



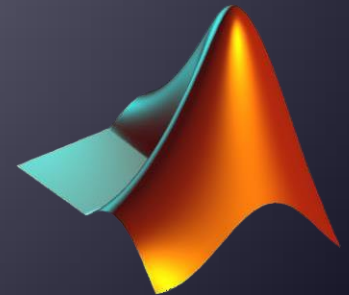
## Computer Skills and Computing for BME II



# Lecture 1

## Introduction to Graphical User Interfaces

MSc. Reem Salah hassan



# MATLAB GUI (Graphical User Interface)

- Why use a GUI in MATLAB?
- The main reason for using GUI is because it makes things easier for end-users to use the application or program, it is difficult to let users execute a program through the command window or command lines.
- Users looking for easy apps to use.
- GUI make programs friendly and attractive to be used.

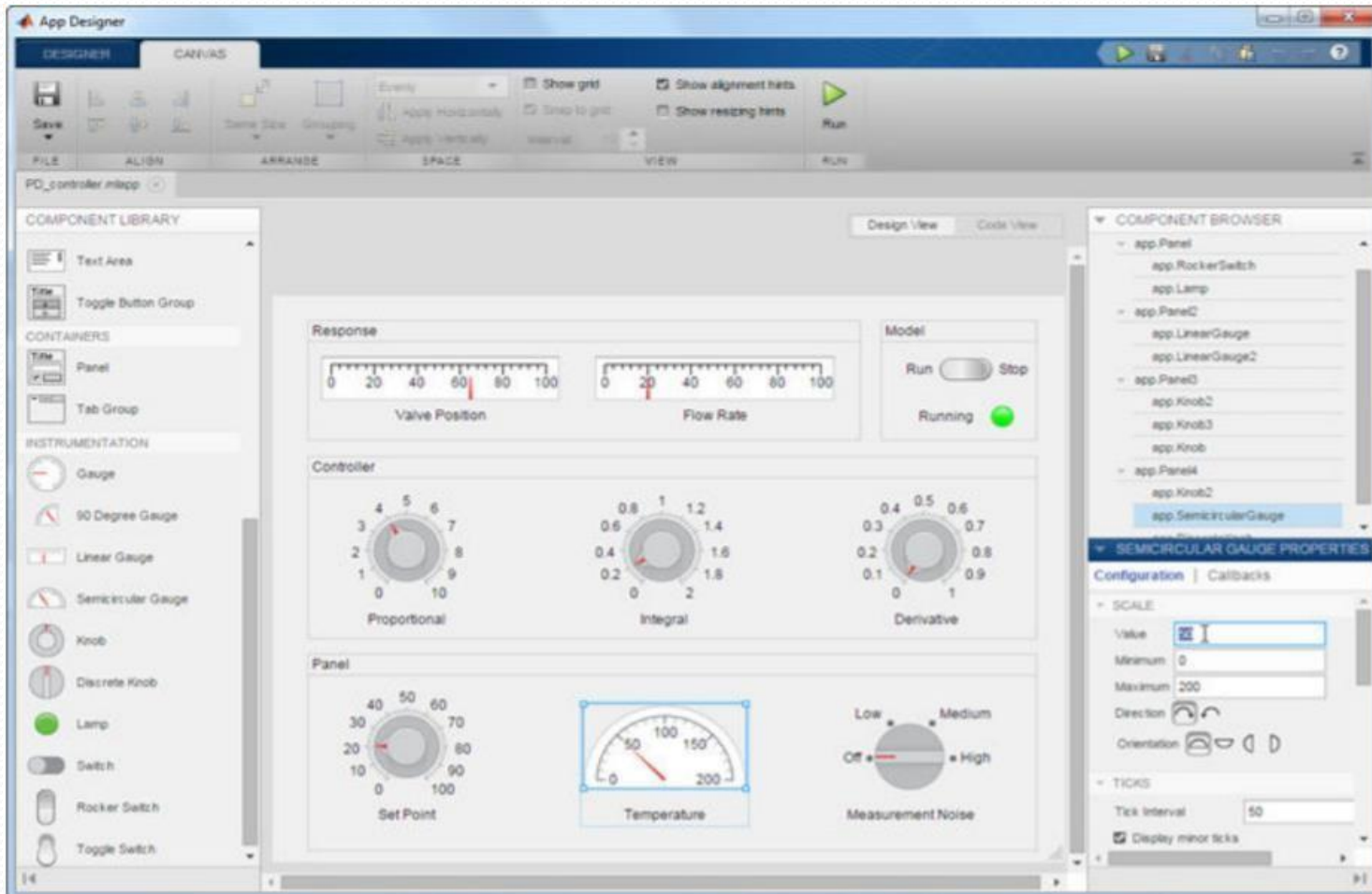
# Ways to Build Apps:

