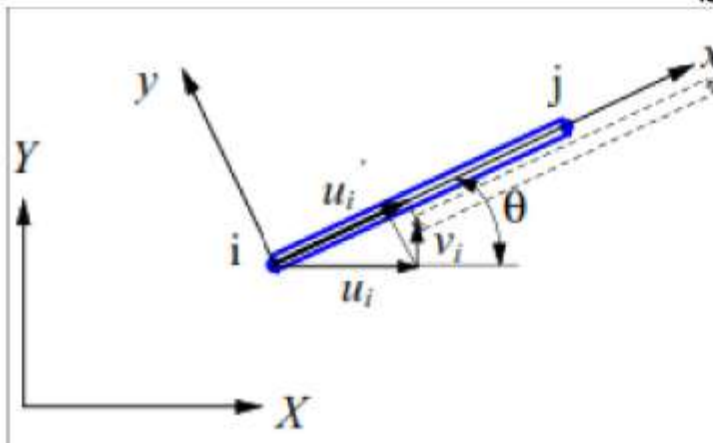


مطلوب است کد نویسی برنامه ای با متلب، که مشخصات هر نوع خرپای دو بعدی را از اکسل خوانده و ضمن تحلیل خرپا با روش اجزا محدود، اعضای فشاری و نیروی بحرانی آن را محاسبه و نمایش دهد.

مراحل انجام کار :

- ۱- تعیین مساحت اعضای خرپا
- ۲- تعیین ممان اینرسی خرپا
- ۳- تعیین مدول یانگ اعضای خرپا
- ۴- شماره گذاری گره های خرپا
- ۵- تعیین مختصات گره های خرپا
- ۶- تعیین راستاهایی که تکیه گاه وجود دارد
- ۷- تعیین نیروهای خارجی وارد بر گره ها
- ۸- محاسبه ماتریس سختی هر عضو
- ۹- محاسبه ماتریس سختی کل
- ۱۰- محاسبه قسمت هایی از فرمول تنش هر عضو
- ۱۱- محاسبه ماتریس دوران هر عضو
- ۱۲- تعیین طول هر عضو برای محاسبه نیروی کمانش بحرانی
- ۱۳- پاره سازی ماتریس سختی کل
- ۱۴- محاسبه جابجایی
- ۱۵- محاسبه نیروهای تکیه گاهی
- ۱۶- محاسبه نیروی هر عضو در مختصات محلی
- ۱۷- محاسبه نیروی هر عضو در مختصات سراسری
- ۱۸- محاسبه تنش هر عضو
- ۱۹- محاسبه نیروی هر عضو
- ۲۰- محاسبه نیروی کمانش
- ۲۱- تعیین اینکه نیروی عضو از بار بحرانی کمتر است یا بیشتر
- ۲۲- ذخیره اطلاعات در فایل اکسل
- ۲۳- رسم نمودار خرپا و شماره گذاری گره ها

• Bar Element in 2D Space



• یک المان میله در فضای ۲ بعدی در نظر بگیرید.

- مختصات X - Y ، مختصات کلی
- مختصات x - y ، مختصات محلی
- جابجایی ها در دو مختصات

Local	Global
x, y	X, Y
u'_i, v'_i	u_i, v_i
1 dof at node	2 dof's at node

• توجه شود در مختصات محلی فقط **یک درجه آزادی** در هر **نود** وجود دارد ولی در مختصات کلی **۲ درجه آزادی** در هر **نود** وجود دارد.

• استخراج ماتریس سفتی المان میله دو بعدی

• برای استخراج ماتریس سفتی ابتدا مولفه های جابجایی در مختصات محلی بر حسب مولفه های جابجایی در مختصات کلی نوشته می شوند.

$$\begin{aligned} u'_i &= u_i \cos q + v_i \sin q = \begin{bmatrix} l & m \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \\ v'_i &= -u_i \sin q + v_i \cos q = \begin{bmatrix} -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \end{aligned}$$

where $l = \cos q$, $m = \sin q$.

