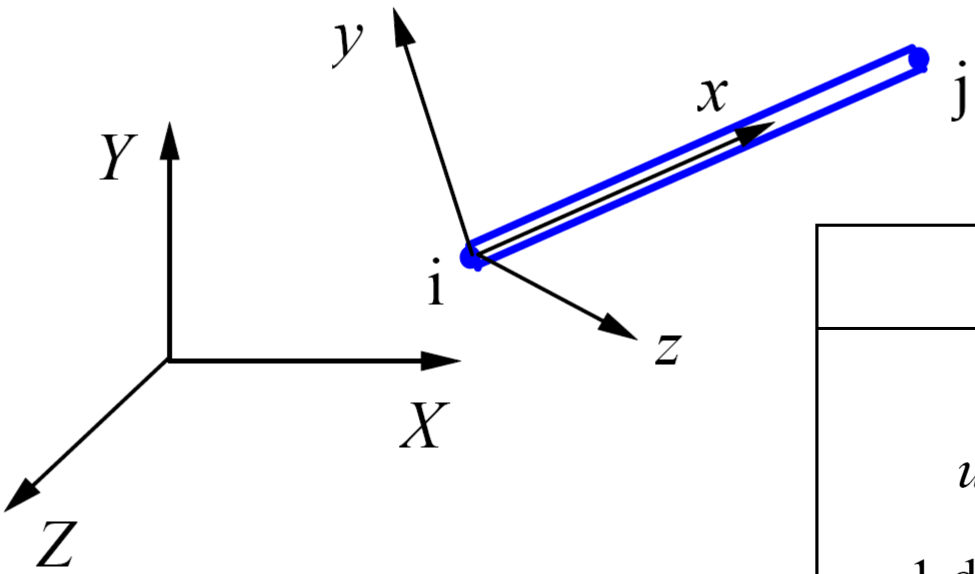
Example of a truss structure

المان میله در فضای سه بعدی



Local	*Global*
x, y, z	X, Y, Z
$u_i^{'}, v_i^{'}, w_i^{'}$	u_i, v_i, w_i
1 dof at node	3 dof s at node

❖ پس از محاسبه ماتریس سختی در دستگاه مختصات محلی، با انتقال به دستگاه مختصات سراسری (X, Y, Z) ماتریس سختی در آن دستگاه محاسبه می شود.

ماتریس سختی میله در فضای سه بعدی

معادلات تعادل در دستگاه محلی $x'y'$

$$\frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_i' \\ u_j' \end{Bmatrix}=\begin{Bmatrix} f_i' \\ f_j' \end{Bmatrix}$$

با افزودن ابعاد ماتریس:

$$\frac{AE}{L}\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} u_i' \\ v_i' \\ w_i' \\ u_j' \\ v_j' \\ w_j' \end{Bmatrix}=\begin{Bmatrix} f_i' \\ \mathbf{0} \\ \mathbf{0} \\ f_j' \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}$$

یا

$$\mathrm{k'u'}=\mathrm{f'}$$

ماتریس سختی میله در فضای سه بعدی

$$k'u' = f' \xrightarrow[\boxed{f'=T\,f}]{u'=T\,u} k'Tu = Tf \xrightarrow{T^T \times} T^T k'Tu = f$$

$$k = T^T k' T$$

$$\begin{cases} u_i' = \cos\theta_X u_i + \cos\theta_Y v_i + \cos\theta_Z w_i \\ u_j' = \cos\theta_X u_j + \cos\theta_Y v_j + \cos\theta_Z w_j \end{cases}$$

$$\cos\theta_X = \frac{X_j - X_i}{L}$$

$$\cos\theta_Y = \frac{Y_j - Y_i}{L}$$

$$\cos\theta_Z = \frac{Z_j - Z_i}{L}$$

$$\begin{Bmatrix} u_i' \\ u_j' \end{Bmatrix} = \begin{bmatrix} \cos\theta_X & \cos\theta_Y & \cos\theta_Z & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta_X & \cos\theta_Y & \cos\theta_Z \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{Bmatrix}$$

