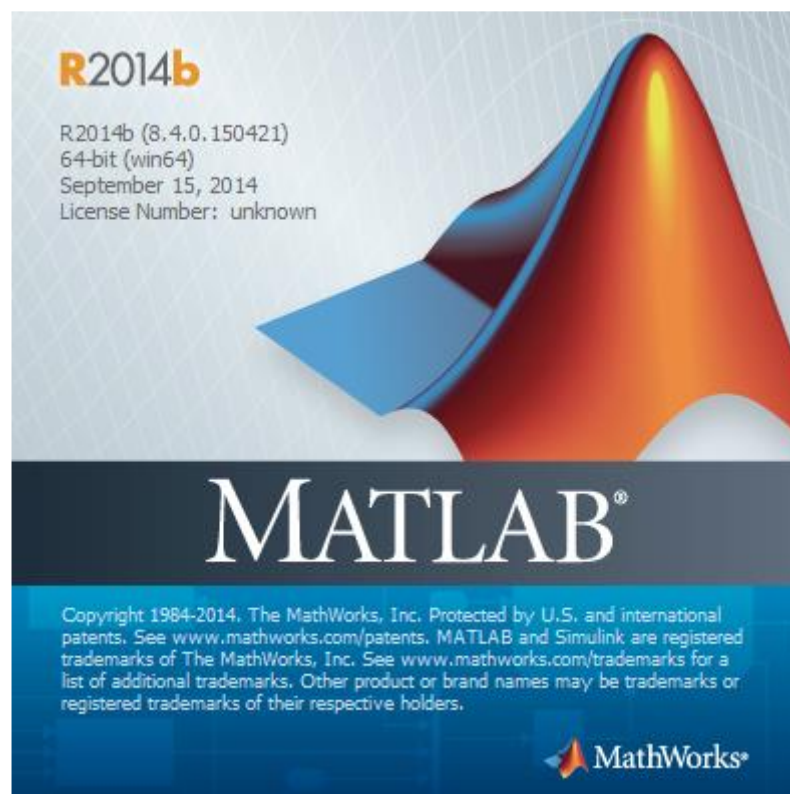


## 第3章b-20200302更新

# MATLAB软件与应用



# 第三章 数据可视化

- 二维数据绘图
- 三维数据绘图
- 图形效果的修饰
- 句柄绘图



## 图形效果的修饰

对图像进行色彩、光照、视角、裁剪、空间变换等方面的处理，使图形更加美观。

# 颜色的修饰

- **colormap(MAP)** —— 色图设定函数，  
MAP为Mx3维色图矩阵,  $M \geq 1$ 。MAP的  
每一行对应一组**RGB**色系设定值。
- 通常曲线图不需要进行色图设置，而曲面图需要（**surf,mesh**等）。
- 色图矩阵**MAP**可依据**RGB**色系任意生成，  
也可以采用系统预设的色图函数。

# 颜色的修饰

常用色图函数：

hsv —— 饱和度色图

gray —— 线性灰度色图

hot —— 暖色色图

cool —— 冷色色图

bone —— 兰色调灰色图

copper —— 铜色色图

pink —— 粉红色图

prism —— 光谱色图

jet —— 饱和度色图II（默认）

flag —— 红、白、蓝交替色图

lines —— 线性色图

autumn —— 红黄色图

# 颜色的修饰

```
x=-4:4; y=x;
```

```
[x,y]=meshgrid(x,y);
```

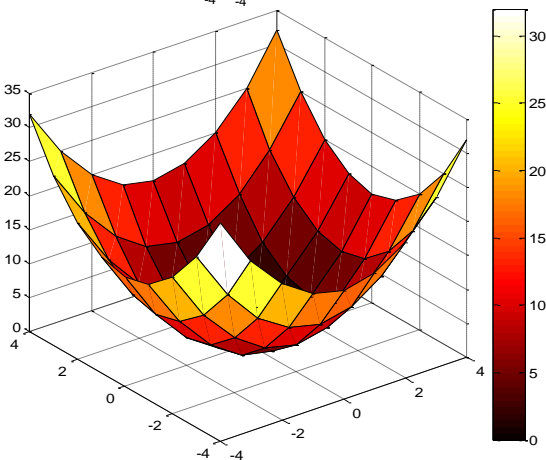
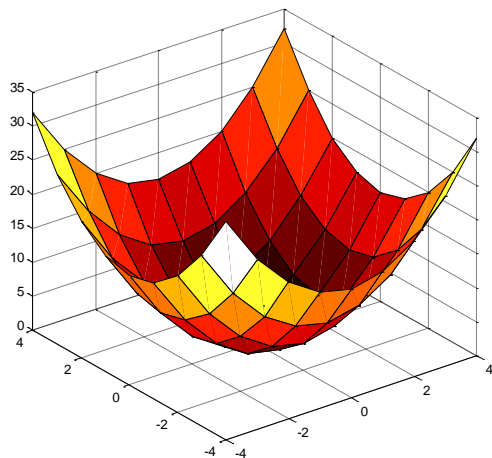
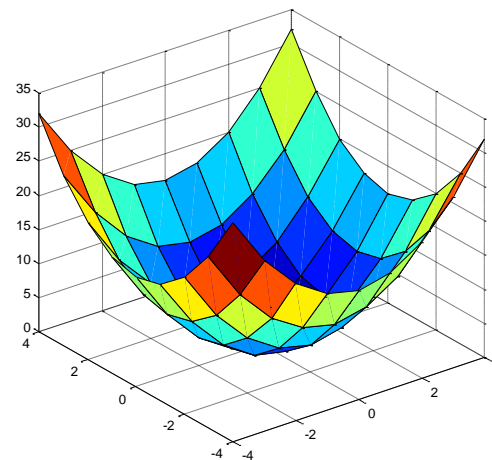
```
z=x.^2+y.^2;
```

```
surf(x,y,z);
```

```
colormap(hot);
```

```
colorbar()
```

