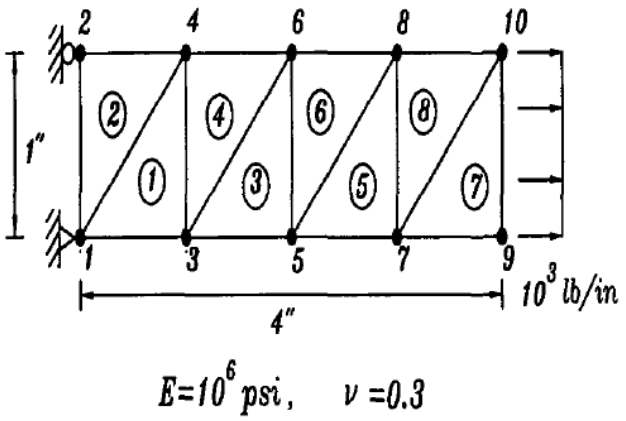

Figure 9.7.1 Plate Subjected to Axial Load

```
%------------------------------------------------------------------
% Example 9.7.1
% plane stress analysis of a solid using linear triangular elements
% (see Fig. 9.7.1 for the finite element mesh)
%
% Variable descriptions
% k = element matrix
% f = element vector
% kk = system matrix
% ff = system vector
% disp = system nodal displacement vector
% eldisp = element nodal displacement vector
% stress = matrix containing stresses
% strain = matrix containing strains
% gcoord = coordinate values of each node
% nodes = nodal connectivity of each element
% index = a vector containing system dofs associated with each element
% bcdof = a vector containing dofs associated with boundary conditions
% bcval = a vector containing boundary condition values associated with
%         the dofs in bcdof
%------------------------------------------------------------------
%
%--------------------------
% input data for control parameters
%--------------------------
nel=8;                                  % number of elements
nnel=3;                                 % number of nodes per element
ndof=2;                                 % number of dofs per node
nnode=10;                               % total number of nodes in system
sdof=nnode*ndof;                        % total system dofs
edof=nnel*ndof;                         % degrees of freedom per element
emodule=100000.0;                       % elastic modulus
poisson=0.3;                            % Poisson's ratio
```

```
%
%--------------------------------------
% input data for nodal coordinate values
% gcoord(i,j) where i-> node no. and j-> x or y
%--------------------------------------
gcoord=[0.0 0.0; 0.0 1.0; 1.0 0.0; 1.0 1.0; 2.0 0.0;
2.0 1.0; 3.0 0.0; 3.0 1.0; 4.0 0.0; 4.0 1.0];
%
%------------------------------------------------
% input data for nodal connectivity for each element
% nodes(i,j) where i-> element no. and j-> connected nodes
%------------------------------------------------
nodes=[1 3 4; 1 4 2; 3 5 6; 3 6 4;
5 7 8; 5 8 6;7 9 10; 7 10 8];
%
%------------------------------
% input data for boundary conditions
%------------------------------
bcdof=[1 2 3];                        % first three dofs are constrained
bcval=[0 0 0];                         % whose described values are 0
%
%--------------------------------
% initialization of matrices and vectors
%--------------------------------
ff=zeros(sdof,1);                                    % system force vector
kk=zeros(sdof,sdof);                                        % system matrix
disp=zeros(sdof,1);                          % system displacement vector
eldisp=zeros(edof,1);                       % element displacement vector
stress=zeros(nel,3);              % matrix containing stress components
strain=zeros(nel,3);              % matrix containing strain components
index=zeros(edof,1);                                         % index vector
kinmtx=zeros(3,edof);                                    % kinematic matrix
matmtx=zeros(3,3);                                    % constitutive matrix
%
%-------------
% force vector
%-------------
ff(17)=500;                          % force applied at node 9 in x-axis
ff(19)=500;                         % force applied at node 10 in x-axis
%
%----------------------------------------------
% compute element matrices and vectors, and assemble
%----------------------------------------------
matmtx=fematiso(1,emodule,poisson);                 % constitutive matrix
%
for iel=1:nel                      % loop for the total number of elements
%
nd(1)=nodes(iel,1);              % 1st connected node for (iel)-th element
```

```
nd(2)=nodes(iel,2);              % 2nd connected node for (iel)-th element
nd(3)=nodes(iel,3);              % 3rd connected node for (iel)-th element
%
x1=gcoord(nd(1),1); y1=gcoord(nd(1),2);      % coord values of 1st node
x2=gcoord(nd(2),1); y2=gcoord(nd(2),2);      % coord values of 2nd node
x3=gcoord(nd(3),1); y3=gcoord(nd(3),2);      % coord values of 3rd node
%
index=feeldof(nd,nnel,ndof);      % extract system dofs for the element
%
%-----------------------------------------
% find the derivatives of shape functions
%-----------------------------------------
area=0.5*(x1*y2+x2*y3+x3*y1-x1*y3-x2*y1-x3*y2); % area of triangule
area2=area*2;
dhdx=(1/area2)*[(y2-y3) (y3-y1) (y1-y2)];        % derivatives w.r.t. x
dhdy=(1/area2)*[(x3-x2) (x1-x3) (x2-x1)];        % derivatives w.r.t. y
%
kinmtx2=fekine2d(nnel,dhdx,dhdy);                 % kinematic matrix