$$L = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2 + (Z_j - Z_i)^2}$$

$$\mathrm{k} = \mathrm{T}^{\mathrm{T}}\mathrm{k}'\mathrm{T}$$

$$[\mathbf{k}]_{6\times6} = [\mathbf{T}]^{\mathrm{T}}_{6\times2}[\mathbf{k}']_{2\times2}[\mathbf{T}]_{2\times6} \qquad [\mathbf{F}]_{6\times1} = [\mathbf{T}]^{\mathrm{T}}_{6\times2}\{\mathbf{F}'\}_{2\times1}$$

$$\cos\theta_X = l,\quad \cos\theta_Y = m,\quad \cos\theta_Z = n$$

$$\boldsymbol{k} = \frac{AE}{L}\begin{bmatrix} l^2 & lm & ln & -l^2 & -lm & -ln \\ lm & m^2 & mn & -lm & -m^2 & -mn \\ ln & mn & n^2 & -ln & -mn & -n^2 \\ -l^2 & -lm & -ln & l^2 & lm & ln \\ -lm & -m^2 & -mn & lm & m^2 & mn \\ -ln & -mn & -n^2 & ln & mn & n^2 \end{bmatrix}$$

المان میله در فضای سه بعدی

Input data for bar elements:

- *(X, Y, Z)* for each node
- *E* and *A* for each element

Calculate:

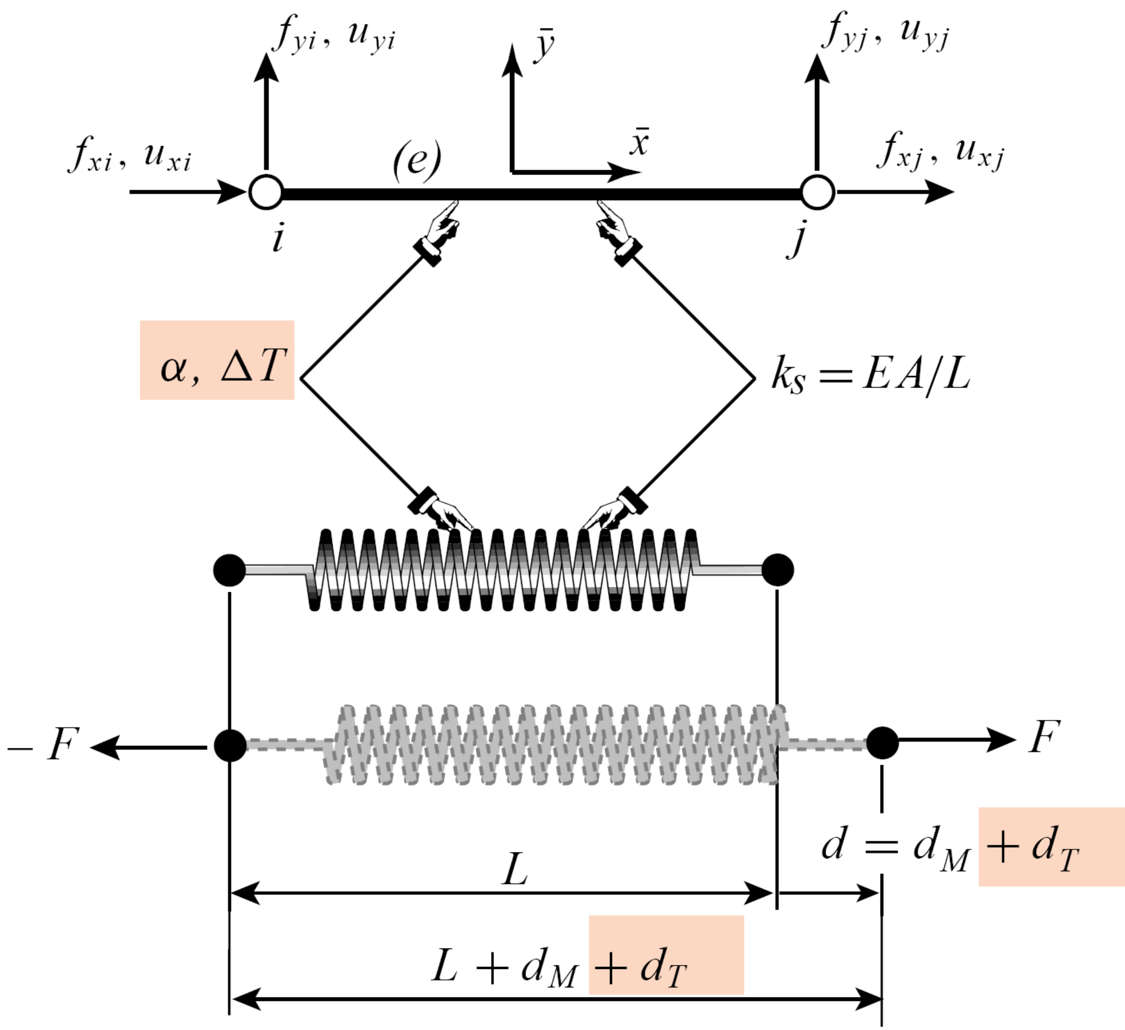
- The directional cosines
- The element stiffness matrix in global coordinates
- The element force vector in global coordinates
- Assemble the stiffness matrices to obtain the global stiffness matrix
- Assemble the load vectors to obtain the global load vector
- Solve the final equation to obtain the displacement at different nodes

