Computational Mechanics 2
Homework 1 Solutions

Adithya Nair

AID23002

March 14, 2024

For all the followin three trusses (Figures 1 to 3), first draw the free body diagram of the structure nd write the equations for LMB and the AMB about point A. Then write a finite element method code in Octave to calculate the reaction forces at the supports. Finally, plot the deformed shape of the truss on top of the original shape. The Procedure For Question 1:

1. Input all given values, keep track of order of evaluations using the nodeVector. Loop through the structure using the localStiffnessGenerator function as well as the elementCount.
2. Apply boundary conditions, and find displacementVector by $K^{-1}f$
3. Add the displacementVector to the original positionVector, plot over both of them.

1 Figure 1

```
% Input
E = 11.4e6;
areaVector = 50e-4*ones(5,1);
lengthVector = [0.5;1.0;sqrt(5)/2;1/sqrt(2);0.5];
angleVector = [0;0;pi- atan(1/2);pi/4;pi/2];
positionVector = [-0.5;0;0;0;1;0;0;0.5]; % Found by taking B as origin

nodeVector = [1;2;2;3;3;4;1;4;2;4]; % AB, BC, CD, DA, DB
elementCount = 5;
nodeCount = 4;

displacementVector = zeros(2*nodeCount,1);
forceVector = zeros(2*nodeCount,1);
forceVector(6) = -1500;
forceVector(7) = -1000;


stiffnessGlobal = zeros(2*nodeCount,2*nodeCount);

function stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    stiffnessConstant = E*A/l;
    R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
    stiffnessMatrix = [stiffnessConstant 0 -stiffnessConstant 0; zeros(1,4);-stiffnessConsta
    R4 = [R zeros(2,2); zeros(2,2) R];
    stiffnessLocal = R4*stiffnessMatrix*R4';
end
function stiffnessLocalGlobal = local2Global(stiffnessLocal,node1,node2,nodeCount)
    stiffnessLocalGlobal = zeros(2*nodeCount,2*nodeCount);
    i = 2*node1 - 1;
    j = 2*node2 - 1;
    stiffnessLocalGlobal(i:(i+1),i:(i+1)) = stiffnessLocal(1:2,1:2);
    stiffnessLocalGlobal(i:(i+1),j:(j+1)) = stiffnessLocal(1:2,3:4);
    stiffnessLocalGlobal(j:(j+1),i:(i+1)) = stiffnessLocal(3:4,1:2);
    stiffnessLocalGlobal(j:(j+1),j:(j+1)) = stiffnessLocal(3:4,3:4);
end
```

```
function forceAxial = tension(stiffnessLocal,theta,displacementSelected)
    R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
    R4 = [R zeros(2,2); zeros(2,2) R];
    displacementLocal = R4'*displacementSelected;
    forceAxial = stiffnessLocal*displacementLocal;
end
% Looping through the entire structure.
A = 0; theta = 0; l = 0; stiffnessLocal = zeros(4,4); nodeAxialForces = zeros(nodeCount,1);
for element = 1:5 % For first three bars
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    nodeCounter = element*2 - 1;
    stiffnessLocalGlobal = local2Global(stiffnessLocal,nodeVector(nodeCounter),nodeVector(no
    stiffnessLocal
    stiffnessLocalGlobal
    stiffnessGlobal += stiffnessLocalGlobal;
end

% Applying Boundary Conditions,
forceEval = forceVector([3 5:end]);
displacementEval = displacementVector([3 5:end]);
stiffnessEval = stiffnessGlobal([3 5:end],[3 5:end]);
displacementEval = stiffnessEval\forceEval;

displacementVector([3 5:end]) = displacementEval;
forceVector = stiffnessGlobal*displacementVector;

forceAxial = zeros(4,1);
% 2 Finding Axial Forces
for element = 1:elementCount
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    nodeCounter = element*2 - 1;
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    displacementVectorSelected = zeros(4,1);
    displacementVectorSelected(1:2) = displacementVector(nodeVector(nodeCounter):(nodeVector
    displacementVectorSelected(3:4) = displacementVector(nodeVector(nodeCounter+1):(nodeVect
    forceAxial = tension(stiffnessLocal,theta,displacementVectorSelected);
    nodeAxialForces(nodeVector(nodeCounter)) += forceAxial(1);
    nodeAxialForces(nodeVector(nodeCounter+1)) += forceAxial(3);
end

% Calculating Stresses And Strains
```

```
stressVector = nodeAxialForces/A;
strainVector = displacementVector/l;
```

