Tarea 1

Por: Cesar Fernando Gamba Tiusaba

Código: 215524

Problema.

Resultados programa.

Reacciones.

|  |  |
| --- | --- |
| Reacciones Programa | Reacciones Excel |
| -29,01234568 | -29,01234568 |
| -25 | -25 |
| -30,98765432 | -30,98765432 |
| 85 | 85 |

Fuerzas globales y locales elemento.

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento 1 | Globales | Locales | Excel Global |
| val(:,:,1) = | val(:,:,1) = | F 1-2 |
| 0 | -3,2407 | 0 |
| -3,2407 | 0 | -3,2407407 |
| 0 | 3,2407 | 0 |
| 3,2407 | 0 | 3,24074074 |

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento 2 | val(:,:,2) = | val(:,:,2) = | F 1-4 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento 3 | val(:,:,3) = | val(:,:,3) = | F 4-3 |
| 0 | 61,7593 | 0 |
| 61,7593 | 0 | 61,7592593 |
| 0 | -61,7593 | 0 |
| -61,7593 | 0 | -61,759259 |

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento 4 | val(:,:,4) = | val(:,:,4) = | F 2-3 |
| -30,9877 | -30,9877 | -30,987654 |
| 0 | 0 | 0 |
| 30,9877 | 30,9877 | 30,9876543 |
| 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento 5 | val(:,:,5) = | val(:,:,5) = | F 1-3 |
| -29 | -36,2654 | -29,012346 |
| -22 | 0 | -21,759259 |
| 29 | 36,2654 | 29,0123457 |
| 21,7593 | 0 | 21,7592593 |

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento 6 | val(:,:,6) = | val(:,:,6) = | F 2-4 |
| 31 | 38,7346 | 30,9876543 |
| -23 | 0 | -23,240741 |
| -31 | -38,7346 | -30,987654 |
| 23,2407 | 0 | 23,2407407 |

Código:

%Datos iniciales de entrada

nnd=4

nel=3

nsec=1

nmat=1

GIC=2

GL=nnd\*2;

nrest=GL-GIC;

nFext=0

ndesplazacono=1

%Dimensionamiento de matrices

Coord=zeros(nnd,2);

CAp=zeros(nnd,2);

Secc=zeros(nsec,1);

Mat=zeros(nmat,1);

Fext=zeros(nnd,2);

Elem=zeros(nel,4);

MGL=zeros(nnd,2);

KT=zeros(GL,GL);

GLel=zeros(nel,4);

ndesplazconocidosapoyos=zeros(nnd,2);

%Entrada de datos de los nudos

for i=1:1:nnd;

nudo=i;

Coord(i,1)=input(['Coordenada x del nudo ' num2str(i) '\n']);

Coord(i,2)=input(['Coordenada y del nudo ' num2str(i) '\n']);

end

%Entrada de datos de las restricciones

for ires=1:1:nrest;

i=input(['Nudo restringido:' '\n']);

j=input(['restringido en x=1, o en y=2?? ']);

CAp(i,j)=1;

end

%Entrada de desplazamientos

for cont1=1:1:ndesplazacono

i=input(['Nudo con desplazamiento' '\n']);

j=input(['Dirección de desplazamiento en x=1, y=2 ' '\n']);

if j<3

ndesplazconocidosapoyos(i,j)=input(['Magnitud del desplazamiento del nudo ' '\n']);

end

end

%Matriz de grados de libertad

cont1=1;

cont2=GIC+1;

for i=1:1:nnd

for j=1:1:2

if CAp(i,j)==1

MGL(i,j)=cont2;

cont2=cont2+1;

else

MGL(i,j)=cont1;

cont1=cont1+1;

end

end

end

%Secciones y materiales

for i=1:1:nsec;

Secc(i,1)=input(['Área de la sección ' num2str(i) '\n']);

end

for i=1:1:nmat;