- There are different ways to build MATLAB apps:
- “Use App Designer”
- “Use GUIDE”
- “Use MATLAB Functions to Create Apps Programmatically”
- Each of these approaches offers a different workflow and a slightly different set of functionality. The best choice for you depends on your project requirements and how you prefer to work.

# Use App Designer

App Designer is a rich drag-and-drop environment introduced in R2016a, and it is the recommended environment for building most apps. It includes a fully integrated version of the MATLAB editor. The layout and code views are tightly linked, so changes you make in one view immediately affect the other. A larger set of interactive controls is available, including gauges, lamps, knobs, and switches. Most graphics functionality is supported.

# Use App Designer: Design Tools



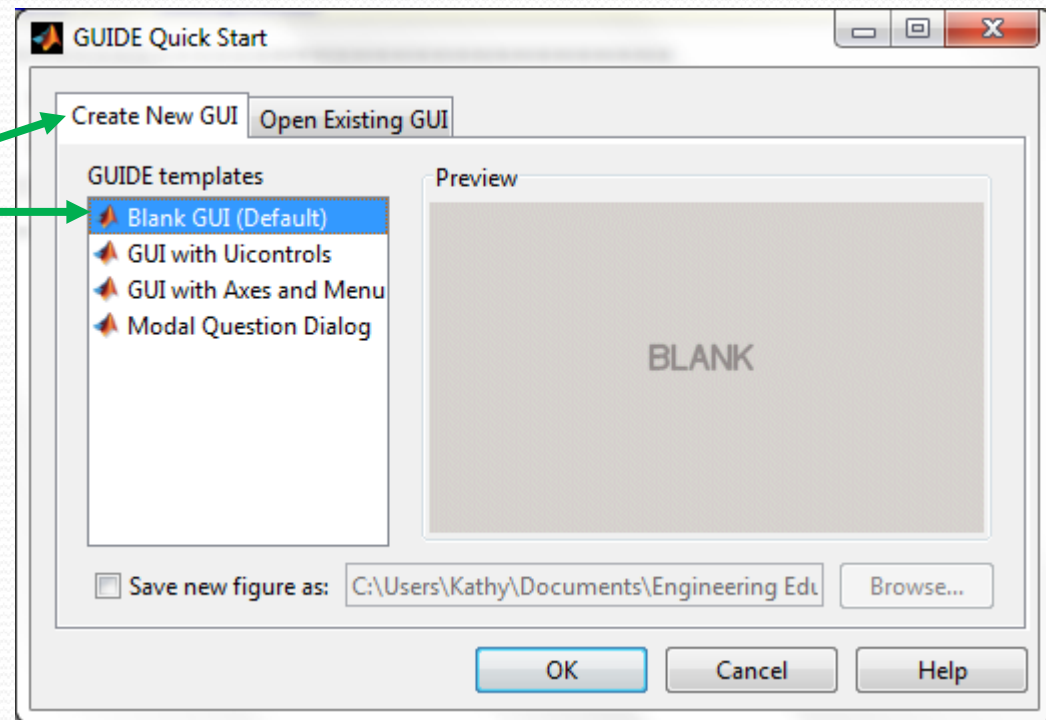
# Use GUIDE

GUIDE is a drag-and-drop environment for laying out user interfaces (UIs). You code the interactive behavior of your app separately, in the MATLAB editor. Apps created with GUIDE are compatible with almost all other releases, and they support all the graphics functionality in MATLAB.

