Computer Implementation 4.2 *(Matlab)* *Space truss (p. 231)*

The analysis of space trusses can be performed conveniently by writing two small *Matlab* functions, one for defining the element stiffness matrix and the other for computing the element strain, stress, and the axial force.

MatlabFiles\Chap4\SpaceTrussElement.m

```
function k = SpaceTrussElement(e, A, coord)
% k = SpaceTrussElement(e, A, coord)
% Generates stiffness matrix of a space truss element
% e = modulus of elasticity
% A = Area of cross-section
% coord = coordinates at the element ends

x1=coord(1,1); y1=coord(1,2); z1=coord(1,3);
x2=coord(2,1); y2=coord(2,2); z2=coord(2,3);
L=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);
ls=(x2-x1)/L; ms=(y2-y1)/L; ns=(z2-z1)/L;
k = e*A/L*[ls^2, ls*ms, ls*ns, -ls^2, -(ls*ms), -(ls*ns);
   ls*ms, ms^2, ms*ns, -(ls*ms), -ms^2, -(ms*ns);
   ls*ns, ms*ns, ns^2, -(ls*ns), -(ms*ns), -ns^2;
   -ls^2, -(ls*ms), -(ls*ns), ls^2, ls*ms, ls*ns;
   -(ls*ms), -ms^2, -(ms*ns), ls*ms, ms^2, ms*ns;
   -(ls*ns), -(ms*ns), -ns^2, ls*ns, ms*ns, ns^2];
```

MatlabFiles\Chap4\SpaceTrussResults.m

```
function results = SpaceTrussResults(e, A, coord, disps)
% results = SpaceTrussResults(e, A, coord, disps)
% Compute space truss element results
% e = modulus of elasticity
% A = Area of cross-section
% coord = coordinates at the element ends
% disps = displacements at element ends

x1=coord(1,1); y1=coord(1,2); z1=coord(1,3);
x2=coord(2,1); y2=coord(2,2); z2=coord(2,3);
L=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);
ls=(x2-x1)/L; ms=(y2-y1)/L; ns=(z2-z1)/L;
T=[ls,ms,ns,0,0,0; 0,0,0,ls,ms,ns];
d = T*disps;
eps= (d(2)-d(1))/L;
sigma = e.*eps;
force = sigma.*A;
results=[eps, sigma, force];
```

Using these functions now we consider solution of the three bar truss. The steps are exactly those explained in similar examples in Chapter 1.

MatlabFiles\Chap4\ThreeBarSpaceTrussEx.m

```
% Three bar space truss example
a1 = 200; a2 = 600; e = 200000; P = 20000;
nodes = 1000*[.96, 1.92, 0; -1.44, 1.44, 0; 0, 0, 0; 0, 0, 2];
dof=3*length(nodes);
conn=[1,4; 2,4; 3,4];
lmm = [1, 2, 3, 10, 11, 12;
   4, 5, 6, 10, 11, 12;
   7, 8, 9, 10, 11, 12];
debc = [1:9];
ebcVals = zeros(length(debc),1);

%load vector
R = zeros(dof,1); R(11) = -P;

% Assemble global stiffness matrix
K=zeros(dof);
for i=1:2
   lm=lmm(i,:);
   con=conn(i,:);
   k=SpaceTrussElement(e, a1, nodes(con,:));
   K(lm, lm) = K(lm, lm) + k;
end
```

```
lm=lmm(3,:);
con=conn(3,:);
k=SPaceTrussElement(e, a2, nodes(con,:));
K(lm, lm) = K(lm, lm) + k

% Nodal solution and reactions
[d, reactions] = NodalSoln(K, R, debc, ebcVals)
results=[];
for i=1:2
   results = [results; SpaceTrussResults(e, a1, ...
         nodes(conn(i,:),:), d(lmm(i,:)))];
end
results = [results; SpaceTrussResults(e, a2, ...
       nodes(conn(3,:),:), d(lmm(3,:)))];
format short g
results
```

```
>> ThreeBarSpaceTrussEx

K =

  Columns 1 through 6

    1459.7    2919.3   -3040.9         0         0         0
    2919.3    5838.6   -6081.9         0         0         0
   -3040.9   -6081.9    6335.3         0         0         0
         0         0         0    3566.7   -3566.7    4953.8
         0         0         0   -3566.7    3566.7   -4953.8
         0         0         0    4953.8   -4953.8    6880.3
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
   -1459.7   -2919.3    3040.9   -3566.7    3566.7   -4953.8
   -2919.3   -5838.6    6081.9    3566.7   -3566.7    4953.8
    3040.9    6081.9   -6335.3   -4953.8    4953.8   -6880.3

  Columns 7 through 12

         0         0         0   -1459.7   -2919.3    3040.9
         0         0         0   -2919.3   -5838.6    6081.9
         0         0         0    3040.9    6081.9   -6335.3
         0         0         0   -3566.7    3566.7   -4953.8
         0         0         0    3566.7   -3566.7    4953.8
         0         0         0   -4953.8    4953.8   -6880.3
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0     60000         0         0    -60000
```

```
         0         0         0    5026.4   -647.45    1912.9
         0         0         0   -647.45    9405.4   -11036
         0         0    -60000    1912.9   -11036     73216


d =

         0
         0
         0
         0
         0
         0
         0
         0
         0
   -0.18705
    -2.592
   -0.3858


reactions =

    6666.7
     13333
    -13889
   -6666.7
    6666.7
   -9259.3
         0
         0
     23148


results =

  0.00050936     101.87      20375
  0.00033036     66.072      13214
  -0.0001929     -38.58     -23148
```