cyli
mate 1
thic 1
sfi -1;;; ts 0 0 0 0 0 1 %rad1 0 %rad2
end

```

```

start
1 2 6 7; 1 145; 1 2 6 7; c tire
[%rad1-%sig] [%rad1-%sig] [%rad1+%sig] [%rad1+%sig]
0 360
[-%sig] [-%sig] %sig %sig
cyli
mate 2
di 1 2 0 3 4;;1 2 0 3 4;
sfi -1 -4;;-1 -4; ts 0 0 0 0 0 1 %rad1 0 %rad2
end

```

```
start
  1 3; 1 37; 1 3; c wheel nave
  %rad6 %rad3
  0 360
  [-%rad7] %rad7
cyli
mate 3
end

gct 1;
rxy rz 10; lev 1 grep 0 1;;
pslv 1

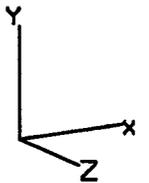
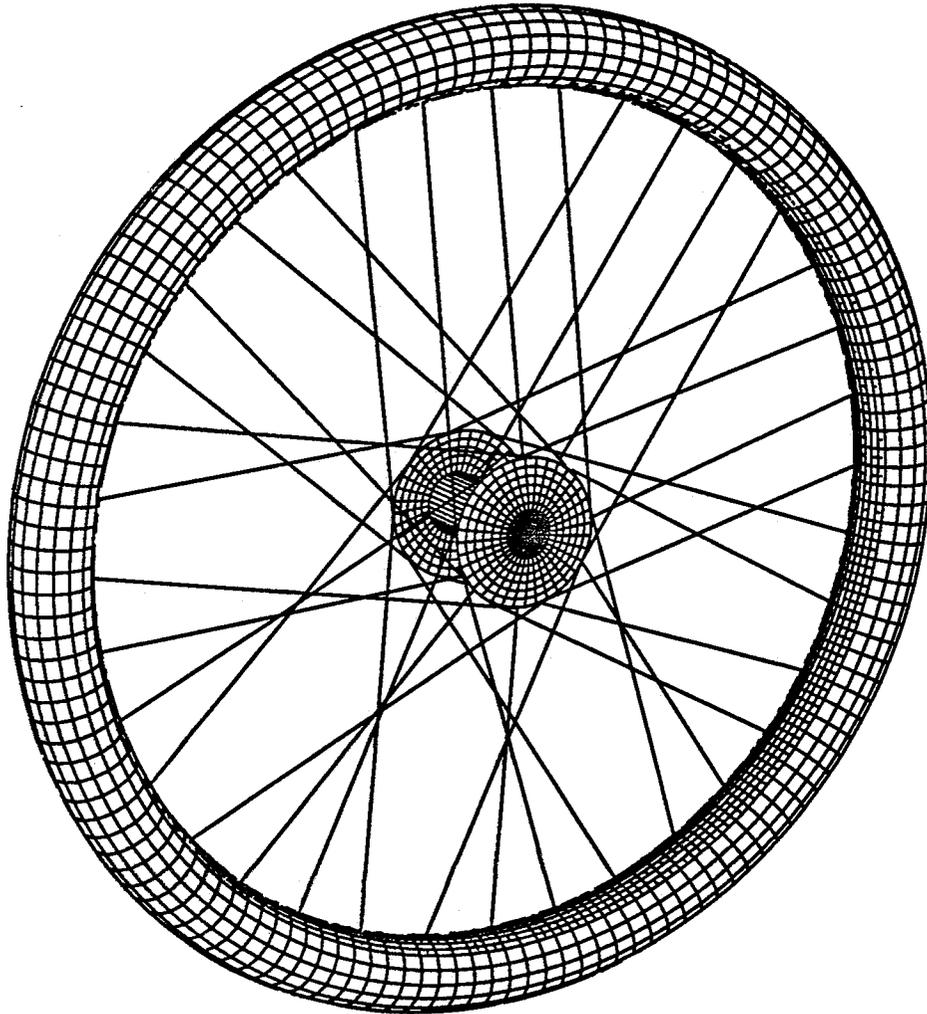
start
  1 6; 1 37; -1;
  %rad3 %rad4
  0 360
  %wnl
pb 1 0 0 1 0 0 z %rad7
cyli
mate 3
end

