

Matlab绘图

Matlab提供了大量的**高层绘图函数**，图形每一部分的属性都是按缺省方式设置的，绘图即方便又高效。

同时，Matlab还提供了**底层绘图函数**，通过对图形对象属性的设置与操作，可以对图形的每一部分进行控制。（这种绘图方式常称为**句柄绘图**）

一、2d绘图

二、图形属性控制

三、3d绘图

一、2d绘图

- 1、绘制2d图形的基本步骤
- 2、直角坐标系绘图: `plot()`
- 3、对数坐标绘图: `loglog()`, `semilogx()`, ...
- 4、双y轴绘图: `plotyy()`
- 5、极坐标系绘图: `polar()`
- 6、二维绘图函数汇总

1、绘制2d图形的基本步骤

如何画出 $y=\sin(x)$ 在 $[0, 2*\pi]$ 上的图像？

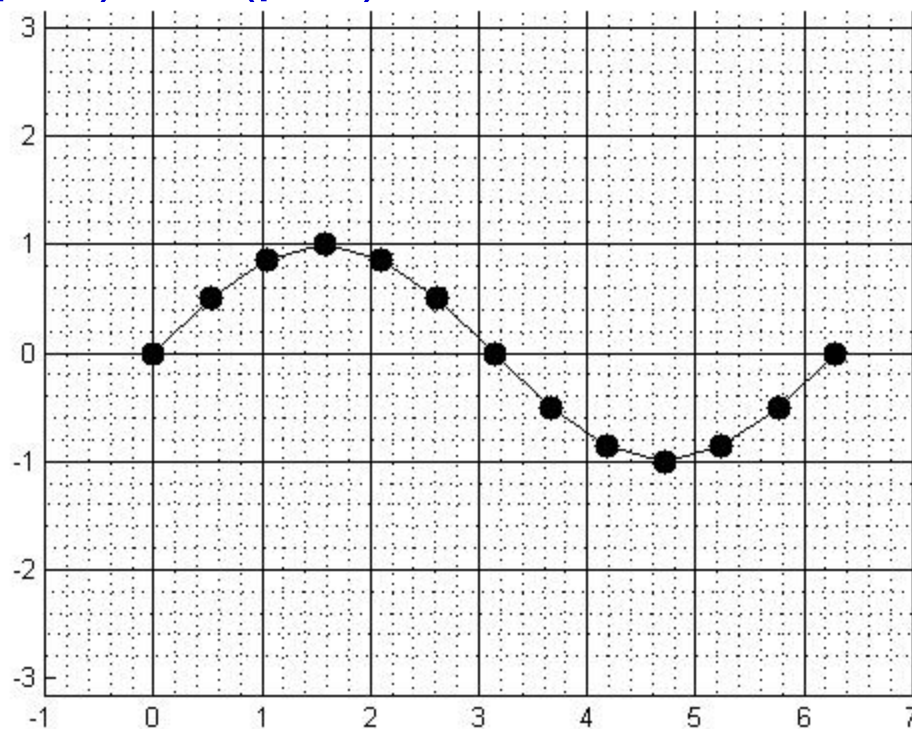
手工作图

1、找点： $x=0, \pi/6, \pi/3, \pi/2, 2*\pi/3, 5*\pi/6, \dots$

2、计算函数值： $y=\sin(0), \sin(\pi/6), \sin(\pi/3), \dots$

3、描点：在坐标系中画出这些离散点

4、用直线或曲线连接这些点，得到函数的大致图形



Matlab 作图的基本步骤

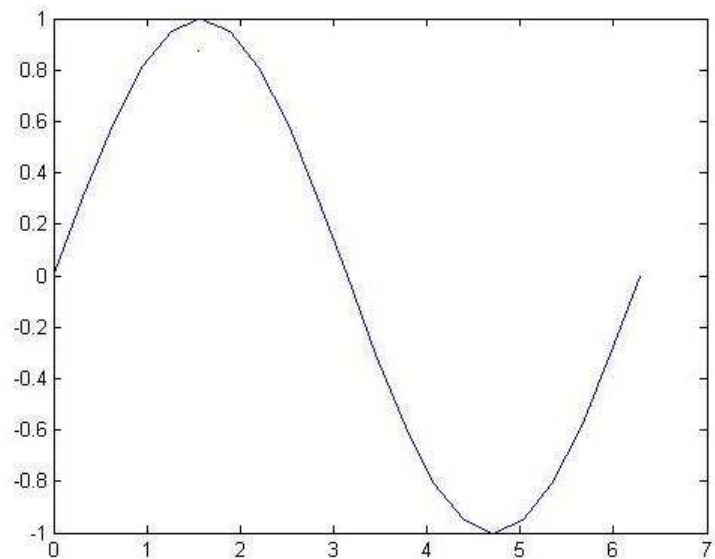
1、准备绘图数据：

```
x=[0:pi/10:2*pi]
```

```
y=sin(x)
```

2、调用绘图函数作出图形，

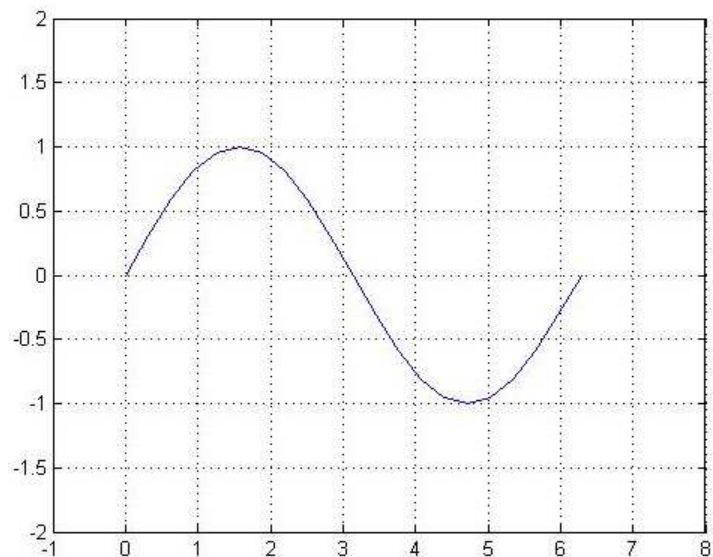
如： `plot(x,y)`



3、调用相关函数和命令调整图形特性，如：

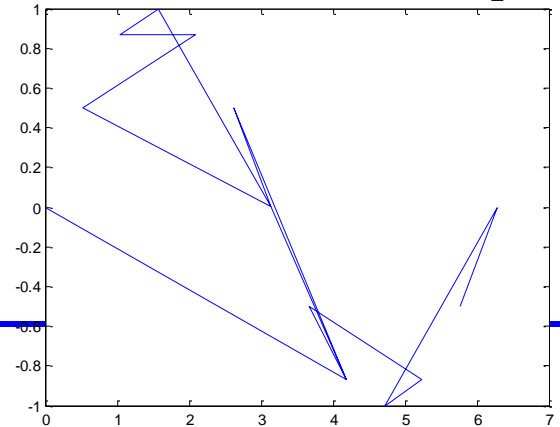
```
grid on
```

```
axis([-1,8,-2,2])
```

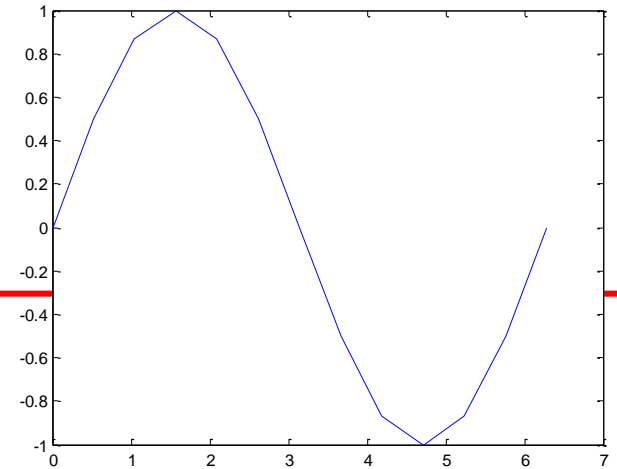


注意： 准备matlab绘图数据时，数据应进行排序。

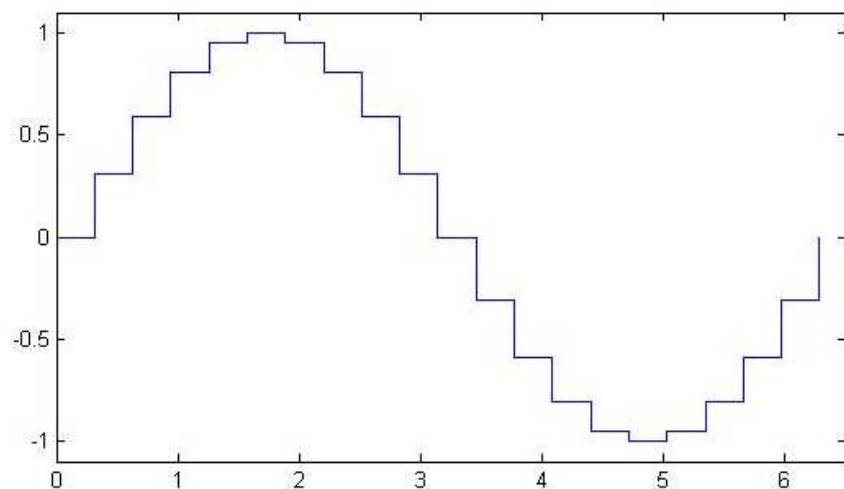
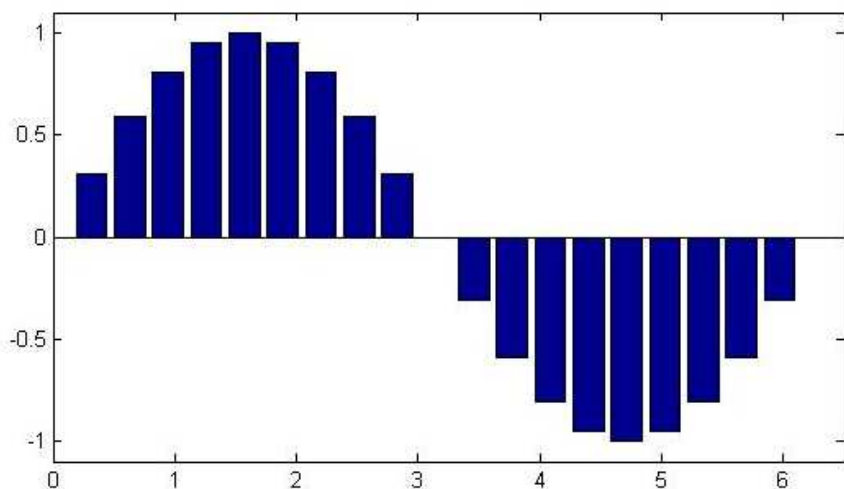
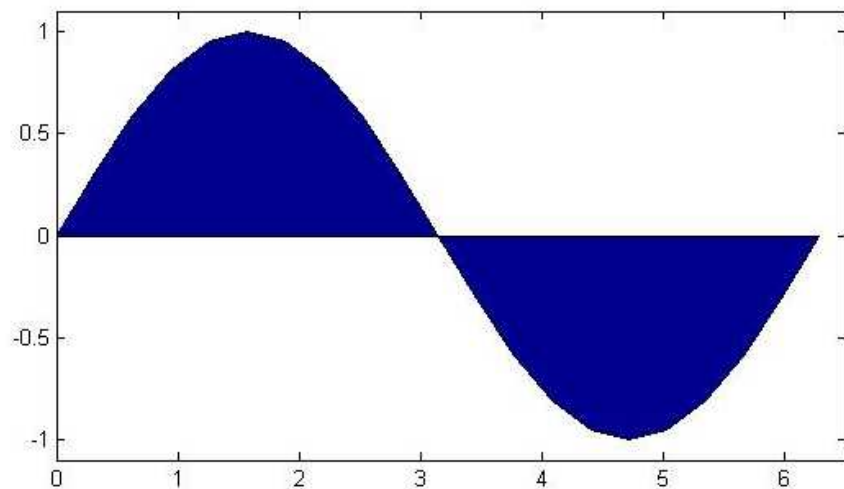
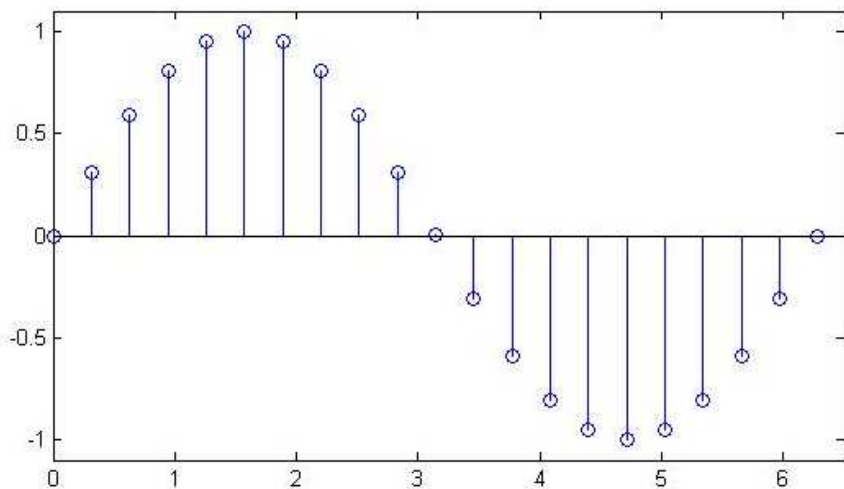
```
x = [0,4/3,1,1/2,1/3,2/3,1/6,1,5/6,4/3,7/6,5/3,3/2,2,11/6]
x = x*pi
y = sin(x)
plot(x,y)
```



```
x = [0,4/3,1,1/2,1/3,2/3,1/6,1,5/6,4/3,7/6,5/3,3/2,2,11/6]
x = sort(x)*pi
y = sin(x)
plot(x,y)
```



Matlab 作图比手工作图功能强大



2、直角坐标系2d绘图：plot()

plot()是最基本的二维绘图函数，功能包括：

- 自动打开一个图形窗口(Figure)，如果已经存在一个图形窗口，则清除当前图形，绘制新图形
- 用直线连接相邻两数据点来绘制图形
- 可以一次绘制一条曲线，也可以一次绘制多条曲线
- 可以设定绘图颜色、点型、线型、线宽等特性

```
plot(Y)
```

```
plot(X1,Y1,...,Xn,Yn)
```

```
plot(X1,Y1,LineSpec,...,Xn,Yn,LineSpec)
```

```
plot(...,'PropertyName',PropertyValue,...)
```

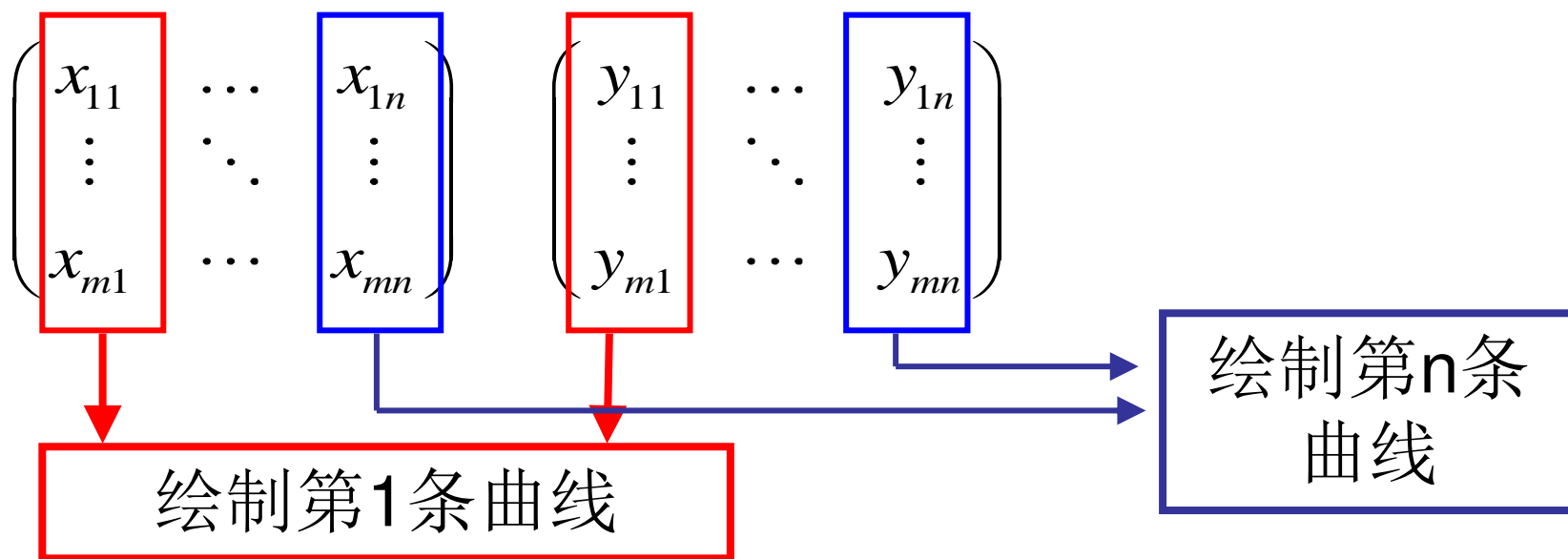
```
plot(axes_handle,...)
```

```
h = plot(...)
```

plot(X, Y)

当X, Y 都是实数一维数组，以 X 中元素为横坐标， Y 中元素为纵坐标作平面曲线。此时 X, Y 必须具有相同长度。

当 X, Y 都是实数二维数组，将 X 的列和 Y 相应的列相组合，绘制多条平面曲线。此时 X, Y 必须具有相同的维数。



当 X, Y 是复数数组时，忽略虚部，等效于：

```
plot(real(X), real(Y))
```


plot(x, Y) x 是一维数组, Y 是二维数组

若 x 的长度与 Y 的行数相等, 则将 x 与 Y 中的各列相对应, 绘制多条平面曲线;

若 x 的长度与 Y 的列数相等, 则将 x 与 Y 中的各行相对应, 绘制多条平面曲线。

plot(X, y) X 是二维数组, y 是一维数组

若 y 的长度与 X 的行数相等, 则将 X 中的各列与 y 相对应, 绘制多条平面曲线;

若 y 的长度与 X 的列数相等, 则将 X 中的各行与 y 相对应, 绘制多条平面曲线。

plot(Y)

当Y 是实数一维数组时，等价于：

```
x = 1:length(Y)
plot(x,Y)
```

当Y 是实数二维数组时，等价于：

```
x = 1:size(Y,1)
plot(x,Y)
```

当Y为复数数组时，等价于：

```
plot(real(Y),imag(Y))
```

例：利用二维数组绘制图形：

$$y = k \cos(x), k = 1, 2, 3$$

$$x \in [0, 2\pi]$$

```
x = 0:0.1:2*pi
```

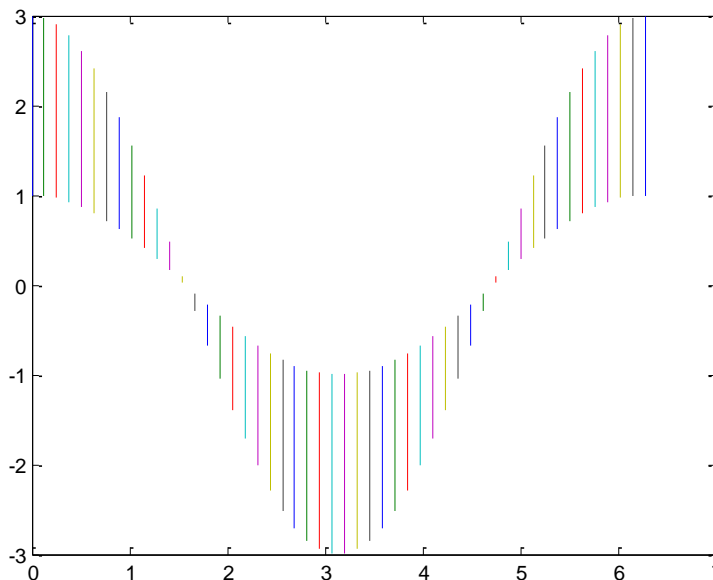
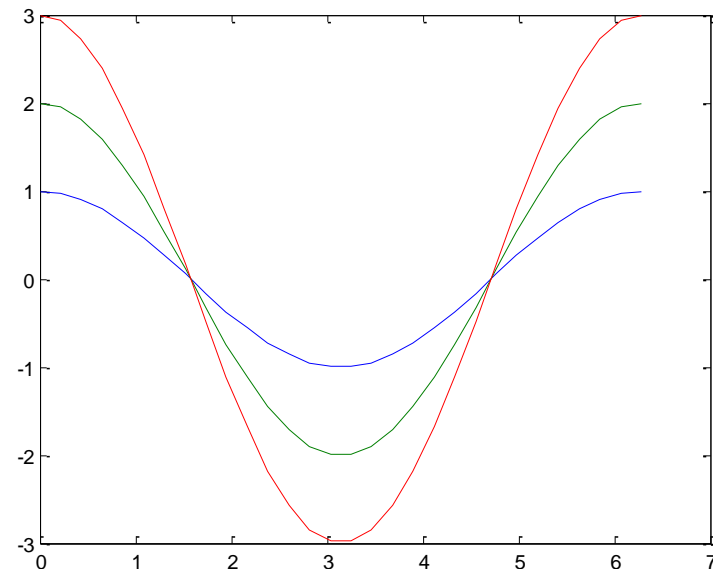
```
x = x'
```

```
X = [x, x, x]
```

```
Y = [cos(x), 2*cos(x), 3*cos(x)]
```

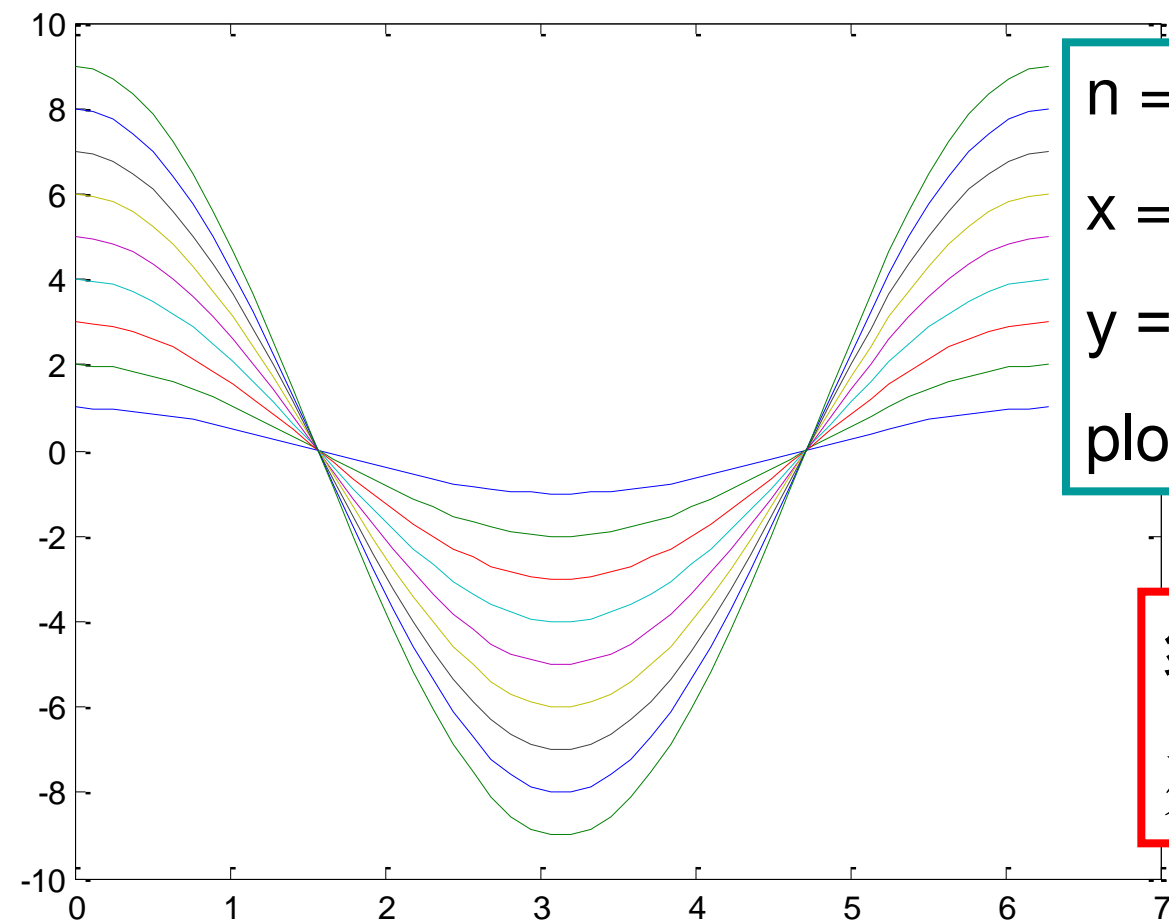
```
plot(X,Y)
```

```
plot(X',Y')
```



例：利用矩阵乘法生成绘图数据，并绘制图形：

$$y = k \cos(x), x \in [0, 2\pi], k = 1, 2, \dots, n$$



`n = 9`

`x = linspace(0, 2*pi, 50)`

`y = cos(x)'*(1:n)`

`plot(x, y)`

线性代数中：

列 \times 行 = 矩阵

plot(X1,Y1,LineSpec,...,Xn,Yn,LineSpec)