%以上colorbar添加颜色标尺, 是最简单的调用格式, 完整的输入参数设置见doc



# 颜色的修饰

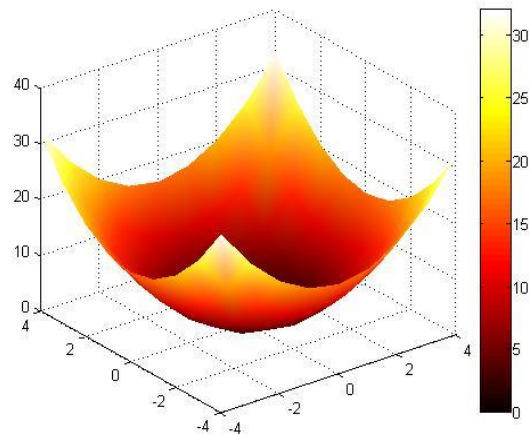
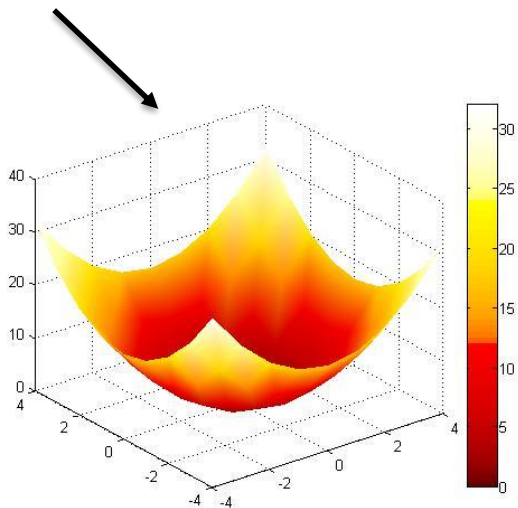
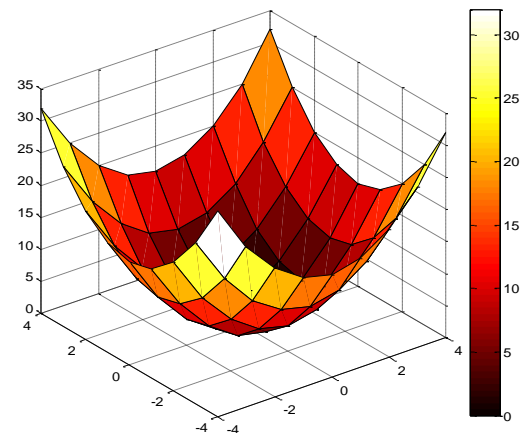
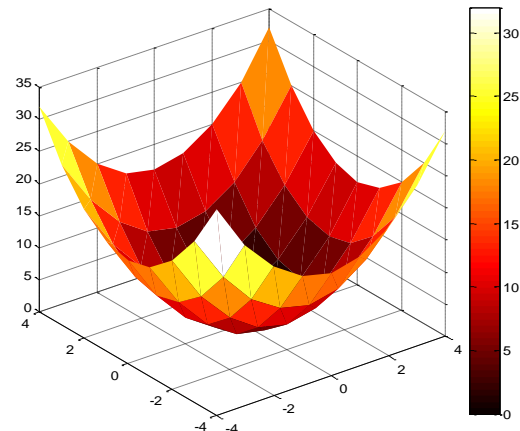
%接上页指令

**shading flat;** %删掉网格线条

**shading faceted;** %添加网格线条(默认)

**shading interp;** %颜色光滑过渡模式

**brighten(0.7);** %设置明暗



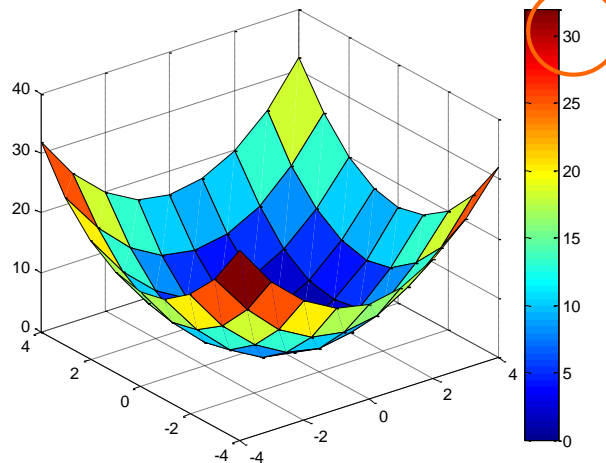
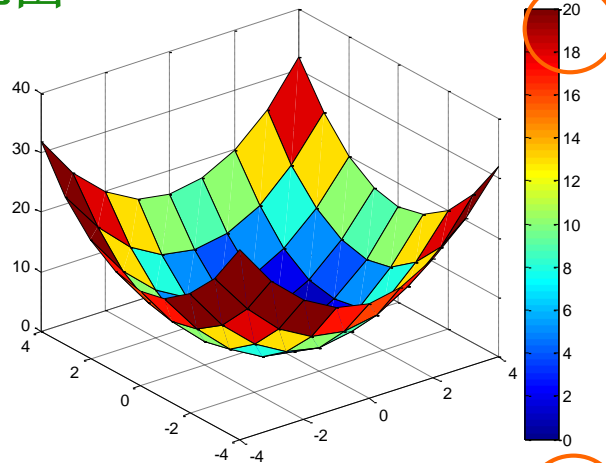
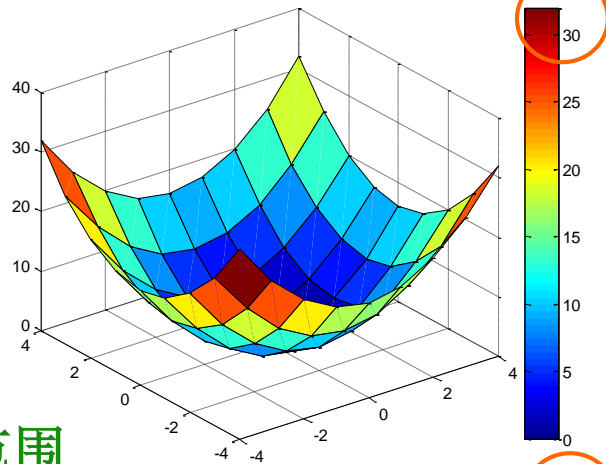
# 颜色的修饰

%接上页指令

**colormap(jet); shading faceted;** →

**caxis([0,20]);** %控制数值与颜色刻度的对应范围

**caxis auto;** %更多用法见doc





# 视角效果

- **view** 设置观察点
- **viewmtx** 视角转换
- **rotate3d** 激活鼠标在图形窗口中控制三维视角
- **hidden on, hidden off** 显示或隐藏重叠在后面的网格（对 mesh 绘制的图片）
- 等等...