• در فرم ماتریسی

$$\begin{Bmatrix} u'_i \\ v'_i \end{Bmatrix} = \begin{bmatrix} l & m \\ -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \quad \text{or,} \quad \mathbf{u}'_i = \mathbf{T} \mathbf{u}_i$$

• که $\mathbf{T}$ ماتریس انتقال نام دارد

$$\mathbf{T} = \begin{bmatrix} l & m \\ -m & l \end{bmatrix}$$

is orthogonal, that is, $\mathbf{T}^{-1} = \mathbf{T}^T$.

- رابطه بین مولفه های جابجایی برای ۲ نود المان بصورت زیر نوشته می شود.

$$\begin{Bmatrix} u_i' \\ v_i' \\ u_j' \\ v_j' \end{Bmatrix} = \begin{bmatrix} l & m & 0 & 0 \\ -m & l & 0 & 0 \\ 0 & 0 & l & m \\ 0 & 0 & -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

or,

$$\mathbf{u}' = \mathbf{T} \mathbf{u} \quad \text{with } \mathbf{T} = \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix}$$

بردار نیرو نیز مشابه بردار جابجایی بصورت زیر انتقال می یابد.

$$\mathbf{f}' = \mathbf{T} \mathbf{f}$$

- حال معادله تعادل را در مختصات محلی می نویسیم (توجه شود معادله تعادل در مختصات محلی همان معادله تعادل المان یک بعدی است)

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

- با در نظر گرفتن سایر مولفه ها (v_i', v_j') در مختصات محلی داریم:

$$\frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_i' \\ v_i' \\ u_j' \\ v_j' \end{Bmatrix} = \begin{Bmatrix} f_i' \\ 0 \\ f_j' \\ 0 \end{Bmatrix} \quad \text{or,} \quad \mathbf{k}' \mathbf{u}' = \mathbf{f}'$$

$$\mathbf{k}' \mathbf{T} \mathbf{u} = \mathbf{T} \mathbf{f} \quad \xrightarrow[\mathbf{T}^T \mathbf{T} = \mathbf{I}]{\times \mathbf{T}^T} \quad \mathbf{T}^T \mathbf{k}' \mathbf{T} \mathbf{u} = \mathbf{f}$$

- بنابراین ماتریس سفتی المان میله دو بعدی بصورت زیر استخراج می شود.

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

$$\mathbf{k} = \frac{EA}{L} \begin{bmatrix} l^2 & lm & -l^2 & -lm \\ lm & m^2 & -lm & -m^2 \\ -l^2 & -lm & l^2 & lm \\ -lm & -m^2 & lm & m^2 \end{bmatrix} \quad \begin{matrix} u_i & v_i & u_j & v_j \end{matrix}$$

در فرم صریح ماتریس سفتی بصورت زیر نوشته می شود.

که کسینوسهای هادی l and m بصورت زیر تعریف می شوند.

$$l = \cos q = \frac{X_j - X_i}{L}, \quad m = \sin q = \frac{Y_j - Y_i}{L}$$

• تنش در المان ها

- رابطه تنش در المان دو بعدی از رابطه تنش در المان یک بعدی بدست می آید.

$$\mathbf{s} = E\mathbf{e} = E\mathbf{B}\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = E\begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} l & m & 0 & 0 \\ 0 & 0 & l & m \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

$$\mathbf{s} = \frac{E}{L}\begin{bmatrix} -l & -m & l & m \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

خواندن مشخصات خرپا

برای این کار اطلاعات ممان اینرسی هر عضو، مساحت هر عضو، مدول یانگ، گره های ابتدایی و انتهایی هر عضو، مختصات X و Y هر گره، مولفه های تکیه گاهی و نیروهای خارجی وارد بر خرپا، در فایل اکسل وارد می شود.

