Table of Contents

MAE 150 HW 5 Problem 2

```
clear
close all
```

Initialization

Parameters

```
L = 2; % meters
H = L*sind(60); % meters

d = 0.030; % meters
A = pi*(d/2)^2; % square meters
E = 210*10^9; % GPa

% Define node positions
 % node: 1 2   3   4     5
nodes = [0 L 2*L L/2 3*L/2;  % x
         0 0   0   H     H]; % y

% Define element-node connections
 % element: 1 2 3 4 5 6 7
elements = [1 2 1 2 2 3 4;  % node i
            2 3 4 4 5 5 5]; % node j

n = length(nodes);
m = length(elements);
```

Construct stiffness matrices

Initialize arrays in loop

```
Ke = cell(m,1);
KG = zeros(2*n);

% Loop through elements
for el = 1:m
    i = elements(1,el);
    j = elements(2,el);

    p1 = nodes(:,i);
```

```
    p2 = nodes(:,j);

    beta = atan2(p2(2)-p1(2),p2(1)-p1(1));
    c = cos(beta);
    s = sin(beta);
    c2 = c^2;
    s2 = s^2;
    cs = c*s;
    Ke{el} = A*E/L*[ c2  cs -c2 -cs;
                cs  s2 -cs -s2;
               -c2 -cs  c2  cs;
               -cs -s2  cs  s2];

    Kt = zeros(2*n);
    Kt([2*i-1,2*i,2*j-1,2*j],[2*i-1,2*i,2*j-1,2*j]) = Ke{el};
    KG = KG + Kt;
end
```

Solving for unknowns

Define force vector

```
F = zeros(2,n); % Newtons
F(1,1) = 0;
F(2,1) = NaN;
F(2,2) = -20*10^3;
F(1,3) = NaN;
F(2,3) = NaN;
F(1,4) = -10*10^3;

% Indexing array for reduction (displacement = 0 at reaction forces)
redux = ~isnan(F);

% Reduce KG and F
Kr = KG(redux,redux);
Fr = F(redux);

% Solve for unknown displacements
dr = Kr\Fr;

% Define complete displacement vector
d = zeros(2*n,1);
d(redux) = dr;

% Reduce KG (reduce opposite rows)
Kr = KG(~redux,redux);

% Solve for unknown forces
Fr = Kr*dr;
```

Parts (a)-(d): Print results

```
fprintf('Part(a)\n\n')
```

```
fprintf('K_e1 =\n\n')
disp(Ke{1})
fprintf('K_e2 =\n\n')
disp(Ke{2})
fprintf('K_e3 =\n\n')
disp(Ke{3})
fprintf('K_e4 =\n\n')
disp(Ke{4})
fprintf('(Units: Pa/m)\n')

fprintf('\n\n\n')
fprintf('Part(b)\n\n')
fprintf('K_G =\n\n')
disp(KG)
fprintf('(Units: N/m)\n')

fprintf('\n\n\n')
fprintf('Part(c)\n\n')
fprintf('d =\n\n')
disp(d)
fprintf('(Units: m)\n')

fprintf('\n\n\n')
fprintf('Part(d)\n\n')
fprintf('F_1y =\n\n')
disp(Fr(1))
fprintf('F_3x =\n\n')
disp(Fr(2))
fprintf('F_3y =\n\n')
disp(Fr(3))

fprintf('(Units: N)\n')
```

Part (e): Plotting

```
scale = 500;
linewidth = 2;
fontsize = 14;
color1 = 'b';
color2 = 'r';

close all
figure
clf
hold on
for el = 1:length(elements)
    j = elements(:,el);
    x = nodes(1,j);
    y = nodes(2,j);
    plot(x,y,':','Color',color1,'Linewidth',linewidth);
end
nodes_new = nodes + scale*reshape(d,2,5);
for el = 1:length(elements)
```

```
    j = elements(:,el);
    x = nodes_new(1,j);
    y = nodes_new(2,j);
    plot(x,y,'-','Color',color2,'Linewidth',linewidth);
end
hold off
grid on
axis equal
axis padded
title('Truss Under External Load')
xlabel('X axis (m)')
ylabel('Y axis (m)')
set(gca,'FontSize',fontsize)
h = get(gca,'Children');
legend([h(1),h(m+1)],'Original',sprintf('Deformed (scaled by %dx)',scale))
```

Published with MATLAB® R2023a