

Matlab GUIs

Brief Description for EIT_fem

2014. 4. 4(Fri)
Taeuk Jeong

Dept. of Computational Science & Engineering
Yonsei University

What Is a GUI?

❖ GUI : Graphical User Interface

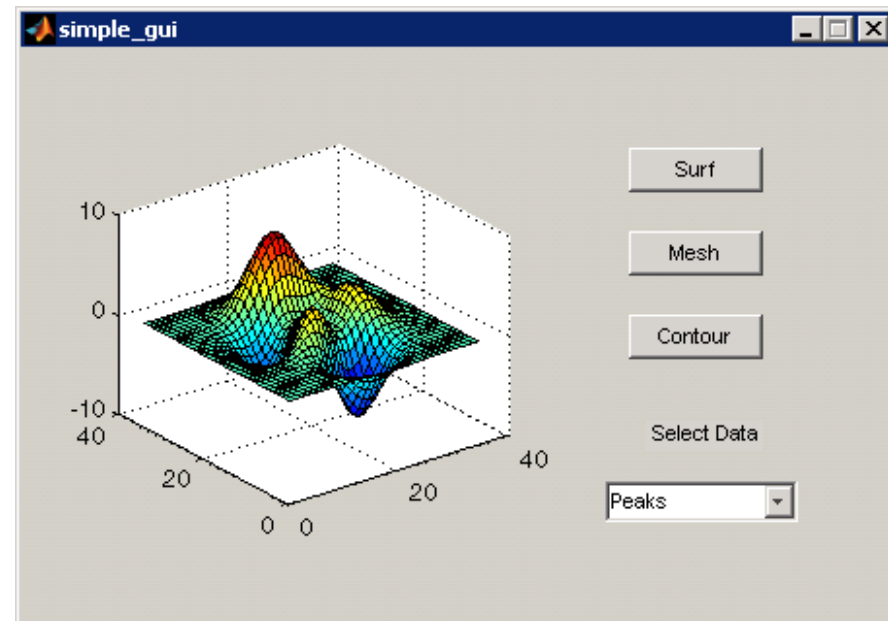
- ✓ a graphical display in one or more windows containing controls, called *components*, that enable a user to perform interactive tasks.

❖ GUI components can include

- ✓ menus, toolbars,
- ✓ push buttons, radio buttons, list boxes, and sliders

❖ GUIs can also perform

- ✓ read and write data files,
- ✓ communicate with other GUIs,
- ✓ display data as tables or as plots



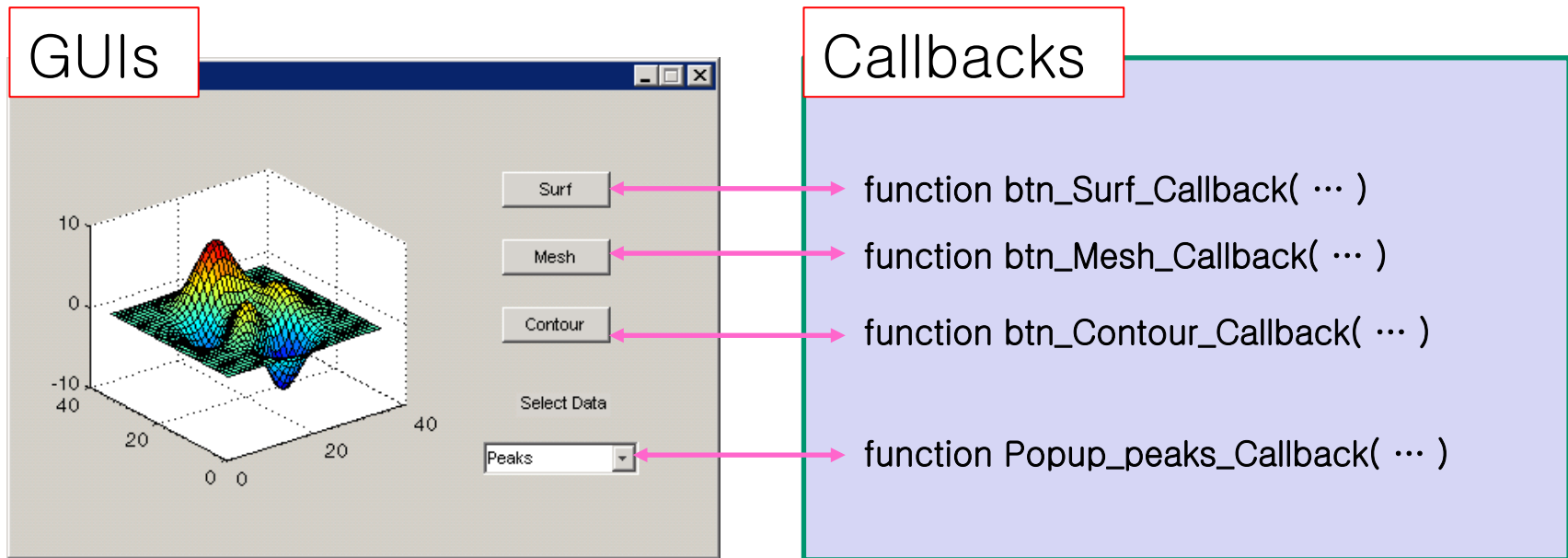
How Does a GUI Work?

❖ GUIs wait

- ✓ for an end user to manipulate a control,
- ✓ and then respond to each user action in turn.

