Tarea 3

Por: Cesar Fernando Gamba Tiusaba

Código: 215524

Problema.

E = 1, I = 1.

10 KN 10 KN 12 KN/m

5 6 7 8

1 2 3 4

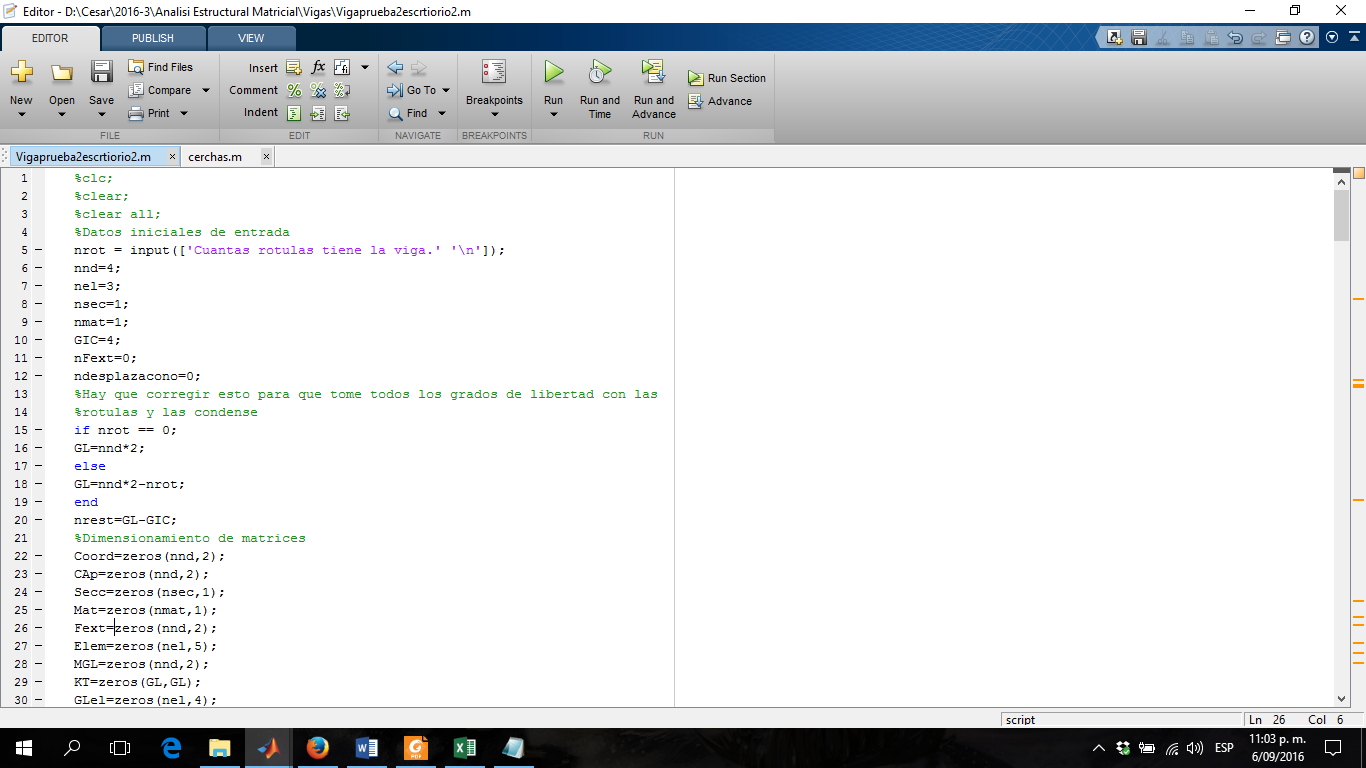
1

2

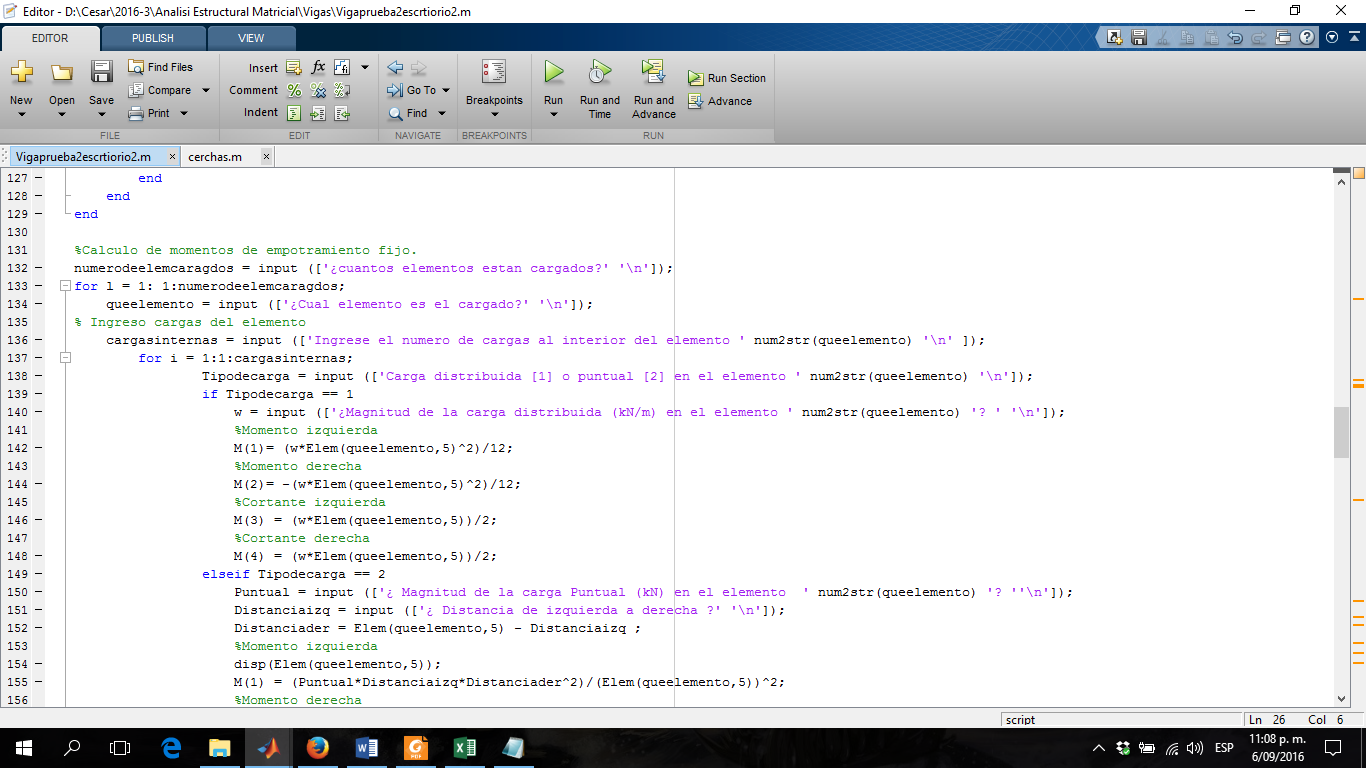
3

2 m 1 m 3 m

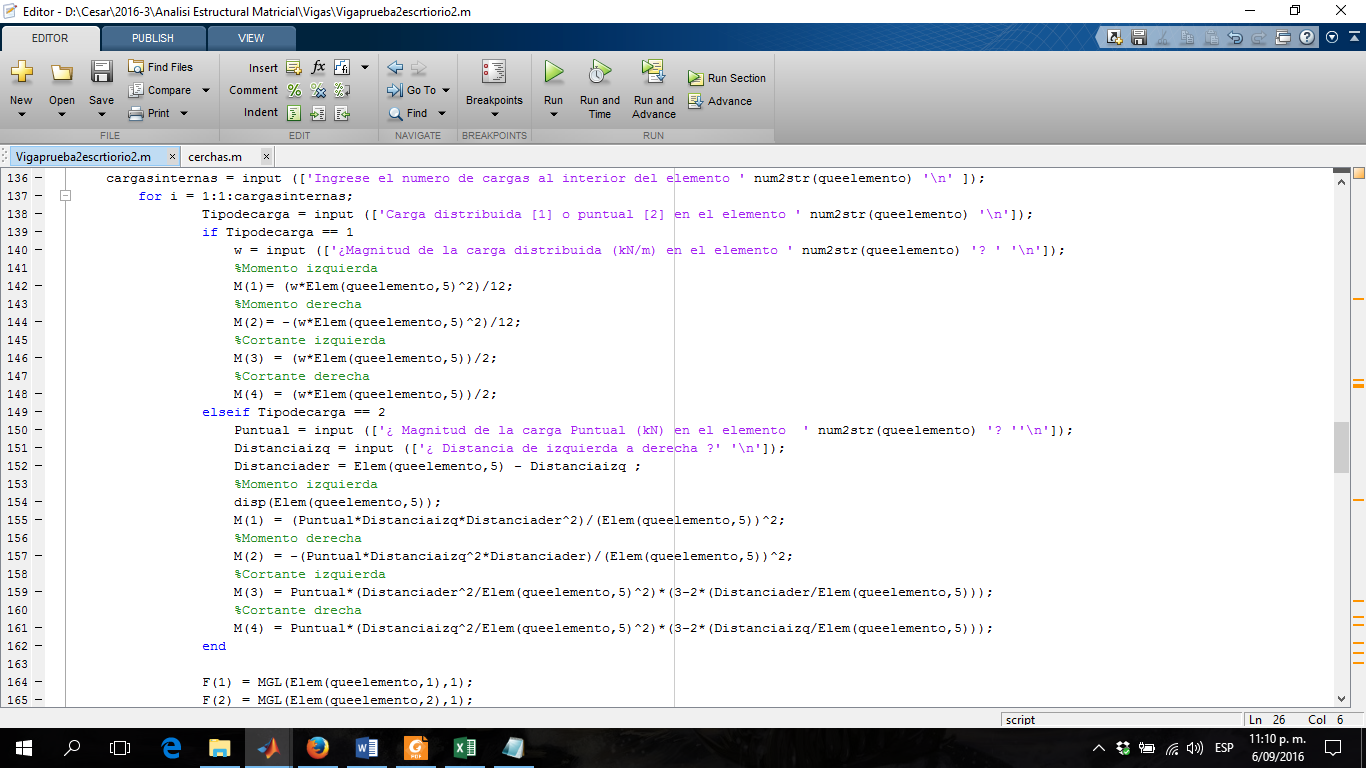
Ingreso de datos:



Se piden el numero de elementos cargados y seguidamente cual elemento es el cargado, tambien se le pregunta el numero de cargas que hay al interior de cada elemento, asi el usuario puede ingresar todas las acciones que quiera.



Luego dependiendo de si es una carga distribuida o una carga puntual ubicada a cualquier distancia del elemento, se realizaran las operacuiones correspondientes.



Finalmente se construye el vector de acciones fijas para nuestro caso.

|  |  |
| --- | --- |
| Grado de libertad | Magnitud de la acción |
|
| 1 | 2,5 |
| 2 | -1,25 |
| 3 | 7,75 |
| 4 | -9 |
| 5 | 5 |
| 6 | 10 |
| 7 | 23 |
| 8 | 18 |

Codigo:

clc;

clear;

clear all;