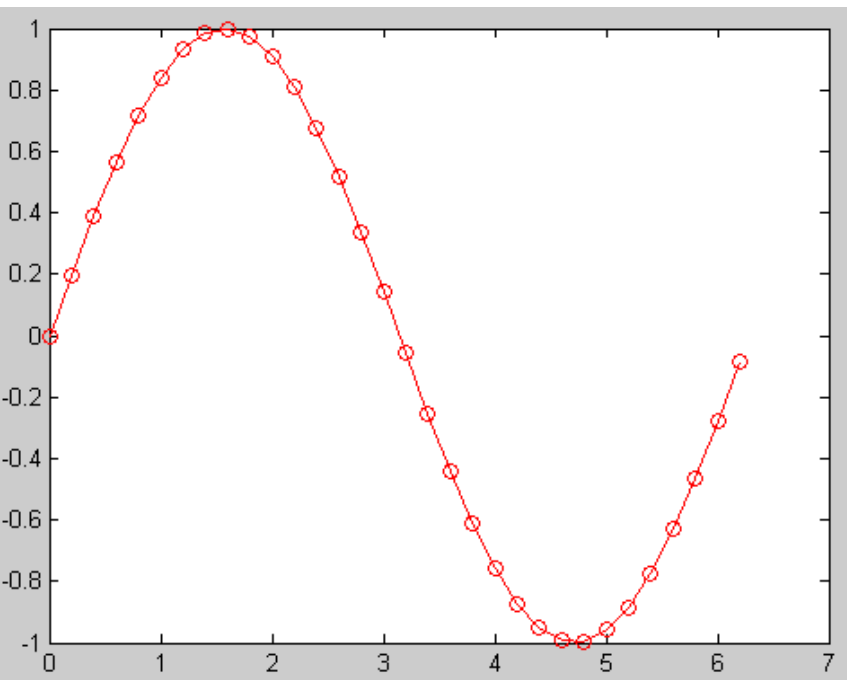
按照三个参数Xn, Yn, LineSpec画线，其中LineSpec指定了线型，点标记和画线的颜色，也可以混合使用三参数Xn,Yn,LineSpec和二参数Xn,Yn:

plot(X1,Y1,X2,Y2,LineSpec,X3,Y3)。

LineSpec的三种属性可以全部指定，也可以只指定其中某一个或两个，排列顺序任意

线型	点标记		颜色	
- 实线	. 点	v 下三角	b 蓝色	m 棕色
: 点线	o 小圆圈	^ 上三角	g 绿色	y 黄色
- . 点划线	x 叉号	< 左三角	r 红色	k 黑色
-- 虚线	+ 加号	> 右三角	c 青色	w 白色
	• 星号	p 五角星		
	s 方格	h 六角星		
	d 菱形			

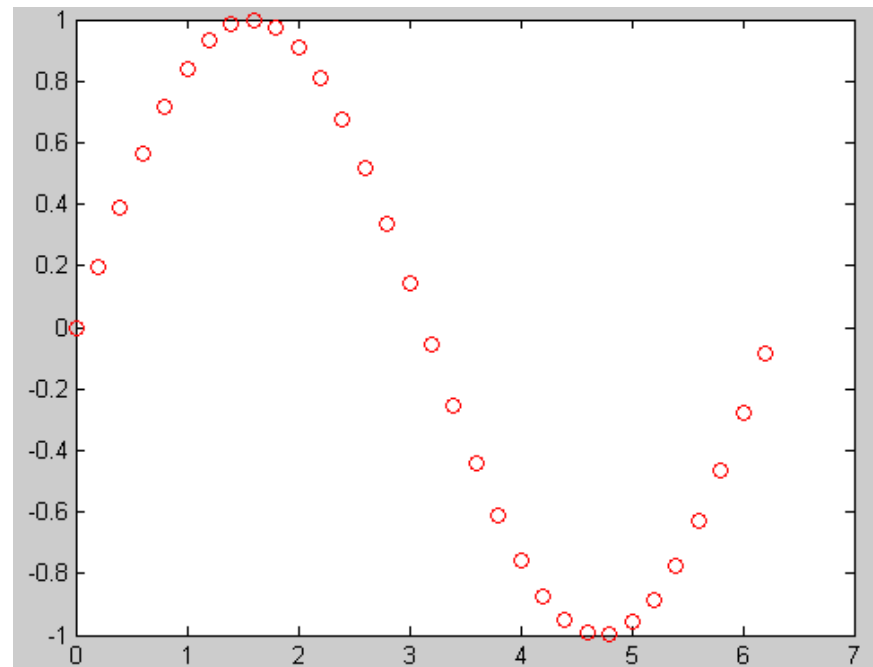
```
x = 0:0.2:2*pi;  
y = sin(x);  
plot(x,y,'r-o')
```



```
x = 0:0.2:2*pi  
y = sin(x)  
plot(x,y,'ro')
```



指定不指定线型的话，
`plot()`只绘制点标记



plot(X1,Y1,LineSpec,'PropertyName',PropertyValue)

对plot生成的图形对象，用指定的properties进行设置。

PropertyName

Color 非标准颜色可用[r,g,b]指定， r,g,b取0和1之间的数值

LineWidth 线宽

LineStyle 线型

Marker 点标记

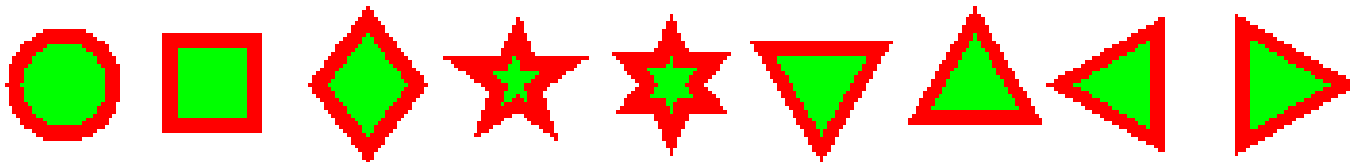
MarkerEdgeColor marker的颜色(或 **filled markers** 边的颜色)

MarkerFaceColor **the face of filled markers** 的颜色

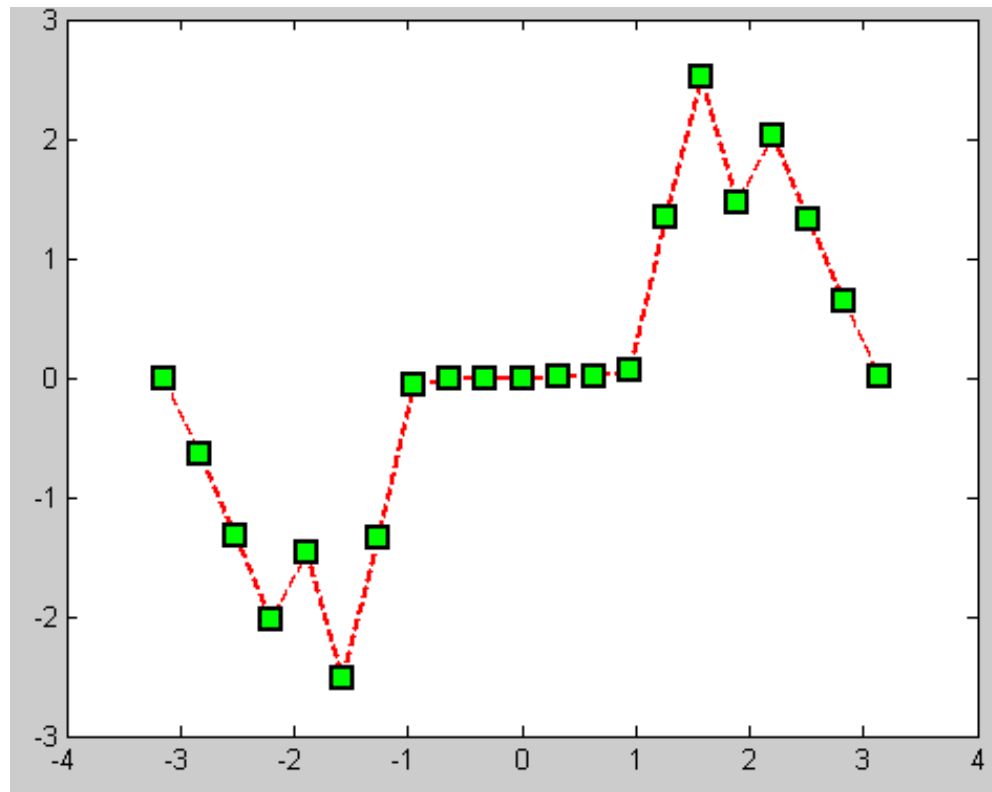
MarkerSize 点标记的大小

filled markers

circle, square, diamond, pentagram, hexagram, and the four triangles)



```
x = -pi:pi/10:pi;  
y = tan(sin(x)) - sin(tan(x));  
plot(x,y,'--rs','LineWidth',2,...  
      'MarkerEdgeColor','k',...  
      'MarkerFaceColor','g',...  
      'MarkerSize',10)
```



`plot(axes_handle, ...)`

将图形绘制在由`axes_handle`指定的坐标系中。

```
x = 0:0.2:2*pi
```

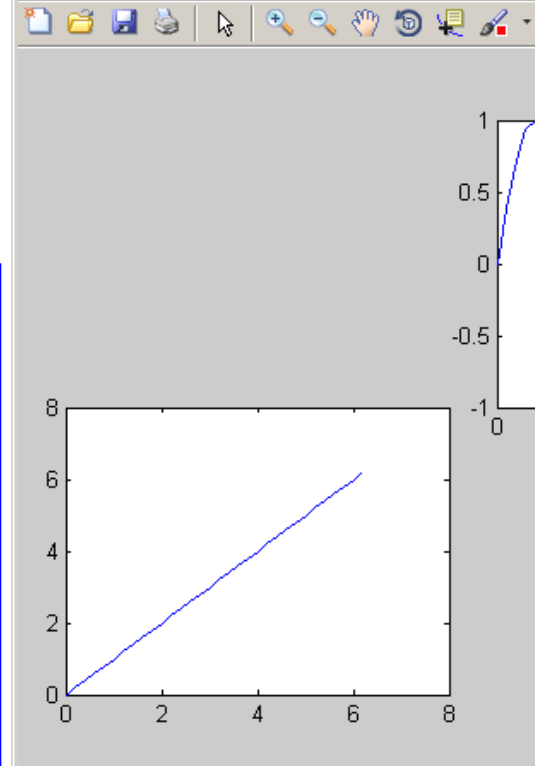
```
h1 = axes('position',[0.05,0.1,0.4,0.4])
```

```
h2 = axes('position',[0.5,0.5,0.4,0.4])
```

```
plot(h1,x,x)
```

```
plot(h2,x,sin(2*x))
```

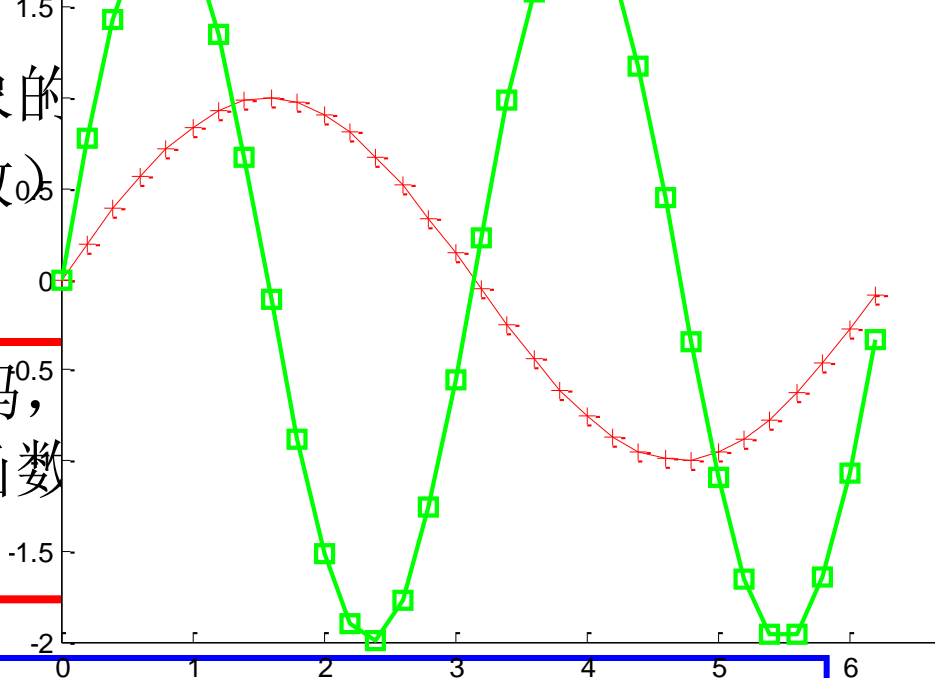
`axes()`是创建坐标轴的函数，后面会具体介绍。



h = plot(...) 返回图形对象的句柄（是一个双精度数），以列向量的形式返回至 **h**。

句柄是图形对象的标识代码，种必要的属性信息，可用 **get()** 函数设置（修改）属性。

```
x = 0:0.2:2*pi  
Y = [sin(x); 2*sin(2*x)]  
h = plot(x,Y)  
get(h(1))  
set(h(1),'Color','r','LineWidth',1,'Marker','+')  
set(h(2),'Color','g','LineWidth',2,'Marker','s')
```



上机练习

1、绘制连续调制波形 $y = \sin(t)\sin(9t)$

$$0 \leq t \leq 2\pi$$

2、绘制圆： $x^2 + y^2 = 4$

3、绘制椭圆： $\frac{x^2}{4} + y^2 = 1$

3、对数坐标绘图

loglog() : 作图时, x轴、y轴都使用常用对数刻度

loglog(Y)

loglog(X1,Y1,...)

loglog(X1,Y1,LineSpec,...)

loglog(...,'PropertyName',PropertyValue,...)

h = loglog(...)

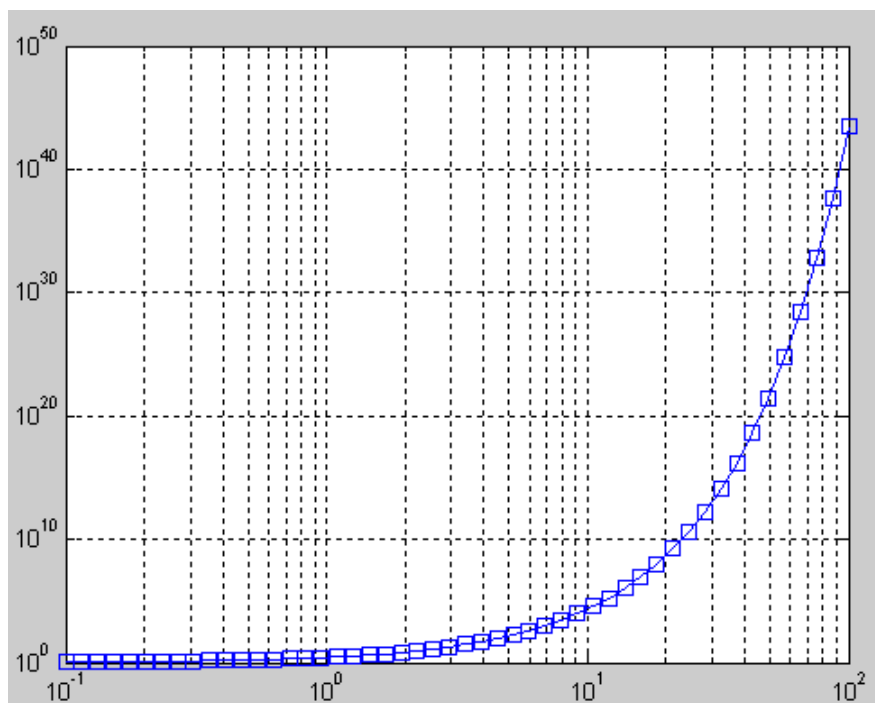
用法与plot()函数完全相同

semilogx() x轴: 常用对数刻度, y轴: 线性刻度

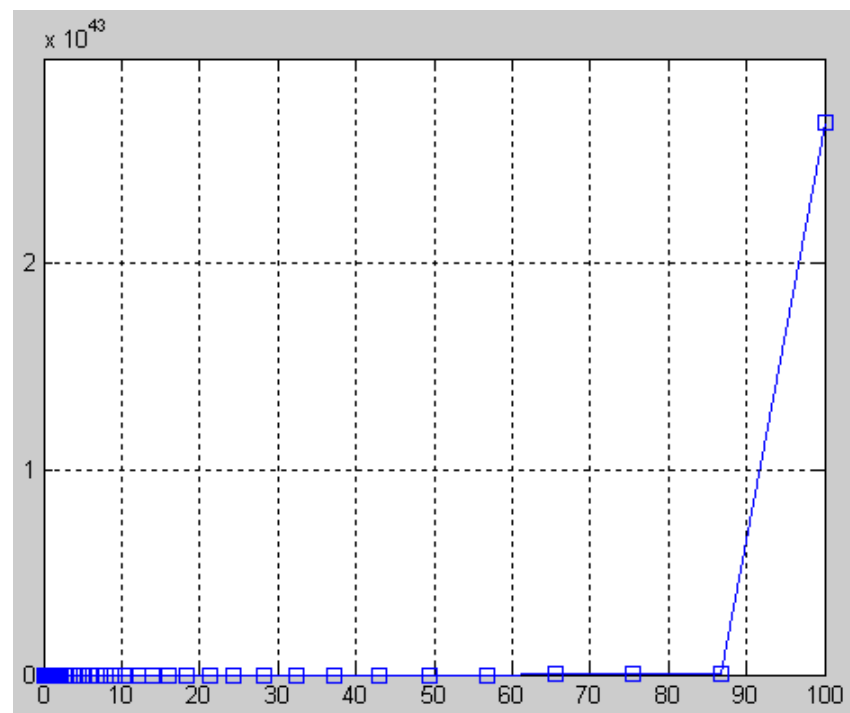
semilogy() x轴: 线性刻度, y轴: 常用对数刻度

loglog() 与 plot() 作图比较

```
x = logspace(-1,2);  
loglog(x,exp(x),'-s')  
grid on
```

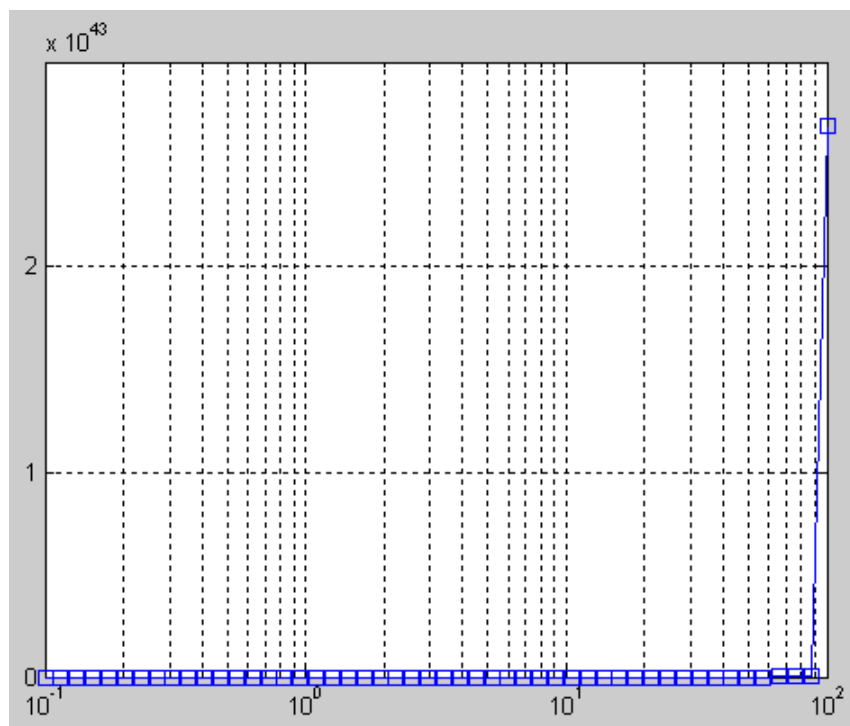


```
x = logspace(-1,2);  
plot(x,exp(x),'-s')  
grid on
```

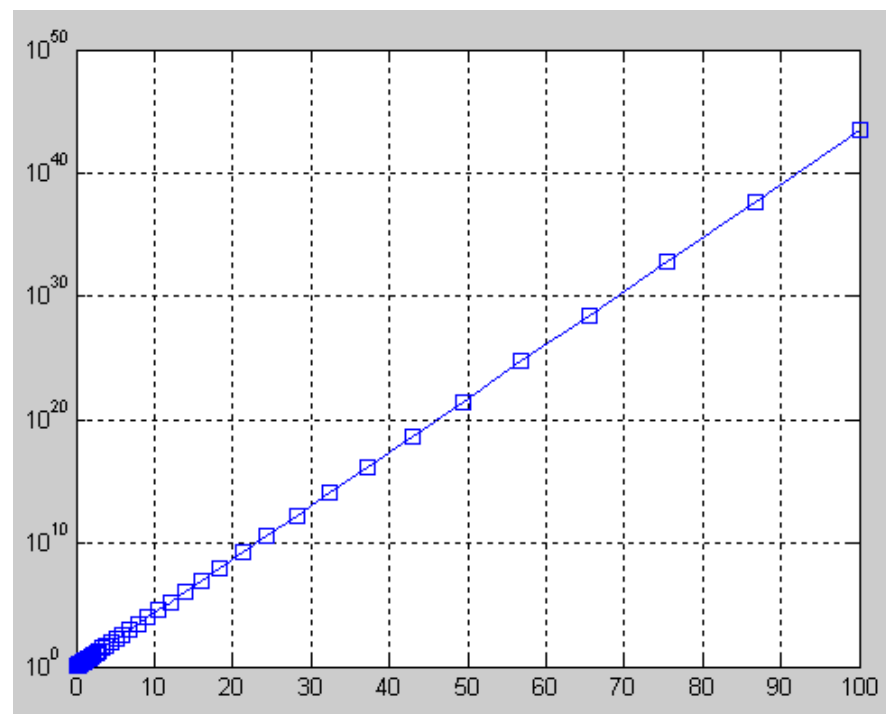


semilogx() 与 semilogy() 作图比较

```
x = logspace(-1,2);  
semilogx(x,exp(x),'-s')  
grid on
```



```
x = logspace(-1,2);  
semilogy(x,exp(x),'-s')  
grid on
```



4、双y轴绘图: plotyy()

如果需要绘制出具有不同纵坐标标度的两个图形，可以使用`plotyy()`绘图函数。

`plotyy(X1,Y1,X2,Y2):`

其中X1,Y1对应一条曲线，X2,Y2对应另一条曲线。横坐标的标度相同，纵坐标有两个，左纵坐标用于x1,y1数据对，右纵坐标用于x2,y2数据对。

`plotyy(X1,Y1,X2,Y2,'function')`

使用由function指定的函数绘图，function可以是：plot, semilogx, semilogy, loglog, stem.....