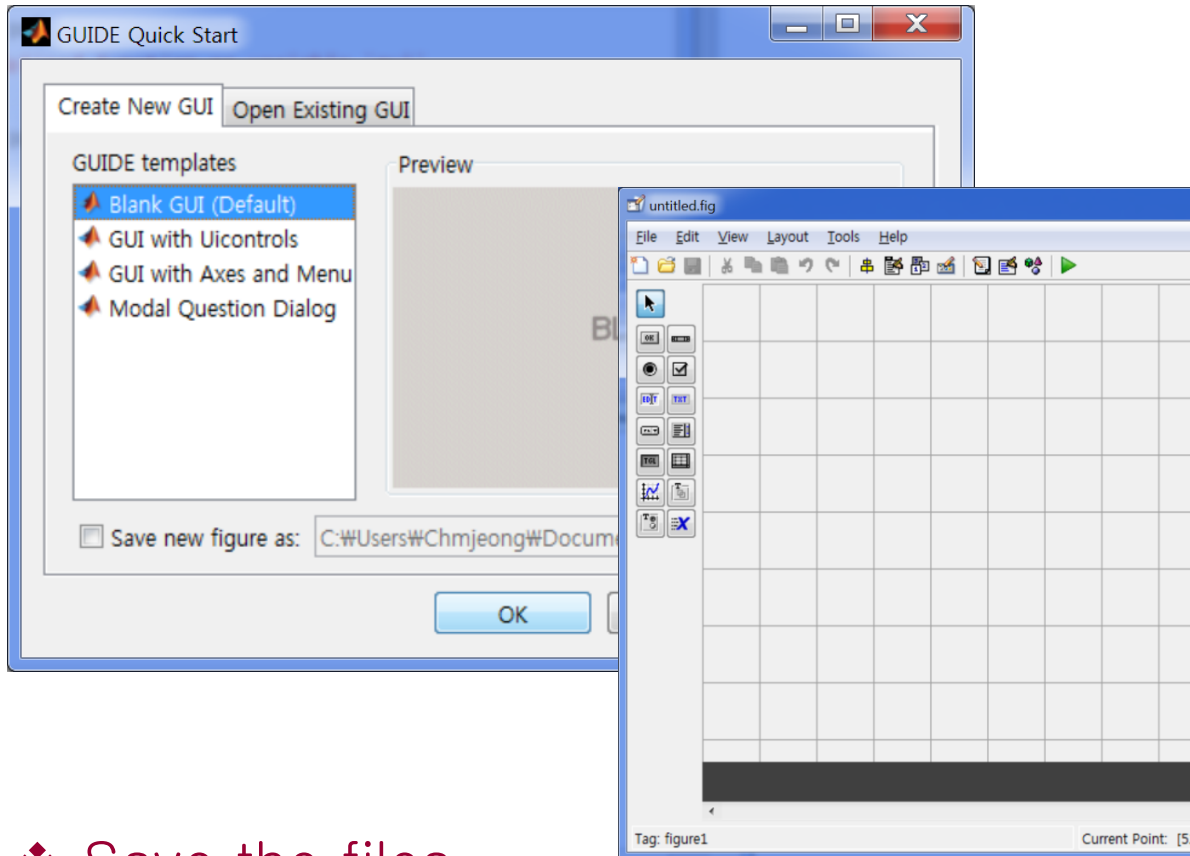
❖ In event-driven programming,

- ✓ callback execution is asynchronous, that is, events(user interactions) external to the software trigger callback execution.



