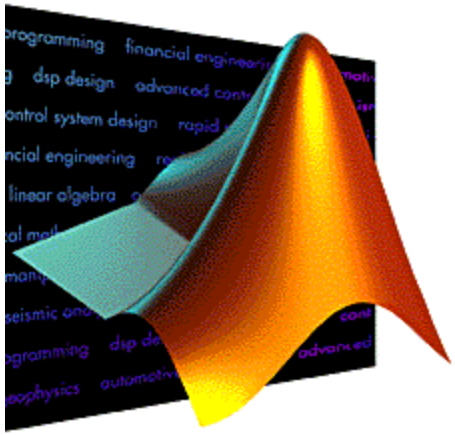
Matlab 程序设计与应用

第8章： MATLAB图形用户界面设计

伍振海



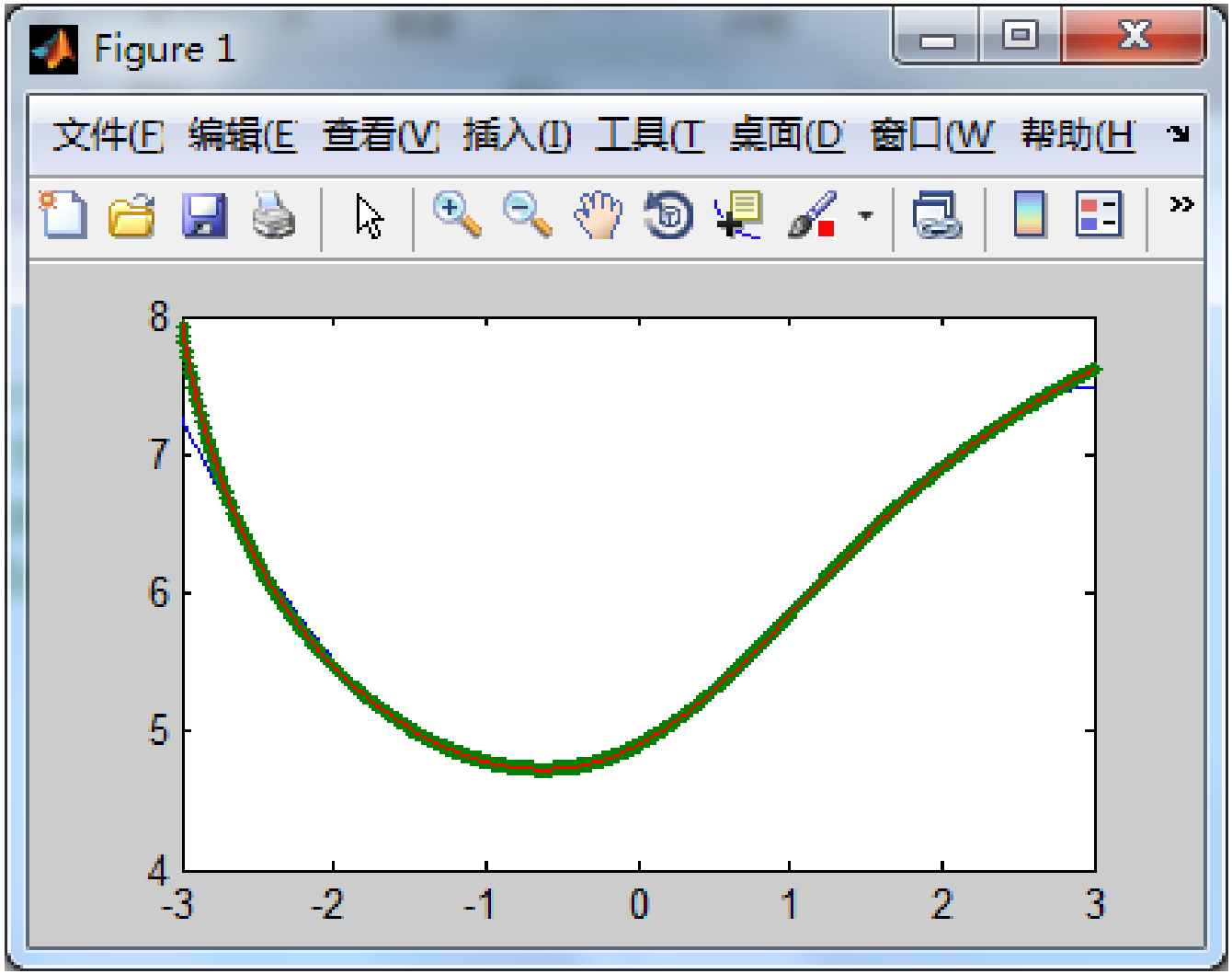
September 24, 2017

Autumn, @swpu

第8章　MATLAB图形用户界面设计

- MATLAB用户界面对象的组成
- MATLAB菜单设计
- MATLAB对话框设计
- MATLAB图形用户界面设计工具

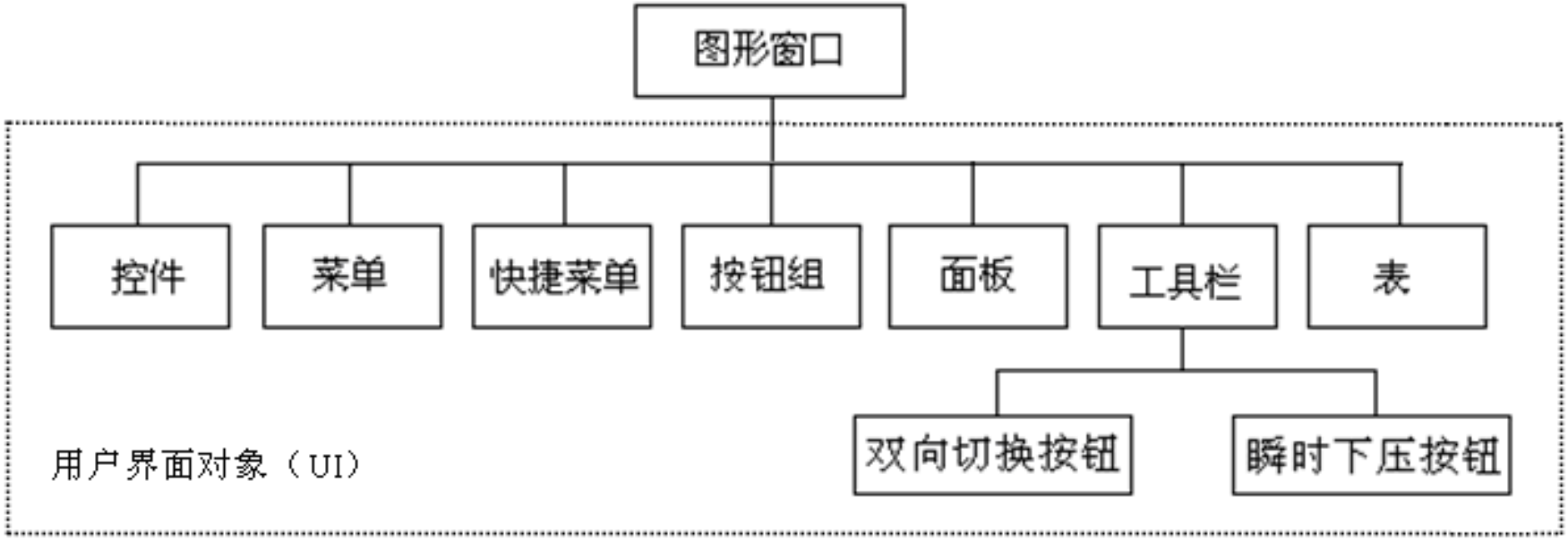
用户界面对象



用户界面对象

每一个图形用户界面(GUI)都是由若干个不同的用户界面对象（UI）组成的。

用户界面对象的层次结构如图所示：



回顾：图形对象及其句柄

- 图形对象

MATLAB的图形对象包括:

