

## 一、 函数编写练习：

结果如下

```
>> ZhiShu
请输入数字=
输入错误，请输入正整数
输入错误，请输入正整数
>> ZhiShu
请输入数字=5
此数为质数
>> ZhiShu
请输入数字=8
此数不是质数，可以被2整除
此数为偶数，其平方根为2.8284
>> ZhiShu
请输入数字=15
此数不是质数，可以被3整除
>> ZhiShu
请输入数字=aaa
输入错误，请输入正整数
输入错误，请输入正整数
```

函数 ZhiShu.m 代码如下

```
function ZhiShu
i=2;
try
    X=input('请输入数字=' , 's');
    X=str2num(X);
    if isempty(X)    %判断是否输入字母
        disp('输入错误，请输入正整数')
    else
        if X<0 || mod(X, 1) ~= 0
            disp('输入错误，请输入正整数') %判断是否输入负数或非整数
        elseif X==0
            disp('0不是质数')
        elseif X==1
            disp('1不是质数')
        elseif X==2
            disp('此数为质数')
            Bool=1;
        else
            while i<X
                if mod(X, i)==0
                    disp(['此数不是质数，可以被', num2str(i), '整除'])
                    break;
                else
                    i=i+1;
                if X==i
```

```

        disp('此数为质数')
    end
end
end
end

EvenNum(X);
catch
    disp('输入错误, 请输入正整数')
end

function EvenNum(X)
if X<0 || mod(X, 1) ~= 0
    disp('输入错误, 请输入正整数') %判断是否输入负数或非整数
elseif mod(X, 2)==0
    disp(['此数为偶数, 其平方根为', num2str(sqrt(X))])
end

```

## 二、 流程控制语句练习:

结果如下

PrimeNum =

1 至 13 列

2	3	5	7	11	13	17	19	23	29	31	37	41
---	---	---	---	----	----	----	----	----	----	----	----	----

14 至 25 列

43	47	53	59	61	67	71	73	79	83	89	97
----	----	----	----	----	----	----	----	----	----	----	----

