FINITE ELEMENTS IN ENGINEERING (MAE 598)

PROJECT 1 REPORT

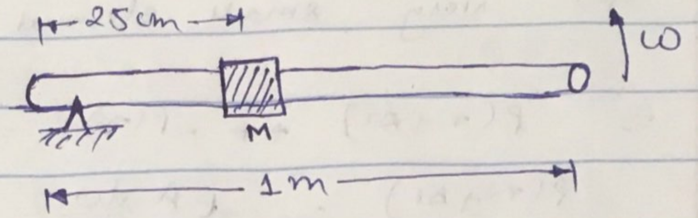
Prof. Jay Oswald

SUBMITTED BY

Varun Agrawal

(1215318065)

ANALYTICAL

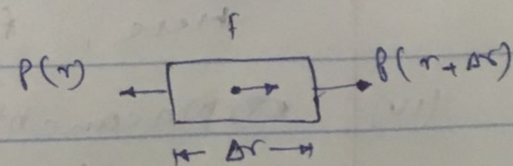


Dia. of rod = 1m, density = 8200 kg/m³
N = 2000 rpm
Mass of collar, M = 10 kg at 25 cm from fixed end.

$E(T) = 201.5 - 0.056T$ E = Young's Modulus, GPa
T = temperature, °C
T = 150°C at fixed end & T = 1200°C at free end

⇒ Temperature varying linearly along 'x', using line equation,
Points → (0, 150) & (1, 1200)
∴ $y - 150 = 1050x$
$y = 1050x + 150$
or $\boxed{T = 1050r + 150}$

for a small element at a distance 'r' m,
r_m = radius at which collar is attached.



$$P(r+\Delta r) - P(r) + f = \Delta r\, \rho A \omega^2 r_m$$

$P \rightarrow$ load,

for very small element, $\Delta r = 0$,

$$\Rightarrow \quad P(r+\Delta r) + F - P(r) = 0$$

$$P(r+\Delta r) = EA\frac{du}{dr} \quad \text{for } r_m < r < L$$

$$\& \quad P(r) = EA\frac{du}{dr} \quad \text{for } 0 < r < r_m$$

$$f = M r_m \omega^2 \quad (\text{Body force})$$

$$\Rightarrow \frac{d}{dr}\left(EA\frac{du}{dr}\right) = -\rho A \omega^2 r\, dr$$

Boundary conditions :

(i) $E\left.\frac{du}{dr}\right|_{r=L} = 0$ [No force at free end]

(ii) $u(0) = 0$ (no displacement at fixed point)

(iii) $EA\left.\frac{du}{dr}\right|_{r_m^+} = EA\left.\frac{du}{dr}\right|_{r_m^-} - F$ (Jump property)

Here, $f = M r_m \omega^2$

(iv) Displacement at r_m on both sides is equal.

$$u(r_m^+) = u(r_m^-)$$

Now,

for $\quad 0 < r < r_m \quad , \quad r_m < r < L$

$$EA\frac{du}{dr} = \int -\rho A\omega^2 r\,dr \qquad\qquad EA\frac{du}{dr} = -\int \rho A\omega^2 r\,dr$$

$$\Rightarrow EA\frac{du}{dr} = -\frac{\rho A\omega^2 r^2}{2} + C_1 \quad \text{— (A)} \qquad\qquad EA\frac{du}{dr} = -\frac{\rho A\omega^2 r^2}{2} + C_2 \quad \text{— (B)}$$

for eqn (B), at $r = L$, $EA\frac{du}{dr} = 0$

$$\Rightarrow 0 = -\frac{\rho A\omega^2 L^2}{2} + C_2$$

$$\Rightarrow \boxed{C_2 = \frac{\rho A\omega^2 L^2}{2}}$$

$$\Rightarrow EA\frac{du}{dr} = \frac{\rho A\omega^2}{2}(L^2 - r^2)$$

for eqn (A), using B.C. (iii)

$$\Rightarrow C_1 - \cancel{\frac{\rho A\omega^2 r_m^2}{2}} - M r_m\omega^2 = +\frac{\rho A\omega^2 L^2}{2} - \cancel{\frac{\rho A\omega^2 r_m^2}{2}}$$

$$\boxed{C_1 = \frac{\rho A\omega^2 L^2}{2} + M r_m\omega^2}$$

$$\Rightarrow EA\frac{du}{dr} = \frac{\rho A\omega^2}{2}(L^2 - r^2) + M r_m\omega^2$$

Scanned with CamScanner

1 for the strain,

$0 < r < r_m$

$$u(r) = \frac{1}{EA}\int \left\{ \frac{\rho A \omega^2}{2}(L^2 - r^2) + M r_m \omega^2 \right\} dr + C_3$$

$r_m < r < L$

$$u(r) = \frac{1}{EA}\int \frac{\rho A \omega^2}{2}(L^2 - r^2)\, dr + C_4$$

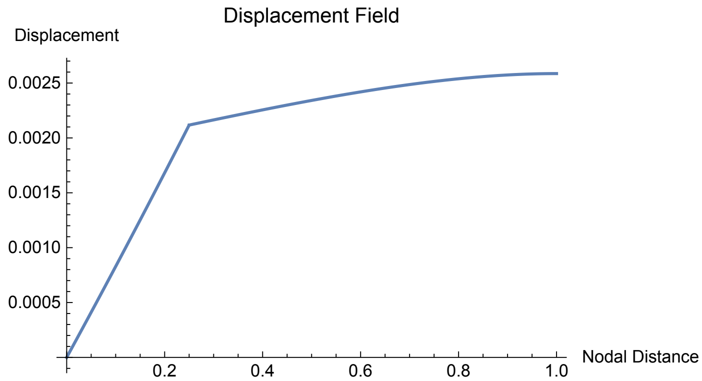
Using Mathematica,

Scanned with CamScanner

Analytical Solution

```
(*dudr1 = strain from r = 0 to rm, dudr2 = strain from r = rm to R*)

In[2]:= dudr1 = 1/(YM * A) (Integrate[-rho * A * w^2 * r, r] + c1);
        dudr2 = 1/(YM * A) (Integrate[-rho * A * w^2 * r, r] + c2);

(*T is the given variation of temperature in the rod,
YM is the given young's modulus varying through the rod*)

In[4]:= T = T0 + dTdr * r;
        YM = Y0 + dYdT * T;
        (*u1 is the displacement from r = 0 to rm, u2 is the displacement from r = rm to R*)
        u1 = Integrate[dudr1, r] + c3;
        u2 = Integrate[dudr2, r] + c4;

(*boundary conditions*)

In[8]:= bc1 = (u1 /. r → 0) == 0;
        bc2 = (dudr2 /. r → R) == 0;
        bc3 = (u1 /. r → rm) == (u2 /. r → rm);
        bc4 = (YM * A * dudr1 /. r → rm) == (YM * A * dudr2 /. r → rm) + M * rm * w^2;
        (*Solving for constants c1, c2, c3, c4 using boundary conditions*)
        soln = Simplify[Solve[bc1 && bc2 && bc3 && bc4, {c1, c2, c3, c4}][[1]]];

(* R = Radius of rod, rm = radius at which collar is attached to the rod,
w = angular velocity of rod, rho = density of rod,
A = cross-sectional area of rod, M = mass of collar*)

(*value of the parameters*)

In[13]:= param = {T0 → 150, dTdr → 1050, Y0 → 201.5 * 10^9, dYdT → -0.056 * 10^9,
           R → 1, rm → 0.25, w → 2000 * 2 * π/60, rho → 8200, A → 0.01^2 * π/4, M → 10};

(*displacement in the rod or the displacement field*)

In[14]:= u = Piecewise[{{u1, r < rm}}, u2] /. soln /. param;

(*Plot of the displacement field*)
```

In[15]:=
```
Plot[u, {r, 0, 1}, PlotLabel → "Displacement Field",
 AxesLabel → {"Nodal Distance", "Displacement"}]
```

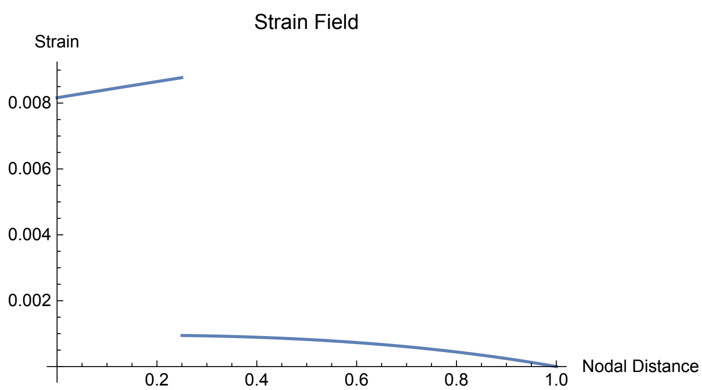
Out[15]=



```
(*Strain in the rod or the strain field*)
```

In[16]:=
```
strain = Piecewise[{{dudr1, r < rm}}, dudr2] /. soln /. param;
```

```
(*Plot of the strain field*)
```

In[17]:=
```
Plot[strain, {r, 0, 1}, PlotLabel → "Strain Field",
 AxesLabel → {"Nodal Distance", "Strain"}]
```

Out[17]=



```
(*defining stress with the varying r*)
```

In[18]:=
```
stress1 = dudr1 * YM;
```

In[19]:=
```
stress2 = dudr2  * YM;
```

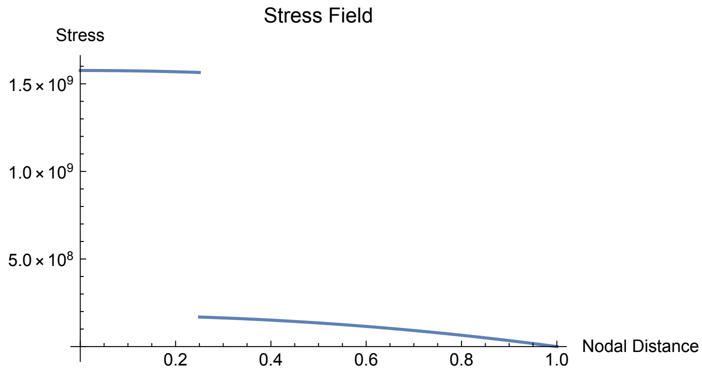
```
(*Stress in the rod*)
```

In[20]:=
```
stress = Piecewise[{{stress1, r < rm}}, stress2] /. soln /. param;
```

```
(*Plot of stress*)
```

In[21]:=
```
Plot[stress, {r, 0, 1}, PlotLabel → "Stress Field",
 AxesLabel → {"Nodal Distance", "Stress"}]
```

Out[21]=



Displacement and Strain Field for Analytical, MATLAB, and ABAQUS on same plot for 20 linear elements