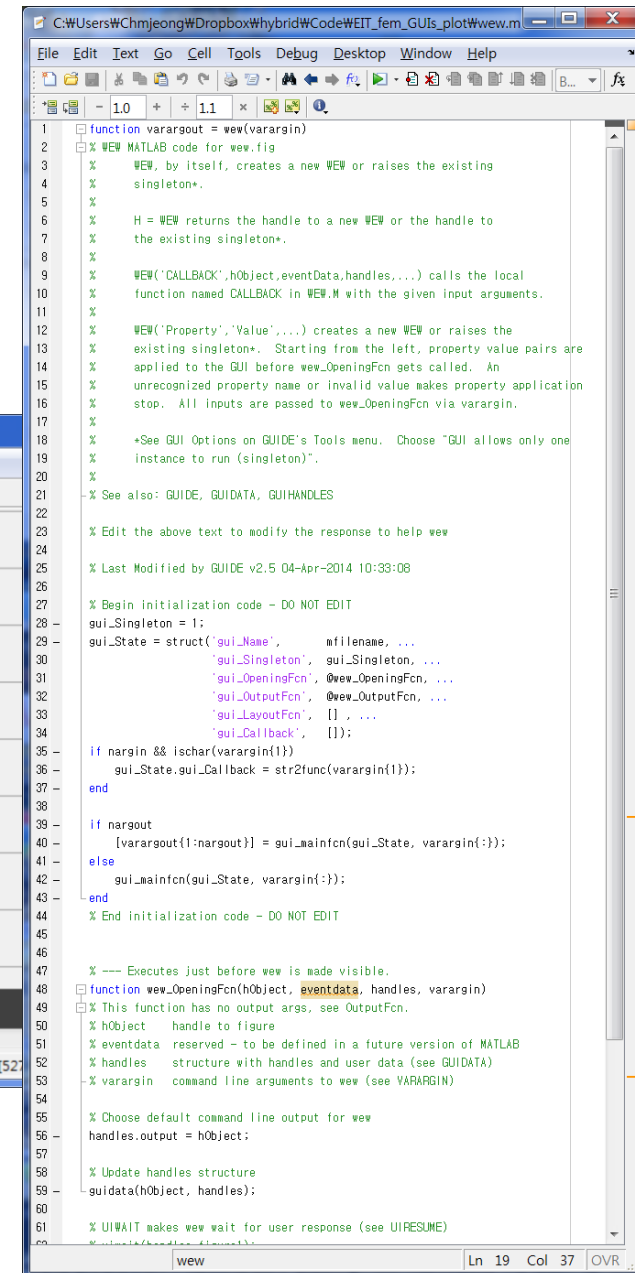
Create Blank GUI

❖ File>New>GUI

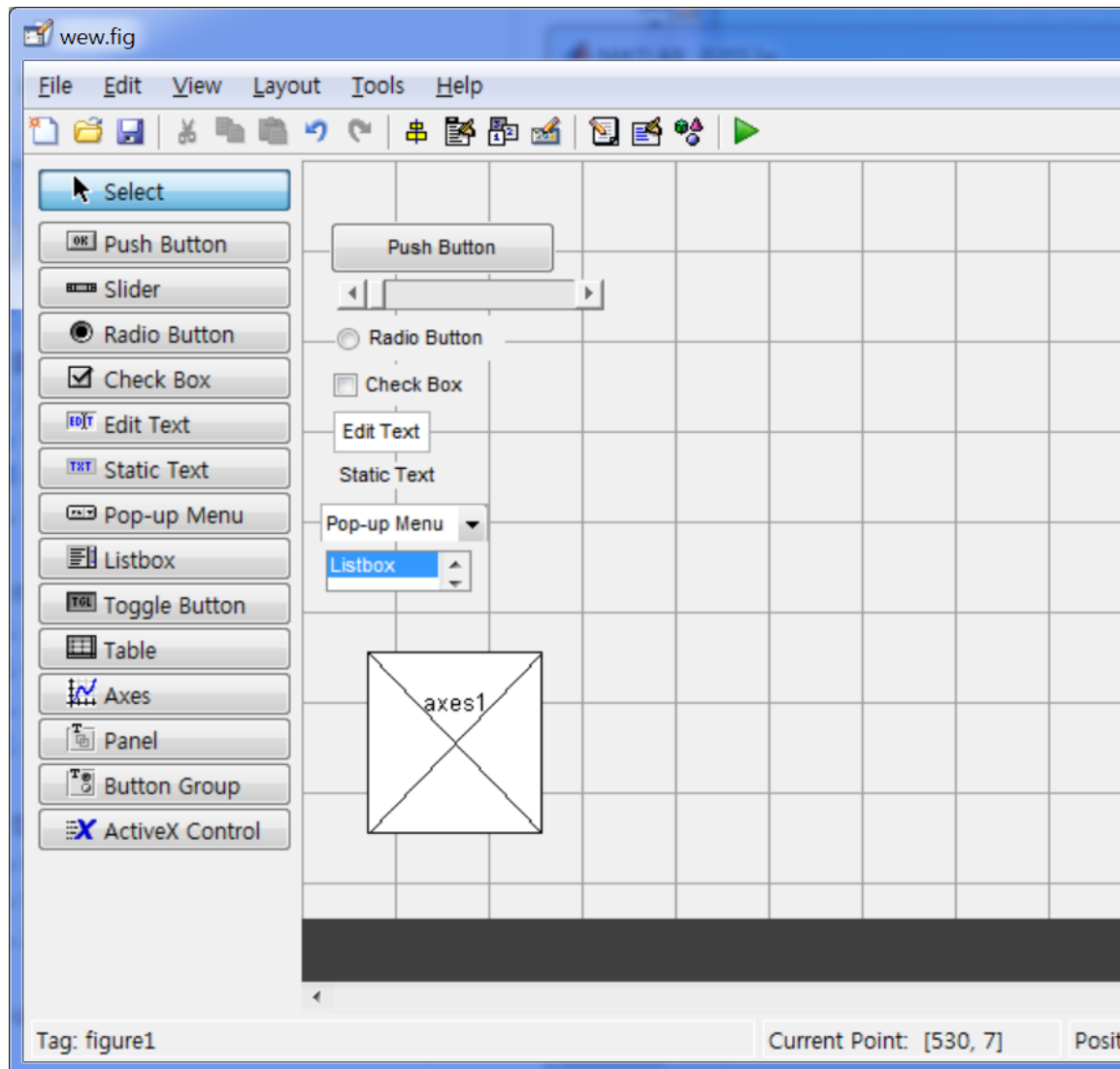


❖ Save the files

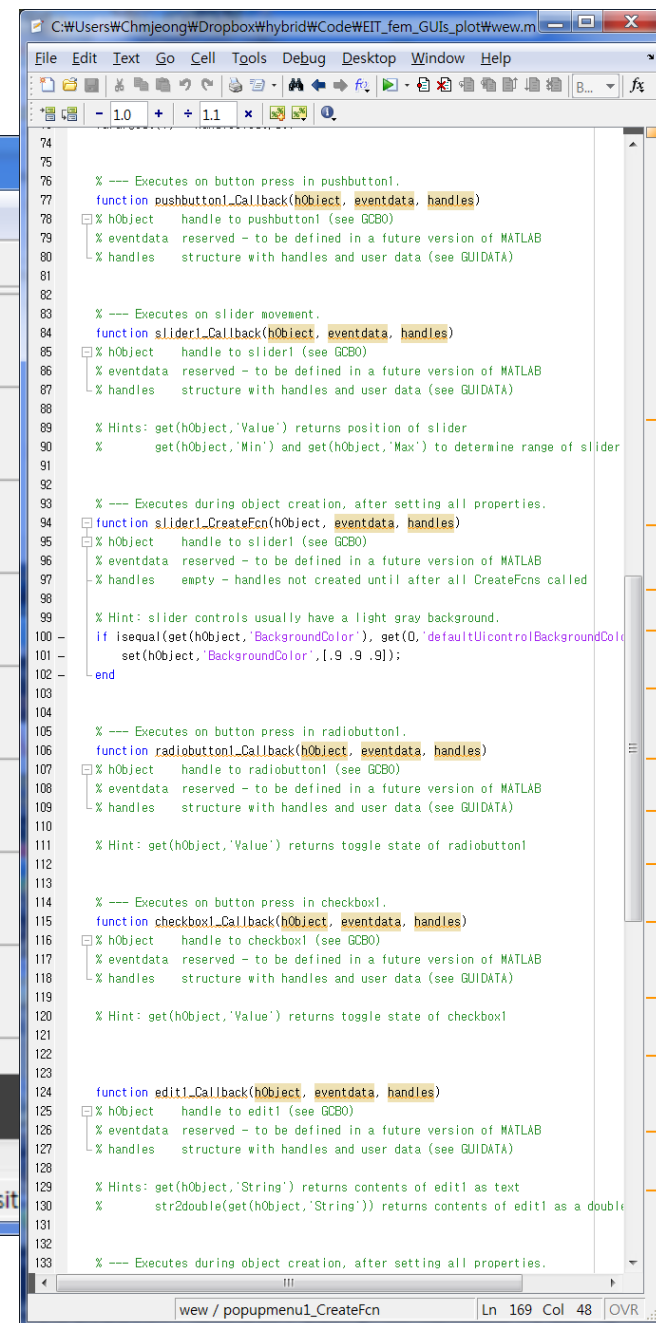
- ✓ xxx.fig → figure file
- ✓ xxx.m → callback file(control code)



Components of GUIs



❖ File>Preferences : Click “Show names in component palette”



Handles structure

- ❖ All functions in the M-file have the following input arguments corresponding to the handles structure:
 - ✓ hObject -- the handle to the figure or Callback object
 - ✓ eventdata -- input from keyboard or mouse click to do something
 - ✓ handles -- structure with handles and user data

