Computer Implementation 8.2 ***(Matlab)*** *Modal analysis of a plane truss (p. 568)*

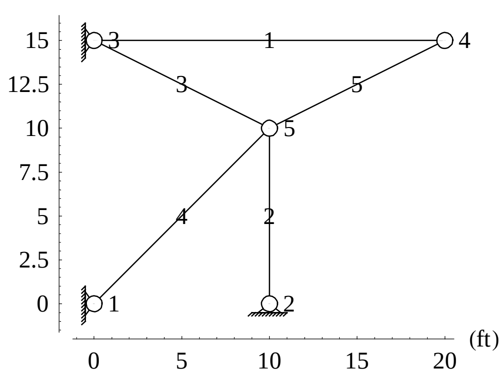
The following TransientPlaneTrussElement function returns the mass and the stiffness matrix of a plane truss element. To generate the mass matrix the function needs the mass density (ρ).

MatlabFiles\Chap8\TransientPlaneTrussElement.m

```
function [m, k] = TransientPlaneTrussElement(e, A, rho, coord)
% [m, k] = TransientPlaneTrussElement(e, A, rho, coord)
% Generates mass & stiffness matrices for a plane truss element
% rho = mass density
% e = modulus of elasticity
% A = area of cross-section
% coord = coordinates at the element ends

x1=coord(1,1); y1=coord(1,2);
x2=coord(2,1); y2=coord(2,2);
L=sqrt((x2-x1)^2+(y2-y1)^2);
ls=(x2-x1)/L; ms=(y2-y1)/L;
k = e*A/L*[ls^2, ls*ms,-ls^2,-ls*ms;
   ls*ms, ms^2,-ls*ms,-ms^2;
   -ls^2,-ls*ms,ls^2,ls*ms;
   -ls*ms,-ms^2,ls*ms,ms^2];
m = ((rho*A*L)/6)*[2, 0, 1, 0; 0, 2, 0, 1;
   1, 0, 2, 0;0, 1, 0, 2];
```

Using this function we consider free vibration analysis of the plane truss structure shown. All members have the same cross-sectional area and are made of the same material, $\rho = 490\,\text{lb}/\text{ft}^3$, $E = 30\times10^6\,\text{lb}/\text{in}^2$ and $A = 1.25\,\text{in}^2$. The truss supports a rotating machine weighing 2000 lbs at its tip.



The weight of the machine is an added mass at node 4. This is defined as ma in the following line and is added directly to the global mass matrix corresponding to degrees of freedom 7 and 8 associated with node 4. The procedure for generating the global mass and stiffness matrix is exactly the same as that for the static analysis used in the previous chapters. Using in-lb units the calculations are as follows.

MatlabFiles\Chap8\ModalTrussEx.m

```
% Modal analysis of a plane truss
g = 386.4; e = 30*10^6; A = 1.25;
rho = (490/(12^3))/g; ma = 2000/g;
nodes = 12 * [0, 0; 10, 0; 0, 15; 20, 15; 10, 10];
conn = [3, 4; 2, 5; 5, 3; 1, 5; 5, 4];
elems = size(conn,1);
lmm=[];
for i=1:elems
   lmm = [lmm; [2*conn(i,1)-1, 2*conn(i,1),2*conn(i,2)-1, 2*conn(i,2)]];
end
debc = [1:6]; ebcVals=zeros(length(debc),1);
dof=2*size(nodes,1);
M=zeros(dof); K=zeros(dof);
% Add nodal masses to the global M matrix.

M(7,7)=ma; M(8,8) = ma;
% Generate equations for each element and assemble them.
for i=1:elems
   con = conn(i,:);
   lm = lmm(i,:);
   [m, k] = TransientPlaneTrussElement(e, A, rho, nodes(con,:));
   M(lm, lm) = M(lm, lm) + m;
   K(lm, lm) = K(lm, lm) + k;
end

% Adjust for essential boundary conditions
dof = length(R);
df = setdiff(1:dof, debc);
Mf = M(df, df);
Kf = K(df, df);

% Compute frequencies and mode shapes
[V, lam] = eig(Kf, Mf);
freq=sqrt(lam)
modeShapes = V

>> ModalTrussEx

freq =
```

55.835	0	0	0
0	235.46	0	0
0	0	1598.4	0
0	0	0	1971.6

modeShapes =

0.099847	-0.42131	-0.01827	-0.036118
-0.42305	-0.098538	0.0011071	-0.020645
-0.041882	-0.18022	1.6041	1.8044
-0.01476	-0.062462	-1.8123	1.6046