beam
cy 000000 [%rad5*.5] 10 [%wnl*.5]
cy 000000 %rad4 100 %wnl cy 000000 %rad5 20 0
cy 000000 %rad4 -80 %wnl cy 000000 %rad5 0 0
0 2 3 3 4 1 1 4 5 3 4 1 1 0
lct 8; rz 40;
repe 8; lrep 0 1 2 3 4 5 6 7 8;
end

pplv

end
ry 30
rx 15
scale 1.1
set tv display
```

wheel - tire, felly, spokes and wheel-nave
INGRID display



ACKNOWLEDGEMENTS

This manual is based upon the work of Bill Cook, Doug Stillman, John Hallquist, Robert Rainsbugger and Brett Wayne. The authors wish to acknowledge the numerous suggestions provided by INGRID collaborators and users. Special thanks are given to our colleagues in the Methods Development Group at LLNL: Bruce Englemann, Brad Maker, Jerry Goudreau, and Robert Whirley. This manual would not have been possible without the efforts of Nikki Falco and Jeanette Tootle. Special thanks are due to Brad Maker for contributing Chapter 3.

The authors also wish to acknowledge Ed Zywicz, Roger Logan and Joe Stoner for their suggestions in preparing this document.

Finally, the authors gratefully acknowledge the pioneering work of Doug Stillman and John Hallquist in developing INGRID.

Bibliography

- Cook, W.A., "INGEN: A General Purpose Mesh Generator for Finite Element codes," Los Alamos National Laboratory, Report LA-7135-MS, 1978.
- Lovejoy, S. C., and R.G. Whirley, "DYNA3D Example Problem Manual, University of California, Lawrence Livermore, National Laboratory, UCRL-MA-105259, 1990.
- Maker, B.N., R.M Ferencz, and J.O. Hallquist, "NIKE3D: An Implicit, Finite-deformation, Finite Element Code for Analyzing The Static and Dynamic Response of Three Dimensional Solids," University of California, Lawrence Livermore National Laboratory, UCRL-MA-105268, 1991.
- Shapiro, A.B., "TOPAZ3D - A three dimensional finite element heat transfer code," University of California, Lawrence Livermore National Laboratory, Rept. UCID-20484, 1985.
- Spelce, T., and J.O. Hallquist, "TAURUS: An Interactive Post-Processor for the Analysis Codes, NIKE3D, DYNA3D and TOPAZ3D," University of California, Lawrence Livermore, National Laboratory, UCRL-MA-105401, 1991.
- Whirley, R.G., and J.O. Hallquist, "DYNA3D A Nonlinear, Explicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics - User Manual," University of California, Lawrence Livermore National Laboratory, UCRL-MA-107254, 1991.

Index

A

adina 48
 ADINA Material Type 1
 Isotropic Linear Elastic 132
 ADINA Material Type 10
 Thermo-Elastic Plastic 133
 ADINA Material Type 9
 Elastic Plastic 132
 ADINA Materials 132
 Advanced Modelling Examples 387
 ANALYSIS OPTIONS 19
 Analysis Options 47
 Analytic 3-D Surface Definitions 140
 ansys 48
 ape 48

B

BASIC CONCEPTS 23
 bdiv 48
 Beam Element Attributes 211
 Beam Element Generation 192
 Beam Part Options 211
 Beam Parts 208
 Beam Section Definitions 66
 brode 50

C

Coordinate Specification 210, 211, 212
 COORDINATE SYSTEMS 19
 Coordinate Systems 235
 cube 48

D

Definitions 23, 24, 25, 28, 31
 Discrete 3-D Surface Definitions 138
 Display Commands 219, 239
 Displaying Part and Material Subsets 249
 dn2d 48
 dn3d 48
 dn3d88 48
 dn3d88s 48
 DYNA3D Analysis Options 50
 DYNA3D Equation of State Types 112

DYNA3D Equation-of-State Form 1
 Linear Polynomial 113
 DYNA3D Equation-of-State Form 11
 Tensor Pore Collapse 120
 DYNA3D Equation-of-State Form 2
 JWL 114
