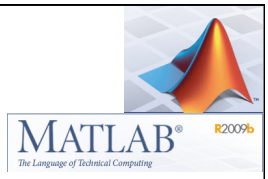
CSCI 130
Introduction to Engineering
Computing
Class Meeting #27



The Last Class Meeting of the Semester!!

- Creating GUI's in Matlab
- Review of Course Learning Goals

11

Creating GUI's in Matlab

GUI = graphical user interface (similar to userforms in VBA)

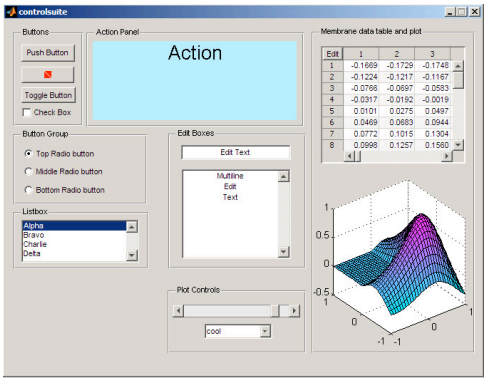
Typical applications:

- **select sources of data**
- **carry out engineering/scientific calculations**
- **make option choices**
- **set values**
- **display results, including graphs directly on GUI**

22

Creating GUI's in Matlab

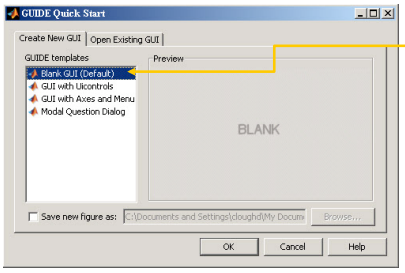
A prototype GUI with many elements



Creating GUI's in Matlab

Matlab provides a tool to aid in the design of GUI's ⇒ GUIDE

Launch GUIDE from the Command window ⇒ `>> guide`

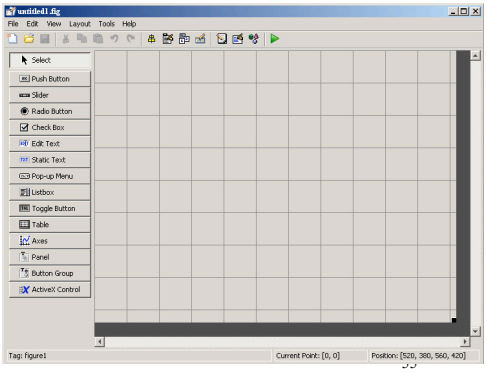


Pick a template for your GUI

44

Creating GUI's in Matlab

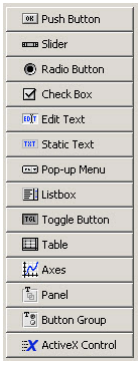
Initial blank template



55

Creating GUI's in Matlab

Tools you can place on the GUI:



You must design and select the tools you require

Placing tools on the GUI is easy, just drag and drop with the mouse

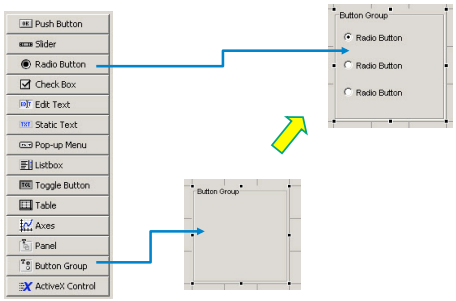
The more difficult step is associating program code with the tools

GUIDE helps with that by creating a skeleton code the first time you save the GUI figure

66

Creating GUI's in Matlab

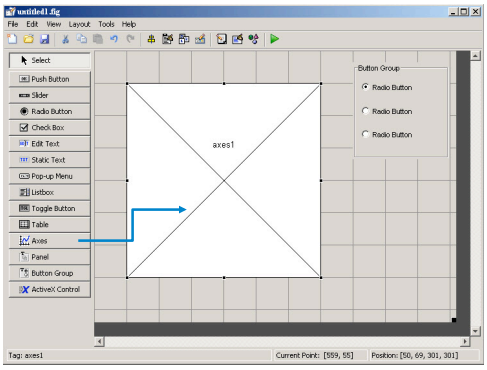
For example, place 3 option buttons (also called radio buttons) in a button group (like a VBA frame)