معرفی کدهای متلب

```
clc
clear
close all
format long

Name_File='Problem.xls';

Name_Sheet='Problem_2';

Module_E=xlsread(Name_File,Name_Sheet,'A2:A100');

Inertia=xlsread(Name_File,Name_Sheet,'B2:B100');

Area=xlsread(Name_File,Name_Sheet,'C2:C100');

Nodes=xlsread(Name_File,Name_Sheet,'D2:E100');

Matrix_XY=xlsread(Name_File,Name_Sheet,'F2:G100');

Nodes_Tekegah=xlsread(Name_File,Name_Sheet,'H2:H100');

force=xlsread(Name_File,Name_Sheet,'I2:I100');

Number_Element=size(Module_E,1);

Number_Node=size(Matrix_XY,1);

AllDof=2*Number_Node;
```

با این دستورات صفحه نمایش پاک و تمام متغیرها حذف، پنجره های رسم نمودار بسته و فرمت نمایش اعداد در حالت طولانی قرار می گیرد. نام فایل اکسل به برنامه معرفی می شود. نام شیتی که اطلاعات خرپا از آن قرار دارد به برنامه معرفی می شود. مدل یانگ تمام اعضا در متغیر Module_E ذخیره می شود ممان اینرسی تمام اعضا در متغیر Inertia ذخیره می شود مساحت تمام اعضا در متغیر Area ذخیره می شود شماره گره های ابتدایی و انتهایی در متغیر Nodes ذخیره می شود مختصات هر گره در متغیر Matrix_XY ذخیره می شود شماره گره های تکیه گاهی در متغیر Nodes_Tekegah ذخیره می شود نیروهای افقی و عمودی هر گره در متغیر force ذخیره می شود تعداد المانها محاسبه می شود تعداد گره ها محاسبه می شود ابعاد ماتریس سختی کل سازه محاسبه می شود

با دستورات زیر ماتریس سختی کل با تعداد درایه های AllDof ساخته شده و بعد برای هر المان ماتریس سختی محاسبه و در محل خود در ماتریس سختی کل قرار می گیرد. برای این منظور شماره گره هر المان در متغیر ElementCourse ذخیره می شود. محل قرارگیری درایه ها در ماتریس سختی کل در متغیر ElementDof ذخیره می شود. اختلاف مختصات های هر گره المان ، در راستای افقی و عمودی در متغیر های Delta_x و Delta_y ذخیره می شود. سپس اقدام به محاسبه طول عضو و بردارهای هادی می شود (متغیر های s, c, L). مرحله بعدی محاسبه ماتریس سختی المان با متغیر k است. چونکه طول هر المان ، ماتریس سختی هر المان ، ماتریس دوران و ماتریس تنش هر المان در مراحل بعدی مورد نیاز است اقدام به ذخیره آنها در متغیر های Ke و Sigmae و Te و Le می شود.

```
%Ejade Matrice Sakhti
stiffness=zeros(AllDof,AllDof);
for e=1:Number_Element
    ElementCourse=Nodes(e,:);
    ElementDof=[2*Nodes(e,1)-1 [2*Nodes(e,1)] [2*Nodes(e,2)-1 [2*Nodes(e,2)]];
    X1=Matrix_XY(ElementCourse(1),1);
    Y1=Matrix_XY(ElementCourse(1),2);
    X2=Matrix_XY(ElementCourse(2),1);
    Y2=Matrix_XY(ElementCourse(2),2);
    Delta_x=X2-X1;
    Delta_y=Y2-Y1;
    c=Delta_x/L;
    s=Delta_y/L;

    EAL=(Module_E(e)*Area(e)/L);

    k=(EAL)*[c^2,c*s,-c^2,-c*s
             c*s,s^2,-c*s,-s^2
             -c^2,-c*s,c^2,c*s
             -c*s,-s^2,c*s,s^2];
    stiffness(ElementDof,ElementDof)=stiffness(ElementDof,ElementDof)+k;
    Ke(1:4,e*4-3:e*4)=k;
    Sigmae(e,:)=(Module_E(e)/L)*[-c,-s,c,s];
    T=[c,s,0,0
       -s,c,0,0
        0,0,c,s
       -s,c,0,0];
    Te(1:4,e*4-3:e*4)=T;
    Le(e,1)=L;
end
```

```
%mohasebeye Displacements
U=zeros(AllDof,1);
activeDof=setdiff([1:AllDof]', [Nodes_Tekehah]);
U=stiffness(activeDof,activeDof)\force(activeDof);
Displacements=zeros(AllDof,1);
Displacements(activeDof)=U;
for e=1:AllDof
    disp(['Displacement None ',num2str(e),' : ' ,
num2str(Displacements(e))])
end
disp('*****')
%mohasebeye Reactions
F=stiffness*Displacements; %F=KU
reactions=F(Nodes_Tekehah);
for e=1:size(Nodes_Tekehah)
    disp(['Reactions None ',num2str(Nodes_Tekehah(e)),' : ' ,
num2str(reactions(e))])
end
disp('*****')
```

