



EP375 Computational Physics

Topic 6

MATLAB GUI



Department of
Engineering Physics

University of Gaziantep

Feb 2014

Content

- 1. Introduction**
- 2. Initializing GUI**
- 3. Adder Program**
- 4. Plotting**

MATLAB®
The Language of Technical Computing

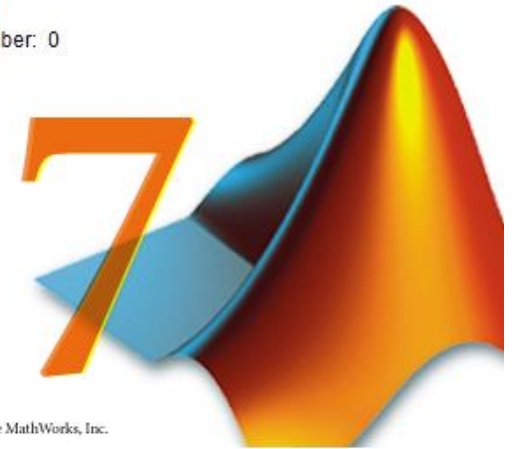
Version 7.0.0.19920 (R14)

May 06, 2004

License Number: 0

Ahmet

GU



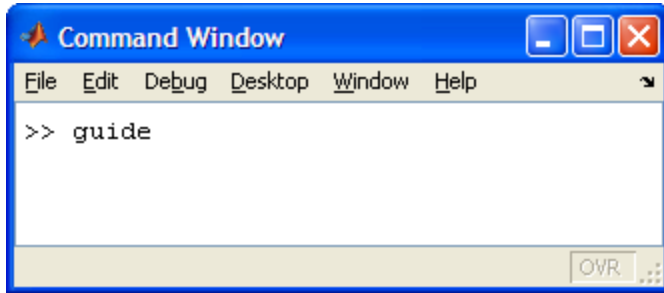
Copyright 1984–2004, The MathWorks, Inc.

1. Introduction

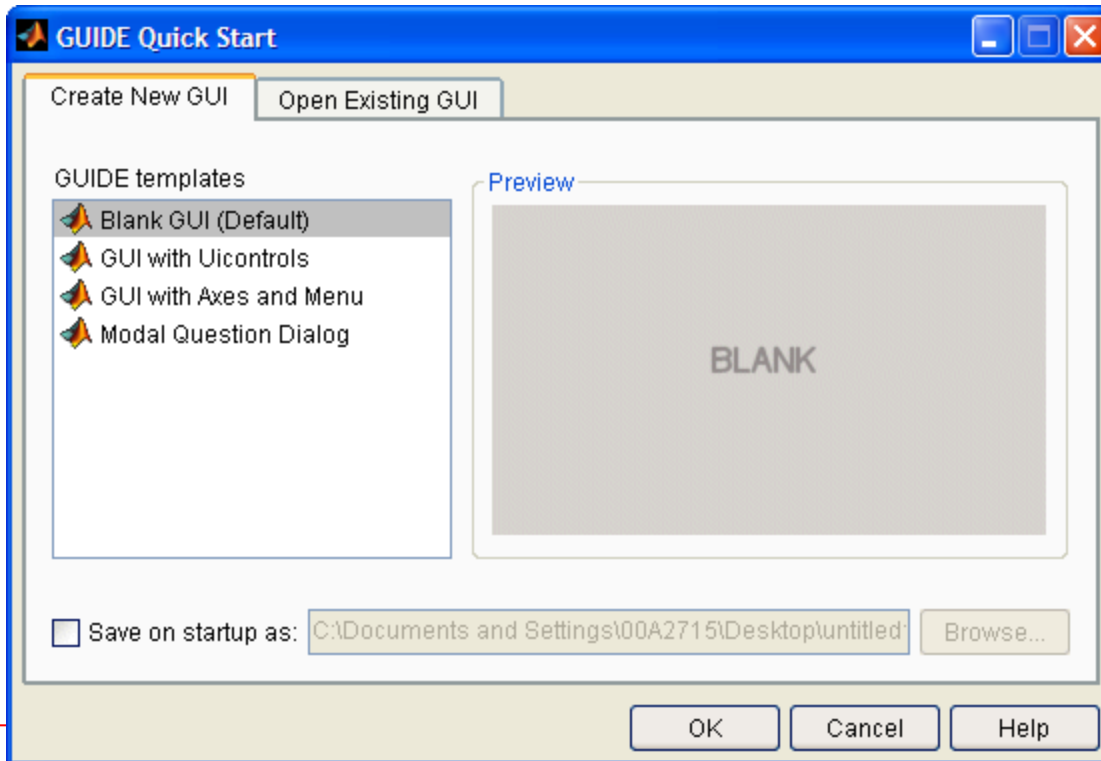
- A graphical user interface (GUI) is a type of user interface that allows users to interact with electronic devices with images rather than text commands.
- GUIs are used is because it makes things simple for the end-users of the program.
- See Also:
 - * `http://en.wikipedia.org/wiki/Graphical_user_interface`
 - * `http://blinkdagger.com/matlab/matlab-gui-graphical-user-interface-tutorial-for-beginners/`

