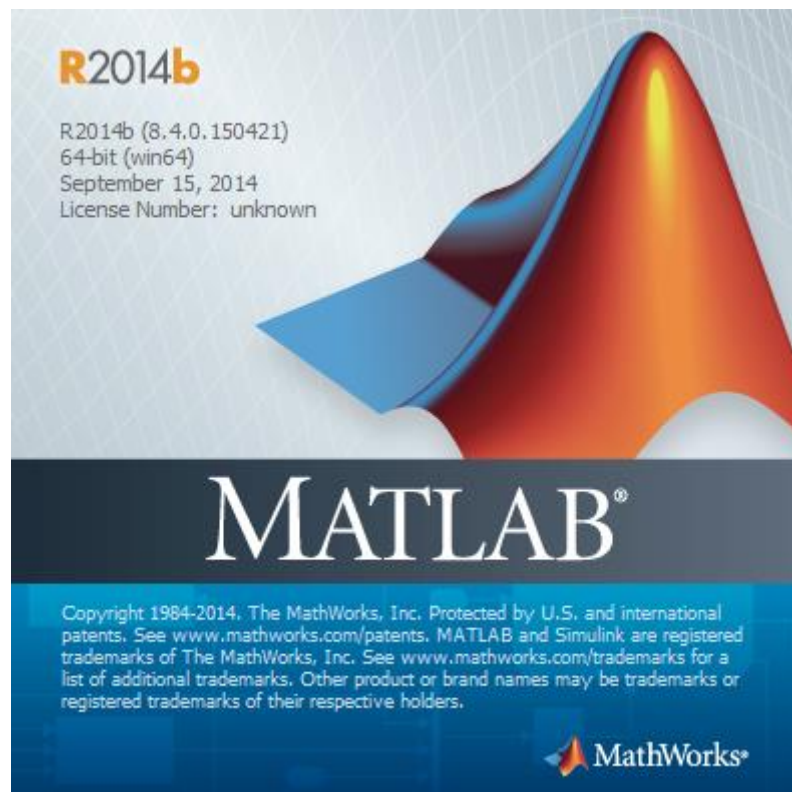


MATLAB软件与应用



第七章 GUI设计

-
- ◆ GUIDE工具简介
 - ◆ GUI 向导设计
 - ◆ GUI 程序设计

什么是GUI

- GUI是Graphic User Interface图形用户接口的简称，允许用户在图形显示环境下，通过鼠标、键盘等输入设备来操纵屏幕上的菜单或图标，完成，选择命令、调用函数、启动程序等常用任务。
- 在MATLAB中，可通过GUIDE向导设计和GUI程序设计两类模式来开发GUI程序。
- demo所打开的演示程序是典型的GUI程序。

GUI工具简介 启动GUIDE

我们主要通过**GUI向导设计（GUIDE）**模式来开发GUI程序

Graphic User Interface Development Environment

- 方便、高效的图形界面集成开发环境
- 一套界面设计工具的集合（实质上是一种工具箱）
- GUI控件都集成在这个环境(外观、属性和响应)
- 设计好的GUI界面保存为FIG文件中
- 自动生成一个包含GUI初始化和布局控制代码的M文件

整个GUI的实现过程可以分为以下几步：

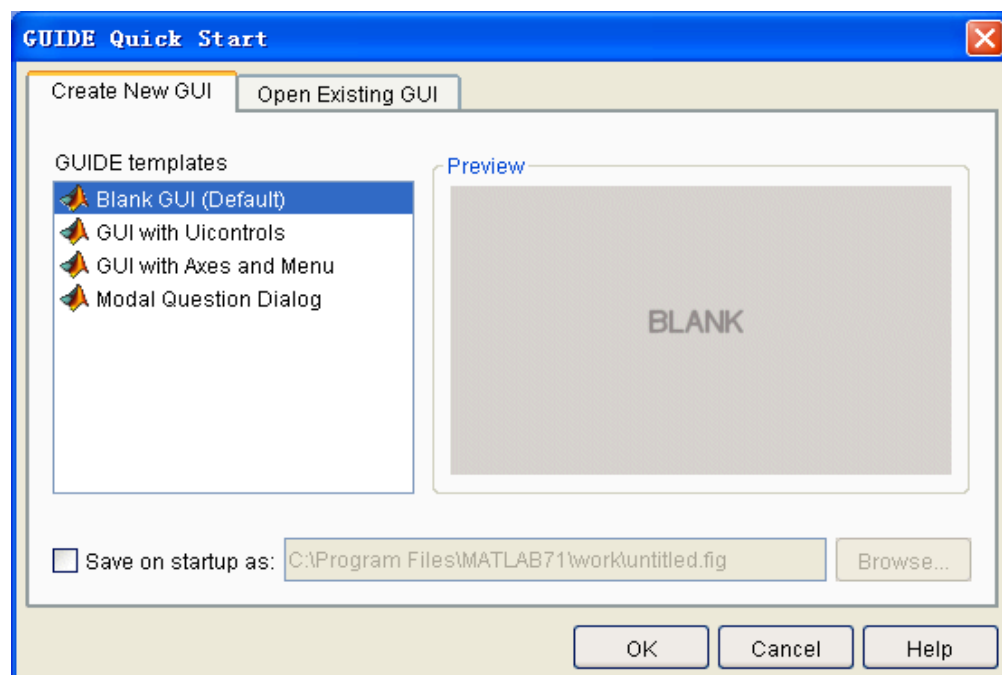
- 使用界面设计编辑器进行GUI界面布局设计；
- 理解应用程序M文件中所使用的编程技术；
- 编写GUI组件行为响应控制（即回调函数）代码。

GUI工具简介 启动GUIDE

GUIDE提供了多种设计模板（以及对应的M文件）以方便用户使用GUI。

启动方式

1. 直接输入`guide`命令，打开GUI模版界面
2. 菜单File->New也可打开GUI模板界面



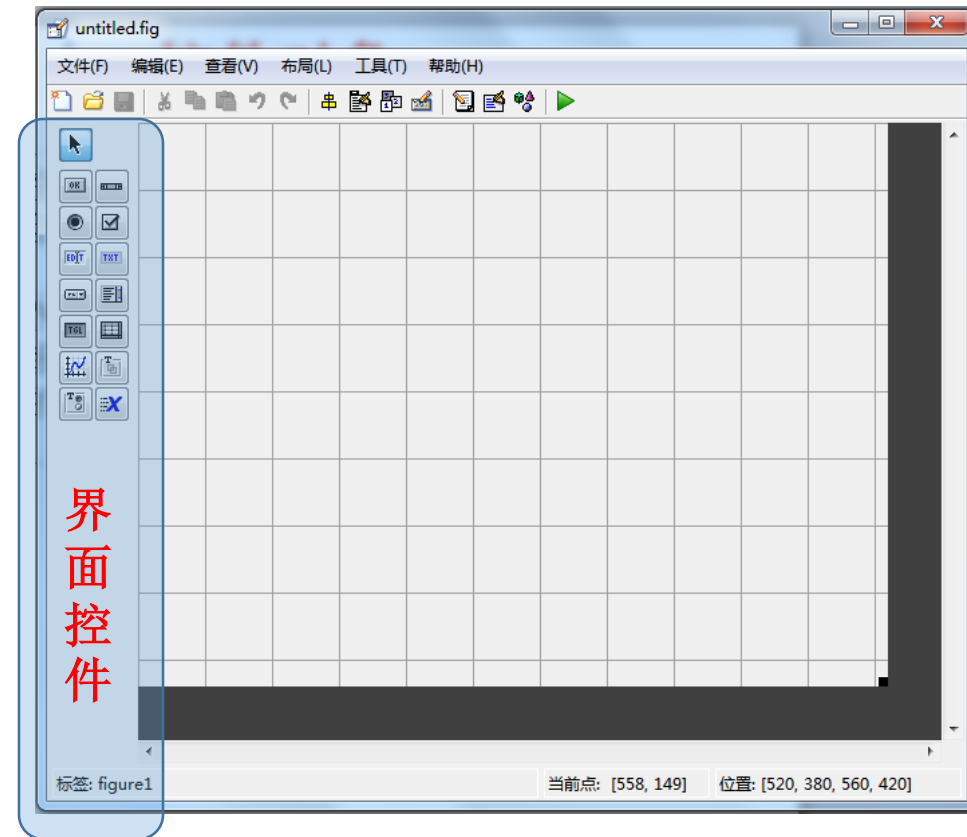
GUI工具简介

GUI设计模版

Matlab提供4种GUI设计种模板

创建GUI的同时，会生成.fig和.M文件，前者保存GUI外观的设计信息，后者吧保存控件所执行的功能即回调函数。














1. 空白模板
2. 带有控制按钮的模板
3. 带有坐标轴和菜单的模板
4. 问答式对话框



GUI空白界面编辑器

GUI工具简介 控件对象

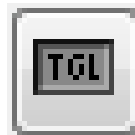
在GUI界面编辑器中可使用控件来设计GUI的外观。

		
按下按钮立即触发	 	滑动条，通过滑动改变值
单选按钮	 	复选框
编辑框（文本和图形）	 	静态文本，只显示不接受输入
弹出式菜单	 	列表框（可接受多个值）
开关按钮	 	表格，用于输入数据
坐标轴，用于输出图形、函数	 	面板，同按钮组类似
按钮组，可将单选按钮，复选框等集成在一起	 	ActiveX控件，可重复使用的程序

GUI工具简介 控件对象



普通按钮，是小的矩形面。将鼠标指针移动至按钮，单击鼠标，按钮被按下随即自动弹起，并执行回调程序。按钮的“Style”属性的默认值是“pushbutton”。



开关按钮，和普通按钮形状相同，区别在于它有两种状态，用鼠标单击按钮，它会从一种状态变成另一种状态，并执行相应的回调程序（两种状态各对应不同的回调程序）。按钮“开”时，Value属性的值为在Max属性中指定的值；按钮“关”时，Value属性的值为在Min属性中指定的值。按钮的“Style”属性的默认值是“togglebutton”。



