Table of Contents

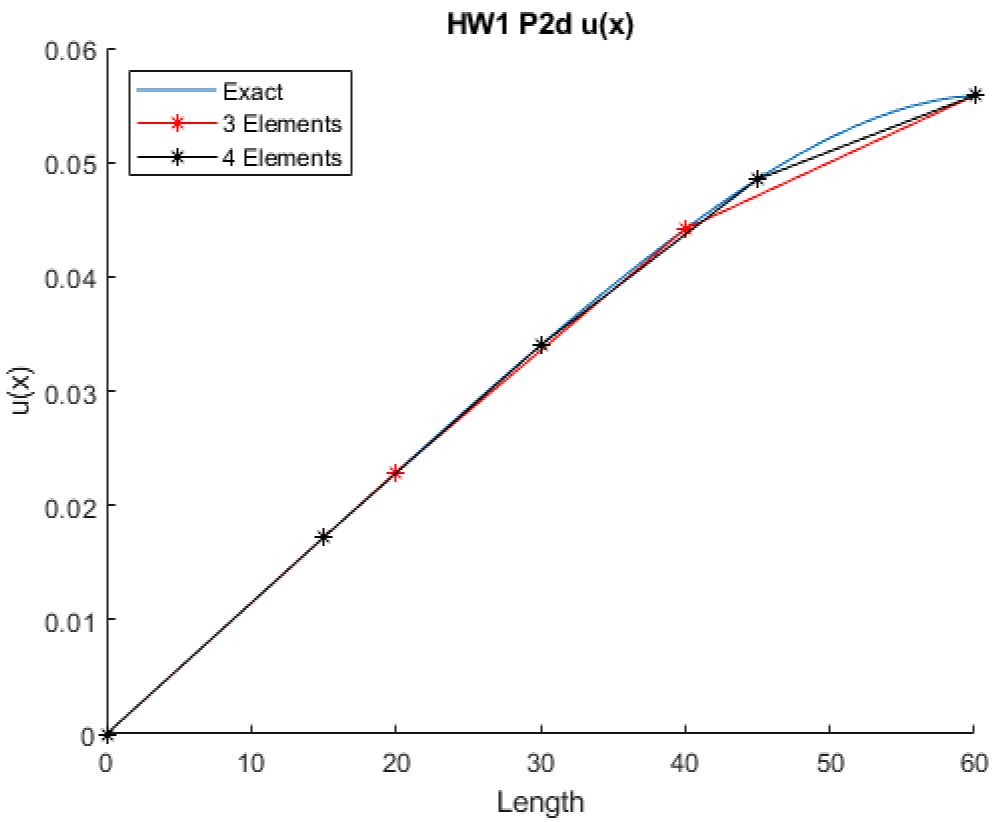

```
clc
clear all
close all
%Zhaoyi Jiang
%CESG504 HW1
```

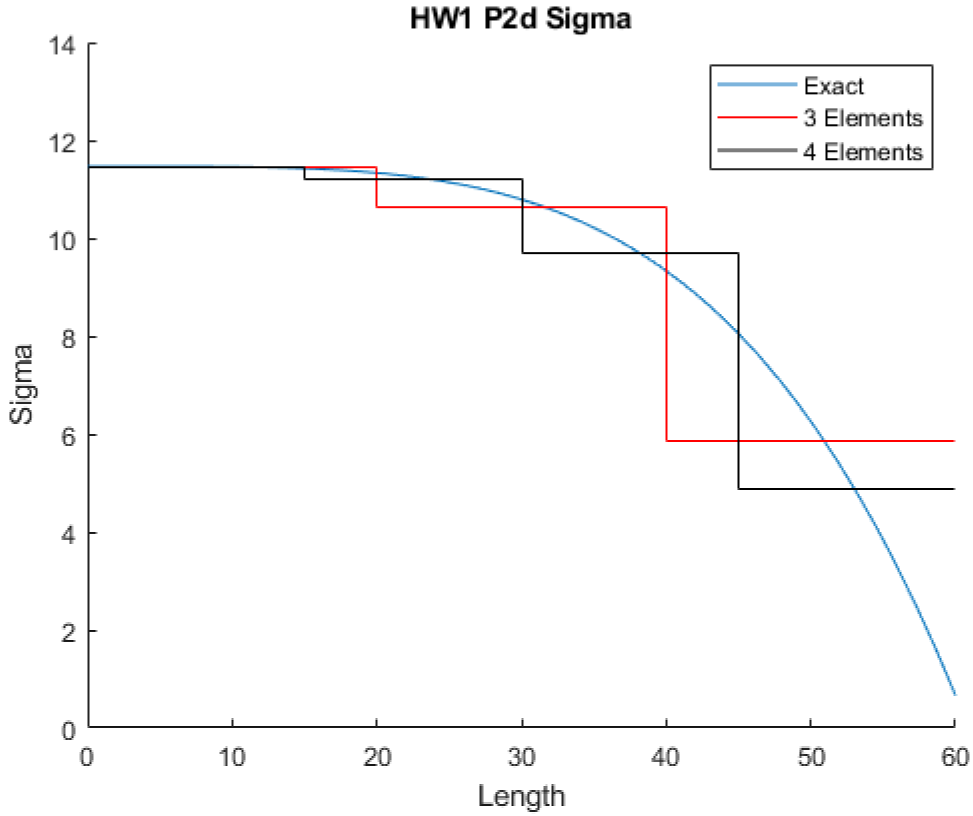
HW1 P2d

```
figure(1)
A=3;
E=10000000;
hold on
fplot(@(x) (34400*x-x^5/2000)/(A*E))
xlim([0 60])
ylim([0 0.06])
fem3=[0 0.02288 0.04416 0.05584];
sp3=[0 20 40 60];
fem4=[0 0.01719 0.03399 0.04852 0.05584];
sp4=[0 15 30 45 60];
plot(sp3,fem3,'-r*',sp4,fem4,'-k*')
title('HW1 P2d u(x)')
xlabel('Length')
ylabel('u(x)')
legend({'Exact','3 Elements','4 Elements'},'Location','northwest')
hold off

figure(2)
A=3;
E=10000;
hold on
fplot(@(x) (34400-x^4/400)/3000)
xlim([0 60])
ylim([0 14])
sigma3=[0.001145*E 0.001064*E 0.000584*E 0.000584*E];
sp3=[0 20 40 60];
sigma4=[0.001146*E 0.00112*E 0.000969*E 0.000488*E 0.000488*E];
sp4=[0 15 30 45 60];
stairs(sp3,sigma3,'-r')
stairs(sp4,sigma4,'-k')
title('HW1 P2d Sigma')
xlabel('Length')
```

```
ylabel('Sigma')
legend('Exact','3 Elements','4 Elements')
hold off
```

```
Warning: Function behaves unexpectedly on array inputs. To improve
 performance,
properly vectorize your function to return an output with the same
 size and
shape as the input arguments.
Warning: Function behaves unexpectedly on array inputs. To improve
 performance,
properly vectorize your function to return an output with the same
 size and
shape as the input arguments.
```





HW1 P4 2elements

```
clc
clear all
close all

% NOTE: YOU WILL HAVE TO EDIT THE FUNCTION getArea.m
    %                        %  %
%  TO ENSURE THAT THE A(x) FUNCTION MATCHES WHAT IS SPECIFIED IN
 PROBLEM 4    %

leg={'Exact'};
flag=1;
%BEGIN _____________________________ u s e r   i n p u t
 s_______________________________________%  %
                         %  %
                         %  %
A0=4; %in^2 <- area at LHS of rod
                           %  %
                         %  %
A1=2; %in^2 <- area at RHS of rod
                           %  %
                         %  %
L=24;      %m
                            %  %
```

```
E=10000000;  %Pa
                                    %  %
                               %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
%%%%%%%%% nodal information matrix %%%%%%%%%
                                %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
        % x location | fixity (1=fixed, 0=free)
                                %  %
n_info=     [0          1;