# GUI Development Environment (guide)

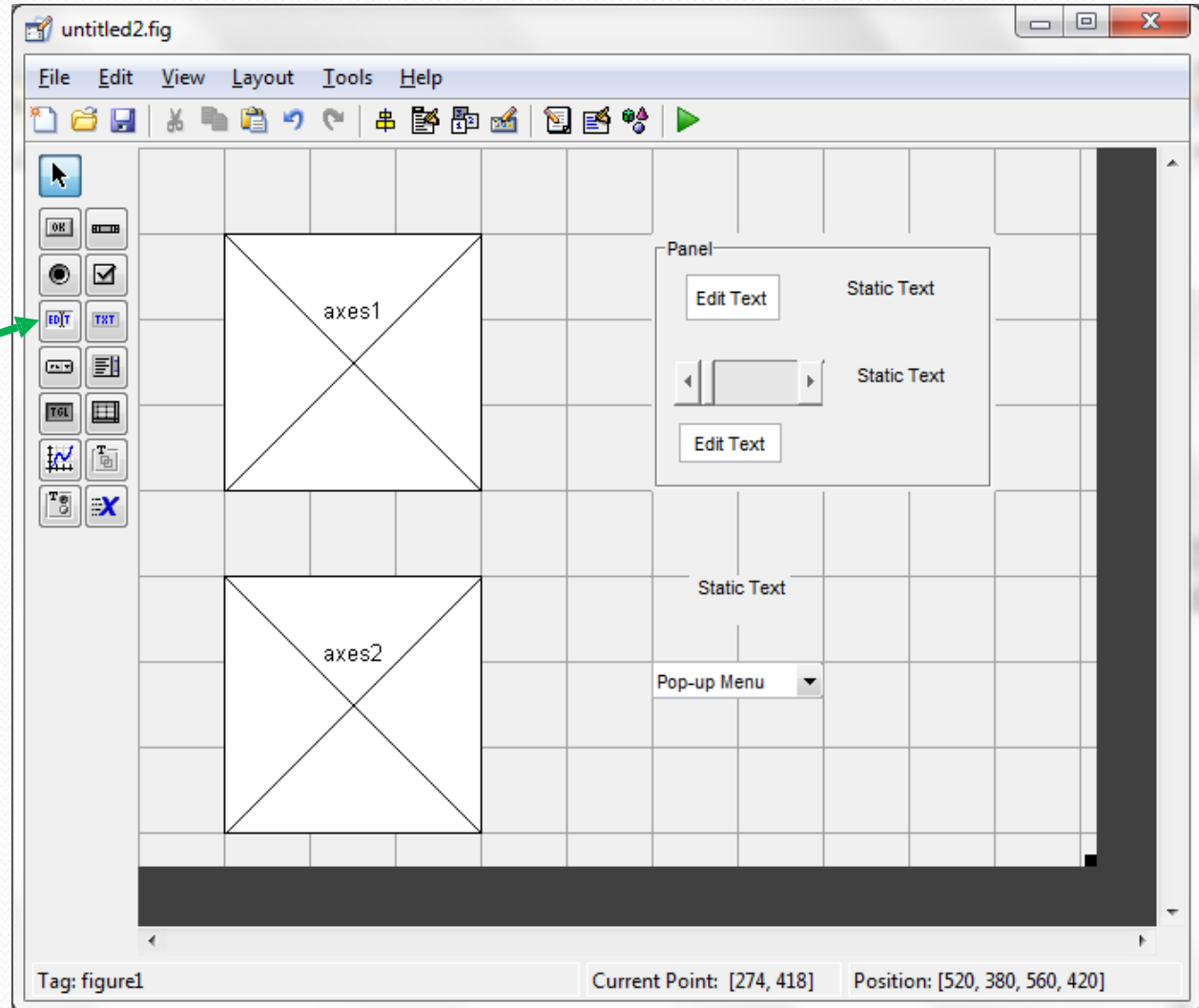
- At the command prompt in MATLAB, type  
`>> guide`

- Select the option to  
Create New GUI,  
Blank GUI (Default)



# Add Objects

- Drag and drop components from the menu of icons to create this GUI:





# MATLAB Function

- Use MATLAB Functions to Create Apps Programmatically You can also code the layout and behavior of your app entirely using MATLAB functions.
- In this approach, you create a traditional figure and place interactive components in that figure programmatically. These apps support the same types of graphics and interactive components that GUIDE supports, as well as tabbed panels.