2. Initializing GUI

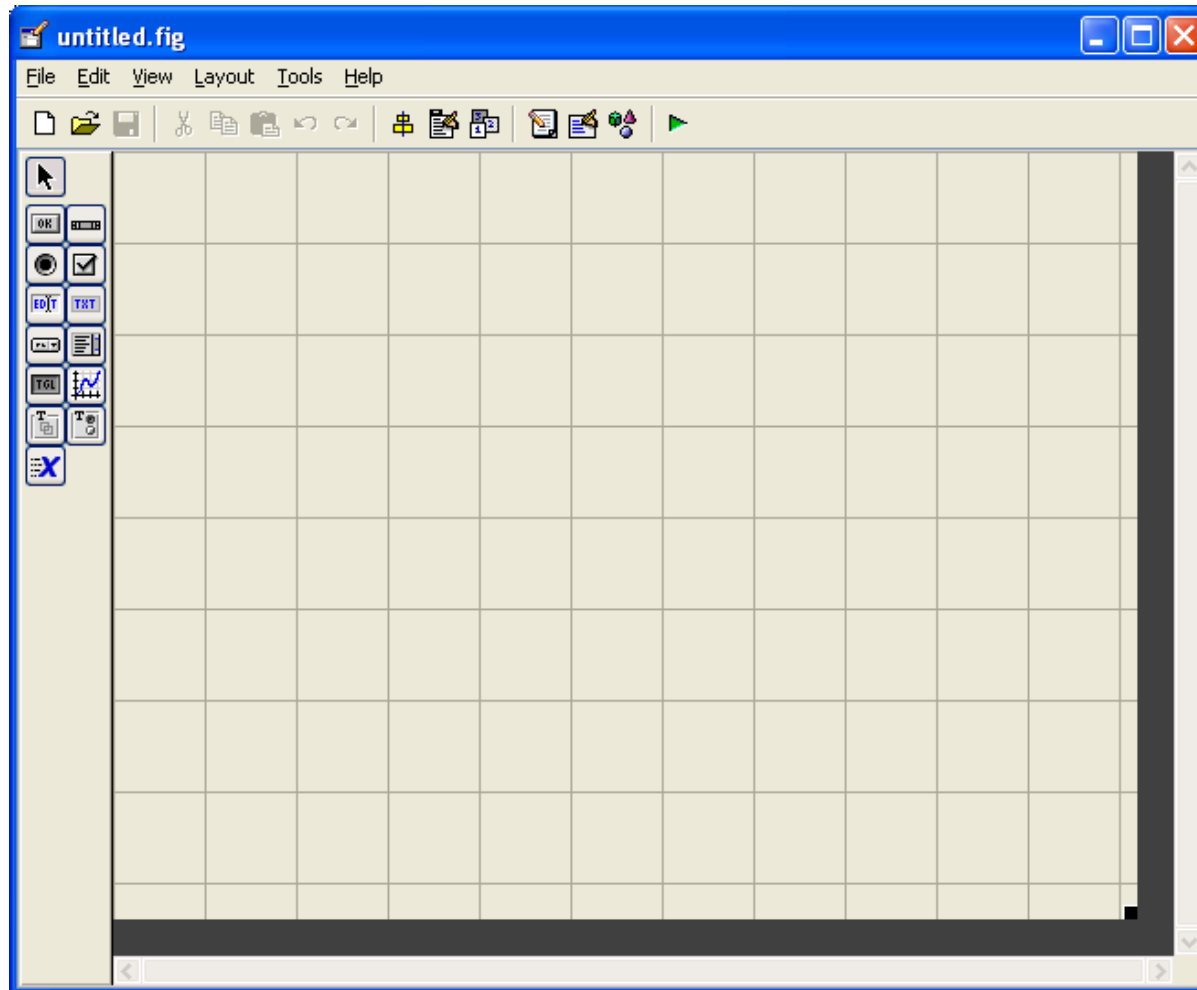
Open up MATLAB. Go to the command window and type in guide.