The calculated displacement vectors are:

$$\begin{bmatrix} 0 \\ 0 \\ -0.0263 \\ 0 \\ -0.0789 \\ -0.4964 \\ 0.0803 \\ -0.0307 \end{bmatrix}$$

The resulting forces are -

$$\begin{bmatrix} 1.0000e+03 \\ -2.0000e+03 \\ -9.0949e-13 \\ 3.5000e+03 \\ 4.5475e-13 \\ -1.5000e+03 \\ -1.0000e+03 \\ -3.6676e-14 \end{bmatrix}$$

2 Figure 2

```
    % Input
E = 200e6;
elementCount = 10;
nodeCount = 7;

areaVector = 8e-3*ones(elementCount,1);
% AB, BC, CD, DE, EG, GH, AG, BG, BE, CE
% 1 m down from C, so E is (6,3)
AB = 3;
BC = 3;
CD = 2;
```

```
DE = CD*asecd(26.7);
DH = sqrt(8^2 + 4^2);
GH = sqrt(3^2 + 1.5^2);
EG = DH - DE - GH;
AG = 2.5*acscd(39.8);
BG = 2.5;
BE = 1*acscd(18.4);
CE = 1;
lengthVector = [AB;BC;CD;DE;EG;GH;AG;BG;BE;CE];
angleVector = [0;0;0;180+26.7;180+26.7;180+26.7;360-39.8;90;360-18.4;90];
positionVector = [0;4;3;4;6;4;8;4;6;3;3;1.5;0;0] % ABCDEGH

nodeVector = [1;2;2;3;3;4;4;5;5;6;6;7;1;6;6;2;2;5;5;3]; % AB, BC, CD, DA, DB
displacementVector = zeros(2*nodeCount,1);
forceVector = zeros(2*nodeCount,1);
forceVector(7) = 12e3; % F2
forceVector(9) = +1.5e3; % F1
% DE, EG,

stiffnessGlobal = zeros(2*nodeCount,2*nodeCount);

function stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    stiffnessConstant = E*A/l;
    R = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];
    stiffnessMatrix = [stiffnessConstant 0 -stiffnessConstant 0; zeros(1,4);-stiffnessConsta
    R4 = [R zeros(2,2); zeros(2,2) R];
    stiffnessLocal = R4*stiffnessMatrix*R4';
end
function stiffnessLocalGlobal = local2Global(stiffnessLocal,node1,node2,nodeCount)
    stiffnessLocalGlobal = zeros(2*nodeCount,2*nodeCount);
    i = 2*node1 - 1;
    j = 2*node2 - 1;
    stiffnessLocalGlobal(i:(i+1),i:(i+1)) = stiffnessLocal(1:2,1:2);
    stiffnessLocalGlobal(i:(i+1),j:(j+1)) = stiffnessLocal(1:2,3:4);
    stiffnessLocalGlobal(j:(j+1),i:(i+1)) = stiffnessLocal(3:4,1:2);
    stiffnessLocalGlobal(j:(j+1),j:(j+1)) = stiffnessLocal(3:4,3:4);
end
function forceAxial = tension(stiffnessLocal,theta,displacementSelected)
    R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
    R4 = [R zeros(2,2); zeros(2,2) R];
    displacementLocal = R4'*displacementSelected;
    forceAxial = stiffnessLocal*displacementLocal;
end
% Looping through the entire structure.
A = 0; theta = 0; l = 0; stiffnessLocal = zeros(4,4);
for element = 1:elementCount % For first three bars
```

```
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    nodeCounter = element*2 - 1;
    stiffnessLocalGlobal = local2Global(stiffnessLocal,nodeVector(nodeCounter),nodeVector(nc
    stiffnessGlobal += stiffnessLocalGlobal;
end

% Applying Boundary Conditions,
range = [3:nodeCount*2-2]
forceEval = forceVector(range);
displacementEval = displacementVector(range);
stiffnessEval = stiffnessGlobal(range,range);
displacementEval = stiffnessEval\forceEval;

displacementVector(range) = displacementEval;
forceVector = stiffnessGlobal*displacementVector;

positionVectorNew = positionVector + displacementVector;
plot(positionVector(1:2:end),positionVector(2:2:end),'--or');
hold on;
plot(positionVectorNew(1:2:end),positionVectorNew(2:2:end),'--ob');

forceAxial = zeros(4,1);
% 2 Finding Axial Forces
nodeAxialForces =  zeros(nodeCount,1);
for element = 1:elementCount
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    nodeCounter = element*2 - 1;
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    displacementVectorSelected = zeros(4,1);
    displacementVectorSelected(1:2) = displacementVector(nodeVector(nodeCounter):(nodeVector
    displacementVectorSelected(3:4) = displacementVector(nodeVector(nodeCounter+1):(nodeVect
    forceAxial = tension(stiffnessLocal,theta,displacementVectorSelected);
    nodeAxialForces(nodeVector(nodeCounter)) += forceAxial(1);
    nodeAxialForces(nodeVector(nodeCounter+1)) += forceAxial(3);
end
% Calculating Stresses And Strains
stressVector = nodeAxialForces/A;
strainVector = displacementVector/l;
```