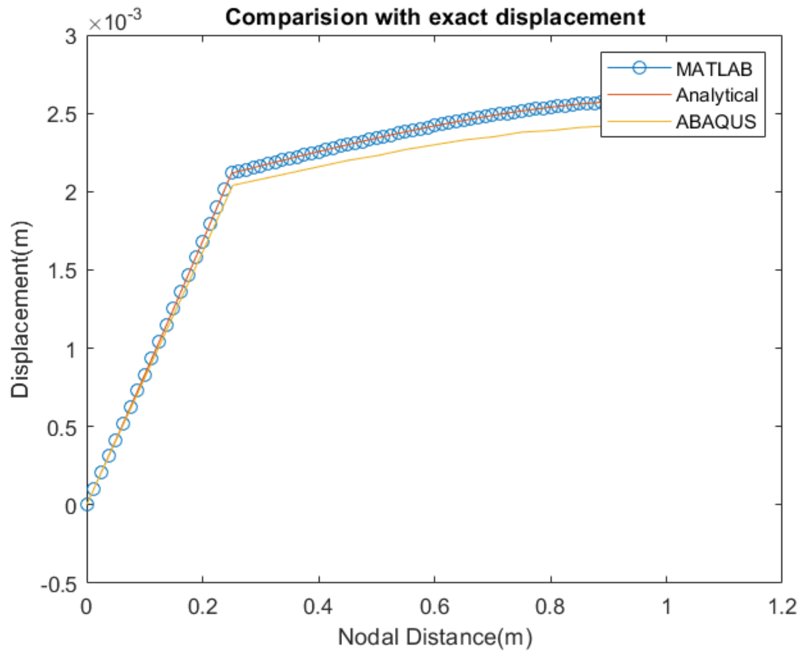


Figure 1

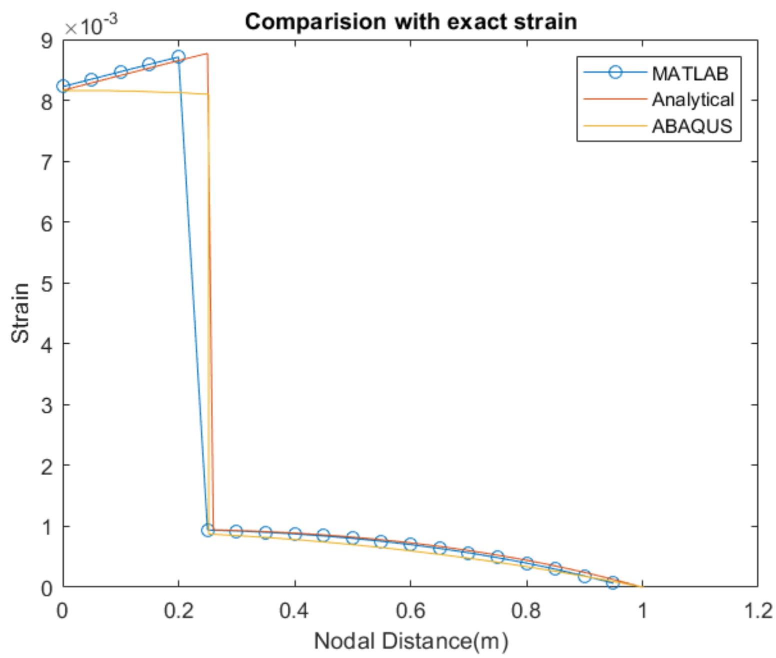


Figure 2

Displacement and Strain Field for Analytical, MATLAB, and ABAQUS on same plot for 20 quadratic elements

