

FEAP - - A Finite Element Analysis Program

Version 7.4 Example Manual

Robert L. Taylor
Department of Civil and Environmental Engineering
University of California at Berkeley
Berkeley, California 94720-1710
E-Mail: rlt@ce.berkeley.edu

October 2001

Contents

1	Introduction	1
2	Example 1. Patch test	2
3	Example 2. Axisymmetric patch test	8
4	Example 2. Truss problem	14
5	Example 3. Circular disk	17
6	Example 4. Strip with hole and slit	29
7	Example 5. Thermal Problem	36
8	Example 6. Coupled Thermo-mechanical	41
9	Example 7. Contact Problem	46

Chapter 1

Introduction

In this manual we provide some examples of problems which can be set up and solved using the *FEAP* program. We begin by describing some of the methods which may be used to define an input data file for some simple finite element analyses. The manual is organized to start with very basic methods for inputs and proceeds to more general methods to describe input data and problem solutions.

FEAP is controlled using a set of *commands*. Each command performs a basic step in either describing a problem or solving a problem. Commands are divided into three basic groups:

1. Mesh description commands;
2. Problem solution commands; and
3. Graphical display commands.

The appendices of the *User Manual* contain the optional forms which each input command may have.

It is suggested that new users of *FEAP* carefully read this manual in its entirety before starting to generate their own input data. The later examples provide ways to manage the data for problems in separate files using an include option. Also, the form of the data input records may be constructed using *parameters* to which numerical values may be assigned either in numeric or expression form (see also Chapter 4 of the *User Manual*).

Chapter 2

Example 1. Patch Test.

The first problem considered is a simple patch test in which all the data is specified explicitly. Later examples will illustrate how *FEAP* can add missing data or use repeating similar parts to generate a mesh.

The *patch test* is a simple test which should be performed when first using any finite element program (see Chapter 11, Zienkiewicz and Taylor, 4th ed. Vol 1, for description of patch test). The patch test both ensures that the theory for the finite element formulation has been correctly implemented and that installation of the analysis system is also correct. For a problem in linear elasticity one patch test is uniform stress in the x direction. This condition may be imposed on a square region divided into 4-elements and loaded by a constant x -traction on the right side as shown in Figure 2.1 (node 6 is located at $y = 5$ units). For a square with side lengths 10 and a 10 unit per length traction, the nodal forces on the boundary are 2.5 at the corners and 5.0 at the midside. The left side of the mesh is restrained so that no x -direction displacements occur and the lower left corner is also restrained in the y -direction. These restraints prevent rigid body motions as well as ensure that a correct solution to the constant stress problem may be obtained.

For a linear elastic problem with the isotropic properties: Young's modulus 1000 and Poisson ratio 0.25, the plane strain solution has displacements

$$u = 0.009375 x \quad ; \quad v = -0.003125 y$$

and stresses

$$\sigma_{xx} = 1.0 \quad ; \quad \sigma_{yy} = 0.0 \quad ; \quad \sigma_{xy} = 0.0 \quad ; \quad \sigma_{zz} = 0.25$$

The input data for each problem to be solved by *FEAP* is prepared and placed in a file on disk. It is recommended that the filename for the input data have a first character of **I**. The filename should not exceed 14 characters (12 in PC mode, of which

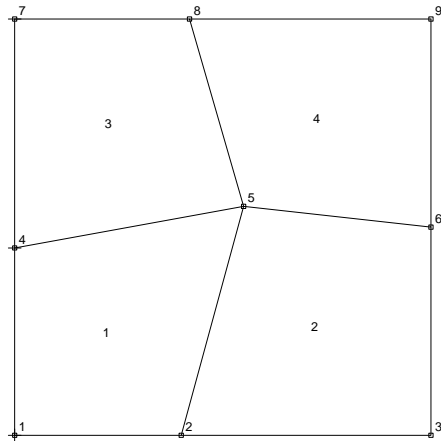


Figure 2.1: Patch Test Mesh

the last 4 may be used for an extender **.xxx**). When using the program on a PC it is recommended that no extender be used for the main data input file. Feap uses this file for generating other files which do contain extenders and, thus, an error could occur.

The filename for the input data of the patch test will be called **Ipatch** for the discussion below. When *FEAP* is run the names for other files will be assigned by replacing the first character (i.e., the **I**) by one indicating the type of file. For example, the file containing the description of the mesh and solution results is called the *output* file and for the above choice for the name of the input data file will be named **Opatch**. The complete input data file for the patch test problem is shown in Table 2.1 and a description for each part of this data follows.

Input records for *FEAP* are free format. Each data item is separated by a comma and/or blank characters. If blank characters are used without commas, each data item **must** be included. That is multiple blank fields are not considered to be a zero. Each data item is restricted to 14 characters (15 including the blank or comma). Comments may be appended to any data record after the character **!** (e.g., see Table 2.1).

The input file **Ipatch** consists of several *data sets*. For the patch test mesh given in Table 2.1 the data sets are given by the commands (shown without indentation in the table):

```
FEAP * * 4-Element Patch Test
MATERial
COORdinates
ELEMeNts
BOUNdary restraints
FORCes
END
BATCh
```

```

FEAP * * 4-Element Patch Test
  9,4,1,2,2,4
MATERial,1
  SOLId
    PLANe STRAIn
      ELAStic ISOTropic 1000.0 0.25
        ! Blank termination record
COORDinates
  1 0 0.0 0.0
  2 0 4.0 0.0
  3 0 10.0 0.0
  4 0 0.0 4.5
  5 0 5.5 5.5
  6 0 10.0 5.0
  7 0 0.0 10.0
  8 0 4.2 10.0
  9 0 10.0 10.0
        ! Blank termination record
ELEMents
  1 1 1 1 2 5 4
  2 1 1 2 3 6 5
  3 1 1 4 5 8 7
  4 1 1 5 6 9 8
        ! Blank termination record
BOUNDary restraints
  1 0 1 1
  4 0 1 0
  7 0 1 0
        ! Blank termination record
FORCes
  3 0 2.5 0.0
  6 0 5.0 0.0
  9 0 2.5 0.0
        ! Blank termination record
END

```

Table 2.1: Data for Patch Test

```

BATCh
  FORM residual
  TANGent
  SOLVe
  DISPlacement ALL
  STREss ALL
END
STOP

```

Table 2.2: Data for Patch Test

END
STOP

FEAP interprets only the first four characters of each command. These are shown as upper case letters to indicate the minimum amount which may be used to identify each command. Either upper case or lower case letters may be used to identify each command. Thus, **MATE** or **mate** may be used to identify the material property data sets. After a **FEAP** command and the control record, the commands before the first **END** may be in any order and define the mesh for the problem. The commands after the **BATCH** describe the solution algorithm for the problem and terminate with the second **END** command. Finally, the **STOP** command informs *FEAP* that no more data exists. *Only one problem may appear in the file which contains the problem initiation command FEAP.* See Example 4 in Chapter 6 for a way to run multiple problems.

The control record defines the size of finite element problem to be solved. The first field defines the number of nodes (**NUMNP**), the next field is the number of elements (**NUMEL**), followed by the number of material sets (**NUMMAT**), the spatial dimension for the mesh (**NDM**), the number of degrees of freedom for each node (**NDF**), and the maximum number of nodes on any element (**NEN**). The patch test has a mesh with 9 nodes, 4 elements, and 1 material set. The problem is 2 dimensional, has 2 degrees of freedom at each node, and each element has 4 nodes.

The first data set is identified by the **MATER**ial command. This record also must contain a material set number (ranging from 1 to the maximum number of sets needed). The next records consist of commands which describe the type of element (see Chapter 6 of the *User Manual* for the types of elements included with *FEAP*) and the material parameters associated with the set. The data shown in Table 2.1 indicates a **SOLID** (continuum) element is used, the problem is *plain strain* and the material parameters are associated with a *linear elastic isotropic material*. Except for the element type record, other data may be in any order and terminates with a blank record (comments are permitted on records and begin with the exclamation point, "!").

The values of the nodal coordinates for the patch are specified using the **COORD**inate command. Each record defines a node number, a generation parameter, and the x and y coordinate values. Nodes may be in any order, but are shown in increasing order in Table 2.1. Input terminates with a blank record.

The manner in which nodes are connected to form individual finite elements and their association to the type of element and material parameters is described by the data following the **ELEM**ent command. Each record defines the element number (which *must* be in increasing numerical order), a generation parameter (to be described later), the material data set associated with the element, and the list of nodes connected to the element. For the elements shown in Figure 2.1 the node sequence must start with a node at one vertex and then proceed with the nodes on vertices traversed counter

clockwise around the element. Input terminates with a blank record.

The degree of freedoms for each node may have known applied loads (nodal forces) or may be restrained to satisfy specified nodal displacements. In *FEAP* all degree of freedoms are assumed to have specified loading applied unless a restraint code is set. The **BOUND**ary restraint command may be used to assign restraints to degree of freedoms which are to have specified displacements. Each record defines a node number, a generation parameter, and the restraint codes for each degree of freedom associated with the node. A non-zero value for the restraint code indicates that the associated degree of freedom must satisfy a specified nodal displacement value (default is zero); whereas, a zero restraint code indicates the associated degree of freedom has a specified nodal force (also zero by default). Thus, for the data shown in Table 2.1, node 1 has both the u and v displacements restrained; nodes 4 and 7 have the u displacement restrained and the y force specified. All other nodes have both the x and the y forces specified since no restraints are specified. Input terminates with a blank record.

It is evident from the remaining data that no data is provided to impose non-zero displacements (methods to input non-zero values are described in Section 5.7 of the *User Manual*); however, data is given to impose non-zero forces using the **FORCE** command. Each force record defines a node number, a generation parameter, and the force values for each degree of freedom associated with the node. Thus, for the data given in Table 2.1, nodes 3 and 9 have x force values of 2.5 units and node 6 has an x force value of 5.0 units. Input terminates with a blank record.

The final command after the force values is the **END** command which terminates input of the data describing the finite element mesh.