# 光照效果

- **light** 设置光源
- **lighting** 设置曲面光源
- **material** 设置光照反应模式
- 等等...

# 裁剪与空间变换

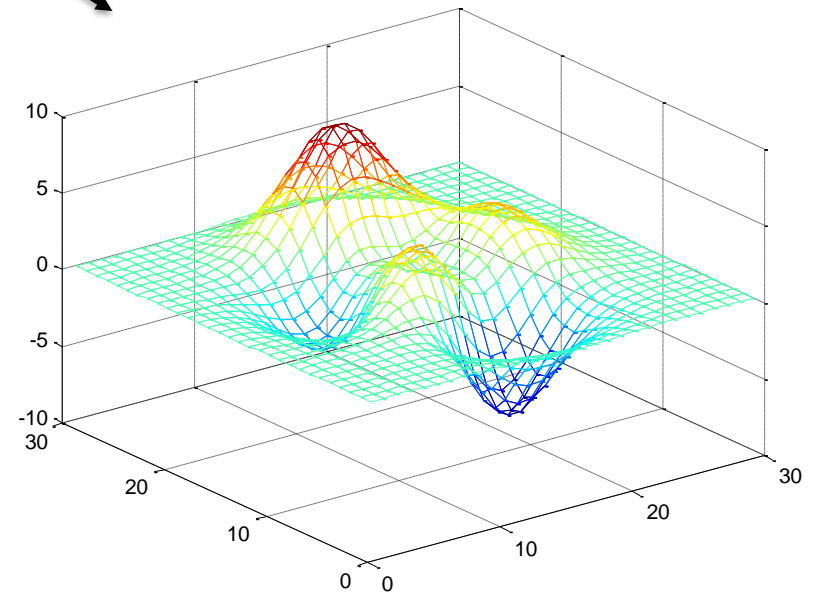
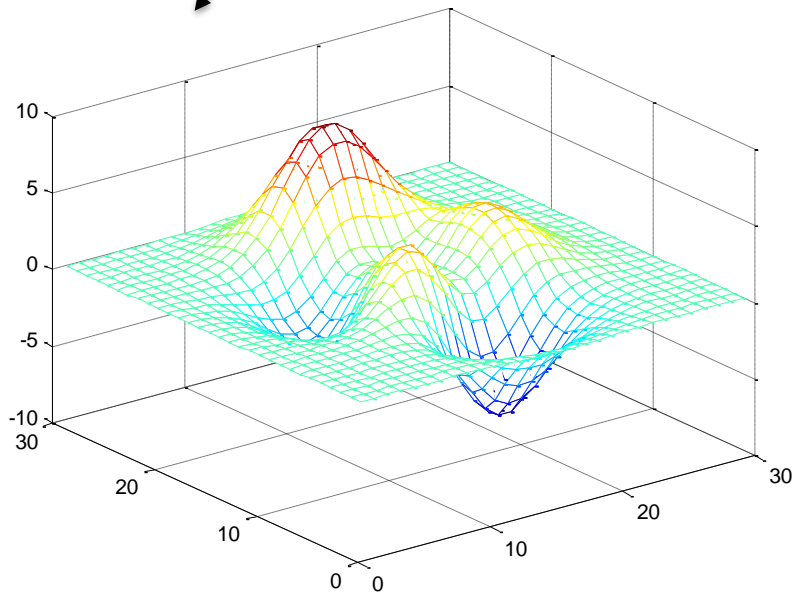
- **imcrop** 裁剪图片
- 等等...

补充:

```
p=peaks(30);mesh(p);
```

**hidden on** %默认

**hidden off**



# 补充:

函数 `view(az,el)`

`az` ---- 方位角; `el` ---- 俯视角

默认值为: `az=-37.5`; `el=30`

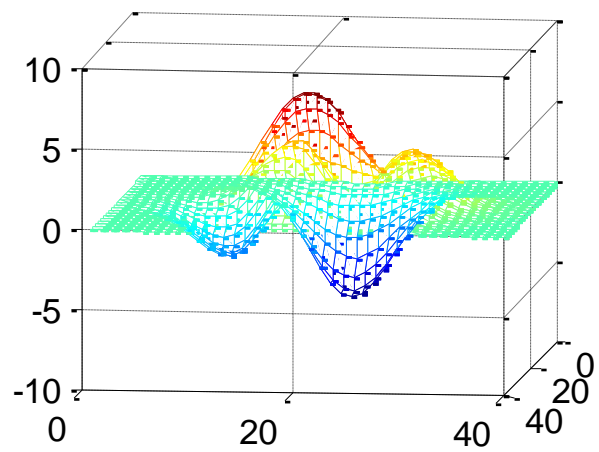
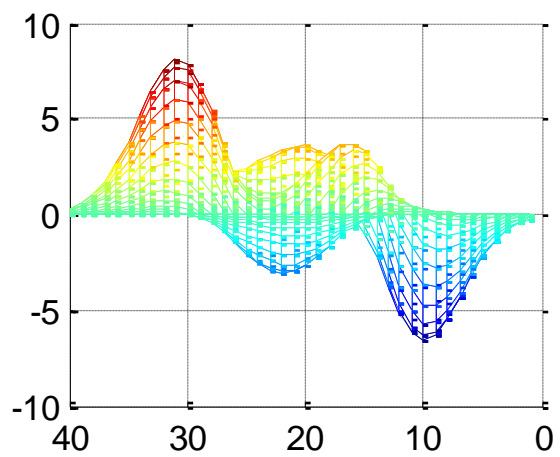
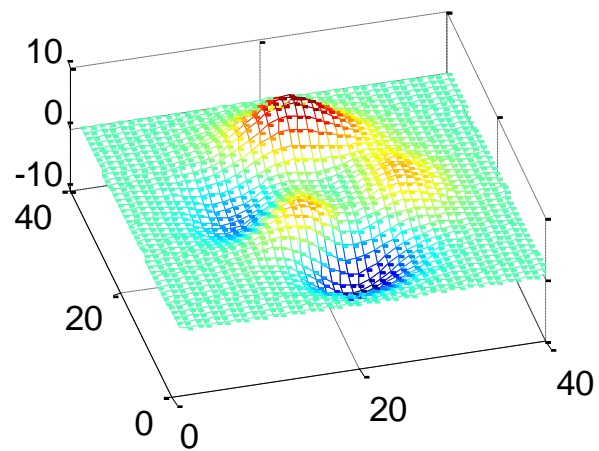
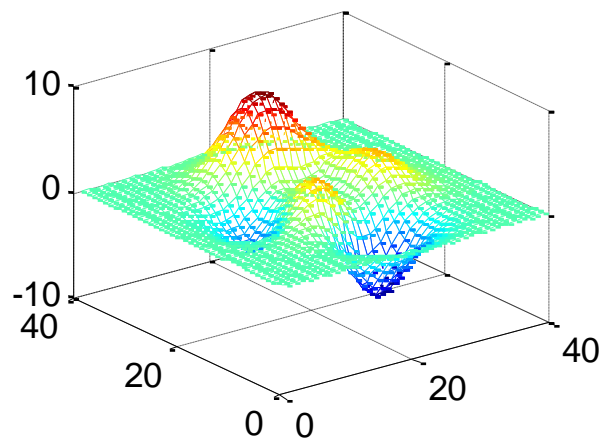
```
z=peaks(40);
```

```
subplot(2,2,1); mesh(z);
```

```
subplot(2,2,2); mesh(z); view(-15,60);
```

```
subplot(2,2,3); mesh(z); view(-90,0);
```

```
subplot(2,2,4); mesh(z); view(-7,-10);
```