计算机屏幕、图形窗口、坐标轴、用户菜单、用户控件、曲线、曲面、文本、图像、光源、区域块和方框等。

系统将每一个对象按树型结构组织起来。

回顾：图形对象及其句柄

●图形对象句柄

MATLAB在创建每一个图形对象时，都为该对象分配唯一的一个值，称其为图形对象**句柄(Handle)**。

句柄是图形对象的唯一标识符，不同对象的句柄不可能重复和混淆。

计算机屏幕作为根对象，其句柄值为**0**；

图形窗口对象的句柄值为一正整数；

其他图形对象的句柄为浮点数。

回顾：图形对象及其句柄

- 图形对象句柄的获取：
  - 创建时获取
  - L=plot(1:10,rand(1,10),'r--');
  - 通过函数获取
  - **gcf**：获取当前图形窗口的句柄
  - **gca**：获取当前坐标轴的句柄
  - **gco**：获取当前被选中的图形对象的句柄
  - **findobj**：按照指定的属性来获取图形对像的句柄

```
H=findobj(gca,'LineStyle','--');
```

回顾：图形对象及其句柄

- 图形对象属性
- 属性名与属性值

如：属性名：**FontName**，属性值：**Arial**

- get：获得对象属性的值

```
» yVals=get(L,'YData');
```

- set：设置对象属性的值

```
» A=gca;
» L=plot(1:10,rand(1,10),'r--');
» set(A,'FontName','Arial','XScale','log');
» set(L,'LineWidth',1.5,'Marker','*');
```

回顾：图形对象及其句柄

- 图形窗口对象的建立: figure
  - hf=figure('name','演示图形窗口')
- 关闭图形窗口： close
  - close(hf)：关闭句柄为hf的图形窗口
  - close all：关闭所有的图形窗口
  - clf：清除当前图形窗口的内容，但不关闭窗口
- 坐标轴对象的建立：
  ha=axes('Position',[0.1,0.1,0.7,0.7])

contour(peaks(20))

- 设定当前坐标轴：

axes(ha)

回顾：图形对象及其句柄

例：建立一个图形窗口，绘制正弦图形，获取其曲线颜色，设置曲线颜色与线型

```
f=figure('name','正弦图像','NumberTitle','off');
        %创建图形窗口对像
h=ezplot('sin');     %绘制曲线，并返回句柄
c=get(h, 'color');   %获取颜色属性
set(h, 'color',[1 0 0]);    %设置曲线颜色
set(h, 'LineStyle', '--');    %设置曲线线型
```

8.1 菜单设计

- 建立用户菜单：**uimenu**
- 建立一级菜单项：

  一级菜单项句柄=uimenu(图形窗口句柄，属性名1，属性值1，属性名2，属性值2，…)
- 建立子菜单项：

  子菜单项句柄=uimenu(一级菜单项句柄，属性名1，属性值1，属性名2，属性值2，…)

8.1 菜单设计

```
hf=figure('name','UI
Test','NumberTitle','off','MenuBar','none');
                %建立图形对象
hm1=uimenu(gcf, 'Label', '&Plot');
                %主菜单，gcf=hf
Hm11= uimenu(hm1, 'Label', '&SinWave');
                %子菜单
Hm12= uimenu(hm1, 'Label','&CosWave');
```

8.1 菜单设计：菜单对象常用属性

- 公共属性：

  children、Parent、Tag、Type、UserData、Visible

- 特殊属性：（p218）
  - **Label:菜单项名字**
  - **Accelerator:定义快捷键字母**
  - **Callback: 执行的命令**
  - **Checked: 是否选中：on 或 off**
  - **Enabled: 是否可用：on 或 off**
  - **Position: 菜单的相对位置**
  - **Separator：是否添加分隔线**
    - **on：添加，　off：不添加**
  - 所有的属性名称都可以只写前四个字母（缩写）

8.1 菜单设计：回调函数

Callback:回调函数，决定菜单点击后执行什么命令

```
hf=figure('name','UI Test','NumberTitle','off','MenuBar','none');
                %建立图形对象
hm1=uimenu(gcf, 'Label', '&Plot');
                %主菜单，gcf=hf
Hm11= uimenu(hm1, 'Label', '&SinWave','call','ezplot("sin")');
                %子菜单
Hm12= uimenu(hm1, 'Label','&CosWave','call','ezplot("cos")');
```

Note：单引号中要使用单引号，用两个"

8.1 菜单设计：回调函数

若命令语句较长，可将多个语句连接成一个字符串

[‘command 1’，‘command 2’….]

```
uimenu(hplot,'Label','Sine Wave','call',…
                ['t=linspace(0,2*pi,100);','plot(t,sin(t));']);
```

也可以调用自定义脚本或函数

```
uimenu(hplot,'Label','Sine Wave','call',’plotsin');
```

函数文件：

```
function plotsin
    t=linspace(0,2*pi,100);
    plot(t,sin(t));
end
```

8.1 菜单设计：快捷菜单

- **快捷菜单：**鼠标右键单击某对象时屏幕上弹出的菜单
  - **hcm=uicontextmenu;**
    - ➢创建快捷菜单，返回句柄hcm
  - **set(hl,'UIContextMenu',hcm);**
    - ➢设置图形对象hl的快捷菜单为hcm

步骤：

(1)利用uicontextmenu函数建立快捷菜单。

(2)利用uimenu函数为快捷菜单建立菜单项。

(3)利用set函数将该快捷菜单和某图形对象联系起来。

8.1 菜单设计：快捷菜单

```
hl=ezplot('sin(x)');
hc=uicontextmenu;
hlc=uimenu(hc,'Label','颜色');
uimenu(hlc,'Label','红色','call','set(hl,"color","r")');
uimenu(hlc,'Label','蓝色','call','set(hl,"color","b")');
set(hl,'UIContextMenu',hc);
```

8.2 对话框设计

- 对话框的控件
  - 按钮（Push Button）
  - 双位(开关)按钮（Toggle Button）
  - 单选按钮（Radio Button）
  - 复选框（Check Box）
  - 列表框（List Box）
  - 弹出框（Popup Menu）
  - 编辑框（Edit Box）
  - 滑动条（Slider）
  - 静态文本（Static Text）
  - 边框（Frame）

8.2 对话框设计

- 建立控件对象：uicontrol

  对象句柄=uicontrol(图形窗口句柄，属性名1，属性值1，属性名2，属性值2，…)

- 控件对象的属性
  - **Style:控件类型**
    - **push、toggle、radio、check、list、popup、edit、text、slider、frame**
  - **Position:** 控件位置：**[x y w h]**
  - **Callback:** 响应的命令
  - **String:** 控件对象的说明文字
  - **max/min:** p222