`plotyy(X1,Y1,X2,Y2,'function1','function2')`

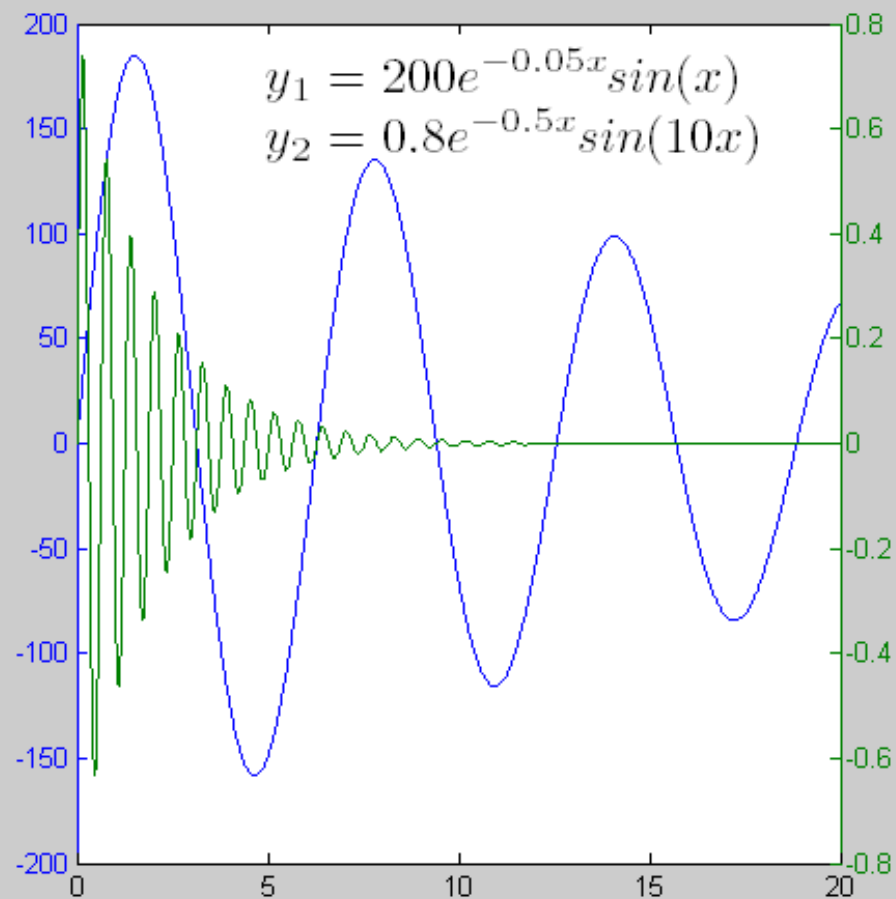
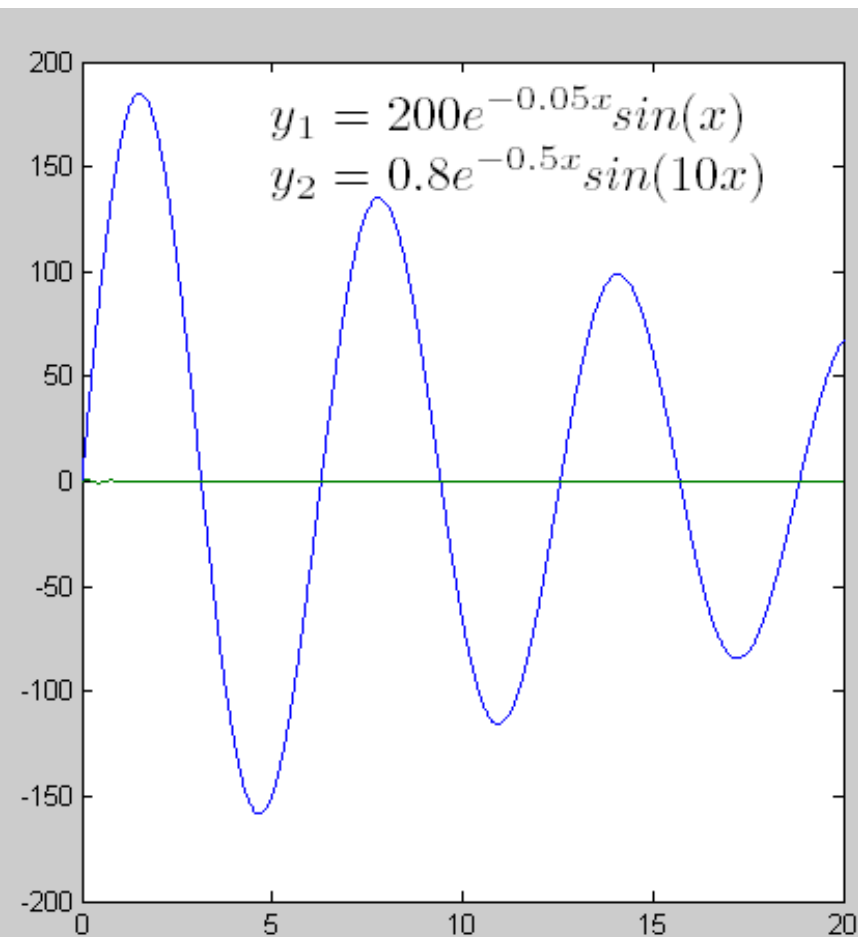
使用由function1指定的函数作X1, Y1的图

使用有function2指定的函数作X2, Y2的图

`[AX,H1,H2] = plotyy(...)`

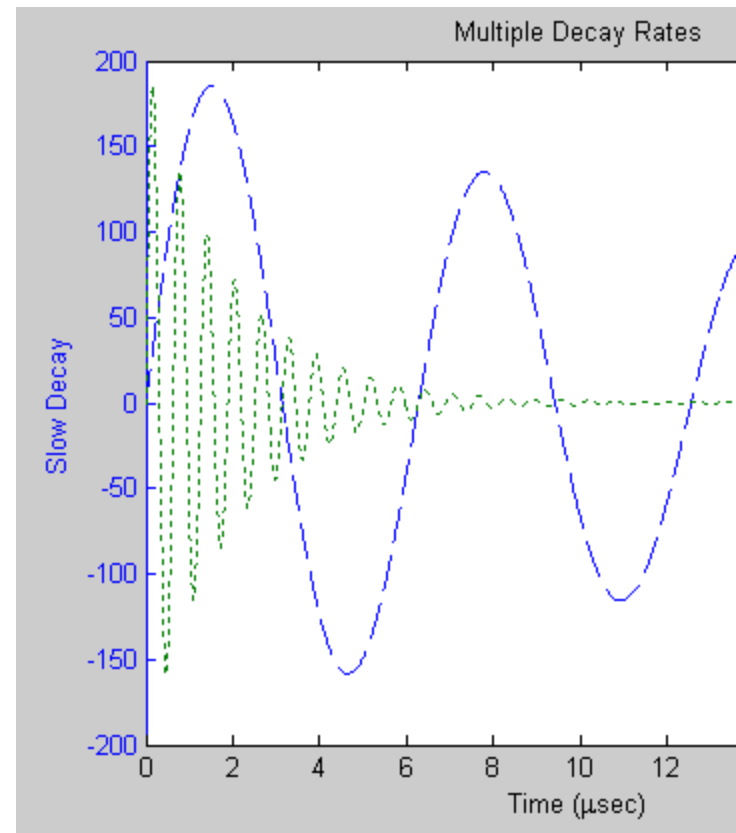
AX: 存储两个Y轴的句柄，AX(1)为左Y轴的句柄，AX(2)为右Y轴的句柄。H1和H2分别为两个图形对象的句柄。

plotyy()和plot()作图比较



plotyy 举例

```
x = 0:0.01:20;  
y1 = 200*exp(-0.05*x).*sin(x);  
y2 = 0.8*exp(-0.5*x).*sin(10*x);  
[AX,H1,H2] = plotyy(x,y1,x,y2,'plot');  
  
title('Multiple Decay Rates')  
xlabel('Time (\musec)')  
  
set(get(AX(1),'Ylabel'),'String','Slow Decay')  
set(get(AX(2),'Ylabel'),'String','Fast Decay')  
  
set(H1,'LineStyle','--')  
set(H2,'LineStyle',':')
```



5、极坐标系绘图：polar()

polar()接受极坐标形式的函数 $\rho=f(\theta)$,在笛卡尔坐标系平面上画出该函数，且在平面上画出极坐标形式的栅格。

polar(Q,r)

用极角 θ 和极径 ρ 画出极坐标图形。 θ 是从x轴到指定矢量半径的夹角，单位为弧度， ρ 是数据空间单位指定的矢量半径的单位。

polar(Q,r,LineSpec)

参数LineSpec指定极坐标图中线条的线型、标记符号和颜色。

polar(axes_handle,...)

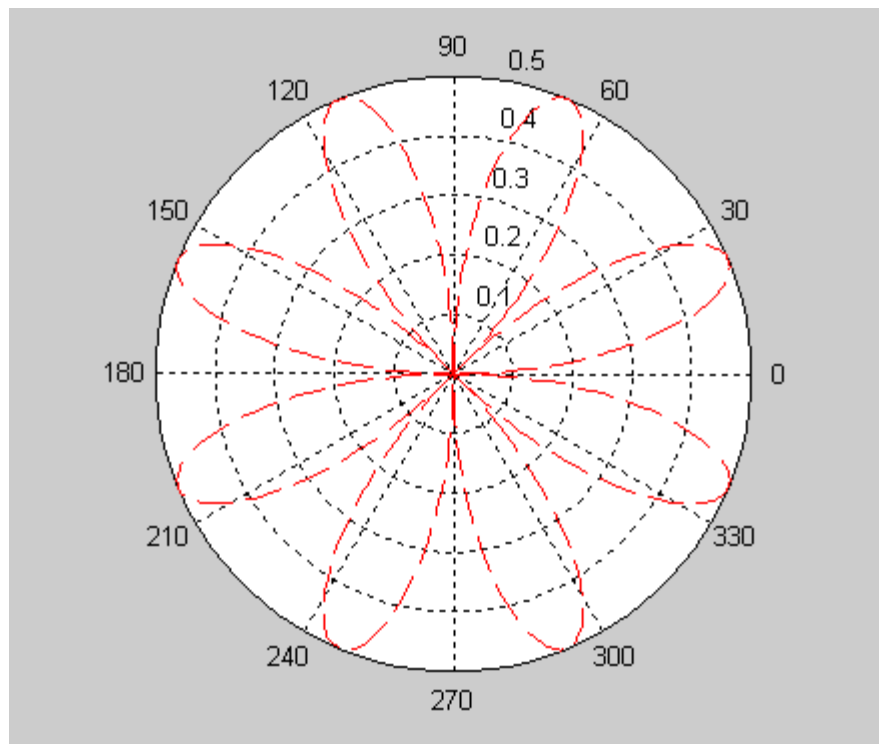
用坐标系axes_handle中绘图。

h = polar(...)

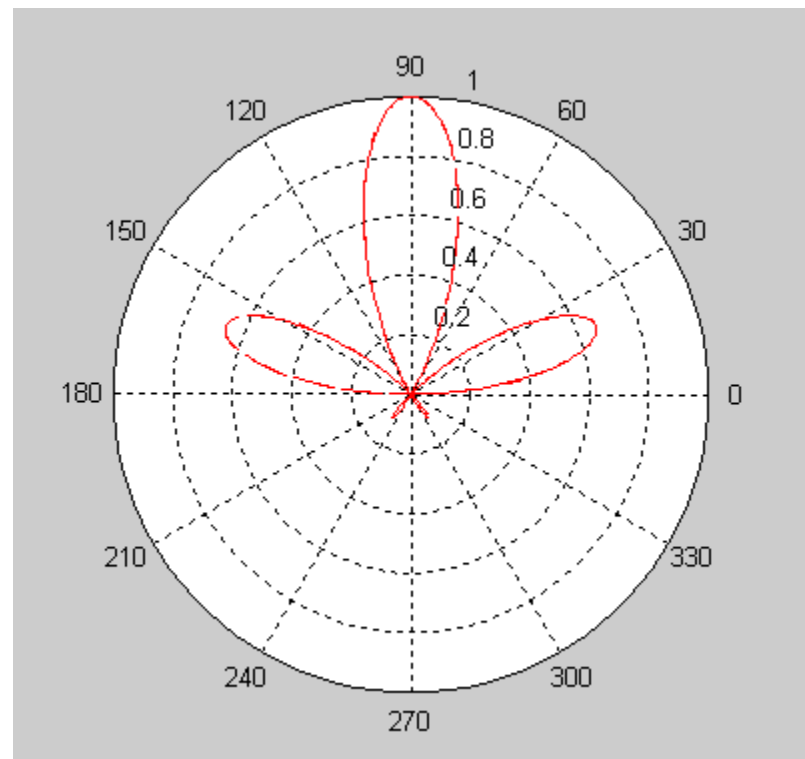
在h中返回图形对象的句柄。

polar()绘图举例

```
t = 0:.01:2*pi;  
polar(t,sin(2*t).*cos(2*t),'--r')
```

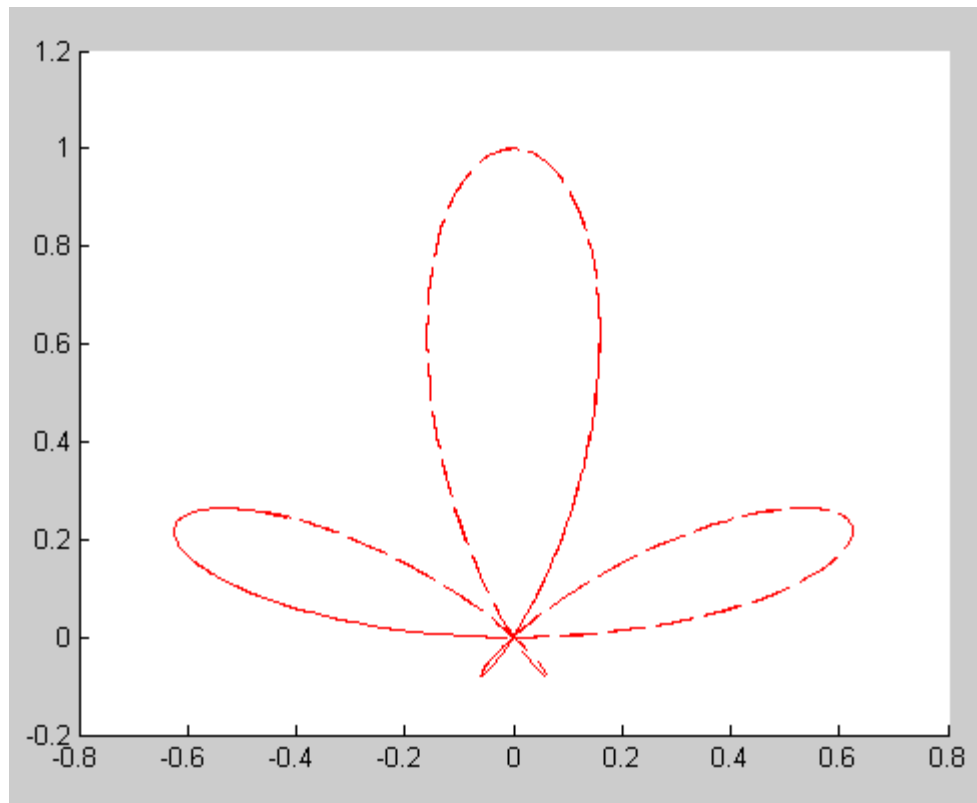


```
t = 0:.01:2*pi;  
polar(t,sin(3*t).*cos(2*t),'--r')
```



hold状态对polar()绘图的影响

```
hold on  
t = 0:.01:2*pi;  
polar(t,sin(3*t).*cos(2*t),'--r')
```

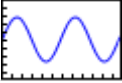
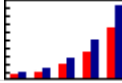
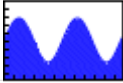
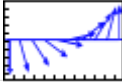

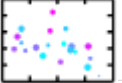
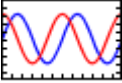
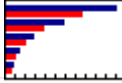

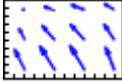

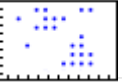
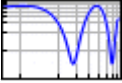
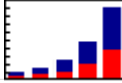

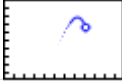

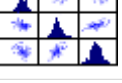
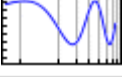
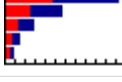
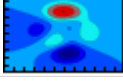

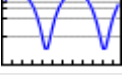

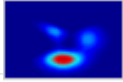


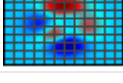
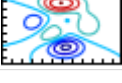
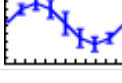
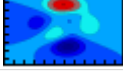
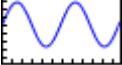
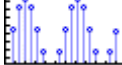


练习：在极坐标系中绘制如下函数的图形

$$\rho = \frac{100}{100 + \left(t - \frac{1}{2\pi}\right)^8} \left(2 - \sin(7t) - \frac{1}{2} \cos(30t)\right), \quad t \in \left[-\frac{1}{2}\pi, \frac{3}{2}\pi\right]$$

你能用`plot()`函数绘制出上面的极坐标图形吗？

6、二维绘图函数汇总

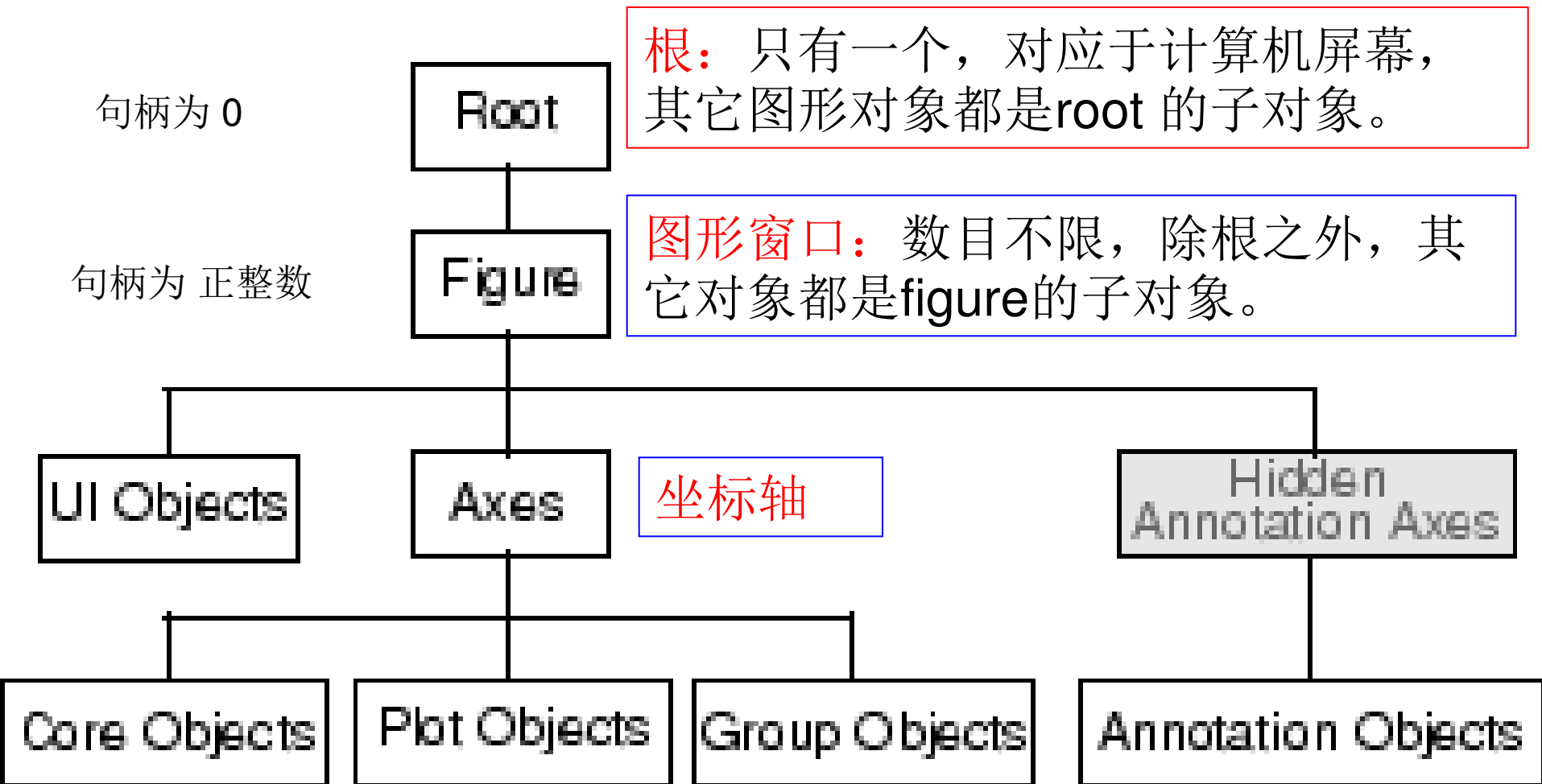
Line Graphs	Bar Graphs	Area Graphs	Direction Graphs	Radial Graphs	Scatter Graphs
plot 	bar (grouped) 	area 	feather 	polar 	scatter 
plotyy 	barh (grouped) 	pie 	quiver 	rose 	spy 
loglog 	bar (stacked) 	fill 	comet 	compass 	plotmatrix 
semilogx 	barh (stacked) 	contourf 		ezpolar 	
semilogy 	hist 	image 			
stairs 	pareto 	pcolor 			
contour 	errorbar 	ezcontourf 			
ezplot 	stem 				

二、图形属性控制

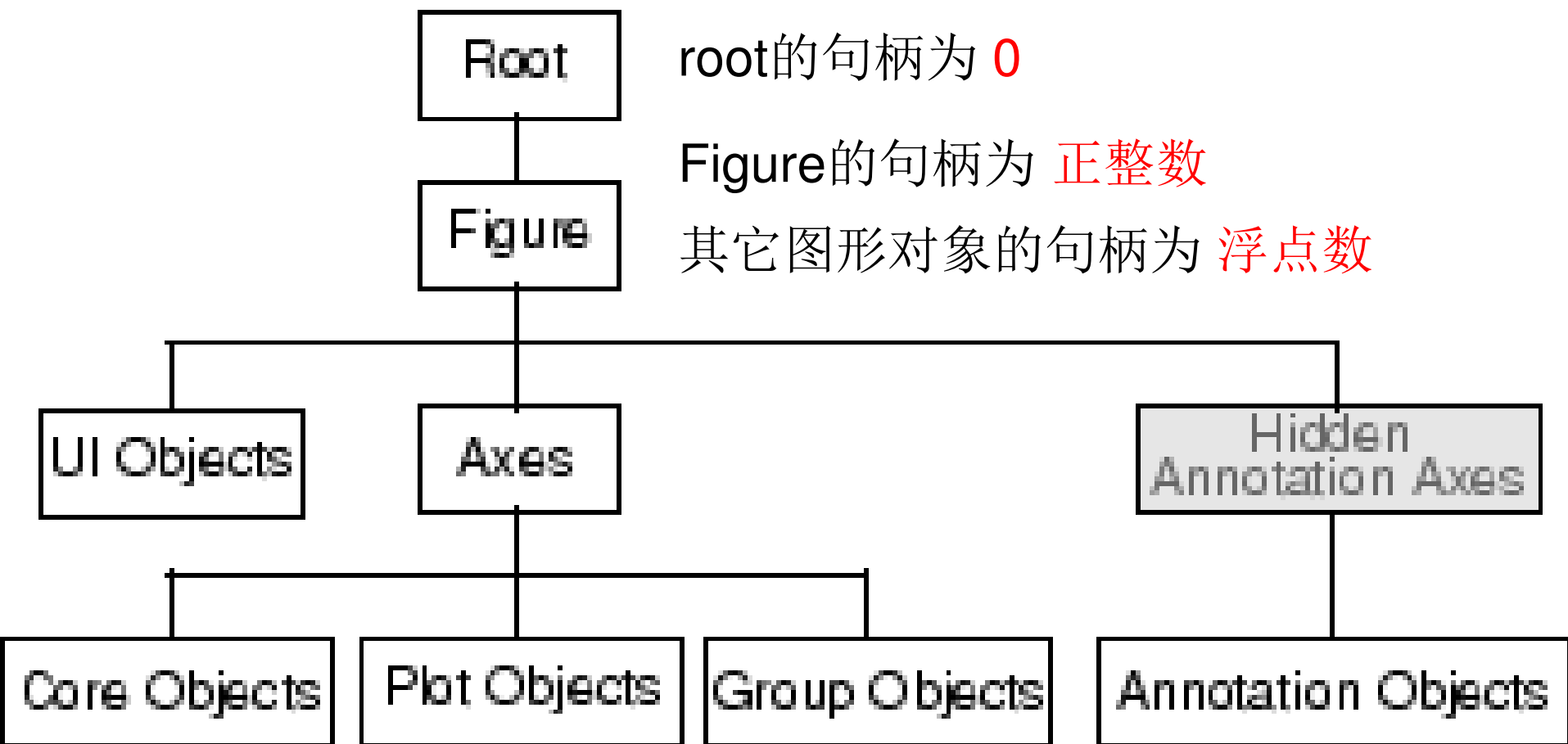
- 1、Matlab图形系统简介
- 2、根(Root)
- 3、图形窗口(Figure)
- 4、坐标轴(Axes)
- 5、保持图形: hold
- 6、subplot()
- 7、图形标注: title(), xlabel(), text(), ...