The set of commands shown in Table 2.2 define the solution algorithm to be used in solving the problem. The execution is initiated by the **BATCH** command (alternatively, it is possible to perform an interactive execution where users enter each command as needed, see next example and Chapter 11 of the *User Manual*). The **FORM** command instructs *FEAP* to form the residual for the equilibrium equations written as:

$$\mathbf{R}(\mathbf{u}) = \mathbf{F} - \mathbf{P}(\mathbf{u})$$

where \mathbf{F} is the vector of applied nodal forces, \mathbf{u} is the vector of nodal displacements, and for a static linear elastic problem \mathbf{P} is defined as

$$\mathbf{P} = \mathbf{K} \mathbf{u}$$

in which \mathbf{K} is the stiffness matrix. A solution to the problem is defined by requiring the residual to be zero. In *FEAP* the solution may be computed using Newton's method which solves a sequence of linear problems. Thus, the **TANG**ent command requests the tangent matrix to the residual about the current solution state, \mathbf{u} (at start of execution the value is zero) where the tangent matrix is defined as:

$$\mathbf{K}_{tang} = - \frac{\partial \mathbf{R}}{\partial \mathbf{u}}$$

Thus, for a linear elastic static problem the tangent matrix is just \mathbf{K} . The **SOLVE** command instructs *FEAP* to solve the incremental (Newton) equations

$$\mathbf{K}_{tang} \Delta \mathbf{u} = \mathbf{R}$$

and to update the solution as

$$\mathbf{u} \leftarrow \mathbf{u} + \Delta \mathbf{u}$$

For a linear problem this solution sequence should converge in one iteration; thus, the residual would be zero if the command **FORM** is given again.

At this point *FEAP* has computed the solution; however, it is necessary to issue additional commands to output the values for each type of solution quantity. The **DISPLACEMENT** command instructs *FEAP* to output values for the nodal generalized displacement parameters associated with each nodal degree of freedom (for linear elasticity using solid elements these are the values of the u and v displacements at a node). The option **ALL** requests the displacement values for all active nodes (see manual pages in Appendix A for other options). Similarly, the command **STRESS, ALL** requests output values for stresses within all the active elements (see Appendix A for other options). All output is placed in the output file.

Chapter 3

Example 2. Axisymmetric patch test

A three dimensional solid is described by a body of revolution and is to be analyzed for the case of *axisymmetric* behavior. Only small deformations are considered.

For this case the displacement field is given by:

$$\mathbf{u} = \begin{Bmatrix} u_r(r, z) \\ u_z(r, z) \end{Bmatrix} \quad (3.1)$$

That is, the displacements do not depend on the θ coordinate. The stress and strain for an axisymmetric case may be ordered as:

$$\boldsymbol{\sigma} = [\sigma_{rr} \quad \sigma_{zz} \quad \sigma_{\theta\theta} \quad \sigma_{rz}]^T \quad (3.2)$$

and

$$\boldsymbol{\epsilon} = [\epsilon_{rr} \quad \epsilon_{zz} \quad \epsilon_{\theta\theta} \quad \gamma_{rz}]^T \quad (3.3)$$

where $\gamma_{ij} = 2\epsilon_{ij}$. Using the displacements given by Eq. 3.1 the strain-displacement relations are given by (Reference: I.S. Sokolnikoff, *Mathematical Theory of Elasticity*, McGraw-Hill, New York, 1956, pp 182-184.)

$$\boldsymbol{\epsilon} = \left[\frac{\partial u_r}{\partial r}, \quad \frac{\partial u_z}{\partial z}, \quad \frac{u_r}{r}, \quad \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right]^T \quad (3.4)$$

Similarly, for the above displacement field the equations of equilibrium are given by:

$$\begin{aligned} \frac{\partial \sigma_{rr}}{\partial r} + \frac{\partial \sigma_{rz}}{\partial z} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} + b_r &= 0 \\ \frac{\partial \sigma_{rz}}{\partial r} + \frac{\sigma_{rz}}{r} + \frac{\partial \sigma_{zz}}{\partial z} + b_z &= 0 \end{aligned} \quad (3.5)$$

in which body forces also are axisymmetric and thus are given by $\mathbf{b} = \mathbf{b}(r, z)$.

For a displacement formulation, a variational problem for linear elasticity may be written as

$$\Pi(\mathbf{u}) = 2\pi \left[\int_{\Omega} W(\boldsymbol{\epsilon}) r dr dz - \int_{\Omega} \mathbf{u}^T \mathbf{b} r dr dz - \int_{\Gamma_t} \mathbf{u}^T \bar{\mathbf{t}} r ds \right] = \min. \quad (3.6)$$

where W is the stored energy function and ds is a differential length on the two dimensional $r - z$ boundary.

Introducing an isoparametric interpolation for the finite element model as

$$\begin{aligned} r &= N_{\alpha}(\boldsymbol{\xi}) r^{\alpha} \\ z &= N_{\alpha}(\boldsymbol{\xi}) z^{\alpha} \\ u_r &= N_{\alpha}(\boldsymbol{\xi}) u_r^{\alpha} \\ u_z &= N_{\alpha}(\boldsymbol{\xi}) u_z^{\alpha} \end{aligned}$$

where $\boldsymbol{\xi}$ are natural coordinates, the strain-displacement relations become

$$\boldsymbol{\epsilon} = \begin{bmatrix} \frac{\partial N_{\alpha}}{\partial r} & 0 \\ 0 & \frac{\partial N_{\alpha}}{\partial z} \\ \frac{N_{\alpha}}{r} & 0 \\ \frac{\partial N_{\alpha}}{\partial z} & \frac{\partial N_{\alpha}}{\partial r} \end{bmatrix} \begin{Bmatrix} u_r^{\alpha} \\ u_z^{\alpha} \\ u_{\theta}^{\alpha} \end{Bmatrix}$$

The variational problem generates the residual equation

$$\mathbf{R}_{\alpha} = \mathbf{F}_{\alpha} - 2\pi \int_{\Omega} \mathbf{B}_{\alpha}^T \boldsymbol{\sigma} r dr dz$$

in which nodal forces are given by

$$\mathbf{F}_{\alpha} = 2\pi \left[\int_{\Omega} N_{\alpha} \mathbf{b} r dr dz + \int_{\Gamma_t} N_{\alpha} \bar{\mathbf{t}} r ds \right]$$

Linearizing generates the tangent stiffness used in solution of problems by a Newton method. Accordingly, after the linearization step we obtain

$$\mathbf{K}_{\alpha\beta} = 2\pi \int_{\Omega} \mathbf{B}_{\alpha}^T \mathbf{D}_T \mathbf{B}_{\beta} r dr dz$$

as the stiffness matrix in which \mathbf{D}_T is the matrix of tangent moduli for each material.

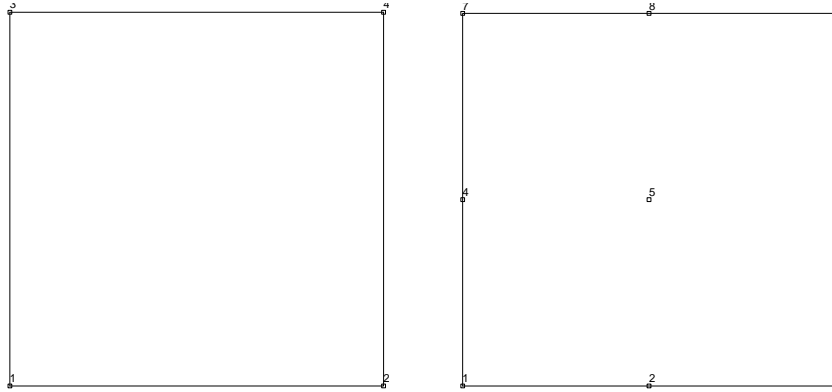


Figure 3.1: (a) 4-node element (b) 9-node element.

The patch test satisfaction requires two parts: A stability test and a consistency test. Here the stability is assessed by computing the eigenvalues for a single element.

An eigenpair assessment is performed for the 4-node element shown in Fig. 1(a). The command language statements needed are:

```
BATCh
  TANG,,-1 ! No factoring performed
  EIGE      ! Eigenpair for last element
END
```

and results obtained are:

Eigenvalue			
1	2	3	4
1.5094E+04	6.2967E+03	5.9726E+03	4.1488E+03
4.1332E+03	8.7238E+02	1.6841E+02	-4.8317E-13

Eigenvectors			
1	2	3	4
2.8467E-01	-1.4779E-01	3.6123E-01	5.7917E-01
2.7360E-01	-3.0785E-01	-3.1771E-01	1.5711E-01
-4.2180E-01	-5.3722E-01	-3.4016E-01	-3.3940E-01
4.0766E-01	3.0785E-01	-3.9098E-01	-1.5711E-01
-4.2180E-01	5.3722E-01	-3.4016E-01	3.3940E-01
-4.0766E-01	3.0785E-01	3.9098E-01	-1.5711E-01
2.8467E-01	1.4779E-01	3.6123E-01	-5.7917E-01
-2.7360E-01	-3.0785E-01	3.1771E-01	1.5711E-01

Eigenvectors

5	6	7	8
2.2696E-02	-5.3662E-01	3.7778E-01	7.8404E-15
-5.6183E-01	-9.2489E-02	-3.6131E-01	5.0000E-01
6.1055E-02	-4.5016E-01	3.1016E-01	6.4563E-15
4.2439E-01	-2.8981E-02	3.6131E-01	5.0000E-01
6.1055E-02	-4.5016E-01	-3.1016E-01	-5.9879E-15
-4.2439E-01	2.8981E-02	3.6131E-01	5.0000E-01
2.2696E-02	-5.3662E-01	-3.7778E-01	-7.1650E-15
5.6183E-01	9.2489E-02	-3.6131E-01	5.0000E-01

The test is repeated for the mesh shown in Fig. 1(b) and the resulting eigenvalues are (vectors are not included here)

Eigenvalue			
1	2	3	4
5.1049E+04	4.9787E+04	2.3214E+04	2.0397E+04
1.2530E+04	1.2488E+04	8.1176E+03	7.0663E+03
6.2949E+03	5.8856E+03	4.3601E+03	3.1329E+03
2.3443E+03	2.1822E+03	1.3440E+03	3.6688E+02
9.5560E+01	-6.3612E-12		

In both cases there are two *rigid body modes*: a z -translation and a free rotation in the θ -direction. That is displacements which cause no strain are restricted to

$$u_z = A \quad \text{and} \quad u_\theta = B r$$

where A and B are constants.