Choose the first option Blank GUI (Default).



You should see the following screen appear.



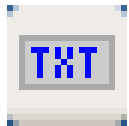
You can design your gui program using the tool box left.

3. Adder Program

Add the following components to the canvas:



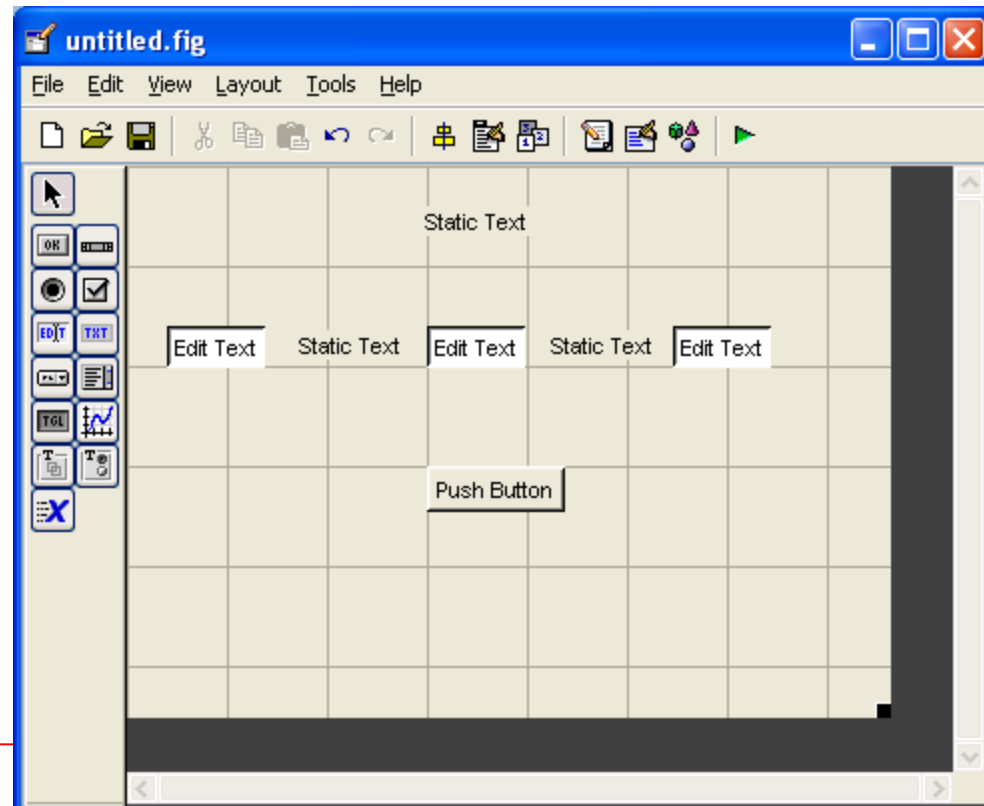
3 Edit Text components