编辑框，允许用户动态地编辑文本字符串或数字，就象使用文本编辑器或文字处理器一样。编辑框一般用于让用户输入或修改文本字符串和数字。编辑框的“String”属性“的默认值是“Edit Text”。

GUI工具简介 控件对象



单选按钮，由一个标注字符串（在“String”属性中设置）和字符串左侧的一个小圆圈组成。当它被选择时，圆圈被填充一个黑点，且属性“Value”的值为“1”；若未被选择，圆圈为空，属性的“Value”值为“0”。单选按钮一般用于在一组互斥的选项中选择一项。为了确保互斥性，各单选按钮的回调程序需要将其它各项的“Value”值设为“0”。单选按钮“style”的属性的默认值是“Radio Button”。

GUI工具简介 控件对象



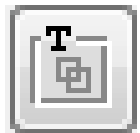
弹出式菜单(Pop-up Menu)，向用户提出互斥的一系列选项清单，用户可以选择其中的某一项。弹出式菜单下拉式菜单，它不受菜单条的限制，可以位于图形窗口内的任何位置。

通常状态下，弹出式菜单以矩形的形式出现，矩形中含有当前选择的选项，在选项右侧有一个向下的箭头来表明该对象是一个弹出式菜单。当指针处在弹出式菜单的箭头之上并按下鼠标时，出现所有选项。移动指针到不同的选项，单击鼠标左键就选中了该选项，同时关闭弹出式菜单，显示新的选项。

选择一个选项后，弹出式菜单的“Value”属性值为该选项的序号。

弹出式菜单的“Style”属性的默认值是“popupmenu”，在“string”属性中设置弹出式菜单的选项字符串，在不同的选项之间用“|”分隔，类似于换行。

GUI工具简介 控件对象



图文面板框，图文框是填充的矩形区域。一般用来把其它控件放入图文框中，组成一组。图文框本身没有回调程序。注意**只有用户界面控件可以在图文框中显示**。由于图文框是不透明的，因而定义图文框的顺序就很重要，必须先定义图文框，然后定义放到图文框中的控件。因为先定义的对象先画，后定义的对象后画，后画的对象覆盖到先画的对象上。



静态文本框(text)，静态文本框用来显示文本字符串，该字符串内容由属性“string”确定。静态文本框之所以称为“静态”，是因为文本不能被动态的修改，而只能通过改变“String”属性来更改。静态文本框一般用于显示标记、提示信息及当前值。静态文本框的“Style”属性的默认值是“text”。

GUI工具简介 控件对象

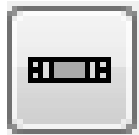


列表框，列出一些选项的清单，并允许用户选择其中的一个或多个选项，一个或多个的模式由Min和Max属性控制。Value 属性的值为被选中选项的序号，同时也指示了选中选项的个数。当单击鼠标按钮选中该项后，Value属性的值被改变，释放鼠标按钮的时候Matlab执行列表框的回调程序。列表框的“Style”属性的默认值是“listbox”。

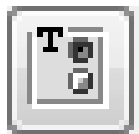


复选框，又称检查框，由一个标注字符串（在“String”属性中设置）和字符串左侧的一个小方框所组成。选中时在方框内添加“√”符号，“Value”属性值设为“1”；未选中时方框变空，“Value”属性值设为“0”。复选框一般用于表明选项的状态或属性。

GUI工具简介 控件对象



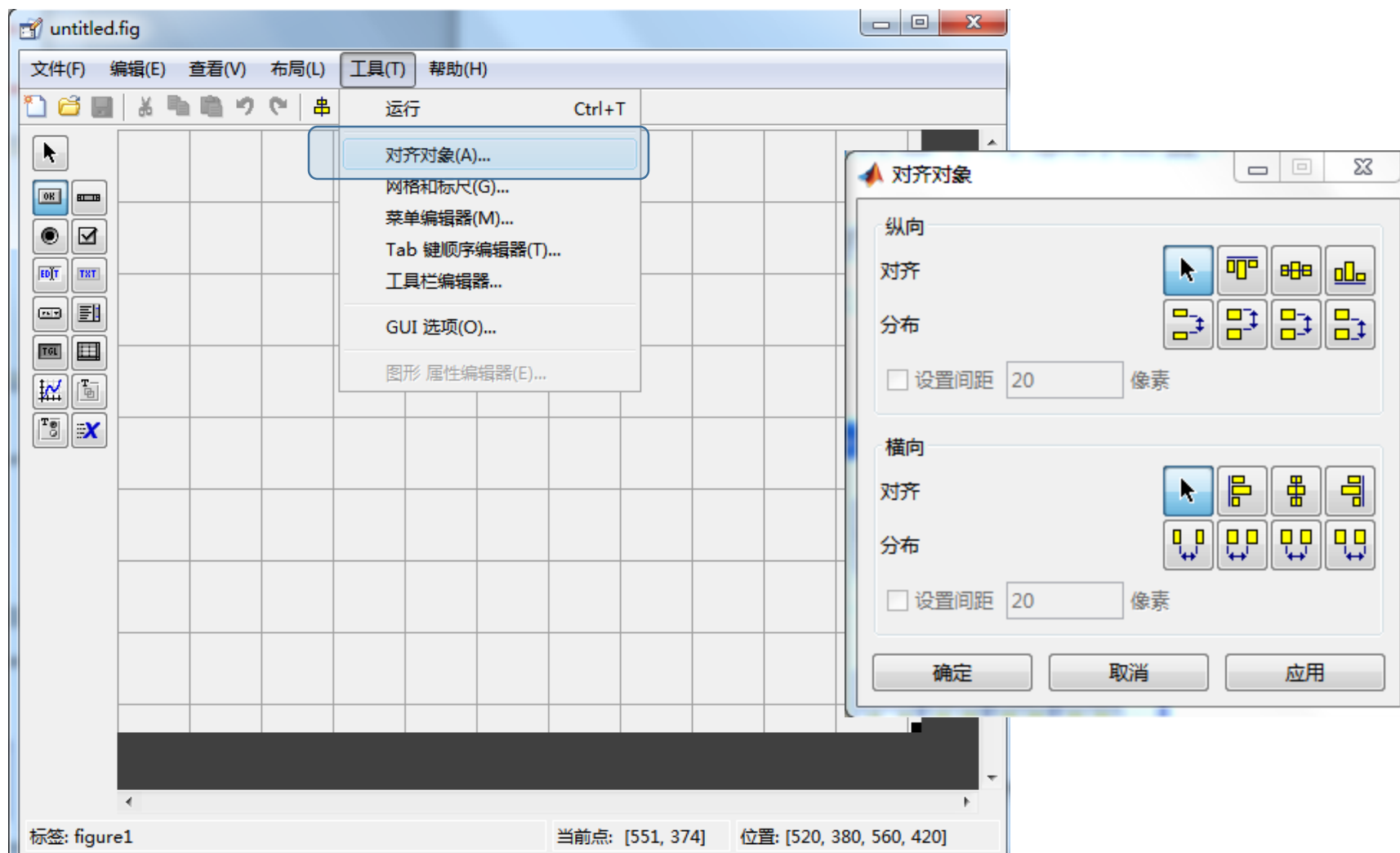
滑动条，又称滚动条，包括三个部分，分别是滑动槽，表示取值范围；滑动槽内的滑块，代表滑动条的当前值；以及在滑动条两端的箭头，用于改变滑动条的值。滑动条一般用于从一定的范围中取值。改变滑动条的值有三种方式，一种是用鼠标指针拖动滑块，在滑块位于期望位置后放开鼠标；另一种是当指针处于滑块槽中但不在滑块上时，单击鼠标按钮，滑块沿该方向移动一定距离，距离的大小在属性“SliderStep”中设置，缺省情况下等于整个范围的10%；第三种方式是在滑块条的某一端用鼠标单击箭头，滑块沿着箭头的方向移动一定的距离，距离的大小在属性“SliderStep”中设置，缺省情况下为整个范围的1%。滑动条的“Style”属性的默认值是“slider”



按钮组，放到按钮组中的多个单选按钮具有排它性，但与按钮组外的单选按钮无关。制作界面时常常会遇到有几组参数具有排它性的情况，即每一组中只能选择一种情况。此时，可以用几组按钮组表示这几组参数，每一组单选按钮放到一个按钮组控件中。