For a part of the consistency test we now consider the test applied to a displacement field expressed as:

$$u_r = A + B r \quad \text{and} \quad u_z = 0$$

we obtain strains

$$\epsilon = \begin{pmatrix} B \\ 0 \\ \frac{A}{r} + B \\ 0 \end{pmatrix}$$

and for an isotropic material with constitution

$$\sigma_{ij} = \lambda \varepsilon_v \delta_{ij} + 2 \mu \epsilon_{ij}$$

where $\varepsilon_v = \varepsilon_{ij}$ is the volume change in small deformation, the stresses

$$\boldsymbol{\sigma} = \begin{pmatrix} \lambda \left(\frac{A}{r} + 2B \right) + 2\mu B \\ \lambda \left(\frac{A}{r} + 2B \right) \\ \lambda \left(\frac{A}{r} + 2B \right) + 2\mu \left(\frac{A}{r} + B \right) \\ 0 \end{pmatrix}$$

Inserting these stresses into the equilibrium equation requires a non-zero radial body force to be:

$$b_r = (\lambda + 2\mu) \frac{A}{r^2}$$

This body force distribution *must* be coded into each element to permit the testing of a patch performance.

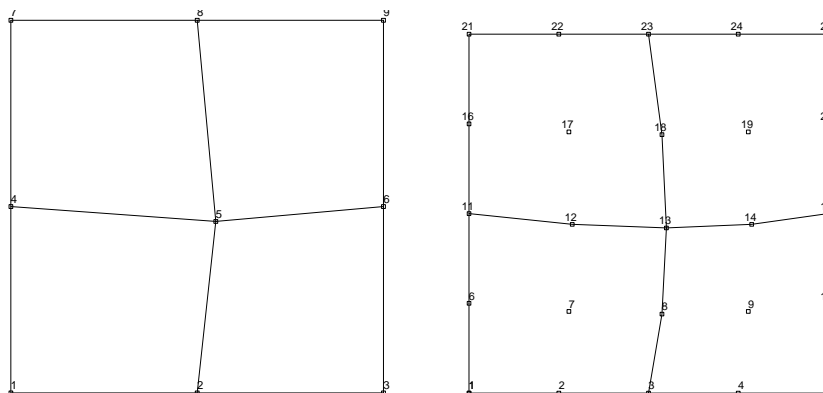


Figure 3.2: (a) 4-node mesh (b) 9-node mesh.

A rectangular region between $5 \leq r \leq 10$ and $0 \leq z \leq 5$ is used (Note that it is important not to use a patch which includes a zero radius as no radial displacement is allowed there). The meshes shown in Fig. 2 are used for the patch tests and results for the 9-node mesh are shown in Fig. 3 (results for a 4-node element are very similar). Based on the above assessment we find that the element, as coded, passes all the patch test requirements for stability and consistency.

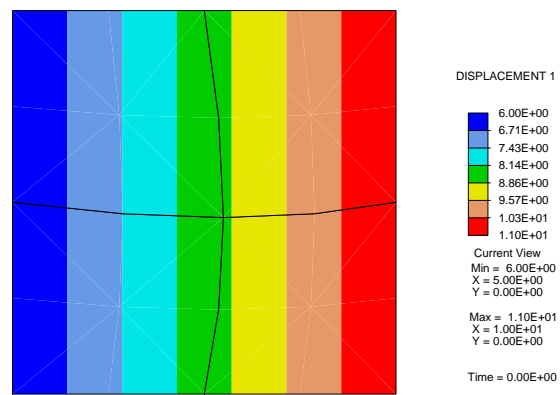


Figure 3.3: 9-node radial displacement contour.

Chapter 4

Example 2. Truss Problem.

As a next example for the use of *FEAP*, consider a simple truss problem. The mesh, nodal and element numbers, loading, restraints, and material properties are shown in Figure 4.1.

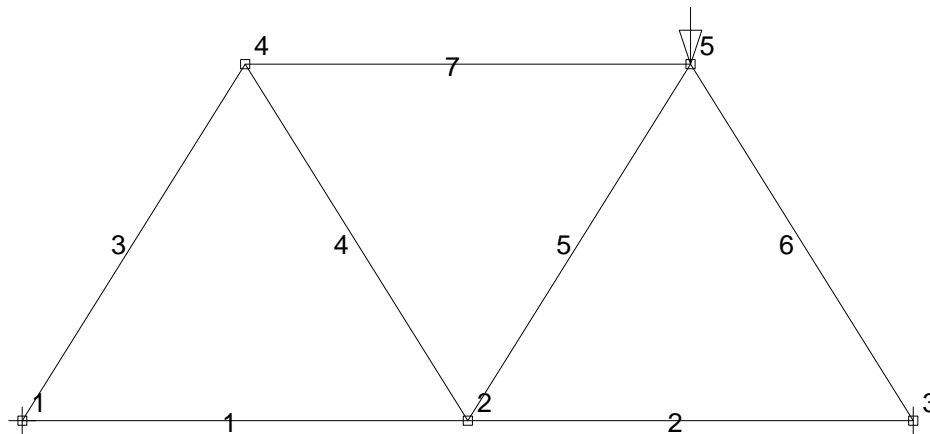


Figure 4.1: Mesh for Truss Example

The complete data file for the problem is shown in Table 4.1 and a description for each part of this data follows.

The control record indicates the mesh has 5 nodes, 7 elements, 2 material sets. It is a 2 dimensional problem with 2 degrees of freedom at each node and 2 nodes for each truss element.

The **MATE**rial property sets for all members are identical except for the cross sectional area of the members (the first numerical field on the **CROSS SECTION** record). Two types are indicated: Set 1 has an area of 10 units while set 2 has an area of 5 units.

The coordinates are input as for the patch example described above. Element properties

```

FEAP * * 2-D Truss Problem
  5 7 2 2 2 2
MATERial,1
  TRUSS
    ELAStic ISOTropic 1000.0
    CROSS SECTIon    10.0
                                ! Blank termination record
MATERial,2
  TRUSS
    ELAStic ISOTropic 1000.0
    CROSS SECTIon    5.0
                                ! Blank termination record
COORDinates
  1 0 0.0 0.0
  2 0 200.0 0.0
  3 0 400.0 0.0
  4 0 100.0 160.0
  5 0 300.0 160.0
                                ! Blank termination record
ELEMents
  1 0 1 1 2
  2 0 1 2 3
  3 0 1 1 4
  4 0 2 2 4
  5 0 2 2 5
  6 0 1 3 5
  7 0 1 4 5
                                ! Blank termination record
BOUNDary restraints
  1 0 1 1
  3 0 0 1
                                ! Blank termination record
FORCe
  5 0 0.0 -10.0
                                ! Blank termination record
END
INTERactive
STOP

```

Table 4.1: Data for Truss Analysis Problem

are also input in a similar way; however, note that the material property set in the third field now is set to either 1 or 2 depending on whether the cross section has 5 or 10 units of area. Boundary restraint conditions are imposed so that node 1 is restrained to have zero u and v displacements while node 3 has only a restrained v displacement. Finally, a single load in the vertical direction with magnitude -10.0 is applied to node 5.

This problem requests an **INTE**ractive mode of solution. In the interactive mode users must give the solution commands needed for each solution step. For example, when the command **FORM** is given *FEAP* will compute the residual and then prompt for another command input. Similarly, giving the commands for output will display the request on the screen and also place the information in the output file.

Chapter 5

Example 3. Circular Disk Subjected to Point Loading.

The next problem considered is a circular disk loaded by two concentrated forces of 10 units each directed along a diagonal (see Figure 5.1). The material of the disk is assumed to be linearly elastic. Furthermore, for simplicity we consider the loads to be slowly applied so that inertial effects may be ignored. Thus, the model to be solved is a simple linear elastostatics problem. Since the loading is symmetric and we assume the material to be isotropic (and thus also symmetric), it is only necessary to construct a mesh for one quadrant of the circular disk.

This problem has curved boundaries and requires a general mesh to define the finite element solution, thus, more details are described for the input data options available in *FEAP*.

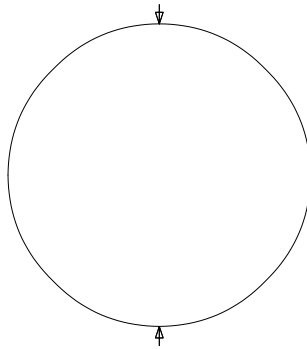


Figure 5.1: Circular Disk with Point Loading

In the two dimensional capabilities included in *FEAP*, the solid elements permit an analyst to use elements with between 3 and 9 attached nodes. A 3-node element is a plane triangle with nodes located at each vertex. A 4-node element is a quadrilateral

with nodes at each vertex. A 9-node element is quadrilateral in shape but may have curved sides defined by nodes located in the mid part of each edge, as well as one additional node in the interior. Omitting some midside nodes and/or the center node produces quadrilateral elements with between 5 and 8 nodes.

Let us assume that a mesh will be constructed using 4-node quadrilateral elements. The nodes are defined by a sequence starting with *Node 1* and concluding with the maximum number *Node NUMNP*. The elements also are defined by a sequence starting with *Element 1* and concluding with *Element NUMEL*. A 4-node quadrilateral element is defined by the node numbers associated with each vertex. A simple mesh for one quadrant of the disk is shown in Figure 4.1. The figure shows the numbers associated with each node and element. This mesh has 19 nodes ($NUMNP = 19$) and 12 elements ($NUMEL = 12$).

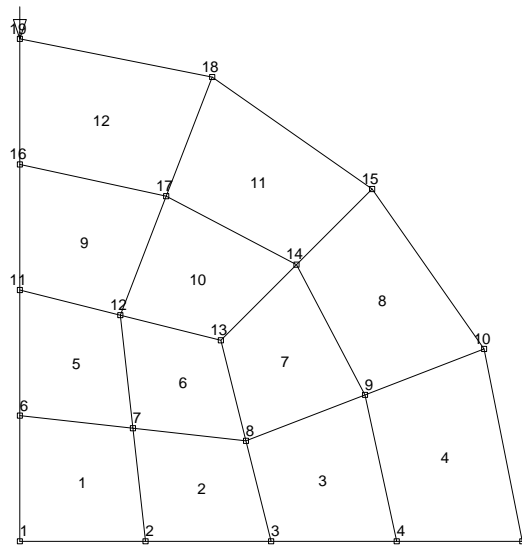


Figure 5.2: Finite Element Mesh for Circular Disk with Point Loading

The control data for the mesh shown in Figure 5.2 is:

```
FEAP * * Example 1. Circular Disk: Basic inputs
19 12 1 2 2 4
```

The remainder of the finite element mesh is described using the data set concepts introduced in the previous examples.