77

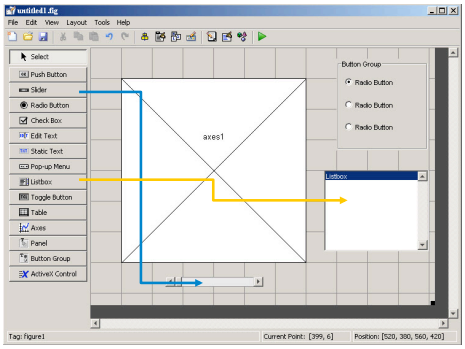
Creating GUI's in Matlab

Add a set of axes for displaying a graph



Creating GUI's in Matlab

Add a slider for setting the value of a numerical parameter and a listbox for making a choice from a list

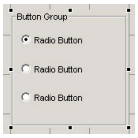


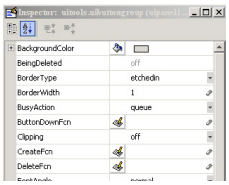
Creating GUI's in Matlab

Once you have placed your selected tools on the GUI, you adjust their appearance using the Property Inspector (like the Properties window in VBA)

activate the Property Inspector from the View menu

select an element on the GUI





Inspector shows properties for that element

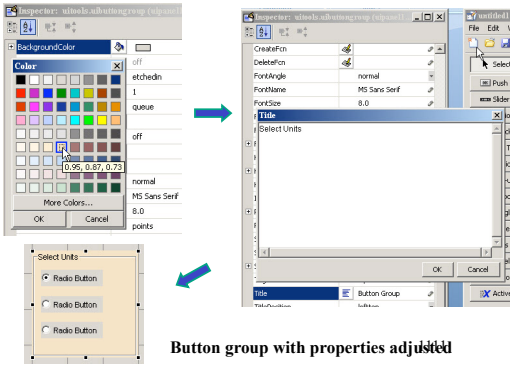
1010

Creating GUI's in Matlab

Change the desired properties of the element

Background color

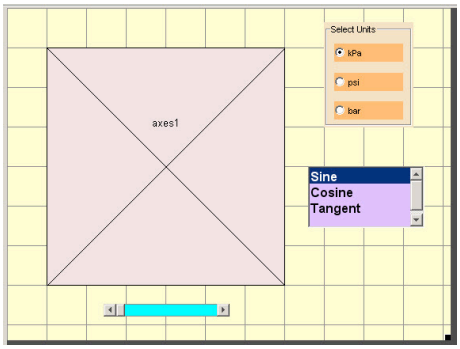
Title



Button group with properties adjusted

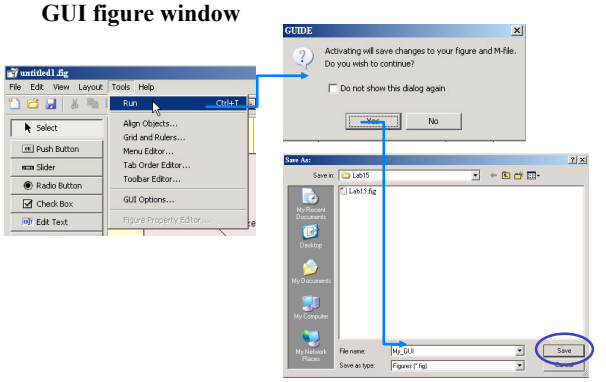
Creating GUI's in Matlab

Complete the detailed work to adjust the properties of all elements on the GUI



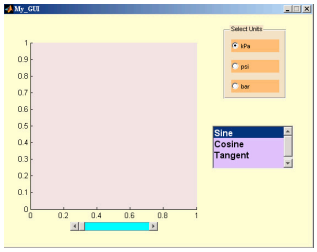
Creating GUI's in Matlab

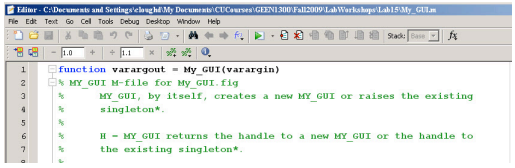
Choose Run from the Tools menu in the GUI figure window