%Datos iniciales de entrada

nrot = input(['Cuantas rotulas tiene la viga.' '\n']);

nnd=3;

nel=2;

nsec=1;

nmat=1;

GIC=3;

nFext=0;

ndesplazacono=0;

%Hay que corregir esto para que tome todos los grados de libertad con las

%rotulas y las condense

if nrot == 0;

GL=nnd\*2;

else

GL=nnd\*2-nrot;

end

nrest=GL-GIC;

%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,5);

MGL=zeros(nnd,2);

KT=zeros(GL,GL);

GLel=zeros(nel,4);

ndesplazconocidosapoyos=zeros(nnd,2);

cRt = zeros(nnd,1);

Femporamiento=zeros(GL,1);

%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)=0;

end

%Entrada de datos de las restricciones

for ires=1:1:nrest;

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

j=input('restringido en giro=1, o en y=2?? ''\n');

CAp(i,j)=1;

disp (CAp);

end

%Entrada de rotulas

for irot=1:1:nrot

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

cRt(i,1)=1;

disp (cRt);

end

%Matriz de grados de libertad

% 1 = apoyo restringido, 0 = apoyo sin restricción.

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;

disp (MGL);

else

MGL(i,j)=cont1;

disp (MGL);

if cRt(i) == 1 && j == 1;

cont1=cont1;

else

cont1=cont1+1;

end

end

end

end

%Secciones y materiales

for i=1:1:nsec;

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

end

%Modulo de elasticidad

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([' Momento [1], Carga puntual [2]' '\n']);

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

disp (Fext);

end

%Entrada de desplazamientos

for cont1=1:1:ndesplazacono

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

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

ndesplazconocidosapoyos(i,j)=input(['Magnitud del desplazamiento del nudo ' num2str(i) '\n']);

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

%Calculo de la longitud de cada elemnento, recordar que se llee de

%izquierda a derecha se asume que se cuena de izquierda a derecha 1,2,3....