There are options to specify the number and coordinates for each nodal point. The one used to this point is the **COORDinate** option. Since only the first 4 characters of each command are interpreted by *FEAP*, the use of **COOR** or **COORDINATE** produces identical results. After the **COOR** command individual records defining each nodal point and its coordinates (in the present case the x_1 and x_2 coordinates) are specified as:

N, NG, X-1, X-2

where

N Number of nodal point.
 NG Generation increment to next node.
 X-1 value of x-1 coordinate.
 X-2 value of x-2 coordinate.

Thus, for the mesh shown in Figure 4.1, the coordinate data may be specified as:

```
COORdinateS
 1  1  0.0000      0.0000
 5  0  1.0000      0.0000
 6  1  0.0000      0.2500
 8  1  0.4500      0.2000
10  0  0.9239      0.3827
11  1  0.0000      0.5000
13  1  0.4000      0.4000
15  0  0.7010      0.7010
16  0  0.0000      0.7500
17  0  0.2913      0.6869
18  0  0.3827      0.9239
19  0  0.0000      1.0000
      ! Blank termination record
```

The missing node numbers and their coordinate values are generated using linear interpolation on the NG generation sequence given. Thus, the first pair of records also generates nodes 2 to 4 with coordinates:

```
 2  0.2500      0.0000
 3  0.5000      0.0000
 4  0.7500      0.0000
```

Comments may be appended after the second character of any line by using an exclamation followed by the text. Other options to define coordinates are discussed later.

There are also different options which may be used to generate the element connection data. One is the **ELEMent** command which is given as:

N, NG, MA, N-1, N-2, N-3, N-4

where

N	Number of element.
NG	Generation increment for node numbers.
MA	Material identifier associated with element.
N-1	Node number for first vertex.
N-2	Node number for second vertex.
N-3	Node number for third vertex.
N-4	Node number for fourth vertex.

Any vertex of the quadrilateral may be used to define N-1. The remainder, however, should be specified using a counter clockwise sequencing around the nodes as shown in Figure 5.3.

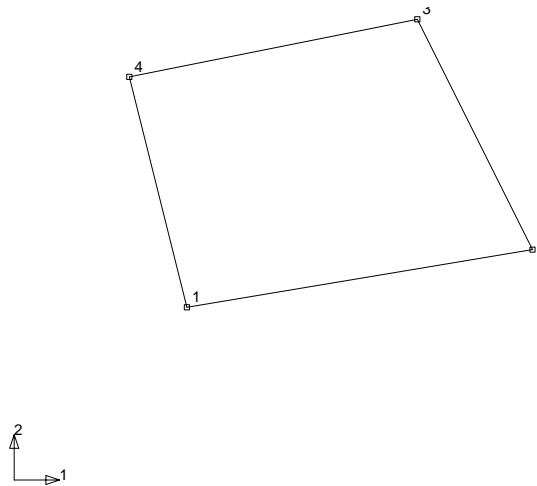


Figure 5.3: Node Sequencing for 4-node Quadrilateral

The element connection data for the example problem is given by:

```

ELEMents
 1 1 1 1 2 7 6
 5 1 1 6 7 12 11
 9 1 1 11 12 17 16
10 1 1 12 13 14 17
11 1 1 14 15 18 17
12 0 1 16 17 18 19
      ! Blank termination record

```

Elements within a data set must be in order; however, gaps may occur and missing elements will be generated. The second field in each of the element records defines the increment to apply to nodes in order to generate missing elements. Thus, from the first two pairs of records it is evident that elements 2, 3, and 4 are missing. The generation

increment to apply to the nodes on element 2 is specified as 1 on the element 1 record¹. Accordingly, the generated elements will have the following sequence of nodes.

```

2      2  3  8  7
3      3  4  9  8
4      4  5 10  9

```

The material identifier number will be the values specified in the third field and for the data above will be 1 for all generated elements (the material identifier for generated elements is taken from the first record of each generation pair).

Multiple **ELEMent** sets may be used or **ELEMent** may be combined with other options (e.g., **BLOCK** type generations).

To complete the problem specification it is necessary to impose constraints on the nodal displacements along the symmetry axes, specify the applied concentrated load, and define the material set properties. A set of commands which accomplish this is given by

```

BOUNDary restraint codes
 1 1 1 -1
 5 0 0 1
 6 5 -1 0
19 0 1 0
      ! Blank termination record

FORCes on nodes
19 0 0 -5.0
      ! Blank termination record

MATERial,1
SOLId
ELAStic ISOTropic 10000 0.25 ! E and nu
DENSity data 0.1 ! rho
QUADrature data 2 2
      ! Blank termination record
END

```

The set of records following the **BOUNDary** code command imposes the necessary restraints on nodes. A non zero restraint code implies the value of the corresponding displacement component will have specified values (default values are 0.0); whereas, a zero code (the default restraint code value) indicates the component is unknown and must be computed. The inputs use generation, similar to the **COOR** data. Thus the

¹If the field is omitted a default value of 1 is assumed.

first pair of records will also generate restraints for nodes 2, 3, and 4. A negative or zero code will propagate to the generated node, whereas a positive code will become a zero values on the generated nodes. Thus the first pair will generate a set of restraints such that the first displacement component (u_1) will be an unknown and the second component (u_2) will be set to a specified value. The last pair of records also generates restraints for the first degree of freedom of nodes 11 and 16.

For degree of freedoms with zero restraint codes *FEAP* will add a force value to each node (default is 0.0); whereas, for non zero restraint codes *FEAP* will impose a displacement value (default is 0.0). Non zero forces may be specified using a **FORCe** command (other options also exist as described later). Non zero displacements may be specified using a **DISPlacement** command (again, other options exist). Thus, for the example problem a concentrated load ($F_2 = -5$) is applied on node 19 by the **FORCe** command set shown above. The **MATeRial** record specifies the set number as 1. The next record requests the *SOLId* element type (See Chapters 6 and 7 of the *User Manual* for additional information on types of elements and material models permitted). The third record defines the material constitution as isotropic linear elastic and sets the parameters as $E = 1000$, $\nu = 0.25$. The fourth record sets the material density as $\rho = 0.1$. The fifth record specifies a 2×2 Gauss quadrature to compute arrays and output element results. The input of material parameters terminates with a blank record. The data following the *SOLId* specification may be in any order. Also any data not needed may be omitted. Since the disk analysis is quasistatic the material density is unnecessary and could be omitted. Also default quadrature values will be used if this record is omitted. For the current analysis the minimum material data is:

```
MATeRial,1
  SOLId
  ELAStic ISOTropic 10000 0.25 ! E and nu
                                ! Blank termination record
```

The **END** command signals *FEAP* that the specification of the mesh and initial loading conditions is complete. A mesh may be modified during the solution phase by re-entering **MESH** generation or by manipulating the mesh to merge parts using the **TIE** command or setting boundary conditions to satisfy **LINK** or **ELINK** conditions.

We next describe some of the other options which are available to generate and/or manipulate a mesh. Again we consider the example problem to make the discussion specific.

The preceding format to generate the data for a finite element mesh for the example problem is quite restrictive. If it is desired to increase the number of nodes and elements it is necessary to restart the process from the very beginning. Thus, we now consider options for describing the mesh which permit the problem to be described more easily, as well as, permit the number of nodes and elements to be increased or decreased.

FEAP has some powerful options which permit the generation of problems in a form amenable to refinement. For example, the control record pair may be specified with 0 nodes, 0 elements, and/or 0 materials indicated. *FEAP* will use the subsequent data to compute the number of nodes, elements, and material sets in the mesh². Accordingly, it is possible to use:

```
FEAP * * Example 1.  Circular Disk: Block inputs
0 0 0 2 2 4
```

Without additional features this is of little merit. However, nodes and elements may be generated using a **BLOCK** command as:

```
PARAMeter
  m = 2
  n = 2
                                ! End of parameters
BLOCK 1
  CARTesian,m,n,1,1,1
  1  0.0  0.0
  2  0.5  0.0
  3  0.4  0.4
  4  0.0  0.5
                                ! Blank termination record
```

A **BLOCK** command uses a regular subdivision of the parent master element shown in Figure 5.2 to describe the mesh within its perimeter. In Figure 5.2 the coordinate directions 1 and 2 are local axes for the **BLOCK** generation (i.e., the natural coordinates for an isoparametric master element). Using this convention, **BLOCK 1** is described by a 4 node quadrilateral master element whose coordinates are specified as shown above. The first record states that the master element coordinates will be specified in cartesian form (other options are **POLAR** and **SPHERICAL**), the block will be divided into m subdivisions in the 1 direction and n subdivisions in the 2 direction (an $m \times n$ mesh of quadrilaterals), with the first node, element, and material set numbers set to 1. The values for m and n are assigned *prior* to the **BLOCK** specification using the **PARAMETER** command followed by the description of parameters. Each parameter may only be one character between **a** and **z** (*FEAP* converts all data to lower case internally, hence only 26 parameters are available for use at any one time. Parameters may, however, be redefined at later stages of the data.

Parameters for the sine and cosine of the angles at the middle of the circular part of blocks are defined by:

```
PARAMeters
  p = atan(1.)
```

²If user mesh functions are employed this feature may not work correctly

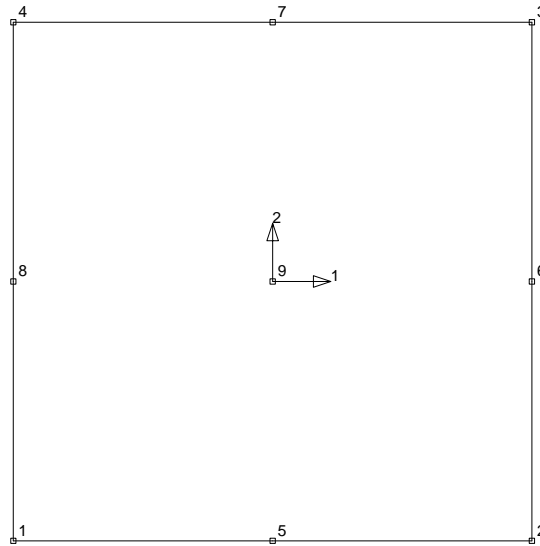


Figure 5.4: Node Sequence to Define Master Nodes on a Block

```

s = sin(0.5*p)
c = cos(0.5*p)
! Blank termination record

```

Alternatively, the inputs may be specified directly in degrees using the expressions

```

PARAMeters
s = sind(22.5)
c = cosd(22.5)
! Blank termination record

```

Consult Chapter 4 of the *User Manual* for a list of all functions which may be used in expressions.