GUI工具简介 控件位置对齐工具

对齐工具可便捷的调整GUI编辑器中控件对象的位置。



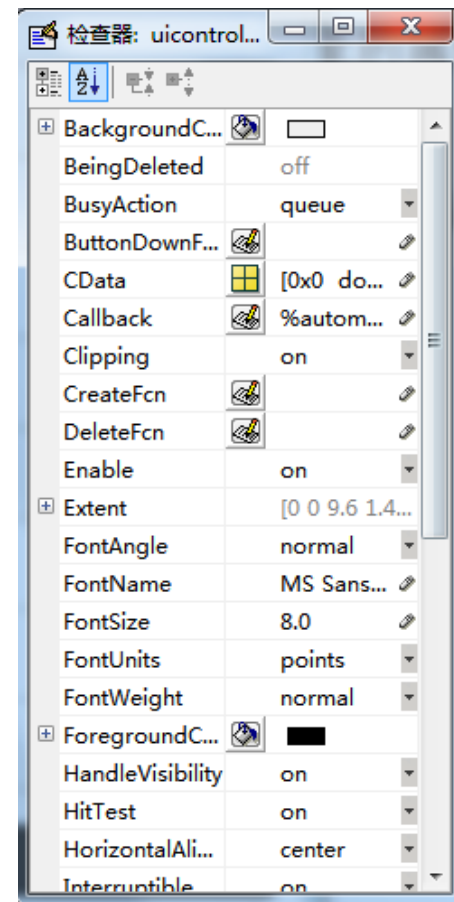
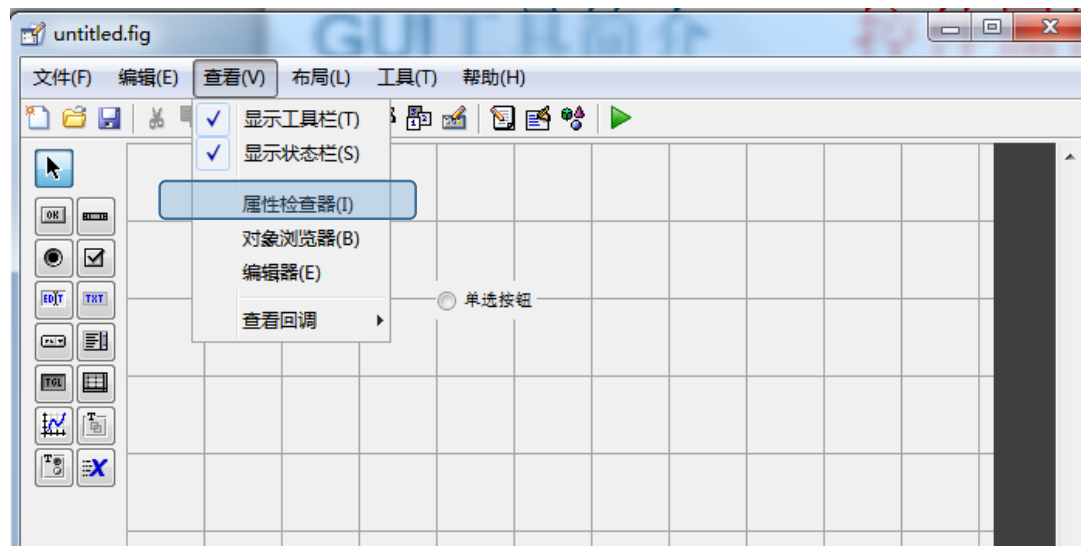
GUI工具简介