1、Matlab图形系统简介

Matlab的图形系统是面向对象的，图形对象之间的关系为父代与子代的关系



每个图形对象都拥有自己的句柄 (handle)。所有能创建图形对象的matlab函数都可给出所创建图形对象的句柄。



图形对象是由其**属性**来描述的。高层绘图函数对图形对象的描述一般是缺省的或由高层绘图函数自动设置。底层绘图函数用**句柄**设置图形对象的属性。

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

delete(h) 删除句柄为h的图形对象

专用函数

gcf 返回当前窗口对象的句柄 Get Current Figure

gca 返回当前轴对象的句柄 Get Current Axes

gco 返回当前图形对象的句柄 Get Current Object

若一个对象的句柄已知，可以获取其“父”或“子”的句柄

```
f = get(h,'parent');
```

```
l = get(h,'children');
```

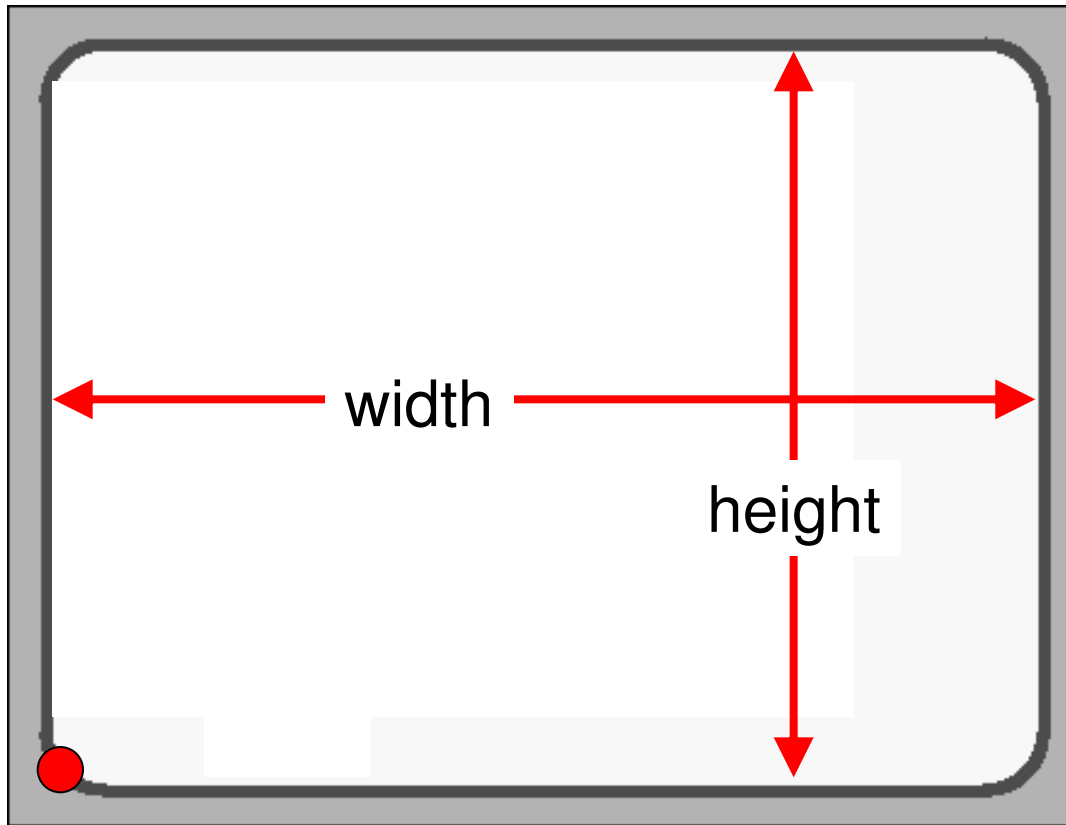
2、根对象(Root)

Root 是关联计算机屏幕的一个图形对象。Matlab系统只有一个Root象，它没有父对象，子对象是Figure。当启动Matlab时，Root对象就创建，用户无法再创建一个Root，也无法删除这个对象，Root的句柄值为 0。

root 的 属性

属性名称	意义	取值
CurrentFigure	当前图形窗口的句柄	figure的句柄
children	所有子对象的句柄	句柄数组
Units	度量单位	pixel, inches等
ScreenSize	屏幕的大小	[left,bottom,width,height]
docsearch ‘root properties’ : 在帮助中查询 root 的所有属性		

root 的 **ScreenSize** 属性



ScreenSize: 只读属性，默认单位为像素 (pixel)，由屏幕的左下角坐标、宽度、高度四个参数确定。

(left, bottom)

- 以像素为单位，左下角点坐标为(1,1)
- 以inches等为单位，左下角点坐标为(0,0)

ss = get(0,'ScreenSize')

返回行向量 ss = [left, bottom, width, height]

3、图形窗口 (Figure)

Figure对象是Matlab系统中显示的图形窗口。用户可建立任意多个**Figure**窗。所有**Figure**对象的父对象都是**Root**对象，而其他所有Matlab图形对象都是**Figure**的子对象。

figure()：创建图形窗口

close()：删除图形窗口

clf()：图形图形窗口中的子对象

gcf：返回当前窗口对象的句柄

创建图形窗口：figure

figure 利用缺省属性值来创建新的图形窗口对象。

figure('PropertyName',propertyvalue,...) 利用指定的属性值来创建图形窗口对象。对于用户没有显式地定义的属性值，将其设置为默认的属性值。

figure(h) 如果句柄h所指示的图形窗口对象存在，则将其设置为**当前窗口**，并将其移动到屏幕的最前方。如果h所指示的图形窗口不存在且h是个整数 ($h \geq 1$)，则创建一个图形窗口，并将窗口的句柄设置为h；如果h不是整数，则返回错误信息。

h = figure(...) 返回图形窗口对象的句柄。

为了在一个已有的图形窗口中绘制图形，这个窗口必须是**激活**的，或者是**当前的**图形窗口。

删除图形窗口：close

- 1、**close** 删除当前figure，相当于close(gcf)
- 2、**close(h)** 删除由h确定的figure。如果h是一个向量或矩阵，就删除由h指定的所有图像
- 3、**close name** 删除指定名称的figure
- 4、**close all** 删除所有句柄没有隐藏的figure
- 5、**close all hidden** 删除所有figure，包括句柄隐藏的。
- 6、**status = close(...)** 如果指定的figure已经被删除则返回1，否则为0。

清除图形窗口中的子对象： `clf`

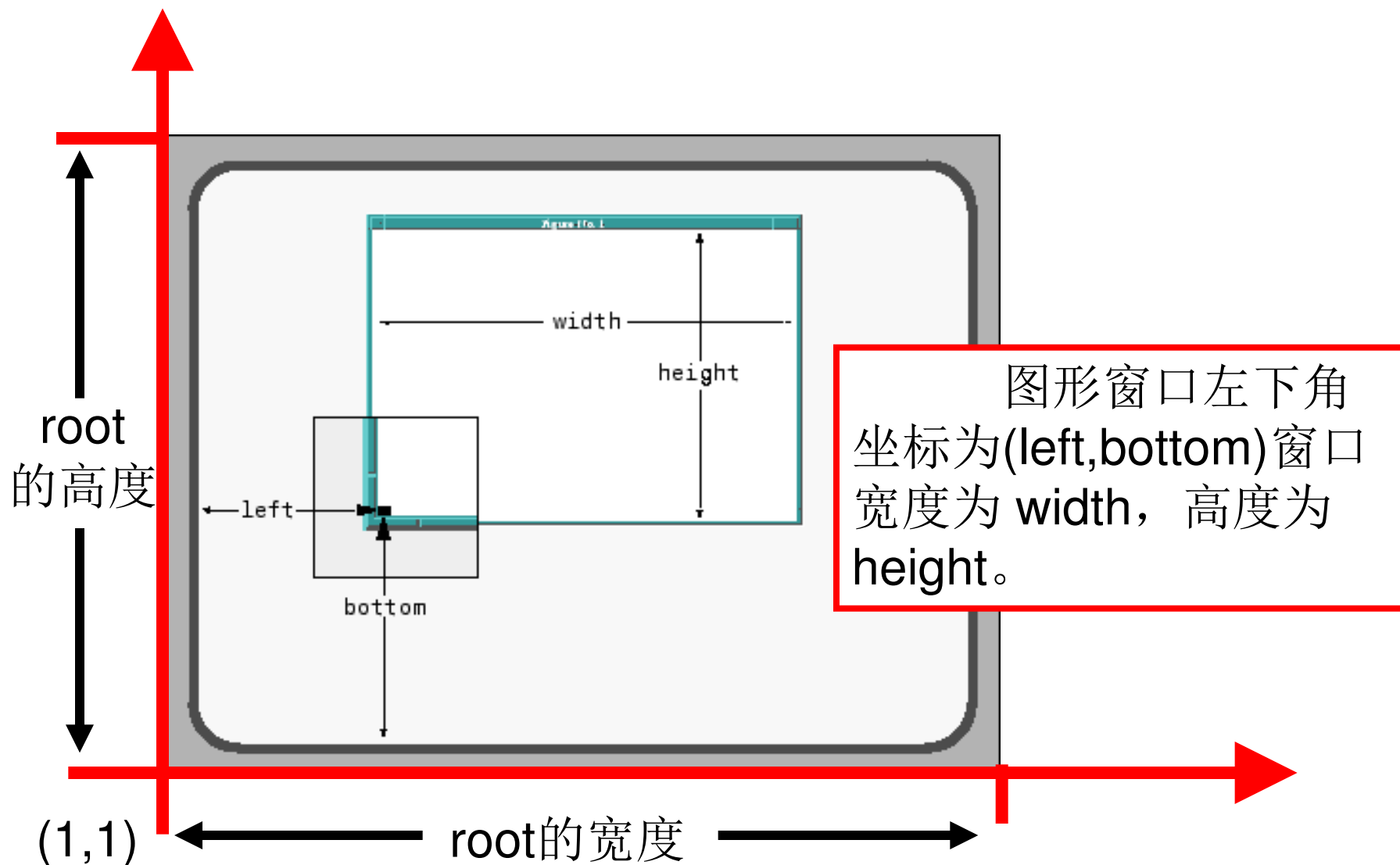
- 1、`clf` 删除当前图形窗口中、句柄未被隐藏(即它们的 `HandleVisibility` 属性为 `on`) 的图形对象。
- 2、`clf('reset')` 或 `clf reset` 删除当前图形窗口中的所有图形对象，无论其句柄是否被隐藏，同时将图形窗口的属性(除 `Position`, `Units`, `PaperPosition`, `PaperUnits` 外)恢复为默认值。
- 3、`clf(fig)` 或 `clf(fig, 'reset')` 清除由句柄为 `fig` 的图形窗口中的内容。
- 4、`figure_handle = clf(...)` 返回图形窗口的句柄，

Figure 的一些属性

属性名称	意义	取值
Color	背景色	颜色字符串或[r g b]
CurrentAxes	当前坐标轴句柄	坐标轴句柄
CurrentObject	当前对象的句柄	对象句柄
HandleVisibility	句柄是否可见	{on} callback off
NumberTitle	figure的数字标识	‘ on off ’
Name	名称	字符串
NextPlot	新图如何输出	new {add} replace replacechildren
Position	窗口位置、大小	[left, bottom, width, height]
Resize	窗口大小是否可调	‘ on off ’
Toolbar	开/关工具条	‘ none figure auto ’
MenuBar	开/关菜单	‘ none figure auto ’
Tag	“文字标识”	字符串
WindowStyle	窗口显示模式	‘ normal modal docked ’
Visible	控制可视性	‘ on off ’

figure 的 Position 属性

Position由行向量 [left, bottom, width, height] 确定



例：figure的Position, Color, ... 属性

```
x = 0:pi/10:2*pi

ss = get(0, 'ScreenSize')
W = ss(3);
H = ss(4);

figure(1)
set(gcf, 'Position', [W/2, H/2, W/3, H/4])
set(gcf, 'Name', 'sin(x) 的图像', 'NumberTitle', 'off')
plot(x, sin(x))

h = figure
set(h, 'Position', [W/2, H/10, W/3, H/4])
set(h, 'Name', 'cos(x) 的图像', 'Color', [1, 1, 1])
set(1, 'Toolbar', 'none', 'Menubar', 'none')
plot(x, cos(x))
```

figure 的 NextPlot 属性

new	创建一个新的图形窗口并作为当前窗口
add	在当前窗口中绘图，不清除任何已存在的图形对象
replacechildren	删除所有HandleVisibility属性为on的子对象，并重置figure的NextPlot属性 值为add，这相当于调用clf命令
replace	移除所有子对象，不管其HandleVisibility属性值，并重置figure的属性值为默认值，但Position、units、PaperPosition和PaperUnits属性值不被重置。NextPlot属性重置为add。这相当于调用clf reset命令。

```
x = 0:0.1:7
```

```
h = figure
```

```
plot(x,sin(x))
```

```
x = 0:0.1:7
```

```
h = figure
```

```
set(h,'NextPlot','new')
```

```
plot(x,sin(x))
```

4、坐标轴 (axes)

Axes对象：在图形窗口中定义一个画图区域，父对象是**Figure**，子对象包括**line**对象、**image**对象、**patch**对象、**surface**对象、**light**对象和**text**对象等。

常用函数

axes(): 创建坐标轴

cla: 清除坐标轴中的子对象

gca: 返回当前轴对象的句柄

axis(): 设置坐标轴的常见属性

xlim() / ylim() / zlim(): 设置x / y / z 轴刻度范围

grid: 设置坐标轴网格线的显示

box: 设置坐标轴边框的显示

xlabel() / ylabel() / zlabel(): 设置x / y / z轴的标注

创建坐标轴 axes()

axes 在当前figure内使用默认属性值来创建一个坐标轴对象。

axes('PropertyName',propertyvalue,...) 创建具有指定的属性值的坐标轴对象。对于其它未明确指定的属性值，MATLAB采用这些属性的默认值。

axes(h) 当句柄为h的坐标轴对象存在时，设置该坐标轴为**当前坐标轴**，并使含有该坐标轴的图形窗口对象获得焦点。同时，将该坐标轴的句柄h 置于图形窗口的Children属性值的第一位，并把图形窗口的CurrentAxes属性值设置为h。**当前坐标轴**为用户用来绘制图像、线条、补片、矩形、曲面以及文本图形对象的目标坐标轴。如果用户想将坐标轴设置为当前坐标轴同时又不改变图形窗口的状态，则可以设置图形窗口的CurrentObject属性：**set(figure_handle,'CurrentAxes',axes_handle)**，这在一个图形窗口被最小化或者在其它窗口的后面，而用户又想设置它所包含的坐标轴为当前坐标轴时是非常有用的。

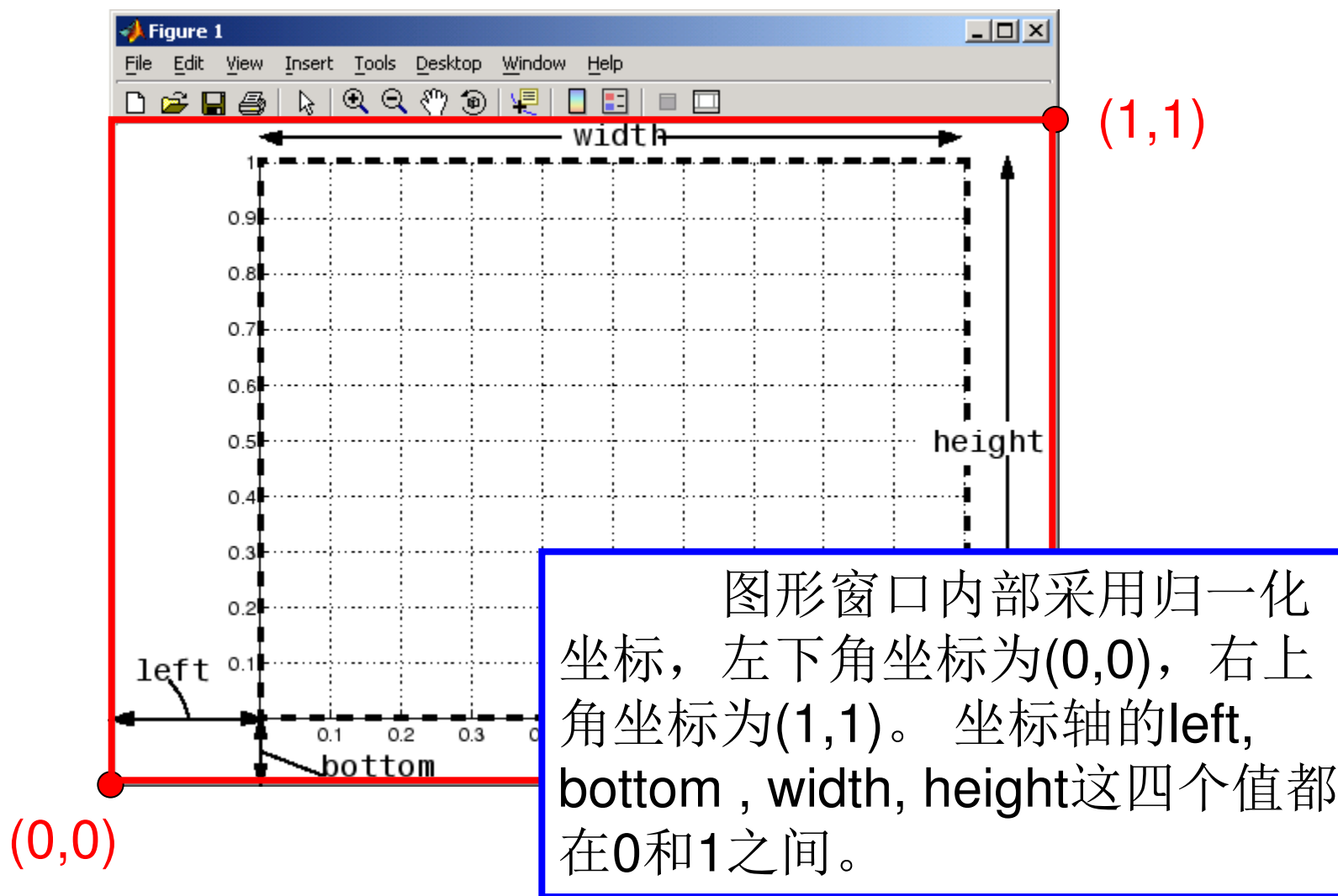
h = axes(...) 返回创建的坐标轴对象的句柄。

axes 的一些属性

属性名称	意义	取值
Color	背景色	[r g b]
ColorOrder	颜色序列	[r1 g1 b1; r2 g2 b2; ...]
LineStyleOrder	线型序列	字符串 - * : o
NextPlot	新图如何绘制	add {replace} replacechildren
Position	指定位置、大小	[200 200 1000 400]
LineWidth	坐标轴线宽	数值
TickDir	刻度线的方向	' in out '
TickLength	刻度线的长度	[2DLength 3DLength]
docsearch 'axes properties': 在帮助中查询 axes 的所有属性		

axes 的 Position 属性

Position由行向量 [left, bottom, width, height] 确定



axes 的 NextPlot 属性

add	在当前坐标轴内绘图，保留所有已存在的图形对象
replacechildren	删除所有HandleVisibility属性为on的子对象，但不重置坐标轴的属性。 这相当于调用cla命令
replace	移除所有子对象，不管其 HandleVisibility 属性值，并重置 figure 的属性值为默认值，但 position 和 Units 属性除外。 这相当于 cla reset命令

例：设置坐标轴属性

figure

```
h1 = axes('position',[0.08,0.1,0.4,0.4])
```

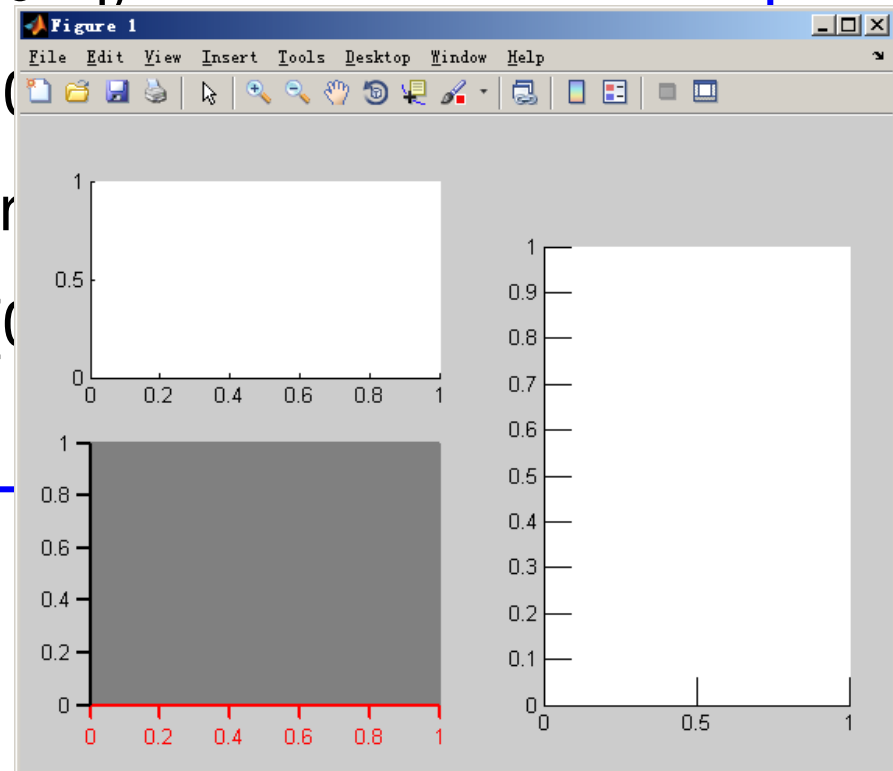
```
h2 = axes('position',[0.6,0.1,0.35,0.7])
```

```
h3 = axes('position',[0.08,0.6,0.4,0.4])
```

```
set(h1,'Color',[0.5,0.5,0.5],'XColor','r')
```

```
set(h1,'TickDir','out','TickLength',[0.06,0.02])
```

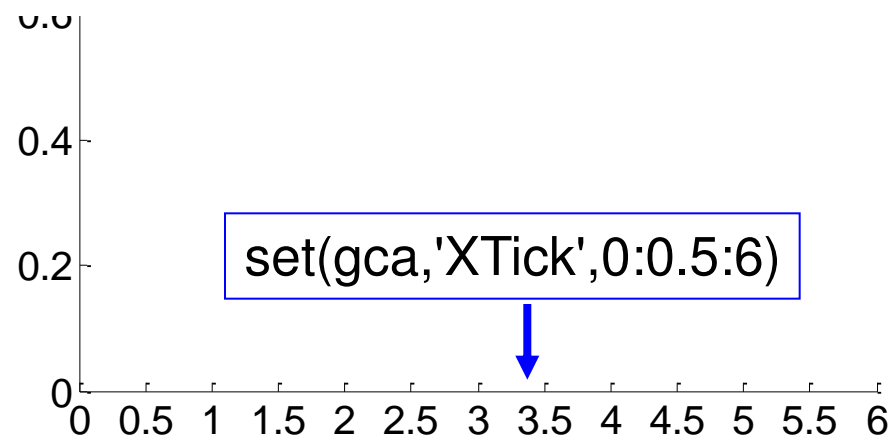
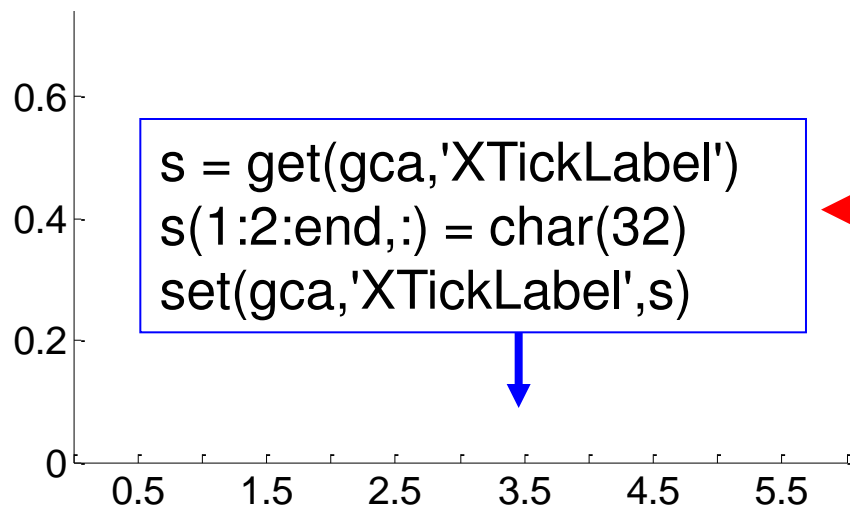
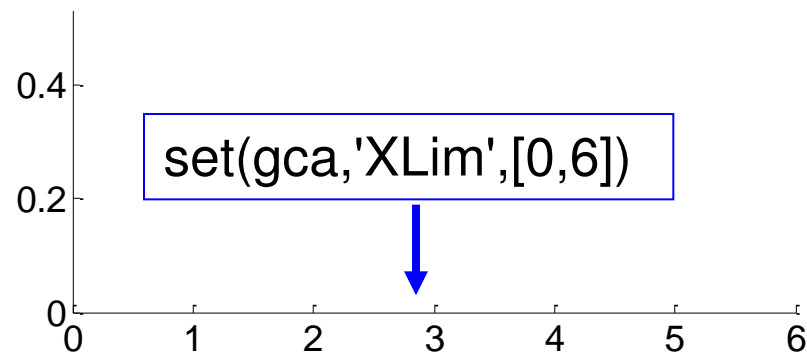
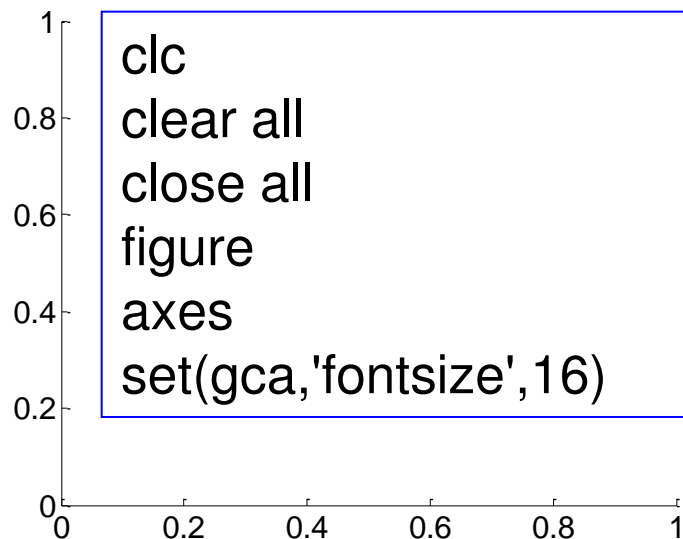
```
set(h2,'TickLength',[0.06,0.02])
```