The additional blocks needed to define the circular disk then may be given as:

```

BLOCK 2
CARTesian,n,n
1 0.5 0.0
2 1.0 0.0
3 0.701 0.701
4 0.4 0.4
6 c s
! Blank termination record

```

```

BLOCK 3
CARTesian,n,n
1 0.4 0.4

```

```

2 0.701 0.701
3 0.0 1.0
4 0.0 0.5
6 s c
! Blank termination record

```

The `PARAMeter` command also permits arithmetic calculations to be performed (see Chapter 4 of the *User Manual* for further information). In the above, the parameter `p` is set to 0.25π (i.e., 45 degrees) and `s` and `c` are the sin and cosine of 22.5 degrees. The second and third blocks are also 2×2 meshes of quadrilaterals since the value of `n` has not been changed; however, each of these blocks are described on an element with one edge curved along the circular boundary of the disk. Note also, that after defining the initial generation sequence for the node and element numbers on the first block all others have zero numbers. *FEAP* will be able to compute the number for the first node and element of each block so that the final mesh has 12 elements and 27 nodes (it is necessary to merge the blocks to produce a mesh with only 19 active nodes needed to define the problem).

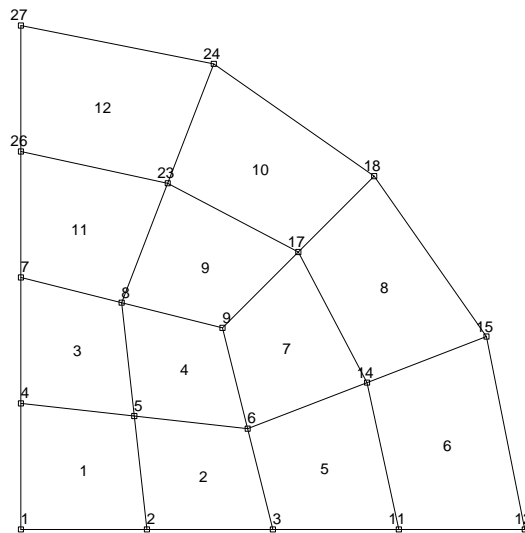


Figure 5.5: Finite Element Mesh for Circular Disk using Block Commands and Merging

The boundary conditions and applied nodal load may also be defined such that it is not necessary to know node numbers by using `EBOUndary` (for edge boundary specification) and `CFORce` commands (for coordinate point force loading), respectively. Accordingly,

```

EBOUndary      ! Edge boundary restraints
  1 0.0 1 0
  2 0.0 0 1
! Blank termination record
CFORce         ! Coordinate specified forces
NODE 0.0 1.0 0.0 -5.0

```

```

                                ! Blank termination record
MATERial,1
  SOLId
  ELAStic ISOTropic 10000 0.25
  DENSty data      0.10
  QUADrature data 2 2
                                ! Blank termination record
END
TIE                                ! Tie nodes with same coordinates.

```

accomplish these steps and complete the description for the mesh and its material properties. Note particularly, the TIE command following the generation of the mesh. This command will merge the three blocks to form a mesh which is equivalent to the original mesh (however, this mesh has different numbering for nodes and elements than the first mesh generated). The numbers for the active nodes after the merge and the element numbers produced by the block commands are shown in Figure 5.5. The advantage of this form, however, is the ease by which the mesh can be refined. Indeed by assigning the parameter n (and m) a value of 5, one obtains a mesh with 75 elements as shown in Figure 5.6.

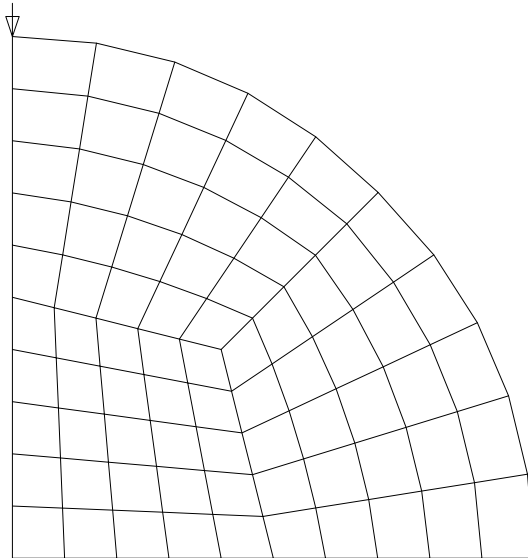


Figure 5.6: Mesh for Circular Disk. 75 Elements

Since the boundary conditions are associated with an edge, all nodes which lie on the edge will be identified and restrained. Similarly, the node with the specified coordinates will be identified for the applied $F_2 = -5.0$ force. Additional details on options for the CFORce command are given in the mesh user manual pages in Appendix A.

Once the mesh is described the problem may be solved using the solution command language. Two modes of execution are available: A batch mode where FEAP enters a solution mode and processes all data without user intervention; and an interactive

mode where the user issues each command or command group and receives prompts for additional data. An analysis may use multiple batch and/or interactive solutions in the same analysis. To enter a batch execution the user inserts a command `BATCh` after the mesh `END` record (and any other commands which manipulate the mesh, e.g., `TIE`), whereas to enter an interactive mode the command `INTERactive` is inserted. The `BATCh` execution command must be immediately followed by the other solution commands and terminates with an `END` command. To perform a batch mode static (steady state) solution for the example problem the commands

```
BATCh
  TANGent      ! Form tangent (K)
  FORM         ! Form residual (R)
  SOLVe        ! Solve  $K*du = R$ 
  END          ! End of batch execution
```

may be included in the input data file after the `TIE` command. Alternatively, the command

```
INTERactive
```

may be included and the other commands given sequentially after receiving the

```
Macro 1.>
```

prompt. With either mode *FEAP* process the sequence of commands to produce a solution; however, no report of the results is provided unless specifically requested. To output all the nodal displacement values for the current solution state the command

```
DISPlacement,ALL
```

may be given. Similarly, the commands

```
STREss,ALL
REACTION,ALL
```

produce outputs for all the element variables (stresses and any other values included in the element output section) and reactions for all the nodes. The order of output is controlled by the order of issuing the commands. If the solution is converged then nodal reactions should be zero (numerically) for all degree-of-freedoms which are not restrained. Restrained degree-of-freedoms report the reaction force necessary to impose the specified displacement constraint.

Alternatively, graphical outputs may be requested. For example, the commands

```
PLOT,MESH  
PLOT,LOAD,,-1
```

produce the results shown in Figure 5.6. Contour plots are also possible. Shaded contours for the vertical displacements are shown in Figure 5.7 and are obtained using the command

```
PLOT,CONT,2
```

Additional information on solution and plot commands is included in Chapters 11 and 12 of the *User Manual*, respectively.

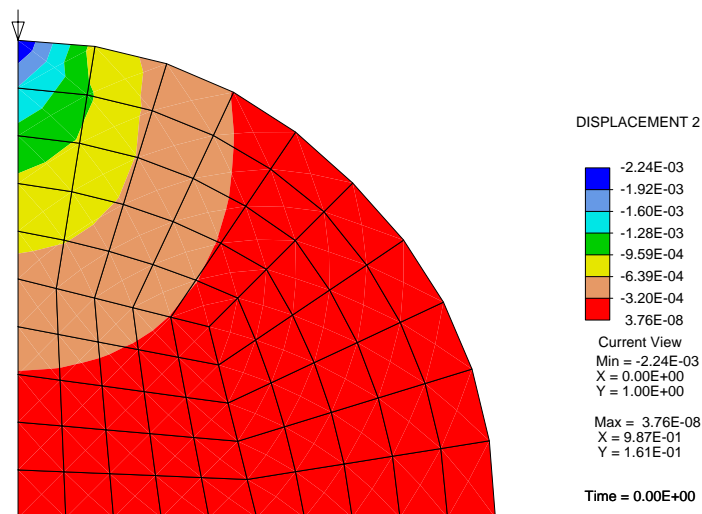


Figure 5.7: Contours of Vertical Displacement for Circular Disk

Chapter 6

Example 4. Strip with Hole and Slit.

As a next example we consider the analysis of a tension strip which contains a hole but has a slit between the hole and the right boundary as shown in Figure 6.1. The strip is to be loaded by applying vertical displacements along the top and bottom. The possibility of having different materials for the top and bottom halves is anticipated and, thus, the entire mesh needs to be constructed.

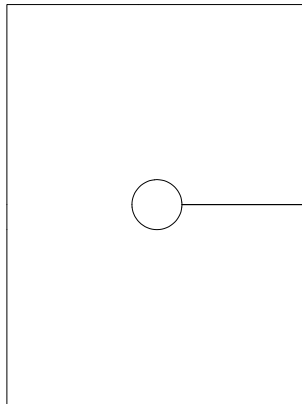


Figure 6.1: Tension Strip with Hole and Slit

To construct the mesh a set of 8 blocks of nodes and elements will be constructed and merged to form the final analysis. Since the mesh has a considerable amount of symmetry it is proposed to generate the two blocks for one quadrant using the `BLOCK` mesh command and then use the `TRANSformation` command to form the other quadrants. The steps to form the mesh may be summarized as follows.

1. Assign `REGIon` 1 as the first quadrant.

2. Use the **BLOCK** command to form the two blocks for the first quadrant. Save the mesh for the first quadrant in a file called **IHQQUAD**.
3. Use an **INCL**ude, **IHQQUAD** in a problem data file (e.g., file **ISTRIP**) to import the data for quadrant 1 (see Chapter 4 of the *User Manual* for more information on use of the include option).
4. Assign the second and third quadrants to **REGI**on 2.
5. Set the **TRANS**formation to reflect the x-axis and use an **INCL**ude, **tt IHQQUAD** to generate the 2 blocks for the second quadrant. Note that since the coordinate transformation is not a rotation the generated elements may have negative volume.
6. Set the **TRANS**formation to reflect the x-axis and y-axes. Use an **INCL**ude, **IHQQUAD** to generate the 2 blocks for the third quadrant.
7. Assign the fourth quadrants to **REGI**on 3.
8. Set the **TRANS**formation to reflect y-axes. Use an **INCL**ude, **IHQQUAD** to generate the 2 blocks for the fourth quadrant.

The commands to perform the transformations and read the include files are summarized in Figure 6.2.

The outlines for all the blocks formed after this step are shown in Figure 6.3. It is necessary now to merge these blocks to form the final mesh, while retaining the slit, which if not properly treated will be merged also during the use of a **TIE** command.