The calculated displacement vectors are -

$$\begin{bmatrix} 0 \\ 0 \\ 0.0242 \\ -0.0004 \\ 0.0467 \\ -0.1255 \\ 0.0617 \\ -0.2789 \\ -0.0155 \\ -0.1255 \\ 0.0009 \\ 0.0001 \\ 0 \\ 0 \end{bmatrix}$$

The calculated force vectors are -

$$\begin{bmatrix} -1.3128e+04 \\ 1.8728e+02 \\ 3.6380e-12 \\ 0 \\ -1.0914e-11 \\ 0 \\ 1.2000e+04 \\ -1.1369e-13 \\ 1.5000e+03 \\ -2.9104e-11 \\ 8.6622e-15 \\ -9.9476e-13 \\ -3.7237e+02 \\ -1.8728e+02 \end{bmatrix}$$

3 Figure 3

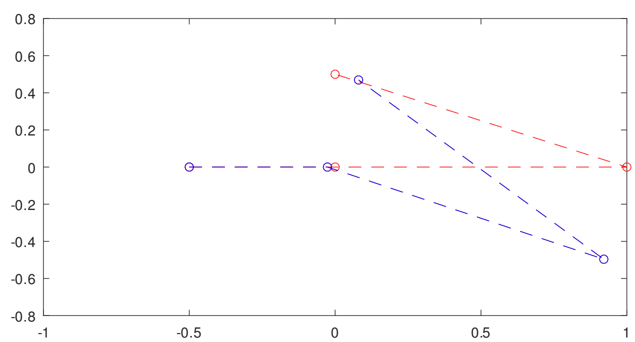
```
    % Input
E = 200e7;
elementCount = 9;
nodeCount = 6;

areaVector = 2.5e-3*ones(elementCount,1);
% Following from the node order
% 12 23 34 45 56 13 35 24 46
lengthVector = [4;3;5;3;5;5;4;4;4];
angleVector = [0;90;360-atan(3/4);90;360-atan(3/4);45;0;0;0];
positionVector = [0;0;4;0;4;3;8;0;8;3;12;0] % ABCDEGH

nodeVector = [1;2;2;3;3;4;4;5;5;6;1;3;3;5;2;4;4;6]; % AB, BC, CD, DA, DB
displacementVector = zeros(2*nodeCount,1);
forceVector = zeros(2*nodeCount,1);
forceVector(8) = 400; % F2
forceVector(7) = -1200; % F1

% DE, EG,

stiffnessGlobal = zeros(2*nodeCount,2*nodeCount);

function stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    stiffnessConstant = E*A/l;
    R = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];
    stiffnessMatrix = [stiffnessConstant 0 -stiffnessConstant 0; zeros(1,4);-stiffnessConsta
    R4 = [R zeros(2,2); zeros(2,2) R];
    stiffnessLocal = R4*stiffnessMatrix*R4';
end
function stiffnessLocalGlobal = local2Global(stiffnessLocal,node1,node2,nodeCount)
    stiffnessLocalGlobal = zeros(2*nodeCount,2*nodeCount);
    i = 2*node1 - 1;
    j = 2*node2 - 1;
    stiffnessLocalGlobal(i:(i+1),i:(i+1)) = stiffnessLocal(1:2,1:2);
    stiffnessLocalGlobal(i:(i+1),j:(j+1)) = stiffnessLocal(1:2,3:4);
    stiffnessLocalGlobal(j:(j+1),i:(i+1)) = stiffnessLocal(3:4,1:2);
    stiffnessLocalGlobal(j:(j+1),j:(j+1)) = stiffnessLocal(3:4,3:4);
end

function forceAxial = tension(stiffnessLocal,theta,displacementSelected)
    R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
    R4 = [R zeros(2,2); zeros(2,2) R];
    displacementLocal = R4'*displacementSelected;
    forceAxial = stiffnessLocal*displacementLocal;
```

```
end

% Looping through the entire structure.
A = 0; theta = 0; l = 0; stiffnessLocal = zeros(4,4);
for element = 1:elementCount % For first three bars
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    nodeCounter = element*2 - 1;
    stiffnessLocalGlobal = local2Global(stiffnessLocal,nodeVector(nodeCounter),nodeVector(no
    stiffnessGlobal += stiffnessLocalGlobal;
end

% Applying Boundary Conditions,
range = [3:nodeCount*2-2]
forceEval = forceVector(range);
displacementEval = displacementVector(range);
stiffnessEval = stiffnessGlobal(range,range);
displacementEval = stiffnessEval\forceEval;