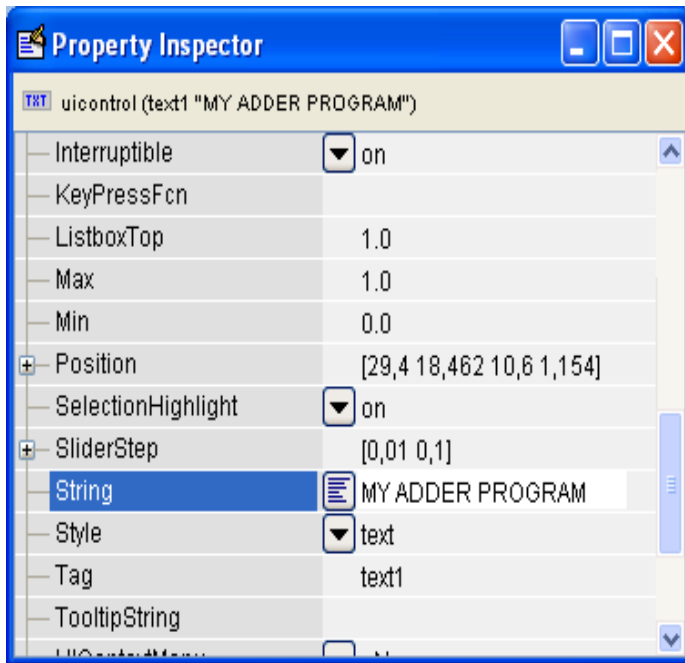
4 Static Text component



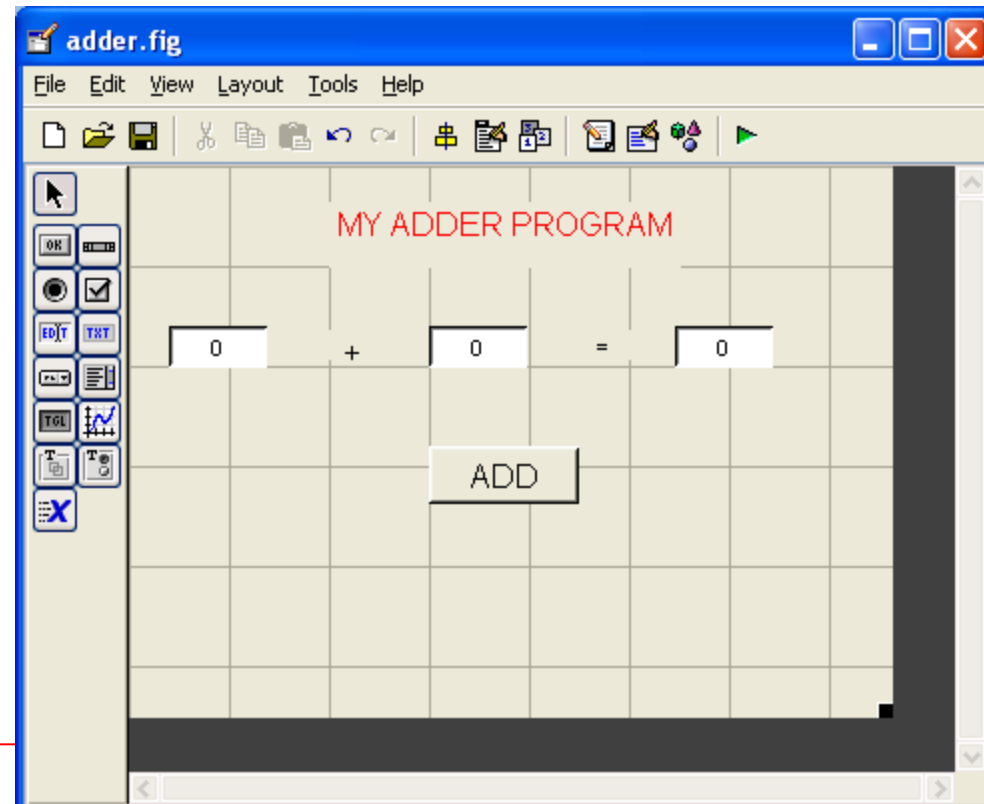
1 Push Button



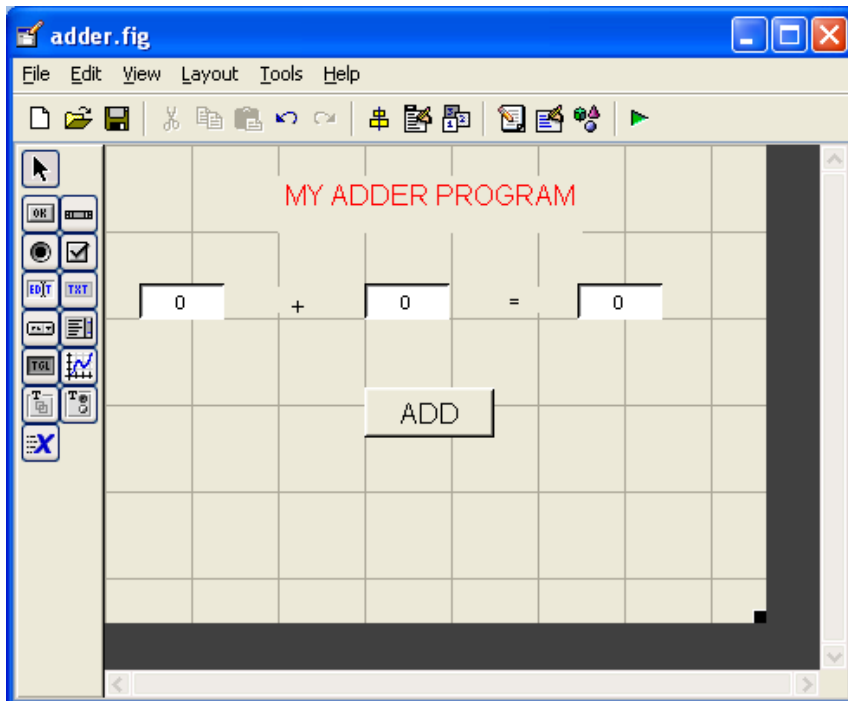
Double click each component. You will see the *Property Inspector* window. This allows you to modify the properties of a component.