            L/2         0;%  %
            L         0];
                               %  %
%  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
%%%%%%%%% element information matrix %%%%%%%%%
                                %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
        % i node, j node, E
                                %  %
e_info=[1,     2,     E;
        2,     3,     E];
                                %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
numelem=size(e_info,1);  %number of elements
                                %  %
                               %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
%%%%%%%%%%%%%%%% LOADING %%%%%%%%%%%%%%%%%%%
                                %  %
P=50000;    %Applied Load
                                %  %
b=0; %m/s^2
                                %  %
                               %  %
P_loaded_Node=numelem+1; %<- in this case, the point load is applied
 at the end of the rod         %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
%%%%% EXACT SOLUTION for displacement (used for plotting) %%%%%%
                                %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                %  %
syms u(x) x
                                %  %
% u(x)=?; %<- exact solution (Change this using your acquired solution
 to problem 2a)             %  %
                               %  %
```

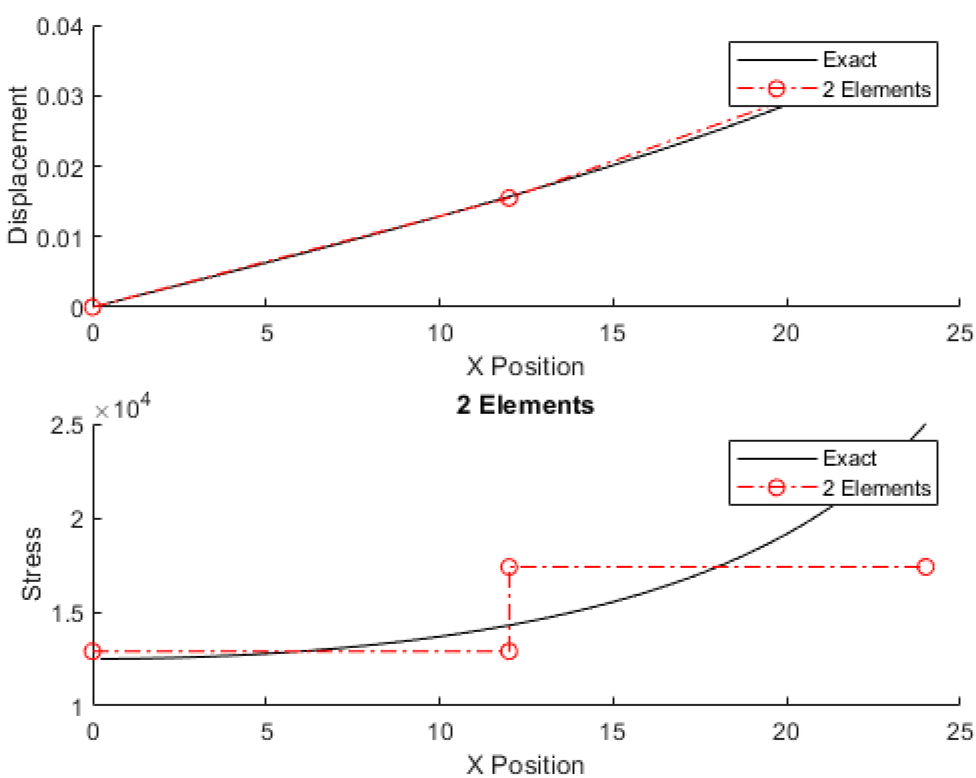
```
u(x)=(P*L/(E*(A0*(A1-A0))^0.5))*atan((x/L)*(1/sqrt(A0))*sqrt(A1-A0))
  %(exact soln - problem 4)   %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %  %
%END _______________________________ u s e r    i n p u t
 s_______________________________________% %



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DO NOT MODIFY BELOW, UNLESS YOU
 ARE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BRAVE/HAVE A COPY OF ORIGINAL
 SAVED


nnodes=numelem+1;     %number of nodes in the linear element system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5555

% initializing arrays for Stiffness Matrix and FVector = fixed end
 forces + applied loads

Kglob=zeros(nnodes,nnodes);
Fglob=zeros(nnodes,1);
uFE_soln=zeros(length(Fglob),1);

endNodesLocList=[]; %< used for plotting only
for i=1:size(e_info,1);

    currentElemCenterXloc=(n_info(e_info(i,1),1) +
 n_info(e_info(i,2),1))/2; %element center=(x location of i node + x