## 句柄绘图

利用底层绘图函数，对图形各基本对象进行进一步细致的控制，其操作都是针对图形对象属性的（Handle Graphics），体现了面向对象的程序设计概念。

# 句柄绘图

## 高层绘图与底层绘图的区别

- **高层绘图函数**：对整个图形进行操作，图形每一部分属性都按缺省方式设置，体现了**matlab**的实用性。
- **底层绘图函数**：对图形每一部分进行控制，可用来开发用户界面以及各专业的专用图形。体现了**matlab**的开发性。

# 句柄绘图      基本功能

- ❖ 可随意改变matlab生成图形的方式
- ❖ 允许定制图形的许多特性，无论是对图形做一点小改动，还是影响所有图形输出的整体改动
- ❖ 句柄绘图的许多功能是高层绘图函数无法实现的

# 图形对象

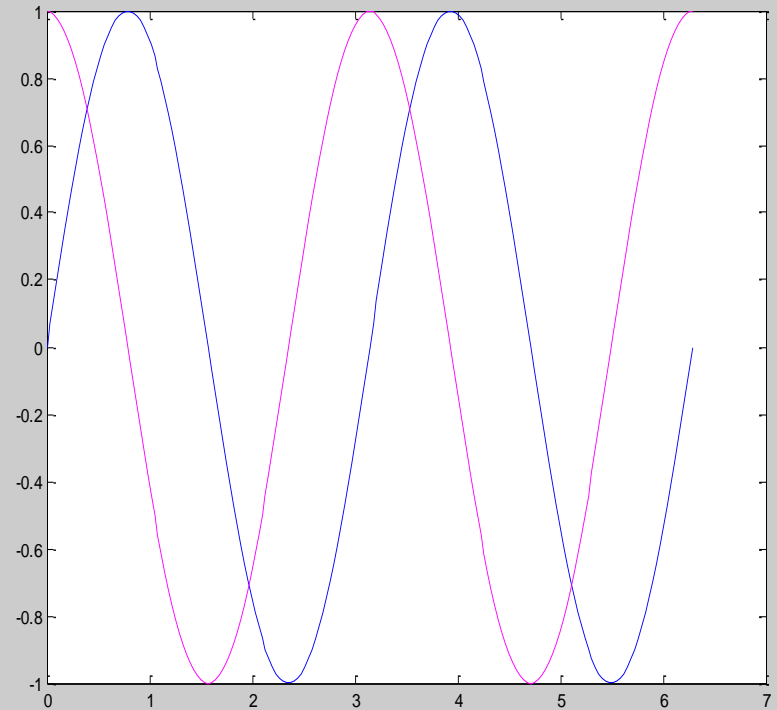
**图形对象：** 图形系统中最基本、最底层的单元

```
x=0:2*pi/180:2*pi;
```

```
y1=sin(2*x);
```

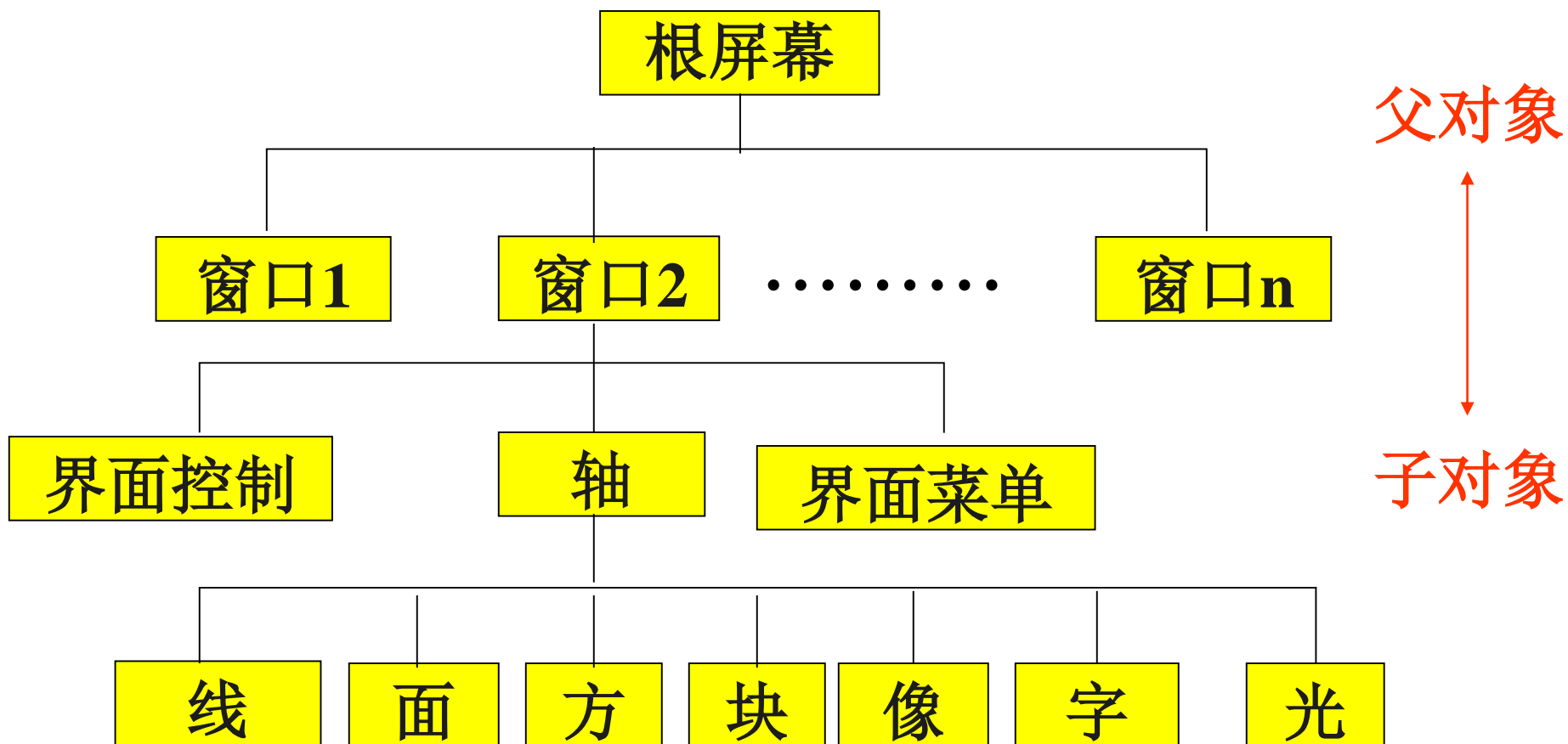
```
y2=cos(2*x);
```

```
plot(x,y1,x,y2)
```