函数 ZhiShu.m 代码如下

```

function Bool=ZhiShu(X)
Bool=0;
% function ZhiShu
i=2;
try
%     X=input('请输入数字=','s');
%     X=str2num(X);
%     if isempty(X)    %判断是否输入字母
%         disp('输入错误, 请输入正整数')
%     else
%         if X<0 || mod(X, 1) ~= 0
%             disp('输入错误, 请输入正整数') %判断是否输入负数或非整数
%         elseif X==0
%
```

```

        disp('0不是质数')
elseif X==1
    disp('1不是质数')
elseif X==2
    disp('此数为质数')
    Bool=1;
else
    while i<X
        if mod(X, i)==0
            disp(['此数不是质数, 可以被', num2str(i), '整除'])
            break;
        else
            i=i+1;
            if X==i
                disp('此数为质数')
                Bool=1;
            end
        end
    end
end
%     end

EvenNum(X);
catch
    disp('输入错误, 请输入正整数')
end

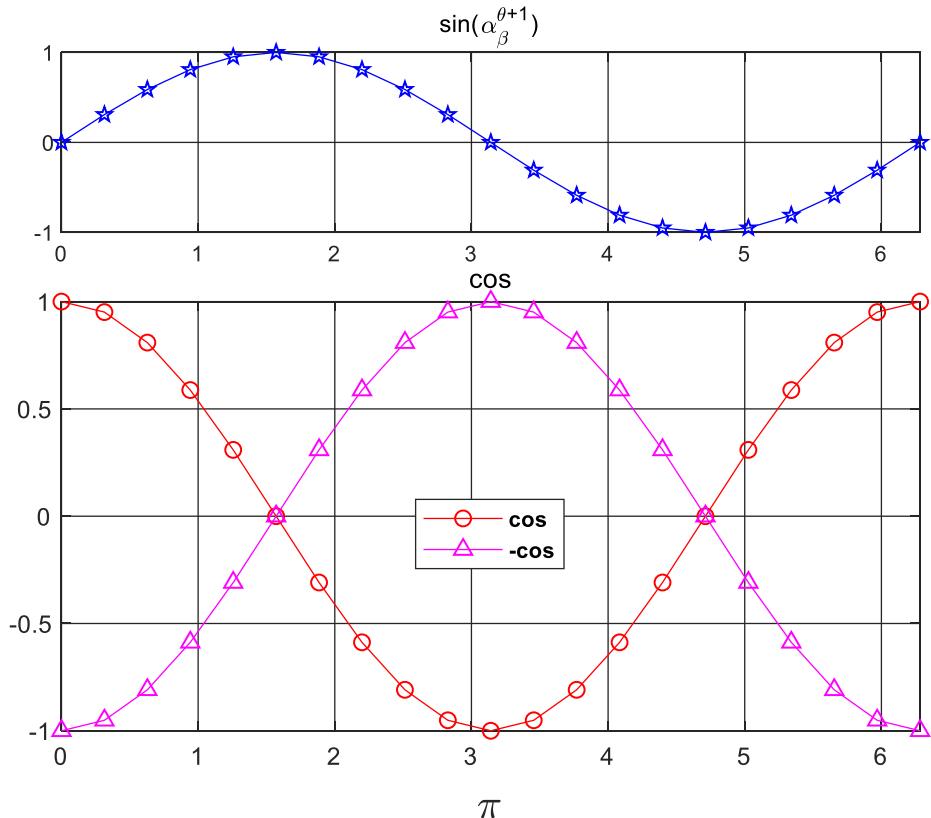
function EvenNum(X)
if X<0 || mod(X, 1) ~= 0
    disp('输入错误, 请输入正整数') %判断是否输入负数或非整数
elseif mod(X, 2)==0
    disp(['此数为偶数, 其平方根为', num2str(sqrt(X))])
end
脚本文件ZhiShu_0_100.m代码如下
clear, clc
index=1;
for i=0:100
    if ZhiShu(i)      %函数返回值为1, 才给数组赋值
        PrimeNum(index)=i;
        index=index+1;
    end
end