控件属性检查/编辑器

控件属性检查/编辑器可查看和修改控件对象的属性值。

打开方式

1. GUI编辑器工具栏中View->Inspect Properties选项
2. 命令窗口的输入inspect
3. 双击

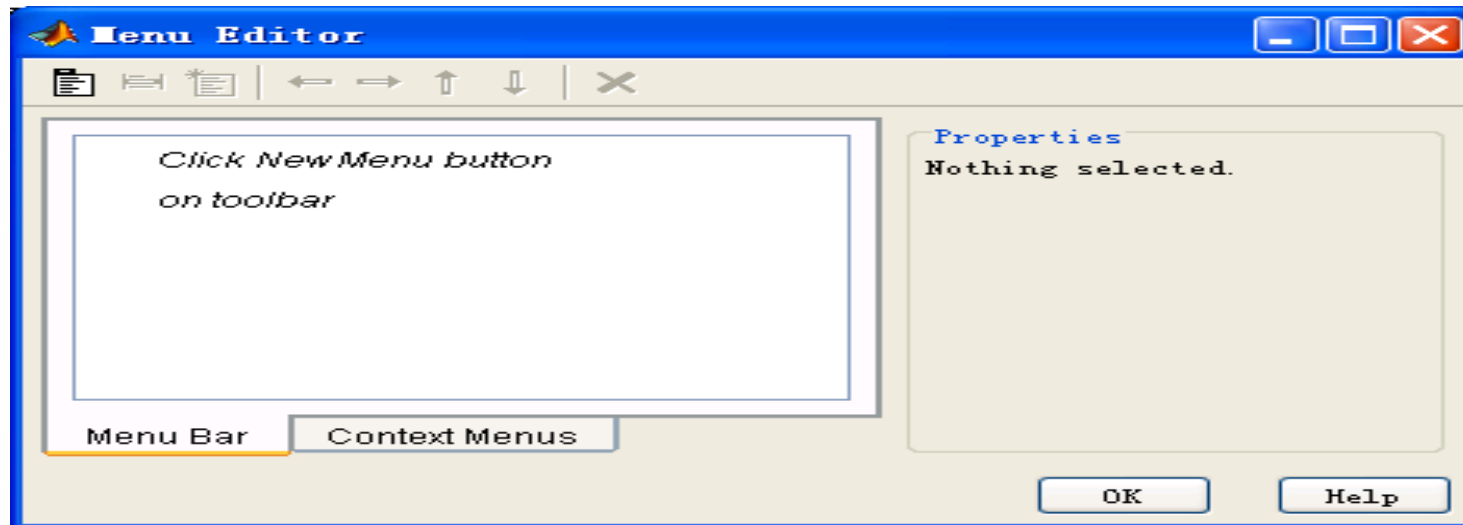


GUI工具简介 菜单编辑器

菜单编辑器，可以创建、设置、修改下拉式菜单（图形窗口菜单栏中的菜单），以及弹出式菜单（在图形对象上单击鼠标右键时弹出的上下文菜单）

打开方式

1. GUIDE中工具栏上的按钮
2. 菜单Tools->Menu Editor

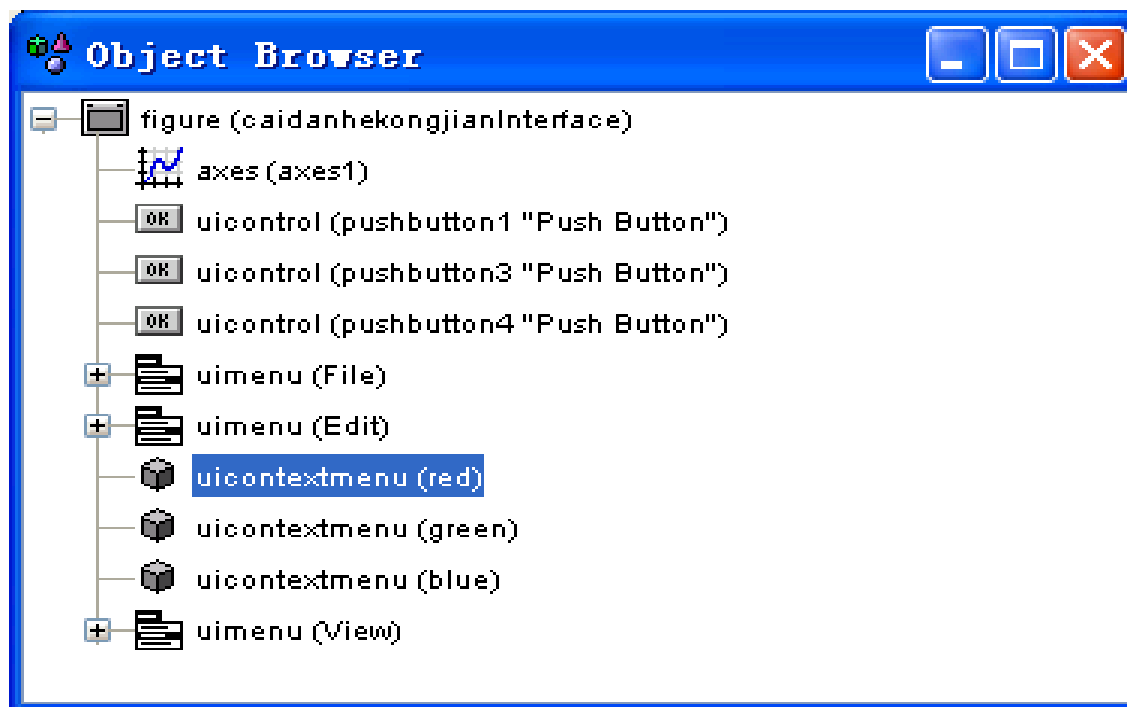


GUI工具简介 对象浏览器

对象浏览器，可查看当前所创建的图形对象。

打开方式

1. 从对象设计编辑器界面的工具栏上选择
2. 菜单View -> Object Browser



GUI向导设计

【例】 设计一个带有三个按钮和一个坐标轴的图形用户界面，当用鼠标点击三个按钮时，分别在坐标轴内画sphere，peaks和membrane三个图形

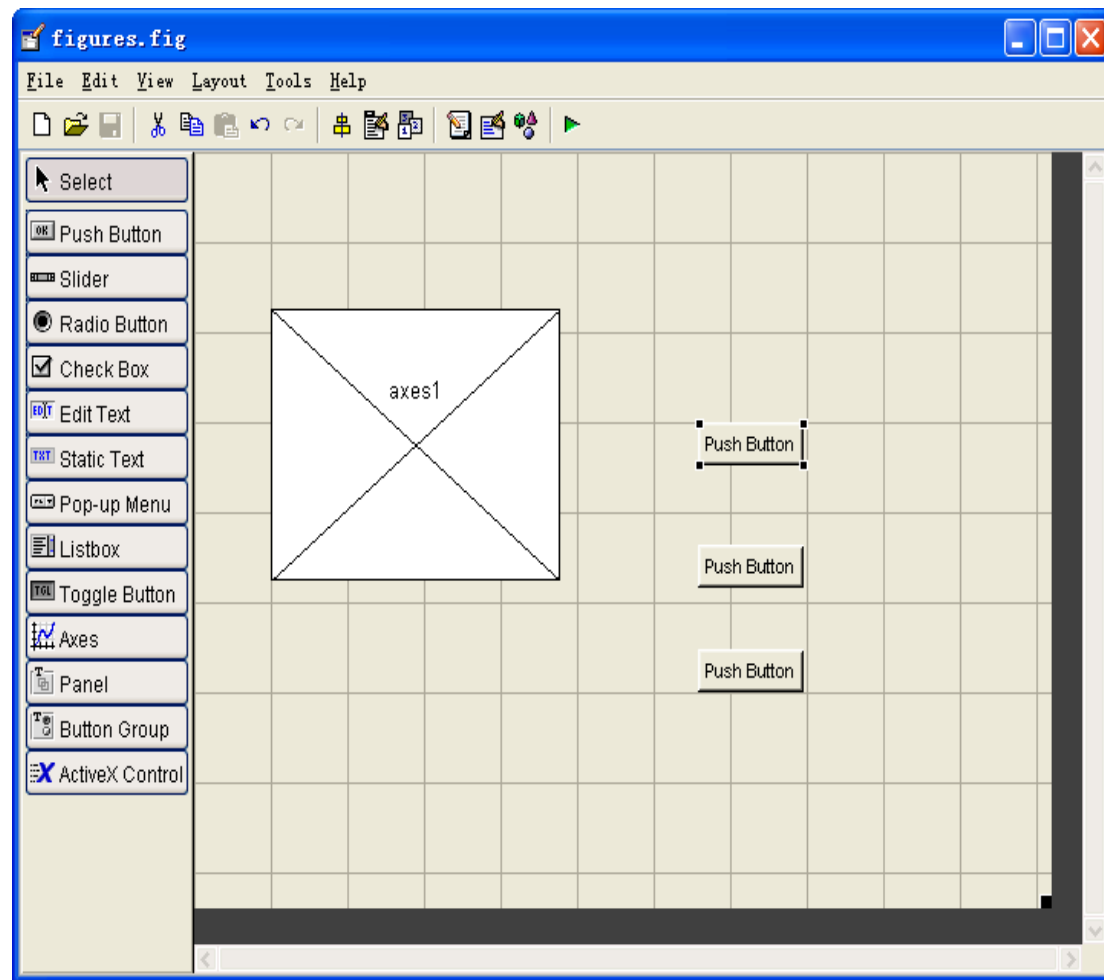
1. 启动GUI空白模板

建立带有一个坐标轴和三个按钮控件的图形界面

2. 用位置排列工具使控件对齐

将待调整的控件同时选中（按住Ctrl键，鼠标左键逐一点击要调整的控件，或按住鼠标左键进行拖拽框选），再后点击工具栏上的打开几何位置排列工具窗口，进行对齐操作。

GUI向导设计

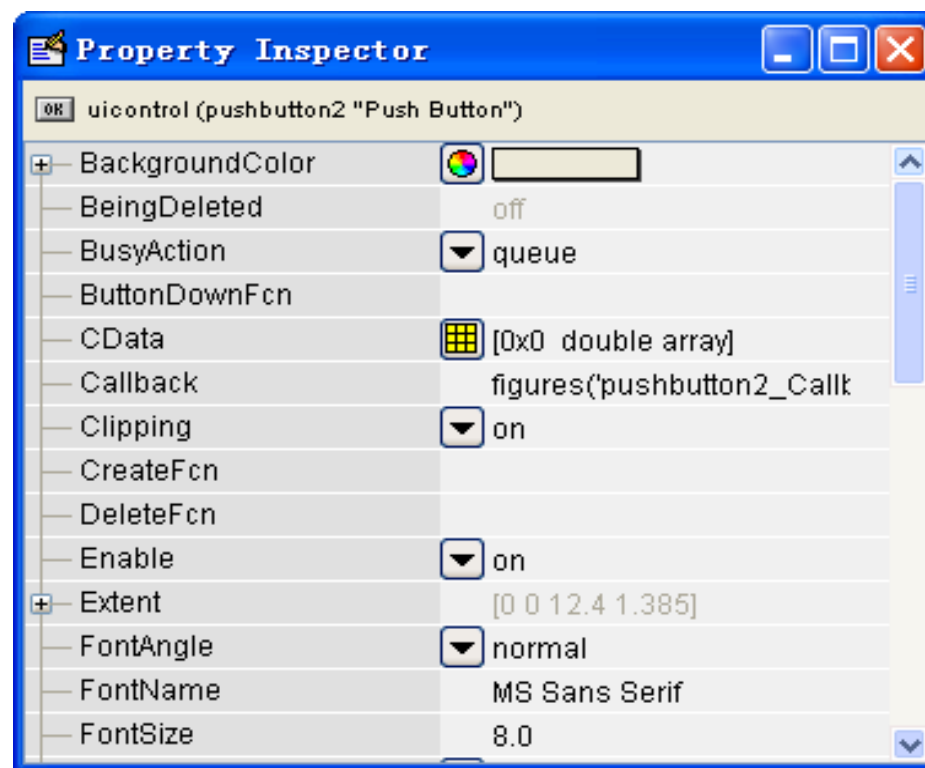


调整空间位置后的界面设计编辑器

GUI向导设计

3. 设置控件属性

在按钮上双击鼠标左键可以打开按钮属性编辑器，如下图。该图左侧是按钮的所有属性，右侧是其属性值。

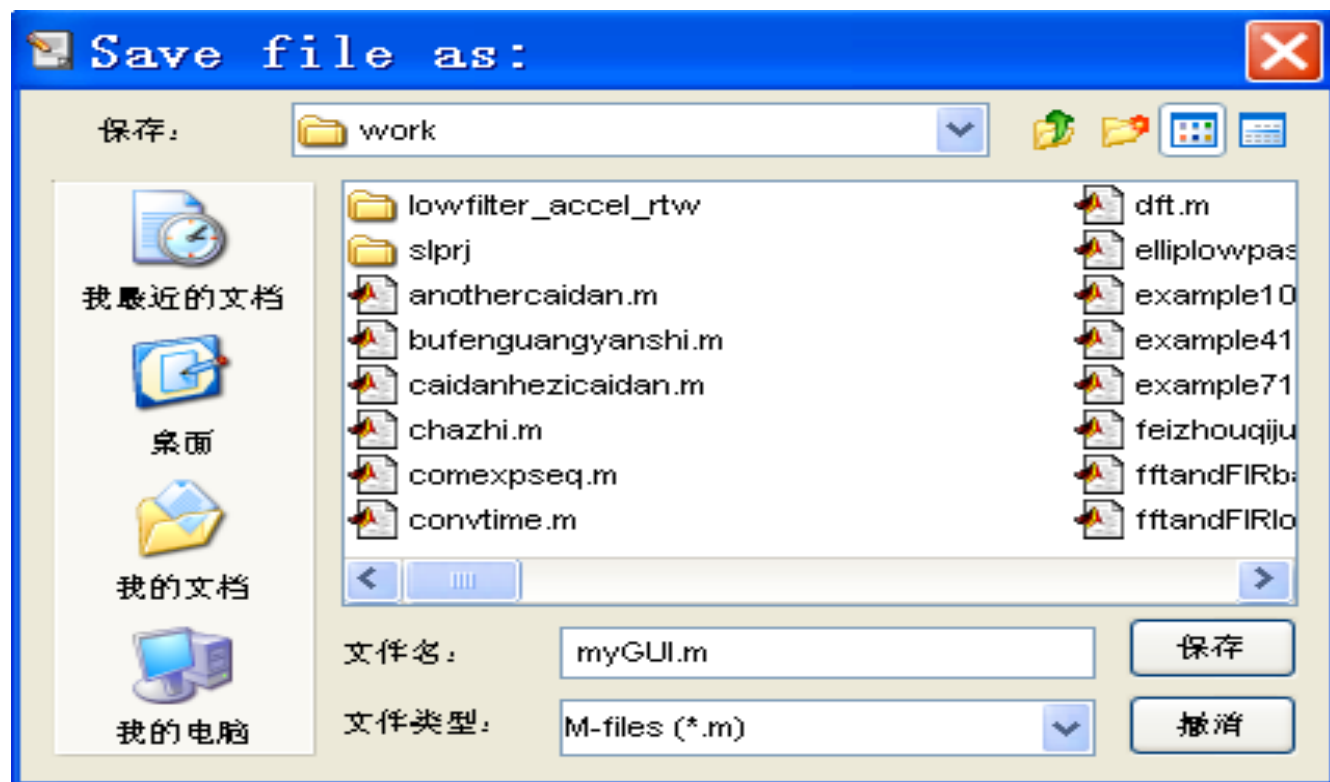


属性编辑器

GUI向导设计

4. 编写回调程序

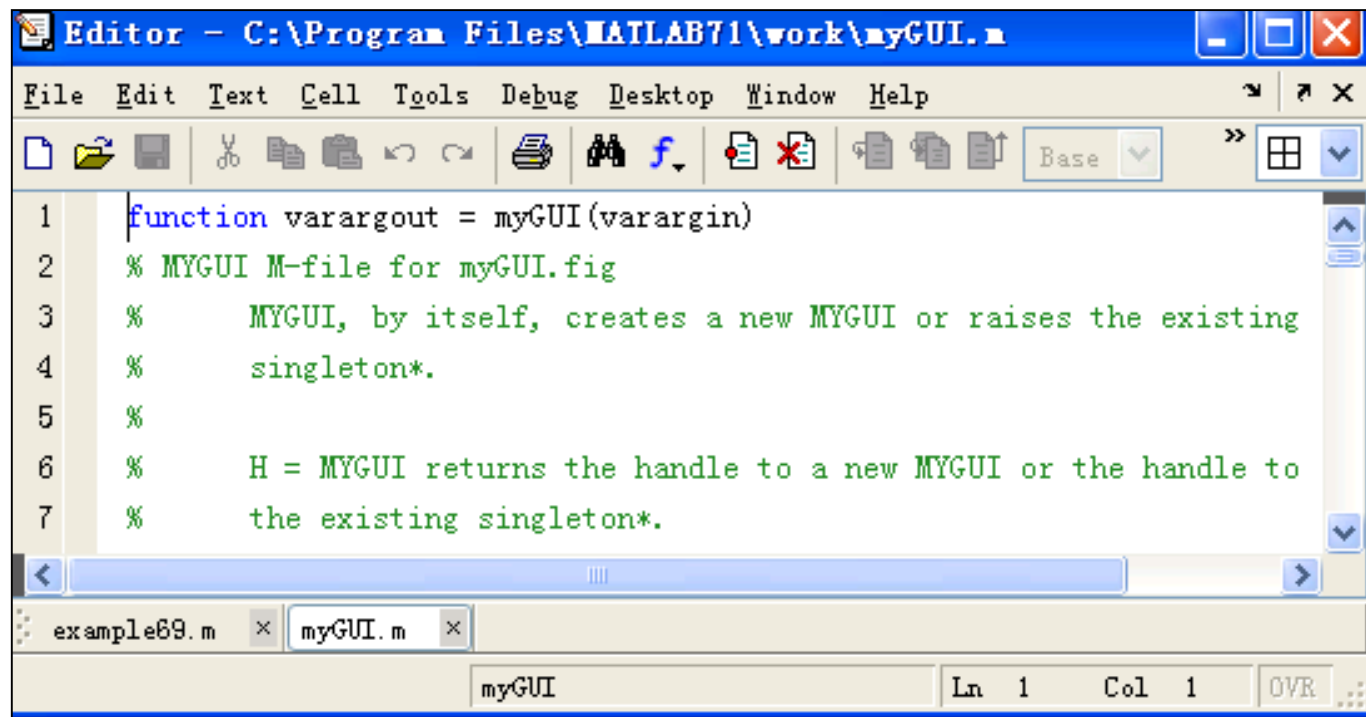
以上工作结束后，要定义三个按钮的功能。首先点击菜单栏上的保存按钮进行文件的保存，会弹出“save as”对话框，进行FIG文件保存，文件命名为myGUI。