EDITOR PUBLISH VIEW

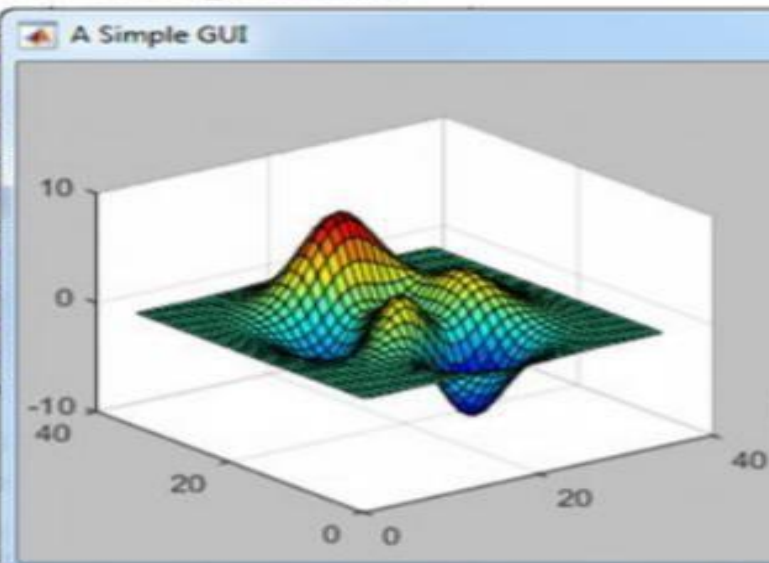
New Open Save Source Control Find Files Compare Print Go To Find Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE BREAKPOINTS RUN

simple\_gui2.m x +

```
1 function simple_gui2
2 % SIMPLE_GUI2 Select a data set from the pop-up menu, then
3 % click one of the plot-type push buttons. Clicking the button
4 % plots the selected data in the axes.
5
6 % Create and then hide the GUI as it is being constructed.
7 f = figure('Visible','off','Position',[360,500,450,285]);
8
9 % Construct the components.
10 hsurf = uicontrol('Style','pushbutton','String','Surf',...
11 'Position',[315,220,70,25],...
12 'Callback',@surfbutton_Callback);
13 hmesh = uicontrol('Style','pushbutton','String','Mesh',...
14 'Position',[315,255,70,25],...
15 'Callback',@meshbutton_Callback);
16 hcontour = uicontrol('Style','pushbutton','String','Contour',...
17 'Position',[315,290,70,25],...
18 'Callback',@contourbutton_Callback);
19 htext = uicontrol('Style','text','String','Select Data',...
20 'Position',[315,325,70,25],...
21 'Callback',@selectdata_Callback);
22 hpopup = uicontrol('Style','listbox','String',{'peaks'},...
23 'Position',[315,360,70,25],...
24 'Callback',@selectdata_Callback);
25
26 ha = axes('Units','normalized','Position',[315,395,70,25]);
27 align([hsurf,hmesh,hcontour,htext,hpopup,ha]);
28
```

A Simple GUI



Surf

Mesh

Contour

Select Data

peaks

# MATLAB GUI (Graphical User Interface)

- GUI's in MATLAB consists of two files:
  - An “m-file” [\*\*\*\*\*.m]
  - A “fig-file” [\*\*\*\*\*.fig]
- The m-file has all of the code that controls the GUI
- The fig-file has all of the graphical objects, positions, and default values, and links it all together.

# MATLAB GUI

The image shows a MATLAB GUI titled "Bullet Trajectory" with a "Help" menu. The interface is divided into several sections:

- Ballistic Information:** Includes input fields for Grain of Bullet (gr.) (140), Muzzle Velocity (ft/s) (2780), Caliber of Bullet (7), and a pop-up menu for unit (mm).
- Atmospheric Conditions:** Includes input fields for Temperature (F) (32) and Pressure (in Hg) (29.5).
- Shooting Conditions:** Includes input fields for Heading (N), Wind Direction (NNW), Wind Speed (MPH) (12), and Distance to Target (Yds.) (100).
- Scope MOA:** Includes radio buttons for 1/4 MOA (selected) and 1/8 MOA.
- Zero:** Includes an input field for Zero (Yds.) (100).
- Buttons:** A "Run Trajectory Calculations" push button and a "Clear Graph" button.
- 3D Plot:** A 3D plot showing Drop (ft.) on the vertical axis, Drift (ft.) on the horizontal axis, and Downrange (ft.) on the depth axis. The axes range from -1 to 1.
- Final Statistics:** A table showing results for Time to Reach Target, Bullet Drop, Bullet Drift, and Kinetic Energy.
- Scope Adjustments:** A table showing MOA Clicks for Elevation and Windage.