Creating GUI's in Matlab

When you first Run the GUI, it appears and an initial set of functions are written in the Matlab editor

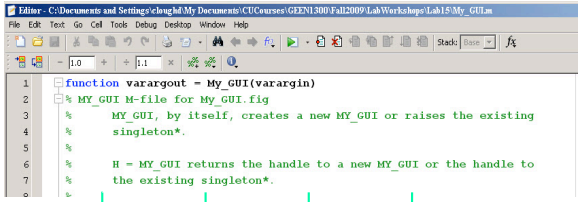




```
function varargout = My_GUI(varargin)
% MY_GUI M-file for My_GUI.fig
%      MY_GUI, by itself, creates a new MY_GUI or raises the existing
%      singleton*.
%
%      H = MY_GUI returns the handle to a new MY_GUI or the handle to
%      the existing singleton*.
```

Creating GUI's in Matlab

You now have to add the code to make the GUI do what you want



```
function varargout = My_GUI(varargin)
% MY_GUI M-file for My_GUI.fig
%      MY_GUI, by itself, creates a new MY_GUI or raises the existing
%      singleton*.
%
%      H = MY_GUI returns the handle to a new MY_GUI or the handle to
%      the existing singleton*.
```

Many of the code sections are "callbacks" – these are like event handlers in VBA where the code executes in response to an action or event, such as moving a slider or selecting an item in a listbox.

1515

Creating GUI's in Matlab

Skeleton callbacks for the slider and listbox

```
% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (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,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

Code goes in here that responds to action

```
% --- 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
```

1616

Creating GUI's in Matlab

The main issue then is to learn how to create the code necessary to the operation of the GUI

Apart from studying background information from Matlab Help, look at examples that relate to what you are trying to accomplish – here is an example:

GUI to Interactively Explore Data in a Table

On this page…

About the tablestat Example

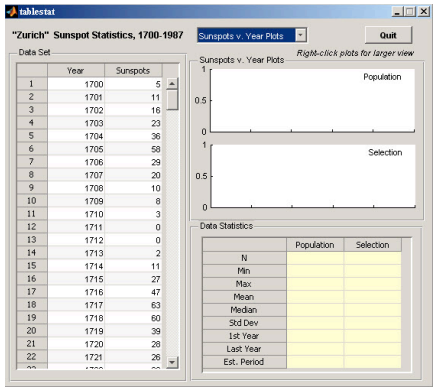
This example shows how to program callbacks for interactive data exploration, including:

- An Opening Function to initialize a table and a plot.
- A uitable's Cell Selection Callback to do plot selected data in real time as the user selects data observations.
- A Pop-up menu's callback to generate line graphs that display different views of data.
- A context menu attached to an axes.

1717

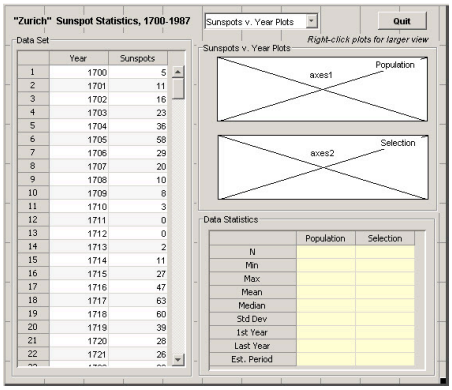
Creating GUI's in Matlab

This GUI allows for statistical exploration of sunspot data



Creating GUI's in Matlab

This is what it looks like in the GUIDE editor



Creating GUI's in Matlab