displacementVector(range) = displacementEval;
forceVector = stiffnessGlobal*displacementVector;


positionVectorNew = positionVector + displacementVector;
plot(positionVector(1:2:end),positionVector(2:2:end),'--or');
hold on;
plot(positionVectorNew(1:2:end),positionVectorNew(2:2:end),'--ob');
forceAxial = zeros(4,1);
nodeAxialForces = zeros(nodeCount,1);
% 2 Finding Axial Forces
for element = 1:elementCount
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    nodeCounter = element*2 - 1;
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    displacementVectorSelected = zeros(4,1);
    displacementVectorSelected(1:2) = displacementVector(nodeVector(nodeCounter):(nodeVecto
    displacementVectorSelected(3:4) = displacementVector(nodeVector(nodeCounter+1):(nodeVect
    forceAxial = tension(stiffnessLocal,theta,displacementVectorSelected);
    nodeAxialForces(nodeVector(nodeCounter)) += forceAxial(1);
    nodeAxialForces(nodeVector(nodeCounter+1)) += forceAxial(3);
end
```

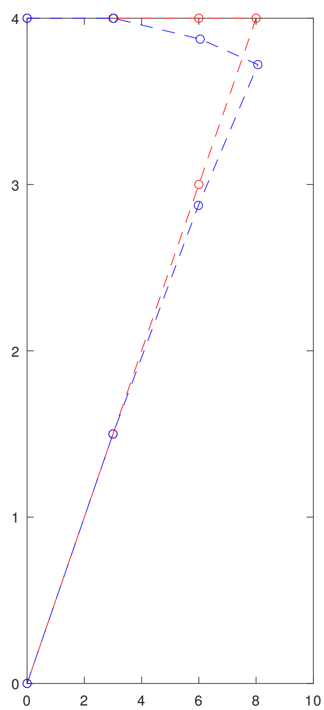
```
% Calculating Stresses And Strains
stressVector = nodeAxialForces/A;
strainVector = displacementVector/l;

% When maximum stress is 300 MPa, Forces need a new area,
stressVector = 300e6;
areaVector = stressVector./forceVector;
displacementVector = zeros(2*nodeCount,1);
forceVector = zeros(2*nodeCount,1);
forceVector(8) = 400; % F2
forceVector(7) = -1200; % F1

% DE, EG,

stiffnessGlobal = zeros(2*nodeCount,2*nodeCount);

function stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    stiffnessConstant = E*A/l;
    R = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];
    stiffnessMatrix = [stiffnessConstant 0 -stiffnessConstant 0; zeros(1,4);-stiffnessConsta
    R4 = [R zeros(2,2); zeros(2,2) R];
    stiffnessLocal = R4*stiffnessMatrix*R4';
end
function stiffnessLocalGlobal = local2Global(stiffnessLocal,node1,node2,nodeCount)
    stiffnessLocalGlobal = zeros(2*nodeCount,2*nodeCount);
    i = 2*node1 - 1;
    j = 2*node2 - 1;
    stiffnessLocalGlobal(i:(i+1),i:(i+1)) = stiffnessLocal(1:2,1:2);
    stiffnessLocalGlobal(i:(i+1),j:(j+1)) = stiffnessLocal(1:2,3:4);
    stiffnessLocalGlobal(j:(j+1),i:(i+1)) = stiffnessLocal(3:4,1:2);
    stiffnessLocalGlobal(j:(j+1),j:(j+1)) = stiffnessLocal(3:4,3:4);
end

function forceAxial = tension(stiffnessLocal,theta,displacementSelected)
    R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
    R4 = [R zeros(2,2); zeros(2,2) R];
    displacementLocal = R4'*displacementSelected;
    forceAxial = stiffnessLocal*displacementLocal;
end

% Looping through the entire structure.
A = 0; theta = 0; l = 0; stiffnessLocal = zeros(4,4);
for element = 1:elementCount % For first three bars
    A = areaVector(element);
    theta = angleVector(element);
```