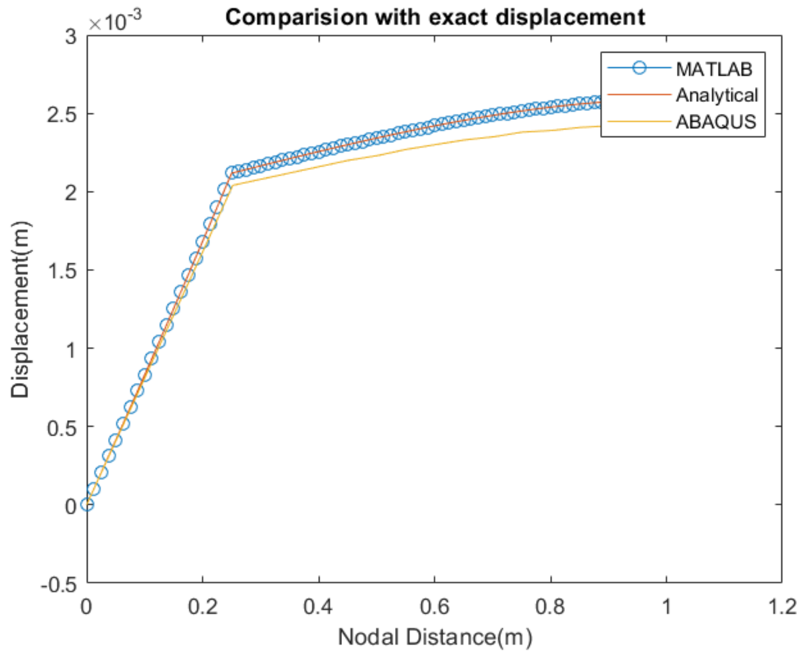


Figure 3

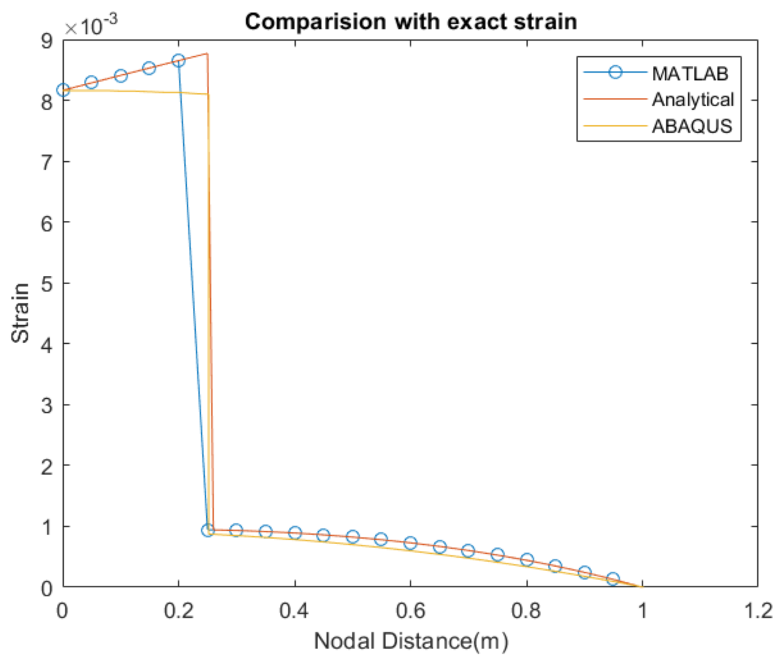


Figure 4

Displacement field for 4 Linear Elements

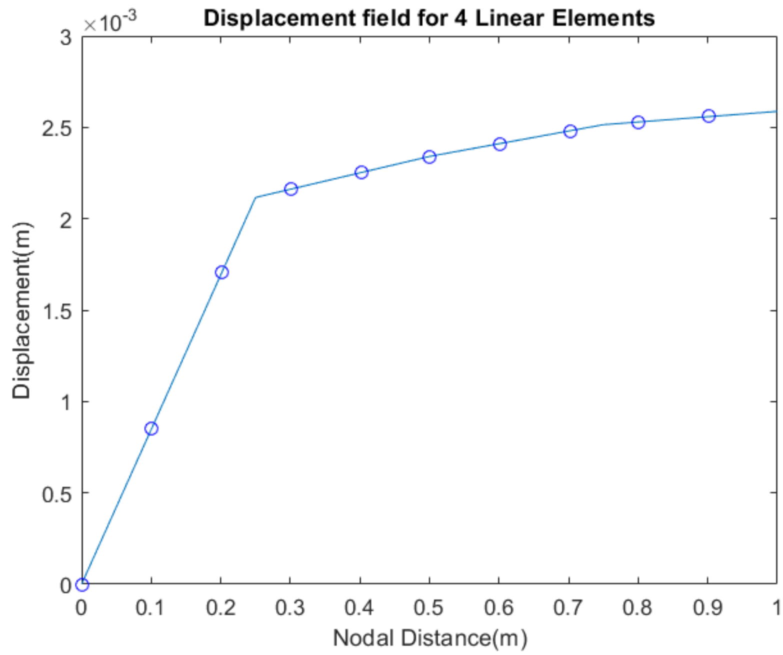


Figure 5

Strain field for 4 Linear Elements

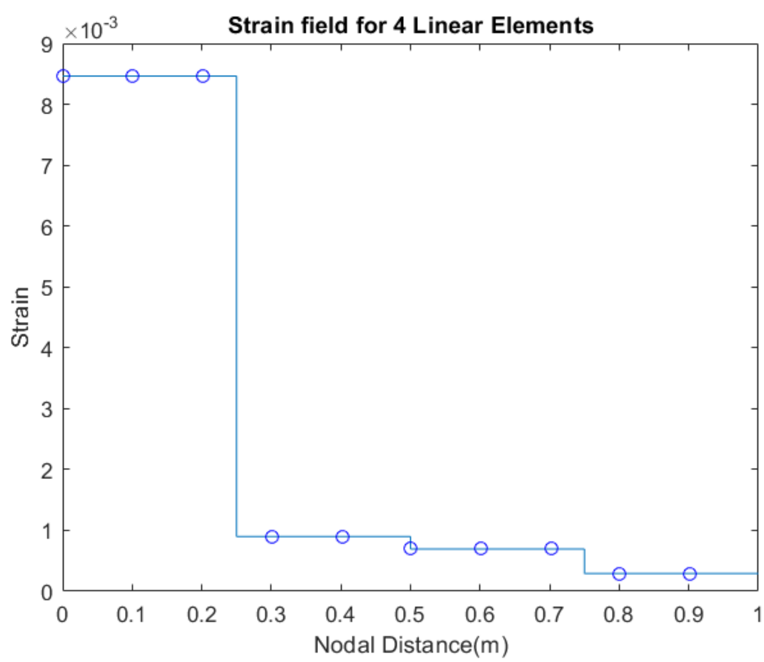


Figure 6

Displacement field for 20 Linear Elements

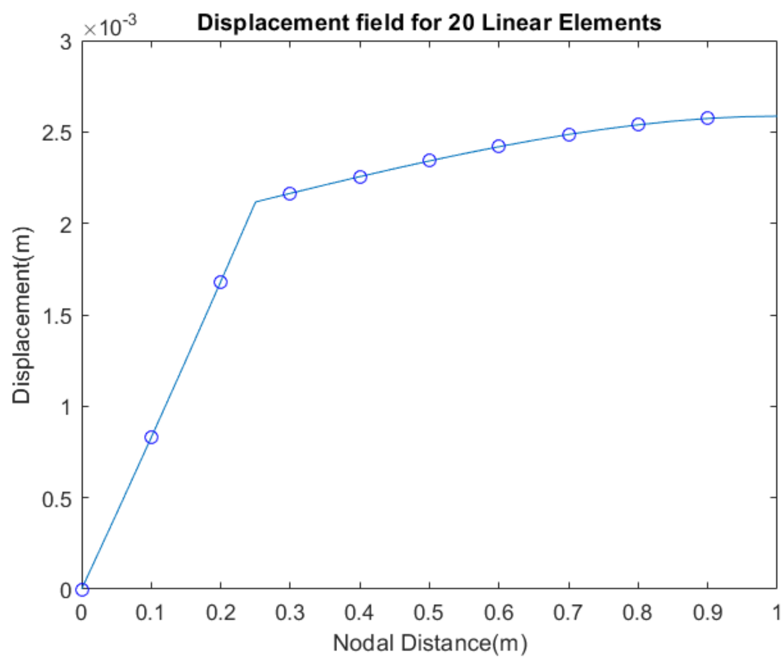


Figure 7

Strain field for 20 Linear Elements

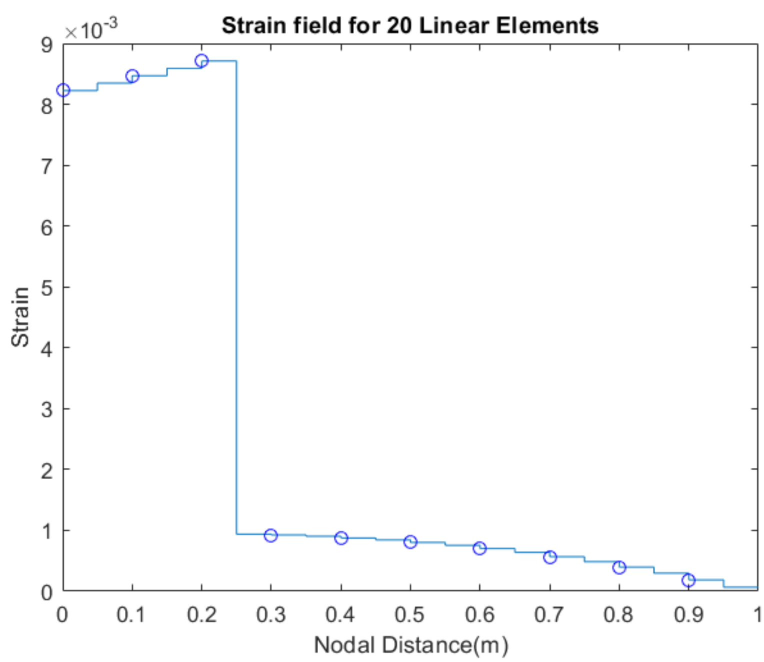


Figure 8

Displacement field for 4 Quadratic Elements

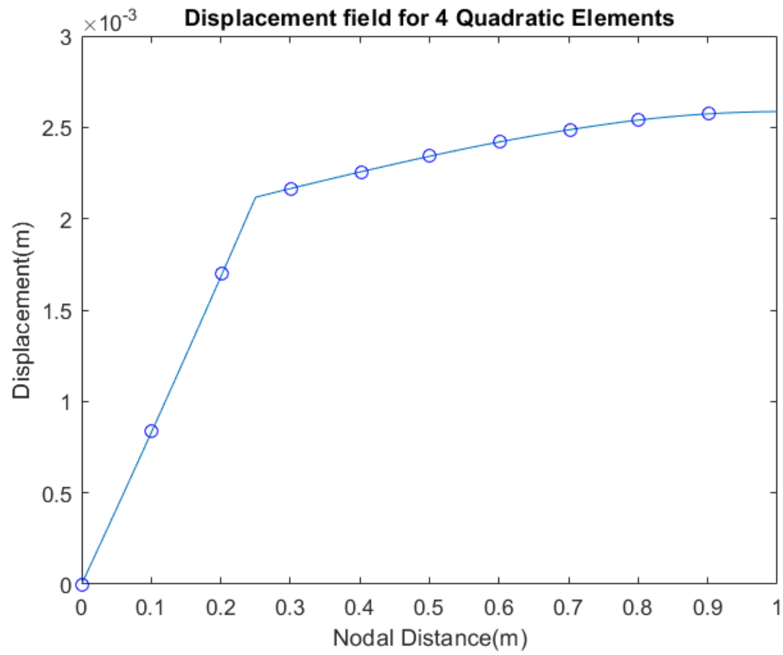


Figure 9

Strain field for 4 Quadratic Elements

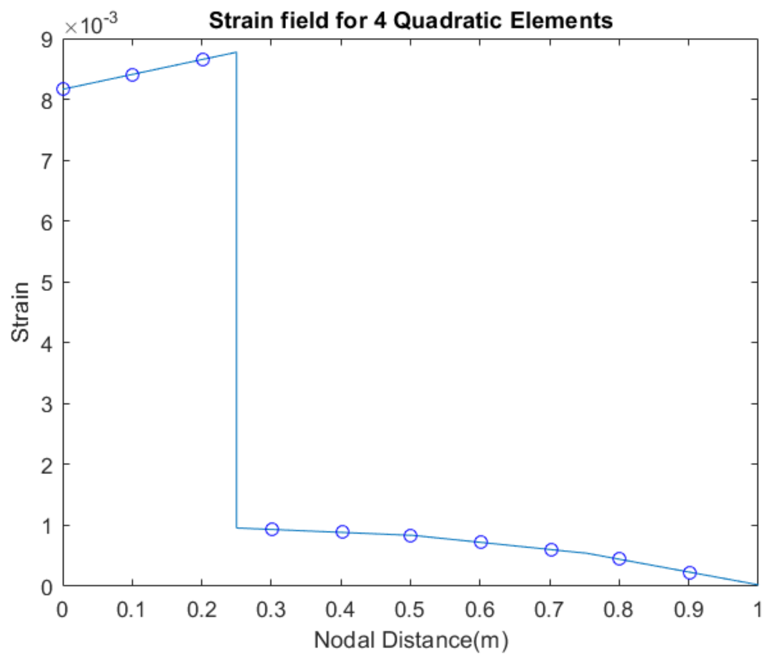


Figure 10

Displacement field for 8 Quadratic Elements

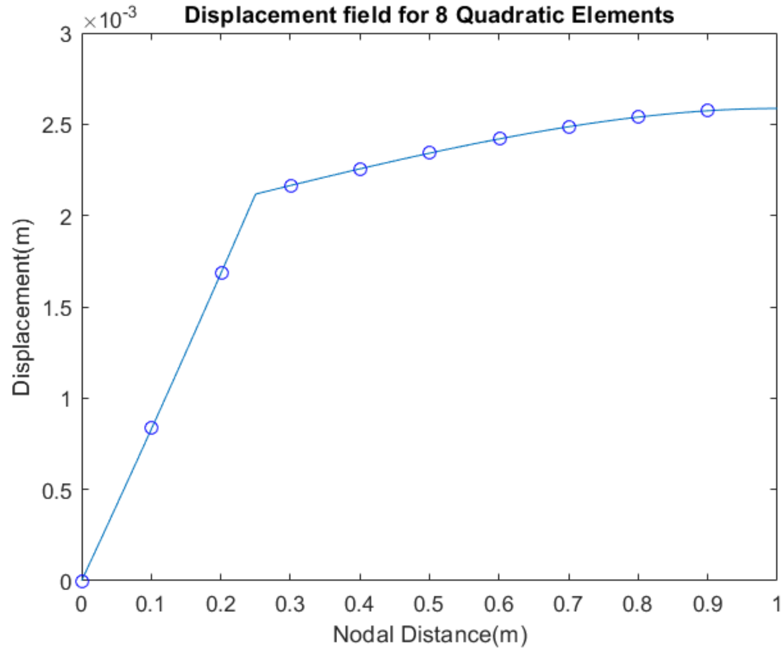


Figure 11

Strain field for 8 Quadratic Elements