# 图形对象



图形对象之间的关系为父代与子代关系

# 图形对象

**根:**图形对象的根，对应于计算机屏幕，根只有一个，其它所有图形对象都是根的后代。

**图形窗口:**根的子代，窗口的数目不限，所有图形窗口都是根屏幕的子代，除根之外，其它对象则是窗的后代。

**界面控制:**图形窗口的子代，创建用户界面控制对象，使得用户可采用鼠标在图形上作功能选择，并返回句柄。

# 图形对象

**界面菜单:** 图形窗口的子代, 创建用户界面菜单对象

**轴:** 图形窗口的子代, 创建轴对象, 并返回句柄, 线面字块像的父辈

**线:** 轴的子代, 创建线对象

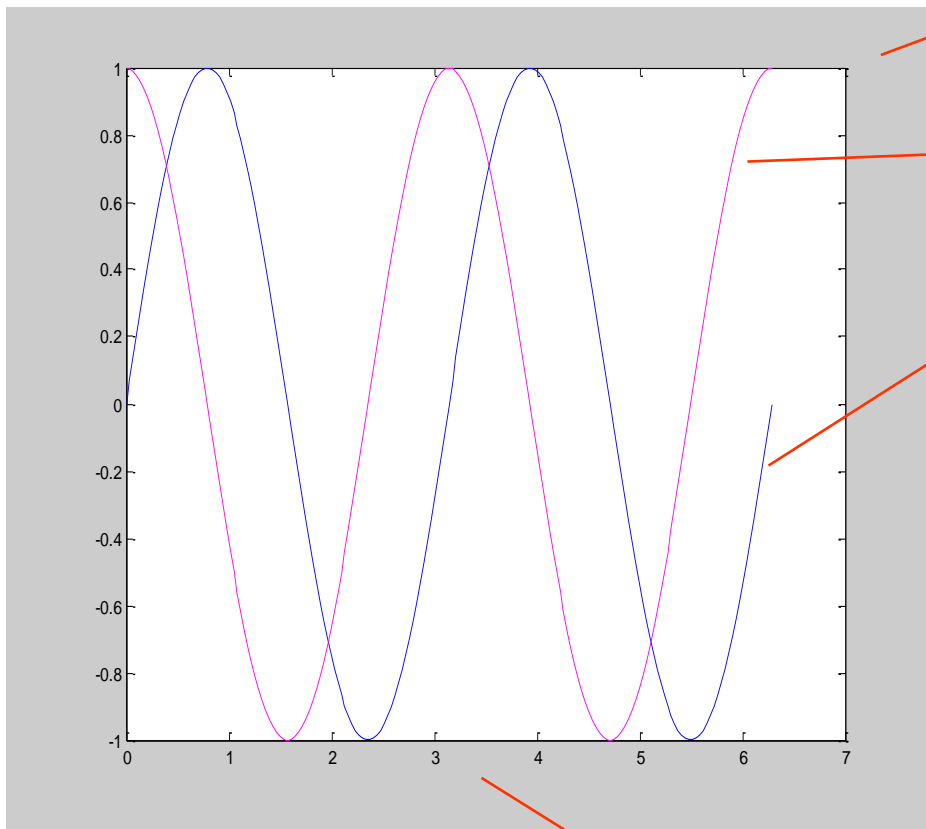
**面:** 轴的子代, 创建块对象

**字:** 轴的子代, 创建字对象

**块:** 轴的子代, 创建块对象

**像:** 轴的子代, 创建图像对象

# 图形对象



窗口对象

线对象

轴对象

# 创建图形对象的底层函数

**figure**——创建图形窗口

调用格式: `h=figure(n)` `n`为窗口序号

**uicontrol**——图形界面控制

调用格式: `h=uicontrol('property',value)`

`property/value`确定控制类型

# 创建图形对象的底层函数

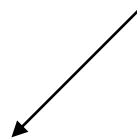
**uimenu**——创建用户界面菜单

`h=uimenu('property',value)`

`property/value` 确定菜单形式

**axes**——创建轴对象

`h=axes('property', [ left, bottom  
width,height ])`



定义轴对象的位置与大小

# 创建图形对象的底层函数

**line** —— 创建线对象

plot, plot3, contour也能创建

$h = \text{line}(x, y, z)$

**patch** —— 创建块对象

fill, fill3, contour3也能创建

$h = \text{patch}(x, y, z, c)$

$x, y, z$ 定义多边形， $c$ 确定填充颜色。

**surface** —— 创建面对象

$h = \text{surface}(x, y, z, c)$

$x, y, z$ 是三维曲面坐标， $c$ 是颜色矩阵。

# 创建图形对象的底层函数

**image**——显示图像

`h=image(x)`; x为图像矩阵

**text**——标注文字

`h=text(x,y,'string')`

确定标注位置

标注字符串

每个底层函数只能创建一个图形对象，并将它们置于适当的父辈对象中。



# 图形对象的属性

图形对象是由属性来描述的

例:

**figure** 大小位置 [1 1 640 480]

颜色 [0 0 0]

**line** x, y数据; 线的颜色; 线型; 线宽

# 图形对象的属性

- 高层绘图中，图形对象的属性一般是缺省或由高层绘图函数自动设置的，几乎不需要用户干预
- 句柄绘图中，图形对象的属性基本都需要用户通过句柄来设置

# 句柄 (handle)

- 句柄是图形对象的标识代码，含有图形对象各种必要的属性信息。
- 各图形对象的句柄数据格式：

根屏幕

0

---

图形窗口

整数，表示图形窗口数

---

其它对象

对应的浮点数

# 句柄 (handle)

- 所有能创建图形对象的matlab函数都可返回所创建图形对象的句柄
- 通过修改句柄可改变该句柄对应的对象属性

例：创建1号窗口，返回句柄值

`h=figure(1)` %关键是有个返回值

`h=1` %返回值为窗口号数

例：`h=line(1:6,1:6)`创建线对象的同时也建立了一个唯一的句柄

变量h是句柄值——浮点数

类似的，`plot`等也有句柄返回值

# 句柄的获得与设置

## ■ 句柄属性的设置与修改

**get:** 获得句柄图形对象的属性和返回某些对象的句柄值

**set:** 改变图形对象的属性

## ■ 专用函数:

**gcf:** 当前窗口对象的句柄

**gca:** 当前轴对象的句柄

**delete(gcf):** 删除当前窗口的属性

# 句柄的获得与设置

例：创建一个线对象,返回其句柄值并查看具体的句柄属性

```
h1=line([0:10],[0:10])
```

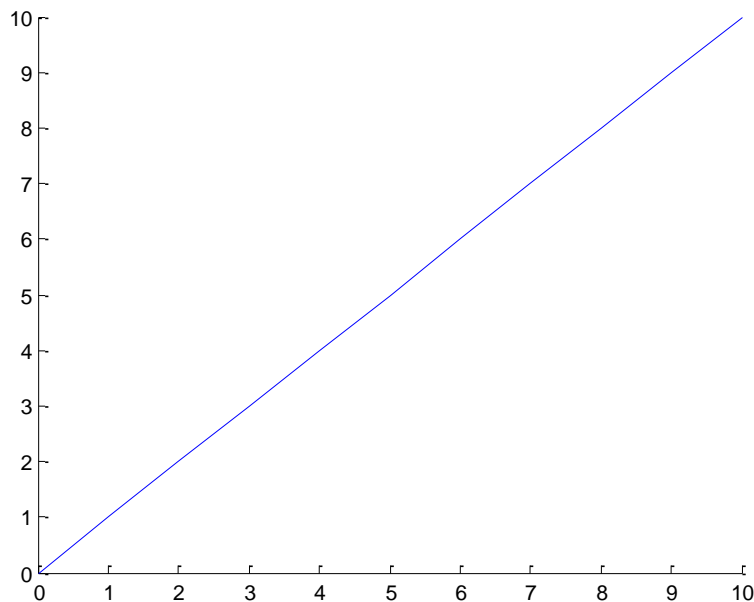
结果是

```
h1 =  
57.0001
```

```
get(h1)
```

结果是

```
DisplayName =  
Annotation = [ (1 by 1) hg.Annotation array]  
Color = [0 0 1] 注意此处线条颜色为默认值蓝色  
LineStyle = -  
... 后面还有很多
```



# 句柄的获得与设置

%接上页指令

**get(gca)** %轴是线对象的父代，可查轴的句柄

结果是

ActivePositionProperty = outerposition

ALim = [0 1]

ALimMode = auto 注意此处坐标轴范围设置为默认值

AmbientLightColor = [1 1 1]

Box = off

**get(gca,'colororder')** %可查色序

结果是

0	0	1.0000
0	0.5000	0
1.0000	0	0
0	0.7500	0.7500
0.7500	0	0.7500
0.7500	0.7500	0
0.2500	0.2500	0.2500

# 句柄的获得与设置

%接上页指令

**set(h1,'color',[1 0 0])** ;%设置线颜色为红色

**get(h1);** %再次查看属性

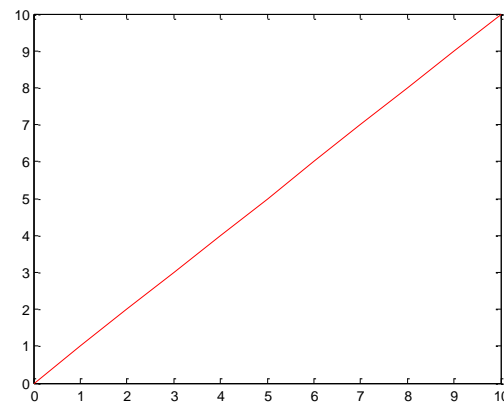
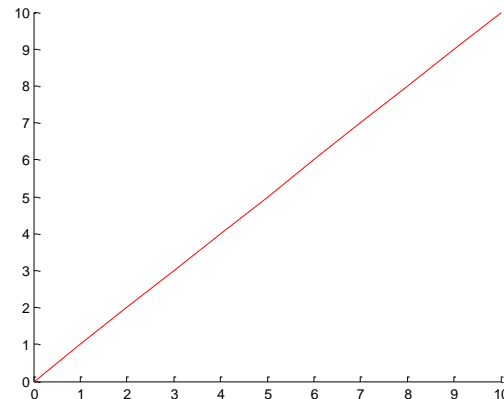
结果是 isplayName =  
Annotation = [ (1 by 1) hg.Annotation array]  
Color = [1 0 0] 这里表示线为红色  
LineStyle = -

...