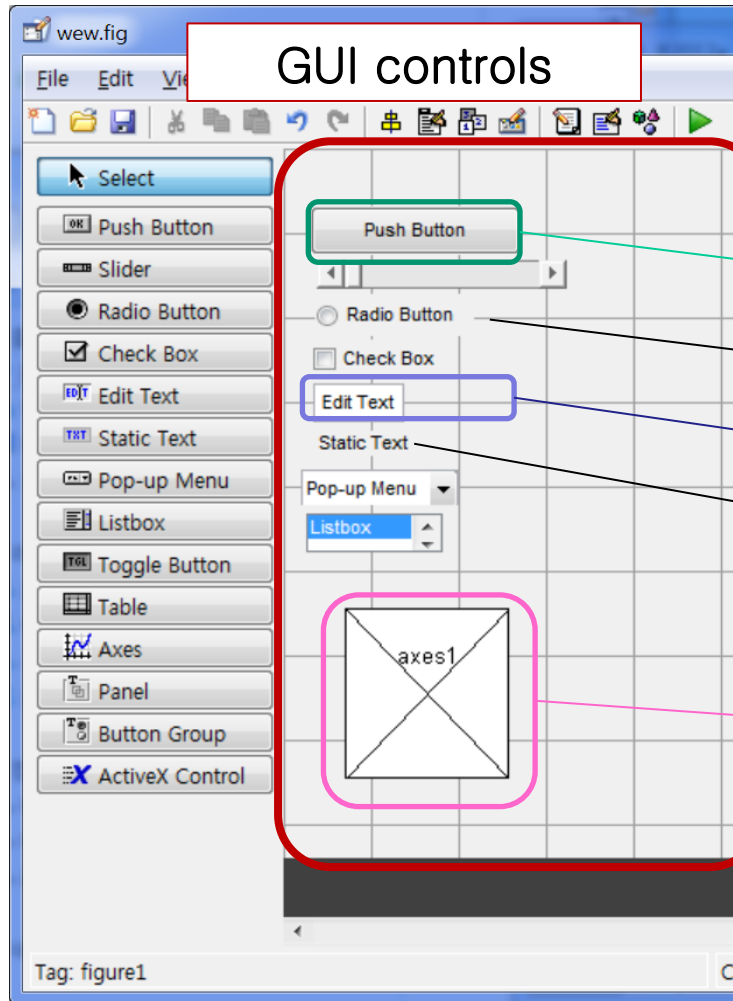
- ✓ `function test_OpeningFcn(hObject, eventdata, handles, varargin)`

- ❖ Update of the handles structure
 - ✓ `guidata(hObject, handles);`

Handles structure

❖ handle : identification of objects in GUI

✓ Assign to a real value



handles structure

handles.figure1

handles.pushbutton1

handles.radiobutton1

handles.edit1

handles.text1

handles.axes1

Inspector : figure

The image shows the MATLAB GUI Inspector for a figure window titled "figure (EIT_fem_plot)". The interface is divided into three main sections:

- GUI controls:** A list of controls on the left, including Select, Push Button, Slider, Radio Button, Check Box, Edit Text, Static Text, Pop-up Menu, Listbox, Toggle Button, Table, Axes, Panel, Button Group, and ActiveX Control. A red box highlights the "GUI controls" label.
- hanles.figure1:** A red box highlights the "axes1" control in the main workspace, which is a square plot area with an "X" inside. A red arrow points from this box to the "Tag" property in the Inspector.
- Inspector:** A list of properties for the figure window on the right. A red box highlights the "Tag" property, which is set to "figure1". A red arrow points from the "hanles.figure1" box to this property.

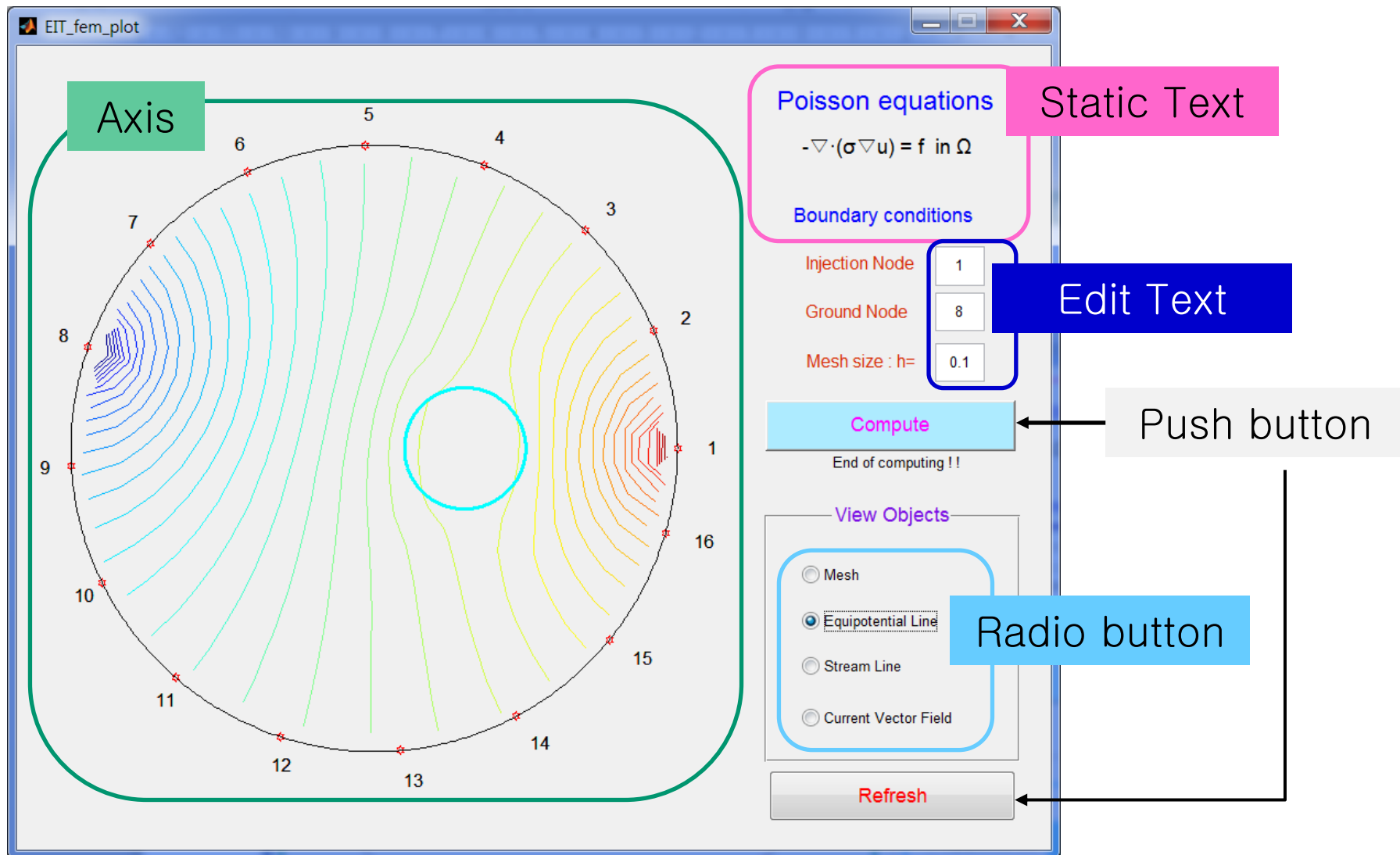
The Inspector properties include:

- BeingDeleted: off
- BusyAction: queue
- ButtonDownFcn: on
- Clipping: on
- CloseRequestFcn: closereq
- Color: [0 0 0]
- CreateFcn: on
- CurrentCharacter: 0
- CurrentPoint: [-0.143 -0.05]
- DeleteFcn: on
- DockControls: on
- FileName: C:\Users\Chmjeong\WD...
- HandleVisibility: callback
- HitTest: on
- IntegerHandle: off
- Interruptible: on
- InvertHardcopy: on
- KeyPressFcn: on
- KeyReleaseFcn: on
- MenuBar: none
- Name: EIT_fem_plot
- NextPlot: add
- NumberTitle: off
- PaperOrientation: portrait
- PaperPosition: [0.25 2.5 8 6]
- PaperPositionMode: manual
- PaperSize: [20.984 29.677]
- PaperType: A4
- PaperUnits: centimeters
- Pointer: arrow
- PointerShapeCData: [16x16 double array]
- PointerShapeHotSpot: [1x2 double array]
- Position: [103.714 25.6 132.286 35.9]
- Renderer: painters
- RenderMode: auto
- Resize: off
- ResizeFcn: on
- SelectionHighlight: on
- SelectionType: normal
- Tag: figure1
- ToolBar: auto
- UIContextMenu: <None>
- Units: characters
- UserData: [1x0 double array]
- Visible: on
- WVisual: 0
- WVisualMode: auto
- WindowButtonDownFcn: on
- WindowButtonMotionFcn: on
- WindowButtonUpFcn: on
- WindowKeyPressFcn: on
- WindowKeyReleaseFcn: on
- WindowScrollWheelFcn: on
- WindowStyle: normal

