function varargout = classworkmatlab(varargin)

% CLASSWORKMATLAB MATLAB code for classworkmatlab.fig

% CLASSWORKMATLAB, by itself, creates a new CLASSWORKMATLAB or raises the existing

% singleton\*.

%

% H = CLASSWORKMATLAB returns the handle to a new CLASSWORKMATLAB or the handle to

% the existing singleton\*.

%

% CLASSWORKMATLAB('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in CLASSWORKMATLAB.M with the given input arguments.

%

% CLASSWORKMATLAB('Property','Value',...) creates a new CLASSWORKMATLAB or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before classworkmatlab\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to classworkmatlab\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help classworkmatlab

% Last Modified by GUIDE v2.5 09-Jun-2018 12:41:37

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @classworkmatlab\_OpeningFcn, ...

'gui\_OutputFcn', @classworkmatlab\_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

% End initialization code - DO NOT EDIT

% --- Executes just before classworkmatlab is made visible.

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

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to classworkmatlab (see VARARGIN)

% Choose default command line output for classworkmatlab

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes classworkmatlab wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

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

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

function input1\_Callback(hObject, eventdata, handles)

% hObject handle to input1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input1 as text

% str2double(get(hObject,'String')) returns contents of input1 as a double

% --- Executes during object creation, after setting all properties.

function input1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to input1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

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

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

end

function input2\_Callback(hObject, eventdata, handles)

% hObject handle to input2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input2 as text

% str2double(get(hObject,'String')) returns contents of input2 as a double

% --- Executes during object creation, after setting all properties.

function input2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to input2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

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

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

end

% --- Executes on button press in plot.

function plot\_Callback(hObject, eventdata, handles)

% hObject handle to plot (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%%

%%input f1 and f2

zoom on

zoom out

in1 =get(handles.input1,'String');%input f1

in2 =get(handles.input2,'String');%input f2

FS1 =['@(x)',char(in1)];%add @(x) to f1

FS2 =['@(x)',char(in2)];%add @(x) to f2

FS3 =['@(x)',char(in1),'-(',char(in2),')'];%add @(x) to f1-f2

fs1 = str2func(FS1);%Turn the f1 into a function

fs2 = str2func(FS2);%Turn the f2 into a function

fs3=str2func(FS3);%Turn f1-f2 into a function

xx = -100:0.0001:100;%X, Vector points of functions

fx1=fs1(xx);%Y1 vector

fx2=fs2(xx);%y2 vector

%%

%%find complex points and Define those points in infinity

fx1(imag(fx1) ~= 0) = inf ;%If there are complex points in f1, we will make them points in infinity

fx2(imag(fx2) ~= 0) = inf ;%If there are complex points in f2, we will make them points in infinity

%%

%%if the fx1 or fx2 is a number Turn it into a vector

if length(fx1)==1

a=fx1;

fx1= rot90(a\*ones(length(xx),1));

end

if length(fx2)==1

a=fx2;

fx2= rot90(a\* ones(length(xx),1));

end

%%

%%polt f1 and f2

plot(xx,fx1,'r',xx,fx2,'b')%show the graph,f1 in red , f2 in blue

hold on

%%

%%Delete points in infinity

t = fs3(xx)> 0;%find when F3 bigger then 0

i0 = find(diff(t(:))~=0);%Found different points from t

[oo1]=find(isinf(fx1));%Finding infinity points f1

[oo2]=find(isinf(fx2));%Finding infinity points f2

l1 = length(oo1);

l2 = length(oo2);

i0 = [i0(:)';i0(:)'+1];%Add a line to the matrix. The line will be +1

for ll1=1:l1

LL1=oo1(ll1);

i0(:,any(i0==LL1,1)) = [];%Delete points in infinity f1

end

for ll2=1:l2

LL2=oo2(ll2);

i0(:,any(i0==LL2,1)) = [];%Delete points in infinity f2

end

%%

%%find cutting points

nold = size(i0,2);%Number of cutting points

xout1 = zeros(nold,1);%The size X vector is like the number of cutting point

for jj = 1:nold

xout1(jj) = fzero(fs3,xx(i0(:,jj))); %Placement of the X vector points

end

%%

%%Deleting points unreasonable cutting Matlab functions

xout = unique(xout1);%Delete duplicate points

yout1=fs1(xout);%Y1 (cutting point)