Change
String,
FontSize and
ForegroundColor parameters.



Save the settings as **adder.fig**
MATLAB automatically
generates an m-file to go
along with the figure that you
just put together.



```
Editor - C:\Documents and Settings\Peng-User\Desktop\Programming\MATLAB\adder.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
1 function varargout = adder(varargin)
2 % ADDER M-file for adder.fig
3 % ADDER, by itself, creates a new ADDER or raises the existing
4 % singleton*.
5 %
6 % H = ADDER returns the handle to a new ADDER or the handle to
7 % the existing singleton*.
8 %
9 % ADDER('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in ADDER.M with the given input arguments.
11 %
12 % ADDER('Property','Value',...) creates a new ADDER or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before adder_OpeningFunction gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to adder_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Copyright 2002-2003 The MathWorks, Inc.
24
25 % Edit the above text to modify the response to help adder
26
27 % Last Modified by GUIDE v2.5 28-Dec-2011 09:58:16
28
29 % Begin initialization code - DO NOT EDIT
30 gui_Singleton = 1;
31 gui_State = struct('gui_Name', mfilename, ...
32 'gui_Singleton', gui_Singleton, ...
33 'gui_OpeningFcn', @adder_OpeningFcn, ...
34 'gui_OutputFcn', @adder_OutputFcn, ...
35 'gui_LayoutFcn', [], ...
36 'gui_Callback', []);
37 if nargin && ischar(varargin{1})
38 gui_State.gui_Callback = str2func(varargin{1});
39 end
40
41 if nargin
42 [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
43 else
44 gui_mainfcn(gui_State, varargin{:});
45 end
46 % End initialization code - DO NOT EDIT
47
```

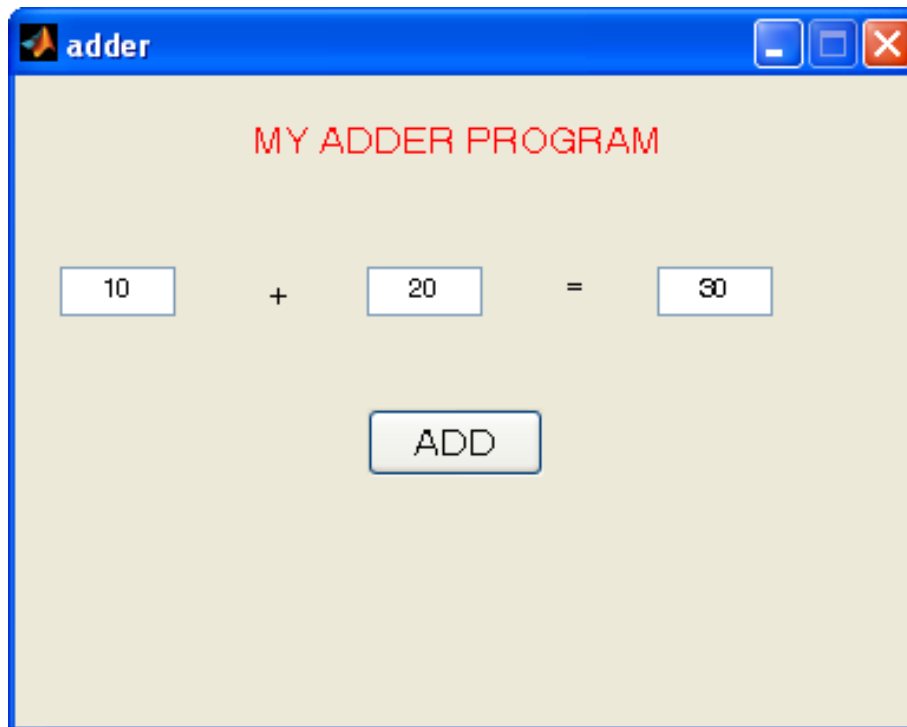
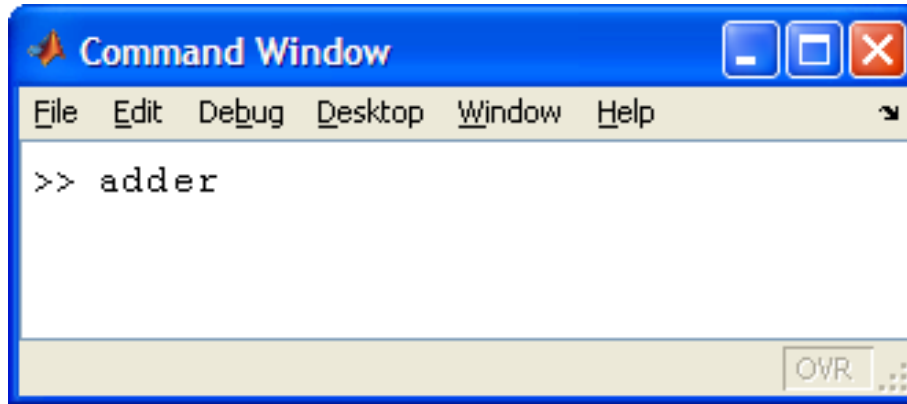