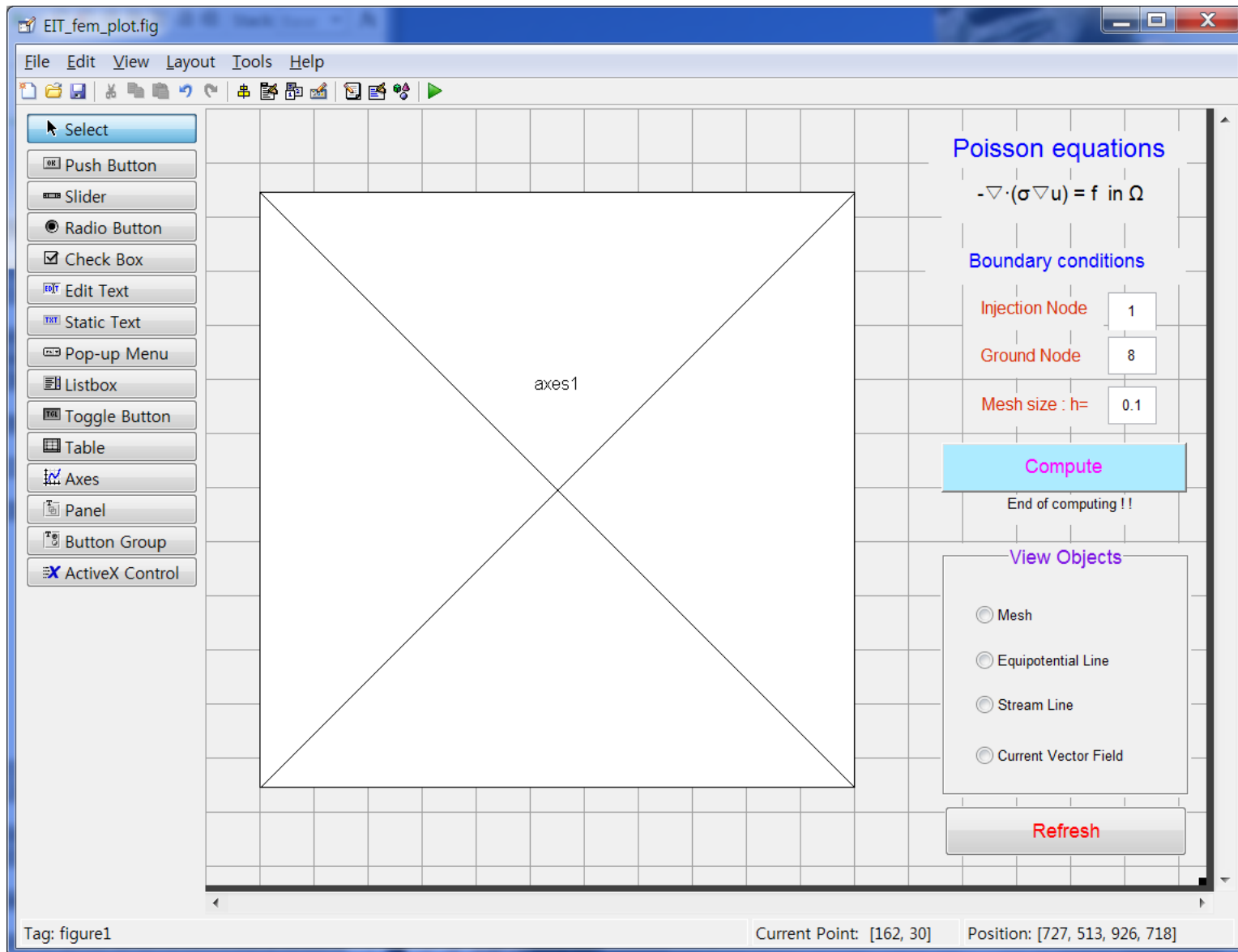
Planning of GUIs



EIT_fem_plot.exe



Create GUI controls



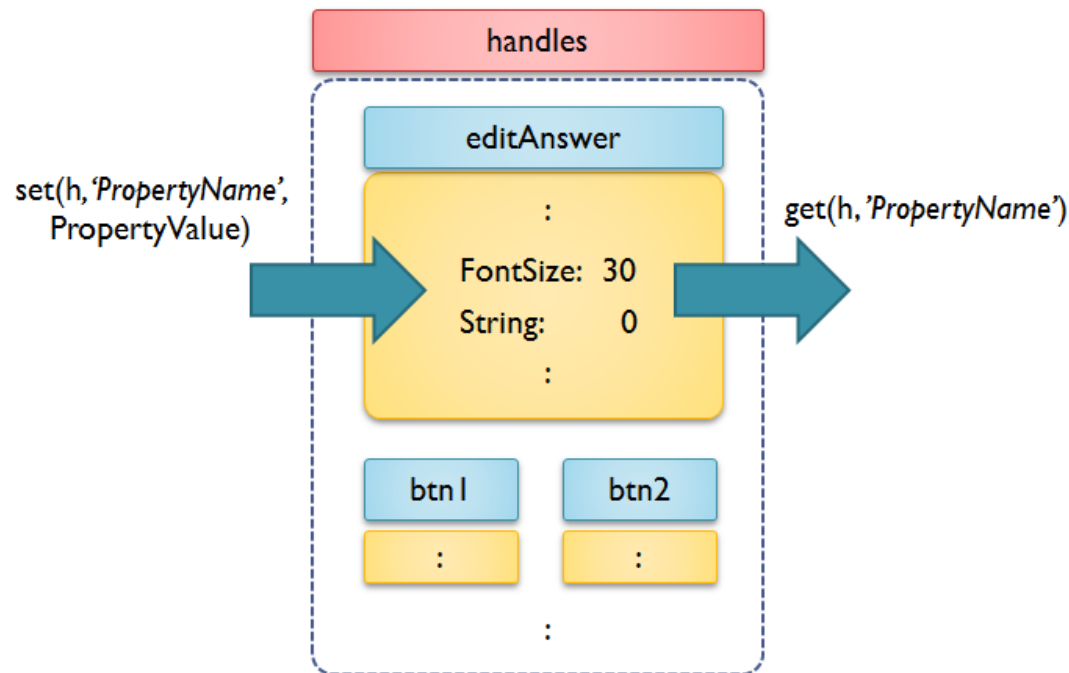
Access to properties of handles

❖ `get(handle, 'PropertyName');`

- ✓ `str = get(handles.edit1, 'String');`
- ✓ `str_val = str2double(get(handles.edit1, 'String'));`

❖ `set(handle, 'PropertyName', 'PropertyValue');`

- ✓ `set(handles.edit1, 'String', '1');`



Static text

Inspector: uicontrol (text1 "Poisson eq...")

- BackgroundColor:
- BeingDeleted: off
- BusyAction: queue
- ButtonDownFcn:
- CData: [0x0 double array]
- Callback:
- Clipping: on
- CreateFcn:
- DeleteFcn:
- Enable: on
- Extent: [0 0 28.714 1.75]
- FontAngle: normal
- FontName: MS Sans Serif
- FontSize: 15.000000000000004
- FontUnits: points
- FontWeight: normal
- ForegroundColor:
- HandleVisibility: on
- HitTest: on
- HorizontalAlignment: center
- Interruptible: on
- KeyPressFcn:
- ListboxTop: 1.0
- Max: 1.0
- Min: 0.0
- Position: [95.371 31.931 34.2 2.846]
- SelectionHighlight: on
- SliderStep: [0.01 0.1]
- String: Poisson equations
- Style: text
- Tag: text1
- TooltipString: <None>
- UIContextMenu: <None>
- Units: characters
- UserData: [0x0 double array]
- Value: [0.0]
- Visible: on

hanles.text1

Poisson equations

Static Text

$-\nabla \cdot (\sigma \nabla u) = f$ in Ω

Boundary conditions

Injection Node: 1

Ground Node: 8

Mesh size : h= 0.1

Compute

End of computing !!

View Objects

☐ Mesh

☐ Equipotential Line

☐ Stream Line

☐ Current Vector Field

```
str = get(handles.text1, 'String');
set(handles.text1, 'String', 'EIT_fem');
```

Current Point: [8/8, 7/15] Position: [727, 513, 926, 718]

Edit text : initialization

The image illustrates the initialization of an edit text field in a MATLAB GUI. The Inspector window on the left shows the 'handles.edit1' object with properties like String, Style, and Tag. The main window shows a GUI with a plot of Poisson equations and boundary conditions. The Code window on the right shows the initialization function EIT_fem_plot_OpeningFcn, which sets the handles structure and initializes the edit text field.

handles.edit1

Edit Text

```

1 % --- Executes just before EIT_fem_plot is made visible.
2 function EIT_fem_plot_OpeningFcn(hObject, eventdata, handles, varargin)
3 % This function has no output args, see OutputFcn.
4 % hObject    handle to figure
5 % eventdata  reserved - to be defined in a future version of MATLAB
6 % handles    structure with handles and user data (see GUIDATA)
7 % varargin   command line arguments to EIT_fem_plot (see VARARGIN)
8
9 % Choose default command line output for EIT_fem_plot
10 handles.output = hObject;
11 handles.inB = str2double(get(handles.edit1, 'String'));
12 handles.outB = str2double(get(handles.edit2, 'String'));
13 handles.h_size = str2double(get(handles.edit3, 'String'));
14
15 % Update handles structure
16 guidata(hObject, handles);
  