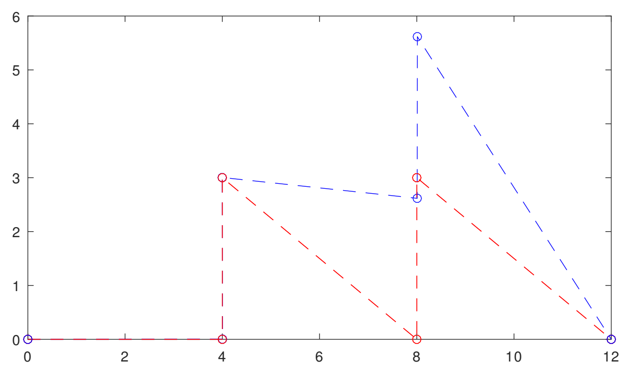
```
    l = lengthVector(element);
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    nodeCounter = element*2 - 1;
    stiffnessLocalGlobal = local2Global(stiffnessLocal,nodeVector(nodeCounter),nodeVector(no
    stiffnessGlobal += stiffnessLocalGlobal;
end

% Applying Boundary Conditions,
range = [3:nodeCount*2-2]
forceEval = forceVector(range);
displacementEval = displacementVector(range);
stiffnessEval = stiffnessGlobal(range,range);
displacementEval = stiffnessEval\forceEval;


displacementVector(range) = displacementEval;
forceVector = stiffnessGlobal*displacementVector;


positionVectorNew = positionVector + displacementVector;
plot(positionVector(1:2:end),positionVector(2:2:end),'--or');
hold on;
plot(positionVectorNew(1:2:end),positionVectorNew(2:2:end),'--ob');
forceAxial = zeros(4,1);
nodeAxialForces = zeros(nodeCount,1);
% 2 Finding Axial Forces
for element = 1:elementCount
    A = areaVector(element);
    theta = angleVector(element);
    l = lengthVector(element);
    nodeCounter = element*2 - 1;
    stiffnessLocal = localStiffnessGenerator(E,A,l,theta);
    displacementVectorSelected = zeros(4,1);
    displacementVectorSelected(1:2) = displacementVector(nodeVector(nodeCounter):(nodeVector
    displacementVectorSelected(3:4) = displacementVector(nodeVector(nodeCounter+1):(nodeVect
    forceAxial = tension(stiffnessLocal,theta,displacementVectorSelected);
    nodeAxialForces(nodeVector(nodeCounter)) += forceAxial(1);
    nodeAxialForces(nodeVector(nodeCounter+1)) += forceAxial(3);
end

% Calculating Stresses And Strains
stressVector = nodeAxialForces/A;
strainVector = displacementVector/l;
```

The calculated force vectors are : -



0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1


0
1
2
3
4
0
2
4
6
8
10


0
1
2
3
4
5
6
0
2
4
6
8
10
12