```

### 三、画图练习

## 1. Plot 命令的练习

结果如下：



代码如下：

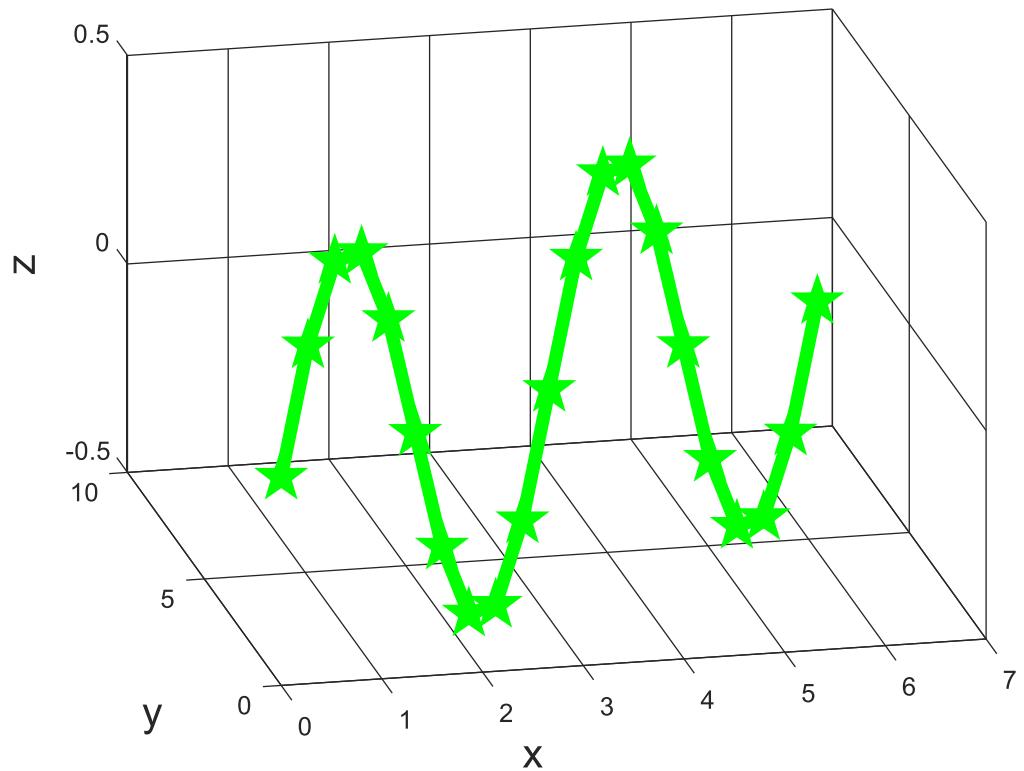
```
%% plot命令练习
t=0:0.1*pi:2*pi;

subplot(3, 1, 1); %画3行1列第一幅子图
plot(t, sin(t), 'b-p');
grid on; axis tight; hold on; %显示轴线、坐标区适应、保持
title('sin(\alpha_\beta^{\{\theta+1\}})');
ax=gca; %必须将gca句柄赋给临时句柄ax
ax.GridAlpha=1; %直接gca.GridAlpha=1无法更改透明度

subplot(3, 1, [2, 3]); %画3行1列第二、三幅子图
plot(t, cos(t), 'r-o');
plot(t, -cos(t), 'm-^');
grid on; axis tight; hold on;
legend({'cos', '-cos'}, 'Location', 'best', 'FontWeight', 'bold');
xlabel('\pi', 'fontsize', 16);
title('cos')
ax=gca;
ax.GridAlpha=1;
```