 DYNA3D Equation-of-State Form 3
 Sack "Tuesday" 114
 DYNA3D Equation-of-State Form 4
 Gruneisen 115
 DYNA3D Equation-of-State Form 5
 Ratio of Polynomials 115
 DYNA3D Equation-of-State Form 6
 Linear Polynomial with Energy Leak 117
 DYNA3D Equation-of-State Form 7
 Ignition and Growth of Reaction in HE 118
 DYNA3D Equation-of-State Form 8
 Tabulated Compaction 119
 DYNA3D Equation-of-State Form 9
 Tabulated 119
 DYNA3D Examples 272
 DYNA3D Material Type 1
 Elastic 86
 DYNA3D Material Type 10
 Isotropic-Elastic-Plastic-Hydrodynamic 93
 DYNA3D Material Type 11
 Temperature Dependent, Elastoplastic, Hydrodynamic 93
 DYNA3D Material Type 12
 Isotropic-Elastic-Plastic 95
 DYNA3D Material Type 13
 Elastic-Plastic with Failure Model 95
 DYNA3D Material Type 14
 Soil and Crushable Foam with Failure Model 96
 DYNA3D Material Type 15
 Johnson/Cook Strain and Temperature Sensitive Plasticity 97
 DYNA3D Material Type 16
 Pseudo TENSOR Concrete/Geological Model 98
 DYNA3D Material Type 17
 Isotropic Elastic-Plastic Oriented Crack Model 99
 DYNA3D Material Type 18
 Power Law Isotropic Plasticity 99

-
- DYNA3D Material Type 19
 - Strain Rate Dependent Isotropic Plasticity 100
 - DYNA3D Material Type 2
 - Orthotropic Elastic 87
 - DYNA3D Material Type 20
 - Rigid 101
 - DYNA3D Material Type 21
 - Thermal Orthotropic Elastic 101
 - DYNA3D Material Type 22
 - Composite Damage Model 103
 - DYNA3D Material Type 23
 - Thermal Orthotropic Elastic with 12 Curves 104
 - DYNA3D Material Type 24
 - Piecewise Linear Isotropic Plasticity 106
 - DYNA3D Material Type 25
 - Inviscid Two Invariant Geologic Cap Model 107
 - DYNA3D Material Type 26
 - Metallic Honeycomb 107
 - DYNA3D Material Type 27
 - Compressible Mooney-Rivlin Rubber 109
 - DYNA3D Material Type 28
 - Resultant Plasticity 109
 - DYNA3D Material Type 30
 - Closed-Form Update Shell Plasticity 110
 - DYNA3D Material Type 31
 - Frazer-Nash Rubber Model 111
 - DYNA3D Material Type 4
 - Thermo-Elastic-Plastic 89
 - DYNA3D Material Type 5
 - Soil and Crushable Foam 90
 - DYNA3D Material Type 6
 - Viscoelastic 91
 - DYNA3D Material Type 7
 - Blatz-Ko Rubber 91
 - DYNA3D Material Type 8
 - High Explosive Burn 92
 - DYNA3D Material Type 9
 - Null Hydrodynamics 92
 - DYNA3D Material Types 84
 - DYNA3D Materials 83
- E**
- Edit and the In-line Calculator 193
 - Electromagnetic Loads and Boundary Conditions 181, 202
 - Electrostatic Loads and Boundary Conditions 183, 203
 - Element Generation Commands 210
 - ELEMENT TYPES 14
 - Equation-of-State Definitions 113
 - es3d 48
 - Establishing Local T-axis Orientation 78
 - Executing INGRID 234
- F**
- face 48
 - facrx 50
 - Fourier Series Definitions 61
- G**
- gem3 48
 - gem3d 48
 - gemini 48
 - General Control Commands 62, 165
 - General Line Display 227
 - Geometric Transformations 63
 - Geometric Transformations - Point/Line Modifications 175, 176
 - Geometric Transformations - Projections 167
 - Geometrical Part Control 66, 179
 - Global Control Commands 47
 - Global Material Commands 79
 - Global Material Control Commands 77
 - Global Material Specification 79
 - Graphical Diagnostic Information Display - the di commands 260
 - Graphics Diagnostic Commands 224
 - grav 50
- H**
- hydt 50
- I**
- Identification of Nodes in Regions Subjected to Load Curves 225
 - Identification of Regions Subject to Prescribed Boundary Conditions 226