با این دستورات عملیات پاره سازی انجام و با تقسیم ماتریس سختی بر نیرو جابجایی محاسبه می شود. با این دستورات جابجایی هر گره در متغیر Displacements ذخیره و نمایش داده می شود.

با ضرب ماتریس سختی در جابجایی نیرو محاسبه و عکس العمل های تکیه گاهی نمایش داده می شود

با دستورات زیر محاسبات مربوط به نیروهای هر المان در مختصات محلی و نیروهای هر المان در مختصات سراسری و تنش هر عضو می شود. برای این منظور جابجایی های هر المان در متغیر ue ، ذخیره می شود. ماتریس سختی و ماتریس دوران هر عضو از متغیر های Ke و Te جدا و در

متغیر ke و te ذخیره می شود. با ضرب رابطه ماتریس سختی در جابجایی ماتریس نیرو در مختصات محلی بدست می آید ($fe=ke*ue$). با ضرب ماتریس دوران در ماتریس نیرو در مختصات محلی، ماتریس نیرو در مختصات سراسری بدست می آید ($F=te*fe$). برای محاسبه بار کمانش، اگر مقدار تنش (σ) کمتر از صفر باشد، عضو فشاری بوده و ضمن نمایش فشاری بودن عضو، بار بحرانی کمانش محاسبه و با نیروی فشاری که از تحلیل اجزا محدود بدست آمده است مقایسه می شود. از نیروی فشاری عضو کمتر از نیروی بحرانی باشد عبارت **Member Is Safe** نمایش داده می شود، در غیر این صورت عبارت **Member Is Not Safe** نمایش داده می شود. در این مرحله از محاسبات نتایجی که باید در اکسل ذخیره شود، در متغیرهای مورد نظر قرار می گیرد.

```
%mohasebeye Internal Forces
g=1;
for e=1:Number_Element%
    Ut=Displacements;
    ue=[Ut(2*Nodes(e,1)-1);Ut(2*Nodes(e,1));Ut(2*Nodes(e,2)-1);Ut(2*Nodes(e,2))];
    ke=Ke(1:4,e*4-3:e*4);
    te=Te(1:4,e*4-3:e*4);
    fe=ke*ue;
    F=te*fe;
    Sigma=Sigmae(e,:)*ue;

    Matrix_fe(e,:)=fe';
    Matrix_F(e,:)=F';
    Matrix_Sigma(e,\)=Sigma;

    disp(strcat('Local Displacement Matrix For Element : ',num2str(e)));
    disp(ue')

    disp(strcat('Local Stiffness Matrix For Element : ',num2str(e)));
    disp(ke')

    disp(strcat('Local Force Matrix For Element : ',num2str(e)));
    disp(fe')

    disp(strcat('global Force Matrix For Element : ',num2str(e)));
    disp(F')

    disp(strcat('global Stress For Element : ',num2str(e)));
    disp(Sigma)

    F1=sqrt(F(1)^2+F(2)^2);
    disp(strcat('global Force For Element : ',num2str(e)));
    disp(F1)

    if Sigma<0
        disp(strcat('Element : ',num2str(e),' : Feshari'))
        Matrix_Member(e,1)={'Feshari'};
        Fcr=((pi^2)*Module_E(e)*Inertia(e))/(Le(e,1)^2);
        disp('F Critical')
        disp(Fcr)

        if F1<Fcr
            disp('Member Is Safe')
        else
            disp('Member Is Not Safe')
        end
        Matrix_Fcr(g,1)=e;
        Matrix_Fcr(g,2)=F1;
        Matrix_Fcr(g,3)=Fcr;
        g=g+1;
    else
        disp(strcat('Element : ',num2str(e),' : Kesheshi'))
        Matrix_Member(e,1)={'Kesheshi'};
    end
    disp('*****')
end
end
```

```

Displacements=[1:AllDof]',Displacements];
xlswrite(['Result_',Name_Sheet, '.xlsx'],Displacements
,'Displacements');
reactions=[Nodes_Tekegah,reactions];
xlswrite(['Result_',Name_Sheet, '.xlsx'],reactions,'re
actions');

xlswrite(['Result_',Name_Sheet, '.xlsx'],stiffness,'st
iffness');

xlswrite(['Result_',Name_Sheet, '.xlsx'],Ke,'Ke');

xlswrite(['Result_',Name_Sheet, '.xlsx'],Matrix_Sigma,
'Sigma');

xlswrite(['Result_',Name_Sheet, '.xlsx'],Matrix_F,'Mat
rix_F');

xlswrite(['Result_',Name_Sheet, '.xlsx'],Matrix_fe,'Ma
trix_fe');

if (g >1)
xlswrite(['Result_',Name_Sheet, '.xlsx'],Matrix_Fcr,'M
atrix_e_F_Fcr');
end

xlswrite(['Result_',Name_Sheet, '.xlsx'],Matrix_Member
,'Matrix_Type_Member');