%
k=kinmtx2'*matmtx*kinmtx2*area;               % element stiffnes matrix
%
kk=feasmbl1(kk,k,index);                    % assemble element matrices
%
end                      % end of loop for the total number of elements
%
%--------------------------
% apply boundary conditions
%--------------------------
[kk,ff]=feaplyc2(kk,ff,bcdof,bcval);
%
%------------------------
% solve the matrix equation
%------------------------
disp=kk\ff;
%
%---------------------------------------------
% element stress computation (post computation)
%---------------------------------------------
for ielp=1:nel                  % loop for the total number of elements
%
nd(1)=nodes(ielp,1);             % 1st connected node for (iel)-th element
nd(2)=nodes(ielp,2);             % 2nd connected node for (iel)-th element
nd(3)=nodes(ielp,3);             % 3rd connected node for (iel)-th element
%
x1=gcoord(nd(1),1); y1=gcoord(nd(1),2);      % coord values of 1st node
x2=gcoord(nd(2),1); y2=gcoord(nd(2),2);      % coord values of 2nd node
x3=gcoord(nd(3),1); y3=gcoord(nd(3),2);      % coord values of 3rd node
%
```

```
index=feeldof(nd,nnel,ndof);         % extract system dofs for the element
%
%-----------------------------------
% extract element displacement vector
%-----------------------------------
for i=1:edof
eldisp(i)=disp(index(i));
end
%
area=0.5*(x1*y2+x2*y3+x3*y1-x1*y3-x2*y1-x3*y2);   % area of triangle
area2=area*2;
dhdx=(1/area2)*[(y2-y3) (y3-y1) (y1-y2)];          % derivatives w.r.t. x
dhdy=(1/area2)*[(x3-x2) (x1-x3) (x2-x1)];          % derivatives w.r.t. y
%
kinmtx2=fekine2d(nnel,dhdx,dhdy);                   % kinematic matrix
%
estrain=kinmtx2*eldisp;                              % compute strains
estress=matmtx*estrain;                              % compute stresses
%
for i=1:3
strain(ielp,i)=estrain(i);                          % store for each element
stress(ielp,i)=estress(i);                          % store for each element
end
%
end
%
%---------------------
% print fem solutions
%---------------------
num=1:1:sdof;
displace=[num' disp]                            % print nodal displacements
%
for i=1:nel
stresses=[i stress(i,:)]                                   % print stresses
end
%
%---------------------------------------------------------------------


function [kinmtx2]=fekine2d(nnel,dhdx,dhdy)
%---------------------------------------------------------------------
% Purpose:
% determine the kinematic equation between strains and displacements
% for two-dimensional solids
%
% Synopsis:
% [kinmtx2]=fekine2d(nnel,dhdx,dhdy)
```

```
%
% Variable Description:
% nnel - number of nodes per element
% dhdx - derivatives of shape functions with respect to x
% dhdy - derivatives of shape functions with respect to y
%------------------------------------------------------------------
%
for i=1:nnel
i1=(i-1)*2+1;
i2=i1+1;
kinmtx2(1,i1)=dhdx(i);
kinmtx2(2,i2)=dhdy(i);
kinmtx2(3,i1)=dhdy(i);
kinmtx2(3,i2)=dhdx(i);
end
%
%------------------------------------------------------------------


function [matmtrx]=fematiso(iopt,elastic,poisson)
%------------------------------------------------------------------
% Purpose:
% determine the constitutive equation for isotropic material
%
% Synopsis:
% [matmtrx]=fematiso(iopt,elastic,poisson)
%
% Variable Description:
% elastic - elastic modulus
% poisson - Poisson's ratio
% iopt=1 - plane stress analysis
% iopt=2 - plane strain analysis
% iopt=3 - axisymmetric analysis
% iopt=4 - three dimensional analysis
%------------------------------------------------------------------
%
if iopt==1                                          % plane stress
matmtrx= elastic/(1-poisson*poisson)* ...
[1 poisson 0; ...
poisson 1 0; ...
0 0 (1-poisson)/2];
%
elseif iopt==2                                      % plane strain
matmtrx= elastic/((1+poisson)*(1-2*poisson))* ...
[(1-poisson) poisson 0;
poisson (1-poisson) 0;
0 0 (1-2*poisson)/2];
```

```
%
elseif iopt==3                                              % axisymmetry
matmtrx= elastic/((1+poisson)*(1-2*poisson))* ...
[(1-poisson) poisson poisson 0;
poisson (1-poisson) poisson 0;
poisson poisson (1-poisson) 0;
0 0 0 (1-2*poisson)/2];
%
else                                                    % three-dimension
matmtrx= elastic/((1+poisson)*(1-2*poisson))* ...
[(1-poisson) poisson poisson 0 0 0;
poisson (1-poisson) poisson 0 0 0;
poisson poisson (1-poisson) 0 0 0;
0 0 0 (1-2*poisson)/2 0 0;
0 0 0 0 (1-2*poisson)/2 0;
0 0 0 0 0 (1-2*poisson)/2];
%
end
%
%------------------------------------------------------------------
```