 - Import Part Control Commands 218
 - Importing Existing Mesh Parts 213
-

- INGRID Input Format 45
- INGRID Part Library 311
- INGRID USER'S GUIDE 231
- Initial & Boundary Conditions 70
- INTERACTIVE COMMANDS 231
- Interactive Commands 219
- INTERACTIVE FEATURES 21
- Interactive Input Errors 270
- itrx 50
- itss 51

- J**
- Joint Definition 211
- Joint Definitions 67

- L**
- lcmx 51
- Line Definitions 73
- LLNL Octopus Specific Commands 229
- Load Curve Definitions 72
- Load Curve Display 227

- M**
- Magnification Commands 222
- MANUAL ORGANIZATION 10
- Material Detonation Points 79
- MATERIAL MODELS 21
- Material Specification 77
- Material Type 3
 - Kinematic / Isotropic Elastic-Plastic 88
- maxwell3 48
- MAZE Cross Section Definition 198
- MAZE Part Boundary Conditions 202
- MAZE Part Control 199
- Maze Parts 196
- MESH GENERATION COMMANDS 35, 145
- MESH GENERATION STRATEGIES 19
- Modifying Density and Viscosity 78
- MONT3D Analysis Options 59
- MONT3D Examples 307
- monte 48
- Multiple Surface Projections 174

- N**
- nast 48
- NIKE3D Analysis Options 54

- NIKE3D Examples 299
- NIKE3D Material Type 1
 - Elastic 121
- NIKE3D Material Type 2
 - Orthotropic Elastic 122
- NIKE3D Material Type 3
 - Elastic-Plastic 123
- NIKE3D Material Type 4
 - Thermo-Elastic-Plastic 123
- NIKE3D Material Type 5
 - Soil and Crushable Foam 125
- NIKE3D Material Type 6
 - Viscoelastic 126
- NIKE3D Material Type 7
 - Thermo-Orthotropic Elastic 126
- NIKE3D Material Type 8
 - Thermo-Elastic-Creep 128
- NIKE3D Materials 121
- nk2d 48
- nk3d 48
- nk3d84 48
- nk3d89 48
- Node and Element Identification 225
- NOMENCLATURE AND CONVENTIONS 12
- Non-Graphical Diagnostic Information Display 266
- Non-Graphical Diagnostics Commands 228

- O**
- Output Commands. 228
- Output Generation 269
- OVERVIEW 11

- P**
- pafe 48
- Panning and Zooming Commands 244
- Part and Materials Display Commands 223
- Part Tolerancing 135, 256
- Part Tolerancing Commands 226, 227
- PART TYPES 18
- Perspective 237
- poly3d 49

- R**
- Restoring the Display 253

- rigid 49
- Rigid Body Motion 71
- Rotation Commands 221
- Rotational Commands 242

- S**
- sap4 49
- Scaling 247
- Scaling Commands 222
- Shell Materials 78
- Slide Surface Specification 136, 212
- Sliding Interface Definitions 203
- Sliding Interface Display Commands 224
- Sliding Interfaces Commands 183
- Solid Mechanics Boundary Conditions 186, 204
- Spring - Mass - Damper Systems 69
- Standard Part 161
- Standard Part Boundary Conditions 181
- Standard Part Control 164
- Surface Definitions 138
- Surface Projections 171
- Symmetry Plane, Stone Wall and Non-Reflecting Boundary Identification 224

- T**
- Thermal Conditions 71
- Thermal Loads and Boundary Conditions 189, 205
- TOPAZ3D Analysis Options 56
- TOPAZ3D Electrostatic Material Type 1
 - Isotropic 131
- TOPAZ3D Examples 302
- TOPAZ3D Material Type 1
 - Isotropic 129
- TOPAZ3D Material Type 2
 - Orthotropic 130
- TOPAZ3D Material Type 3
 - Isotropic Temperature Dependent 130
- TOPAZ3D Material Type 4
 - Orthotropic Temperature Dependent 131
- TOPAZ3D materials 129
- Translational Commands 221
- tz2d 49
- tz3d 49

- U**
- UNITS 13

- V**
- Verbatim Materials 134
- Viewing Line information 254
- Volume Definitions 144

- Z**
- Zooming and Panning Commands 222