Modify the function

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

as follows:

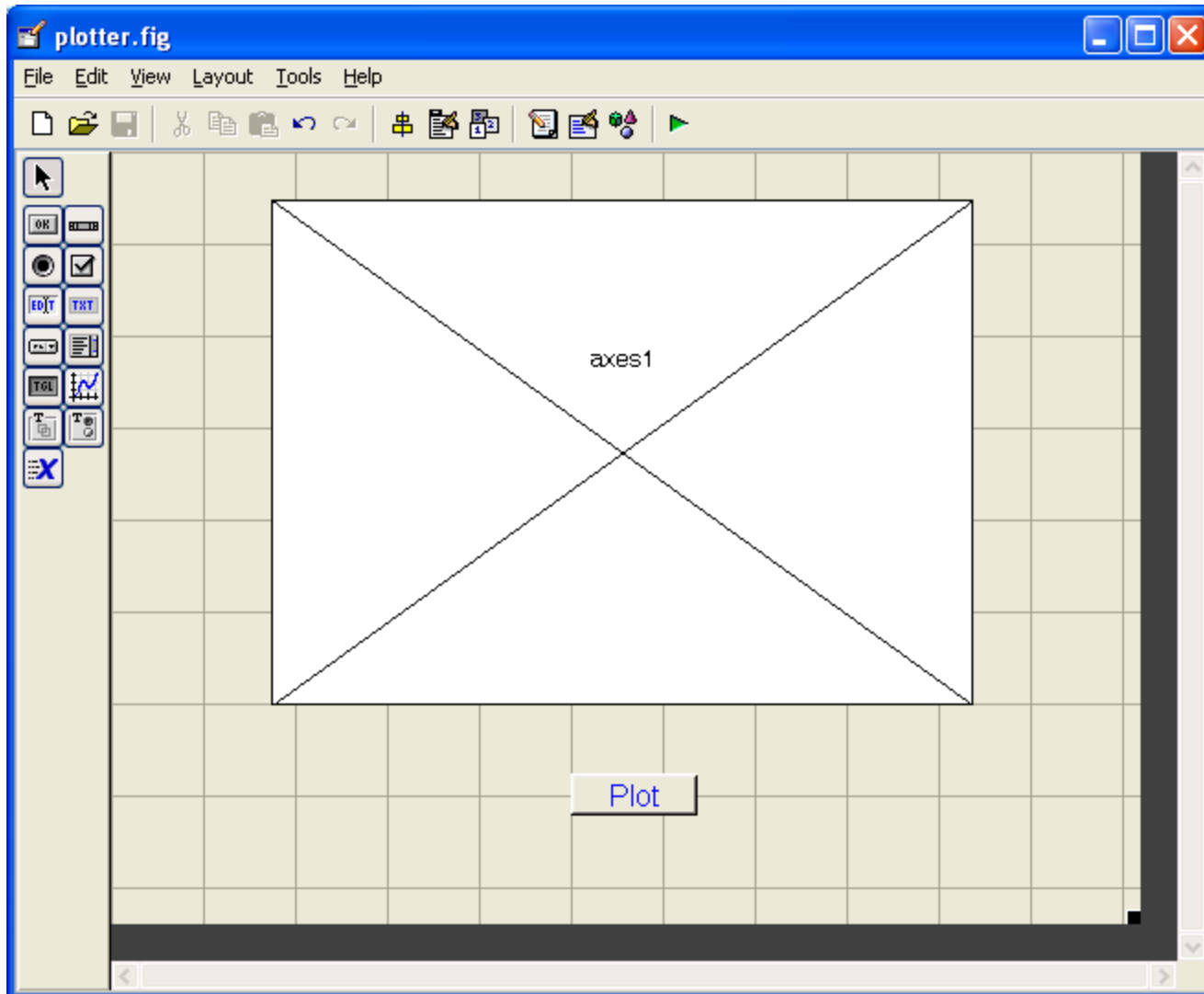
```
function pushbutton1_Callback(hObject, eventdata, handles)
    x = str2num(get(handles.edit1, 'String'));
    y = str2num(get(handles.edit2, 'String'));
    z = num2str(x+y);
    set(handles.edit3, 'String', z);
    guidata(hObject, handles); % update handles structures
end
```