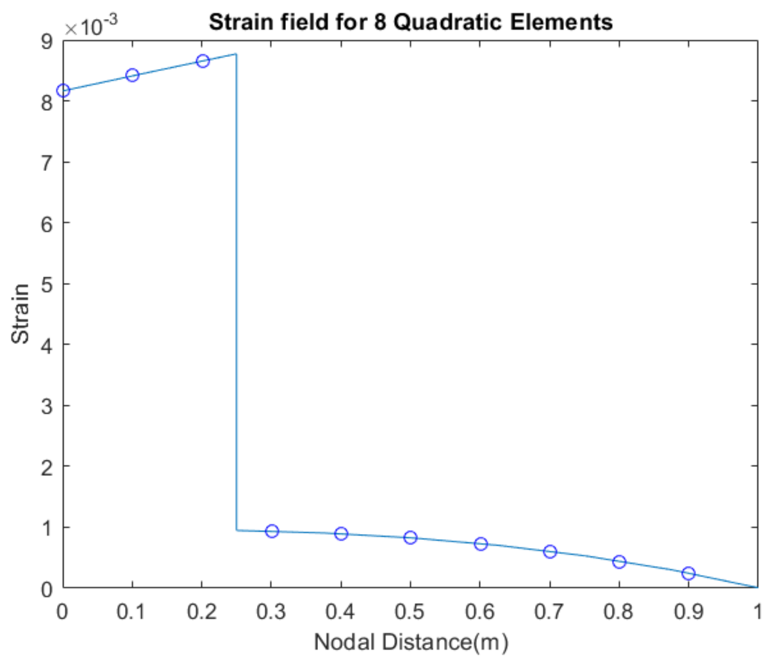


Figure 12

Displacement field for 1 Cubic Element

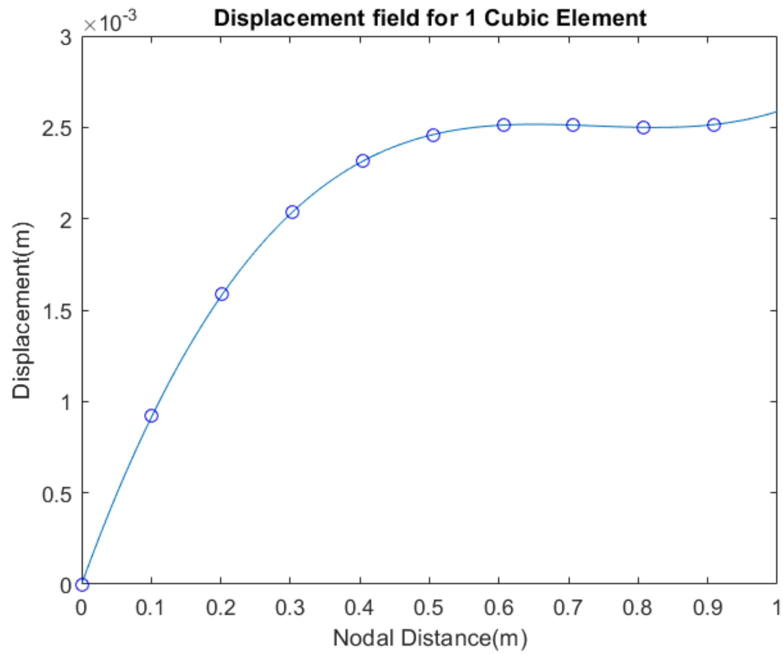


Figure 13

Strain field for 1 Cubic Element

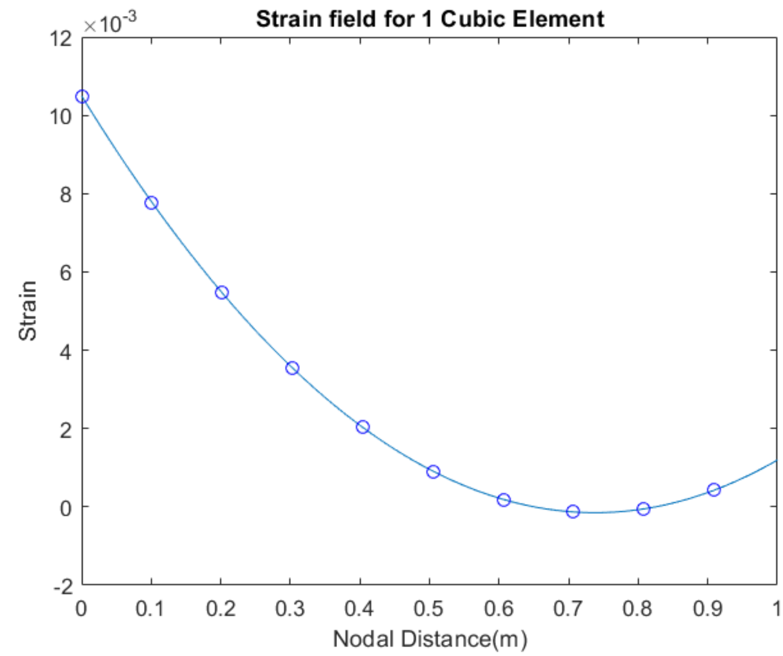


Figure 14

Displacement field for 4 Cubic Elements

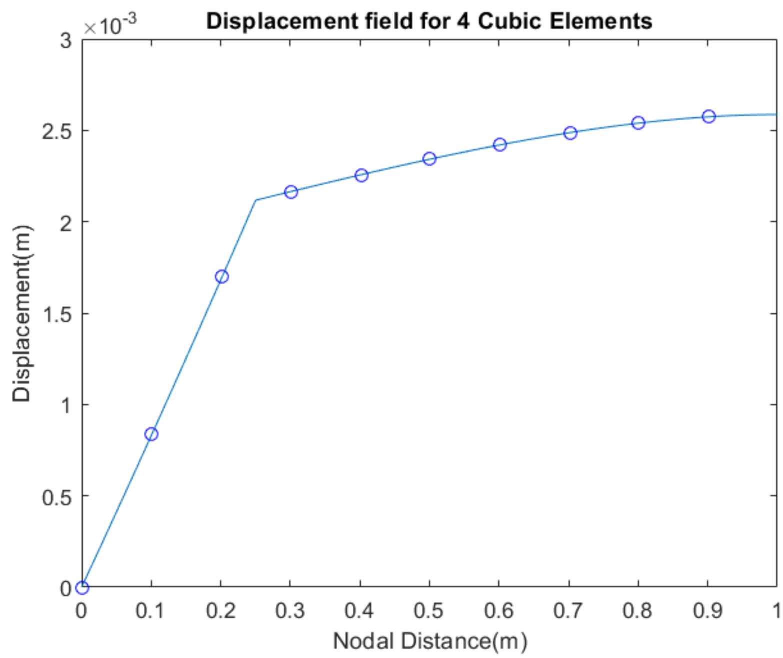


Figure 15

Strain field for 4 Cubic Elements

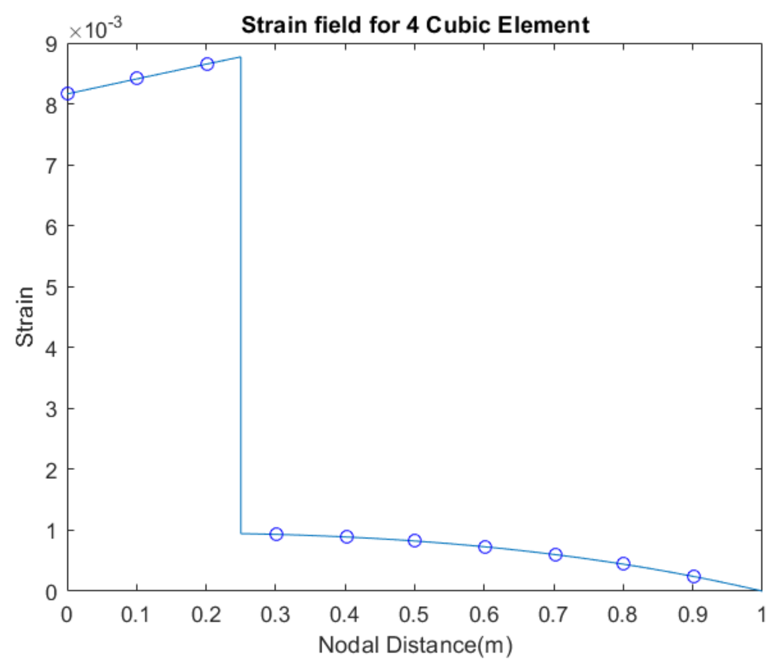


Figure 16

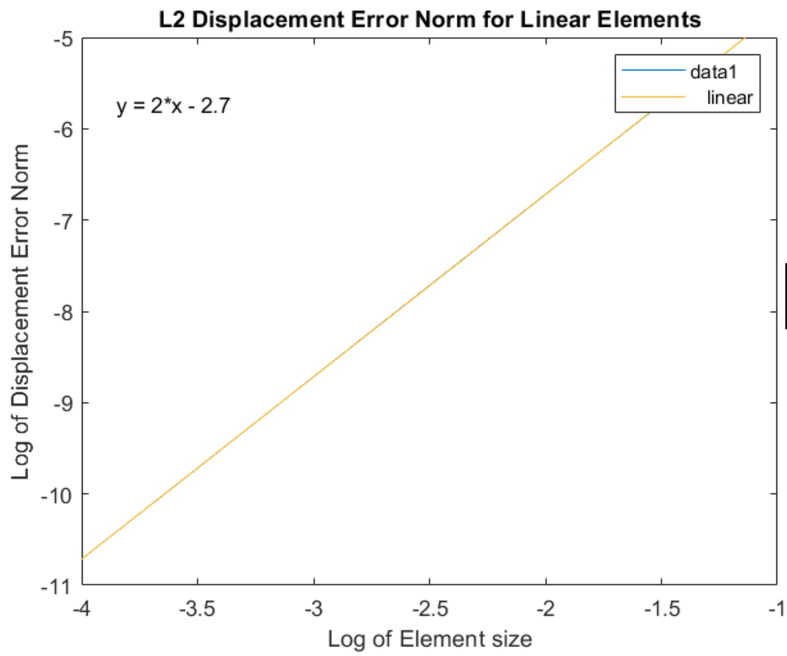
ASSEMBLY TIME (in seconds) COMPARISON (for 1000 elements)

TYPE OF ELEMENT	FULL STORAGE	SPARSE STORAGE
Linear	0.2091	1.1933
Quadratic	0.0860	0.1249
Cubic	0.0837	0.0850

SOLVING TIME (in seconds) COMPARISON (for 1000 elements)

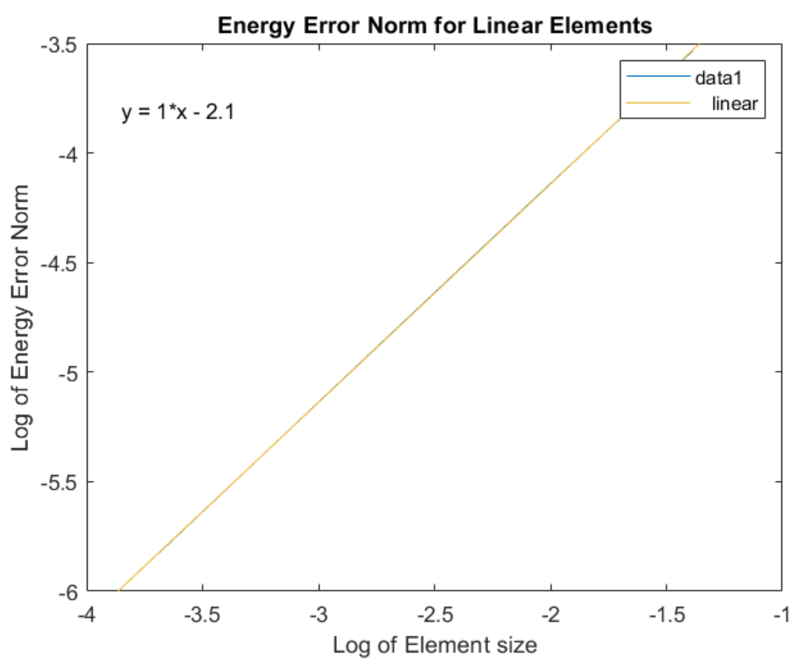
TYPE OF ELEMENT	FULL STORAGE	SPARSE STORAGE
Linear	0.1778	0.2400
Quadratic	0.3406	0.0016
Cubic	0.9477	5.9641*10^-4