```

```

figure
hold on
for e=1:Number_Element
    ElementCourse=Nodes(e,:);
    ElementDof=[2*Nodes(e,1)-1 2*Nodes(e,1) 2*Nodes(e,2)-1
2*Nodes(e,2)];

x=[Matrix_XY(ElementCourse(2),1),Matrix_XY(ElementCourse(1),1)];

y=[Matrix_XY(ElementCourse(2),2),Matrix_XY(ElementCourse(1),2)];
plot(x,y,'r')
end
for e=1:size(Matrix_XY,1)
    text(Matrix_XY(e,1),Matrix_XY(e,2),num2str(e),'FontSize',10,
'Color','b')
end
hold off
grid on

ra_x=0.1*(max(Matrix_XY(:,1))-min(Matrix_XY(:,1)));
ra_y=0.1*(max(Matrix_XY(:,2))-min(Matrix_XY(:,2)));
xlim([min(Matrix_XY(:,1))-ra_x max(Matrix_XY(:,1))+ra_x])
ylim([min(Matrix_XY(:,2))-ra_y max(Matrix_XY(:,2))+ra_y])

disp('Finish')

```

با این دستورات ماتریس جابجایی در شیت **Displacements** ذخیره می شود

با این دستورات عکس العمل های تکیه گاهی در شیت **reactions** ذخیره می شود

با این دستورات ماتریس سختی کل در شیت **stiffness** ذخیره می شود

با این دستورات ماتریس سختی هر عضو در شیت **Ke** ذخیره می شود

با این دستورات ماتریس تنش هر عضو در شیت **Sigma** ذخیره می شود (ستون اول شماره عضو می باشد)

با این دستورات ماتریس نیرو در مختصات سراسری در شیت **Matrix_F** ذخیره می شود

با این دستورات ماتریس نیرو در مختصات محلی در شیت **Matrix_fe** ذخیره می شود

با این دستورات ماتریس نیروهای فشاری و نیروی بحرانی کمانش در شیت **Matrix_e_F_Fcr** ذخیره می شود. (ستون اول شماره عضو فشاری ستون دوم نیروی عضو و ستون سوم نیروی بحرانی کمانش می باشد)

با این دستورات ماتریس نوع عضو (فشاری یا کشش بودن) در شیت **Matrix_Type_Member** ذخیره می شود

با این دستورات المانهای خرپا رسم می شود.

با این دستورات شماره گره ها نمایش داده می شود.

با این دستورات محدوده رسم خرپا در صفحه بهتر نمایش داده می شود.

پیام خاتمه محاسبات نمایش داده می شود.