Properties That Control the X-, Y-, or Z-Axis

XColor, YColor, ZColor	Color of axis lines
XDir, YDir, ZDir	{normal} reverse
XGrid, YGrid, ZGrid	on {off}
XLabel, YLabel, ZLabel	handle of text object
XLim, YLim, ZLim	[minimum maximum]
XLimMode, YLimMode, ZLimMode	{auto} manual
XMinorGrid, YMinorGrid, ZMinorGrid	on {off}
XMinorTick, YMinorTick, ZMinorTick	on {off}
XScale, YScale, ZScale	{linear} log
XTick, YTick, ZTick	vector of data values locating tick marks
XTickLabel, YTickLabel, ZTickLabel	string
XTickMode, YTickMode, ZTickMode	{auto} manual
XTickLabelMode, YTickLabelMode, ZTickLabelMode	{auto} manual

例：XLim, XTick, XTickLabel设置



清除坐标轴中的子对象： `cla`

`cla`: 删除当前坐标系下的所有未设置隐藏的图形对象。

`cla reset` : 删除当前坐标系下所有图形对象，不论是否设置了隐藏。同时将**`axes`**的属性(除**`Position`**, **`Units`**外)设置为默认值。

`cla(ax)` 或 **`cla(ax,'reset')`** : 清除由句柄**`ax`**所指坐标系。

axis()：设置坐标轴的常用属性

v = axis 返回包含当前坐标范围的行向量v，v具有四个或六个元素，这取决于当前坐标轴是二维的还是三维的。

axis([xmin, xmax, ymin, ymax]) **2d**

axis([xmin, xmax, ymin, ymax, zmin, zmax]) **3d**

人工设置坐标范围，允许取 inf 或 -inf，即坐标上限或下限时自动产生的，即坐标范围“半自动确定”。

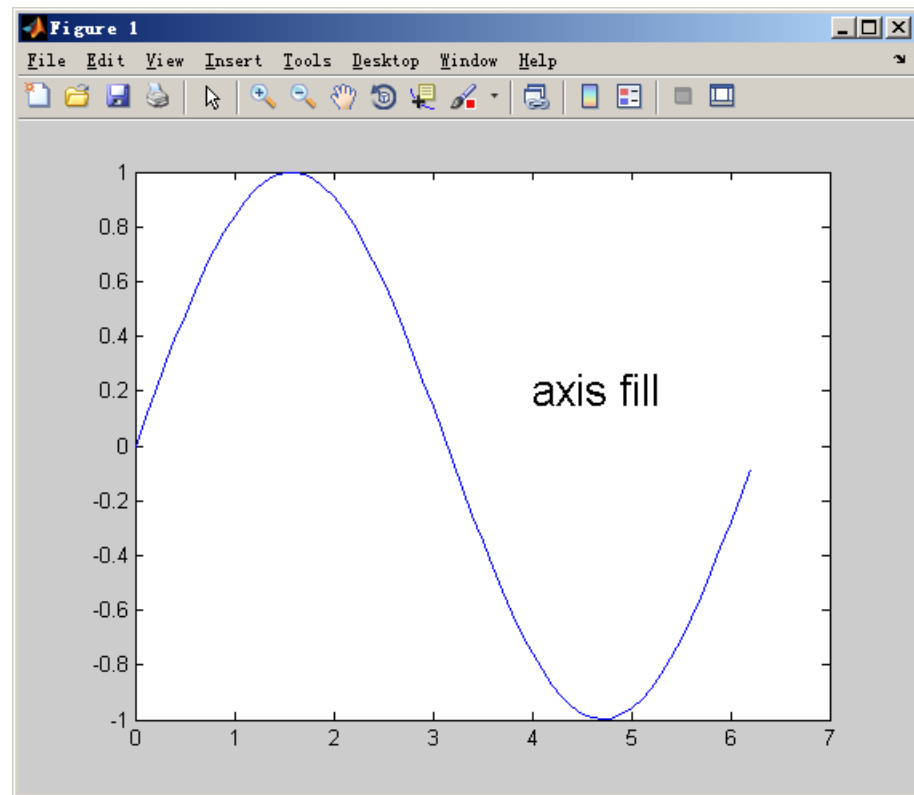
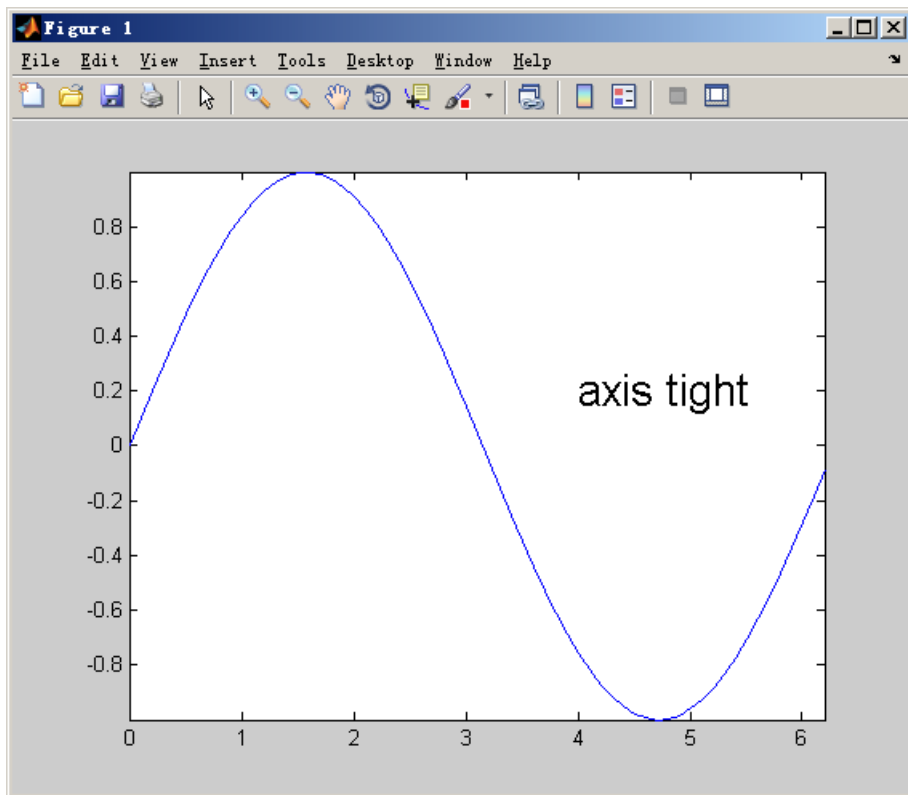
axis auto 基于x、y和z的最大值和最小值来**自动设置坐标轴范围**。用户可以只限定某一个坐标轴，如：axis 'auto x' 用来限定x轴的范围；axis 'auto yz' 用来限定y轴和z轴的范围。

axis manual和**axis(axis)** **冻结当前坐标轴范围**。如果打开了hold on命令，则后续的图形都使用同样的坐标范围。该函数设置XLimMode、YLimMode和ZLimMode属性为manual。

axis tight 采用紧密模式设置当前坐标轴范围，即以用户数据范围为坐标轴范围。

axis fill 设置坐标轴范围及PlotBoxAspectRatio属性使得坐标轴填充由 position 确定的矩形局域。

只有PlotBoxAspectRatioMode 或 DataAspectRatioMode 属性值为 manual时，该方法才起作用。

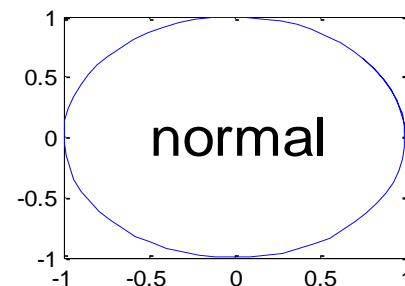
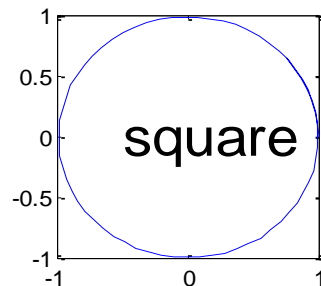
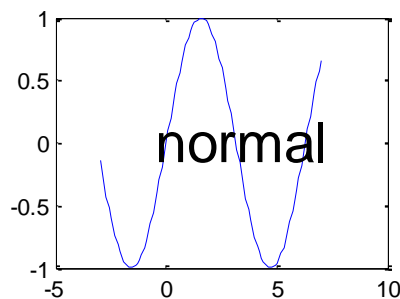
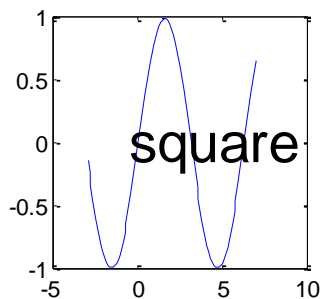
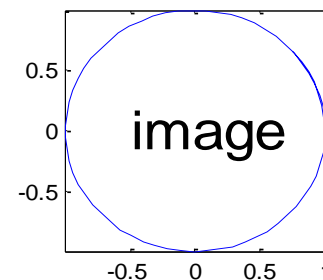
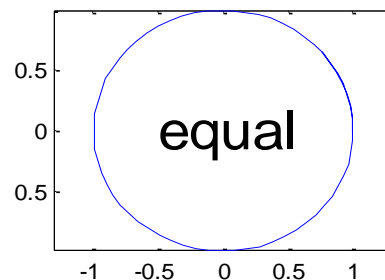
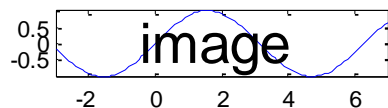
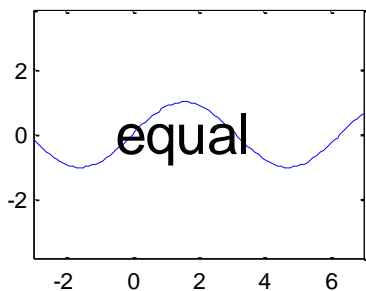


axis equal 设置当前坐标轴的横纵轴具有相同的单位长度。

axis image 功能与**axis equal**相同。但坐标轴的边框紧贴在数据的四周。多用来显示图片。

axis square 调整坐标系的x、y和z轴，使它们有相同的长度。

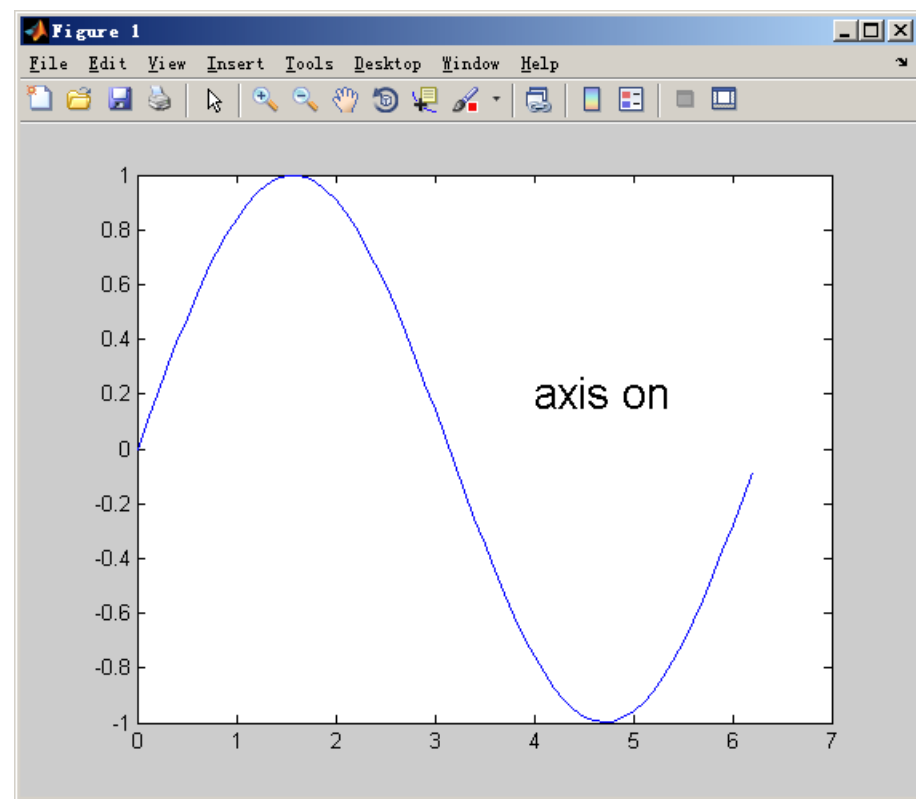
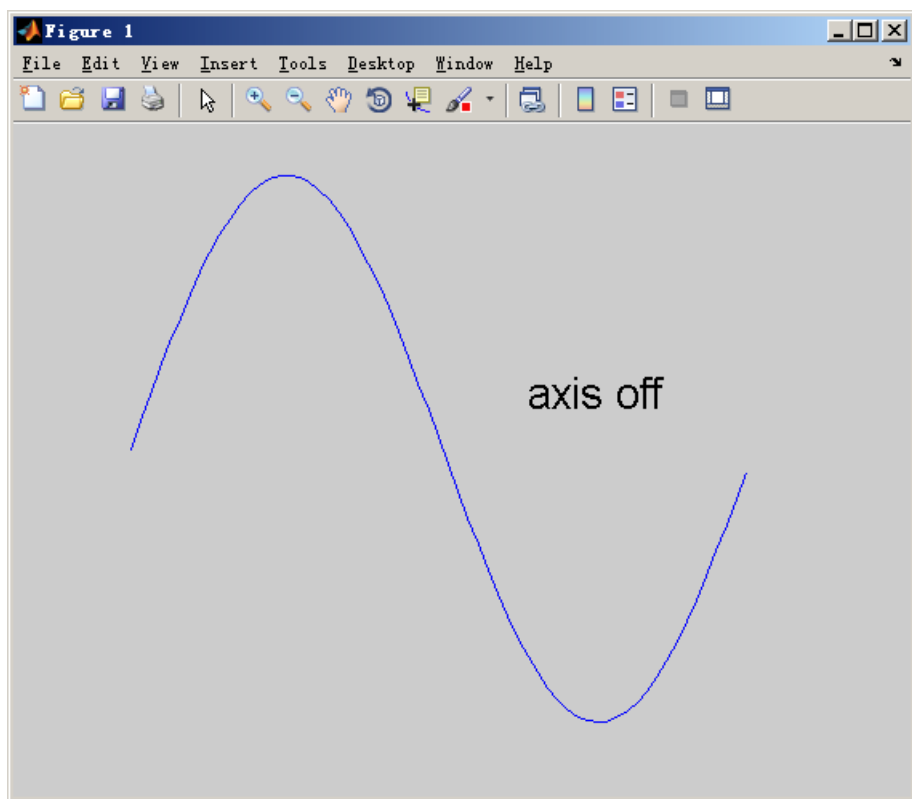
axis normal 自动调整横纵轴比例，使图形显示达到最佳效果



axis vis3d 锁定坐标轴的纵横比，避免三维旋转时变化。

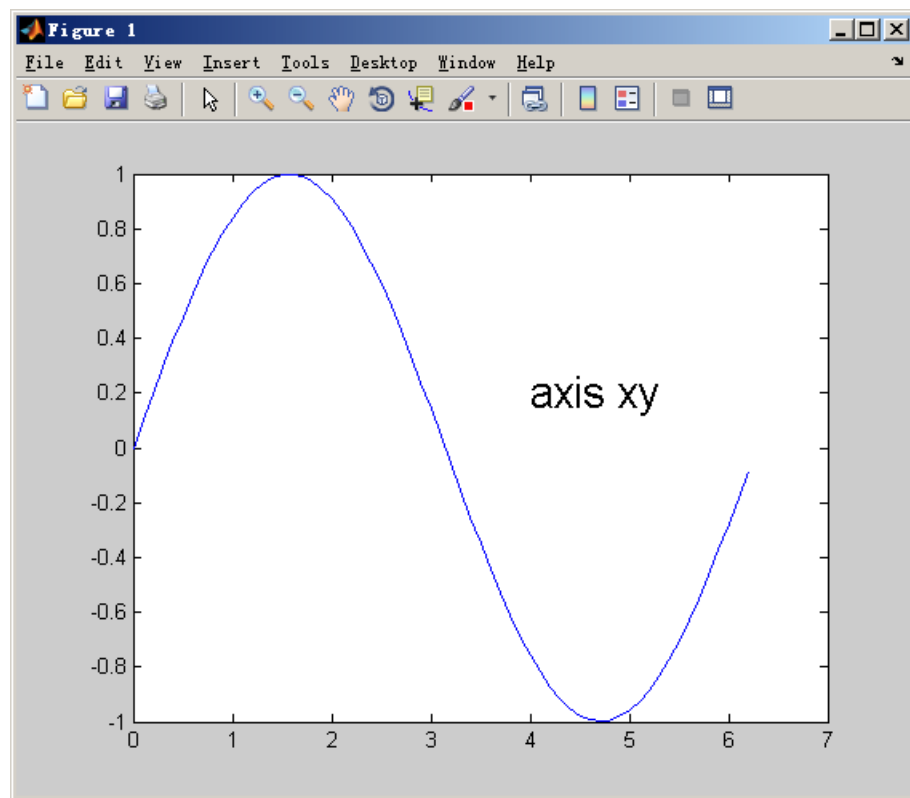
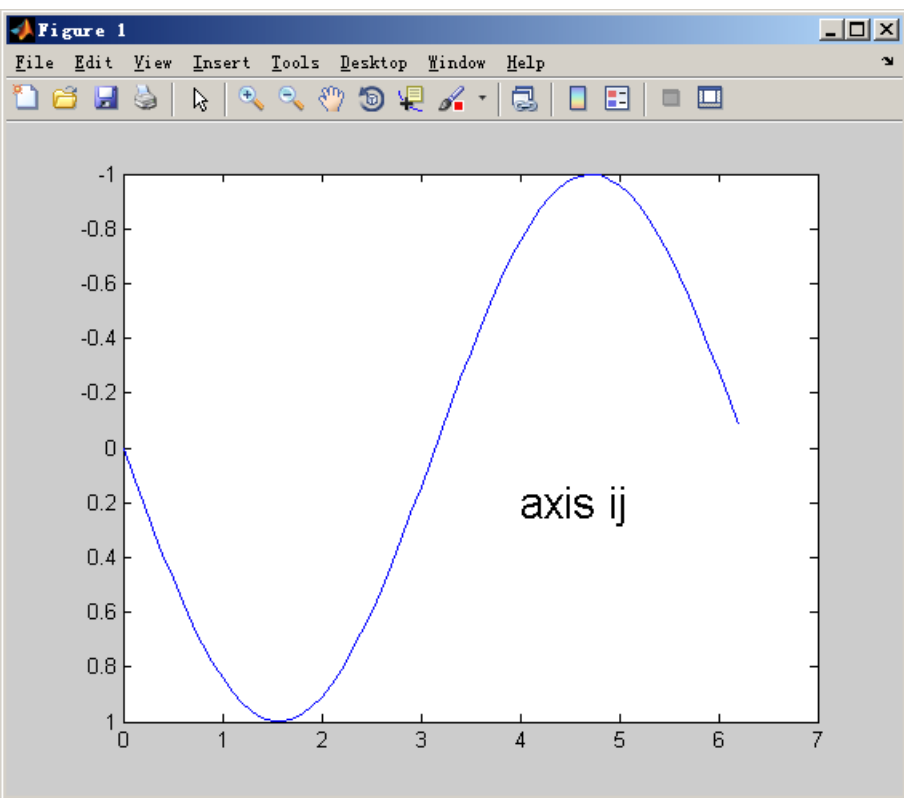
axis off 关闭所有坐标轴线、刻度标记和标签。

axis on 打开所有坐标轴线、刻度标记和标签。



axis ij 坐标系的原点设置到左上角。 i 轴为垂直轴，正方向为从上到下。 j 轴为水平轴，正方向为从左到右。

axis xy 坐标系的原点在左下角。 x 轴为水平坐标轴，正方向为从左到右， y 轴为垂直坐标轴，正方向为从下到上。



axis(axes_handles,...) 将**axis**命令应用于指定的坐标轴对象。
例如，下面的代码将两个坐标轴都设置为**square**

```
h1 = subplot(221);  
h2 = subplot(222);  
axis([h1 h2],'square')
```

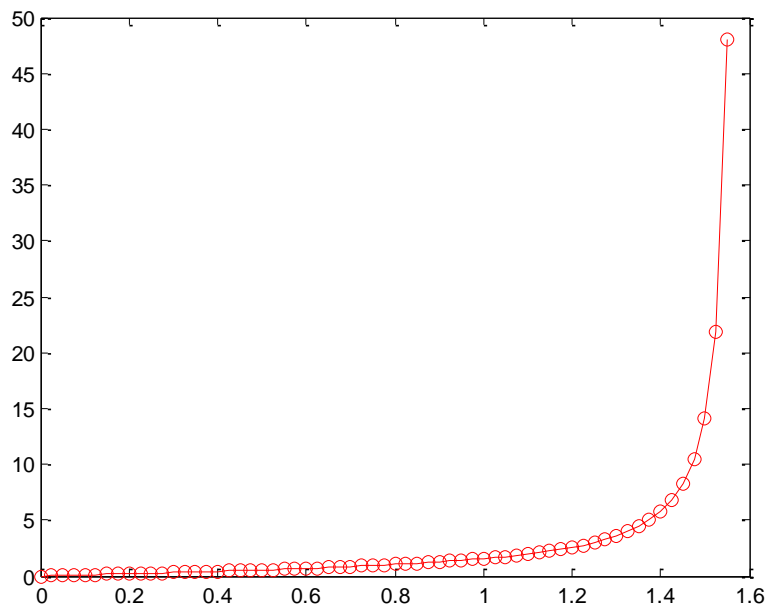
[mode,visibility,direction] = axis('state') 返回三个字符串，用来表示坐标轴的当前属性值。

mode	' auto ' ' manual '
visibility	' on ' ' off '
direction	' xy ' ' ij '

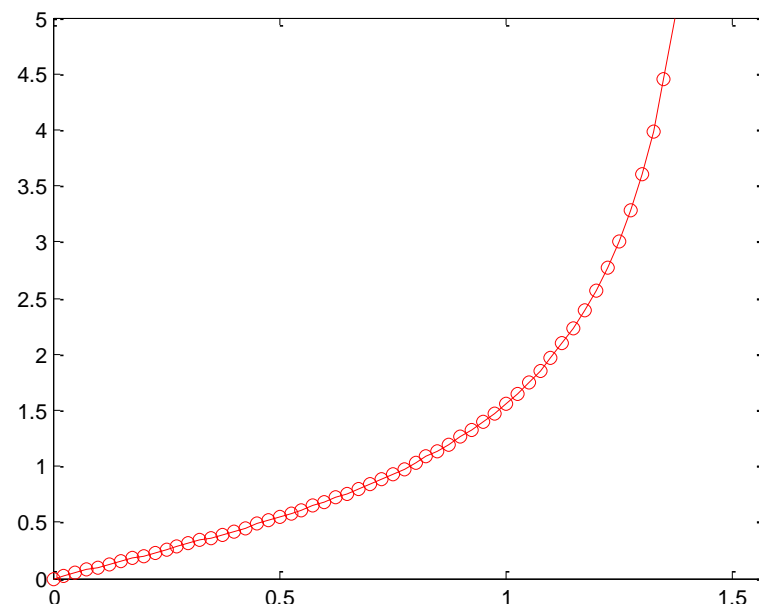
如果XLimMode、YLimMode和ZLimMode都设置为auto，则mode的值为auto;如果XLimMode、YLimMode和ZLimMode中有一个的值设为manual，则mode的值为manual。

axis()应用举例

```
x = 0:.025:pi/2;  
plot(x,tan(x),'-ro')
```



```
x = 0:.025:pi/2;  
plot(x,tan(x),'-ro')  
axis([0 pi/2 0 5])
```



xlim() / ylim() / zlim()

当用户只想改变一个坐标轴的坐标限的时候，axis命令就显得很麻烦，这时可以使用xlim、ylim和zlim函数进行设置。

xlim : 返回包含x坐标范围的行向量

xlim([xmin xmax]) : 设置x的坐标范围

xlim('mode') : 返回x坐标的设置模式 (auto 或 manual)

lim('auto') : 设置x坐标的模式为 auto.

xlim('manual') : 设置x坐标的模式为 manual.