保存图形界面的对话框图

GUI向导设计

同时，Matlab会自动创建一个同名的M文件，并且自动打开。



The screenshot shows the MATLAB Editor window titled "Editor - C:\Program Files\MATLAB71\work\myGUI.m". The menu bar includes File, Edit, Text, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations and editing. The main text area displays the following code:

```
1 function varargout = myGUI(varargin)
2 % MYGUI M-file for myGUI.fig
3 %     MYGUI, by itself, creates a new MYGUI or raises the existing
4 %     singleton*.
5 %
6 %     H = MYGUI returns the handle to a new MYGUI or the handle to
7 %     the existing singleton*.
```

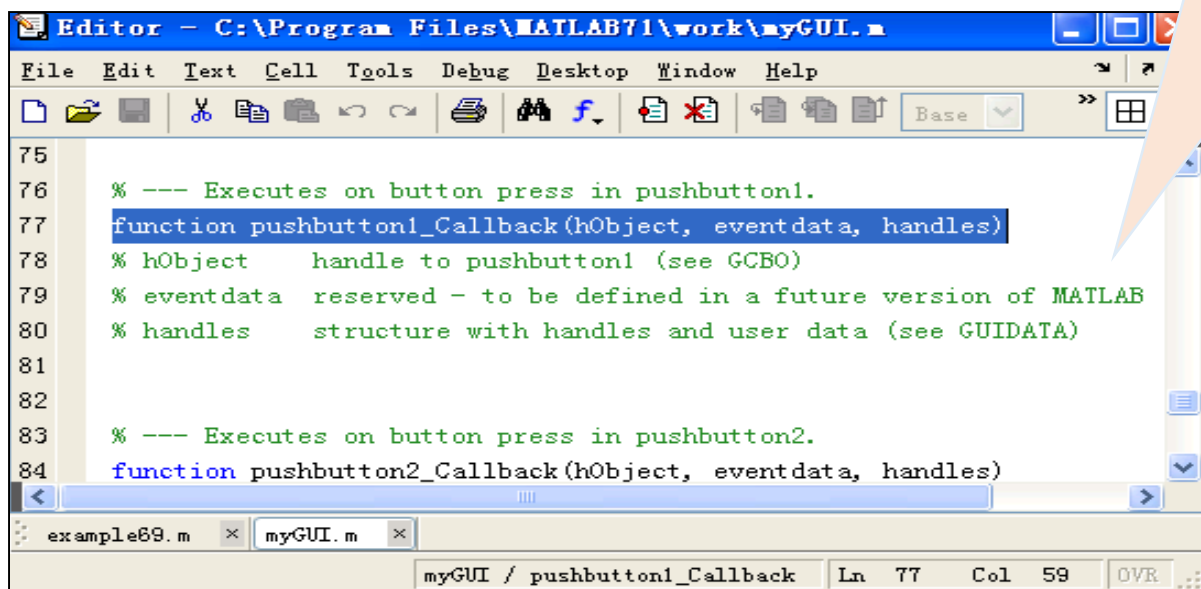
The status bar at the bottom shows "myGUI" and "Ln 1 Col 1".

自动生成的M文件

其次，在自动生成的M文件中，找到与三个按钮有关的回调子函数。

GUI向导设计

按钮1的回调子函数



The image shows a MATLAB Editor window with the following content:

```
75  
76 % --- Executes on button press in pushbutton1.  
77 function pushbutton1_Callback(hObject, eventdata, handles)  
78 % hObject    handle to pushbutton1 (see GCBO)  
79 % eventdata  reserved - to be defined in a future version of MATLAB  
80 % handles    structure with handles and user data (see GUIDATA)  
81  
82  
83 % --- Executes on button press in pushbutton2.  
84 function pushbutton2_Callback(hObject, eventdata, handles)
```

The window title is "Editor - C:\Program Files\MATLAB71\work\myGUI.m". The status bar at the bottom shows "myGUI / pushbutton1_Callback Ln 77 Col 59 OVR".

按钮1的回调子函数

GUI向导设计

下面就可以在该子函数的空白处填写回调程序了。

例如，按钮“sphere”的回调程序为：

`sphere;`

`axis tight;`

按钮“peaks”的回调程序为：

`peaks;`

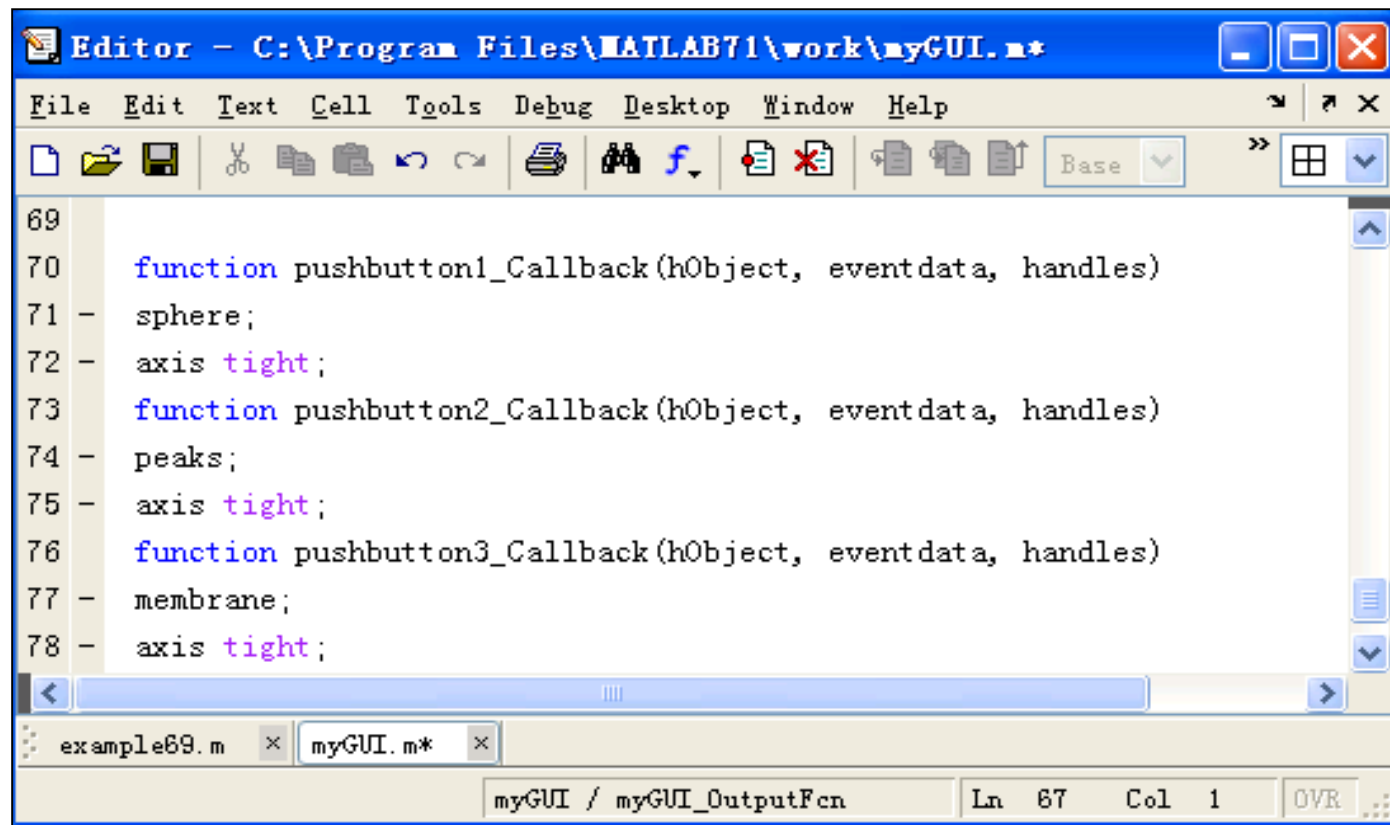
`axis tight;`

按钮“membrane”的回调程序为：

`membrane;`

`axis tight;`

上面的三个回调程序在自动生成的命名为myGUI的M文件编辑器中录入的情况如图所示（注释已删除）



```
Editor - C:\Program Files\MATLAB71\work\myGUI.m*
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Base
69
70 function pushbutton1_Callback(hObject, eventdata, handles)
71 - sphere;
72 - axis tight;
73 function pushbutton2_Callback(hObject, eventdata, handles)
74 - peaks;
75 - axis tight;
76 function pushbutton3_Callback(hObject, eventdata, handles)
77 - membrane;
78 - axis tight;
```