تاثیر نیروهای اولیه

❖ نیروهای اولیه در یک سازه می‌توانند به صورت کرنش اولیه و یا تنش اولیه به سازه اعمال شوند و برای تحلیل سازه باید در نظر گرفته شوند.

- Thermomechanical effects
- Moisture effects
- Prestress effects
- Lack of fit
- Residual stresses

نیروهای اولیه ترمومکانیکی در میله



نیروهای اولیه ترمومکانیکی در میله

❖ کرنش محوری ناشی از اثرات مکانیکی و حرارتی است

$$e_T = d_T / L = \alpha\, \Delta T$$

$$e = d/L \qquad d = \bar{u}_{xj} - \bar{u}_{xi}$$

$$e = e_M + e_T = \frac{\sigma}{E} + \alpha\, \Delta T$$

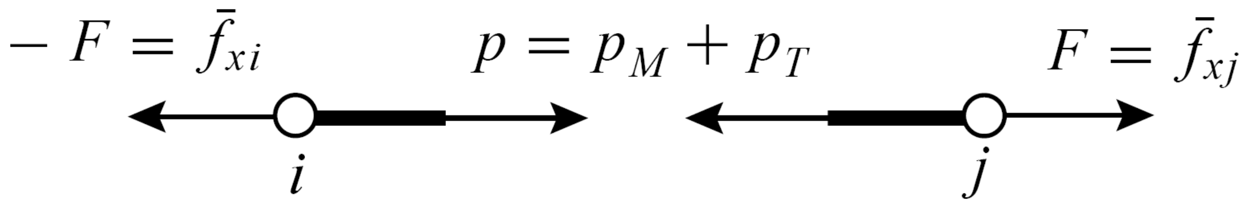
$$\frac{\bar{u}_{xj} - \bar{u}_{xi}}{L} = \frac{\sigma}{E} + \alpha\, \Delta T$$

$$\frac{EA}{L}(\bar{u}_{xj} - \bar{u}_{xi}) = A\sigma + EA\alpha\, \Delta T = p_M + p_T = F$$

$$F = \frac{EA}{L}\begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \bar{u}_{xi} \\ \bar{u}_{yi} \\ \bar{u}_{yi} \\ \bar{u}_{yj} \end{bmatrix}$$

نیروهای اولیه ترمومکانیکی در میله

❖ مشارکت نیروهای اولیه در معادلات تعادل:



$$\bar{\mathbf{f}} = \bar{\mathbf{f}}_M + \bar{\mathbf{f}}_T = \bar{\mathbf{K}}\,\bar{\mathbf{u}}$$

$$\bar{\mathbf{f}}_T = E\,A\,\alpha\,\Delta T \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