**box on;**

**get(gca)** %打开边框，再次查看轴对象属性

结果是 ActivePositionProperty = outerposition  
ALim = [0 1]  
ALimMode = auto  
AmbientLightColor = [1 1 1]  
Box = on 这里表示边框打开

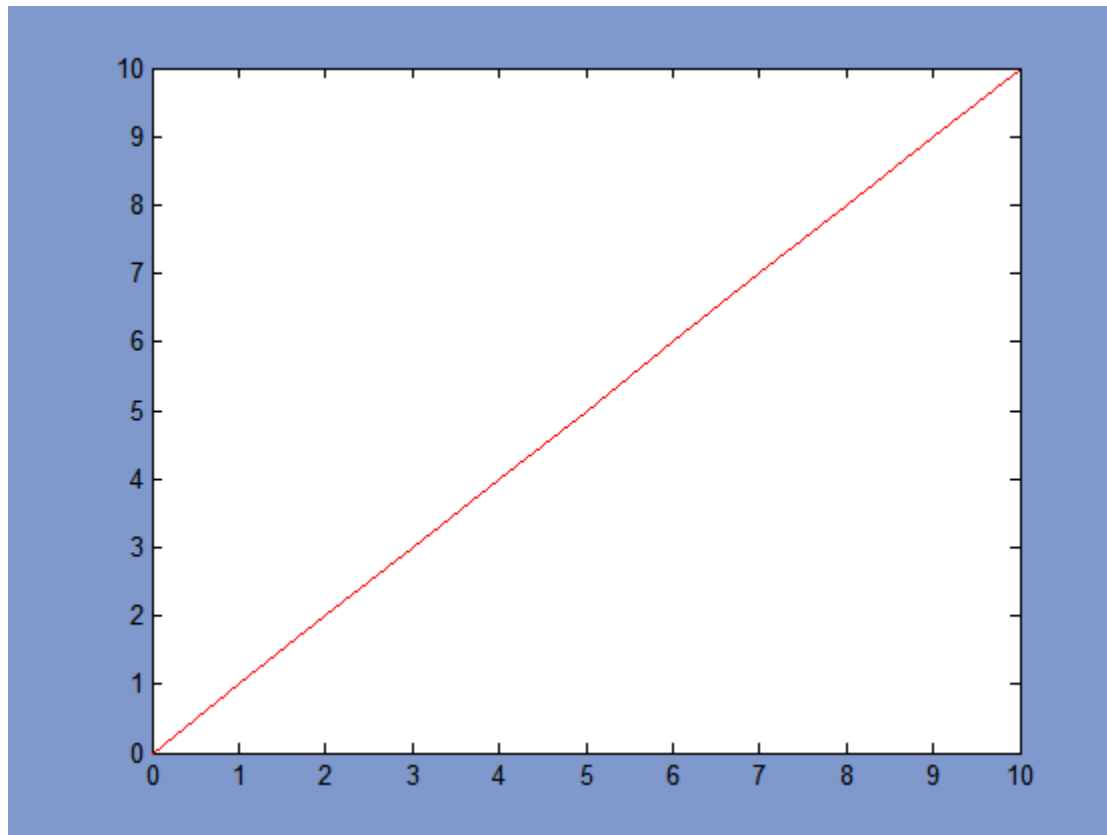




# 句柄的获得与设置

%接上页指令

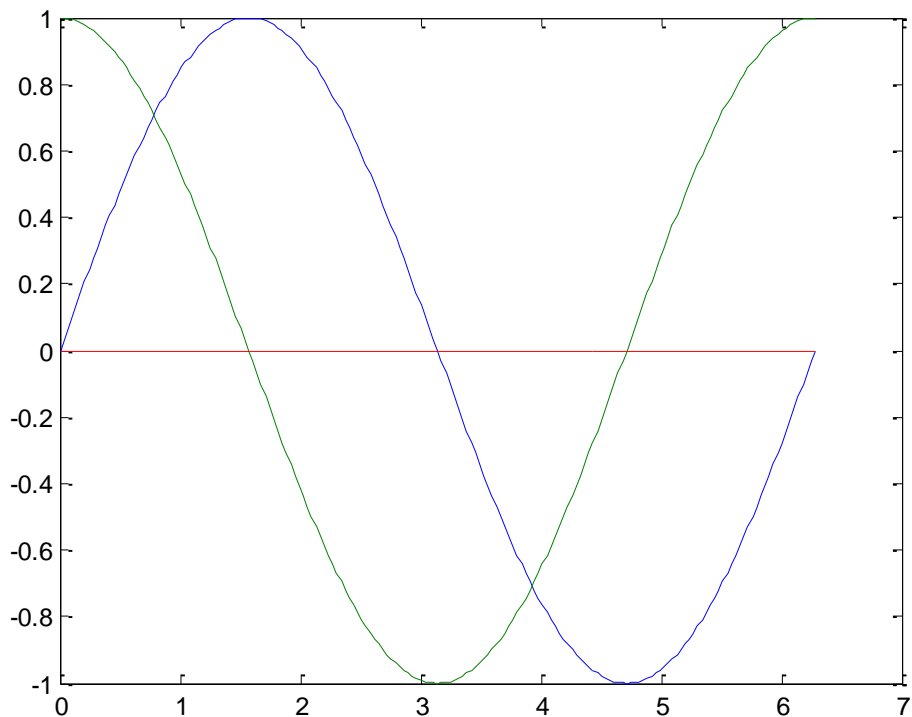
**set(gcf,'color',[0.5 0.6 0.8])** %设置背景颜色



# 句柄的获得与设置

例：作正弦、余弦曲线，将正弦曲线改为虚线，将余弦曲线幅度减为1/2高度，如何用句柄实现。

```
x=0:2*pi/180:2*pi;  
y1=sin(x); y2=cos(x);  
plot(x,y1,x,y2,x,zeros(size(x)))
```