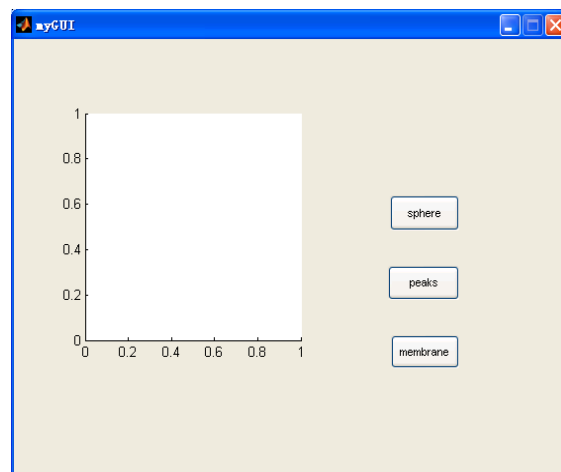

GUI向导设计

程序录入后，点击保存图标，将M文件重新保存，然后点击M文件编辑器上的运行快捷按钮或返回到未被激活的图形界面，点击工具栏上的运行图标，此时则生成被激活的图形界面。当点击“**sphere**”按钮时，在空白的坐标轴处出现球体，点击“**peaks**”按钮时，在空白的坐标轴处出现尖峰图，点击“**membrane**”按钮时，在空白的坐标轴处出现membrane图。

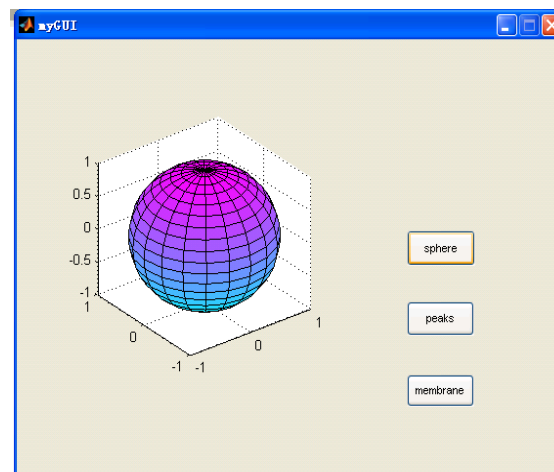
至此，图形界面的GUI向导设计就完成了。需要说明的是，编写回调程序时，也可以在属性编辑器的Callback属性中录入代码。但是此方法只适用于代码语句较少的情况。

GUI向导设计

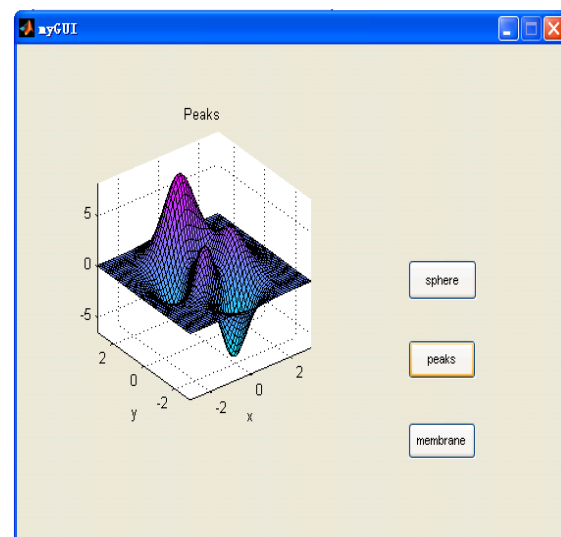
被激活后的界面



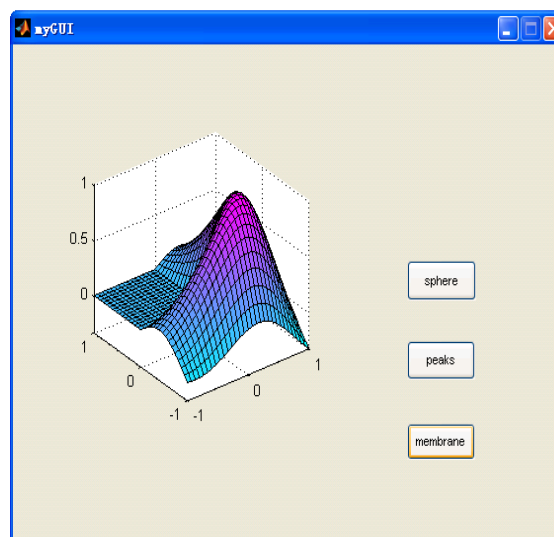
sphere图



peaks图



membrane图



GUI程序设计 通过编写函数文件实现GUI开发

主要涉及三个函数：

1. `uimenu` (菜单)

2. `uicontextmenu` (上下文菜单)

3. `uicontrol` (控件)

GUI程序设计

1. 用户界面菜单对象的建立

自制用户菜单对象，通过函数uimenu创建，调用格式为：

```
h=uimenu( 'PropertyName1' , value1, 'PropertyName2' ,value2, ...)
```

即在当前图形窗口上部的菜单栏创建一个菜单对象，并返回一个句柄值。函数变量PropertyName是所建菜单的属性，value是属性值。菜单对象的属性分为公共属性、基本控制属性和callback管理属性三部分，关于属性及其的详细内容见Matlab帮助文件，这里介绍一些常用重要属性的设置方法。

GUI程序设计

① label和callback

这是菜单对象的基本属性，编写一个具有基本功能的菜单必须要设置label和callback属性。label是在菜单项上显示的菜单内容；callback是用来设置菜单项的回调程序。

② checked和 separator

checked 属性用于设置是否在菜单项前添加选中标记。记为” on”表示添加，” off”表示不添加。因为有些菜单的选中标记相斥，这就要求给一个菜单项添加选中标记的同时去掉另一个选项的标记；

separator 用于在菜单项之前添加分隔符，以便使菜单更加清晰。

③ Background Color和 Foreground Color

Background Color(背景色)是菜单本身的颜色；

Foreground Color(前景色)是菜单内容的颜色。

GUI程序设计

【例】 建立一个包含用户界面菜单项的图形界面，并可执行菜单项的相应功能，分别绘制membrane、peaks和sinc函数图形。

Matlab程序如下：

% 首先建立一个图形窗口，去除窗口本身包含的菜单条和工具条，并命名为myfirstGUI

```
h0=figure('menubar','none','toolbar','none','name','myfirstGUI');
```

% 从左至右依次建立各级菜单% 先建立Draw菜单和其下的Membrane, Peaks和Sinc子菜单项

```
h1=uimenu(h0,'label','Draw');
```

```
h11=uimenu(h1,'label','Membrane','callback','membrane');
```

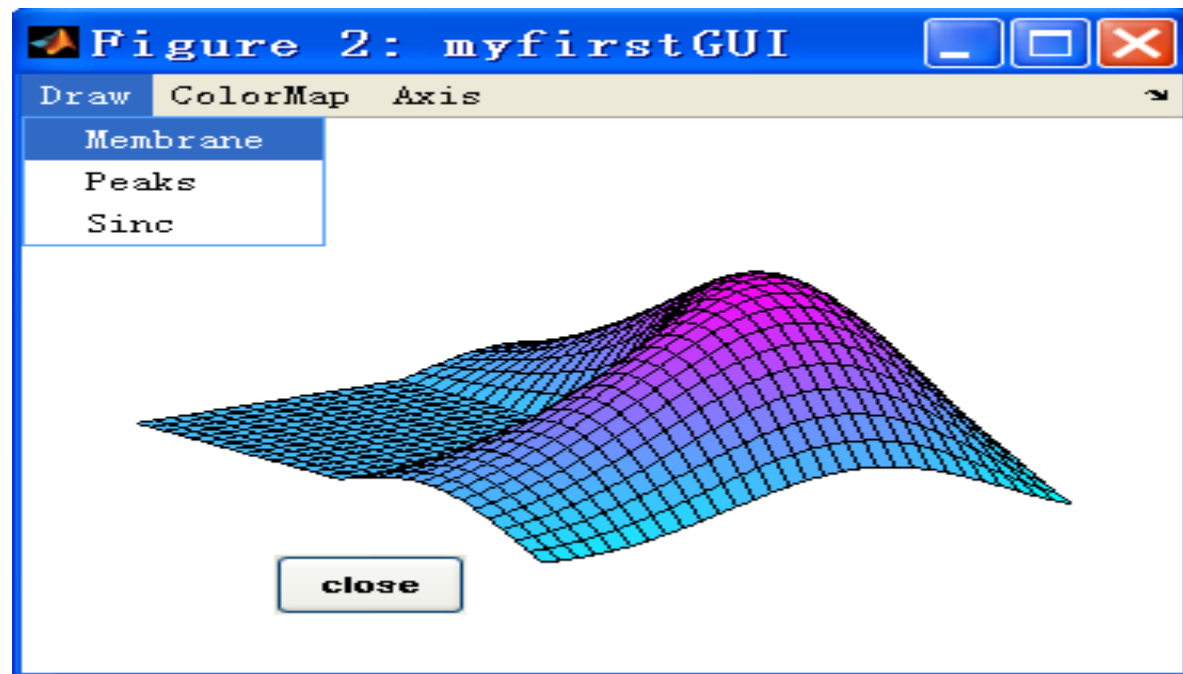
```
h12=uimenu(h1,'label','Peaks','callback','peaks');
```

```
h13=uimenu(h1,'label','Sinc','callback',...
```

```
'[[x,y]=meshgrid(-5:0.5:5);','r=sqrt(x.^2+y.^2)+eps;',...
```

```
'z=sin(r)./r;', 'surf(z);']];
```