 location of j node)/2

    L_elem(i)=(n_info(e_info(i,2),1)-n_info(e_info(i,1),1)); %<-
 determining the length of the current element

    getArea %< getting element x-sec area by calling subroutine (EDIT
 THIS FOR NONPRISMATIC SECTION)

    Ke    %< getting element stiffness by calling subroutine (also
 included at end as a comment)
    KeLinear1DRoddivA(i,:,:)=KeLinear1DRod./ElA;
    % Global Stiffness Matrix assembly
    curr_doflist=[e_info(i,1),e_info(i,2)];
    Fe=[(b*L_elem/2);(b*L_elem/2)];
    for j=1:length(curr_doflist);
        for jj=1:length(curr_doflist);

 Kglob(curr_doflist(j),curr_doflist(jj))=Kglob(curr_doflist(j),curr_doflist(jj))+K
        end
        Fglob(curr_doflist(j))=Fglob(curr_doflist(j)) + Fe(j);
    end
```

```
 endNodesLocList=[endNodesLocList,n_info(e_info(i,1),1),n_info(e_info(i,2),1)];
  %< used for plotting only
end


xExact=L.*[1:100]./100;
uExact=u(xExact); %<- exact solution for the displacement
du=symfun(diff(u(x),x),x);
sExact=E*du(xExact);

constlin=find(n_info(:,2));

Kglob(constlin,:)=[];
Kglob(:,constlin)=[];

Fglob(P_loaded_Node)=Fglob(P_loaded_Node)+P;

Fglob(constlin,:)=[];

ulin=Kglob\Fglob; %<- solving the system of equations

uFE_soln(setdiff(1:end,constlin),:)=ulin;
eFE_soln=uFE_soln./L_elem;
for ijk=1:numelem;
    jkl=ijk+1;
Stress_FE(ijk,:)=abs(reshape(KeLinear1DRoddivA(ijk,:,:),[2
 2])*(uFE_soln(jkl-1:jkl)-uFE_soln(jkl-1)));
end
Stress_FE=reshape(Stress_FE',[numelem*2,1]);


if flag==1;
figure;
subplot(2,1,1)
hold on;
plot(xExact,uExact,'k');
plot(n_info(:,1),uFE_soln,['-.or']);
leg(:,end+1)={string(size(e_info,1))+' Elements'};
xlabel('X Position')
ylabel('Displacement');

subplot(2,1,2)
hold on;
plot(xExact,sExact,'k');
plot(endNodesLocList,Stress_FE,['-.or']);
xlabel('X Position')
ylabel('Stress');
flag=0;
```

```
else
subplot(2,1,1)
hold on;
plot(n_info(:,1),uFE_soln,['-.or']);
leg(:,end+1)={string(size(e_info,1))+' Elements'};

subplot(2,1,2)
hold on;
plot(endNodesLocList,Stress_FE,['-.or']);
end

title('2 Elements')
subplot(2,1,1)
legend(leg)
subplot(2,1,2)
legend(leg)
```

u(x) =

*-(2^(1/2)*atan((2^(1/2)*x*1i)/48)*3i)/100*



HW1 P4 8elements

```
clc
```

```
clear all
close all

% NOTE: YOU WILL HAVE TO EDIT THE FUNCTION getArea.m
    %                        %  %
%  TO ENSURE THAT THE A(x) FUNCTION MATCHES WHAT IS SPECIFIED IN
 PROBLEM 4    %

leg={'Exact'};
flag=1;