It is at this time that the utility of the **REGI**on descriptions is used. A summary of the merge order is as follows

1. Merge each of the regions with itself. The result of this step is shown in Figure 6.4.
2. Merge region 1 with region 2; also merge region 2 with region 3. This will produce the final mesh whose outline was shown in Figure 6.1.

The **TIE** commands to achieve these two steps are:

```
TIE,REGIOn,1,1
TIE,REGIOn,2,2
TIE,REGIOn,3,3
TIE,REGIOn,1,2
TIE,REGIOn,2,3
```

```

FEAP * * Tension Strip With Hole and Slit
0,0,0,2,2,4
PARAMeters
  d=1          ! First node number
  e=1          ! First element number
  m=1          ! Material set number
  n=8          ! Size of blocks
              ! Terminator
REGIon,1      ! Assigns 1st quadrant to region 1
  INCLude,IHQWAD ! Input first quadrant
PARAMeters
  d=0          ! To make feap count nodes
  e=0          ! To make feap count elements
              ! Terminator
REGIon,2      ! Assign 2nd and 3rd quadrant
  TRANSform    ! Reverse x axis for second quadrant
    -1,0,0
    0,1,0
    0,0,1
    0,0,0
  INCLude,IHQWAD
  TRANSform ! Reverse x,y axis for third quadrant
    -1,0,0
    0,-1,0
    0,0,1
    0,0,0
  INCLude,IHQWAD
REGIon,3      ! Assign 4th quadrant to region 3
  TRANSform ! Reverse y axis for fourth quadrant
    1,0,0
    0,-1,0
    0,0,1
    0,0,0
  INCLude,IHQWAD
END

```

Figure 6.2: Region, Transformation, and Include Structure

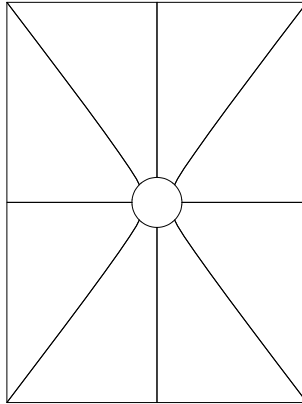


Figure 6.3: Tension Strip: Block Structure Before Merges

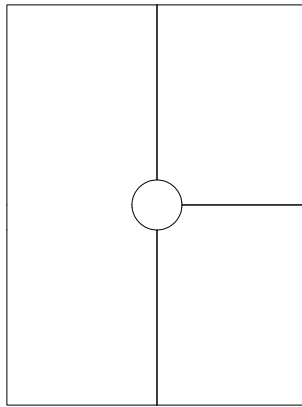


Figure 6.4: Tension Strip: After Merge of Each Region with Itself

The generation of the two blocks forming each quadrant is achieved using the commands shown in Figure 6.5.

Finally, the displacements on the top and bottom edges are specified using the commands given in Figure 6.6.

An analysis using this data produced the results for the σ_{yy} stress shown in Figure 6.

Since no explicit use of node element numbers appears in any of the data it is possible to run a number of problems using the same data but different values for the mesh generation parameter **n**. Indeed all problems may be performed during the same execution of the program by defining a master input file, say `ISLIT` which has the values for the parameters. The file `ISLIT` can contain the data shown in Fig. 6.8. It is necessary to remove the definition of the parameter **n** from the `ISTRIP` file. It is further assumed that the solution commands are also contained in the `ISTRIP` file or in a file which is referenced by an include file. Finally, no **STOP** command should be referenced by the `ISTRIP` file. *FEAP* will perform all three problems and place the output sequentially

```

PARAM
  c=cosd(45.0)
  s=sind(45.0)
  a=cosd(22.5)
  b=sind(22.5)
      ! Termination
BLOCK
  CART,n,n,d,e,m
  1,1,0
  2,6,0
  3,6,8
  4,s,c
  5,2.6,0
  7,2.1,2.8
  8,a,b
  9,2.5,1.2
      ! Termination
BLOCK
  CART,n,n,0,0,m
  1,s,c
  2,6,8
  3,0,8
  4,0,1
  5,2.1,2.8
  7,0,3.1
  8,b,a
  9,1.1,3.0
      ! Termination

```

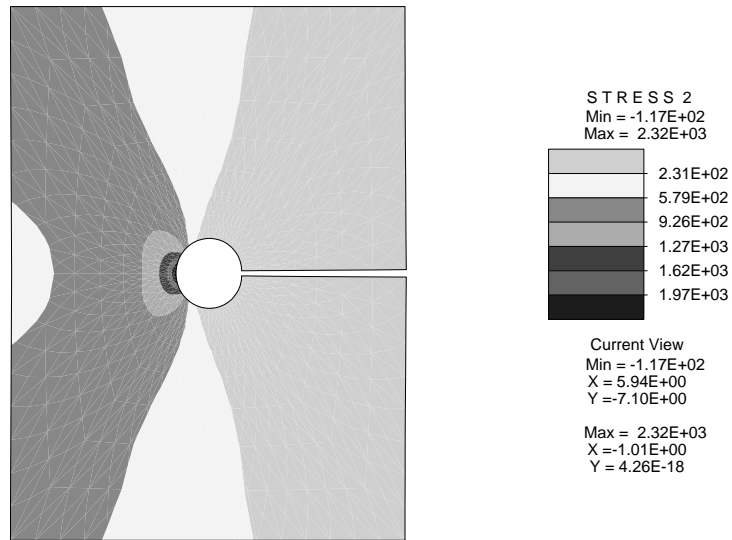
Figure 6.5: Tension Strip with Slit: Block Generation of Quadrant

```

EBOUndary
  2 8 0 1 0
  2 -8 0 1 0
      ! Termination
CBOUndary
  node -6 0 1 0
      ! Termination
EDISplacements
  2 8 0 0.5
  2 -8 0 -0.5
      ! Termination
MATERial 1
  SOLId
  ELASTic ISOTopic 10000 0.25
      ! Termination

```

Figure 6.6: Tension Strip with Slit: Boundary and Material Parameters

Figure 6.7: Tension Strip with Slit: Contours of σ_{yy}

```

PARAMeter
  n = 4

INCLude ISTRIP

PARAMeter
  n = 8

INCLude ISTRIP

PARAMeter
  n = 16

INCLude ISTRIP

STOP

```

Figure 6.8: Include data for tension strip refinements

in the file named `OSLIT`.

The same methodology also may be used to run a sequence of different problems. In this case a reference by the include statement would name each of the problem files to be solved.

Chapter 7

Example 5. Thermal Problem.

In this example we consider the solution of a linear thermal problem. The domain for the solution is a square with side lengths of 5-units. A steady state and a transient thermal analysis are to be performed. For the steady state analysis a temperature of $T = 1$ is applied on the entire left boundary and the right boundary is restrained to have temperature zero. For the transient problem the left side has a specified unit temperature ($T = 1$) suddenly applied at time zero and held constant and the right boundary is insulated ($q_n = 0$). The thermal material parameters are set as follows:

$$k = 10 \quad c = 1 \quad \rho = 0.1$$

The problem is solved using a 10×10 uniform mesh of 9-node quadrilateral elements. For the above properties the material data is specified as:

```
MATERial 1
  THERmal
    FOURIER ISOTROPIC 10.0 1.0
    DENSITY MASS      0.10
```

The mesh is generated using the block command with the data

```
FEAP
  0 0 0 2 1 9

BLOCK
  CARTESIAN 20 20 0 0 1 0 9
  1 0.0 0.0
  2 5.0 0.0
  3 5.0 5.0
  4 0.0 5.0

EBOUnd
```

```

1 0 1
1 5 1 ! Use for steady state problem only

EDISpl
1 0 1

MATE 1
... (material properties as above)

END

```

A plot of the mesh is shown in Fig 7.1

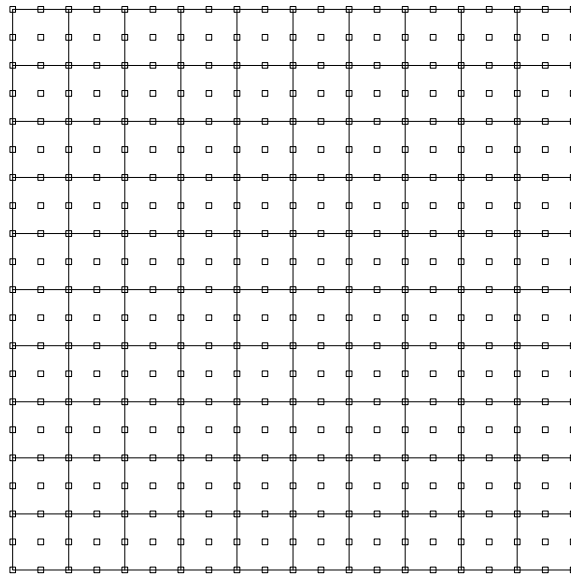


Figure 7.1: Mesh of 9-node elements

The solution to the steady state heat problem is given by:

$$T(x, y) = 1 - \frac{x}{5}$$

and is exactly captured by the finite element solution as indicated in Fig. 7.2. This solution was computed using the commands:

```

BATCh
  TANGent,,1      ! Solve problem
  PLOT,CONTOur,1 ! Contour solution
END

```

The transient solution is computed using a transient solution. This may be accomplished by inserting an `ORDER` command after the mesh `END` command and before the first solution execution. The order command is given as

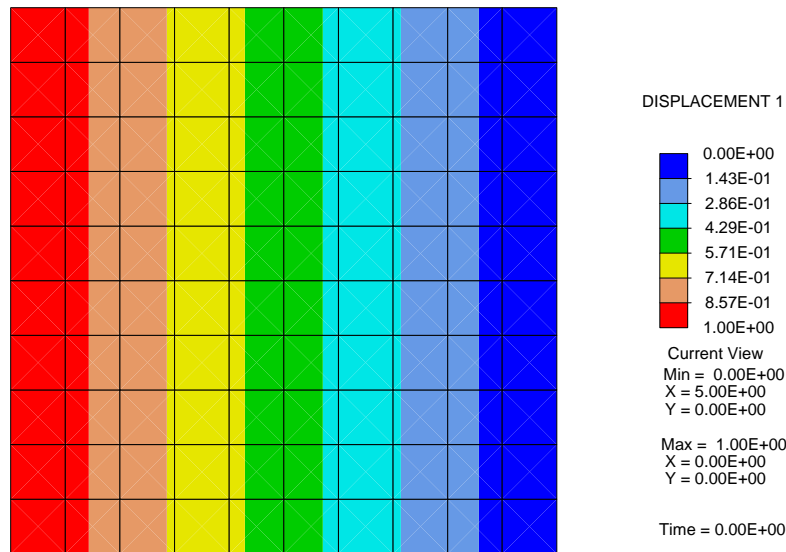


Figure 7.2: Steady state temperature solution (Patch test).

```
ORDER
1
```

where the 1 restricts transients for the heat equation to a first-order differential equation. Any time integrator could be tried, however, in the results given below the backward Euler scheme was used (i.e., `TRANSient,BACKward`). The solution was performed for 20 time steps of $\Delta t = 0.005$ with the command language program:

```
BATCH
PARAMeter
DT,,dt          ! Sets to parameter value
TRANS,BACK
LOOP,,20
TIME
TANG,,1        ! Problem linear, no iterations
PLOT,RANG,0.4,0.9
PLOT,WIPE
PLOT,CONT,3    ! Temperature contours
NEXT
END
dt= 0.0001
                ! End of parameter input
```