# 句柄的获得与设置

`get(gcf)` %首先查询各对象的属性

`h=get(gcf,'children')`

结果是h =

60.0004

`hh=get(gca,'children')`

结果是hh =

70.0004

69.0004

68.0004

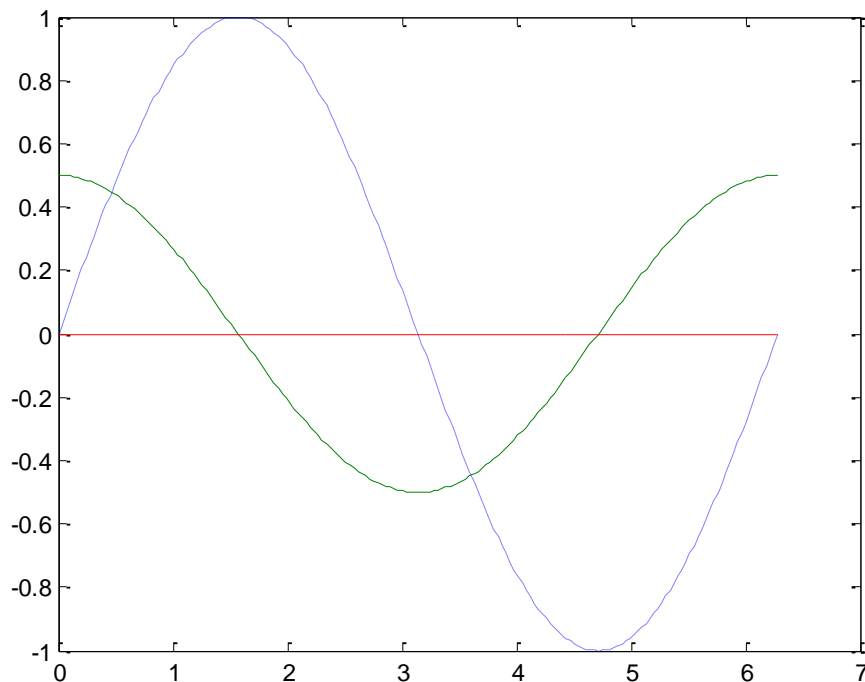
**%修改属性，蓝线发生变化**

`y11=get(hh(2),'ydata');`

`y22=y11/2;`

`set(hh(2),'ydata',y22)`

`set(hh(3),'linestyle',':')`



# 对象属性的继承

除了通过句柄来直接操作对象属性，还能进行对象属性的继承。

- 对象属性的继承操作是通过父代对象，设置缺省对象属性来实现的。
- 父代句柄属性中设置缺省值后，所有子代对象均可以继承该属性的缺省值。
- 属性缺省值的描述结构为：

**Dfault+对象名称+对象属性**

例如

DefaultFigureColor —— 图形窗口的颜色

DefaultAxesAspaceRatio —— 轴的视图比率

DefaultLineLineWide —— 线的宽度

DefaultLineColor —— 线的颜色

缺省值的获得与设置也是由get,set函数实现的。

# 对象属性的其他操作

default —— 由父代继承

factory —— 厂家设定缺省值

remove —— 清除设定缺省值

例： `set(gca,'defaultlinecolor','remove')`  
`set(h,'color','factory')`

注意：

`plot,line`遵循`colororder`色序，对设置缺省值无效。

# 补充

例：在轴对象上（父代对象）设置线的颜色缺省值为红色

```
close,clear
```

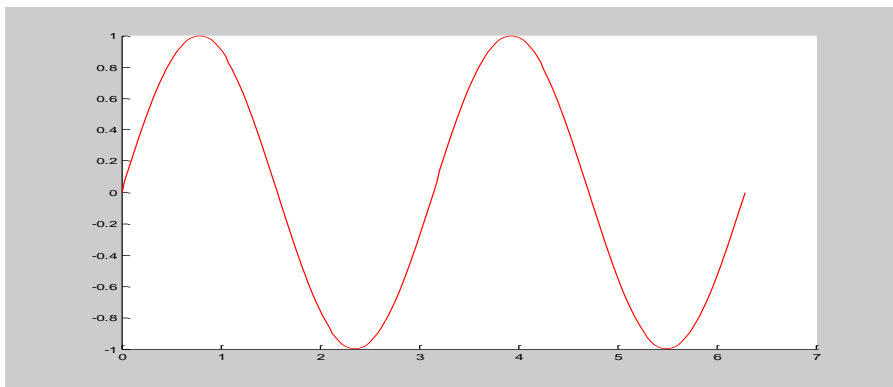
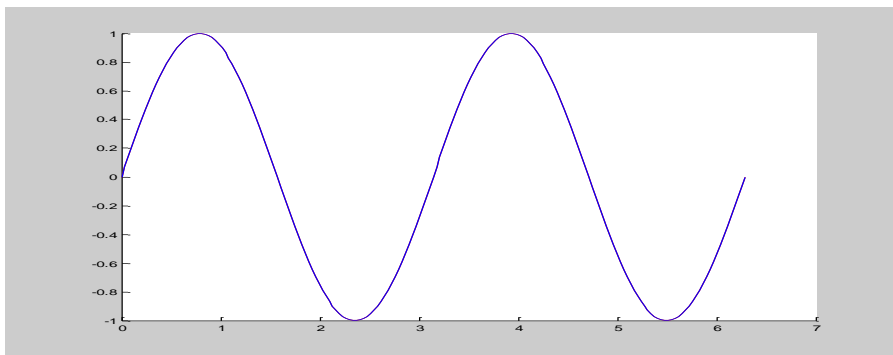
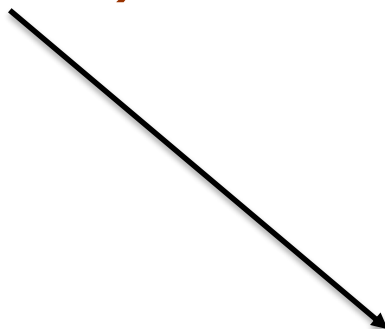
```
x=0:2*pi/180:2*pi;
```

```
y=sin(2*x);
```

```
set(gca,'DefaultLineColor',[1 0 0]);
```

```
h=line(x,y);
```

```
set(h,'color','default')
```



# 第3章

1. 二维曲线绘图 **plot**基本用法（类似的还有**fplot** 等）
2. 一图多线，一窗多子图，双纵坐标
3. 图片的常用设置（曲线颜色、线型、附加符号、图片标题、坐标名称、图例、文本、网格）
4. 三维曲线/曲面绘图**plot3** **mesh** **surf**
5. 以下内容了解基本概念

图片的读写等操作，准四维绘图、饼状图、柱状图、杆图、极坐标图、对数坐标图等，图片的更详细设置如字体字号等，图形修饰和句柄