%BEGIN _____________________________ u s e r   i n p u t
 s_______________________________________%  %
                          %  %
                          %  %
A0=4; %in^2 <- area at LHS of rod
                            %  %
                          %  %
A1=2; %in^2 <- area at RHS of rod
                            %  %
                          %  %
L=24;      %m
                             %  %
E=10000000;  %Pa
                               %  %
                          %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%% nodal information matrix %%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
        % x location | fixity (1=fixed, 0=free)
                            %  %
n_info=     [0          1;
             L/8        0;
             L*2/8        0;
             L*3/8        0;
             L*4/8        0;
             L*5/8        0;
             L*6/8        0;
             L*7/8        0;%  %
             L        0];
                          %  %
%  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%%% element information matrix %%%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
        % i node, j node, E
                            %  %
e_info=[1,     2,     E;
        2,     3,     E
```

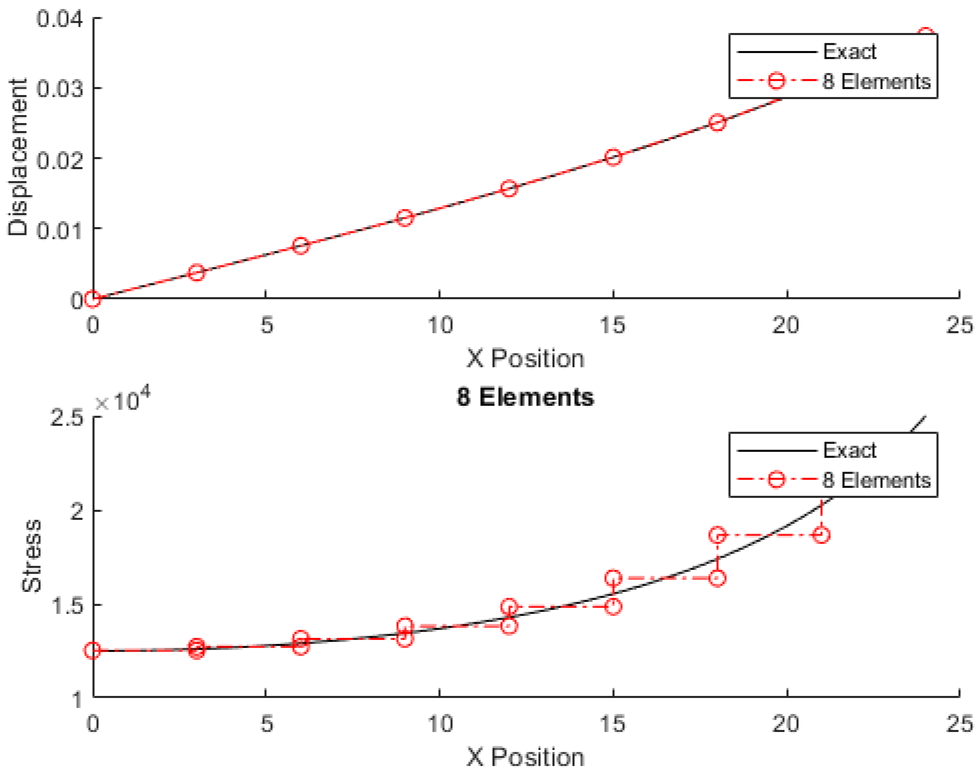
```
        3,      4,      E
        4,      5,      E
        5,      6,      E
        6,      7,      E
        7,      8,      E
        8,      9,      E];
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
numelem=size(e_info,1);  %number of elements
                            %  %
                          %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%% LOADING %%%%%%%%%%%%%%%%%%%
                            %  %
P=50000;    %Applied Load
                            %  %
b=0; %m/s^2
                            %  %
                          %  %
P_loaded_Node=numelem+1; %<- in this case, the point load is applied
 at the end of the rod        %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%% EXACT SOLUTION for displacement (used for plotting) %%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
syms u(x) x
                            %  %
% u(x)=?; %<- exact solution (Change this using your acquired solution
 to problem 2a)             %  %
                          %  %
u(x)=(P*L/(E*(A0*(A1-A0))^0.5))*atan((x/L)*(1/sqrt(A0))*sqrt(A1-A0))
