```
% Input File: Two Trianglular Elements Under Axial Load
% Copyright (C) Arif Masud and Tim Truster
% This input file should be run prior to executing the FEA_Program routine.
% Format of required input:
응
응
                     = number of nodes in the mesh (length(NodeTable))
    numnp:
응
                     = number of elements in the mesh
응
    numel:
응
응
    nen:
                     = maximum number of nodes per element (4)
2
응
    PSPS:
                     = flag for plane stress ('s') or plane strain ('n')
응
응
                     = table of mesh nodal coordinates defining the
    NodeTable:
                       geometry of the mesh; format of the table is as
응
                       follows:
응
응
                           Nodes
                                                x-coord y-coord
응
                                    NodeTable = [x1]
                           n1
                                                          у1
%
                           n2
                                                   x2
                                                          у2
응
                            . . .
응
                           nnumnp
                                                   xnumnp ynumnp];
2
응
                     = table of mesh connectivity information, specifying
    ix:
                       how nodes are attached to elements and how materials
응
                       are assigned to elements; entries in the first nen
읒
                       columns correspond to the rows of NodeTable
응
2
                       representing the nodes attached to element e;
응
                       entries in the last nen+1 column are rows from MateT
응
                       signifying the material properties assigned to
                       element e; format of the table is as follows:
응
응
                           Elements
                                               n1
                                                    n2
                                                           n3
                                                                  n4
응
                                         ix = [eln1 eln2 eln3 eln4 elmat]
2
                                               e2n1 e2n2 e2n3 e2n4 e2mat
                           e2.
응
                                                     . .
                            . . .
                                                . .
                                                            . .
                                                                  . .
2
                                              values for element numel ];
                           enumel
읒
%
                     = table of mesh material properties for each distinct
    MateT:
응
                       set of material properties; these sets are
응
                       referenced by element e by setting the value of
응
                       ix(e,nen+1) to the row number of the desired
                       material set; format of the table is as follows:
%
                                                  E v t
응
                           Materials
응
                           mat1
                                         MateT = [E1 v1 t1]
2
                                                  E2 v2 t2
                           mat2
응
                                                   .. .. ..];
                            . . .
2
응
    BCLIndex:
                     = list of the number of boundary conditions and loads
응
                       applied to the mesh; first entry is the number of
응
                       prescribed displacements at nodes; second entry is
```

```
응
                        the number of nodal forces
응
응
    NodeBC:
                     = table of prescribed nodal displacement boundary
                        conditions; it contains lists of nodes, the
응
응
                        direction of the displacement prescribed (x=1, y=2),
응
                        and the value of the displacement (set 0 for fixed
                        boundary); the length of the table must match the
2
                        entry in BCLIndex(1), otherwise an error will result
                        if too few conditions are given or extra BCs will be
읒
                        ignored in the model input module; format of the
응
                        table is as follows:
                                              nodes direction value
응
                            BCs
                                    NodeBC = [bc1n]
응
                            bc1
                                                     bc1dir
응
                            bc2
                                              bc2n
                                                     bc2dir bc2u
응
                            . . .
2
응
    NodeLoad:
                     = table of prescribed nodal forces; it contains lists
응
                        of nodes, the direction of the force prescribed
응
                        (x=1, y=2), and the value of the force; the length
응
                        of the table must match the entry in BCLIndex(2),
                        otherwise an error will result if too few conditions
응
응
                        are given or extra loads will be ignored in the
응
                       model input module; format of the table is as
응
                        follows:
응
                            Loads
                                                  nodes direction value
응
                            P1
                                      NodeLoad = [ Pln
                                                           P1dir
                                                                    P1P
2
                            P2
                                                   P2n
                                                           P2dir
                                                                    P2P
                                                                    ..];
                            . . .
2
읒
응
응
응
응
응
2
clear all; close all; clc
% Mesh Data:
L = 1;
H = 1;
thick = 1;
%Parameters for strain-energy function (Units consistent throughout)
mu1=100;
mu2=1000;
kappa=1000;
```

```
thick = 1.0;
MateT = [mu1 mu2 kappa thick]; PSPS = 'nl';
% Body Force:
Fb = zeros(2,1);
% Time-Integration Parameters :
alpha = 0; gamma = (1-2*alpha)/2; beta = (1-alpha)^2/4; delt = 0.001;
del_t = delt; maxiter = 50; tmax=10; tSteps = tmax/delt + 1;
% Mesh Nodal Coordinates
NodeTable = [0]
             L
             L
                   Η
             0
                   Hl;
numnp = length(NodeTable);
% Mesh Element Connectivities
ix = [1 \ 2 \ 3 \ 4 \ 1];
nen = 4;
numel = 1;
% Mesh Boundary Conditions and Loads
BCLIndex = [4 2]';
NodeBC = [1 \ 1 \ 0]
          1 2 0
          2 1 0
          2 2 0];
NodeLoad = [3 2
                   0];
            4 2
LoadDist=[0;0]; % Distribution of the Load on the two points.
tol=10^-12;
                      %tolerance used in N-R Method
                      %increments in which external load are applied
% Magnitude of the Initial Applied Quasi-Static Displacement :
d1 = 0;
d2 = 0.2;
d3 = 0;
d4 = 0.2;
% Magnitude of the Initial Applied Velocity :
v1 = 0;
v2 = 0;
v3 = 0;
v4 = 0;
% Magnitude of the Initial Displacement :
do = [d1 d2 d3 d4]';
```

```
vo = [v1 \ v2 \ v3 \ v4]';
% % Mass Matrix remains constant :
% ElemM = 0.25* [0.44667 0 0.22333 0 0.11222 0 0.22444 0
                  0 0.44667 0 0.22333 0 0.11222 0 0.224444
응
                  0 0 0.44667 0 0.22444 0 0.11222 0
                  0 0 0 0.44667 0 0.22444 0 0.11222
응
                  0 0 0 0 0.45111 0 0.22555 0
                  0 0 0 0 0 0.45111 0 0.22555
응
                  0 0 0 0 0 0 0.45111 0
                  0 0 0 0 0 0 0 0.451111];
% ElemM = ElemM + ElemM';
% for im = 1:length(diag(ElemM))
       ElemM(im,im) = ElemM(im,im)/2;
2
  end
% % Pre-assigning Mdd to enhance the Speed :
% Mdd = ElemM(5:8,5:8);
clc
iel = 1;
ndf = 2;
ndm = 2;
nieq = 0;
isw = 3;
% Interpret Boundary Conditions and assign Loads; allocate DOFS
assign_bc_load_data
nneq = neq + nieq;
% Initialization of the Newton Step:
n = 1; storej=1;
dn(:,n) = zeros(neq,1);
vn(:,n) = zeros(neq,1);
an(:,n) = zeros(neq,1);
% Last converged displacements for the first Newton step :
dn(:,1) = do;
vn(:,1) = vo;
% First iteration displacement for the first step : (Same as above)
dis(:,storej) = dn;
vel(:,storej) = vn;
% Definition of the External Force Vector :
time=linspace(0,tmax,tSteps);
FEA_Program
```