The nodal displacements are listed below and they agree with the analytical solutions. On the other hand, the state of stress of each element is $\sigma_x = 1000$ and $\sigma_y = \tau_{xy} = 0$ as expected.

```
displace =
d.o.f.    displ.
1.0000    0.0000                     % x-displacement of node 1
2.0000    0.0000                     % y-displacement of node 1
3.0000    0.0000                     % x-displacement of node 2
4.0000    -0.0030                    % y-displacement of node 2
5.0000    0.0100                     % x-displacement of node 3
6.0000    0.0000                     % y-displacement of node 3
7.0000    0.0100                     % x-displacement of node 4
8.0000    -0.0030                    % y-displacement of node 4
9.0000    0.0200                     % x-displacement of node 5
10.000    0.0000                     % y-displacement of node 5
11.000    0.0200                     % x-displacement of node 6
12.000    -0.0030                    % y-displacement of node 6
13.000    0.0300                     % x-displacement of node 7
14.000    0.0000                     % y-displacement of node 7
15.000    0.0300                     % x-displacement of node 8
16.000    -0.0030                    % y-displacement of node 8
17.000    0.0400                     % x-displacement of node 9
18.000    0.0000                     % y-displacement of node 9
19.000    0.0400                     % x-displacement of node 10
20.000    -0.0030                    % y-displacement of node 10
```

‡

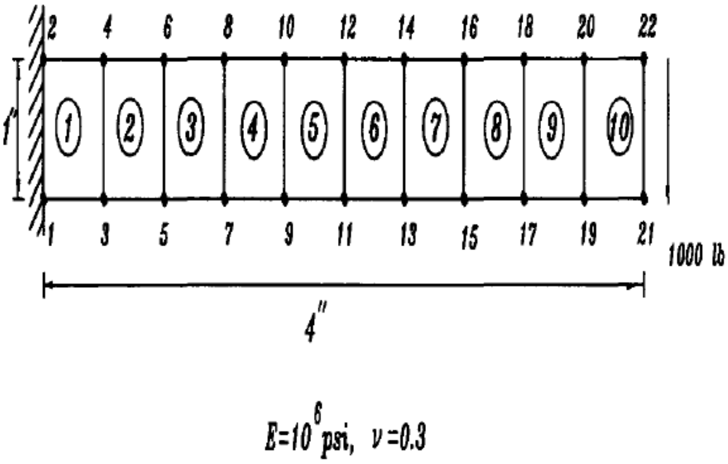

Figure 9.7.2 Cantilever Beam Subjected to a Tip Load

♣ **Example 9.7.2** We want to analyze a short cantilever beam using two-dimensional isoparametric elements assuming plane stress condition. To this end, the beam is modeled using ten four-node quadrilateral elements as seen in Fig. 9.7.2.

```
%--------------------------------------------------------------------
% Example 9.7.2
% plane stress analysis of a cantilever beam using isoparametric
% four-node elements
% (see Fig. 9.7.2 for the finite element mesh)
%
% Variable descriptions
% k = element matrix
% f = element vector
% kk = system matrix
% ff = system vector
% disp = system nodal displacement vector
% eldisp = element nodal displacement vector
% stress = matrix containing stresses
% strain = matrix containing strains
% gcoord = coordinate values of each node
% nodes = nodal connectivity of each element
% index = a vector containing system dofs associated with each element
% point2 = matrix containing sampling points
% weight2 = matrix containing weighting coefficients
% bcdof = a vector containing dofs associated with boundary conditions
% bcval = a vector containing boundary condition values associated with
%            the dofs in bcdof
%--------------------------------------------------------------------
%
%--------------------------------
% input data for control parameters
```