8.2 对话框设计

```
figure('name','UI Test','NumberTitle','off','MenuBar','none');
                %建立图形对象
pbplot=uicontrol(gcf,'style','push','position',[20,5,100,30], 'string','plot sine wave', 'callback', 'ezplot(''sin(x)'')');
                %建立按钮对象
ptgrid=uicontrol(gcf,'style','toggle','position',[130,5,100,30],  'string','grid','callback','grid');
                %建立开关按钮对象
```

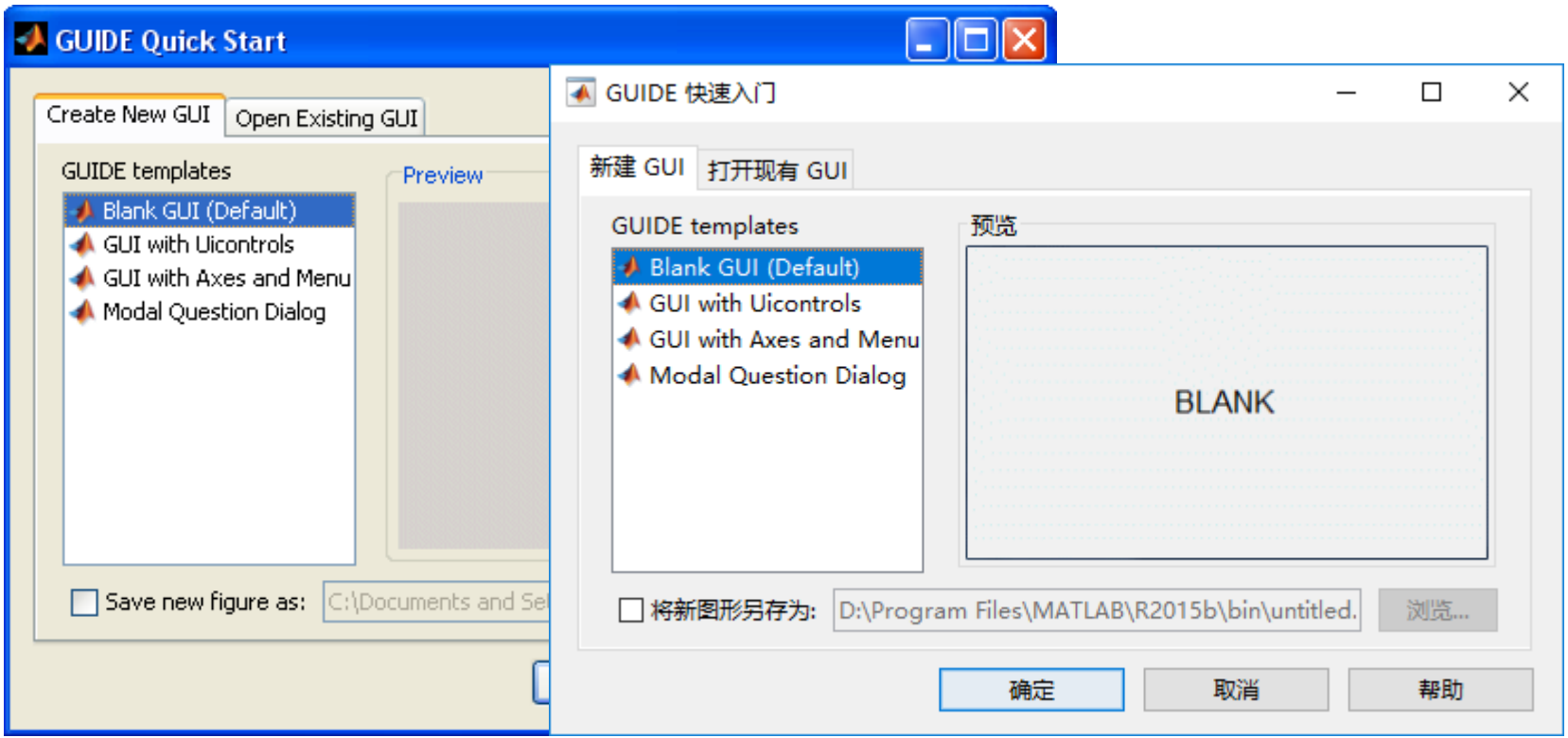
8.2 对话框设计

控件展示：

```
figure('name','UI Test','NumberTitle','off','MenuBar','none');
uicontrol(gcf,'style','push','position',[20,5,100,30],'string','push button');
uicontrol(gcf,'style','toggle','position',[130,5,100,30],'string','toggle button');
uicontrol(gcf,'style','check box','position',[240,5,100,30],'string','check box');
uicontrol(gcf,'style','radio','position',[370,5,100,30],'string','radio button');
uicontrol(gcf,'style','edit','position',[20,45,100,30],'string','edit');
uicontrol(gcf,'style','text','position',[130,45,100,30],'string','text');
uicontrol(gcf,'style','popup','position',[240,45,100,30],'string','popup');
uicontrol(gcf,'style','slider','position',[370,45,100,30],'string','slider');
uicontrol(gcf,'style','frame','position',[370,95,100,30],'string','frame');
```