xlim(axes_handle,...) : 设置指定的坐标轴的x坐标范围

grid

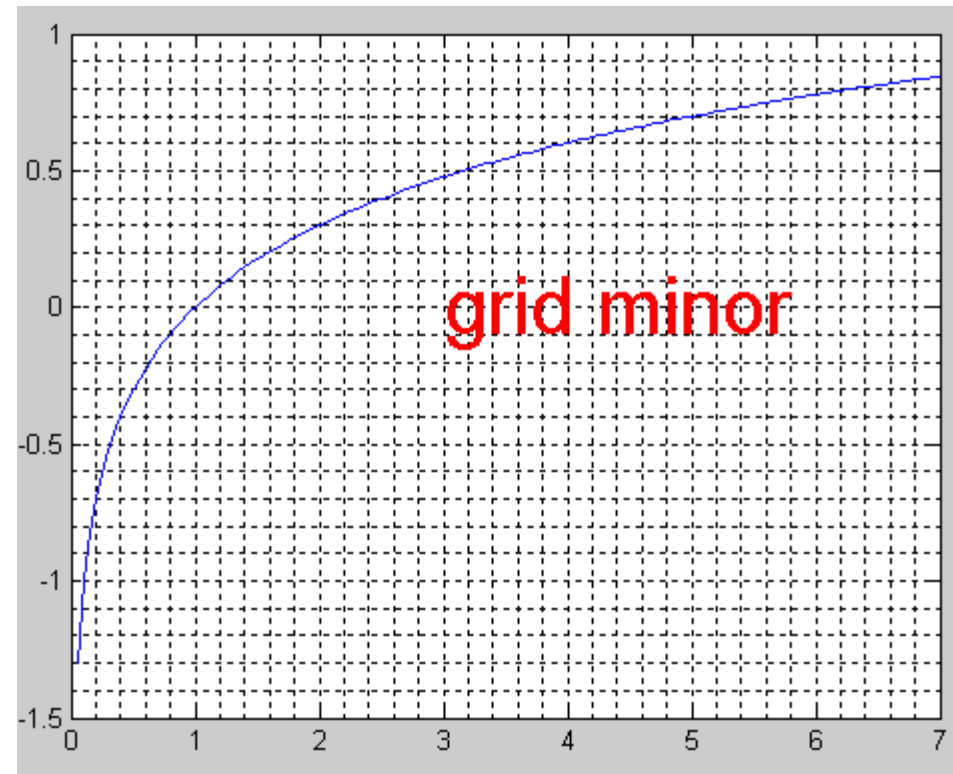
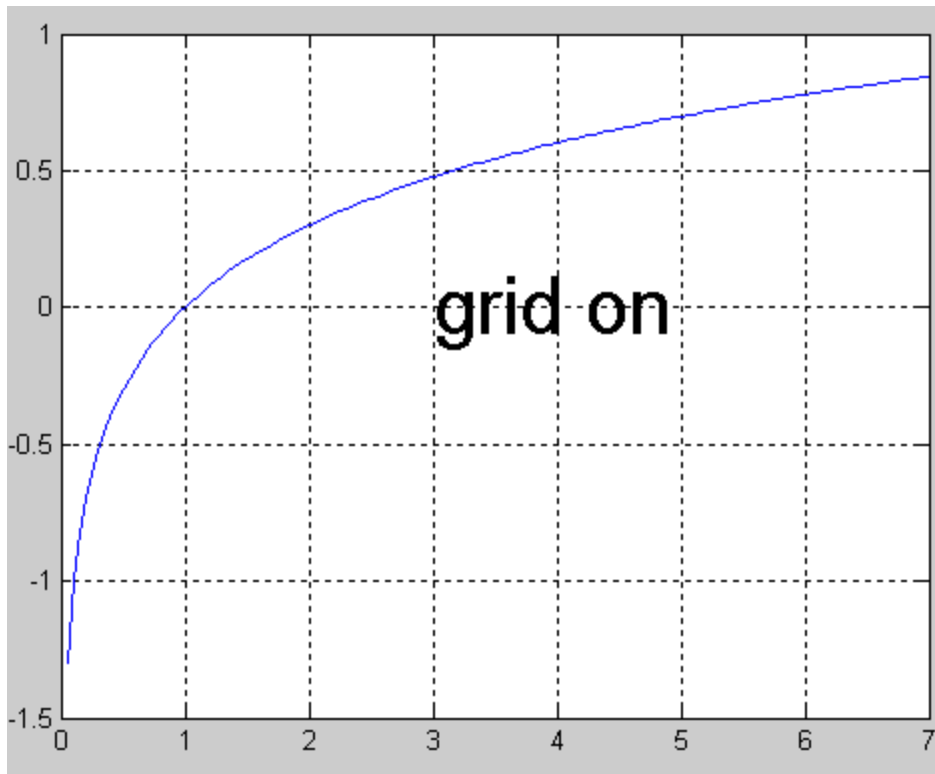
grid on: 对当前坐标系添加主要网格线。

grid off: 对当前坐标系移除主要/细节网格线。

grid : 在grid on / grid off两个状态间切换。

grid(axes_handle,...) 对指定的坐标系设置网格线是否显示

grid minor : 对当前坐标系设置添加细节网格



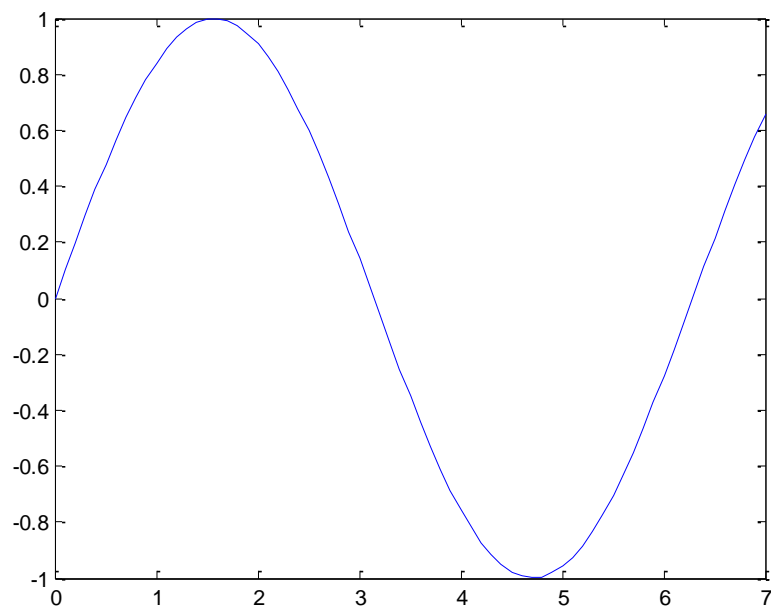
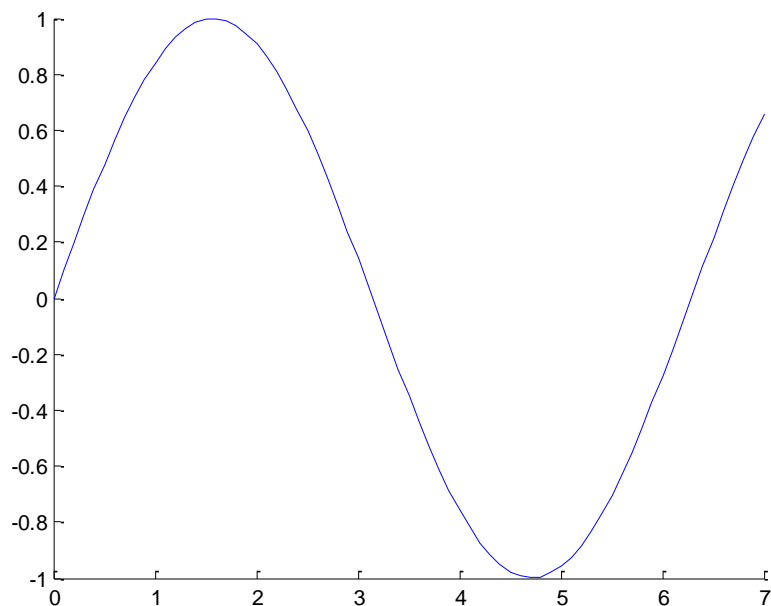
box

box on : 当前坐标系的坐标轴框边界。

box off : 当前坐标系不显示坐标轴边框。

box: 切换当前坐标系坐标轴边框的显示。

box(axes_handle,...) : 对指定的坐标系显示坐标轴开关。



5、保持图形：hold

hold on: 启动图形保持功能，当前坐标轴和图形都将保持，此后绘制的图形都将添加在这个图形的基础上，并且自动调整坐标轴的范围。**hold on** 就是将当前窗口和当前坐标轴的‘NextPlot’属性设置为‘add’。

hold all: 启动图形保持功能，但不重置`ColorOrder`和`LineStyleOrder`两个属性。

hold off: 关闭图形保持功能，将当前的窗口中的图形释放，用以绘制新的图形。**hold off** 就是将当前坐标轴的‘NextPlot’属性设置为‘replace’。

hold: 在hold on/off两种状态之间进行切换。

hold(axes_handle,...)：在句柄指定的坐标轴上设置hold状态

若不存在axes，hold命令会创建一个axes

ishold: 查询hold当前是on还是off，是hold on的话就返回逻辑1

例：使用hold绘制多个图形

```
x = 0.1:0.5:4
```

```
y = sqrt(x)
```

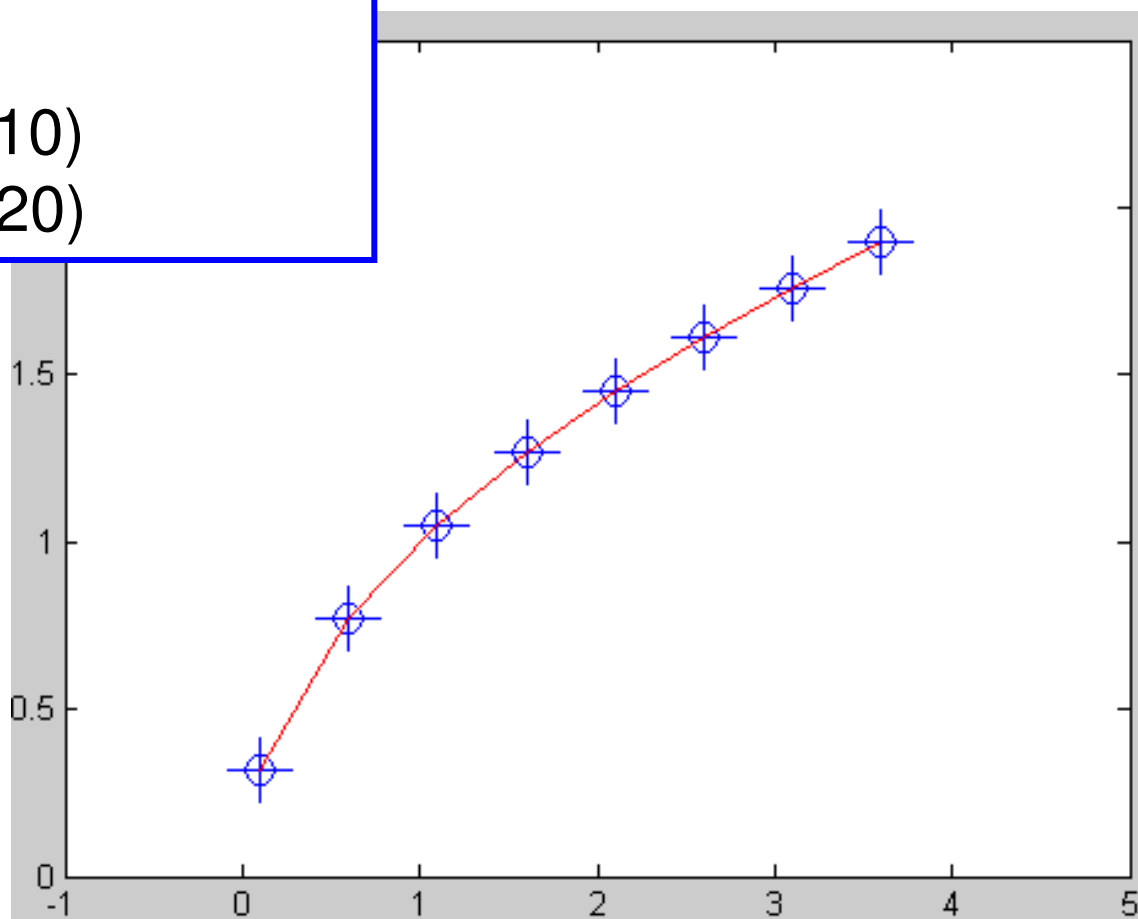
```
plot(x,y,'r')
```

```
axis([-1,5,0,2.5])
```

```
hold on
```

```
plot(x,y,'bo','markersize',10)
```

```
plot(x,y,'b+','markersize',20)
```



例题： 利用hold实现简单的动画

用`plot()`函数动态演示
利萨如图形的轨迹。

$$x = \sin(t)$$

$$y = \sin\left(2t + \frac{\pi}{6}\right)$$

.....

.....

```
plot(x([1,2]),y([1,2]))
```

```
hold on
```

```
axis([-1.1,1.1,-1.1,1.1])
```

```
for k = 2:length(t)-1
```

```
    plot(x([k,k+1]),y([k,k+1]))
```

```
    pause(0.1)
```

```
end
```

上机练习

1、绘制衰减振荡曲线 $y = e^{-t/3} \sin 3t$ 及它的包络线 $y_0 = \pm e^{-t/3}$, $t \in [0, 4\pi]$ 。

2、绘制林纳德-琼斯分子势能模型曲线：

林纳德-琼斯分子势能模型为 $E_p = \frac{\lambda}{r^n} - \frac{\mu}{r^m}$ ($n > m$)。其中, λ, μ 为正的常数。多数情况下,

n 取 9 - 15, m 取 4 - 7 之间的整数。公式中的第一项 $\frac{\lambda}{r^n}$ 表示斥力势能, 第二项 $-\frac{\mu}{r^m}$ 表示

引力势能。取 $n=15, m=7, \lambda=\mu=0.01, 0 < r \leq 2$, 绘制该分子势能模型的势能曲线、

斥力势能曲线和引力势能曲线。要求如下：

- (1) 曲线要光滑；
- (2) 三条曲线画在同一个图形窗口的同一个坐标轴中；用不同的颜色区分三条曲线，分子势能曲线用实线显示，斥力势能曲线和引力势能曲线用虚线显示；
- (3) 请将横、纵坐标轴的标签分别设置为 r 和 E_p ；
- (4) 纵坐标的显示范围设置为 $[0.8, 2, -0.01, 0.03]$ 。

第一题

```
t=0:0.1:4*pi
```

```
y=exp(-t/3).*sin(3*t)
```

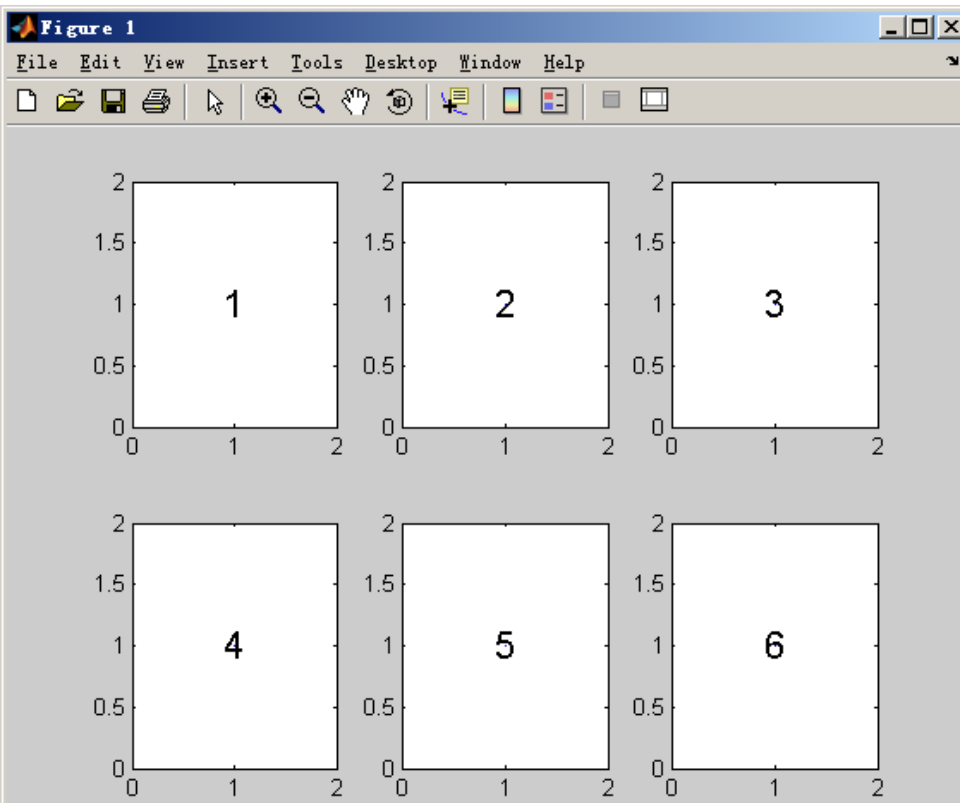
```
yo=exp(-t/3)
```

```
y1=-exp(-t/3)
```

```
plot(t,y,'r-',t,yo,'y--',t,y1,'k-*')
```

6、subplot()

subplot(m,n,p) 将一个绘图窗口分割成 m 行 n 列共 $m \times n$ 个矩形绘图区域（每个区域有自己的坐标轴）并将第 p 个绘图区域选定为当前的绘图区域。编号从最上边一行开始，从左至右、从上至下依次编号。



subplot(1,1,1) 回到默认的模式(整个图形窗口中只用一套坐标轴)

当一个新的subplot命令改变了图形窗口中绘图区域数目的时候，原先的子图就被擦除掉。

subplot(m,n,p,'replace') 如果指定的**axes**已存在，则删除它，创建一个新的**axes**。

subplot('Position',[left bottom width height]) 在当前图形窗口指定的**Position**上画图（创建坐标轴）

subplot(..., prop1, value1, prop2, value2, ...) 创立坐标系时，同时设置坐标系的相关属性。

h = subplot(...) 返回坐标系的句柄。

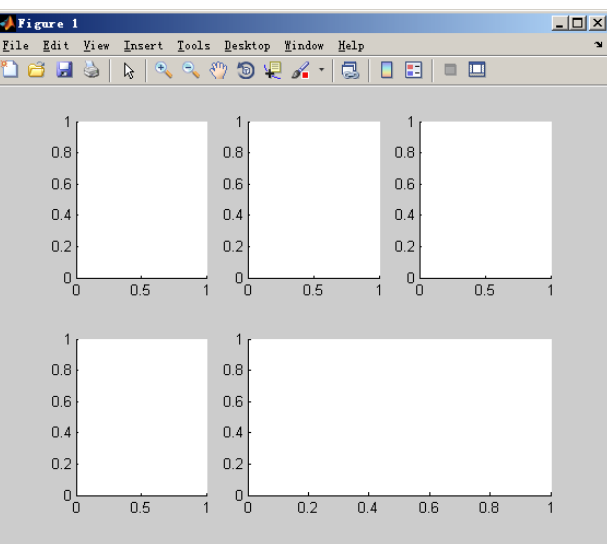
h = subplot(m,n,p) 返回第**p**个坐标轴的句柄

subplot(h) 设置句柄**h**对应的坐标轴为当前坐标轴。

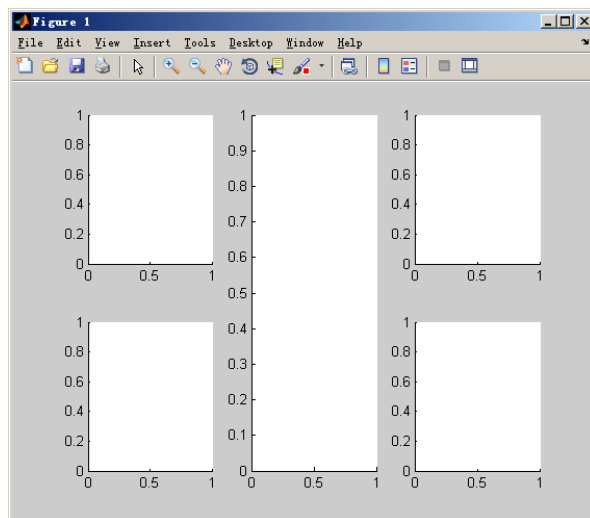
subplot(m,n,P)

P是一个向量，表示向量**P**中指定的若干画图区域连成一个整体，包括那些被**P**跨越的画图区域。

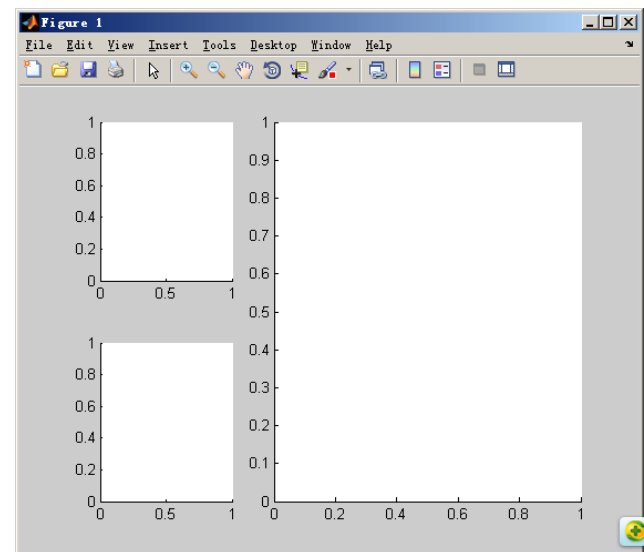
```
subplot(2,3,1)
subplot(2,3,2)
subplot(2,3,3)
subplot(2,3,4)
subplot(2,3,5:6)
```



```
subplot(2,3,1)
subplot(2,3,3)
subplot(2,3,4)
subplot(2,3,6)
subplot(2,3,[2,5])
```



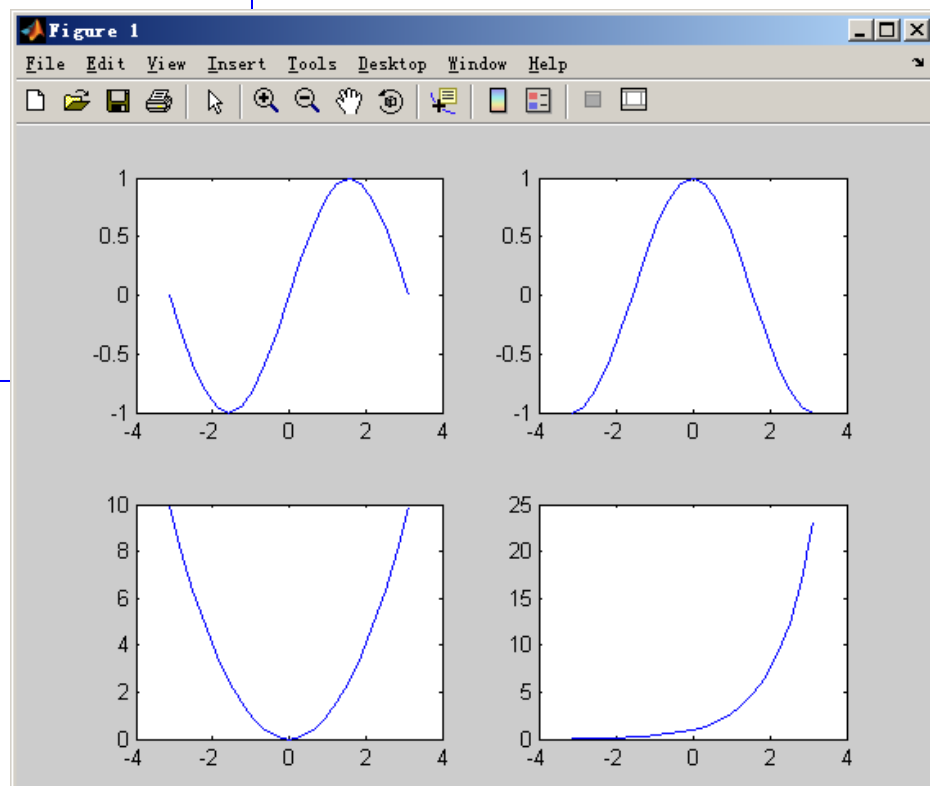
```
subplot(2,3,1)
subplot(2,3,4)
subplot(2,3,[3,5,6])
```



例：使用subplot()

```
x = -pi:pi/10:pi;  
x = x';  
Y = [sin(x),cos(x),x.^2,exp(x)];  
for k = 1:4  
    subplot(2,2,k);  
    plot(x,Y(:,k));  
end
```

先用subplot()
确定在哪个区域中绘图，然后再使用其它的
绘图函数。



被激活的绘图区域在用户输入另一个subplot或者figure命令之前会一直保持被激活状态。

上机练习

某质点的位移由 $x(t)$ 和 $y(t)$ 确定，试绘制其运动轨迹

$$\begin{cases} x(t) = 2\cos(20m\pi t) + \cos(20n\pi t) \\ y(t) = 2\sin(20m\pi t) + \sin(20n\pi t) \end{cases}$$

m	2	2	3	-3
n	3	-3	5	5

要求：

- 1、m、n分别取上表中的四个数值
- 2、绘制1秒内的运动轨迹，每隔0.001秒取一个点；
- 3、四种情况的运动轨迹绘制在一个图形窗口的四个坐标轴中，排成2行2列。

```
m = [2,2,3,-3];  
n = [3,-3,5,5];  
t = 0:0.001:1;  
for k = 1:4  
    x = 2*cos(20*m(k)*pi.*t) +  
cos(20*n(k)*pi.*t);  
    y = 2*sin(20*m(k)*pi.*t) +  
sin(20*n(k)*pi.*t);  
    subplot(2,2,k)  
    plot(x,y)  
end
```