A solution at later times is continued using larger time steps. Generally, the solution can be computed by increasing the time increment by factors of 10 every logarithmic decade of time using a *FEAP function* program. The function program is contained in a file with the extender `.fcn`. For example a routine for the time increment can be given by:

```
dt = 10*dt
```

stored in the file `tinc.fcn`. The solution commands are then performed using the set of commands:

```
BATCh
  DT,,dt
  LOOP,,4
    FUNC,tinc
    LOOP,,9
      TIME
      TANG,,1
    NEXT
  NEXT
END
```

This will continue the solution to time 10.

Results for the temperature contours at selected times are shown in Figs 7.3 to 7.4. Note that this problem produces one dimensional results and could be solved using a much simpler meshing.

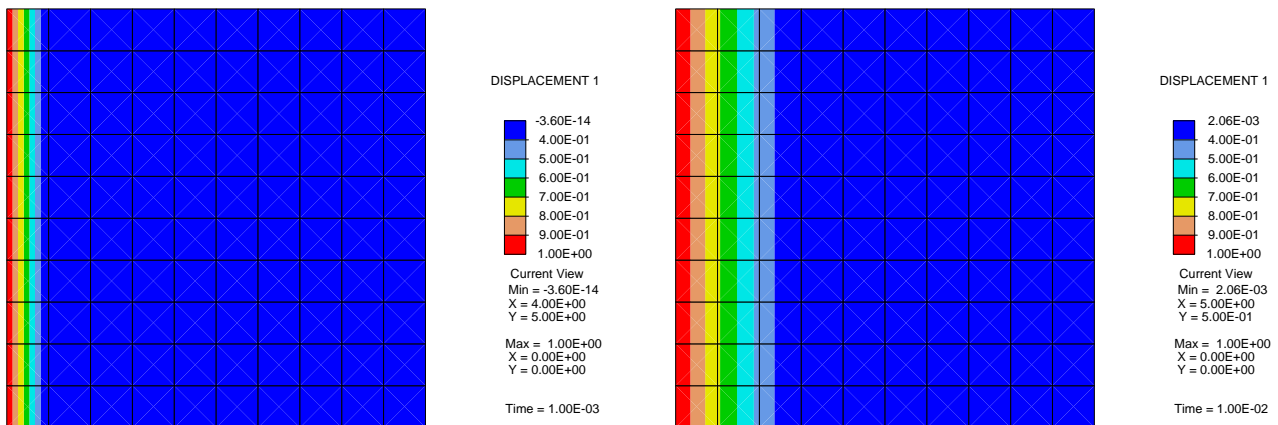


Figure 7.3: Temperature contours at $t = 0.001$ and $t = 0.01$

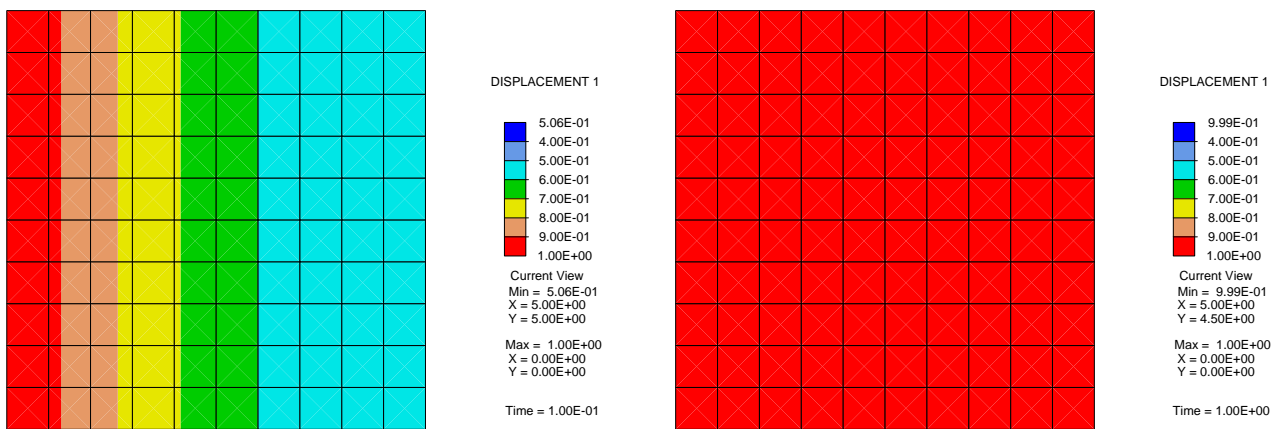


Figure 7.4: Temperature contours at $t = 0.1$ and $t = 1.0$

Chapter 8

Example 6. Coupled Thermo-mechanical Problem.

In this example we consider the solution of a thermo-mechanical problem in a state of plane strain. The domain for the solution is a square with side lengths of 5-units. A vertical roller support is specified at the upper right-hand corner and a pin support at the lower right-hand corner. This is just sufficient to prevent rigid body motions for a static problem. Loading is provided by a transient thermal analysis in which the left side has a specified unit temperature ($T = 1$) suddenly applied at time zero and held constant. The material parameters are set as follows:

1. Solid:

$$E = 100 \quad \nu = 0.4995a \quad \alpha = 0.25 \quad T_0 = 0$$

2. Thermal:

$$k = 10 \quad c = 1 \quad \rho = 0.1$$

The problem is solved using a 10×10 uniform mesh of 9-node quadrilateral elements. The *mixed* formulation is used. With the above properties and element, the material data is specified as:

```
MATeRial 1
SOLID
ELASTIC ISOTROPIC 100 0.4995
THERMAL ISOTROPIC 0.25 0.0
FOURIER ISOTROPIC 10.0 1.0
DENSITY MASS 0.10
MIXED
```

The mesh is generated using the block command with the data

```

FEAP
  0  0  0  2  3  9

BLOCK
  CARTESIAN 20 20 0 0 1 0 9
  1  0.0  0.0
  2  5.0  0.0
  3  5.0  5.0
  4  0.0  5.0

CBOUN
  NODE 5 0 1 1
  NODE 5 5 1 0

EDIS
  1  0  0  0  1

MATE 1
  ... (material properties as above)

END

```

A plot of the mesh is shown in Fig 8.1

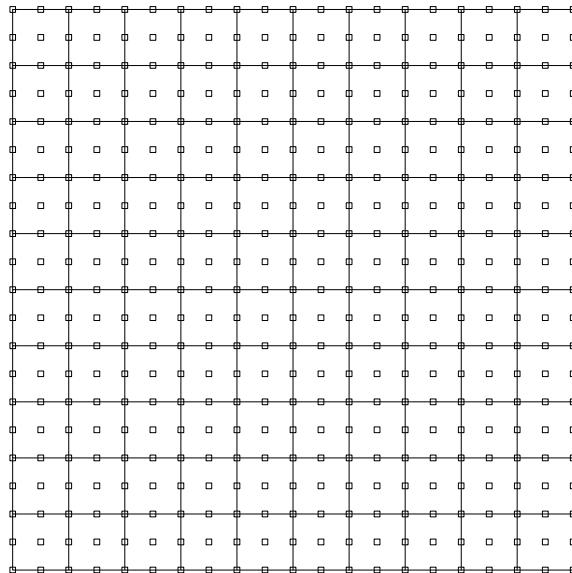


Figure 8.1: Mesh of 9-node elements

The solution of the solid mechanics problem is assumed to be at a time scale for which inertial effects may be ignored. The heat equation, however, is to be solved using a

transient solution. This may be accomplished by inserting an `ORDER` command after the mesh `END` command and before the first solution execution. The order command is given as

```
ORDER
  0 0 1
```

where the 0 denotes a zero-order (i.e., static) differential equation form and the 1 restricts transients for the heat equation to a first-order differential equation. Any time integrator could be tried, however, in the results given below the backward Euler scheme was used (i.e., `TRANSient,BACKward`). The solution was performed for 20 time steps of $\Delta t = 0.005$ with the command language program:

```
BATCH
  DT,,0.005
  TRANS,BACK
  PLOT,DEFORm
  PLOT,FACT,0.7 ! to keep plot in window
  LOOP,,20
    TIME
    LOOP,,5      ! Note that at least 3 iterations
      TANG,,1    ! are needed to converge (since no
    NEXT        ! compling tangent is included)
    PLOT,WIPE
    PLOT,RANG,0.4,0.9
    PLOT,CONT,3 ! Temperature contours
    PLOT,RANG,-1.0,0.2
    PLOT,STRE,1 ! 11-Stress contours
  NEXT
END
```

Results for the contours after the first and last step are shown in Fig 8.2 and 8.3. Numerical results for displacements and temperatures along the row of nodes at $x_1 = 0.25$ (center of left row of elements) are included in Table 8.1

The temperature for the row of nodes shown in the table is constant along the line $x_1 = 0.25$ and equals 0.70261 at time step 1 and 0.96363 at step 20.