Mat(i,1)=input(['Módulo de elasticidad del material ' num2str(i) '\n']);

end

%Fuerzas externas

for cont1=1:1:nFext

i=input(['Nudo con carga' '\n']);

j=input(['Dirección de la carga en x=1, y=2, xy=3' '\n']);

if j<3

Fext(i,j)=input(['Magnitud de la carga en el nudo ' num2str(i) '\n']);

else

F=input(['Magnitud de la carga en el nudo ' num2str(i) '\n']);

Ang=input(['Ángulo de inclinación de la carga con respecto a x positivo ' num2str(i) '\n']);

Fext(i,1)=F\*cosd(Ang);

Fext(i,2)=F\*sind(Ang);

end

end

%Identificación de los elementos

for i=1:1:nel

Elem(i,1)=input(['Nudo inicial del elemento ' num2str(i) '\n']);

Elem(i,2)=input(['Nudo final del elemento ' num2str(i) '\n']);

Elem(i,3)=input(['Tipo de sección del elemento ' num2str(i) '\n']);

Elem(i,4)=input(['Tipo de material del elemento ' num2str(i) '\n']);

end

%Determinación de la matriz de rigidez y de transformación de cada elemento

for i=1:1:nel

xi=Coord(Elem(i,1),1);

yi=Coord(Elem(i,1),2);

xf=Coord(Elem(i,2),1);

yf=Coord(Elem(i,2),2);

Delx=xf-xi;

Dely=yf-yi;

Long=sqrt(Delx^2+Dely^2);

Cs=Delx/Long;

Sn=Dely/Long;

T=[Cs -Sn 0 0;Sn Cs 0 0; 0 0 Cs -Sn;0 0 Sn Cs];

Matrizdetransformacion(:,:,i) = T;

AE=Secc(Elem(i,3),1)\*Mat(Elem(i,4),1);

%Matriz de rigidez local

kel=(AE/Long)\*[1 0 -1 0;0 0 0 0;-1 0 1 0;0 0 0 0];

kelocal(:,:,i)=kel;

%Matriz de rigidez global

keg=T\*kel\*T';

kelglobal(:,:,i)=keg;

%Identificación de grados de libertad por elemento

for j=1:1:2

GLel(i,j)=MGL(Elem(i,1),j);

GLel(i,j+2)=MGL(Elem(i,2),j);

end

%Ensamblaje de la matriz de rigidez

for l=1:1:4

for m=1:1:4

KT(GLel(i,l),GLel(i,m))=KT(GLel(i,l),GLel(i,m))+keg(l,m);

end

end

end

%Vector de desplazamientos conocidos

f = reshape(CAp',1,GL);

u = reshape(Fext',1,GL);

cont =0;

for i=1:1:GL

if f(1,i) == 0

cont = 1+cont;

Mfuezext(cont,1) = u(1,i);

end

end

f = reshape(CAp',1,GL);

p = reshape(ndesplazconocidosapoyos',1,GL);

cont =0;

for i=1:1:GL

if f(1,i) == 1

cont = 1+cont;

Vdesplacono(cont,1) = p(1,i);

end

end

Un = inv(KT(1:GIC,1:GIC))\* (Mfuezext -(KT(1:GIC,GIC+1:GL)\*Vdesplacono));

Fuerzasdescon = (KT(GIC+1:GL,1:GIC)\*Un)+((KT(GIC+1:GL,GIC+1:GL))\*Vdesplacono);

Desplaztotal = zeros(GL,1);

Desplaztotal(1:GIC,1)=Desplaztotal(1:GIC,1)+Un(1:GIC,1);

Desplaztotal(GIC+1:GL,1)=Desplaztotal(GIC+1:GL,1)+Vdesplacono(1:cont,1);

%Construccion matrices de desplazamniento para cada elemento

for i=1:1:nel

for j=1:1:4

Vdesplaglobal(j,1,i)=Desplaztotal(GLel(i,j),1);

end

end

%Construcción matrices de fuerzas globales para cada elemento

for i=1:1:nel

FGlobelem(:,1,i)= kelglobal(:,:,i)\*Vdesplaglobal(:,1,i);

end

%Construcción matrices de fuerzas locales para cada elemento

for i=1:1:nel

FLocelem(:,1,i)= Matrizdetransformacion(:,:,i)'\*FGlobelem(:,1,i);

end