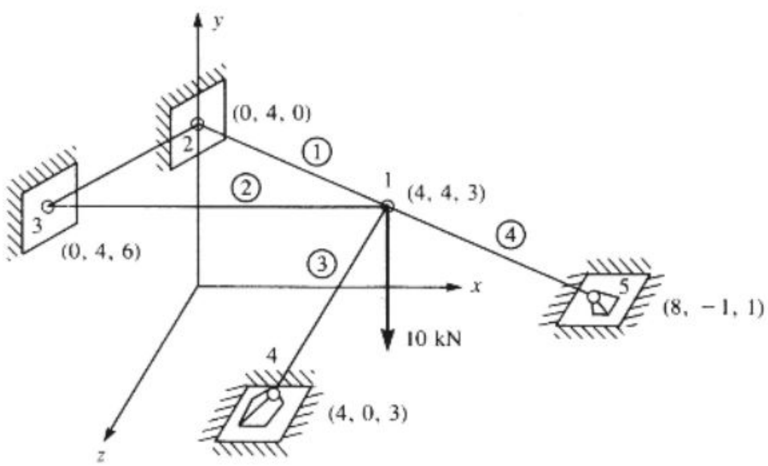
Contents

Problem Statement



Use Mathcad, Matlab, etc to Logan Problem 3-40.

Notes:

1. All elements are push-pull only.
2. Nodes 2,3,4,5 constrain the assembly in all directions.
3. What looks like an element drawn between nodes 2 and 3 is probably an e the drawing.

```
clc
clear all
```

Connectivity table, node coordinate matricies, and problem setup

```
connect = [2 1;
           3 1;
           4 1;
           5 1]; %connectivity table defining 2d trusses

node_coordinates = [4 4 3;
                    0 4 0;
                    0 4 6;
                    4 0 3
                    8 -1 1]; %coordinate table of node locations

F = [0 -10000 0]'; %left DOFs 4-15 out since those are unknown reactions

DOF = 3; %degrees of freedom per node
E = 2.1e11; %modulus of elasticity (Southern Pine)
A = 10e-4; %cross-sectional area of a truss
```

Calculation of element stiffness matrix and population of global stiffness matrix

```
num_nodes = size(node_coordinates, 1);
k_global = zeros(num_nodes*DOF);%initialization of empty global stiffness matrix
T = zeros(size(connect, 1), 6);
DOF_ids = zeros(6, size(connect, 1));

for i = 1:size(connect, 1)
    element_node1 = connect(i, 1); %transforms global node into node numbers relative to element
    element_node2 = connect(i, 2);
```

```
    %pick x coordinates of truss out of node coordinate matrix
    x1 = node_coordinates(element_node1, 1);
    x2 = node_coordinates(element_node2, 1);

    %pick y coordinates of truss out of node coordinate matrix
    y1 = node_coordinates(element_node1, 2);
    y2 = node_coordinates(element_node2, 2);

    %pick y coordinates of truss out of node coordinate matrix
    z1 = node_coordinates(element_node1, 3);
    z2 = node_coordinates(element_node2, 3);

    truss_length = sqrt((x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2); %truss length
    Cx = (x2-x1)/truss_length; %cosine of truss defining angle
    Cy = (y2-y1)/truss_length; %cosine of truss defining angle
    Cz = (z2-z1)/truss_length; %cosine of truss defining angle

    %DOF of nodes associated with truss
    DOFaddress = [(element_node1*3)-2 (element_node1*3)-1 (element_node1*3) ...
                  (element_node2*3)-2 (element_node2*3)-1 (element_node2*3)];

    rotation = [Cx Cy Cz 0 0 0;
                0 0 0 Cx Cy Cz]; %rotation matrix for stress calculation later

    k_local = [Cx 0;
               Cy 0;
               Cz 0;
               0 Cx;
               0 Cy;
               0 Cz];
    k_local = (A*E/truss_length)*k_local*[1 -1;-1 1]*rotation; %local stiffess matrix

    T(i,:) = E/truss_length*[-Cx -Cy -Cz Cx Cy Cz]; %rotation matrix for solving for stresses later
    DOF_ids(:,i) = DOFaddress'; %ids of DOFs associated with this truss

    for i = 1:6 %adds local stiffness matrix to global stiffness matrix
        for j = 1:6
            globalIn1 = DOFaddress(i);
            globalIn2 = DOFaddress(j);
            k_global(globalIn1, globalIn2) = k_global(globalIn1, globalIn2)+k_local(i, j);
        end
    end

end
k_global
```

```
k_global =

   1.0e+07 *

  Columns 1 through 7

    6.4891   -1.3913   -0.5565   -2.6880         0   -2.0160   -2.6880
   -1.3913    6.9892    0.6957         0         0         0         0
   -0.5565    0.6957    3.3023   -2.0160         0   -1.5120    2.0160
   -2.6880         0   -2.0160    2.6880         0    2.0160         0
         0         0         0         0         0         0         0
   -2.0160         0   -1.5120    2.0160         0    1.5120         0
   -2.6880         0    2.0160         0         0         0    2.6880
         0         0         0         0         0         0         0
    2.0160         0   -1.5120         0         0         0   -2.0160
         0         0         0         0         0         0         0
         0   -5.2500         0         0         0         0         0
         0         0         0         0         0         0         0
   -1.1131    1.3913    0.5565         0         0         0         0
    1.3913   -1.7392   -0.6957         0         0         0         0
    0.5565   -0.6957   -0.2783         0         0         0         0

  Columns 8 through 14

         0    2.0160         0         0         0   -1.1131    1.3913
         0         0         0   -5.2500         0    1.3913   -1.7392
         0   -1.5120         0         0         0    0.5565   -0.6957
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0   -2.0160         0         0         0         0         0
         0         0         0         0         0         0         0
         0    1.5120         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0    5.2500         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0    1.1131   -1.3913
         0         0         0         0         0   -1.3913    1.7392
         0         0         0         0         0   -0.5565    0.6957

  Column 15

    0.5565
```

```
    -0.6957
    -0.2783
          0
          0
          0
          0
          0
          0
          0
          0
          0
    -0.5565
     0.6957
     0.2783
```

Reducing global stiffness matrix (did it manually since the partition is obvious)

```
k_reduced = k_global(1:3, 1:3)
```

```
k_reduced =

   1.0e+07 *

    6.4891   -1.3913   -0.5565
   -1.3913    6.9892    0.6957
   -0.5565    0.6957    3.3023
```

Solving simultaneous equations for unconstrained displacements

```
U = (inv(k_reduced)*F);
U = [U;0;0;0;0;0;0;0;0;0;0;0;0];
```

Post-processing

Calculate stress in each element [Pa]

```
%displacements associated with each truss in matrix form
truss_displacements = zeros(4, 6);

%populate it according the the nodes each truss is connected to
for i=1:size(DOF_ids, 1)
    for j=1:size(DOF_ids, 2)
        truss_displacements(i, j) = U(DOF_ids(i,j));
    end
end

stress = diag(T*truss_displacements) %stress is in the diagonal entries cause linear algebra
```

```
stress =

   1.0e+06 *

   -0.3387
   -1.6933
   -7.9681
   -2.7261
```

Calculate strain in each element

```
strain = stress/E
```

```
strain =

   1.0e-04 *

   -0.0161
   -0.0806
   -0.3794
   -0.1298
```

Published with MATLAB® R2019b