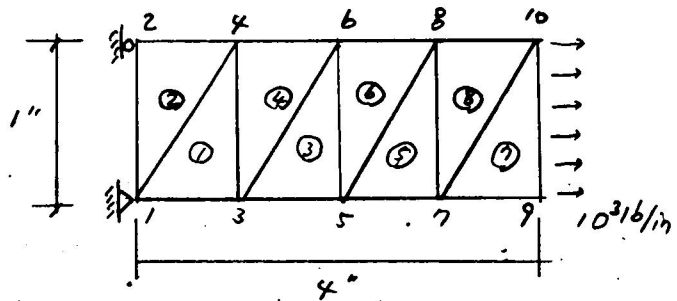


```
%
% Example 9.8.1
% plane stress analysis of a solid using linear triangular elements
% (see Fig. 9.8.1 for the finite element mesh)
```

```
% 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
%
```

```
clear
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
nf=nnode*ndof; % total system dofs
neof=nnel*ndof; % degrees of freedom per element
emodulus=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
)ess=zeros(nel,3);    % matrix containing stress components
strain=zeros(nel,3);  % matrix containing strain components
index=zeros(edof,1);  % index vector
kinmtx2=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

%-----
% computation of element matrices and vectors and their assembly
%-----
)
matmtx=fematiso(1,module,poisson); % compute 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,nel,ndof); % extract system dofs associated with element

%-----
% find the derivatives of shape functions
%-----

```

$$kinmtx2 = \underline{\underline{B}}$$

$$matmtx = \underline{\underline{E}}$$

$$matmtx = \underline{\underline{E}}$$

```

area=0.5*(x1*y2+x2*y3+x3*y1-x1*y3-x2*y1-x3*y2); % area of triangle
area2=area*2;
dhdxdx=(1/area2)*[(y2-y3) (y3-y1) (y1-y2)]; % derivatives w.r.t. x-axis  $\frac{df}{dx}$ 
dhdxdy=(1/area2)*[(x3-x2) (x1-x3) (x2-x1)]; % derivatives w.r.t. y-axis  $\frac{df}{dy}$ 

kinmtx2=fekine2d(nnel,dhdxdx,dhdxdy); % compute kinematic matrix  $B$ 

k=kinmtx2'*matmtx*kinmtx2*area; % element stiffness matrix  $\underline{k} = \underline{B}^T \underline{E} \underline{B} \underline{A} \quad (t=1)$ 

kk=feasmb11(kk,k,index); % assemble element matrices

end

% -----
% apply boundary conditions
% -----
)
[kk,ff]=feaplyc2(kk,ff,bcdof,bcval);

% -----
% solve the matrix equation
% -----

disp(kk\ff);  $\underline{U} = \underline{K}^{-1} \underline{P}$ 

% -----
% element stress computation
% -----

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 associated with 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-axis
dhdy=(1/area2)*[(x3-x2) (x1-x3) (x2-x1)]; % derivatives w.r.t. y-axis
```

```
kinmtx2=fekine2d(nnel,dhdx,dhdy); % compute kinematic matrix
```

```
estrain=kinmtx2*eldisp; % compute strains  $\underline{\epsilon} = \underline{B} \cdot \underline{d}$ 
estress=matmtx*estrain; % compute stresses  $\underline{\sigma} = \underline{E} \underline{\epsilon}$ 
```

```
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
```

$$\underline{B} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & 0 & \frac{\partial f_2}{\partial x} & 0 & \frac{\partial f_3}{\partial x} & 0 \\ 0 & \frac{\partial f_1}{\partial y} & 0 & \frac{\partial f_2}{\partial y} & 0 & \frac{\partial f_3}{\partial y} \\ \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial x} \end{bmatrix}$$

)

```
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];
```

$$\underline{\underline{E}} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix}$$

```
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
```

```
function [index]=feeldof(nd,nnel,ndof)
%-----
% Purpose:
%   Compute system dofs associated with each element
%
% Synopsis:
%   [index]=feeldof(nd,nnel,ndof)
%
% Variable Description:
%   index - system dof vector associated with element "iel"
%   iel - element number whose system dofs are to be determined
%   nnel - number of nodes per element
%   ndof - number of dofs per node
%-----

edof = nnel*ndof;
k=0;
for i=1:nnel
    start = (nd(i)-1)*ndof;
    for j=1:ndof
        k=k+1;
        index(k)=start+j;
    end
end
end
```

)

```
function [kk]=feasmb11(kk,k,index)
%-----
% Purpose:
%   Assembly of element matrices into the system matrix
%
% Synopsis:
%   [kk]=feasmb11(kk,k,index)
%
% Variable Description:
%   kk - system matrix
%   k  - element matrix
%   index - d.o.f. vector associated with an element
%-----
```

```
edof = length(index);
for i=1:edof
    ii=index(i);
    for j=1:edof
        jj=index(j);
        kk(ii,jj)=kk(ii,jj)+k(i,j);
    end
end
end
```

)


```
function [kk,ff]=feaplyc2(kk,ff,bcdof,bcval)
```

```
%-----  
% Purpose:  
%   Apply constraints to matrix equation [kk]{x}={ff}  
%  
% Synopsis:  
%   [kk,ff]=feaplybc(kk,ff,bcdof,bcval)  
%  
% Variable Description:  
%   kk - system matrix before applying constraints  
%   ff - system vector before applying constraints  
%   bcdof - a vector containing constrained d.o.f  
%   bcval - a vector containing contained value  
%  
%   For example, there are constraints at d.o.f=2 and 10  
%   and their constrained values are 0.0 and 2.5,  
%   respectively. Then, bcdof(1)=2 and bcdof(2)=10; and  
%   bcval(1)=1.0 and bcval(2)=2.5.  
%-----
```

```
n=length(bcdof);
```

```
sdof=size(kk);
```

```
for i=1:n
```

```
    c=bcdof(i);
```

```
    for j=1:sdof
```

```
        kk(c,j)=0;
```

```
    end
```

```
    kk(c,c)=1;
```

```
    ff(c)=bcval(i);
```

```
end
```

```
);
```