ERROR NORM PLOTS FOR LINEAR ELEMENTS



ORDER OF CONVERGENCE - 2

Figure 17



ORDER OF CONVERGENCE - 1

Figure 18

ERROR NORM PLOTS FOR QUADRATIC ELEMENTS

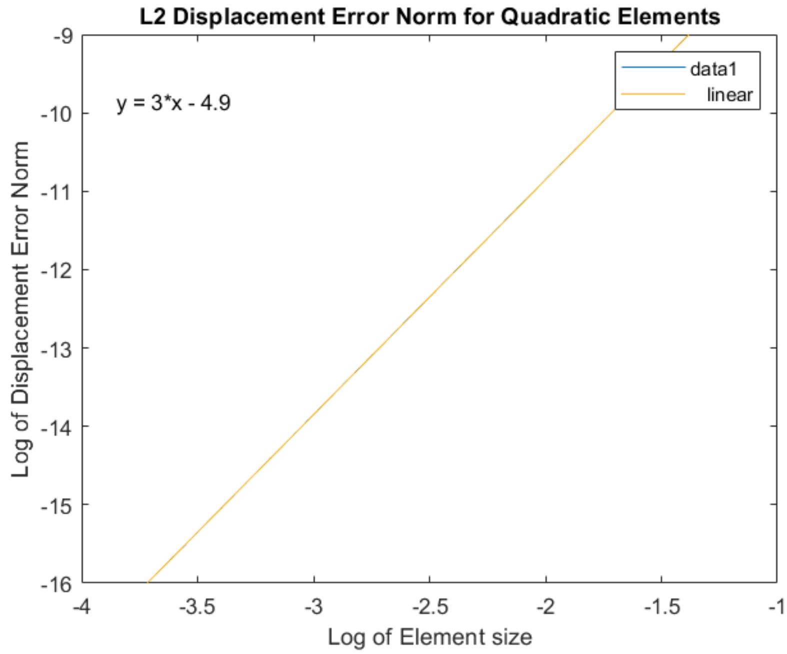


Figure 19

ORDER OF CONVERGENCE - 3

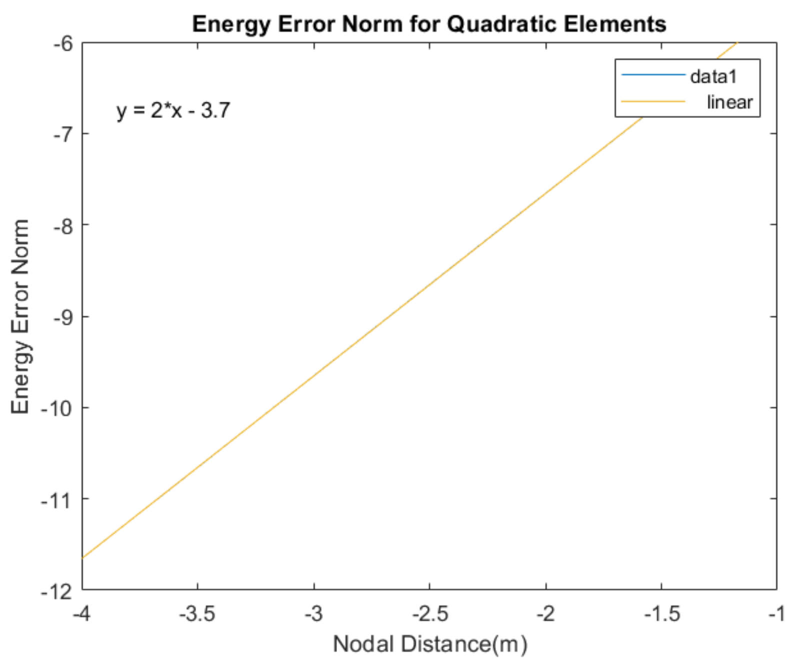


Figure 20

ORDER OF CONVERGENCE - 2

MATLAB CODE

```
clc
clear
close all
% Given parameters, in SI units
A = (pi/4)*0.01^2; %Area of the rod, diameter of the rod = 0.01m
rho = 8200; %density of rod material, kg/m^3
w = 2000*2*pi/60; %angular velocity of the rod, rad/s
M = 10; %mass of the collar attached to the rod, kg
L = 1; %Length of the rod, m

% Code to approximate the stress and displacement in the rod
etype = 'linear';
r = 0;
for ne = 20
    r = r+1;
    size(r) = 1/ne;

if strcmp(etype, 'linear')
    nn = ne + 1;
    mesh.conn = [1:ne; 2:nn];
    qpts = [1/sqrt(3), -1/sqrt(3); 1, 1];
    shape = @shape2;

elseif strcmp(etype, 'quadratic')
    nn = 2*ne + 1;
    mesh.conn = [1:2:nn-2; 2:2:nn-1; 3:2:nn];
    %Integration points
    z1 = sqrt(3/7-2/7*(sqrt(6/5)));
    z2 = sqrt(3/7+2/7*(sqrt(6/5)));
    %weights
    w1 = (18+sqrt(30))/36;
    w2 = (18-sqrt(30))/36;
    qpts=[-z2,-z1,z1,z2; w2,w1,w1,w2];
    shape = @shape3;

elseif strcmp(etype, 'cubic')
    nn = 3*ne + 1;
    mesh.conn = [1:3:nn-3; 2:3:nn-2; 3:3:nn-1; 4:3:nn];
    %Integration points
    z1 = sqrt(3/7-2/7*(sqrt(6/5)));
    z2 = sqrt(3/7+2/7*(sqrt(6/5)));
```

```
    %weights
    w1 = (18+sqrt(30))/36;
    w2 = (18-sqrt(30))/36;
    qpts=[-z2,-z1,z1,z2; w2,w1,w1,w2];
    shape = @shape4;
end

%Radius matrix with number of nodes
rad = zeros(1, nn);
%Temperature matrix, T = 150 degree C at r = 0 and T = 1200 degree C at L=1
T = zeros(1, nn);
%Modulus of Elasticity matrix
YM = zeros(1, nn);

%Defining value of radius for every element
for i = 2:nn
    rad(1,1) = 0;
    rad(1,i) = rad(1, i-1) + L/(nn-1);
end

%Temperature and Modulus of elasticity at every node
for i = 1:nn
    T(1,i) = 1050*rad(1,i) + 150; %varying linearly along the rod
    YM(1,i) = (-0.056*T(1,i) + 201.5) * 10^9; %given Young's Modulus varying with
temperature, MPa
end

x = linspace(0,1,nn);
tic()
K = zeros(nn); %stiffness matrix
F= zeros(nn,1); %force matrix

for c = mesh.conn
    xe = x(:,c);
    Ke = zeros(length(c));
    %Fe = zeros(length(c));
    for q = qpts
        [N, dNdp] = shape(q(1));
        J = xe*dNdp;
        YMv = YM(c)*N;
        dNdx = dNdp/J;
        Ke = Ke + dNdx*YMv*A*dNdx'*J*q(2);
        b = rho*A*(xe*N)*w^2; %body force per unit length on the rod
```

```
        Fe = N*b*J*q(2); %Elemental force
        F(c) = F(c) + Fe;
    end
    K(c,c) = K(c,c) + Ke;
end

assemblytime=toc();
%for the cubic element at 0.25
Fn = xe(length(xe));
p = -0.5;
[N,dNdp]=shape4(p);
if ((xe(1)<0.25)&&(Fn>0.25))
    F(c)=F(c)+N*M*w^2*0.25;
end

Rm = find(x==0.25); %Rm = radius at which collar is attached
F(Rm) = F(Rm) + M*0.25*w^2;

%At the fixed end of the rod
fixed = 1;
K(:,fixed) = 0;
K(fixed,:) = 0;
K(fixed, fixed) = eye(length(fixed));
F(1,1) = 0;

%Displacement in the rod
tic()
d = K\F;

solvingtime=toc();
%for the coordinates in parent coordinate system
xp = [];
%for the displacement in parent coordinate system
dispp = [];
%for the strain in parent coordinate system
strainp = [];
for c = mesh.conn
    de = d(c)';
    xe = x(c);
    for ep = linspace(-1,1,100)
        [N,dNdp] = shape(ep);
        J = xe*dNdp;
        dNdx = dNdp/J;
```

```
        xp(end+1) = xe*N;
        dispp(end+1) = de*N;
        strainp(end+1) = de*dNdx;
    end
end

%Plot of displacement field
figure(1)
plot(xp,dispp)
title('Displacement field for 4 Cubic Elements')
xlabel('Nodal Distance(m)'); ylabel('Displacement(m)');

%Plot of strain field
figure(2)
plot(xp,strainp)
title('Strain field for 4 Cubic Element')
xlabel('Nodal Distance(m)'); ylabel('Strain');

%Code for displacement error norm and energy error norm
eL2num=0;
eL2den=0;
eennum=0;
eenden=0;

for c=mesh.conn
    xe=x(:,c);
    fe=d(c,1)';
    for q=qpts
        [N,dNdp]=shape(q(1));
        J=xe*dNdp;
        fh=fe*N;
        eL2num = eL2num + ((u(xe*N)-fh))^2*J*q(2);
        eL2den = eL2den + (u(xe*N))^2*J*q(2);
        dNdx=dNdp/J;
        dfh=fe*dNdx;
        eennum = eennum + ((du(xe*N)-dfh))^2*J*q(2);
        eenden = eenden + (du(xe*N))^2*J*q(2);
    end
end

e1(r) = sqrt(eL2num/eL2den);
e2(r) = sqrt(eennum/eenden);
end
```

```
%log-log plot for L2 displacement error norm v/s element size
figure(3)
plot(log(size), log(e1))
title('L2 Displacement Error Norm for Quadratic Elements')
xlabel('Log of Element size'); ylabel('Log of Displacement Error Norm');

%log-log plot for energy error norm v/s element size
figure(4)
plot(log(size), log(e2))
title('Energy Error Norm for Quadratic Elements')
xlabel('Log of Element size'); ylabel('Log of Energy Error Norm');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

%Displacement and strain field plots of Analystical, MATLAB, and ABAQUS on
%same plot
i=1;
for r=0:0.01:1
exactx(i)=r;
    exactu(i)=u(r);
    exacts(i)=du(r);
    i=i+1;
end

%Plot of displacement field
load('abaqusu')
figure(5)
plot(xp(1:25:end),dispp(1:25:end), '-o',exactx,exactu,abaqusx,abaqusu)
title('Comparision with exact displacement');
xlabel('Nodal Distance(m)'); ylabel('Displacement(m)');
legend('MATLAB', 'Analytical', 'ABAQUS');

%Plot of strain field
load('abastrain')
figure(6)
plot(xp(1:100:end),strainp(1:100:end),'-o',exactx,exacts,abaSX,abaSS)
title('Comparision with exact strain');
xlabel('Nodal Distance(m)'); ylabel('Strain');
legend('MATLAB', 'Analytical', 'ABAQUS');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Linear Shape Function
function [N, dNdp] = shape2(p)
N= [(1-p)/2; (1+p)/2];
dNdp = [-1/2; 1/2];
end
%Quadratic Shape Function
function [N, dNdp] = shape3(p)
N = [p*(p-1)/2; (p+1)*(1-p); p*(p+1)/2];
dNdp = [(2*p-1)/2; -2*p; (2*p+1)/2];
end
%Cubic Shape Function
function [N, dNdp] = shape4(p)
N = [(3*p+1)*(3*p-1)*(p-1)/(-16); (9)*(p+1)*(3*p-1)*(p-1)/16; (9)*(p+1)*(3*p+1)*(p-1)/(-16); (3*p+1)*(3*p-1)*(p+1)/(16)];
dNdp = [(27*p^2-18*p-1)/(-16); (9/16)*(9*p^2-2*p-3); (-9/16)*(9*p^2+2*p-3); (27*p^2+18*p-1)/(16)];
end

%Displacement Function
    function [ur] = u(r)
        A = (pi/4)*0.01^2; rho = 8200; w = 2000*2*pi/60;
        dTdr = 1050; T0 = 150; Y0 = 201.5*10^9; dYdT = -0.056*10^9; rm = 0.25; R = 1; %rm and R in metres
        if r<=0.25
            ur=-0.16064110451297722+(A*dTdr*dYdT*r*(-(dTdr*dYdT*r)+2*(dYdT*T0+Y0))*rho*w^2-2*(-247574.7054440367*dTdr^2*dYdT^2+A*(dYdT*T0+Y0)^2*rho*w^2)*log(dTdr*dYdT*r+dYdT*T0+Y0))/(4*A*dTdr^3*dYdT^3);
        else
            ur=-0.7758350930553184+(A*dTdr*dYdT*r*(-(dTdr*dYdT*r)+2*(dYdT*T0+Y0))*rho*w^2-2*(-28250.1631976065*dTdr^2*dYdT^2+A*(dYdT*T0+Y0)^2*rho*w^2)*log(dTdr*dYdT*r+dYdT*T0+Y0))/(4*A*dTdr^3*dYdT^3);
        end
    end

%Strain Function
    function [dur] = du(r)
        A = (pi/4)*0.01^2; rho = 8200; w = 2000*2*pi/60; M = 10;
```

```
    dTdr = 1050; T0 = 150; Y0 = 201.5*10^9; dYdT = -0.056*10^9; rm = 0.25; R = 1;
    if r<=0.25
      dur=(-(A*r^2*rho*w^2)/2+((2*M*rm+A*R^2*rho)*w^2)/2)/(A*(dYdT*(dTdr*r+T0)+Y0));
    else
      dur=(-(A*r^2*rho*w^2)/2+(A*R^2*rho*w^2)/2)/(A*(dYdT*(dTdr*r+T0)+Y0));
    end
  end
```

MATLAB CODE FOR CALCULATING ASSEMBLY AND SOLVING TIME

```
clc
clear
close all
% Given parameters, in SI units
A = (pi/4)*0.01^2; %Area of the rod, diameter of the rod = 0.01m
rho = 8200; %density of rod material, kg/m^3
w = 2000*2*pi/60; %angular velocity of the rod, rad/s
M = 10; %mass of the collar attached to the rod, kg
L = 1; %Length of the rod, m
elementtype=["linear" "quadratic" "cubic"]
% Code to approximate the stress and displacement in the rod
r = 0;
for k1=1:2
for eno=1:3

for ne = 1000
etype = elementtype(eno);

  r = r+1;
  size(r) = 1/ne;

if strcmp(etype, 'linear')
  nn = ne + 1;
  mesh.conn = [1:ne; 2:nn];
  qpts = [1/sqrt(3), -1/sqrt(3); 1, 1];
  shape = @shape2;

elseif strcmp(etype, 'quadratic')
```

```
    nn = 2*ne + 1;
    mesh.conn = [1:2:nn-2; 2:2:nn-1; 3:2:nn];
    %Integration points
    z1 = sqrt(3/7-2/7*(sqrt(6/5)));
    z2 = sqrt(3/7+2/7*(sqrt(6/5)));
    %weights
    w1 = (18+sqrt(30))/36;
    w2 = (18-sqrt(30))/36;
    qpts=[-z2,-z1,z1,z2; w2,w1,w1,w2];
    shape = @shape3;

elseif strcmp(etype, 'cubic')
    nn = 3*ne + 1;
    mesh.conn = [1:3:nn-3; 2:3:nn-2; 3:3:nn-1; 4:3:nn];
    %Integration points
    z1 = sqrt(3/7-2/7*(sqrt(6/5)));
    z2 = sqrt(3/7+2/7*(sqrt(6/5)));
    %weights
    w1 = (18+sqrt(30))/36;
    w2 = (18-sqrt(30))/36;
    qpts=[-z2,-z1,z1,z2; w2,w1,w1,w2];
    shape = @shape4;
end

%Radius matrix with number of nodes
rad = zeros(1, nn);
%Temperature matrix, T = 150 degree C at r = 0 and T = 1200 degree C at L=1
T = zeros(1, nn);
%Modulus of Elasticity matrix
YM = zeros(1, nn);

%Defining value of radius for every element
for i = 2:nn
    rad(1,1) = 0;
    rad(1,i) = rad(1, i-1) + L/(nn-1);
end

%Temperature and Modulus of elasticity at every node
for i = 1:nn
    T(1,i) = 1050*rad(1,i) + 150; %varying linearly along the rod
    YM(1,i) = (-0.056*T(1,i) + 201.5) * 10^9; %given Young's Modulus varying with
temperature, MPa
end
```

```
x = linspace(0,1,nn);
tic()
if k1==1
K = zeros(nn); %stiffness matrix
else
K = spalloc(nn,nn,5*nn); %stiffness matrix  bolo tarararararaaa
end
F= zeros(nn,1); %force matrix

for c = mesh.conn
    xe = x(:,c);
    Ke = zeros(length(c));
    %Fe = zeros(length(c));
    for q = qpts
        [N, dNdp] = shape(q(1));
        J = xe*dNdp;
        YMv = YM(c)*N;
        dNdx = dNdp/J;
        Ke = Ke + dNdx*YMv*A*dNdx'*J*q(2);
        b = rho*A*(xe*N)*w^2; %body force per unit length on the rod
        Fe = N*b*J*q(2); %Elemental force
        F(c) = F(c) + Fe;
    end
    K(c,c) = K(c,c) + Ke;
end

asmblytime=toc();
%for the cubic element at 0.25
Fn = xe(length(xe));
p = -0.5;
[N,dNdp]=shape4(p);
if ((xe(1)<0.25)&&(Fn>0.25))
    F(c)=F(c)+N*M*w^2*0.25;
end

Rm = find(x==0.25); %Rm = radius at which collar is attached
F(Rm) = F(Rm) + M*0.25*w^2;

%At the fixed end of the rod
fixed = 1;
K(:,fixed) = 0;
K(fixed,:) = 0;
```

```
K(fixed, fixed) = eye(length(fixed));
F(1,1) = 0;

%Displacement in the rod
tic()
d = K\F;

slvngtime=toc();
%for the coordinates in parent coordinate system
xp = [];
%for the displacement in parent coordinate system
dispp = [];
%for the strain in parent coordinate system
strainp = [];
for c = mesh.conn
    de = d(c)';
    xe = x(c);
    for ep = linspace(-1,1,100)
        [N,dNdp] = shape(ep);
        J = xe*dNdp;
        dNdx = dNdp/J;
        xp(end+1) = xe*N;
        dispp(end+1) = de*N;
        strainp(end+1) = de*dNdx;
    end
end

AssemblyTime(eno,k1)=asmblytime;
SolvingTime(eno,k1)=slvngtime;
end
end
end



%Linear Shape Function
function [N, dNdp] = shape2(p)
N= [(1-p)/2; (1+p)/2];
dNdp = [-1/2; 1/2];
end
%Quadratic Shape Function
function [N, dNdp] = shape3(p)
N = [p*(p-1)/2; (p+1)*(1-p); p*(p+1)/2];
```

```
dNdp = [(2*p-1)/2; -2*p; (2*p+1)/2];
end
%Cubic Shape Function
function [N, dNdp] = shape4(p)
N = [(3*p+1)*(3*p-1)*(p-1)/(-16); (9)*(p+1)*(3*p-1)*(p-1)/16; (9)*(p+1)*(3*p+1)*(p-
1)/(-16); (3*p+1)*(3*p-1)*(p+1)/(16)];
dNdp = [(27*p^2-18*p-1)/(-16); (9/16)*(9*p^2-2*p-3); (-9/16)*(9*p^2+2*p-3);
(27*p^2+18*p-1)/(16)];
end

%Displacement Function
  function [ur] = u(r)
    A = (pi/4)*0.01^2; rho = 8200; w = 2000*2*pi/60;
    dTdr = 1050; T0 = 150; Y0 = 201.5*10^9; dYdT = -0.056*10^9; rm = 0.25; R
= 1; %rm and R in metres
    if r<=0.25
      ur=-0.16064110451297722+(A*dTdr*dYdT*r*(-
(dTdr*dYdT*r)+2*(dYdT*T0+Y0))*rho*w^2-2*(-
247574.7054440367*dTdr^2*dYdT^2+A*(dYdT*T0+Y0)^2*rho*w^2)*log(dTdr*dY
dT*r+dYdT*T0+Y0))/(4*A*dTdr^3*dYdT^3);
    else
      ur=-0.7758350930553184+(A*dTdr*dYdT*r*(-(dTdr*dYdT*r)+
2*(dYdT*T0+Y0))*rho*w^2-2*(-
28250.1631976065*dTdr^2*dYdT^2+A*(dYdT*T0+Y0)^2*rho*w^2)*log(dTdr*dYd
T*r+dYdT*T0+Y0))/(4*A*dTdr^3*dYdT^3);
    end
  end

%Strain Function
  function [dur] = du(r)
    A = (pi/4)*0.01^2; rho = 8200; w = 2000*2*pi/60; M = 10;
    dTdr = 1050; T0 = 150; Y0 = 201.5*10^9; dYdT = -0.056*10^9; rm = 0.25; R
= 1;
    if r<=0.25
      dur=(-
(A*r^2*rho*w^2)/2+((2*M*rm+A*R^2*rho)*w^2)/2)/(A*(dYdT*(dTdr*r+T0)+Y0));
    else
      dur=(-(A*r^2*rho*w^2)/2+(A*R^2*rho*w^2)/2)/(A*(dYdT*(dTdr*r+T0)+Y0));
    end
  end
```