## 2. 三维曲线绘图练习

结果如下：

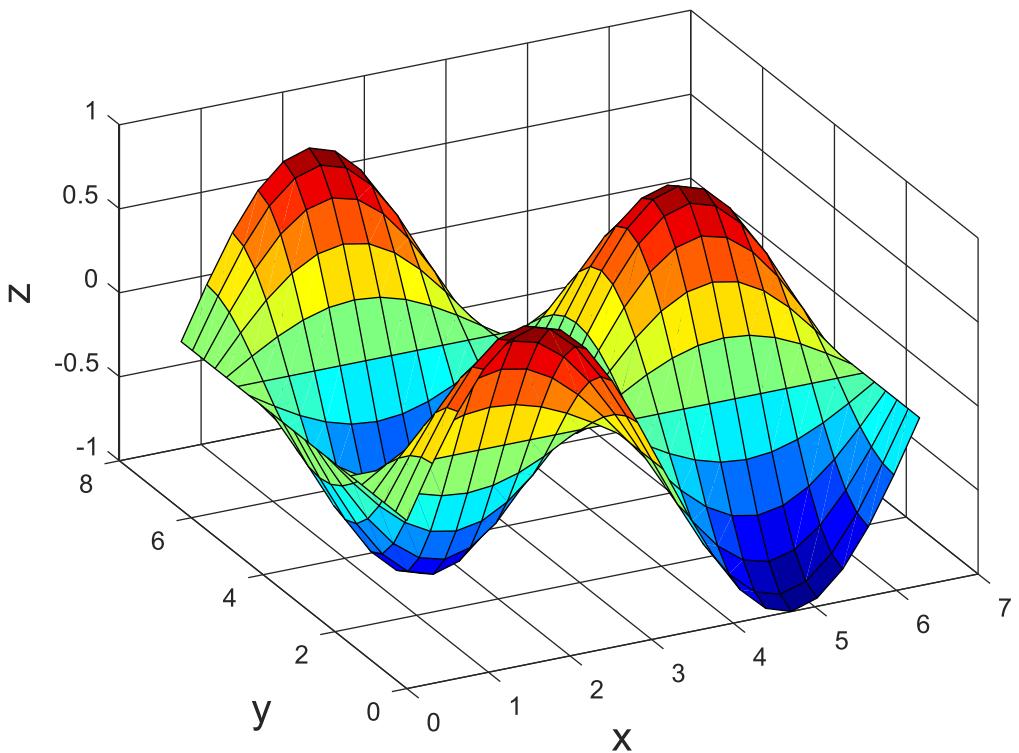


代码如下：

```
%% 三维曲线绘图练习
x=0:0.1*pi:2*pi;      y=0:0.1*pi:2*pi;
Z=sin(x).*cos(y);
plot3(x,y,Z, 'g-p', 'LineWidth', 5);
grid on; hold on;
axis ([0 7 0 10 -0.5 0.5]);
xlabel('x', 'fontsize', 16);
ylabel('y', 'fontsize', 16);
zlabel('z', 'fontsize', 16);
ax=gca;
ax.GridAlpha=1;
```

## 3. 三维曲面绘图

结果如下：

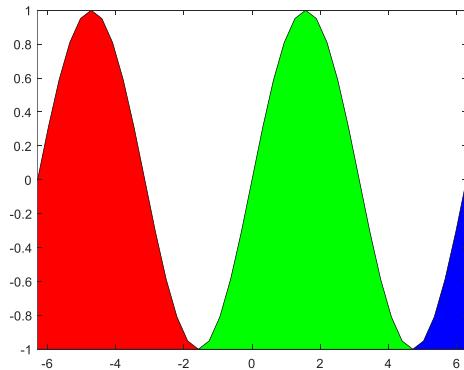
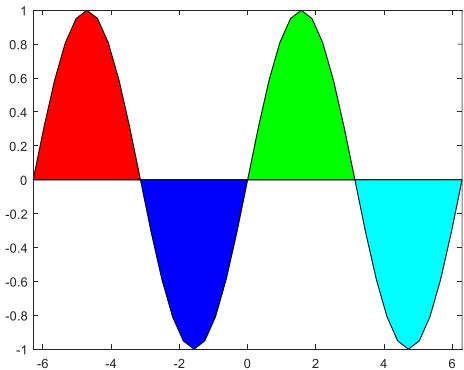


代码如下：

```
%% 三维曲面绘图
x=0:0.1*pi:2*pi;      y=(0:0.1*pi:2*pi)';
Z=cos(y)*sin(x);        %结果为矩阵，不用再meshgrid
surf(x,y,Z);
grid on; hold on;
colormap jet;
axis ([0 7 0 8 -1 1]);
xlabel('x', 'fontsize', 16);
ylabel('y', 'fontsize', 16);
zlabel('z', 'fontsize', 16);
ax=gca;
ax.GridAlpha=1;
```

#### 4. 填充绘图练习

结果如下：

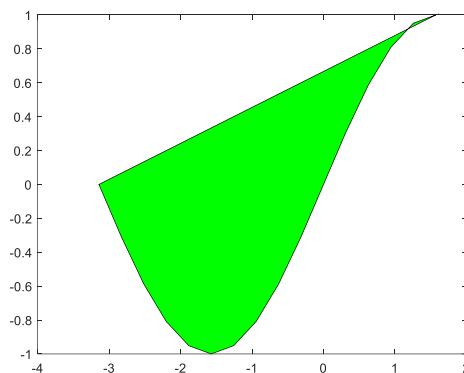
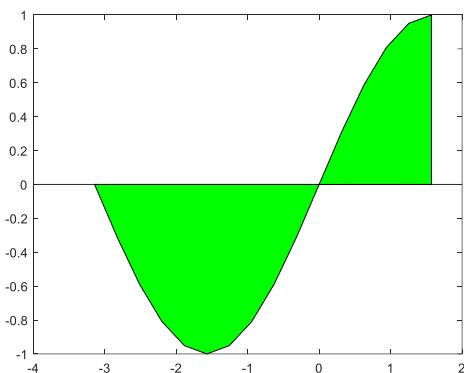