Finally, type in the name of the GUI at the command prompt



4. Plotting on the Canvas

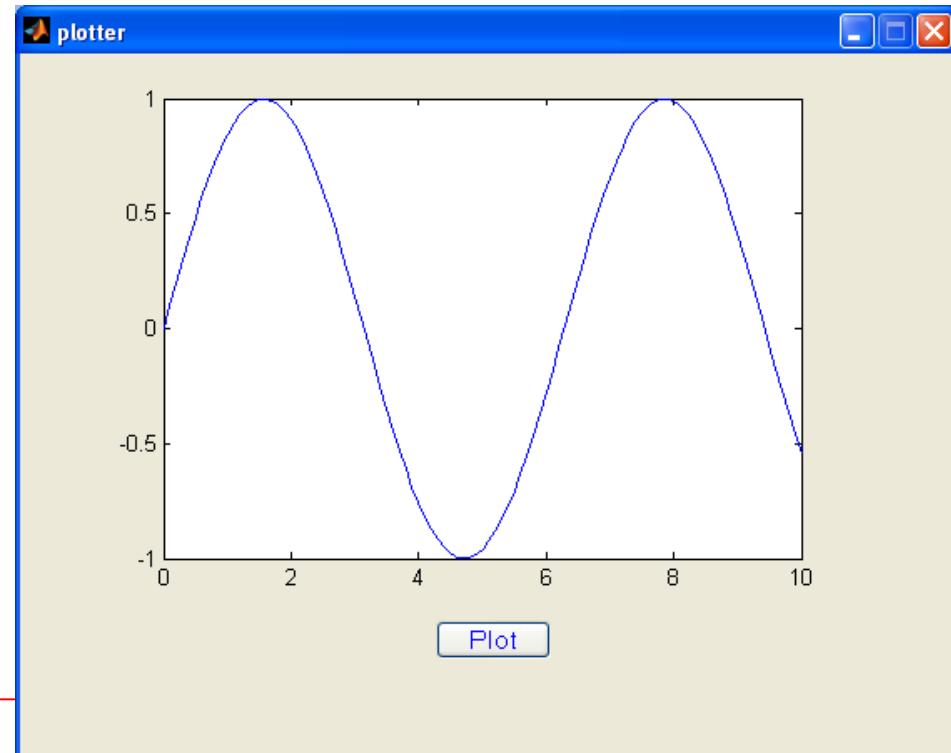
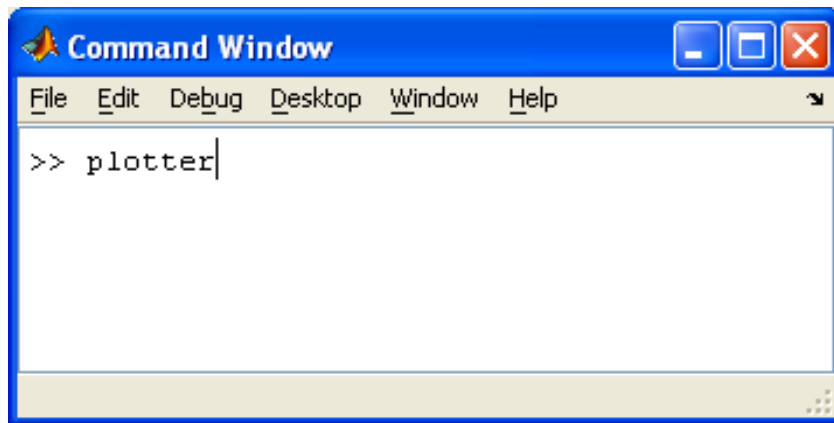
Add the following components to the canvas and save as **plotter.fig**.



