Codigo que define el archive del programa ( lo bota matlab)

function varargout = pfinal(varargin)

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @pfinal\_OpeningFcn, ...

'gui\_OutputFcn', @pfinal\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

Funcion que inicia el programa, en este se definen las condicines iniciales que se desea que tenga el programa al abrirse

function pfinal\_OpeningFcn(hObject, eventdata, handles, varargin)

iniciar con los ejes en blanco

plot(0,0)

title ('Diagrama de Cortante')

xlabel ('Longitud (m)')

ylabel ('Fuerza cortante (N)')

grid off

grid

axes(handles.momento)

plot(0,0)

title ('Diagrama de Momento')

xlabel ('Longitud (m)')

ylabel ( 'Momento (N.m)')

handles.output = hObject;

guidata(hObject, handles);

function varargout = pfinal\_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

Creacion de objetos

function cortante\_CreateFcn(hObject, eventdata, handles)

function momento\_CreateFcn(hObject, eventdata, handles)

function longitud\_Callback(hObject, eventdata, handles)

function longitud\_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

Funcion de accionar el desplegable

function apoyos\_Callback(hObject, eventdata, handles)

global v

almacena v como un valr dependiendo el caso, y asimismo se ven los mensajes correspondientes en los cuadros de texto

v = get(handles.apoyos,'Value');

switch v

case 1

set (handles.msnlongitud,'string','');

set (handles.msnamplitud,'string','');

set (handles.apoyofijo,'string','');

set (handles.apoyopatin,'string','');

axes(handles.diagrama)

handles.diagrama=imread('CASO1.JPG');

imagesc(handles.diagrama)

axis off

fijo en A y patin en B

case 2

set (handles.msnlongitud,'string','Longitud de la viga [m].');

set (handles.msnamplitud,'string','Amplitud de la carga [N/m].');

set (handles.apoyofijo,'string','Distancia A-C [m].');

set (handles.apoyopatin,'string','Distancia B-D [m].');

axes(handles.diagrama)

handles.diagrama=imread('CASO2.JPG');

imagesc(handles.diagrama)

axis off

Fijo en B patin en A

case 3

set (handles.msnlongitud,'string','Longitud de la viga [m].');

set (handles.msnamplitud,'string','Amplitud de la carga [N/m].');

set (handles.apoyofijo,'string','Distancia B-D [m].');

set (handles.apoyopatin,'string','Distancia A-C [m].');

axes(handles.diagrama)

handles.diagrama=imread('CASO3.JPG');

imagesc(handles.diagrama)

axis off

end

cracion de la function del desplegable

function apoyos\_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

definicion de accionar el botoncalcular

function guardar\_Callback(hObject, eventdata, handles)

Definicion de variables

global cmin v longitud amplitud distanciaa distanciab Pmax cmax ancho eje fijo patin x y z C1 C2 C3 area M1 M2 M3 cte Mmax e

Cmin : carga minima

V: valor que toma una variable artificial para crear los ciclos, esta depende del caso seleccionado en el desplegable.

Longitud: almacena el dato de longitud de la viga[m]

Amplitud: almacena el dato de la amplitud de la carga [N/m]

Distanciaa: distancia del punto A al C (véase imagen) [m]

Distanciab: distancia del punto B al D (véase imagen) [m]

Pmax: punto donde la cortante corta el eje horizontal

(distanciaa+ancho-x)[m]

Cmax: carga máxima

Eje: función que grafica una recta horizontal de valor cero, esto para distinguir el ej positivo del negativo en cortante depnede de e

Fijo: valor de la reacción del apoyo fijo

Patin: valor de la reacción con apoyo de patin

X:recorrido desde cero hasta distancia a

Y: recorrido desde distanciaa hasta distanciaa+ancho

Z: Recorrido desde distanciaa+ancho hasta la longitud de la viga

C1,C2,C3: funciones a trozos de cortante, dependen de x, y y z respectivamente

M1,M2,M3: funciones a trozos de momento, dependen de x, y y z respectivamente

Área: multiplicar ancho por amplitud

Cte: constante de la función a trozos de C3

Mmax: calculo del momento maximo

E:recorrido de cero hasta la longitud de la viga

Captura de los datos necesarios brindados por el usuario

longitud = str2double(get(handles.longitud,'String'));

amplitud = str2double(get(handles.amplitud,'String'));

distanciaa = str2double(get(handles.distanciaa,'String'));

distanciab = str2double(get(handles.distanciab,'String'));