  %(exact soln - problem 4)   %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% %  %
%END _______________________________ u s e r    i n p u t
 s_______________________________________%  %


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DO NOT MODIFY BELOW, UNLESS YOU
 ARE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BRAVE/HAVE A COPY OF ORIGINAL
 SAVED


nnodes=numelem+1;     %number of nodes in the linear element system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5555
```

```
% initializing arrays for Stiffness Matrix and FVector = fixed end
 forces + applied loads

Kglob=zeros(nnodes,nnodes);
Fglob=zeros(nnodes,1);
uFE_soln=zeros(length(Fglob),1);

endNodesLocList=[]; %< used for plotting only
for i=1:size(e_info,1);

    currentElemCenterXloc=(n_info(e_info(i,1),1) +
 n_info(e_info(i,2),1))/2; %element center=(x location of i node + x
 location of j node)/2

    L_elem(i)=(n_info(e_info(i,2),1)-n_info(e_info(i,1),1)); %<-
 determining the length of the current element

    getArea %< getting element x-sec area by calling subroutine (EDIT
 THIS FOR NONPRISMATIC SECTION)

    Ke    %< getting element stiffness by calling subroutine (also
 included at end as a comment)
    KeLinear1DRoddivA(i,:,:)=KeLinear1DRod./ElA;
    % Global Stiffness Matrix assembly
    curr_doflist=[e_info(i,1),e_info(i,2)];
    Fe=[(b*L_elem/2);(b*L_elem/2)];
    for j=1:length(curr_doflist);
        for jj=1:length(curr_doflist);

 Kglob(curr_doflist(j),curr_doflist(jj))=Kglob(curr_doflist(j),curr_doflist(jj))+K
        end
        Fglob(curr_doflist(j))=Fglob(curr_doflist(j)) + Fe(j);
    end





 endNodesLocList=[endNodesLocList,n_info(e_info(i,1),1),n_info(e_info(i,2),1)];
  %< used for plotting only
end


xExact=L.*[1:100]./100;
uExact=u(xExact); %<- exact solution for the displacement
du=symfun(diff(u(x),x),x);
sExact=E*du(xExact);

constlin=find(n_info(:,2));

Kglob(constlin,:)=[];
Kglob(:,constlin)=[];
```

```
Fglob(P_loaded_Node)=Fglob(P_loaded_Node)+P;

Fglob(constlin,:)=[];

ulin=Kglob\Fglob; %<- solving the system of equations

uFE_soln(setdiff(1:end,constlin),:)=ulin;
eFE_soln=uFE_soln./L_elem;
for ijk=1:numelem;
    jkl=ijk+1;
Stress_FE(ijk,:)=abs(reshape(KeLinear1DRoddivA(ijk,:,:),[2
 2])*(uFE_soln(jkl-1:jkl)-uFE_soln(jkl-1)));
end
Stress_FE=reshape(Stress_FE',[numelem*2,1]);


if flag==1;
figure;
subplot(2,1,1)
hold on;
plot(xExact,uExact,'k');
plot(n_info(:,1),uFE_soln,['-.or']);