```
% --- Executes when selected cell(s) is changed in data_table.
function data_table_CellSelectionCallback(hObject, eventdata, handles)
% hObject    handle to data_table (see GCBO)
% eventdata  structure with the following field (see UITABLE)
%  Indices:  row and column indices of the cell(s) currently selected
% handles    structure with handles and user data (see GUIDATA)

% ---- Customized as follows ----
% Obtain 1st column of current selection indices from event data;
% this contains row indices. We don't need column indices.
selection = eventdata.Indices(:,1);
% Remove duplicate row IDs
selection = unique(selection);
% Don't process less than a minimum nuber of observations
if size(selection) < 11
    return
end
% Obtain the data table
table = get(hObject,'Data');
% The selection is the table for stats and plot purposes;
% always extract both columns
% Cache the selection in case the plot type changes
handles.currSelection = selection;
guidata(hObject,handles)
% Update the stats and plot for new selection
refreshDisplays(table(selection,:), handles, 2)
```

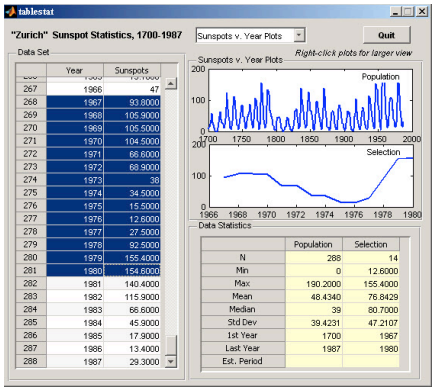
Example callback from code for change in data table selection

2020

Find code that is similar to your needs and adapt it.

Creating GUI's in Matlab

Sunspot GUI in action



Creating GUI's in Matlab

There are many other examples available -- here a just a few in Matlab Help

Examples of GUIDE GUIs

- GUI with Multiple Axes
- GUI for Animating a 3-D View
- GUI to Interactively Explore Data in a Table
- List Box Directory Reader
- Accessing Workspace Variables from a List Box
- A GUI to Set Simulink Model Parameters
- An Address Book Reader
- Using a Modal Dialog Box to Confirm an Operation

2222

Review of Course Learning Goals

1. Problem Solving
- Learn to apply the "engineering method" to the solution of quantitative problems
- Develop the ability to evaluate engineering formulas, carrying units and appropriate precision through calculations

2. Spreadsheet Techniques
- Develop efficient spreadsheet skills
- Learn to set up and interpret "what-if" and case study scenarios
- Learn to organize and layout spreadsheet solutions to engineering problems

2323

Review of Course Learning Goals

3. Programming Fundamentals
- Learn how information is represented by different data types
- Learn program-flow algorithm structure and modularity
- Learn to use features of object-oriented programming

4. Elementary Numerical and Statistical Methods
- Develop the ability to solve single nonlinear algebraic equations using elementary numerical methods, such as bisection, false position or Newton's method
- Learn to solve sets of linear and nonlinear algebraic equations
- Learn to carry out regression calculations

2424

Review of Course Learning Goals

5. Software Tools

- Develop skills with and knowledge of the following software tools:
  - Excel 2007 & Visual Basic for Applications (VBA)
  - Matlab R2009b

2525

Course Summary and Review of Objectives

course title: **Introduction to Engineering Computing**

purpose: get you started
open doors
gain some experience
build reference materials

don't be discouraged that you haven't achieved mastery

challenge: keeping your knowledge and skills alive

look for opportunities to use the computing tools introduced in this course → not just when required by instructors

don't sit there using your calculator (for hours) when, with a little effort, you could get the job done on the computer (and, in *minutes*)

come back for help in future semesters (and later!)
my door is always open

be prepared when computing tools will be required

A couple key final points:

You should judge how much you have learned in this course. That learning is a shared responsibility between you and your instructors. If you learned a lot, as a team, we succeeded. If you learned very little, we failed.

Although you can judge how much you learned, it will be difficult for you to know whether what you learned is on target. But you will be able to assess that with time.

Good luck on the final exam!

[and, of course, ***Go Buffs!!***]

2727