Procedure for an approximate solution using ABAQUS

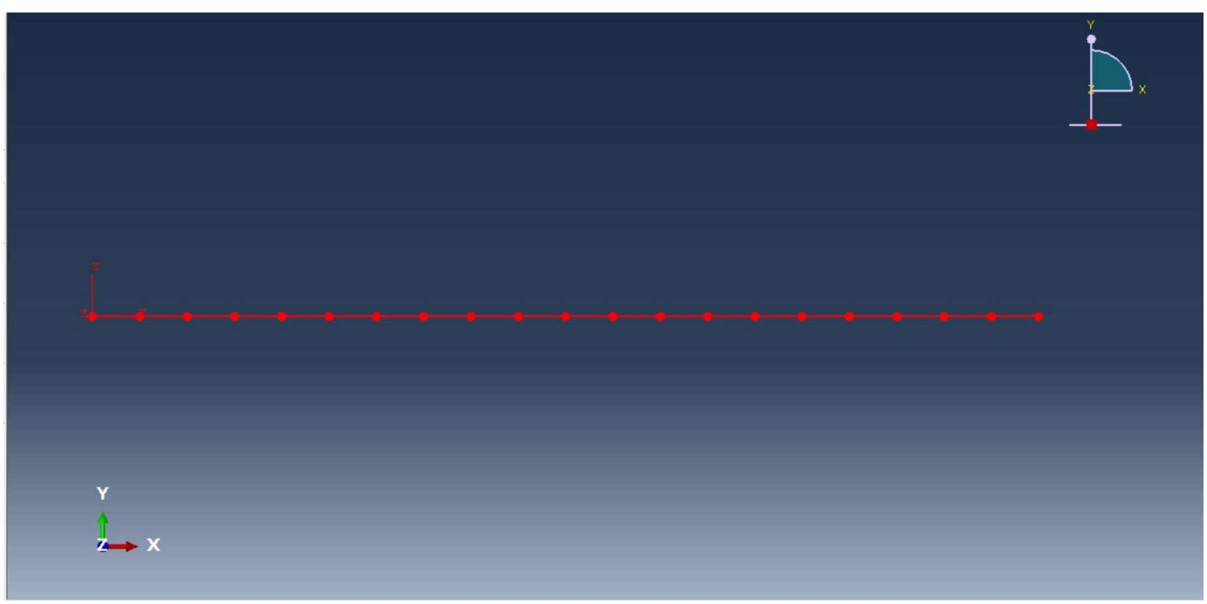
1. A 1D truss component of 1m length was created with Wire as a base feature.

2. The component was given the density of 8200 kg/m^3 (nickel-based, super-alloy) and the temperature dependent Modulus of elasticity.
   E = 193.1 GPa at 150℃ and E = 134.3 GPa at 1200℃

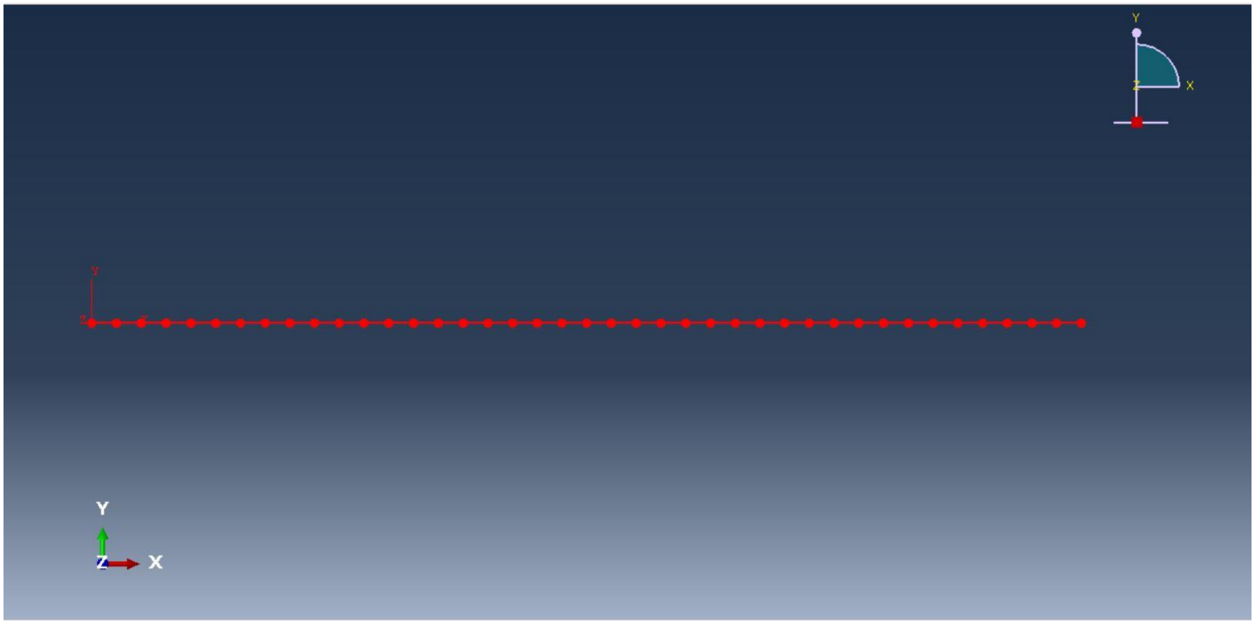
3. Truss section with the cross-section area = 7.85398 x 10^-5 m^2 (diameter = 0.01m).