上机练习

如不考虑任何势场，在平衡态下，理想气体的麦克斯韦速率分布函数为：

$$f(v) = 4\pi \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} e^{-\frac{mv^2}{2kT}} v^2$$

其中， T 为气体的热力学温度， m 为气体分子质量， $k = 1.381 \times 10^{-23} \text{ JK}^{-1}$ 为玻耳兹曼常量。

现要求绘制氖气($m_{Ne} = 20.18 \times 1.66 \times 10^{-27} \text{ kg}$)和氩气($m_{Ar} = 39.94 \times 1.66 \times 10^{-27} \text{ kg}$)

在 $T_1 = 273\text{K}$ ， $T_2 = 500\text{K}$ ， $v \in [0, 2000]$ 时的速率分布曲线。要求将一个图形窗口分成 2

行 2 列共 4 个坐标轴：

- (1) 坐标轴 1：绘制氖气在温度 T_1, T_2 下的速率分布曲线， T_1 用红色， T_2 用蓝色；
- (2) 坐标轴 2：绘制氩气在温度 T_1, T_2 下的速率分布曲线， T_1 用红色， T_2 用蓝色；
- (3) 坐标轴 3：绘制氖气、氩气在温度 T_1 下的速率分布曲线，氖气用实线，氩气用虚线；
- (4) 坐标轴 4：绘制氖气、氩气在温度 T_2 下的速率分布曲线，氖气实线，氩气虚线。

```
v=0:0.1:2000;
m1=20.18*1.66e-27;
m2=39.94*1.66e-27;
T1=273
T2=500
k=1.381e-23
a=4*pi*(m1/2*pi*k*T1)^(3/2)*(exp(-
m1*v.^2/(2*k*T1))).*v.^2;
b=4*pi*(m2/2*pi*k*T1)^(3/2)*(exp(-
m2*v.^2/(2*k*T1))).*v.^2;
c=4*pi*(m1/2*pi*k*T2)^(3/2)*(exp(-
m1*v.^2/(2*k*T2))).*v.^2;
d=4*pi*(m2/2*pi*k*T2)^(3/2)*(exp(-
m2*v.^2/(2*k*T2))).*v.^2;
subplot(2,2,1)
axis([0.0,10,-0.01,0.03])
plot(v,a,'r',v,c,'b');
subplot(2,2,2)
plot(v,b,'r',v,d,'b');
```

矩形波 $f(x) = \begin{cases} +1 & \text{于 } (0, \pi) \text{ 以及 } (2m\pi, (2m+1)\pi), \\ -1 & \text{于 } (-\pi, 0) \text{ 以及 } ((2m-1)\pi, 2m\pi) \end{cases}$ 如下图 1 所示, 显然 $f(x)$ 是奇函数, 可展开为傅里叶正弦级数

$$f(x) = \frac{4}{\pi} \left(\sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x + \dots \right).$$

矩形波是不断跃变的, 它竟然可以用正弦函数 (极平常的连续函数) 的级数

表示出来! 为了阐明上面的级数是如何逐渐逼近矩形波, 我们作图 2. S_n 指的是到 $\sin nx$ 项为止的前几项的部分和。可以

看到, 随着 n 的增大, S_n 逐渐逼近矩形波. n 很大的 S_n 画在图 3 中, 它十分接近矩形波, 但在间断点 $x = -\pi, 0, \pi$ 有“冲过了头”的现象。这种“过冲”是傅里叶级数的特性, 叫做吉布斯现象。

上面是梁昆淼编著的《数学物理方法》一书中的部分内容, 针对这部分内容, 要求:

1、绘制图 1、图 2、图 3, 三张图放在同一图形窗口中;

2、绘制图 1 中的矩形波时, x 的取值范围为 $[-5, 8]$,

作图时请将 x 轴的显示范围设置为从 -7 到 9,
将纵轴的显示范围设置为从 -2 到 2;

3、绘制图 2 中的图形时, x 的取值范围为 $[-\pi, \pi]$,

要求绘制 S_1 , S_3 , S_5 和 S_∞ ,

S_∞ 就是 $[-\pi, \pi]$ 之间的矩形波。

4、绘制图 3 中的图形时, 取 $n = 39$ 。

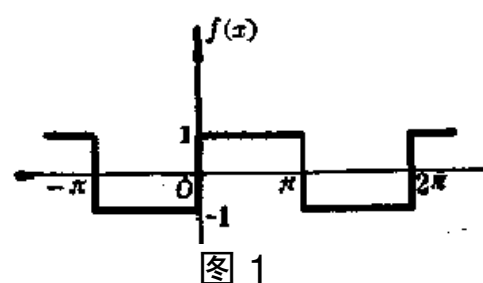


图 1

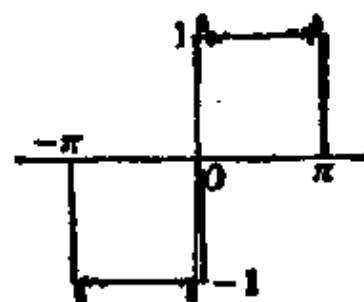


图 3

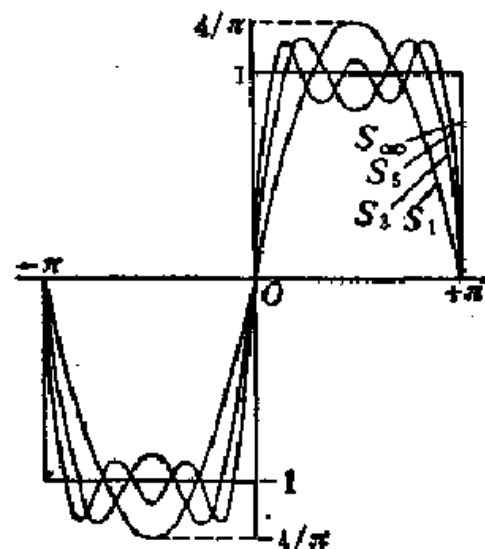


图 2

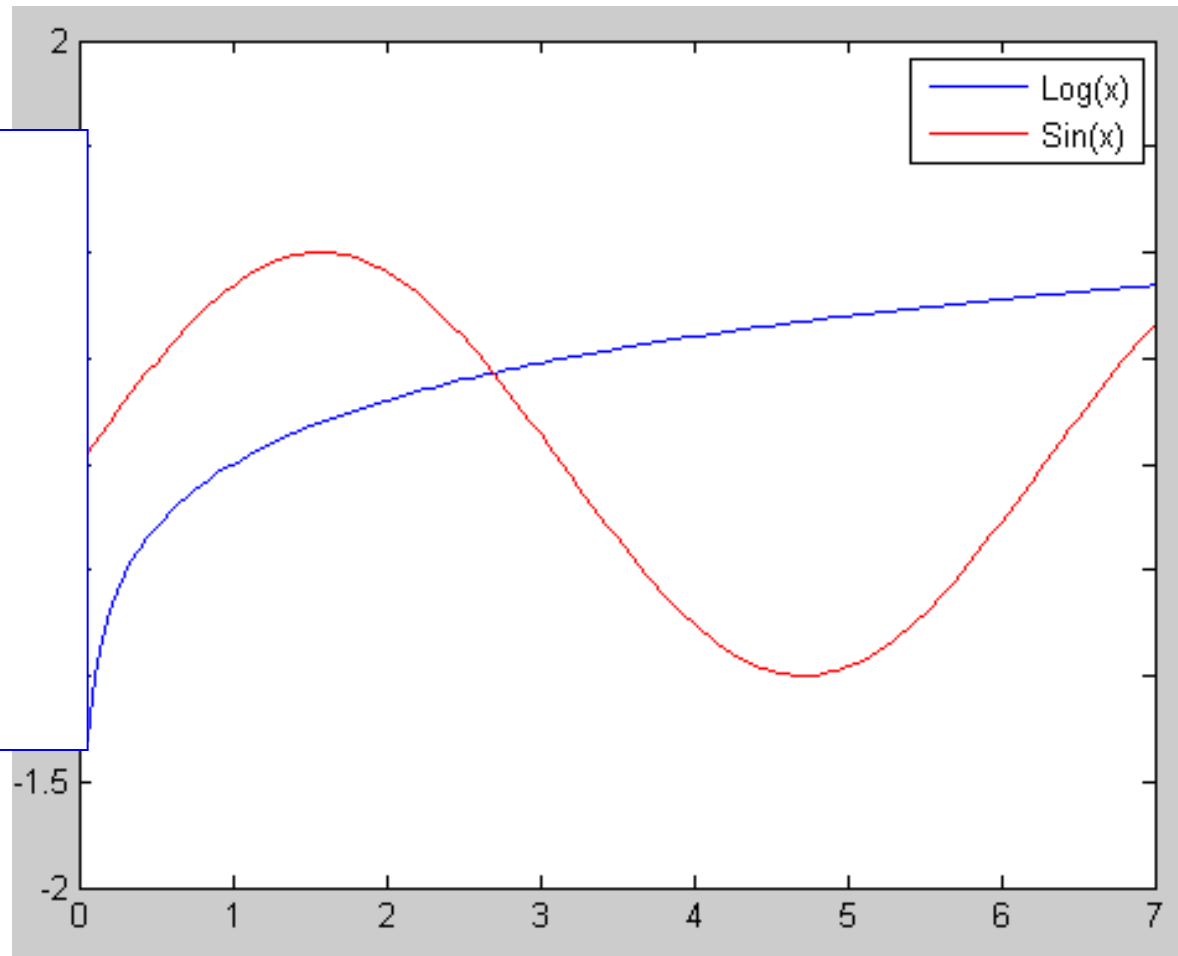
7、图形标注

- legend()
- title()
- xlabel() / ylabel() / zlabel()
- text()
- 使用TeX / LaTeX

legend ()

legend(string1,string2, ...) 添加图例

```
x = 0:0.05: 7  
y1 = log10(x)  
y2 = sin(x)  
plot(x,y1,'b',x,y2,'r')  
axis([0,7,-2,2])  
legend('Log(x)','Sin(x)')
```



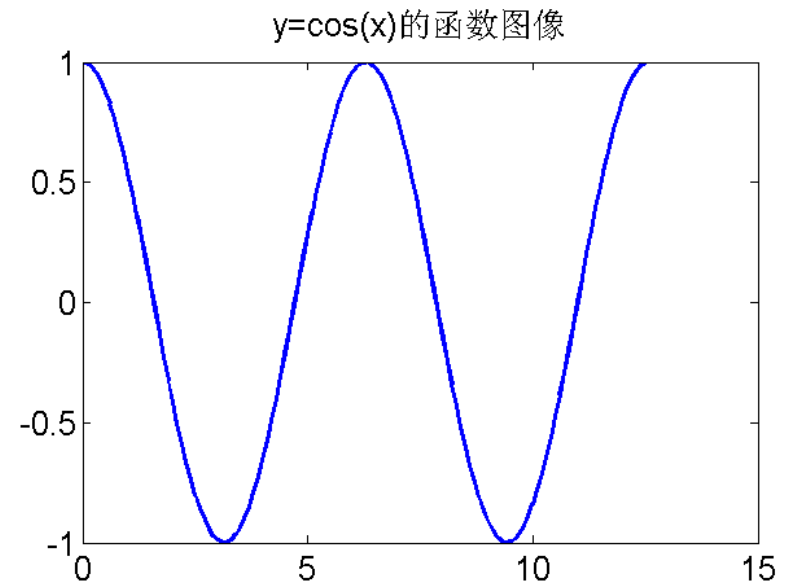
title()

title (' text')

添加标题

例:

```
>> x=[0:0.1:4*pi];  
>> y=cos(x);  
>> plot(x,y)  
>> title('y=cos(x)的函数图像')
```



xlabel() / ylabel() / zlabel()

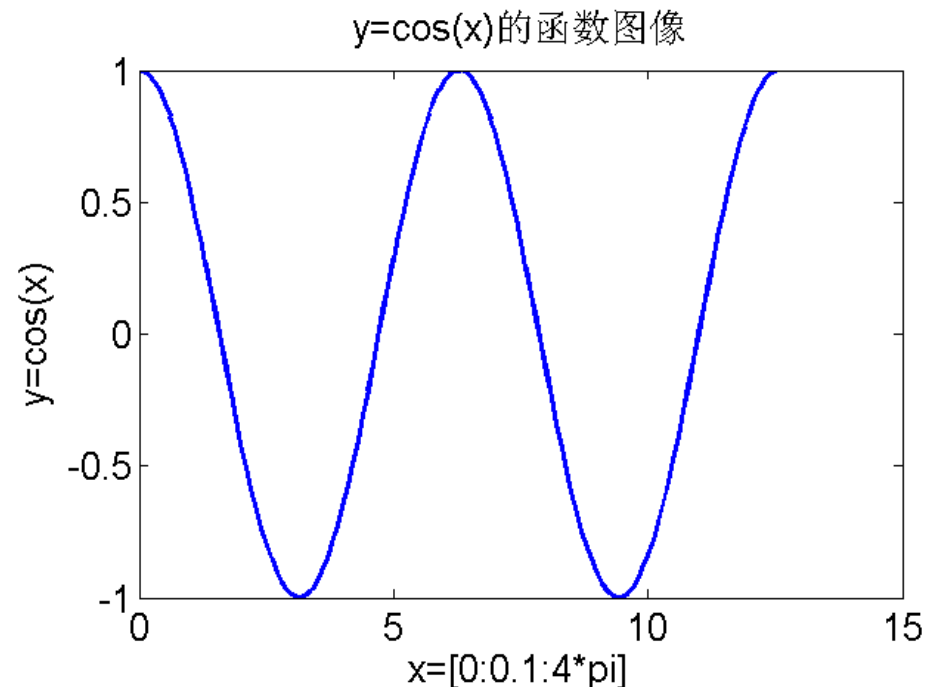
xlabel ('text') 添加 X 坐标轴标注

ylabel ('text') 添加 Y 坐标轴标注

zlabel ('text') 添加 Z 坐标轴标注

例:

```
>> x=[0:0.1:4*pi];  
>> y=cos(x);  
>> plot(x,y)  
>> title('y=cos(x)的函数图像')  
>> xlabel('x=[0:0.1:4*pi]')  
>> ylabel('y=cos(x)')
```

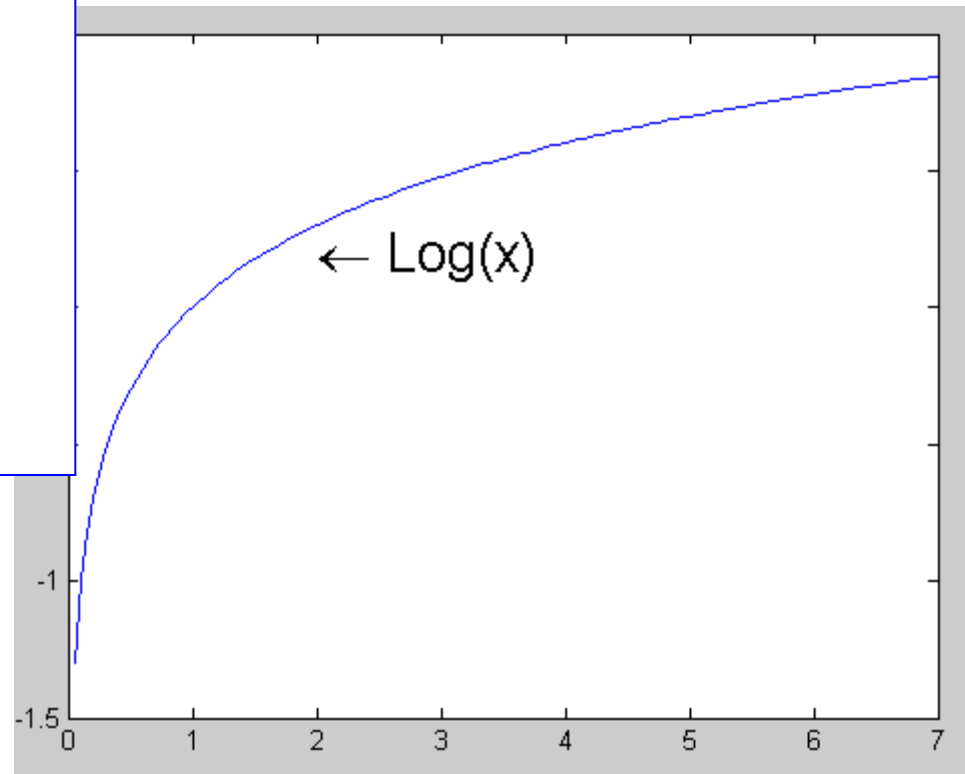


text()

text(x,y,string) 在点 (x,y) 处添加文本

text(x,y,string, 'PropertyName', PropertyValue...)

```
x = 0:0.05: 7  
y = log10(x)  
plot(x,y)  
s = '\leftarrow Log(x)'  
text(2, 0.2, s, 'fontsize',20)
```



使用TeX / LaTeX

在Matlab的text对象的函数中（函数 title、xlabel、ylabel、zlabel、text、legend），说明文字除使用标准的ASCII字符外，还可使用 TeX / LaTeX 格式的控制字符，这样就可以在图形上添加希腊字母、数学符号及公式等内容。Matlab支持两个层次的TeX，通过 text的 [Interpreter](#) 属性来设置：

‘tex’ — 默认设置，支持 TeX 的一个子集

‘latex’ — 支持 LaTeX

‘none’ — 不支持

TeX是D. E. Knuth 开发的免费的功能强大的排版系统，是公认的数学公式排得最好的系统。

LaTeX是由L. Lamport编写的一个目前最流行的TeX宏包（宏集、扩展），特别适合数学类论文、书籍的排版，也可以排版普通的文章和书籍。

TeX Character Sequence

<code>\alpha</code>	α	<code>\upsilon</code>	υ	<code>\sim</code>	≈
<code>\angle</code>	∠	<code>\phi</code>	Φ	<code>\leq</code>	≤
<code>\ast</code>	*	<code>\chi</code>	Χ	<code>\infty</code>	∞
<code>\beta</code>	β	<code>\psi</code>	Ψ	<code>\clubsuit</code>	♣
<code>\gamma</code>	γ	<code>\omega</code>	ω	<code>\diamondsuit</code>	♦
<code>\delta</code>	δ	<code>\Gamma</code>	Γ	<code>\heartsuit</code>	♥
<code>\epsilon</code>	ε	<code>\Delta</code>	Δ	<code>\spadesuit</code>	♠
<code>\zeta</code>	ζ	<code>\Theta</code>	Θ	<code>\leftrightarrow</code>	↔
<code>\eta</code>	η	<code>\Lambda</code>	Λ	<code>\leftarrow</code>	←
<code>\theta</code>	Θ	<code>\Xi</code>	Ξ	<code>\Leftarrow</code>	⇐
<code>\vartheta</code>	ϑ	<code>\Pi</code>	Π	<code>\uparrow</code>	↑
<code>\iota</code>	ι	<code>\Sigma</code>	Σ	<code>\rightarrow</code>	→
<code>\kappa</code>	κ	<code>\Upsilon</code>	Υ	<code>\Rightarrow</code>	⇒
<code>\lambda</code>	λ	<code>\Phi</code>	Φ	<code>\downarrow</code>	↓
<code>\mu</code>	μ	<code>\Psi</code>	Ψ	<code>\circ</code>	∘

Tex字符的字体设置

`\bf`: 设置字体为粗体字。

`\it`: 设置字体为斜体字。

`\sl`: 设置字体为斜体字，很少使用。

`\rm`: 设置字体为正常字体。

`\fontname{字体名}`: 设置字体名。例如: `\fontname{宋体}`

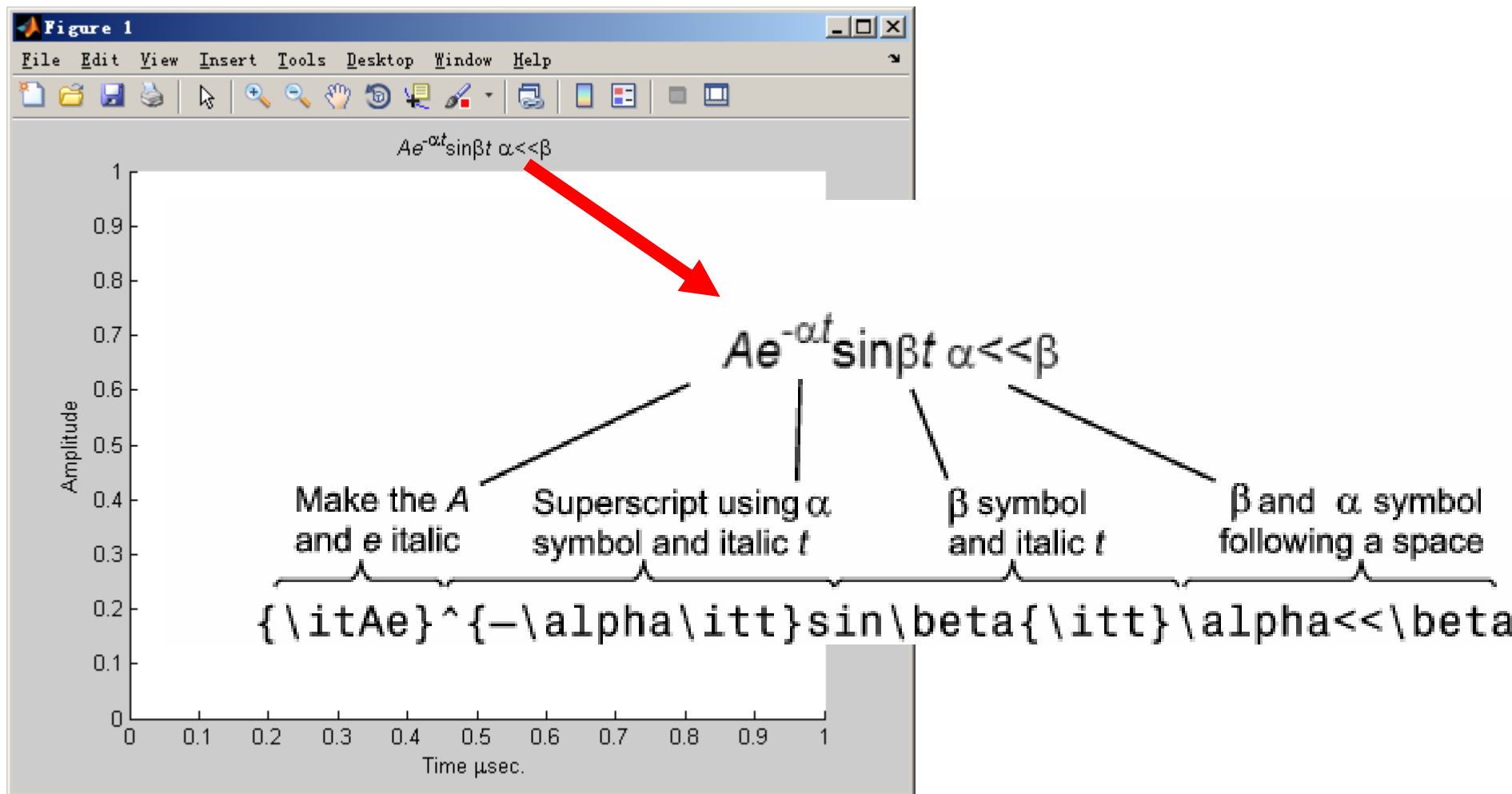
`\fontsize{字体大小}`: 设置字体大小。例如: `\fontsize{16}`

`_ {下标}` 表示下标，若下标只有一个字符，可省略`{ }`

`^ {上标}` 表示上标，若上标只有一个字符，可省略`{ }`

例：使用TeX输入数学表达式

```
title('{\it Ae}^{\alpha\itt}sin\beta\alpha<<\beta')  
xlabel('Time \musec.')  
ylabel('Amplitude')
```



Tex字符的颜色设置

`\color{颜色名}颜色名`

颜色名有12种，分别为red、green、yellow、magenta、blue、black、white、cyan、gray、darkGreen、orange和lightBlue。