for i=1:1:nel

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

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

Elem(i,5)=abs(xf-xi);

end

disp (Elem);

%Transformacion de los vectores de fuerzas externas y vector

%desplazamientos.

for i=1:1:nnd

for j=1:1:2

if MGL(i,j) <= GIC

disp ('Fn 1');

FN(MGL(i,j),1)= Fext(i,j);

disp (FN);

else

UA(MGL(i,j)-GIC,1)= ndesplazconocidosapoyos(i,j);

FA(MGL(i,j)-GIC,1)= Fext(i,j);

disp ('Fn 2');

disp (FA);

disp ('Ua 2');

disp (UA);

end

end

end

%Calculo de momentos de empotramiento fijo.

numerodeelemcaragdos = input (['¿cuantos elementos estan cargados?' '\n']);

for l = 1: 1:numerodeelemcaragdos;

queelemento = input (['¿Cual elemento es el cargado?' '\n']);

% Ingreso cargas del elemento

cargasinternas = input (['Ingrese el numero de cargas al interior del elemento ' num2str(queelemento) '\n' ]);

for i = 1:1:cargasinternas;

Tipodecarga = input (['Carga distribuida [1] o puntual [2] en el elemento ' num2str(queelemento) '\n']);

if Tipodecarga == 1

w = input (['¿Magnitud de la carga distribuida (kN/m) en el elemento ' num2str(queelemento) '? ' '\n']);

%Momento izquierda

M(1)= (w\*Elem(queelemento,5)^2)/12;

%Momento derecha

M(2)= -(w\*Elem(queelemento,5)^2)/12;

%Cortante izquierda

M(3) = (w\*Elem(queelemento,5))/2;

%Cortante derecha

M(4) = (w\*Elem(queelemento,5))/2;

elseif Tipodecarga == 2

Puntual = input (['¿ Magnitud de la carga Puntual (kN) en el elemento ' num2str(queelemento) '? ''\n']);

Distanciaizq = input (['¿ Distancia de izquierda a derecha ?' '\n']);

Distanciader = Elem(queelemento,5) - Distanciaizq ;

%Momento izquierda

disp(Elem(queelemento,5));

M(1) = (Puntual\*Distanciaizq\*Distanciader^2)/(Elem(queelemento,5))^2;

%Momento derecha

M(2) = -(Puntual\*Distanciaizq^2\*Distanciader)/(Elem(queelemento,5))^2;

%Cortante izquierda

M(3) = Puntual\*(Distanciader^2/Elem(queelemento,5)^2)\*(3-2\*(Distanciader/Elem(queelemento,5)));

%Cortante drecha

M(4) = Puntual\*(Distanciaizq^2/Elem(queelemento,5)^2)\*(3-2\*(Distanciaizq/Elem(queelemento,5)));

end

F(1) = MGL(Elem(queelemento,1),1);

F(2) = MGL(Elem(queelemento,2),1);

F(3) = MGL(Elem(queelemento,1),2);

F(4) = MGL(Elem(queelemento,2),2);

for k=1:1:4;

Femporamiento(F(k)) = Femporamiento(F(k))+ M(k);

end

disp (Femporamiento);

end

end

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

for i=1:1:nel

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

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

Long=abs(xf-xi);

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

%Matriz de rigidez global

a = 12\*EI/Long^3;

a1 = 6\*EI/Long^2;

a2 = 4\*EI/Long;

a3 = 2\*EI/Long;

keg=[a a1 -a a1;

a1 a2 -a1 a3;

-a -a1 a -a1;

a1 a3 -a1 a2];

%Matriz de rigidez global

if cRt(Elem(i,1),1)~= 1 && cRt(Elem(i,2),1) ~= 1;

keg=[a a1 -a a1;

a1 a2 -a1 a3;

-a -a1 a -a1;

a1 a3 -a1 a2];

%Rotula al inicio

elseif cRt(Elem(i,1),1)== 1

a = 3\*EI/Long^3;

a1 = 3\*EI/Long^2;

a2 = 3\*EI/Long;

keg=[a 0 -a a1;

0 0 0 0;

-a 0 a -a1;

a2 0 -a1 a2];

%Rotula al final

else

a = 3\*EI/Long^3;

a1 = 3\*EI/Long^2;

a2 = 3\*EI/Long;

keg=[a a1 -a 0;

a1 a2 -a1 0;

-a -a1 a 0;

0 0 0 0;];

end

%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