leg(:,end+1)={string(size(e_info,1))+' Elements'};
xlabel('X Position')
ylabel('Displacement');

subplot(2,1,2)
hold on;
plot(xExact,sExact,'k');
plot(endNodesLocList,Stress_FE,['-.or']);
xlabel('X Position')
ylabel('Stress');
flag=0;
else
subplot(2,1,1)
hold on;
plot(n_info(:,1),uFE_soln,['-.or']);
leg(:,end+1)={string(size(e_info,1))+' Elements'};

subplot(2,1,2)
hold on;
plot(endNodesLocList,Stress_FE,['-.or']);
end

title('8 Elements')
subplot(2,1,1)
legend(leg)
subplot(2,1,2)
legend(leg)
```

u(x) =

```
-(2^(1/2)*atan((2^(1/2)*x*1i)/48)*3i)/100
```



HW1 P4 20elements

```
clc
clear all
close all

% NOTE: YOU WILL HAVE TO EDIT THE FUNCTION getArea.m
    %                        %  %
%  TO ENSURE THAT THE A(x) FUNCTION MATCHES WHAT IS SPECIFIED IN
 PROBLEM 4     %

leg={'Exact'};
flag=1;
%BEGIN _____________________________ u s e r    i n p u t
 s_______________________________________%  %
                       %  %
                       %  %
A0=4; %in^2 <- area at LHS of rod
                         %  %
                       %  %
A1=2; %in^2 <- area at RHS of rod
                         %  %
                       %  %
```

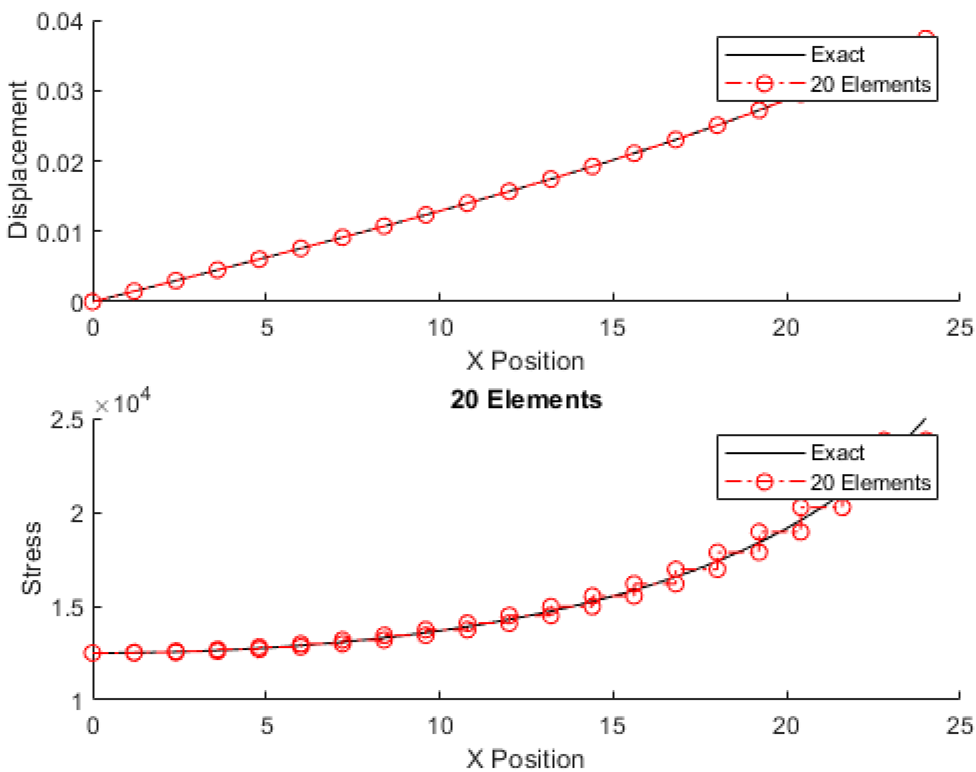
```
L=24;       %m
                                    %  %
E=10000000;  %Pa
                                      %  %
                                 %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                   %  %
%%%%%%%%% nodal information matrix %%%%%%%%%
                                   %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                   %  %
        % x location | fixity (1=fixed, 0=free)
                                   %  %
n_info=     [0           1;
            L*1/20         0;
            L*2/20         0;
            L*3/20         0;
            L*4/20         0;
            L*5/20         0;
            L*6/20         0;
            L*7/20         0;
            L*8/20         0;
            L*9/20         0;
            L*10/20         0;
            L*11/20         0;
            L*12/20         0;
            L*13/20         0;
            L*14/20         0;
            L*15/20         0;
            L*16/20         0;
            L*17/20         0;
            L*18/20         0;
            L*19/20         0;
            L         0;];
                                  %  %
%  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                   %  %