```

Edit text : update value

handles.edit1

The screenshot displays the MATLAB GUI for 'EIT_fem_plot'. The left pane shows the MATLAB code for the callback function `edit1_Callback` and the create function `edit1_CreateFcn`. A red box highlights the `edit1_Callback` function name. A pink dashed arrow points from the `handles.edit1` property access in the code to the 'Edit Text' button in the GUI. The right pane shows the GUI interface with fields for 'Injection Node' (value 2), 'Ground Node' (value 8), and 'Mesh size : h=' (value 0.1). A blue box labeled 'Edit Text' points to the 'Injection Node' field. The bottom pane shows a plot of a curved boundary with nodes 11, 12, 13, and 14. A context menu is open over the 'Injection Node' field, showing options like 'Cut', 'Copy', 'Paste', 'Clear', 'Duplicate', 'Bring to Front', 'Send to Back', 'Object Browser', 'Editor', 'View Callbacks', 'Property Inspector', 'Edit Text Property Editor...', and 'Current Vector Field'.

```

1 function edit1_Callback(hObject, eventdata, handles)
2 % hObject    handle to edit1 (see GCBO)
3 % eventdata  reserved - to be defined in a future version of MATLAB
4 % handles    structure with handles and user data (see GUIDATA)
5
6 % Hints: get(hObject,'String') returns contents of edit1 as text
7 %        str2double(get(hObject,'String')) returns contents of edit1 as a double
8 handles.inB = str2double(get(handles.edit1,'String'));
9 guidata(hObject, handles);
10 end
11
12 % --- Executes during object creation, after setting all properties.
13 function edit1_CreateFcn(hObject, eventdata, handles)
14 % hObject    handle to edit1 (see GCBO)
15 % eventdata  reserved - to be defined in a future version of MATLAB
16 % handles    empty - handles not created until after all CreateFcns called
17
18 % Hint: edit controls usually have a white background on Windows.
19 %        See ISPC and COMPUTER.
20 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
21     set(hObject,'BackgroundColor','white');
22 end
  
```

issson equations

$$\nabla \cdot (\sigma \nabla u) = f \text{ in } \Omega$$

oundary conditions

Injection Node

Ground Node

Mesh size : h=

Compute

View Objects

Mesh

Equipotent

Stream Lin

Current Ve

Ref

11

12

13

14

Context Menu:

- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Clear
- Duplicate (Ctrl+D)
- Bring to Front (Ctrl+F)
- Send to Back (Ctrl+B)
- Object Browser
- Editor
- View Callbacks
- Property Inspector
- Edit Text Property Editor...
- Current Vector Field

Push button

The image displays the MATLAB GUI development environment with three main windows:

- Inspector: uicontrol (btn_comp "C...")**: Shows the properties of the button. Key properties are highlighted with green boxes:
 - BackgroundColor**: (highlighted)
 - Extent**: [0 0 10.857 1.35]
 - FontSize**: 11.0
 - String**: Compute
 - Style**: pushbutton
 - Tag**: btn_comp
 - Visible**: on
- Editor**: Shows the MATLAB code for the GUI.
 - function EIT_fem_plot_OpeningFcn**: Initializes the handles structure. A red box highlights the initialization of `handles.u`, `handles.dxu`, `handles.dyu`, `handles.Node`, `handles.Ord`, `handles.Inj_B`, and `handles.Bindicat`.
 - function btn_comp_Callback**: Executes on button press. It updates the handles structure and calls `EIT_fem_GUIs`. A red box highlights the update of the handles structure.
- Figure**: Shows the GUI for Poisson equations. It includes a title bar, a text area for the equation $-\nabla \cdot (\sigma \nabla u) = f$ in Ω , a section for Boundary conditions with Injection Node (2) and Ground Node (8), a Mesh size input, a **Compute** button (highlighted with a blue box), and a **View Objects** button. A red arrow points from the `set(handles.text7, 'Visible', 'on');` line in the callback function to the `text7` label in the GUI.

Button Group

The screenshot displays a MATLAB GUI titled 'EIT_fem_plot.fig' with several windows open:

- Main GUI Window:** Contains a toolbar with buttons like 'Select', 'Push Button', 'Slider', 'Radio Button', 'Check Box', 'Edit Text', 'Static Text', 'Pop-up Menu', 'Listbox', 'Toggle Button', 'Table', 'Axes', 'Panel', 'Button Group', and 'ActiveX Control'. The 'Button Group' button is highlighted with a red rectangle.
- Inspector: uitools.uibuttongroup (...):** Shows properties for the selected 'Button Group' control. Properties include BackgroundColor, BeingDeleted, BorderType, BorderWidth, BusyAction, ButtonDownFcn, Clipping, CreateFcn, DeleteFcn, FontAngle, FontName, FontSize, FontUnits, FontWeight, ForegroundColor, HandleVisibility, HighlightColor, HitTest, Interruptible, Position, ResizeFcn, SelectedObject, SelectionHighlight, ShadowColor, Tag, Title, TitlePosition, UIContextMenu, Units, UserData, and Visible. The 'Title' property is set to 'View Objects'.
- Inspector: uicontrol (mesh "Mesh"):** Shows properties for the 'Mesh' control. The 'String' property is set to 'Mesh', and the 'Style' property is set to 'radiobutton'. A red rectangle highlights the 'Mesh' radio button in the 'View Objects' section of the main GUI.
- Main Plot Area:** Displays the text 'Poisson equations' with the equation $-\nabla \cdot (\sigma \nabla u) = f$ in Ω . Below it, 'Boundary conditions' are listed: 'Injection Node' (1), 'Ground Node' (8), and 'Mesh size : h=' (0.1). A 'Compute' button is present, followed by the text 'End of computing !!'. The 'View Objects' section contains four radio buttons: 'Mesh' (selected), 'Equipotential Line', 'Stream Line', and 'Current Vector Field'. A red rectangle highlights this section, and a blue box labeled 'handles.mesh' points to the 'Mesh' radio button.