Caso donde el desplegable es “seleccionar tipos de apoyo”, se tiene en ciclo con los otros casos para que realice o no los calculos

En este caso como no se han seleccionado lso apoyos, no calcula nada.

if (v == 1)

set (handles.ax,'string','');

set (handles.ay,'string','');

set (handles.by,'string','');

set (handles.cmax,'string','');

set (handles.mmax,'string','');

set (handles.cmin,'string','');

set (handles.mmin,'string','');

grid on

axes(handles.cortante)

plot(0,0)

title ('Diagrama de Cortante')

xlabel ('Longitud (m)')

ylabel ('Fuerza cortante (N)')

grid off

grid

axes(handles.momento)

plot(0,0)

title ('Diagrama de Momento')

xlabel ('Longitud (m)')

ylabel ( 'Momento (N.m)')

else

Ciclo que agrega los textos e imagenes indicadas segun el caso de los apoyos, para presenter el analisis final(en este las imagenes contienen las reacciones)

if (v==2)

set (handles.msnlongitud,'string','Longitud de la viga [m].');

set (handles.msnamplitud,'string','Amplitud de la carga [N/m].');

set (handles.apoyofijo,'string','Distancia A-C [m].');

set (handles.apoyopatin,'string','Distancia B-D [m].');

set (handles.text9,'string','Ax');

axes(handles.diagrama)

handles.diagrama=imread('RESCASO2.JPG');

imagesc(handles.diagrama)

axis off

else

set (handles.text9,'string','Bx');

set (handles.msnlongitud,'string','Longitud de la viga [m].');

set (handles.msnamplitud,'string','Amplitud de la carga [N/m].');

set (handles.apoyofijo,'string','Distancia B-D [m].');

set (handles.apoyopatin,'string','Distancia A-C [m].');

grid

axes(handles.diagrama)

handles.diagrama=imread('RESCASO3.JPG');

imagesc(handles.diagrama)

axis off

end

Calculo del ancho de la carga distribuida

ancho = longitud-distanciaa-distanciab;

Calculo del valor equivalente de la carga distribuida

area = amplitud\*ancho;

Calculo de las reacciones

patin = (area\*(distanciaa+(ancho/2))/longitud);

fijo = area-patin;

Funciones a trozos junto con sus ejes

Se definen 3 ejes (x,y,z,e) que serán las absisas de cada función a trozos, (C1,M1,C2,M2,C3,M3,eje)

Esto con el fin que cada función sea evaluada únicamente en el intervalo que le corresponde.

x = 0:0.0001:(distanciaa);

C1 = (x<=(distanciaa)).\*(fijo)+(x==0).\*(-fijo);

M1 = (x<=(distanciaa)).\*(fijo\*x);

y = distanciaa:0.0001:(distanciaa+ancho);

C2 =(y <= (distanciaa+ancho)).\*(fijo+amplitud\*distanciaa-(amplitud\*y));

M2 =(y <= (distanciaa+ancho)).\*(fijo\*y +amplitud\*distanciaa\*y -amplitud\*(y.^2)/2 -(amplitud\*distanciaa.^2)/2);

cte= fijo\*(distanciaa+ancho) +amplitud\*distanciaa\*(distanciaa+ancho) -amplitud\*((distanciaa+ancho).^2)/2 -(amplitud\*distanciaa.^2)/2;

z = (distanciaa+ancho):0.0001:longitud;

C3 = (z <= (longitud)).\*(-patin)+(z==longitud).\*(patin);