%%%%%%%%% element information matrix %%%%%%%%%
                                   %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                   %  %
        % i node, j node, E
                                   %  %
e_info=[1,      2,      E;
        2,      3,      E
        3,      4,      E
        4,      5,      E
        5,      6,      E
        6,      7,      E
        7,      8,      E
        8,      9,      E
        9,      10,      E
        10,      11,      E
```

```
        11,     12,     E
        12,     13,     E
        13,     14,     E
        14,     15,     E
        15,     16,     E
        16,     17,     E
        17,     18,     E
        18,     19,     E
        19,     20,     E
        20,     21,     E];
                              %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
numelem=size(e_info,1);  %number of elements
                            %  %
                          %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%% LOADING %%%%%%%%%%%%%%%%%%%
                            %  %
P=50000;    %Applied Load
                            %  %
b=0; %m/s^2
                            %  %
                          %  %
P_loaded_Node=numelem+1; %<- in this case, the point load is applied
 at the end of the rod        %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%% EXACT SOLUTION for displacement (used for plotting) %%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
syms u(x) x
                            %  %
% u(x)=?; %<- exact solution (Change this using your acquired solution
 to problem 2a)             %  %
                          %  %
u(x)=(P*L/(E*(A0*(A1-A0))^0.5))*atan((x/L)*(1/sqrt(A0))*sqrt(A1-A0))
  %(exact soln - problem 4)   %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% %  %
%END _______________________________ u s e r   i n p u t
 s_______________________________________%  %



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DO NOT MODIFY BELOW, UNLESS YOU
 ARE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BRAVE/HAVE A COPY OF ORIGINAL
 SAVED
```

```
nnodes=numelem+1;     %number of nodes in the linear element system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5555

% initializing arrays for Stiffness Matrix and FVector = fixed end
 forces + applied loads

Kglob=zeros(nnodes,nnodes);
Fglob=zeros(nnodes,1);