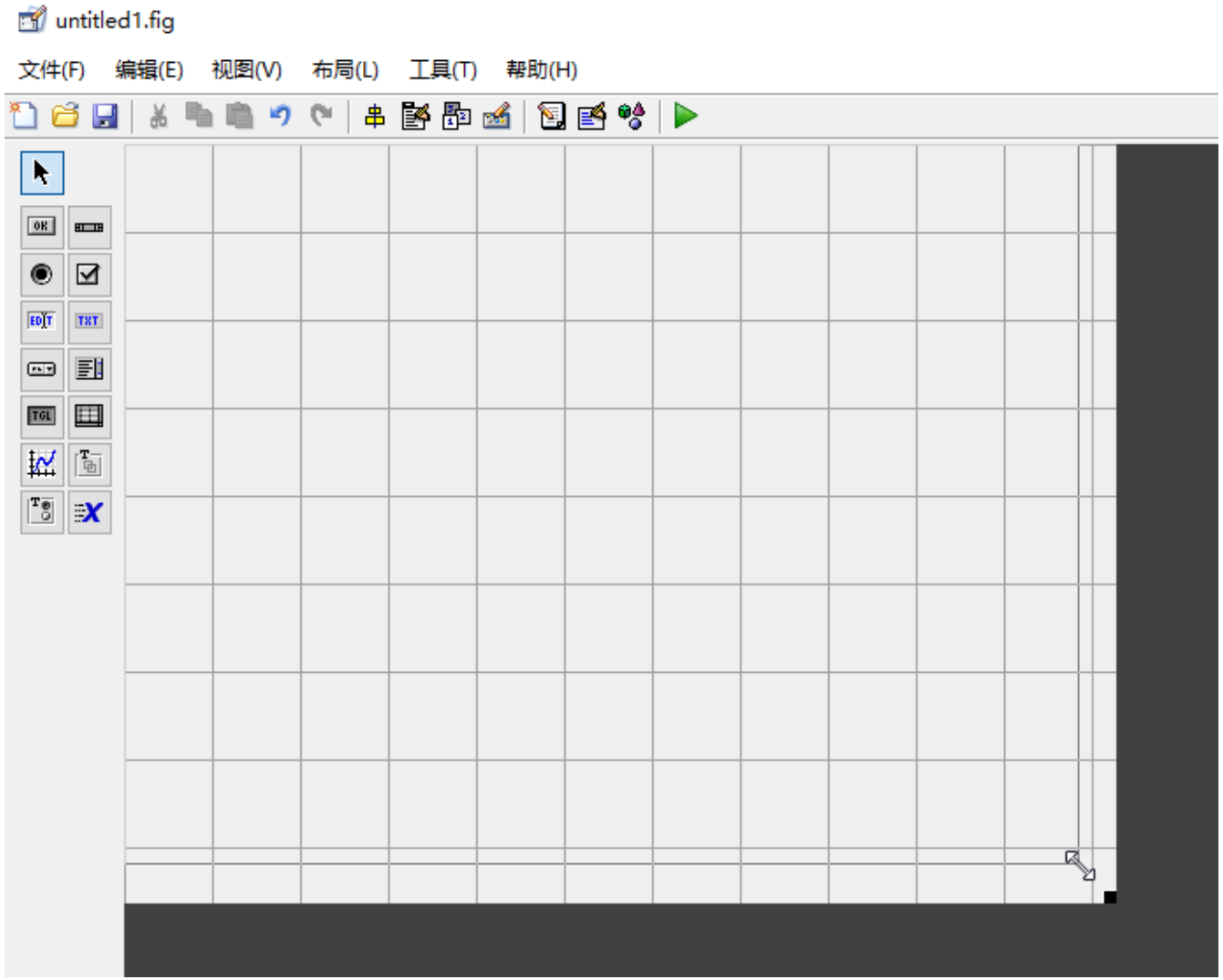
8.3 可视化图形用户界面设计

- 使用GUI向导进行可视化用户界面设计
- 菜单file->new->GUI或在命令窗口输入：guide
- 选择Blank GUI新建一个GUI程序
  若要打开原来的文件，选Open Existing GUI