4. Mesh size was given as 0.05m.

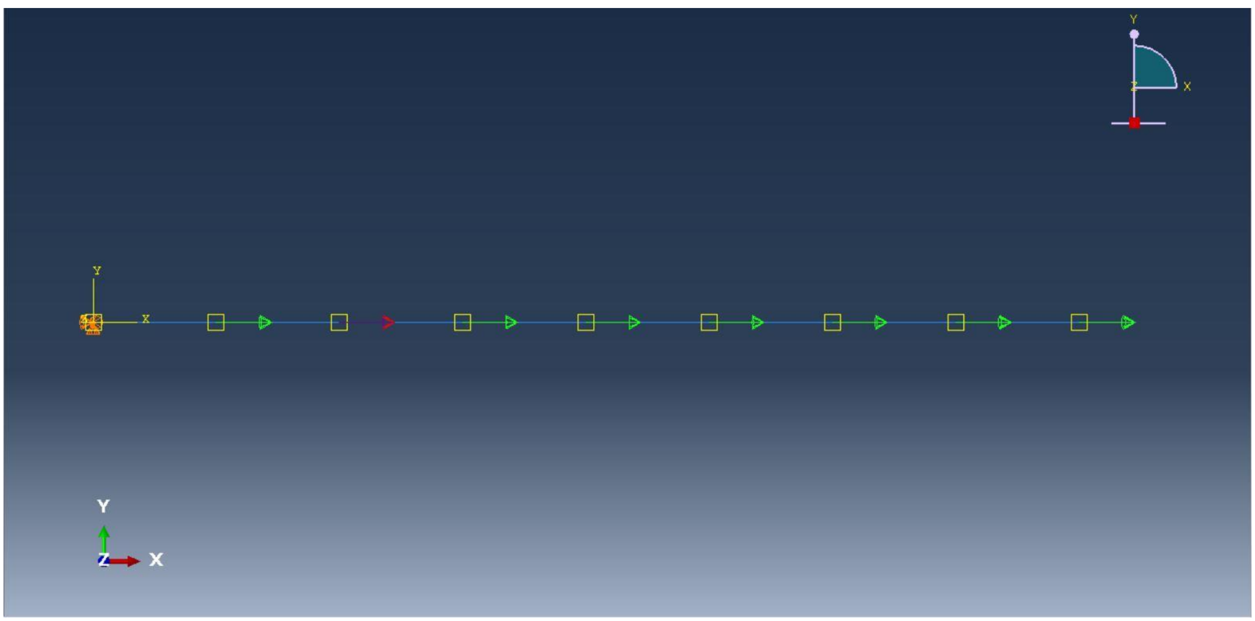
Meshing using Linear Elements



Meshing using Quadratic Elements

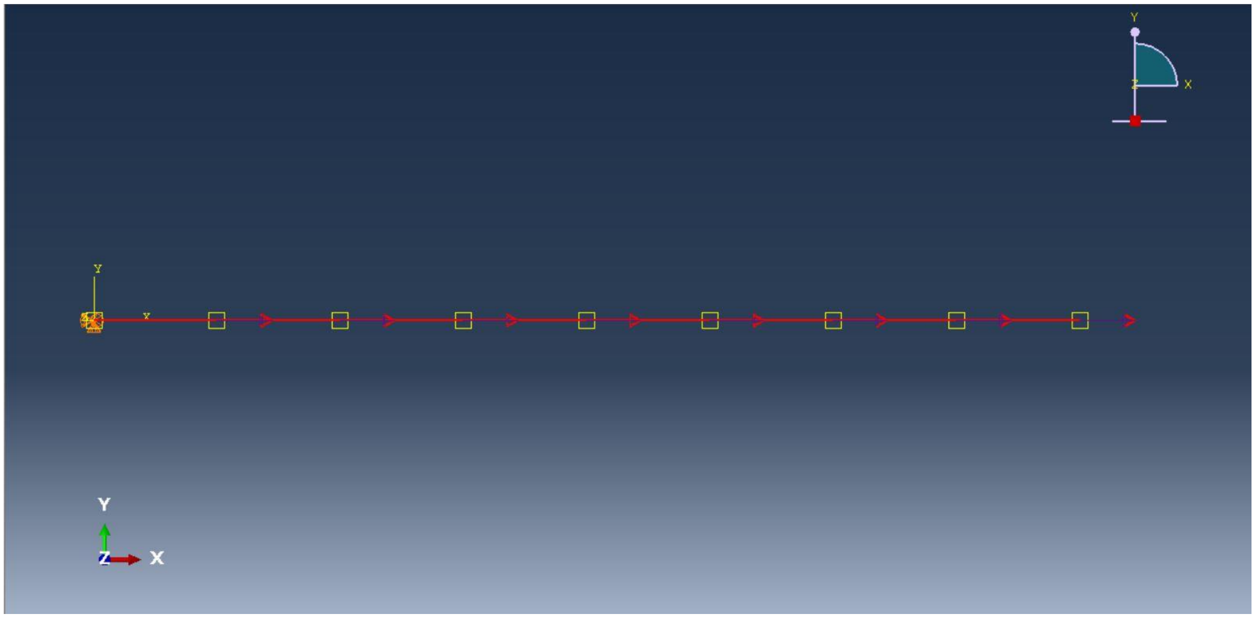


5. A set was created at 0.25m for applying the concentrated load due to the 10 kg collar attached to the rod.



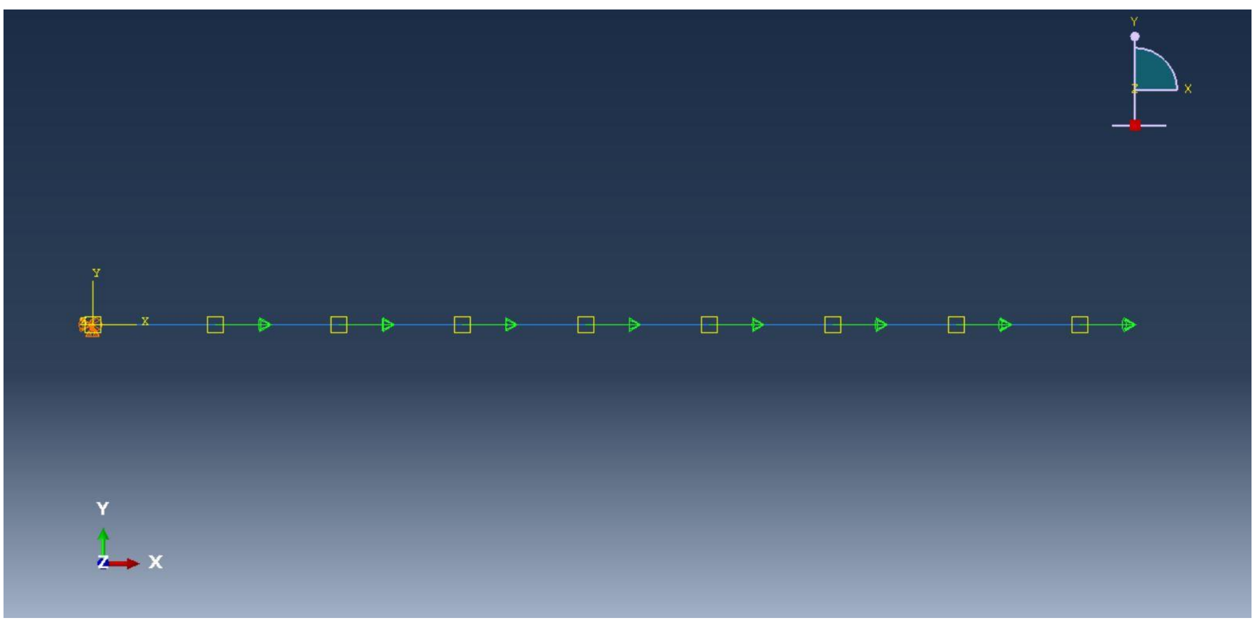
Red Arrow Indicating Concentrated Force

6. The rod was fixed from one end and given the angular velocity of 209.44 rad/s with z-axis as axis of rotation.



Red Arrows indicating Rotational Body Force

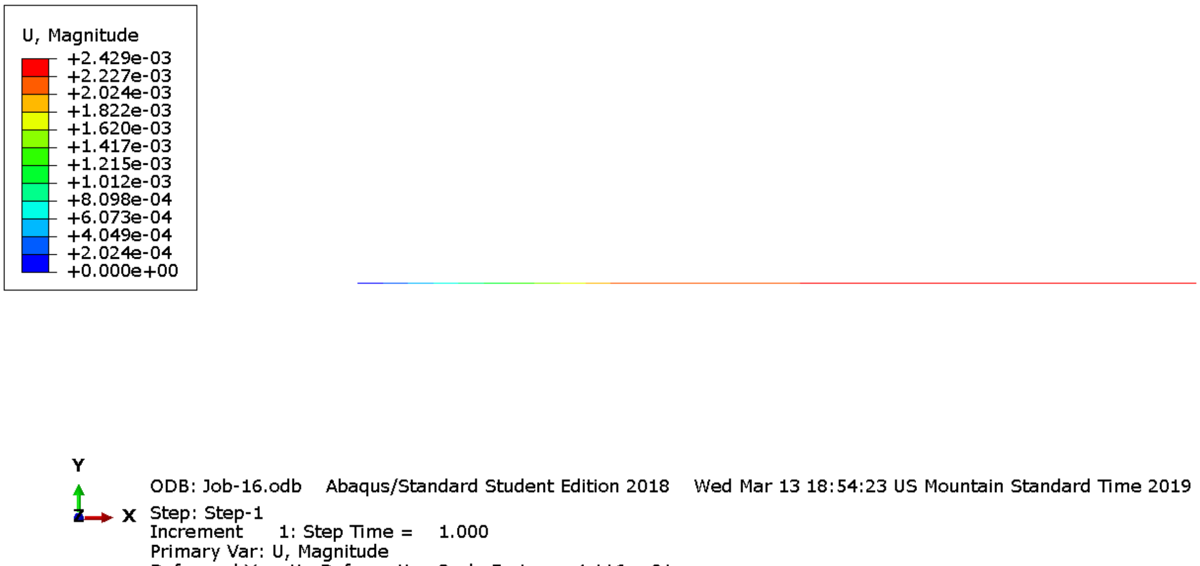
7. A predefined field was created to vary the temperature and modulus of elasticity with the radius of the rod.