uFE_soln=zeros(length(Fglob),1);

endNodesLocList=[]; %< used for plotting only
for i=1:size(e_info,1);

    currentElemCenterXloc=(n_info(e_info(i,1),1) +
 n_info(e_info(i,2),1))/2; %element center=(x location of i node + x
 location of j node)/2

    L_elem(i)=(n_info(e_info(i,2),1)-n_info(e_info(i,1),1)); %<-
 determining the length of the current element

    getArea %< getting element x-sec area by calling subroutine (EDIT
 THIS FOR NONPRISMATIC SECTION)

    Ke    %< getting element stiffness by calling subroutine (also
 included at end as a comment)
    KeLinear1DRoddivA(i,:,:)=KeLinear1DRod./ElA;
    % Global Stiffness Matrix assembly
    curr_doflist=[e_info(i,1),e_info(i,2)];
    Fe=[(b*L_elem/2);(b*L_elem/2)];
    for j=1:length(curr_doflist);
        for jj=1:length(curr_doflist);

 Kglob(curr_doflist(j),curr_doflist(jj))=Kglob(curr_doflist(j),curr_doflist(jj))+K
        end
        Fglob(curr_doflist(j))=Fglob(curr_doflist(j)) + Fe(j);
    end







 endNodesLocList=[endNodesLocList,n_info(e_info(i,1),1),n_info(e_info(i,2),1)];
  %< used for plotting only
end


xExact=L.*[1:100]./100;
uExact=u(xExact); %<- exact solution for the displacement
du=symfun(diff(u(x),x),x);
sExact=E*du(xExact);
```

```
constlin=find(n_info(:,2));

Kglob(constlin,:)=[];
Kglob(:,constlin)=[];

Fglob(P_loaded_Node)=Fglob(P_loaded_Node)+P;

Fglob(constlin,:)=[];

ulin=Kglob\Fglob; %<- solving the system of equations

uFE_soln(setdiff(1:end,constlin),:)=ulin;
eFE_soln=uFE_soln./L_elem;
for ijk=1:numelem;
    jkl=ijk+1;
Stress_FE(ijk,:)=abs(reshape(KeLinear1DRoddivA(ijk,:,:),[2
 2])*(uFE_soln(jkl-1:jkl)-uFE_soln(jkl-1)));
end
Stress_FE=reshape(Stress_FE',[numelem*2,1]);


if flag==1;
figure;
subplot(2,1,1)
hold on;
plot(xExact,uExact,'k');
plot(n_info(:,1),uFE_soln,['-.or']);
leg(:,end+1)={string(size(e_info,1))+' Elements'};
xlabel('X Position')
ylabel('Displacement');

subplot(2,1,2)
hold on;
plot(xExact,sExact,'k');
plot(endNodesLocList,Stress_FE,['-.or']);
xlabel('X Position')
ylabel('Stress');
flag=0;
else
subplot(2,1,1)
hold on;
plot(n_info(:,1),uFE_soln,['-.or']);
leg(:,end+1)={string(size(e_info,1))+' Elements'};

subplot(2,1,2)
hold on;
plot(endNodesLocList,Stress_FE,['-.or']);
end
title('20 Elements')

subplot(2,1,1)
legend(leg)
subplot(2,1,2)
legend(leg)
```

```
u(x) =

-(2^(1/2)*atan((2^(1/2)*x*1i)/48)*3i)/100
```



Published with MATLAB® R2020b