8.3 可视化图形用户界面设计

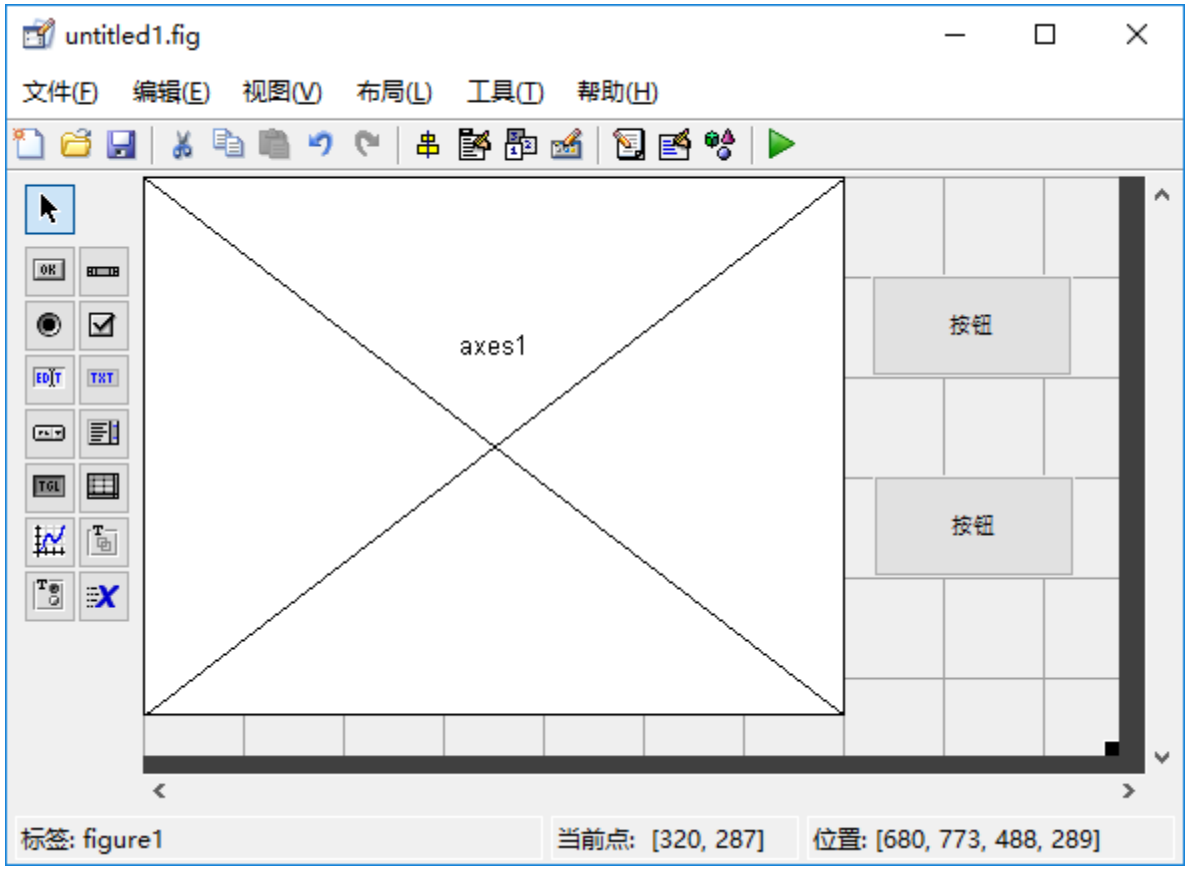
- 可拖曳改变设计区的大小



8.3 可视化图形用户界面设计

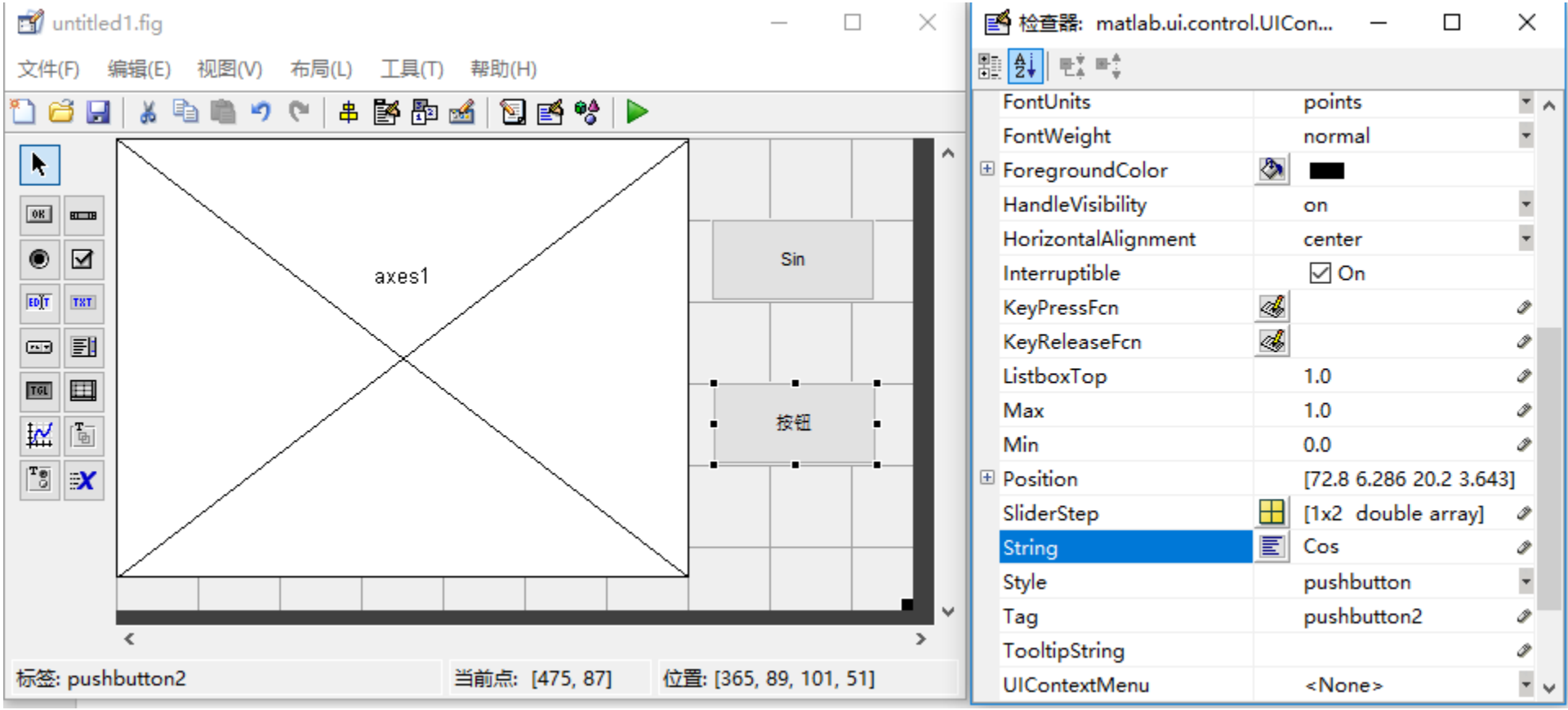
- 将控件拖曳到设计区域

  拖一个axes和两个pushbutton到窗口上



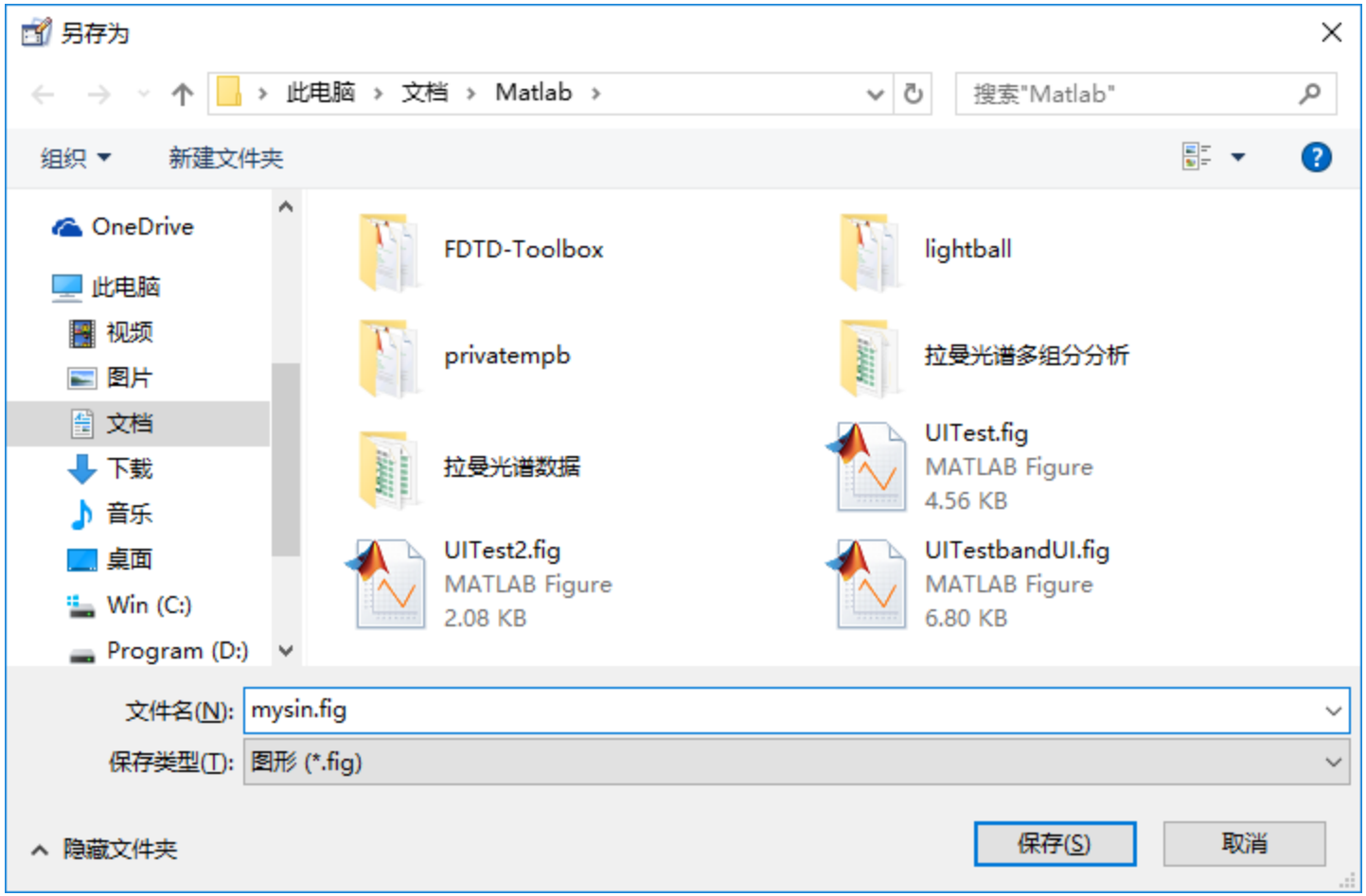
8.3 可视化图形用户界面设计工具

- 设置控件属性：双击控件打开对象属性查看器(Inspector)，修改对象的属性
  - 分别双击pushbutton1与pushbutton2，在弹出的属性设置窗口中，设置其string属性值分别为：Sin和Cos
  - 设置callback属性分别为：ezplot('sin')和ezplot('cos')