نیروهای اولیه ترمومکانیکی در میله

❖ مشارکت نیروهای اولیه در معادلات تعادل:

$$F = \bar{f}_{xj} = -\bar{f}_{xi}, \qquad \bar{f}_{yi} = \bar{f}_{yj} = 0$$

$$\begin{bmatrix} \bar{f}_{xi} \\ \bar{f}_{yi} \\ \bar{f}_{xj} \\ \bar{f}_{yj} \end{bmatrix} = \begin{bmatrix} \bar{f}_{Mxi} \\ \bar{f}_{Myi} \\ \bar{f}_{Mxj} \\ \bar{f}_{Myj} \end{bmatrix} + \begin{bmatrix} \bar{f}_{Txi} \\ \bar{f}_{Tyi} \\ \bar{f}_{Txj} \\ \bar{f}_{Tyj} \end{bmatrix} = \begin{bmatrix} \bar{f}_{Mxi} \\ \bar{f}_{Myi} \\ \bar{f}_{Mxj} \\ \bar{f}_{Myj} \end{bmatrix} + E\alpha\,\Delta T \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{u}_{xi} \\ \bar{u}_{yi} \\ \bar{u}_{xj} \\ \bar{u}_{yj} \end{bmatrix}$$

Assembly Rules with Thermomechanical Effects

1. *Compatibility: The joint displacements of all* members meeting at a joint must be the same.

2. *Equilibrium: The sum of effective forces exerted* by all members that meet at a joint must balance the external force applied to that joint.

No change in application of 1. To account for 2, the thermal forces are globalized and added to the mechanical forces during the merge process.

نیروهای اولیه ترمومکانیکی در میله

$$\mathbf{K}\mathbf{u} - \mathbf{f}_T = \mathbf{f}_M$$

$$\mathbf{K}\mathbf{u} = \mathbf{f}_M + \mathbf{f}_T = \mathbf{f}$$

effective force vector

Solve for node displacements **u**, then recover mechanical forces

$$\mathbf{f}_M = \mathbf{K}\mathbf{u} - \mathbf{f}_T$$

جمع بندی نیروهای اولیه ترمومکانیکی در میله

Disconnection
Localization
Member (Element) Relations ← Include $\bar{f}_I^{(e)}$

Globalization ← Transform to $f_I^{(e)}$
Merge ← Assemble into f_I
Application of BCs
Solution
Recovery of Derived Quantities ← Subtract $f_I^{(e)}$

تدوین یک برنامه در محیط MATLAB برای میله دو بعدی

Preprocessor
Nodes, Elements ,Members Properties (Geometry, Material)

↓

Element Stiffness Matrices
(Local Coordinates)

↓

Element Stiffness Matrices
(Global Coordinates)

↓

Forming Global Stiffness Matrix and Force Vector

↓

Apply the B.C.

↓

Solution Phase

↓

Postprocessing

تدوین یک برنامه در محیط MATLAB برای میله دو بعدی

```
Clear  all
Node=[
             node_no1   x1  y1
             node_no2  x2 y2
             ……………….];
Element=[
            elem_no   node_no1   node_no2   length   theta     E  A
                 ……………….];
BCdof=[…………………];
F_global=[ fx1   fy1   fx2  fy2 …………………]';

Connectivity=[
                         elem_no  Dof1  Dof2  Dof3  Dof4
                         ……………….];

(NN,MN)=size(Node)  ;
(NE,ME)=size(Element);

K_global=zeros(NN*2,NN*2);
```

```
 for i=1:NE
      Ke_local=Kelocal (Element(i,4), Element(i,6), Element(i,7));
% Kelocal is a Matlab function
      Te_rotatoin=T_rotation(Element(i,5));
% T_rotation is a Matlab function
      Ke_global==Te_rotation'*ke_local*Te_rotation;
      ke_assemble=Assemble(Ke_global,Connectivity(i,:),NN);
% Assemble is a Matlab function
      K_global= K_global+ke_assemble
End
F_global=[ fx1   fy1   fx2  fy2 …………………]';
% Assume that the external forces are in global coordinates
% If they are in local coordinates then they have to transfer using
% F_global=Te_rotation*Fe_local

% Applying the B.C
K_globalBC=Boundry_conditionK(K_global,BCdof)
F_globalBC=Boundry_conditionF(F_global, BCdof)

% Solution Phase
U=inv(K_globalBC)*F_globalBC;
```