该段代码运行后，生成“membrane”的图形界面

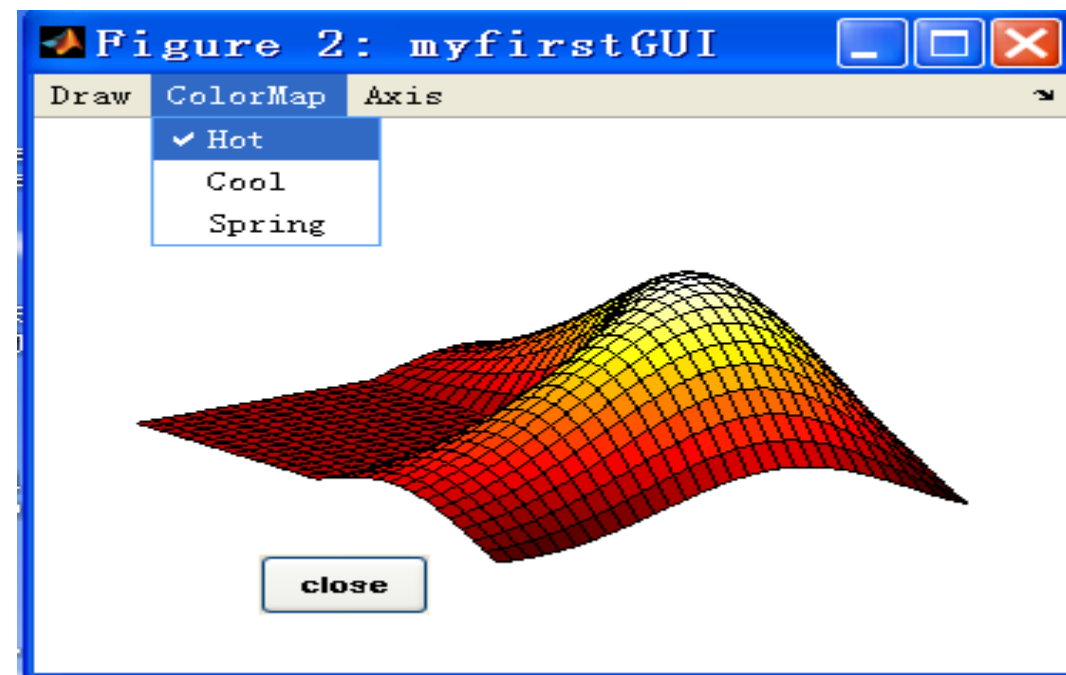


GUI程序设计

% 建立第二个菜单【Colormap】及其下的【Cool】【Hot】【Default】子菜单项，当某项被选中时，添加选中标记，同时去掉其它选项的选中标记

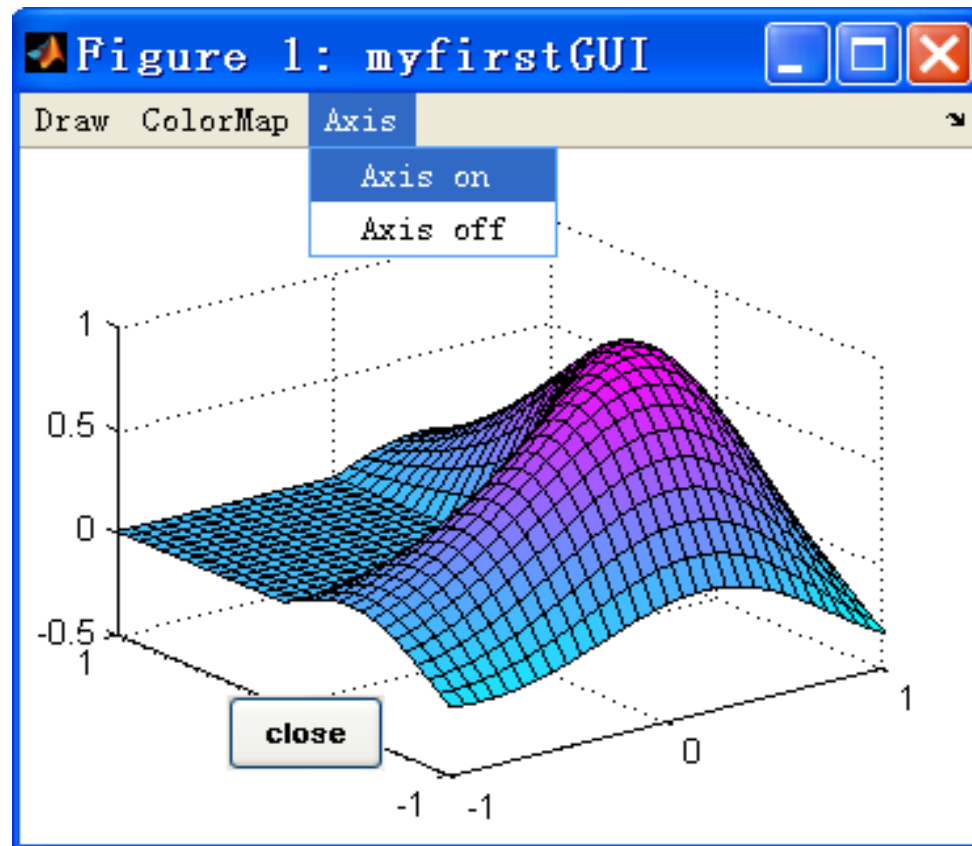
```
h2=uimenu(h0,'label','ColorMap');  
h22(1)=uimenu(h2,'label','Hot',...  
'callback',...  
['set(h22,"checked","off");',...  
'set(h22(1),"checked","on");','colormap(hot);']);  
h22(2)=uimenu(h2,'label','Cool',...  
'callback',...  
['set(h22,"checked","off");',...  
'set(h22(2),"checked","on");','colormap(cool);']);  
h22(3)=uimenu(h2,'label','Spring',...  
'callback',...  
['set(h22,"checked","off");',...  
'set(h22(3),"checked","on");','colormap(Spring);']);
```

该段代码运行后，生成hot效果图形界面。



GUI程序设计

% 建立控制坐标轴的显示菜单【Axis】，用于是否显示坐标轴
h3=uimenu(h0,'label','Axis'); h31=uimenu(h3,'label','Axis on','callback','axis on');
h32=uimenu(h3,'label','Axis off','callback','axis off');
该段代码运行后，生成如下带坐标的图形界面。



GUI程序设计

% 建立关闭图形用户界面按钮【close】

```
hbutton=uicontrol('position',[80 30 60 30], 'string',  
'close', 'fontsize', 8, ...  
'fontweight', 'bold', 'callback', 'close');
```

至此，就形成了一个用户菜单界面，执行结果如以上三图所示。实际上，运行上面的代码后，仅产生一个图形界面，为了让读者看清界面内的主菜单下的子菜单所对应的功能，在相应的位置给出了该子菜单的Callback属性的回调程序结果。

GUI程序设计

2. 用户界面上下文菜单的建立

与固定位置的菜单对象相比，上下文菜单对象的位置不固定，总是与某些图形对象相联系，并通过鼠标右键激活，制作上下文菜单步骤如下：

- ① 用函数`uicontextmenu`创建上下文菜单对象；
- ② 用函数`uimenu`为该上下文菜单对象制作具体的菜单项；
- ③ 用函数`set`将上下文菜单对象和某些图形对象联系在一起。

GUI程序设计

【例】 在一个图形窗口绘制抛物线和余弦曲线，并创建一个与之相联系的上下文菜单，用于控制线条的颜色、线宽、线型及标记点风格。

```
% 画曲线y1，并设置其句柄h=uicontextmenu;  
t=-1:0.1:1;subplot(2,1,1);y1=t.^2;h_line1=plot(t,y1);  
% 建立上下文菜单  
h=uicontextmenu;  
uimenu(h,'label','red','callback','set(h_line1,"color","r")');  
uimenu(h,'label','green','callback','set(h_line1,"color","g")');  
uimenu(h,'label','yellow','callback','set(h_line1,"color","y")');  
uimenu(h,'label','linewidth1.5','callback','set(h_line1,"linewidth",1.5)');  
uimenu(h,'label','linestyle*','callback','set(h_line1,"linestyle","*")');  
uimenu(h,'label','linestyle:','callback','set(h_line1,"linestyle",":")');  
uimenu(h,'label','marker','callback','set(h_line1,"marker","s")');  
set(h_line1,'uicontextmenu',h) % 使上下文菜单与正弦曲线h_line1相联系  
title('抛物线和余弦曲线','fontweight','bold','fontsize',14)  
set(gca,'xtick',[-1:0.5:1]) % 设置坐标轴的标度范围  
set(gca,'xticklabel',{'-1','0.5','0','0.5','1'}) % 设置坐标轴的标度值
```

GUI程序设计

%画曲线y2, 并设置其句柄

```
subplot(2,1,2);t=0:0.1:2*pi;y2=cos(t);h_line2=plot(t,y2);  
h=uicontextmenu;  
uimenu(h,'label','red','callback','set(h_line2,"color","r")');  
uimenu(h,'label','crimson','callback','set(h_line2,"color","m")');  
uimenu(h,'label','black','callback','set(h_line2,"color","k")');  
uimenu(h,'label','linewidth1.5','callback','set(h_line2,"linewidth",1.5)');  
uimenu(h,'label','linestyle*','callback','set(h_line2,"linestyle","*")');  
uimenu(h,'label','linestyle:','callback','set(h_line2,"linestyle",":")');  
uimenu(h,'label','marker','callback','set(h_line2,"marker","s")');  
set(h_line2,'uicontextmenu',h)  
set(gca,'xtick',[0:pi/2:2*pi])  
set(gca,'xticklabel',{'0','pi/2','pi','3pi/2','2pi'})  
xlabel('time 0-2\pi','fontsize',10)
```