Button Group & Radio Button

```

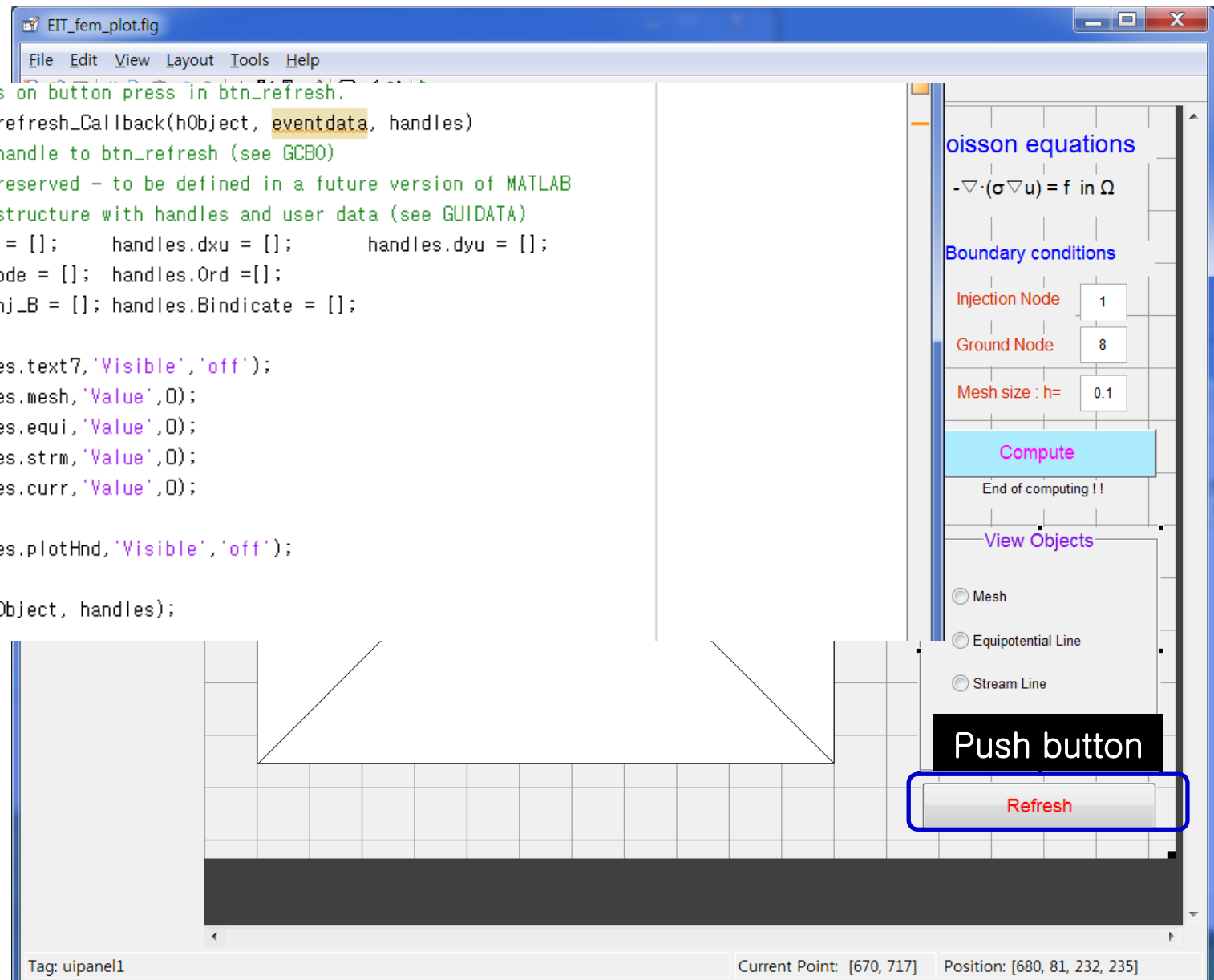
1 % --- Executes when selected object is changed in uipanel1.
2 function uipanel1_SelectionChangeFcn(hObject, eventdata, handles)
3 % hObject    handle to the selected object in uipanel1
4 % eventdata  structure with the following fields (see UIBUTTONGROUP)
5 %   EventName: string 'SelectionChanged' (read only)
6 %   OldValue: handle of the previously selected object or em
7 %   NewValue: handle of the currently selected object
8 % handles    structure with handles and user data (see GUIDA
9 hd = handles;
10 switch hObject
11 case handles.mesh
12     set(handles.plotHnd, 'Visible', 'off');
13     handles.plotHnd = EIT_fem_GUIs_plot_trimesh( hd.h_size, hd.u, hd.dxu, hd.dyu,...
14         hd.Ord, hd.Node, hd.Inj_B, hd.Bindicate);
15 case handles.equi
16     set(handles.plotHnd, 'Visible', 'off');
17     handles.plotHnd = EIT_fem_GUIs_plot_equipotential( hd.h_size, hd.u, hd.dxu, hd.dyu,...
18         hd.Ord, hd.Node, hd.Inj_B, hd.Bindicate);
19 case handles.strm
20     set(handles.plotHnd, 'Visible', 'off');
21     handles.plotHnd = EIT_fem_GUIs_plot_streamLine( hd.h_size, hd.u, hd.dxu, hd.dyu,...
22         hd.Ord, hd.Node, hd.Inj_B, hd.Bindicate);
23 case handles.curr
24     set(handles.plotHnd, 'Visible', 'off');
25     handles.plotHnd = EIT_fem_GUIs_plot_currentWF( hd.h_size, hd.u, hd.dxu, hd.dyu,...
26         hd.Ord, hd.Node, hd.Inj_B, hd.Bindicate);
27 end
28 guidata(hObject, handles);

```

'Value' = 1 if clicked
'Value' = 0 if unclicked

The screenshot shows the EIT GUI interface. At the top, it displays the Poisson equations: $-\nabla \cdot (\sigma \nabla u) = f$ in Ω . Below this, boundary conditions are specified: Injection Node (1), Ground Node (8), and Mesh size: $h = 0.1$. A 'Compute' button is visible, with a message 'End of computing !!' below it. A 'View Objects' panel is open, showing a list of objects: Mesh, Cut, Copy, Paste, Clear, Duplicate, Bring to Front, Send to Back, Object Browser, Editor, View Callbacks, Property Inspector, and Button Group Property Editor... A context menu is also visible, listing actions: CreateFcn, DeleteFcn, ButtonDownFcn, ResizeFcn, and SelectionChangeFcn.

Push button



Summaries

❖ GUIs -- Functions

- ✓ Components(controls) -- callback function

❖ Data access

- ✓ `get(...), set(...)`
- ✓ Handles properties

❖ Planning of working mechanism

- ✓ Plan flow-chart
- ✓ Design GUI controls – push button, check box, edit text, etc

❖ Build the GUIs

- ✓ Matching the controls with functions and handle variables