yout2=fs2(xout);%Y2 (cutting point)

yout3=yout2-yout1;% Subtract the value of the cut points in both functions

[oo3]=find(yout3>0.01|yout3<-0.01);%Finding problematic points

xout(oo3)=[];%delete problematic points(X values)

yout=fs1(xout);%the final value of cutting point (Y values)

n=length(xout);%number of cutting points

%%

%%display the number of cutting points

handles.n=n;

guidata(hObject,handles);

message = [num2str(n),' cutting points between -100 to 100'];%define characters that will be our message with the number of cutting point

set(handles.numofcutpoints,'string',message);%Put the massage in the static text in the gui

%%

%%display the x values and y values

data = zeros(length(xout),2);% matrix of zero : n\*2

data(:,1)=xout;%The X in the left column

data(:,2)=yout;%The y in the left column

set(handles.uitable3,'Data',data)%Put the matrix in table in the gui

%%

%%plot cutting point

if n~=0

plot(xout,yout,'k\*')% show the cutting points on the graph

end

%%

%%audio

if (get(handles.radiobutton1,'Value')) == 0;%if mute is not pushed

if n ==0%0 cutting point

[y,Fs] = audioread('0cuttingpoint.m4a');

sound(y,Fs);

end

if n ==1%1 cutting point

[y,Fs] = audioread('1cuttingpoint.m4a');

sound(y,Fs);

end

if n==2%2 cutting points

[y,Fs] = audioread('2cuttingpoint.m4a');

sound(y,Fs);

end

if n==3 %3 cutting points

[y,Fs] = audioread('3cuttingpoint.m4a');

sound(y,Fs);

end

if n==4 %4 cutting points

[y,Fs] = audioread('4cuttingpoint.m4a');

sound(y,Fs);

end

if n>4%more then 4 cutting points

[y,Fs] = audioread('alotofcuttingpoint.m4a');

sound(y,Fs);

end

end

hold off

function numofcutpoints\_Callback(hObject, eventdata, handles)

% hObject handle to numofcutpoints (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numofcutpoints as text

% str2double(get(hObject,'String')) returns contents of numofcutpoints as a double

% --- Executes during object creation, after setting all properties.

function numofcutpoints\_CreateFcn(hObject, eventdata, handles)

% hObject handle to numofcutpoints (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

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

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

end

% --- Executes on button press in radiobutton1.

function radiobutton1\_Callback(hObject, eventdata, handles)

% hObject handle to radiobutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1

% --- Executes on selection change in listbox1.

function listbox1\_Callback(hObject, eventdata, handles)

% hObject handle to listbox1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1 contents as cell array

% contents{get(hObject,'Value')} returns selected item from listbox1

% --- Executes during object creation, after setting all properties.

function listbox1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to listbox1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.

% See ISPC and COMPUTER.

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

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

end

% --- Executes during object creation, after setting all properties.

function table\_CreateFcn(hObject, eventdata, handles)

% hObject handle to table (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.

function uitable2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to uitable2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% --- Executes when entered data in editable cell(s) in uitable3.

function uitable3\_CellEditCallback(hObject, eventdata, handles)

% hObject handle to uitable3 (see GCBO)

% eventdata structure with the following fields (see MATLAB.UI.CONTROL.TABLE)

% Indices: row and column indices of the cell(s) edited

% PreviousData: previous data for the cell(s) edited

% EditData: string(s) entered by the user

% NewData: EditData or its converted form set on the Data property. Empty if Data was not changed

% Error: error string when failed to convert EditData to appropriate value for Data

% handles structure with handles and user data (see GUIDATA)

% --- Executes when figure1 is resized.

function figure1\_SizeChangedFcn(hObject, eventdata, handles)

% hObject handle to figure1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes on mouse press over figure background, over a disabled or

% --- inactive control, or over an axes background.

function figure1\_WindowButtonDownFcn(hObject, eventdata, handles)

% hObject handle to figure1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)