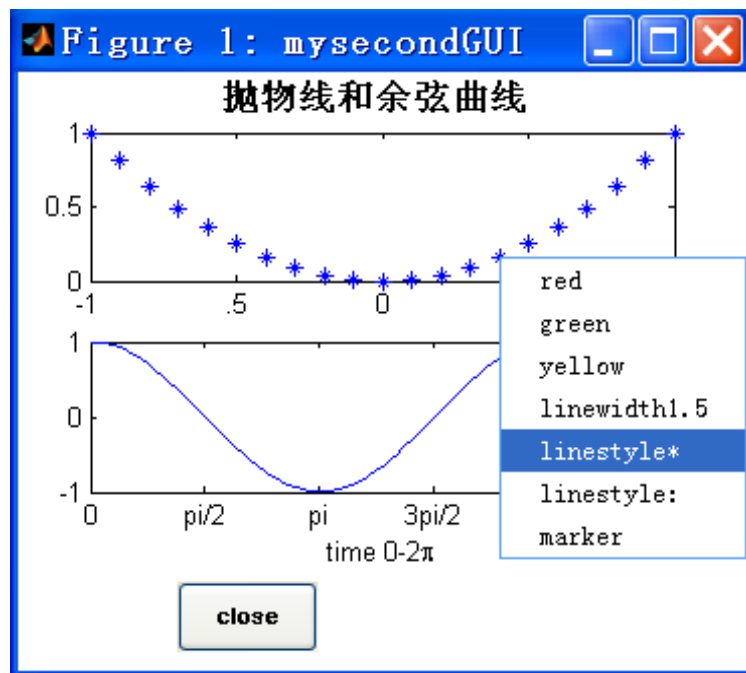
GUI程序设计

% 建立关闭图形用户界面按钮【close】

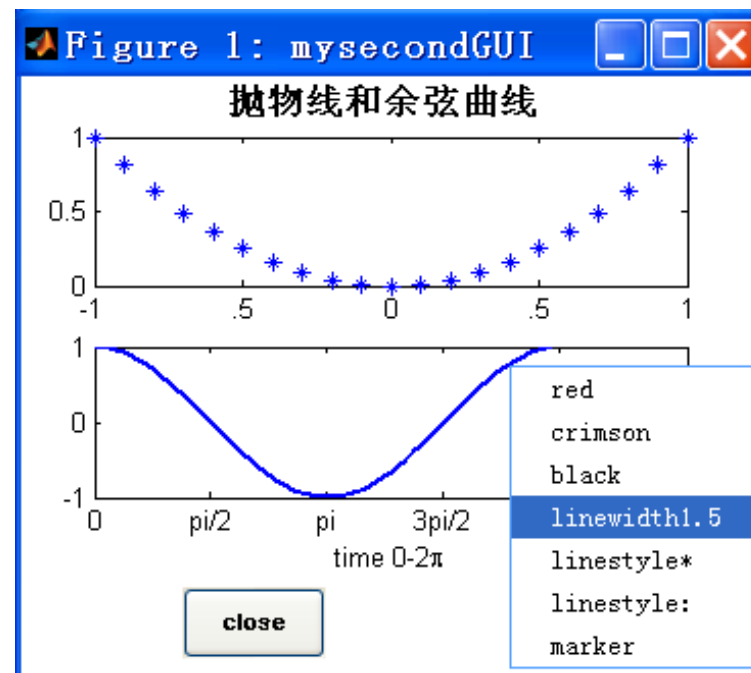
```
hbutton=uicontrol('position',[80 30 60 30],'string','close',  
'fontsize',8,'fontweight','bold','callback','close');
```

运行该程序段，得到如下图所示图形。将鼠标指向线条，单击鼠标右键，弹出上下文菜单，在选中某菜单项后，将执行该菜单项的操作。

GUI程序设计



(a)



(b)

带有上下文菜单的图形界面

(a) 将抛物线的线型设为 “*” 时的图形界面；

(b) 将余弦曲线的线宽设为 “1.5” 时的图形界面

GUI程序设计

3、用户界面控件对象的建立

除了菜单以外，控件对象是另一种实现用户与计算机交互的重要手段。用户界面控件对象是这样一类图形界面的对象：用户用鼠标在控件对象上进行操作，鼠标点击控件时，将激活该控件所对应的后台应用程序，并执行该程序。利用函数命令创建控件对象的格式为：

```
H=uicontrol ('PropertyName1', value1, 'PropertyName2', value2, ...)
```

当用函数创建控件时，这里我们有必要对控件的几个重要属性给予介绍。

GUI程序设计

Value属性：控件的当前值，格式为标量或变量。
该属性对不同的控件有不同的取值方式，分别为：

- 复选框：当此控件被选中时，Value的值为属性Max中的设置的值；未被选中时Value的值为属性中设置的值。
- 列表框：被选中选项的序号，当有多个选项被选中时，Value的属性值为向量。序号指的是选项的排列次序，最上面的选项序号为1，第二个选项序号为2。
- 弹出式菜单：和列表框类似，也是被选中选项的序号，只是弹出式菜单只能有一个选项被选中，因而Value属性值是标量。
- 单选按钮：被选中时Value的值为属性Max中设置的值；未被选中时，Value的值为属性Min中设置的值。
- 滑动条：Value的值等于滑块指定的值。
- 开关按钮：“开”时Value的值为属性Max中设置的值；“关”时Value的值为属性Min中设置的值。

GUI程序设计

Max属性：指定Value属性中可以设置的最大值，格式为标量。
该属性对不同的控件有不同的含义，分别如下所述：

- 复选框：当复选框被选中时Value属性的取值。
- 编辑框：如果Max的值减去Min的值大于1，那么编辑框可以接受多行输入文本；如果Max的值减去Min的值小于或等于1，那么编辑器只能接受一行输入文本。
- 列表框：如果Max的值减去Min的值大于1，那么允许选取多个选项；如果Max的值减去Min的值小于或等于1，那么只能选取一个选项。
- 单选按钮：当单选按钮被选中时Value属性的取值。
- 滑动条：滑动条的最大值，缺省值是1
- 开关按钮：当开关按钮“开”（被选中）时Value属性的取值。缺省值是1。

文本框、弹出式菜单、按钮和静态文本框无此属性。

GUI程序设计

Min属性：指定Value属性中可以设置的最小值，格式为标量。
该属性对不同的控件有不同的含义，分别如下所述：

- 复选框：当复选框被选中时Value属性的取值。
- 编辑框：如果Max 的值减去Min的值大于1，那么编辑框可以接受多行输入文本；如果Max的值减去Min的值小于或等于1，那么编辑器只能接受一行输入文本。
- 列表框：如果Max的值减去Min的值大于1，那么允许选取多个选项；如果Max的值减去Min的值小于或等于1，那么只能选取一个选项。
- 单选按钮：当单选按钮未被选中时Value属性的取值。
- 滑动条：滑动条的最小值，缺省值是0。
- 开关按钮：当开关按钮“开”（被选中）时属性的取值。缺省值是1。

文本框、弹出式菜单、按钮和静态文本框无此属性。

GUI程序设计

【例】 建立一个包含控件的图形用户界面，点击控件时执行该控件的相应功能，要求绘制membrane， peaks和sphere函数图形，并有光照控制效果。

```
% 建立图形窗口和坐标轴，去除窗口本身的菜单条和工具条，并命名为 mysecondGUI
h0=figure('menubar','none','toolbar','none','position',[198 56 408 468],'name','mysecondGUI');
h1=axes('parent',h0,'position',[0.15 0.45 0.7 0.5],'visible','off');
% 建立静态文本框和动态文本框
htext1=uicontrol('parent',h0,'units','points','position',[54 110 45 15],'string','input title','style','text');
hedit=uicontrol('parent',h0,'units','points','position',[100 110 45
16],'callback','title(get(hedit,'string'))','style','edit');
```

GUI程序设计

% 创建三个按钮

```
hbutton1=uicontrol('parent',h0,'units','points','string','Sphere',...  
'position',[20 65 50 18],'callback','mesh(sphere);axis tight');  
hbutton2=uicontrol('parent',h0,'units','points','string','Membrane',...  
'position',[75 65 50 18],'callback','mesh(membrane);axis tight');  
hbutton3=uicontrol('parent',h0,'units','points','string','Sinc','position',[135 65 50 18],'callback',...  
['[x,y]=meshgrid(-5:0.5:5);','r=sqrt(x.^2+y.^2)+eps;','z=sin(r)./r;','mesh(x,y,z)']);
```

% 创建静态文本框和滚动条，鼠标拖动滚动条控制图形的颜色变化

```
htext2=uicontrol('parent',h0,'units','points','position',[20 30 45 15],'string','brightness','style','text');  
hslider=uicontrol('parent',h0,'units','points','position',[65 30 120 15],'min',-1,'max',1,'style','slider','callback',...  
'brighten(get(hslider,"value"))');
```

GUI程序设计

% 建立静态文本框和五个单选按钮

```
htext3=uicontrol('parent',h0,'units','points','position',[200 130 80 15],'string','select color:','style','text');  
hradio(1)=uicontrol('parent',h0,'units','points','position',[200 115 80 15],'string','default','style',...  
'radiobutton','value',1,'callback',...  
['set(hradio,"value",0);','set(hradio(1),"value",1);','colormap("default")']);  
hradio(2)=uicontrol('parent',h0,'units','points','position',[200 100 80 15],'string','spring','style',...  
'radiobutton','value',1,'callback',...  
['set(hradio,"value",0);','set(hradio(2),"value",1);','colormap(spring)']);  
hradio(3)=uicontrol('parent',h0,'units','points','position',[200 85 80 15],'string','summer','style',...  
'radiobutton','value',1,'callback',...  
['set(hradio,"value",0);','set(hradio(3),"value",1);','colormap(summer)']);
```

GUI程序设计

```
hradio(4)=uicontrol('parent',h0,'units','points','position',[200 70 80 15],'string','autumn','style',...  
'radiobutton','value',1,'callback',['set(hradio,"value",0);','set(hradio(4),"value",1);','colormap(autumn)']);
```

```
hradio(5)=uicontrol('parent',h0,'units','points','position',[200 55 80 15],'string','winter','style',...  
'radiobutton','value',1,'callback',['set(hradio,"value",0);','set(hradio(5),"value",1);','colormap(winter)']);
```

% 建立关闭图形用户界面按钮 **【close】**

```
hbutton4=uicontrol('parent',h0,'units','points','string','Close','position',[200 30 50 18],'callback','close');
```

GUI程序设计

至此，建好了带有控件的图形用户界面，下半部分仅带控件，上半部分为空的图形界面。在静态文本框中输入“sinc”，点击“Sinc”按钮，则产生如下图所示的图形界面。

类似的，点击另外两个按钮“Sphere”和“Membrane”，会生成相应的三维图形；在滚动条上按住鼠标左键不动，拖动鼠标会产生不同的明亮度效果；用鼠标点击单选按钮，会产生不同的色彩效果。