代码如下：

```
%% 填充绘图练习
t=-2*pi:0.1*pi:-2*pi+pi;
area(t,sin(t), 'FaceColor', 'red'); hold on;
t=-1*pi:0.1*pi:-1*pi+pi;
area(t,sin(t), 'FaceColor', 'blue'); hold on;
t=0:0.1*pi:pi;
area(t,sin(t), 'FaceColor', 'green'); hold on;
t=pi:0.1*pi:2*pi;
area(t,sin(t), 'FaceColor', 'cyan'); hold on;
axis tight;
```

```
figure
t=-5/2*pi:0.1*pi:-pi/2;
fill(t,sin(t), 'red'); hold on;
t=-pi/2:0.1*pi:3/2*pi;
fill(t,sin(t), 'green'); hold on;
t=3/2*pi:0.1*pi:7/2*pi;
fill(t,sin(t), 'blue');
axis([-2*pi 2*pi -1 1]);
```

区别举例：



举例代码：

```
%difference example
```

```

figure
t=-pi:0.1*pi:pi/2;
area(t,sin(t), 'FaceColor', 'green');
figure
fill(t,sin(t), 'green');

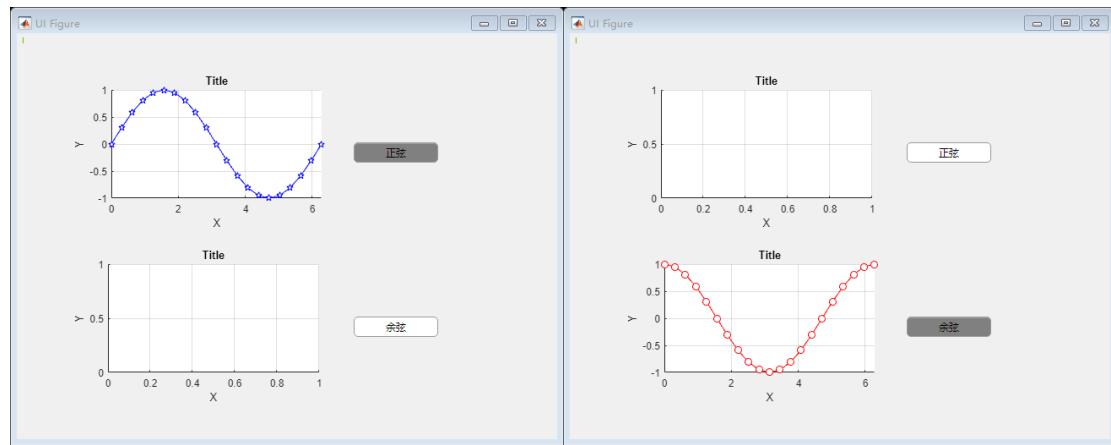
```

结论：area 命令填充区域为定积分表示面积，即  $\int_{t_1}^{t_2} \sin t dt$ ； fill 命令填充区域为二重积分区域，即  $\iint_D f(t, \sin(t)) d\sigma$ 。