M3 = (z < (longitud)).\*(-patin\*z+patin\*longitud);

e = 0:0.0001:longitud;

eje = (e<=longitud).\*(0);

Punto donde la cortante es cero, por ende es maximo momento

Pmax = distanciaa+ancho-(ancho/(1+(fijo/patin)));

Evaluacion de los valores de carga máxima y minima( se compara cual es el mayor para asi presentarlo en los cuadros de texto de forma correcta

if (fijo<=patin)

cmin=fijo;

cmax =patin;

else

cmin=patin;

cmax = fijo;

end

Calculo del momento maximo

Mmax = fijo\*Pmax +amplitud\*distanciaa\*Pmax -amplitud\*(Pmax.^2)/2 -(amplitud\*distanciaa.^2)/2;

Graficar las funciones a trozos calculadas

grid on

axes(handles.cortante)

plot(x,C1,'Blue',y,C2,'Red',z,C3,'Green',e,eje,'Black');

title ('Diagrama de Cortante')

xlabel ('Longitud (m)')

ylabel ('Fuerza cortante (N)')

grid off

grid on

axes(handles.momento)

plot(x,M1,'Blue',y,M2,'Red',z,M3,'Green');

title ('Diagrama de Momento')

xlabel ('Longitud (m)')

ylabel ('Momento (N.m)')

grid off

Presentar los resultados en cuadros de texto

set (handles.ax,'string','0');

set (handles.ay,'string',fijo);

set (handles.by,'string',patin);

set (handles.cmax,'string',cmax);

set (handles.cmin,'string',cmin);

set (handles.mmax,'string',Mmax);

set (handles.mmin,'string',min(M2));

end

Creacion del boton de calculo

function guardar\_CreateFcn(hObject, eventdata, handles)

Captura de amplitud

function amplitud\_Callback(hObject, eventdata, handles)

function amplitud\_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

Definicion de la function que captura distancia ac

function distanciaa\_Callback(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function distanciaa\_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

Definicion de la funcin que captura la distancia bc

function distanciab\_Callback(hObject, eventdata, handles)

function distanciab\_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

Deficicion de funciones

function mmax\_CreateFcn(hObject, eventdata, handles)

function ax\_CreateFcn(hObject, eventdata, handles)

function ay\_CreateFcn(hObject, eventdata, handles)

function cmax\_CreateFcn(hObject, eventdata, handles)

function by\_CreateFcn(hObject, eventdata, handles)

boton de borrar

function pushbutton2\_Callback(hObject, eventdata, handles)

deja en blanclo los cuadros de texto

set (handles.ax,'string','');

set (handles.ay,'string','');

set (handles.by,'string','');

set (handles.cmax,'string','');

set (handles.mmax,'string','');

set (handles.longitud,'string','');

set (handles.amplitud,'string','');

set (handles.distanciaa,'string','');

set (handles.distanciab,'string','');

set (handles.cmin,'string','');

set (handles.mmin,'string','');

deja los ejes de cortante y momento en blanco

grid on

axes(handles.cortante)

plot(0,0)

title ('Diagrama de Cortante')

xlabel ('Longitud (m)')

ylabel ('Fuerza cortante (N)')

grid off

grid

axes(handles.momento)

plot(0,0)

title ('Diagrama de Momento')

xlabel ('Longitud (m)')

ylabel ( 'Momento (N.m)')

axes(handles.diagrama)

handles.diagrama=imread('CASO1.JPG');

imagesc(handles.diagrama)

axis off

funciones que definen los objetos

function pushbutton2\_CreateFcn(hObject, eventdata, handles)

function msnlongitud\_CreateFcn(hObject, eventdata, handles)

function msnamplitud\_CreateFcn(hObject, eventdata, handles)

function apoyofijo\_CreateFcn(hObject, eventdata, handles)

function apoyopatin\_CreateFcn(hObject, eventdata, handles)

function text9\_CreateFcn(hObject, eventdata, handles)

function cmin\_CreateFcn(hObject, eventdata, handles)

function mmin\_CreateFcn(hObject, eventdata, handles)