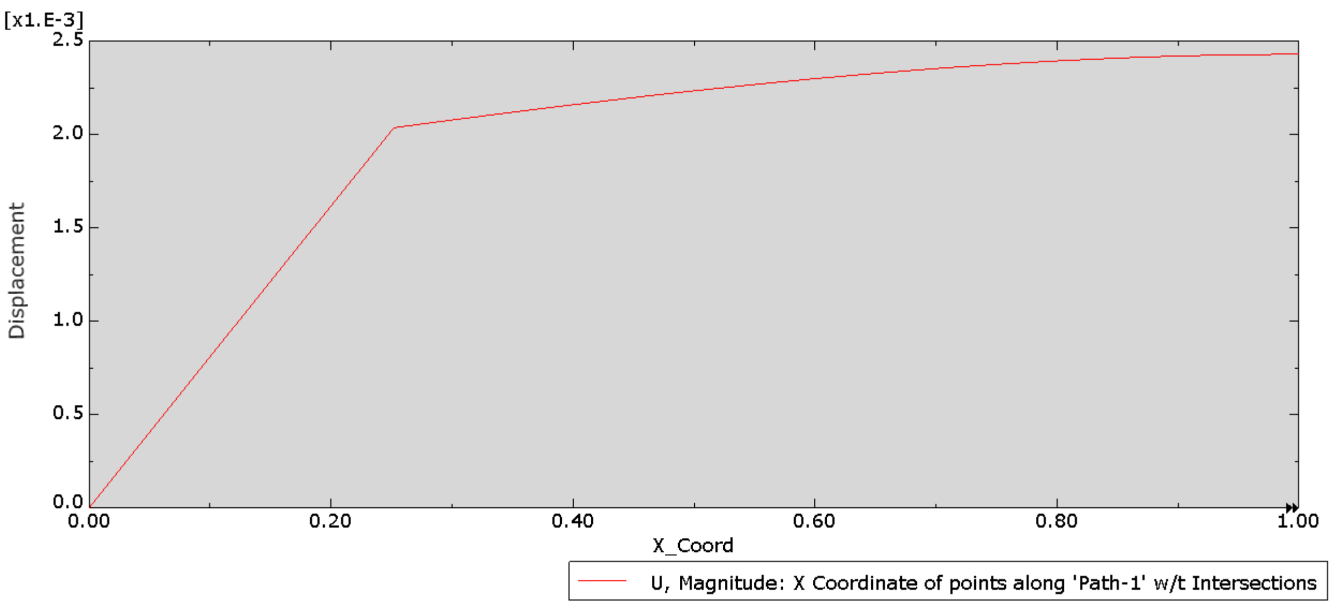
Predefined Field

8. Pinned support was used at one end of the rod.

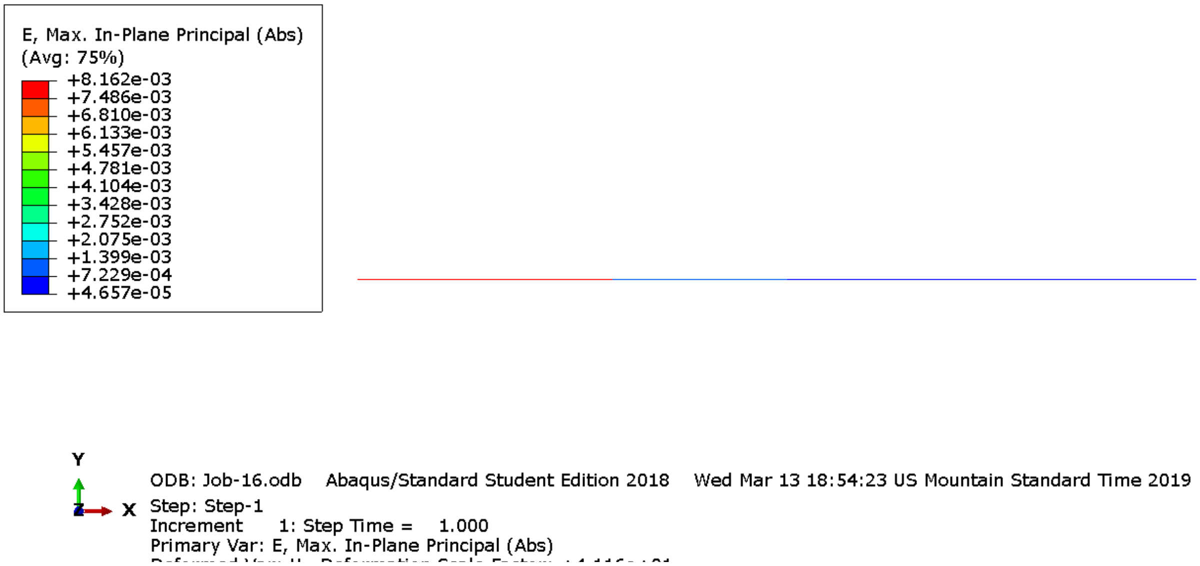
Results



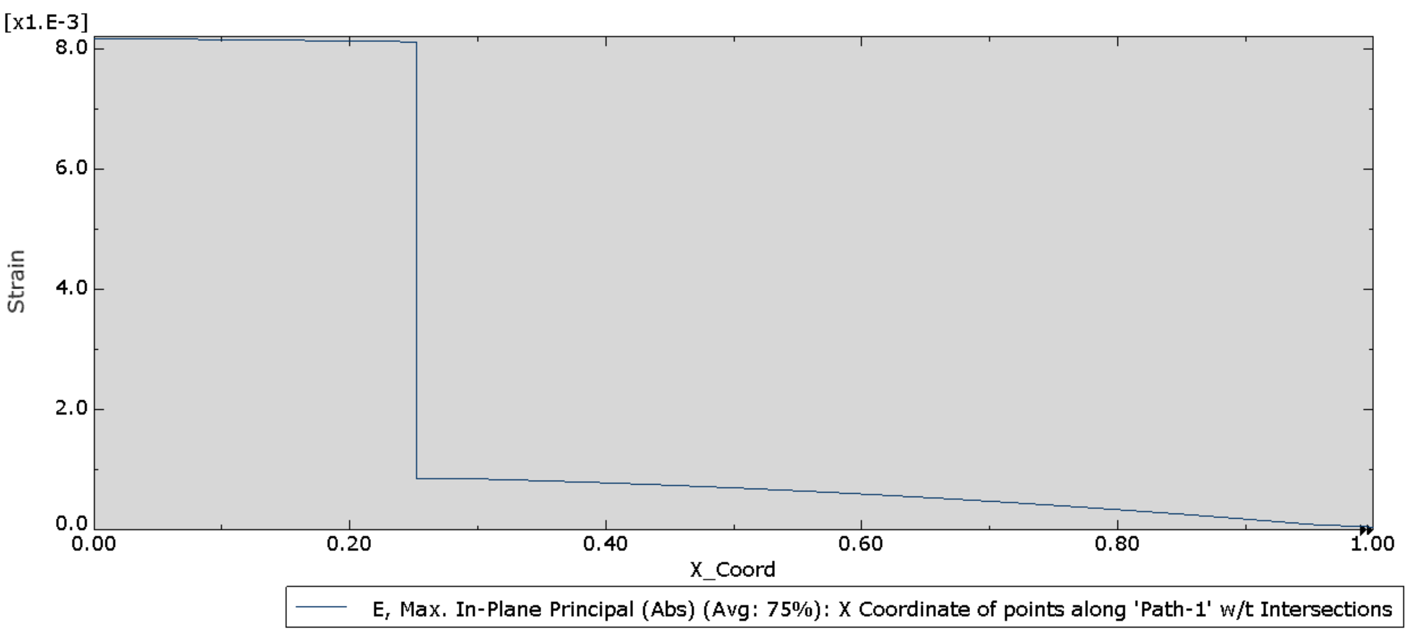
Displacement using Linear Elements



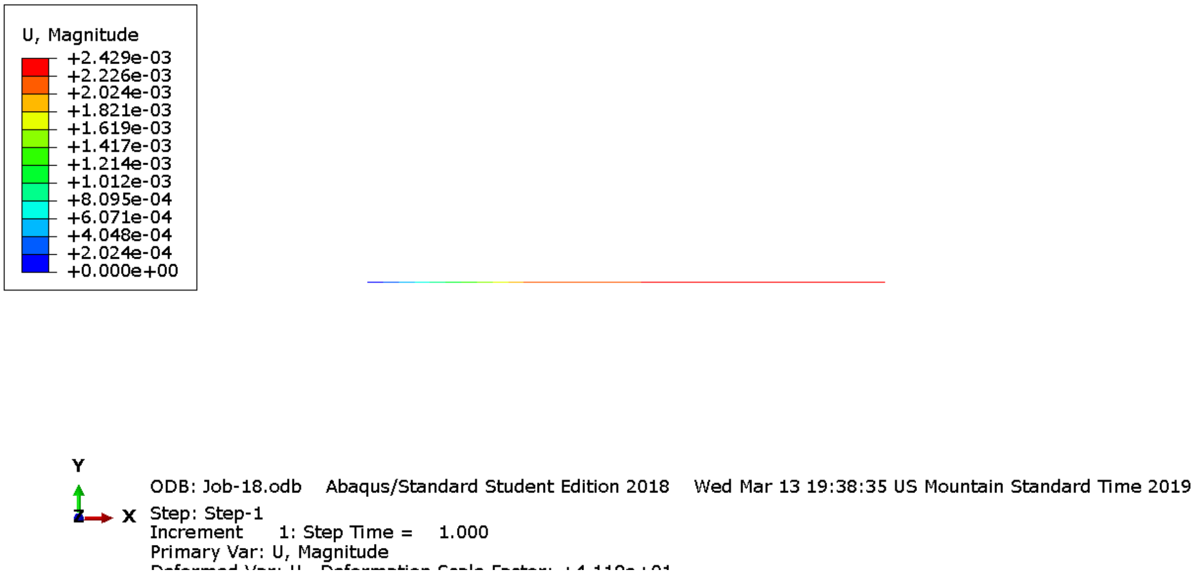
Displacement Field for Linear Elements



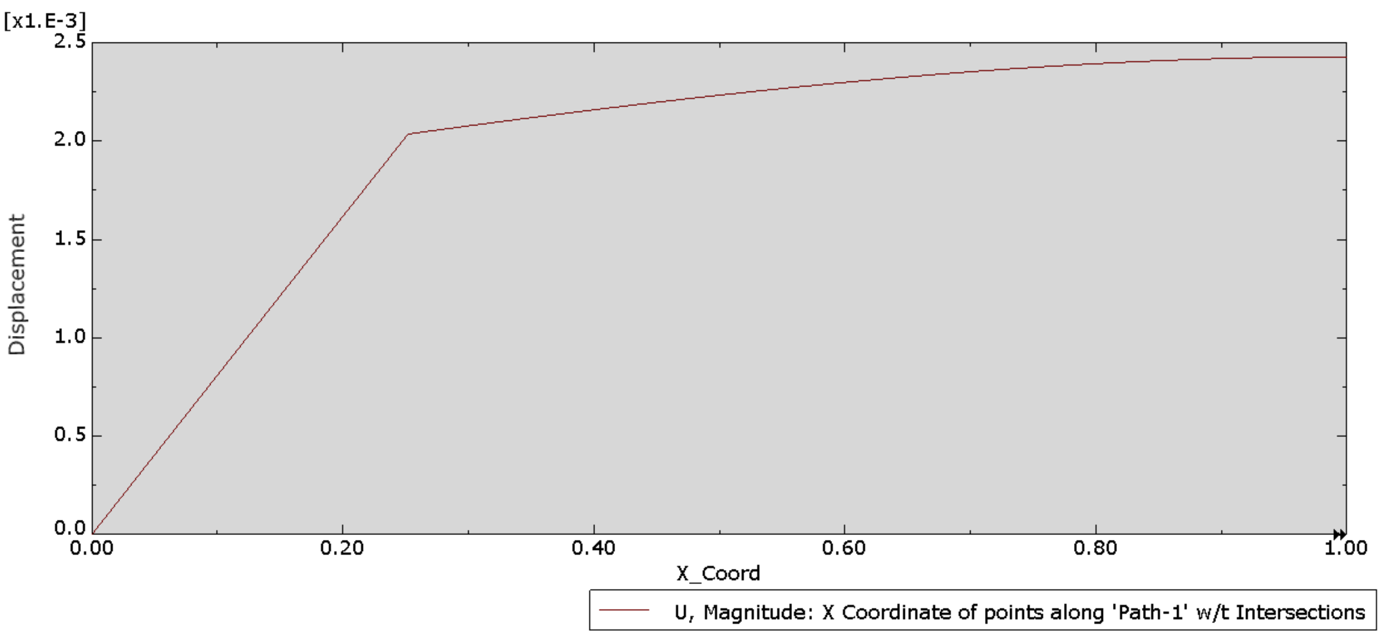
Strain using Linear Elements



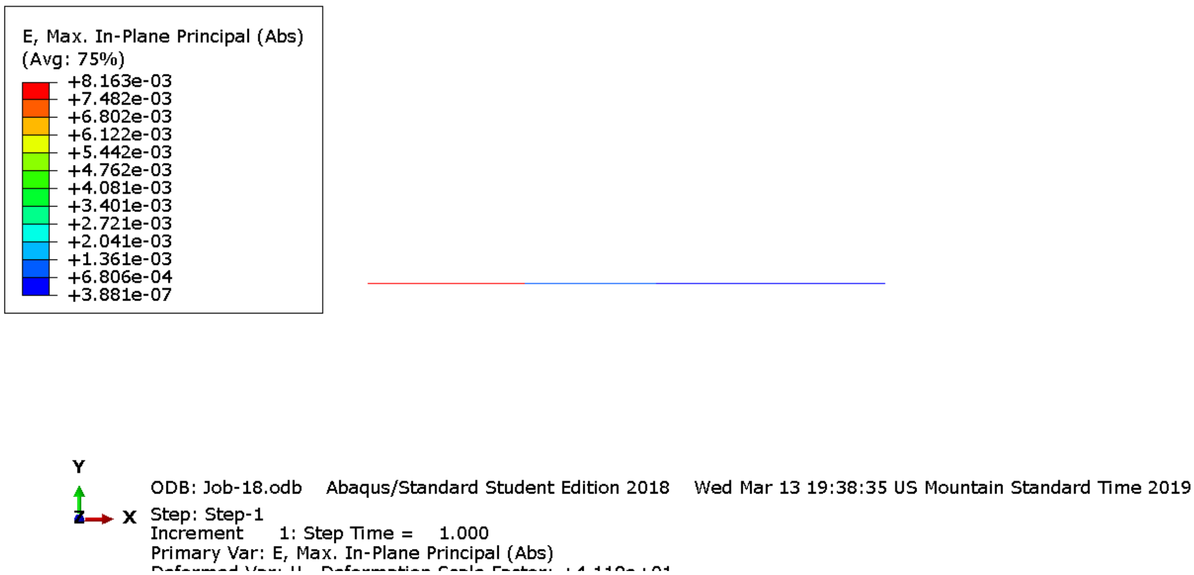
Strain Field for Linear Elements



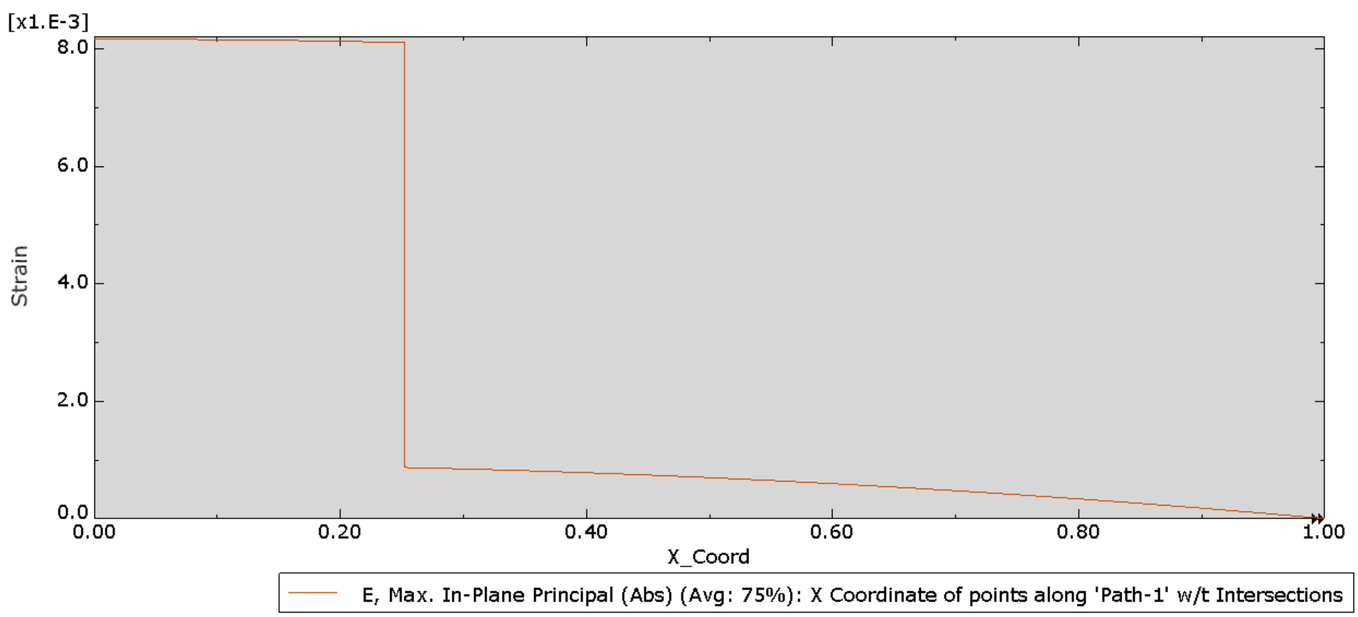
Displacement using Quadratic Elements



Displacement Field for Quadratic Elements



Strain using Quadratic Elements



Strain Field using Quadratic Elements