Modify the `pushbutton1_Callback()` function in `plotter.m`

```
function pushbutton1_Callback(hObject, eventdata, handles)
    x = 0:0.1:10;
    axes(handles.axes1);
    plot(x, sin(x));
```

Type in the name of the GUI at the command prompt:

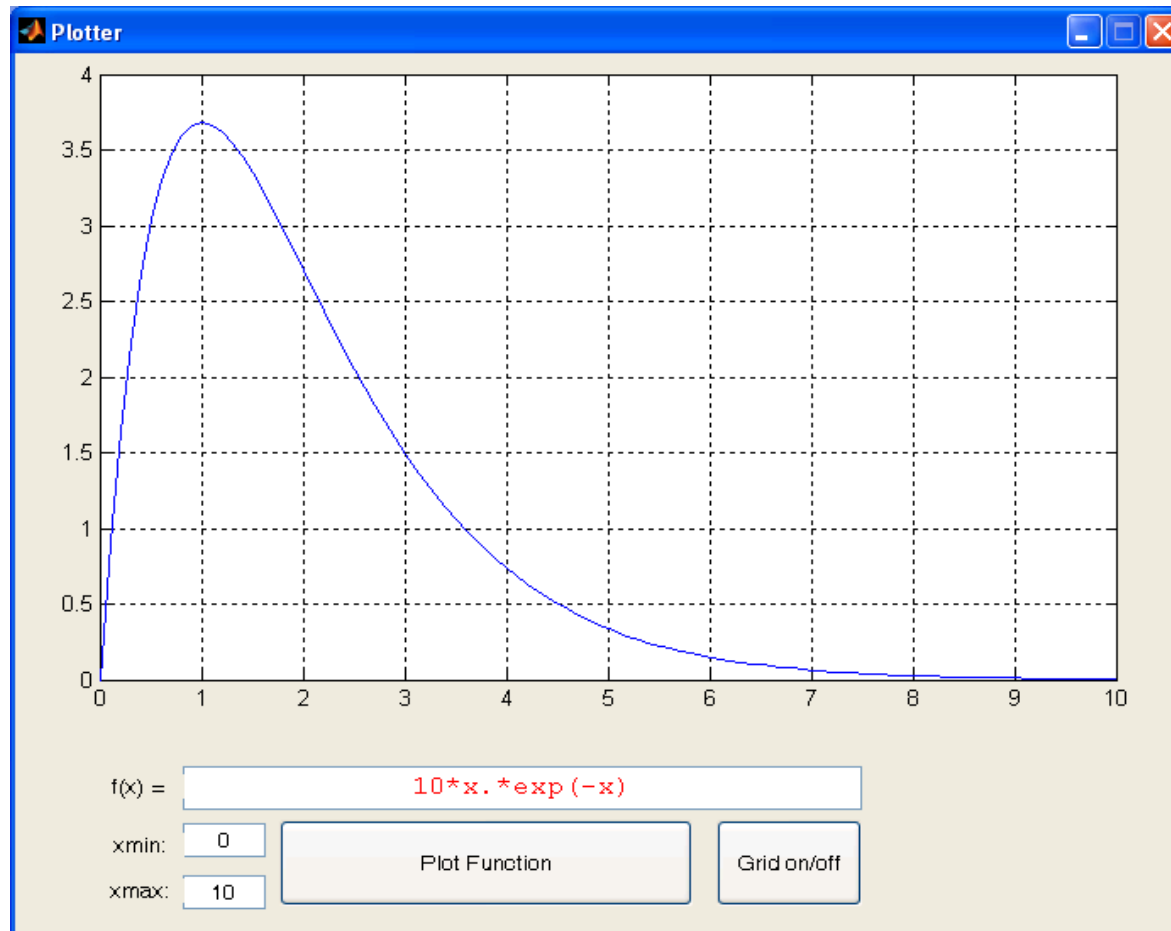


HW 1:

Write a GUI program similar to the Windows standard calculator.



HW 2: Write a GUI program that reads a function in an edit box and plots the function. There must be additional push button for grid on/off.



Hint:

```
>> f = input( 'Enter function (of x) to be plotted: ', 's' );  
x.*x  
>> x = 0:0.01:10;  
>> plot(x, eval(f))
```

References:

[1]. http://en.wikipedia.org/wiki/Graphical_user_interface

[2]. <http://blinkdagger.com/matlab/matlab-gui-graphical-user-interface-tutorial-for-beginners/>