تدوین یک برنامه در محیط MATLAB برای میله دو بعدی

```
% Postprocessing Phase
  for i=1:NE
   stress(i)=Stress_calc(U, Element(i,2), Element(i,3), ...
                              Element(i,4), Element(i,5) );
  end

  % end of the program
    end
```

مثال: یک برنامه MATLABبرای میله دو بعدی

```
E=2.95e11;
A=0.0001;
x1=0;
y1=0;
x2=0.4;
y2=0;
x3=0.4;
y3=0.3;
x4=0;
y4=0.3;
alpha1=0;
alpha2=90;
alpha3=atan(0.75)*180/pi;
k1=Bar2D2Node_Stiffness (E,A,x1,y1,x2,y2,alpha1)
k2=Bar2D2Node_Stiffness (E,A,x2,y2,x3,y3,alpha2)
k3=Bar2D2Node_Stiffness (E,A,x1,y1,x3,y3,alpha3)
k4=Bar2D2Node_Stiffness (E,A,x4,y4,x3,y3,alpha1)
KK=zeros(8,8);
```

مثال: یک برنامه MATLABبرای میله دو بعدی

```
KK=Bar2D2Node_Assembly (KK,k1,1,2);
KK=Bar2D2Node_Assembly (KK,k2,2,3);
KK=Bar2D2Node_Assembly (KK,k3,1,3);
KK=Bar2D2Node_Assembly (KK,k4,4,3)
k=KK([3,5,6],[3,5,6])
p=[20000;0;-25000];
u=k\p
q=[0 0 u(1) 0 u(2) u(3) 0 0]'
P=KK*q
u1=[q(1);q(2);q(3);q(4)]
stress1=Bar2D2Node_Stress(E,x1,y1,x2,y2,alpha1,u1)
u2=[q(3);q(4);q(5);q(6)]
stress2=Bar2D2Node_Stress(E,x2,y2,x3,y3,alpha2,u2)
u3=[q(1);q(2);q(5);q(6)]
stress3=Bar2D2Node_Stress(E,x1,y1,x3,y3,alpha3,u3)
u4=[q(7);q(8);q(5);q(6)]
stress4=Bar2D2Node_Stress(E,x4,y4,x3,y3,alpha1,u4)
```

مثال: یک برنامه MATLABبرای میله دو بعدی

```
function k=Bar2D2Node_Stiffness(E,A,x1,y1,x2,y2,alpha)

L=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
x=alpha*pi/180;
C=cos(x);
S=sin(x);
k=E*A/L*[C*C C*S -C*C -C*S; C*S S*S -C*S -S*S;
  -C*C -C*S C*C C*S; -C*S -S*S C*S S*S];
```

مثال: یک برنامه MATLABبرای میله دو بعدی

```
function forces=Bar2D2Node_Forces(E,A,x1,y1,x2,y2,alpha,u)

L=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
x=alpha*pi/180;
C=cos(x);
S=sin(x);
forces= E*A/L*[-C -S C S]*u;
```

مثال: یک برنامه MATLABبرای میله دو بعدی

```
function z = Bar2D2Node_Assembly(KK,k,i,j)

DOF(1)=2*i-1;
DOF(2)=2*i;
DOF(3)=2*j-1;
DOF(4)=2*j;
for n1=1:4
   for n2=1:4

KK(DOF(n1),DOF(n2))=KK(DOF(n1),DOF(n2))+k(n1,n2);
   end
end
z=KK;
```

مثال: یک برنامه MATLABبرای میله دو بعدی

```
function stress= Bar2D2Node_Stress(E,x1,y1,x2,y2,alpha,u)

L=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
x=alpha*pi/180;
C=cos(x);
S=sin(x);
stress=E/L*[-C -S C S]*u;
```

مراجع:

1- Lecture Notes in NONLINEAR FINITE ELEMENT METHODS, Carlos A. Felippa, University of Colorado, 1999

۲- مطالب درسی اجزای محدود، سعید ضیایی راد، دانشگاه صنعتی اصفهان