Annotations on the left side of the GUI identify various UI elements:

- Edit Text Boxes:** Points to the input fields for Grain of Bullet, Muzzle Velocity, Caliber of Bullet, Temperature, Pressure, Heading, Wind Direction, Wind Speed, and Distance to Target.
- Panel:** Points to the overall GUI window.
- Axes:** Points to the 3D plot axes.
- Pop Up-menu:** Points to the unit selection dropdown for Caliber of Bullet.
- Radio Button:** Points to the 1/4 MOA radio button.
- Push Button:** Points to the "Run Trajectory Calculations" button.
- Static Text:** Points to the "Run Trajectory Calculations" button label.

Final Statistics			
<b>Time to Reach Target</b>	Time	(s)	
<b>Bullet Drop</b>	Drop	(ft.)	
<b>Bullet Drift</b>	Drift	(ft.)	
<b>Kinetic Energy</b>	Energy	(ft lb)	

Scope Adjustments	
MOA Clicks	
<b>Elevation</b>	Clicks E
<b>Windage</b>	Clicks W

# Handle Graphics

- Handle graphics are low-level graphic functions that control characteristics generated by MATLAB.
- They allow the programmer to have precise control of the appearance of plots and graphics.
- Each component has a list of properties that define what it looks like and how it behaves.
- Plot Something:
- Figure Window → View → Property inspector  
→ More Properties

# Handle Graphics

- The 'set' and 'get' commands
- These are the primary commands that you use to ... set and get information about graphic objects, they update the graphical object immediately

- Syntax: 'set'

```
>> set(object_hdl, 'PropertyName', propvalue);
```

- Syntax: 'get'

```
>> propvalue = get(object_hdl, 'PropertyName');
```

# Creating a GUI

➤ **Step 1:** Create the graphical components.

- Using the GUIDE

- Manually configure each component

➤ **Step 2:** Program Components.

-The GUIDE will generate the primary file, you must add to this file all of the actions your components will take.

➤ **Step 3:** Interface with your analysis tools.

# The components and their names are visible

- To add a component to the workspace, you select it and drop it there. The size of the components can be changed by dragging the edges. Changing the position can be done by double-clicking and dragging it.

Let's look at each of the components before using them.

1. Push Button – Its function is to call the callback function for the execution of different programs.



# Con.

2. Slider - It is similar to a scroll bar with a numerical value. The callback function is called and executed when the user changes the pad position. The pad position can be changed by dragging the pad to the required position or clicking the forward or backward arrow.
3. Radio Button – It has on and off states. Its state can be changed by clicking on it. When the button contains a solid circle within the hollow circle, then it is on. But when the hollow circle is empty, it is off. They are grouped and only one can be on at a time.

# Con.

4. Check Box – It has the on and off state. The on the state is represented by a tick inside the box. They are grouped and several checkboxes can be on at the same time.
5. Edit Text – Is for getting strings as an input from the user. The input can then be modified to numbers using `str2num` or the `str2double` function.
6. Static Text – Is to add labels that remain unchanged on the GUI and have no callbacks.

# Con.

7. Pop-up Menu – It is for listing selections. You click on it to view the selections and the callback function is executed depending on the selection.
8. List box – It shows a list of selections just like the pop-up menu. In the list box, all the selections are visible throughout, unlike the pop-up menu where only the current selection is visible.
9. Toggle Button – It functions similarly to the radio button but different appearance in the GUI. It's the appearance that makes the difference.

# Con.

10. Table – It is used for the addition of a spreadsheet to the GUI. In case of a modification in the input, the callback is executed.
11. Axes – Are for the addition of images, charts, and plots to the GUI. They have no callback function.
12. Panel – It is used to group several components and names according to functionality. It doesn't have a callback function.
13. Button Group – It is just like the panel but used to group radio buttons. When we group the radio buttons here. Similar to a panel, we first add the button group before adding the radio buttons.

# Activate Your GUI

- Click on the Green Play arrow to activate the GUI.
- Save the GUI under some name (FirstGui).
- ❖ **Note: name must follow the same rules as naming variables in MATLAB!**
- An m-file will automatically open (with the same name)
- The GUI which currently does almost nothing at this point will launch.
- The m-file has ***callback*** functions where code can be added to control what the various objects (pushbuttons, drop-down menus, plots, etc) will do. Any executable MATLAB code can be placed within these functions.