```
% 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:
%
%   numnp:           = number of nodes in the mesh (length(NodeTable))
%
%   numel:           = number of elements in the mesh
%
%   nen:             = maximum number of nodes per element (4)
%
%   PSPS:            = flag for plane stress ('s') or plane strain ('n')
%
%   NodeTable:       = table of mesh nodal coordinates defining the
%                      geometry of the mesh; format of the table is as
%                      follows:
%                          Nodes  |             x-coord  y-coord
%                          n1     |  NodeTable = [x1     y1
%                          n2     |                x2     y2
%                          ...    |                ..     ..
%                          nnumnp |                xnumnp ynumnp];
%
%   ix:              = table of mesh connectivity information, specifying
%                      how nodes are attached to elements and how materials
%                      are assigned to elements; entries in the first nen
%                      columns correspond to the rows of NodeTable
%                      representing the nodes attached to element e;
%                      entries in the last nen+1 column are rows from MateT
%                      signifying the material properties assigned to
%                      element e; format of the table is as follows:
%                          Elements   |         n1    n2    n3    n4   mat
%                          e1         |  ix = [e1n1  e1n2  e1n3  e1n4 e1mat
%                          e2         |        e2n1  e2n2  e2n3  e2n4 e2mat
%                          ...        |         ..    ..    ..    ..   ..
%                          enumel     |        values for element numel   ];
%
%   MateT:           = table of mesh material properties for each distinct
%                      set of material properties; these sets are
%                      referenced by element e by setting the value of
%                      ix(e,nen+1) to the row number of the desired
%                      material set; format of the table is as follows:
%                          Materials  |           E   v   t
%                          mat1       |  MateT = [E1  v1  t1
%                          mat2       |           E2  v2  t2
%                          ...        |           ..  ..  ..];
%
%   BCLIndex:        = list of the number of boundary conditions and loads
%                      applied to the mesh; first entry is the number of
%                      prescribed displacements at nodes; second entry is
```

```
%                         the number of nodal forces
%
%   NodeBC:             = table of prescribed nodal displacement boundary
%                         conditions; it contains lists of nodes, the
%                         direction of the displacement prescribed (x=1, y=2),
%                         and the value of the displacement (set 0 for fixed
%                         boundary); the length of the table must match the
%                         entry in BCLIndex(1), otherwise an error will result
%                         if too few conditions are given or extra BCs will be
%                         ignored in the model input module;  format of the
%                         table is as follows:
%                             BCs  |            nodes direction value
%                             bc1  |  NodeBC = [bc1n   bc1dir   bc1u
%                             bc2  |            bc2n   bc2dir   bc2u
%                             ...  |             ..     ..       .. ];
%
%   NodeLoad:           = table of prescribed nodal forces; it contains lists
%                         of nodes, the direction of the force prescribed
%                         (x=1, y=2), and the value of the force; the length
%                         of the table must match the entry in BCLIndex(2),
%                         otherwise an error will result if too few conditions
%                         are given or extra loads will be ignored in the
%                         model input module; format of the table is as
%                         follows:
%                             Loads  |              nodes direction value
%                             P1     |  NodeLoad = [ P1n    P1dir    P1P
%                             P2     |               P2n    P2dir    P2P
%                             ...    |               ..     ..       .. ];
%
%
%           4 -------------- 3       2
%           |                |       | \
%           |                |       |  \
%           |                |       |   \
%           |                |       |    \
%           |                |       |     \
%           |                |       |      \
%           1 -------------- 2       3-------1
%


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;
```

```
MateT = [mu1 mu2 kappa thick]; PSPS = 'nl';

% Body Force:
Fb = zeros(2,1);

% Time-Integration Parameters :
alpha = -1/3; 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     0
             L     0
             L     H
             0     H];

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   0];

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 = 0.2;
d2 = 0;
d3 = 0.2;
d4 = 0;

% 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
%                 0 0 0.44667 0 0.22444 0 0.11222 0
%                 0 0 0 0.44667 0 0.22444 0 0.11222
%                 0 0 0 0 0.45111 0 0.22555 0
%                 0 0 0 0 0 0.45111 0 0.22555
%                 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;
%  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
```

Published with MATLAB® R2013a