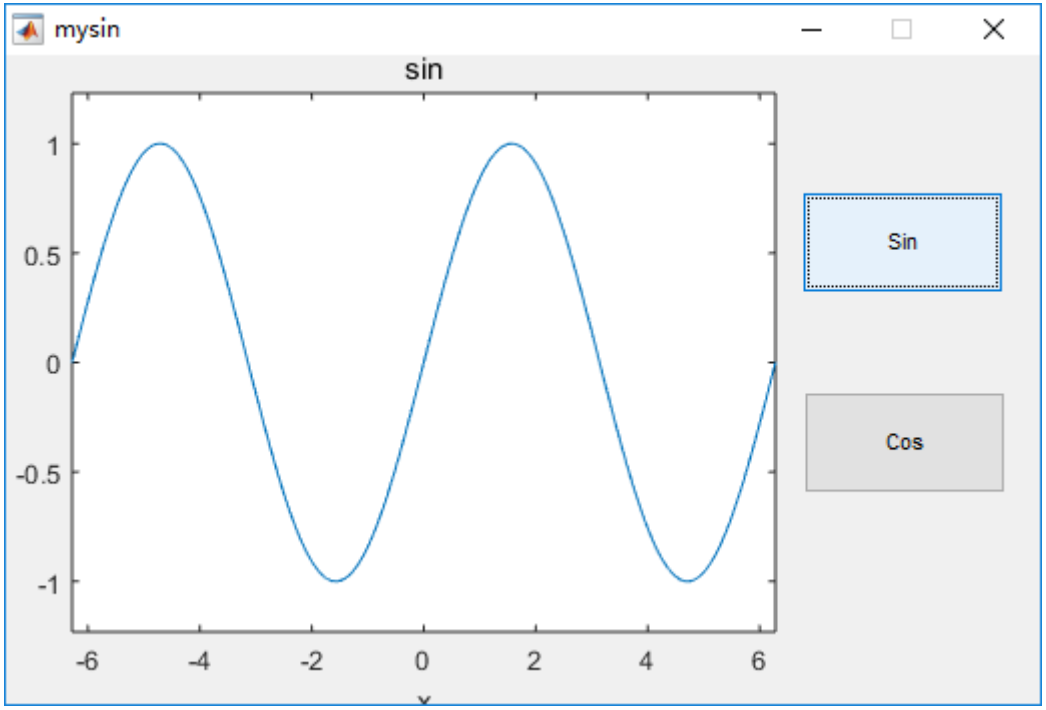
8.3 可视化图形用户界面设计

- 保存**GUI**为**.fig**文件**mysin.m**，同时还会产生一个对应的**.m**文件**mysin.m**



8.3 可视化图形用户界面设计

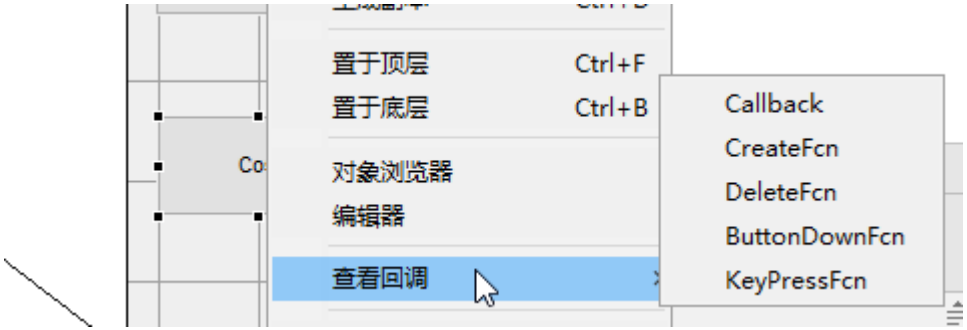
- 运行：按**F5**或点击运行按钮运行

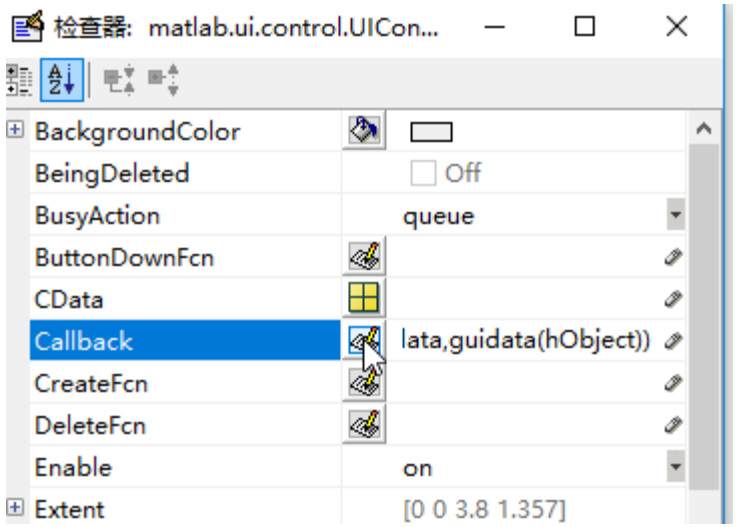


8.3 可视化图形用户界面设计

- 若程序比较复杂，可以这样解决：
  - 法一：把程序保存为函数，在callback中调用该函数
  - 法二：直接在回调程序中写程序

在pushbutton上点右键，查看回调→callback，或在检查器中点击Callback属性后的编辑图标，此时跳转到回调函数的编辑界面，在后面写上相应程序即可。





8.3 可视化图形用户界面设计

- 编写程序：

```
function pushbutton1_Callback(hObject, eventdata, handles)
x=-pi:pi/100:pi;
plot(x,sin(x))

function pushbutton2_Callback(hObject, eventdata, handles)
x=-pi:pi/100:pi;
plot(x,cos(x))
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
x=-pi:pi/100:pi;
plot(x,sin(x))
```