#### 四、 GUI 的练习

##### 1. 示例程序

由于我的 MATLAB 版本为 2018a，新的 GUI 界面为 APP 界面，与之前版本略有不同。于是我在此版本下，对本题示例代码做了修改，也实现了按钮的互斥功能。APP 运行结果和回调函数代码如下：



```

% Button pushed function: Button
function ButtonPushed(app, event)
    x=0:0.1*pi:2*pi;
    y=sin(x);
    plot(app.UIAxes,x,y, 'b-p');
    cla(app.UIAxes2);
    set(app.Button, 'BackgroundColor',[0.5,0.5,0.5]);
    set(app.Button_2, 'BackgroundColor',[1,1,1]);
    grid(app.UIAxes, 'on')
end

% Button pushed function: Button_2
function Button_2Pushed(app, event)
    x=0:0.1*pi:2*pi;
    y=cos(x);
    plot(app.UIAxes2,x,y, 'r-o');
    cla(app.UIAxes);

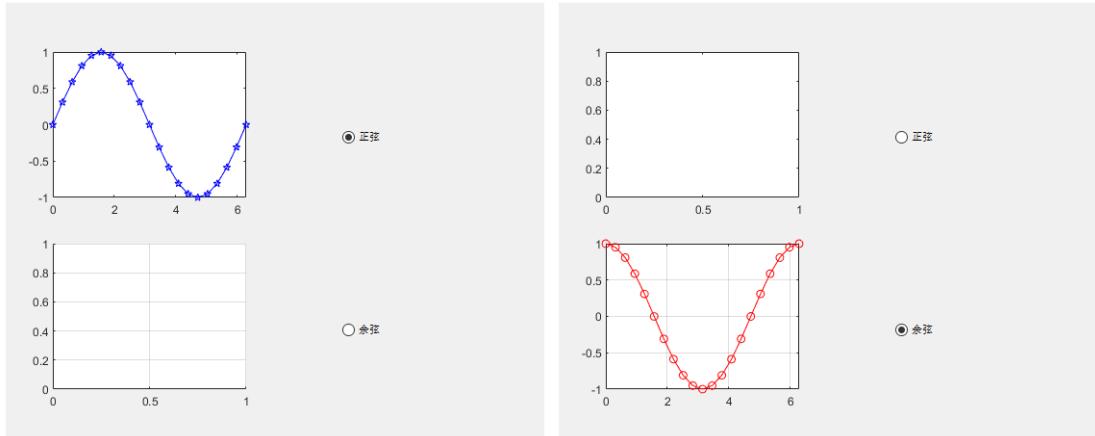
```

```

set(app.Button_2, 'BackgroundColor',[0.5,0.5,0.5]);
set(app.Button, 'BackgroundColor',[1,1,1]);
grid(app.UIAxes2, 'on')
end

```

老的 GUI 运行结果和回调函数代码如下：



```

function radiobutton1_Callback(hObject, eventdata, handles)
x=0:0.1*pi:2*pi;
y=sin(x);
plot(handles.axes1,x,y, 'b-p');
cla(handles.axes2);
set(handles.radiobutton1, 'Value', 1);
set(handles.radiobutton2, 'Value', 0);
grid on