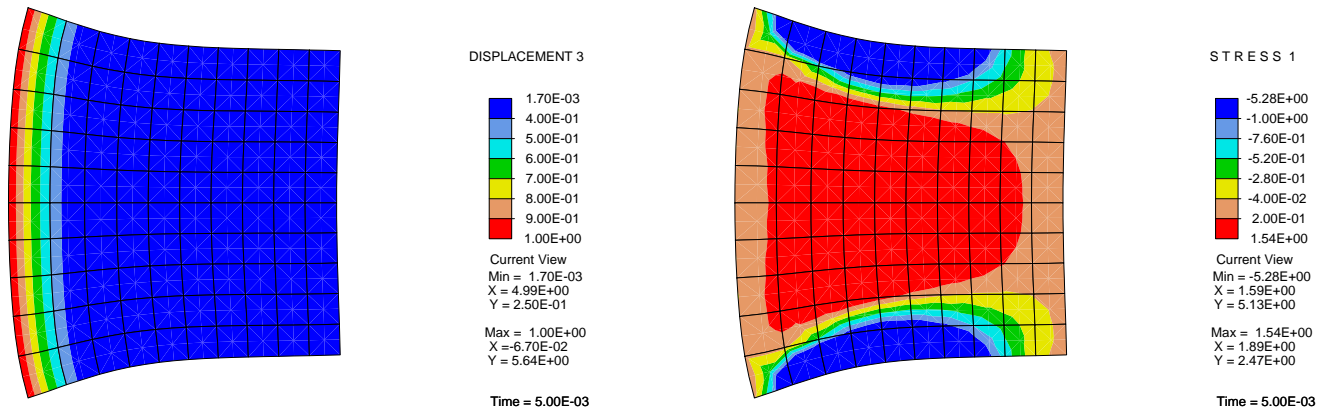


Figure 8.2: Thermal stress problem: Time = 0.005

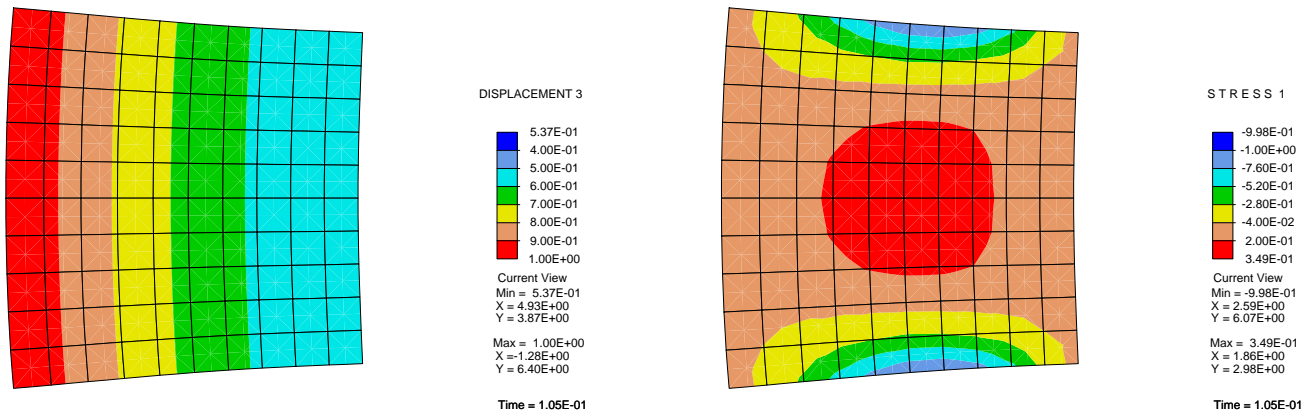


Figure 8.3: Thermal stress problem: Time = 0.100

Node	x_2	Time = 0.005		Time = 0.10	
		u_1	u_2	u_1	u_2
22	5.00	1.527E-02	5.254E-01	-1.188E+00	1.370E+00
23	4.75	-7.526E-02	4.576E-01	-1.216E+00	1.279E+00
24	4.50	-1.358E-01	3.921E-01	-1.240E+00	1.189E+00
25	4.25	-1.788E-01	3.314E-01	-1.261E+00	1.099E+00
26	4.00	-2.101E-01	2.731E-01	-1.278E+00	1.010E+00
27	3.75	-2.328E-01	2.180E-01	-1.293E+00	9.207E-01
28	3.50	-2.496E-01	1.651E-01	-1.304E+00	8.316E-01
29	3.25	-2.614E-01	1.143E-01	-1.313E+00	7.428E-01
30	3.00	-2.695E-01	6.483E-02	-1.319E+00	6.543E-01
31	2.75	-2.740E-01	1.640E-02	-1.322E+00	5.658E-01
32	2.50	-2.756E-01	-3.160E-02	-1.324E+00	4.774E-01
33	2.25	-2.740E-01	-7.961E-02	-1.322E+00	3.891E-01
34	2.00	-2.695E-01	-1.280E-01	-1.319E+00	3.006E-01
35	1.75	-2.614E-01	-1.775E-01	-1.313E+00	2.120E-01
36	1.50	-2.496E-01	-2.283E-01	-1.304E+00	1.233E-01
37	1.25	-2.328E-01	-2.812E-01	-1.293E+00	3.424E-02
38	1.00	-2.101E-01	-3.363E-01	-1.278E+00	-5.505E-02
39	0.75	-1.788E-01	-3.947E-01	-1.261E+00	-1.447E-01
40	0.50	-1.358E-01	-4.553E-01	-1.240E+00	-2.346E-01
41	0.25	-7.526E-02	-5.208E-01	-1.216E+00	-3.248E-01
42	0.00	1.527E-02	-5.886E-01	-1.188E+00	-4.151E-01

Table 8.1: Solution for displacements at time step 1 and 20

Chapter 9

Example 7. Contact Problem.

In this example we consider the solution of a two-dimensional contact problem. Two beams are placed a short distance apart with the top beam loaded by a uniformly distributed vertical load.

Each beam has a length of 20 units and the spacing between the beams is 0.5 units. The upper beam is divided into 11 equal length elements and the lower beam into 10 equal length elements. Each beams is clamped at both ends. Beams are modelled using the finite deformation **FRAMe** element. The material model is elastic ($E = 20,000$) with a cross sectional area $A = 0.1$ and moment of inertia $I = 1$.

The upper beam is loaded by a uniform load which produces a vertical load of $20 \times t$ units at each node (where t is time). During the analysis t increases from zero to 5 in equal increments of 0.1 units.

The mesh is generated using the commands:

```
FEAP
  0 0 0 2 3 2

PARAMeters
  pr = -20 ! Nodal loading
  a  = 20 ! Lower beam length
  b  = 20 ! Upper beam length
  h  = 0.5 ! Spacing between beams

BLOCK
  CARTesian 10 1 0 0 1
    1 0.0 0.0
    2  a 0.0

BLOCK
  CARTesian 11 1 0 0 2
    1 0.0 h
```

```

      2    b  h
EBOUnd
  1  0  1  1  1
  a  0  1  1  1

EFORce
  2  h  0  pr

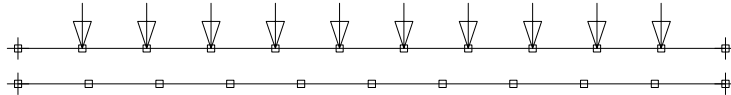
MATE 1
  FRAMe
    ELAStic ISOTropic  20000
    CROSSs section  0.1  1
    FINItE

MATE 2
  FRAMe
    ELAStic ISOTropic  20000
    CROSSs section  0.1  1
    FINItE

END

```

A plot of the mesh is shown in Fig 9.1



Time = 5.0

Figure 9.1: Mesh of two beams

The solution of a contact problem requires the specification of each individual surface to be considered and each pair of surfaces for which possible contacts are to be checked. Each surface is defined using a *right-hand rule* to specify the *outward* pointing normal. In the present example the contact between the two parallel beams is to be defined. The lower beam is designated as contact surface number 1 and surface facets are defined from the right end to the left end to satisfy the right-hand rule giving a outward normal pointing up on the figure. Similarly, the upper beam is designated as contact surface 2 and facets are numbered from left to right to give an outward normal pointing down. The two surfaces are specified to interact using the **PAIR** command with surface

1 being the *slave* and surface 2 the *master*. A node to surface contact with a penalty solution scheme is adopted (command NTOS) with an augmented lagrange correction applied (AUGM). The data for a contact follows the mesh description and for the present example is given by:

```

CONtact

SURFace 1
  LINE 2
  FACeT
  1 -1 11 10
  10 0 2 1

SURFace 2
  LINE 2
  FACeT
  1 1 12 13
  11 0 22 23

PAIR 1
  NTOS 1 2
  SOLM PENAlty 2e3
  AUGMent
  TOLE,,1e-5 1e-5 1e-5

END

```

A solution to the problem is obtained using the commands:

```

BATCh
  PROP
  DT,,0.1
END

BATCh
  LOOP,time,50
  TIME
  LOOP,augment,4
  LOOP,newton,30
  TANG,,1
  NEXT
  AUGMent
  NEXT
  NEXT
END

```

Note especially, the extra loop required to perform the augmentation.

Results for the deformed shape and contours of the final gap achieved (constructed using the plot command `PLOT,CVAR,9,9= gap`) are shown in Fig. 9.2.

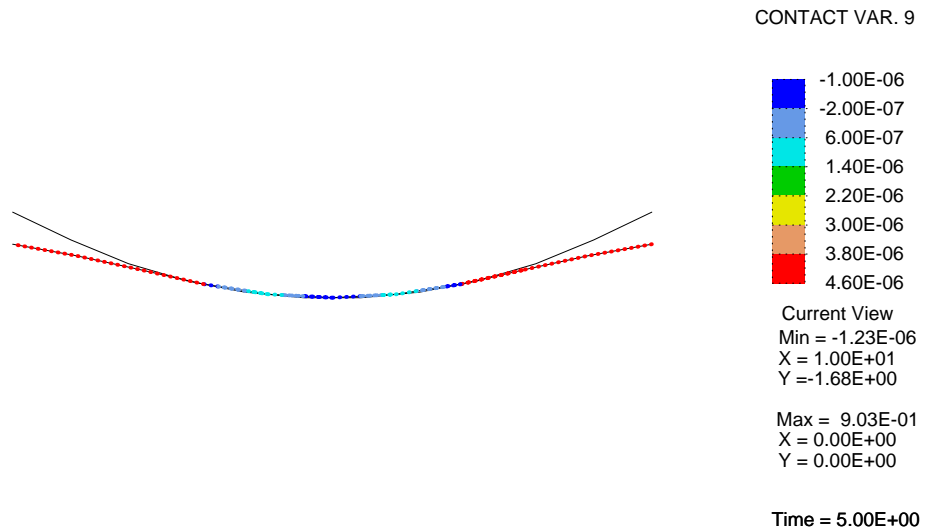


Figure 9.2: Contact deformed shape and gap: Time = 5