8.3 可视化图形用户界面设计

- 五个用户界面设计工具：

➢菜单编辑器(Menu Editor)：

创建、设计、修改下拉式菜单和快捷菜单

➢对象属性查看器(Property Inspector)：

查看及修改对象属性值

➢位置调整工具(Alignment Tool)：

对多个对象的位置进行左右上下调整

➢对象浏览器(Object Browser)：

查看当前设计阶段的各个句柄图形对象

➢Tab顺序编辑器(Tab Order Editor)：

设置当按下Tab键时，对象被选中的先后顺序

8.3 可视化图形用户界面设计

●菜单编辑器

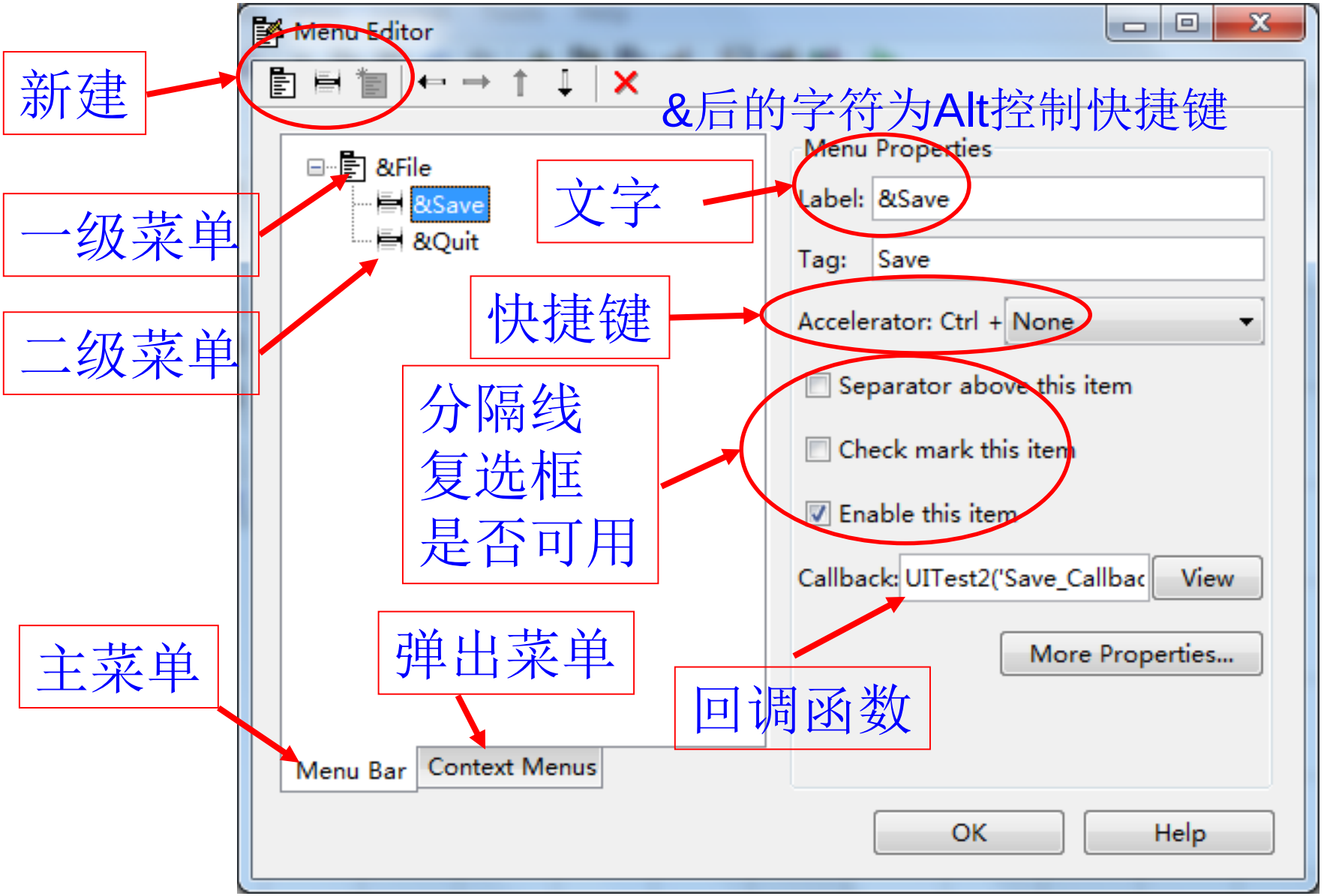
从GUI设计窗口的工具栏上选择Menu Editor命令按钮，或者选择Tools菜单下的Menu Editor子菜单

菜单编辑器的左下角有两个按钮:

选择第一个按钮，可以创建下拉式菜单。

选择第二个按钮，可以创建弹出菜单。选择它后，菜单编辑器左上角的第三个按钮就会变成可用，单击它就可以创建Context Menu主菜单。

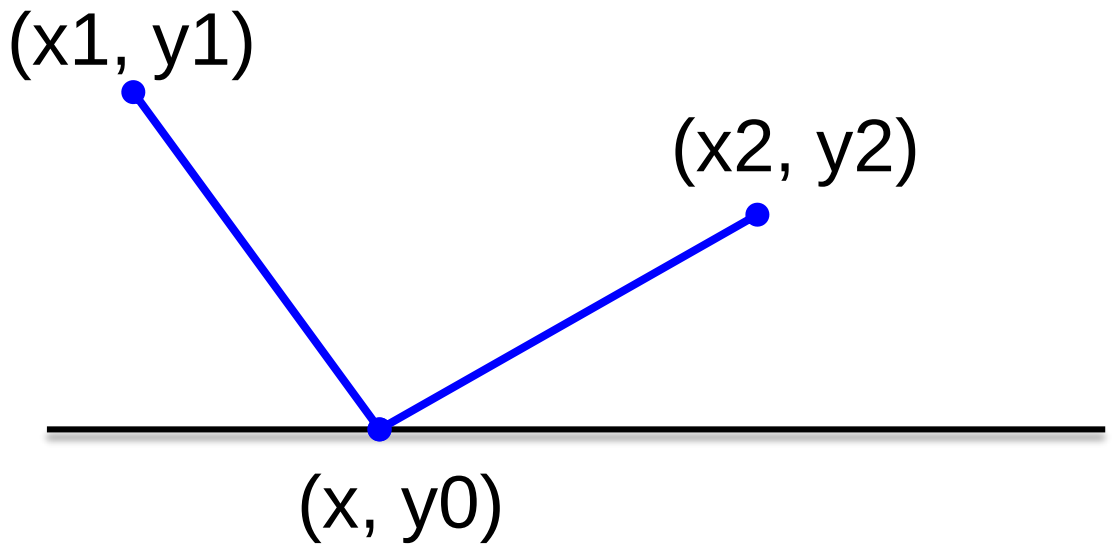
8.3 可视化图形用户界面设计



8.3 可视化图形用户界面设计

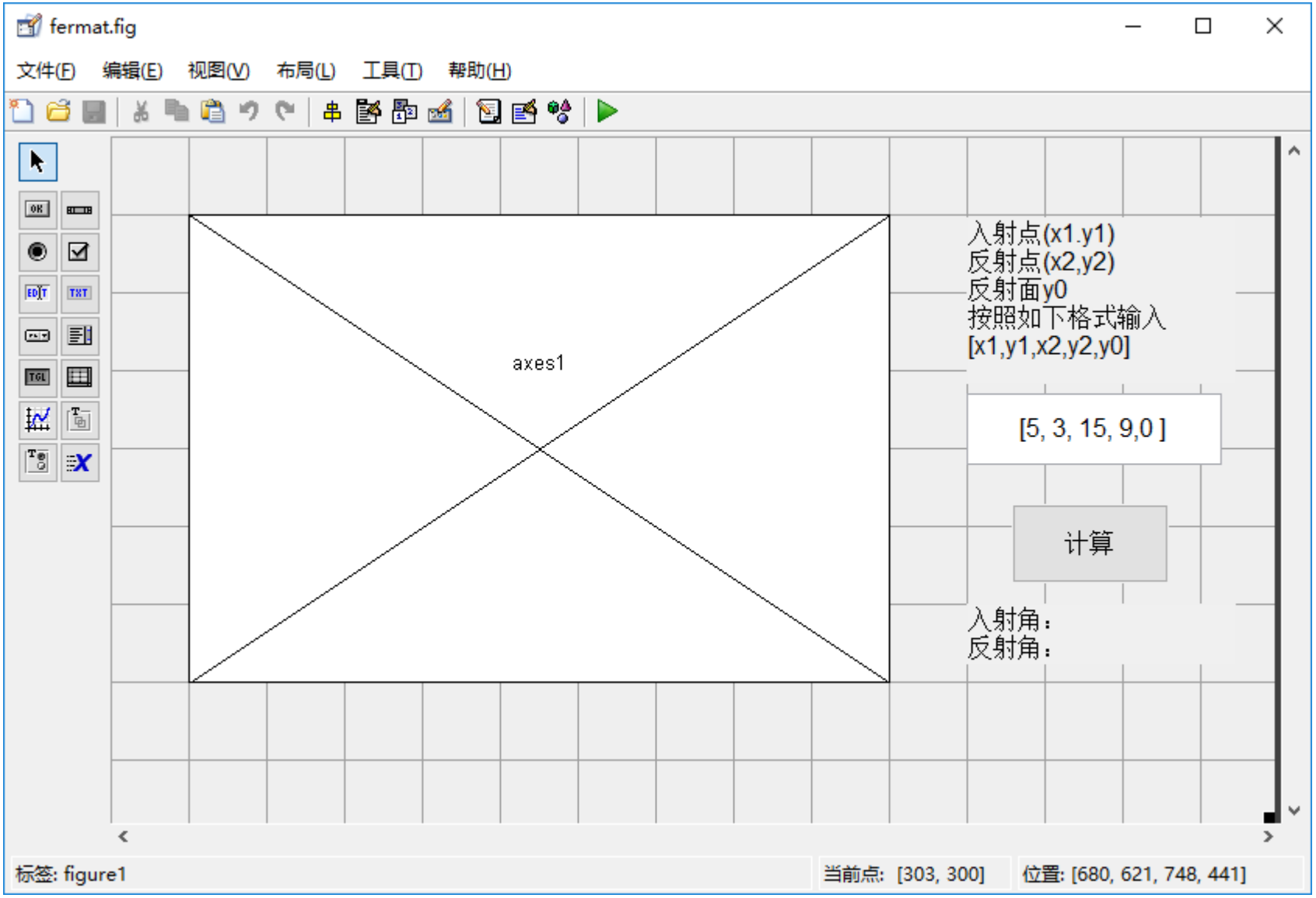
例：基于费马原理使用数值方法证明反射定律

费马原理：光在指定的两点间传播时，实际的光程总是一个极值。在一般情况下，实际光程多是极小值，费马本人最初提出的也是最短光程。



$$s=\sqrt{(x1-x)^2+(y1-y0)^2}+\sqrt{(x2-x)^2+(y2-y0)^2}$$

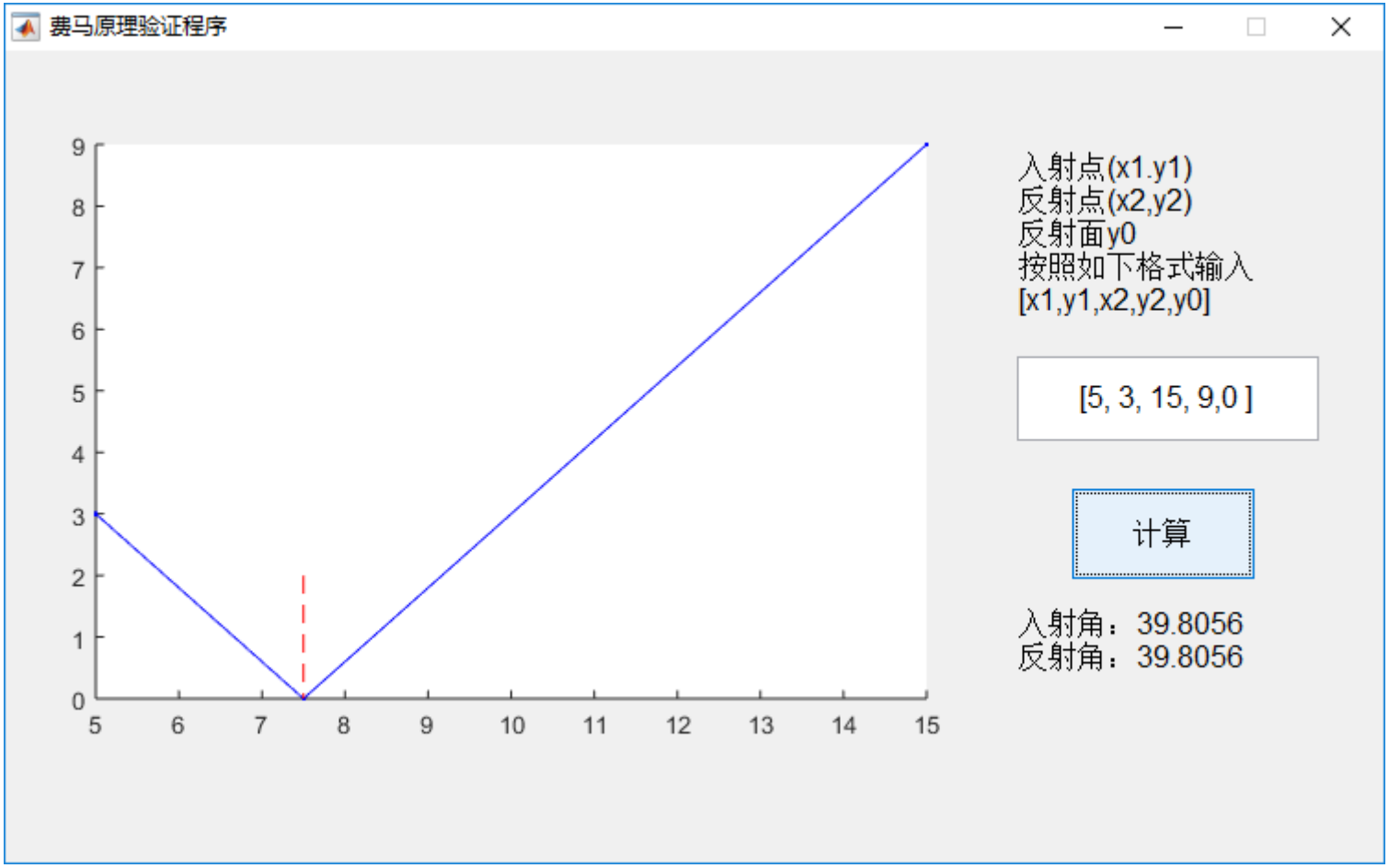
8.3 可视化图形用户界面设计



8.3 可视化图形用户界面设计

```
function pushbutton1_Callback(hObject, eventdata, handles)
xys=get(handles.edit1,'string');          %获得输入
xy=eval(xys');                            %转换为数值向量
x1=xy(1);y1=xy(2);x2=xy(3);y2=xy(4);y0=xy(5);
gcs=@(x) sqrt((x1-x)^2+(y1-y0)^2)+sqrt((x2-x)^2+(y2-y0)^2);
                                          %定义光程匿名函数
[x,fmin]=fminbnd(gcs,x1,x2);              %求最小值
alphain=atan((x-x1)/(y1-y0))*180/pi;      %求入射角
alphaout=atan((x2-x)/(y2-y0))*180/pi;     %求反射角
hold on;
plot([x1 x x2],[y1 y0 y2],'b.-');         %作光线图
plot([x x],[y0+2 y0],'r--');              %作法线
hold off;
set(handles.text3,'string',['入射角：',num2str(alphain); '反射角：
',num2str(alphaout)]);                    %显示结果
```

8.3 可视化图形用户界面设计



8.3 可视化图形用户界面设计

例： 使用可视化界面设计完成例8.4（课后练习）

步骤：

1、file->new->GUI

2、插入axes、pushbutton、edit控件

2、edit控件设置为多行（max-min>1)

3、双击pushbutton控件，进入属性编辑器，在callback属性上点击编辑按钮，在打开的编辑器中相应位置输入代码：

```
str=get(handles.edit1,'string');
eval(str‘);  %注意要转置
```

4、保存执行

8.3 可视化图形用户界面设计