```

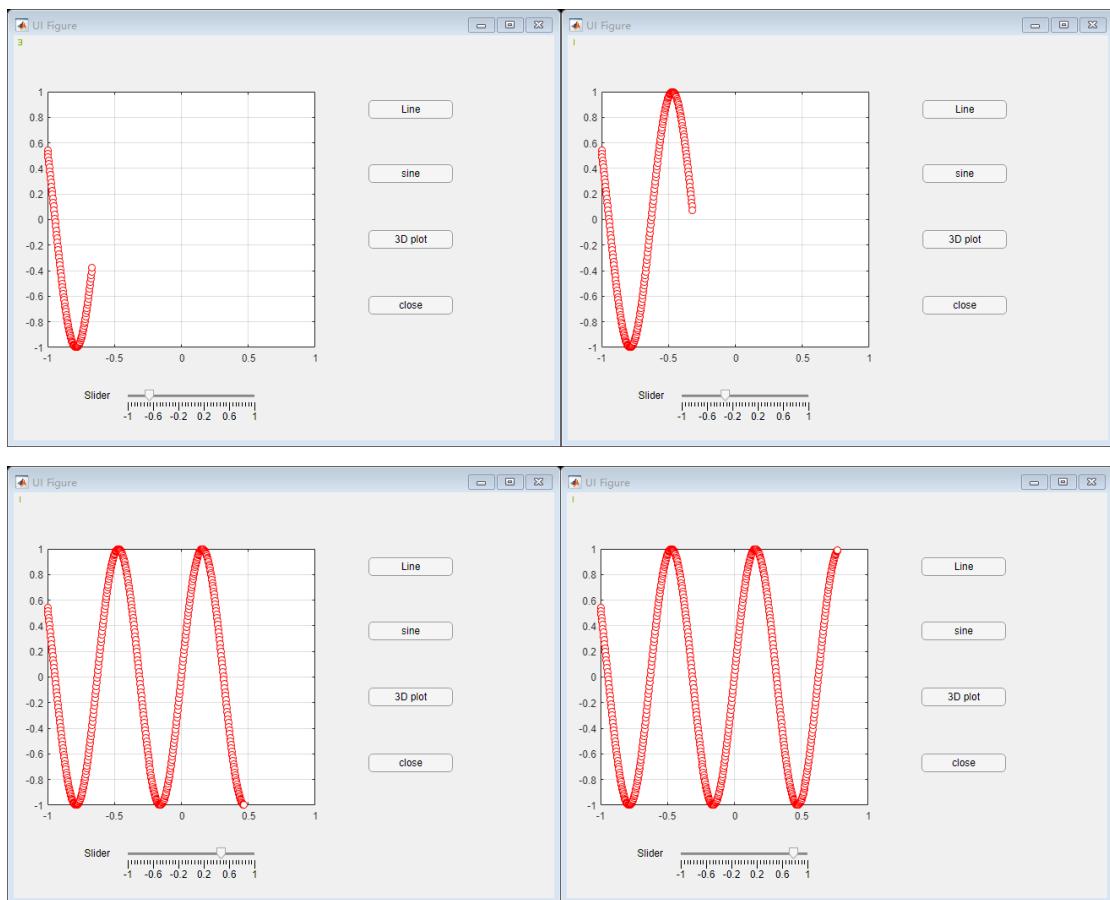
```

function radiobutton2_Callback(hObject, eventdata, handles)
x=0:0.1*pi:2*pi;
y=cos(x);
plot(handles.axes2,x,y, 'r-o');
cla(handles.axes1);
set(handles.radiobutton2, 'Value', 1);
set(handles.radiobutton1, 'Value', 0);
grid on

```

## 2. 滑动条实现功能

APP 运行结果和回调函数代码如下：



```
% Button pushed function: LineButton
function LineButtonPushed(app, event)
    x=-2*pi:0.1*pi:2*pi;
    plot(app.UIAxes,x,x,'r-o')
    axis(app.UIAxes, 'tight');
end

% Button pushed function: sineButton
function sineButtonPushed(app, event)
    x=-2*pi:0.1*pi:2*pi;
    plot(app.UIAxes,x,sin(x), 'g-o')
    axis(app.UIAxes, 'tight');
end

% Button pushed function: DplotButton
function DplotButtonPushed(app, event)
    t = 0:pi/50:10*pi;
    st = sin(t);
    ct = cos(t);
    plot3(app.UIAxes,st,ct,t, 'b-o')
    axis(app.UIAxes, 'tight');
end

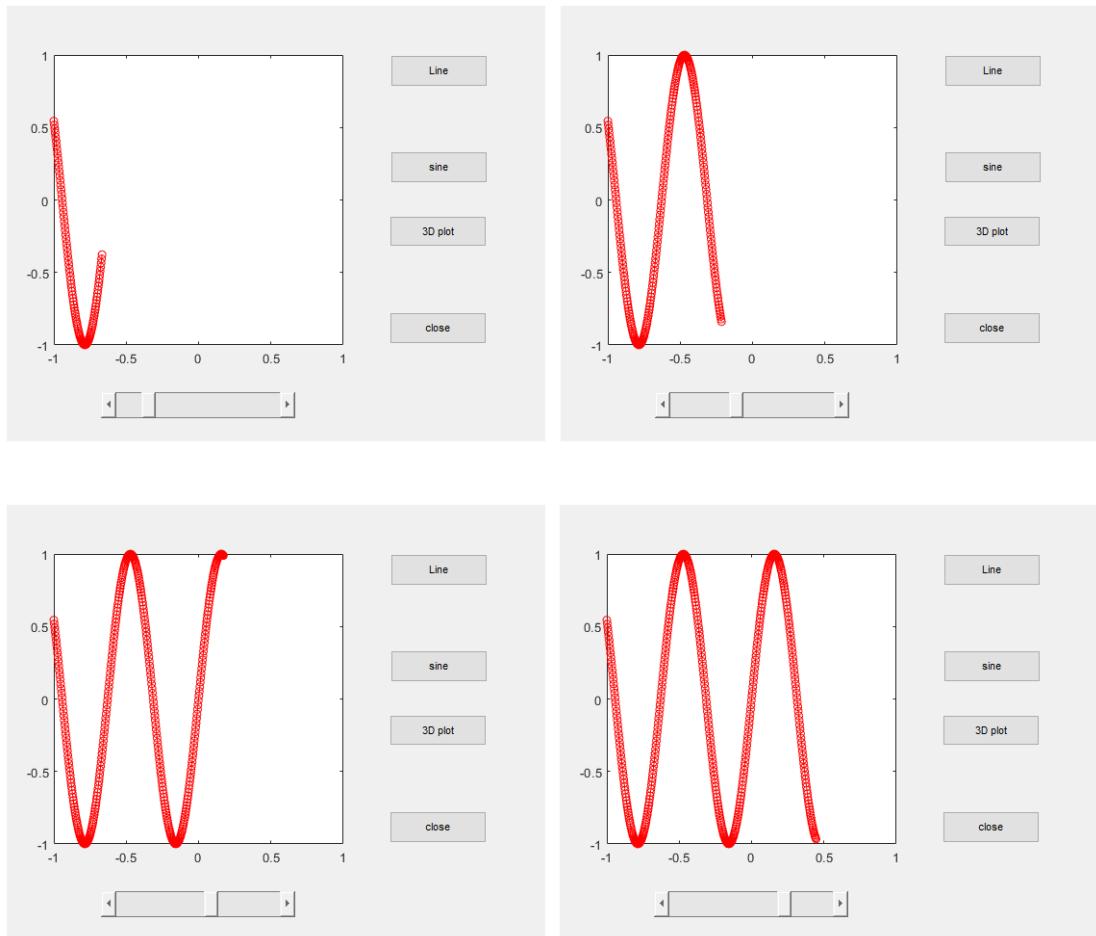
% Button pushed function: closeButton
function closeButtonPushed(app, event)
```

```

    delete(app);
end
% Value changing function: Slider
function SliderValueChanging(app, event)
    changingValue = event.Value;
    x=-1:0.001*pi:changingValue;
    plot(app.UIAxes,x,sin(10*x), 'r-o')
    axis(app.UIAxes,[-1 1 -1 1]);
end

```

老的 GUI 运行结果和回调函数代码如下：



```

function slider2_Callback(hObject, eventdata, handles)
changingValue = get(hObject, 'Value');
x=-1:0.001*pi:changingValue;
plot(x,sin(10*x), 'r-o')
axis([-1 1 -1 1]);
grid on;

function pushbutton1_Callback(hObject, eventdata, handles)
x=-2*pi:0.1*pi:2*pi;

```

```
plot(x, x, 'r-o')
grid on;

function pushbutton2_Callback(hObject, eventdata, handles)
x=-2*pi:0.1*pi:2*pi;
plot(x,sin(x), 'g-o')
grid on;

function pushbutton3_Callback(hObject, eventdata, handles)
t = 0:pi/50:10*pi;
st = sin(t);
ct = cos(t);
plot3(st,ct,t, 'b-o')
grid on;

function pushbutton4_Callback(hObject, eventdata, handles)
delete(old_GUI_exercise);
```