例如：\color{magenta}magenta

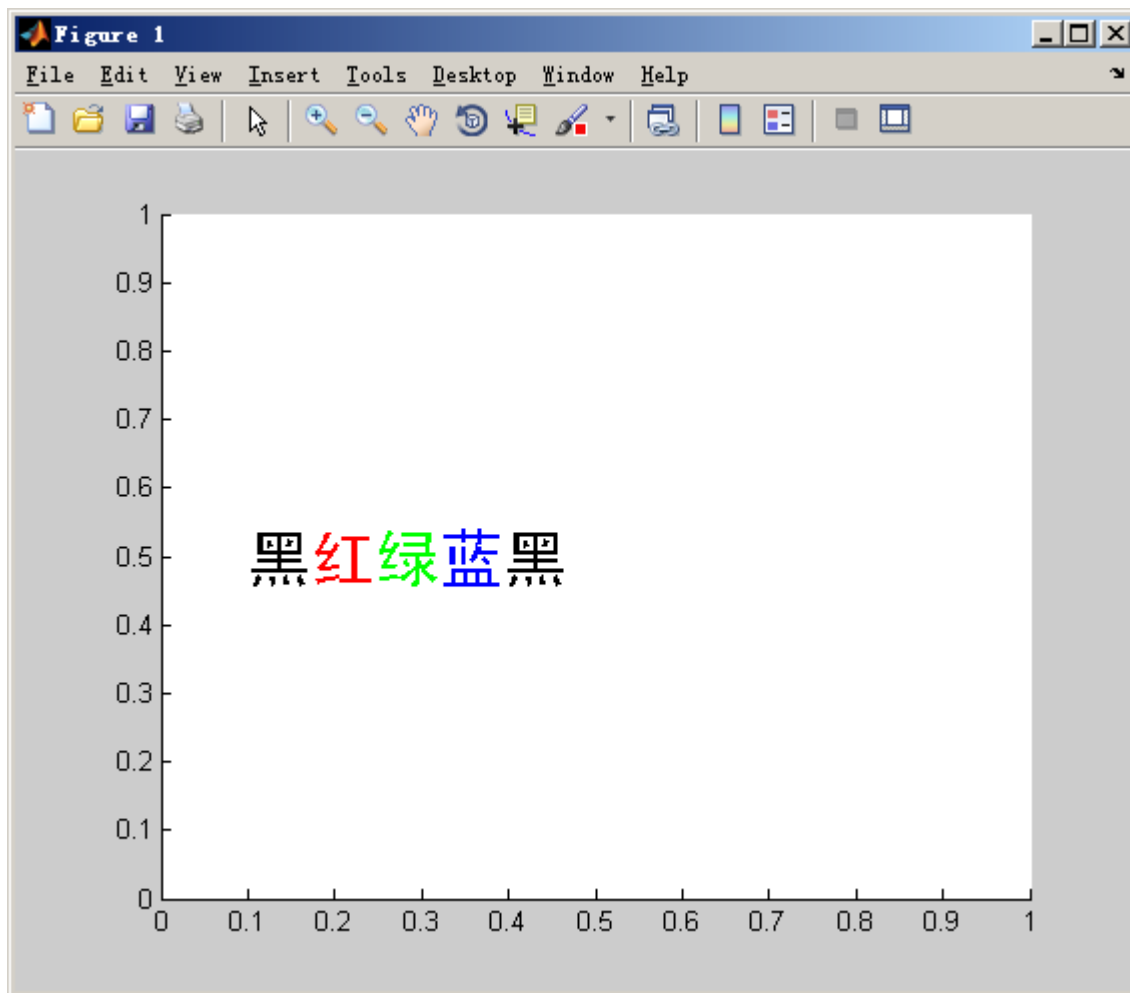
`\color[rgb]{a b c}`

设置字体颜色为RGB矩阵[a b c]所表示的颜色。a、b和c都在[0 1] 范围内。

例如：color[rgb]{0 .5 .5}

例：使用**TeX**指定字符颜色

```
str = '黑{\color{red}红\color{green}绿\color[rgb]{0 0 1}蓝}黑'  
text(0.1,0.5,str,'fontsize',24,'fontname','黑体')
```



使用Latex

在matlab中，使用Latex命令可用下面的方法：

`\(LaTeX命令 \)`

`$ LaTeX命令$`

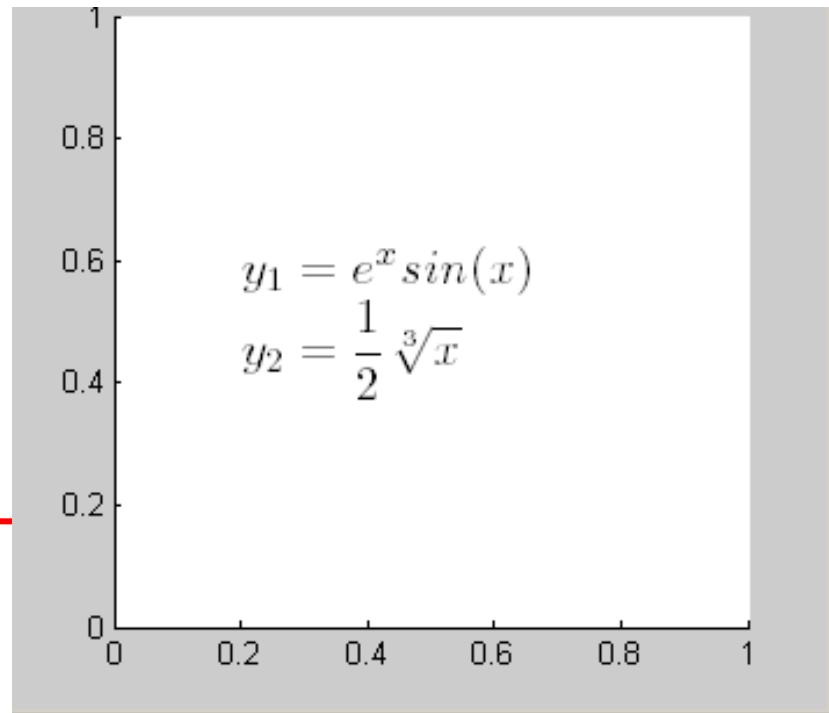
`$$ LaTeX命令 $$`

```
s1 = '$y_1=e^xsin(x)$'
```

```
s2 = '$$y_2=\frac{1}{2}\sqrt[3]{x}$$'
```

```
str = char(s1,s2)
```

```
text(0.2, 0.5, str, 'Interpreter', 'latex', 'FontSize',16)
```



legend()函数必须先获得句柄，才可以设置Interpreter属性

```
h = legend(str1, str2, ... )
```

```
set(h,'interprete','latex')
```

常用 LaTeX 命令

上标用 \wedge {上标}, 下标用 $_$ {下标}, 希腊字母与TeX一样, 用 α 等表示

分式 $\frac{\text{分子}}{\text{分母}}$

根式 \sqrt{x} 、 $\sqrt[n]{x}$

求和 $\sum_{i=1}^n x_i$

积分 \int_0^1

偏微分 $\frac{\partial y}{\partial x}$

极限 $\lim_{n \rightarrow \infty}$ %n趋于无穷符号在lim正下方

$\lim_{n \rightarrow \infty}$ %n趋于无穷符号在lim右下角

上划线 \overline{x}

下划线 \underline{x} %下划线在x的正下方

卧式花括号命令 $\overbrace{x+y+z+w}$

仰式花括号命令 $a+\underbrace{b+c+d}$

更多的 LaTeX 命令请参考相关的 LaTeX 资料

三、3d绘图

- 1、三维曲线: `plot3()`
- 2、二维数据网格: `meshgrid()`
- 3、三维网格图: `mesh() / meshc() / meshz()`
- 4、三维表面图: `surf() / surfc() /`
- 5、利用`surf()`绘制一些常见的三维表面图
- 6、三维绘图函数汇总

1、三维曲线: `plot3()`

```
plot3(X1,Y1,Z1,...)
```

```
plot3(X1,Y1,Z1,LineSpec,...)
```

```
plot3(...,'PropertyName',PropertyValue,...)
```

```
h = plot3(...)
```

`plot3()`的用法与`plot()`类似，只是多了一个 **Z** 数组。

```
t=[0:0.2:10*pi];
```

```
x=2*t;
```

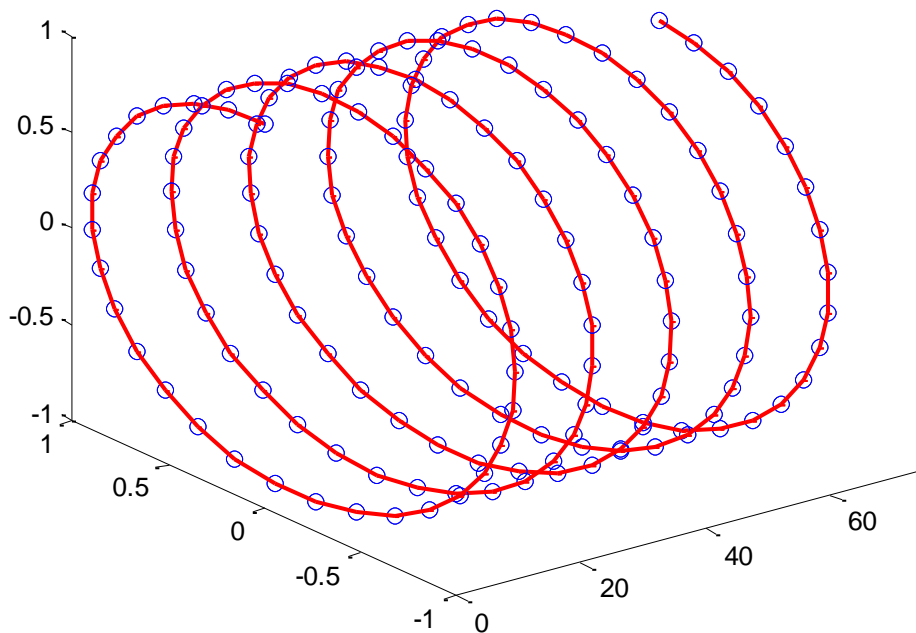
```
y=sin(t);
```

```
z=cos(t);
```

```
plot3(x,y,z,'bo');
```

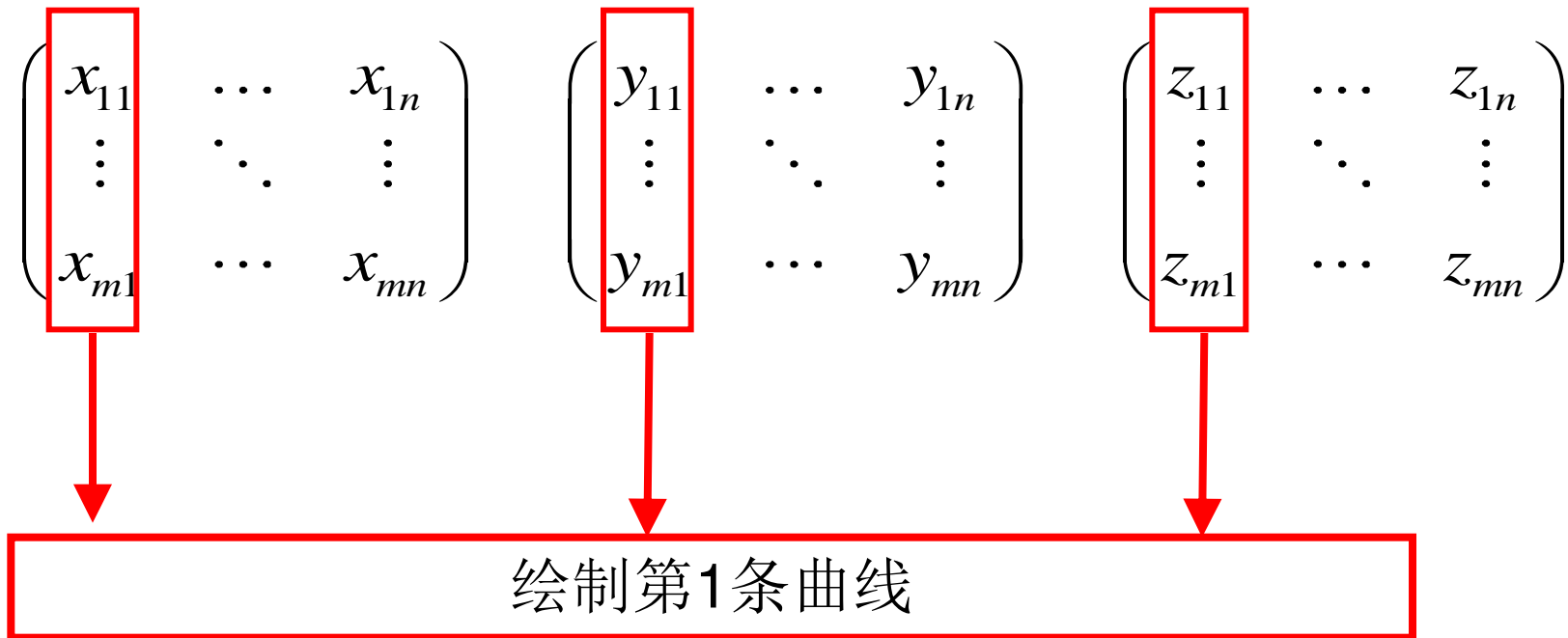
```
hold on
```

```
plot3(x,y,z,'r-','LineWidth',2)
```



用plot3()同时绘制多条3d曲线

当X, Y, Z为同维的二维数组, plot3()将 X、Y、Z 相应的列相组合, 绘制多条3d曲线。



例：使用plot3()绘制长方体的线框图

```
L = 100*randn(1);
```

```
W = L*randn(1);
```

```
H = W*randn(1);
```

```
A = randn(1,3);
```

```
B = A + [L,0,0];
```

```
C = B + [0,W,0];
```

```
D = A + [0,W,0];
```

长度平行于X轴，
宽度平行于Y轴，
高度平行于Z轴，
A为3d空间任意点
均随机生成。

```
A_A = [A;B;C;D;A];
```

```
a_a = A_A + repmat([0,0,H],5,1);
```

```
X = [A_A(:,1),a_a(:,1)];
```

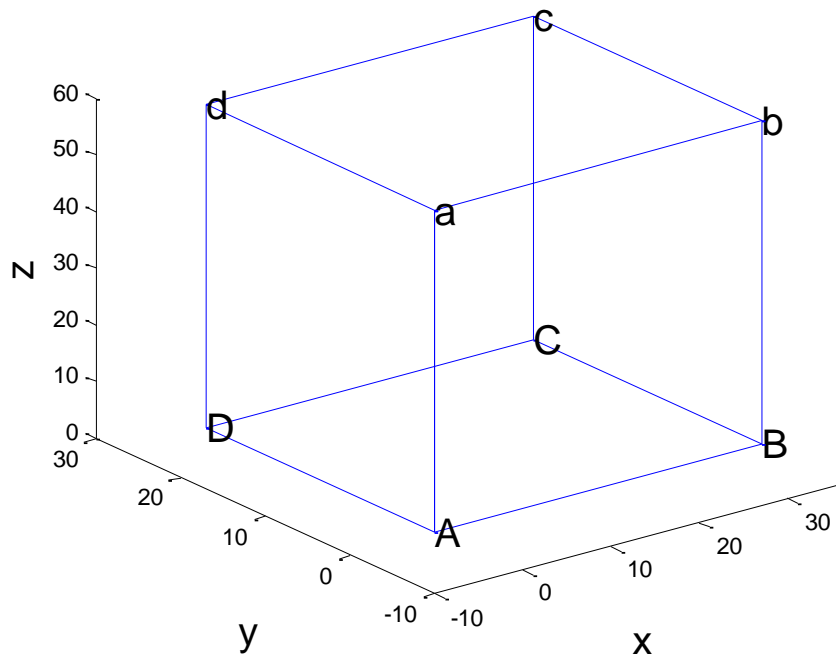
```
Y = [A_A(:,2),a_a(:,2)];
```

```
Z = [A_A(:,3),a_a(:,3)];
```

```
plot3(X,Y,Z,'b')
```

```
hold on
```

```
plot3(X',Y',Z', 'b')
```

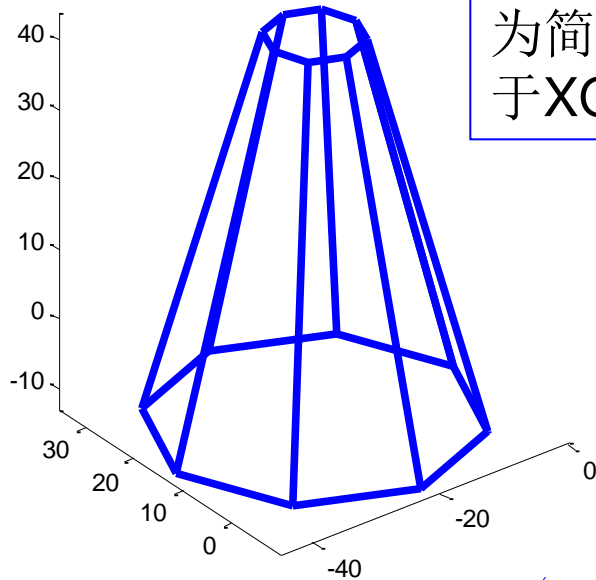


绘图思路（供参考）：
绘制空间曲线

A-B-C-D-A
a-b-c-d-a

A-a
B-b
C-c
D-d

练习：用plot3()绘制正棱柱(台、椎)的线框图



为简单起见，设：中心轴与 Z 轴平行，顶面与底面平行于 XOY 平面，正多边形有个顶点的投影在 X 轴上

设： 底面正多边形的外接圆中心为 $C=[Cx,Cy,Cz]$ ，半径为 R ，边数为 N ，顶面外接圆半径为 r ，棱柱高度为 H 。

$$t = 0:2\pi/N:2\pi$$

则底面各顶点的坐标为：

$$x = Cx + R \cos t$$

$$y = Cy + R \sin t$$

$$z = Cz \cdot \text{ones}(\text{size}(t))$$

则顶面各顶点的坐标为：

$$x = Cx + r \cos t$$

$$y = Cy + r \sin t$$

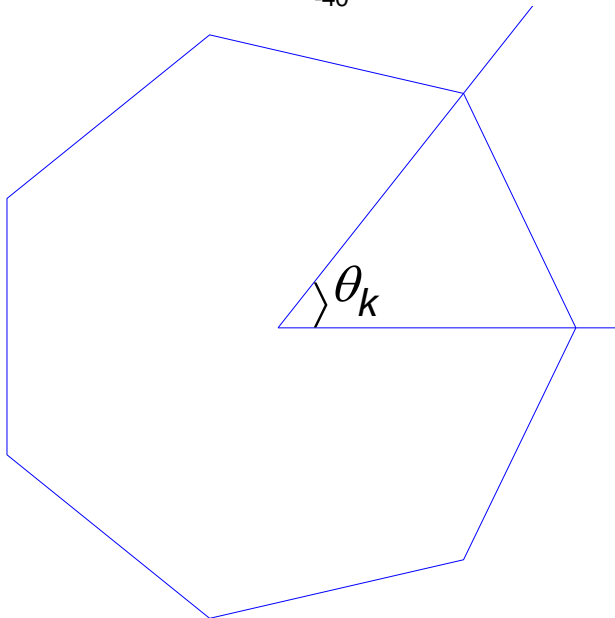
$$z = (Cz + H) \cdot \text{ones}(\text{size}(t))$$

绘图思路（供参考）：
绘制空间曲线

底面多边形

顶面多边形

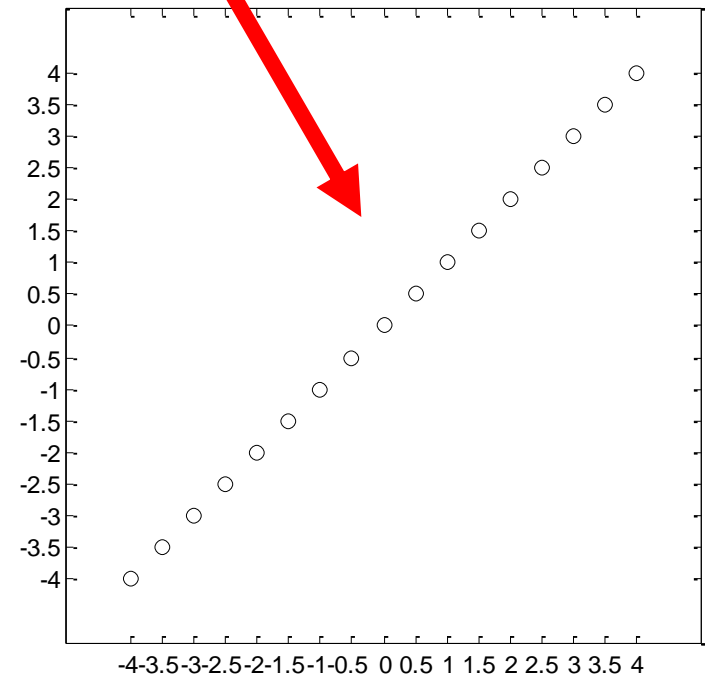
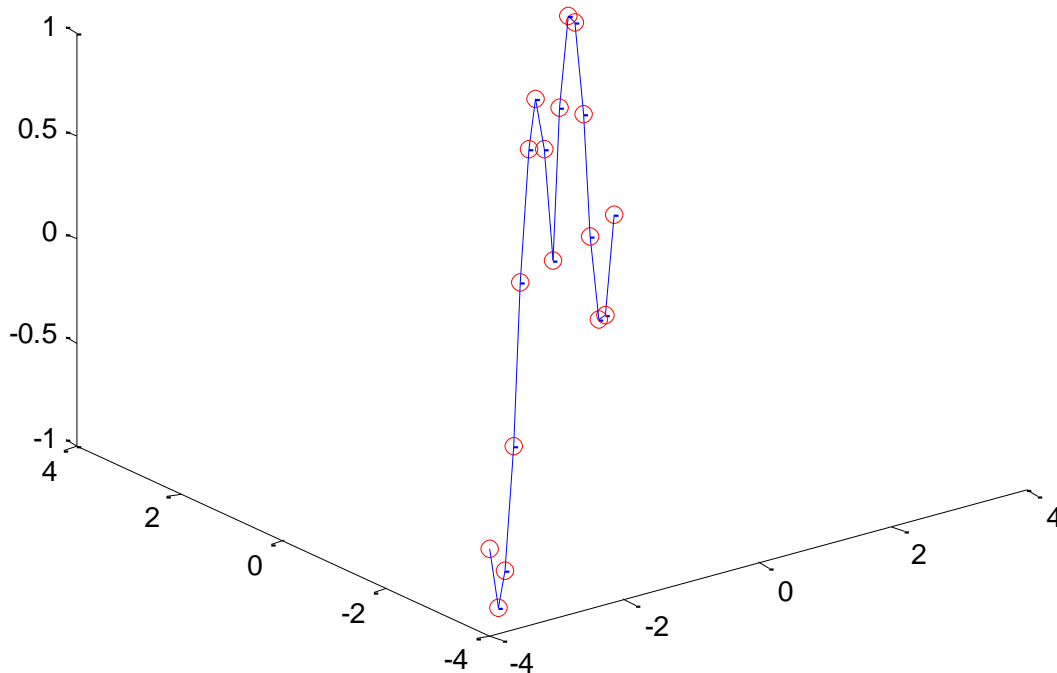
各条棱

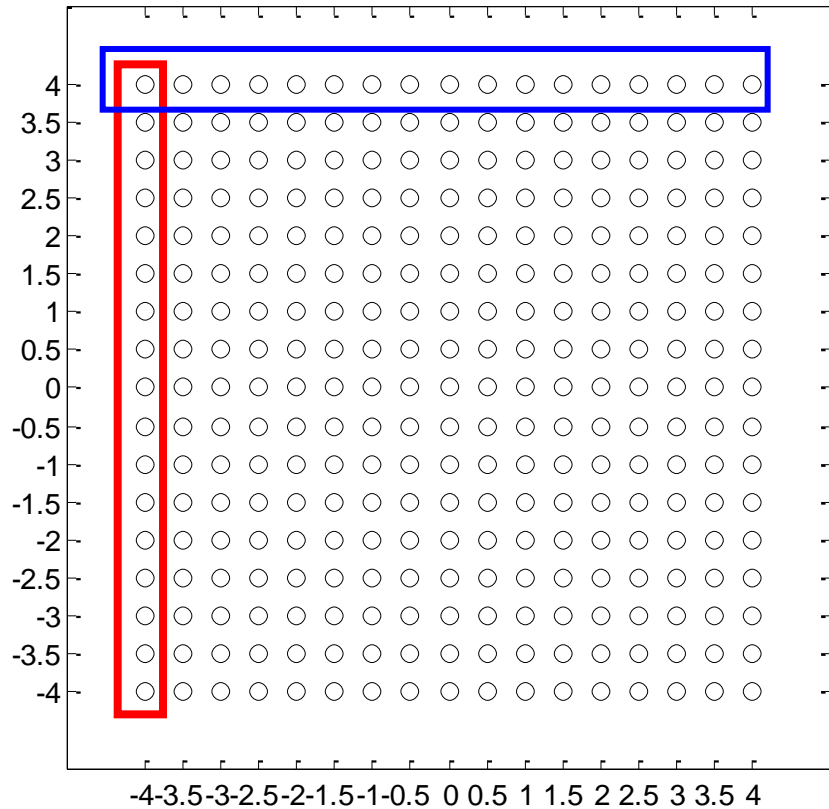


如何用plot3()绘制 $z = \sin(\sqrt{x^2 + y^2})$?

```
x = ( -4:0.5:4 )'  
y = ( -4:0.5:4 )'  
z = sin(sqrt(x.^2+y.^2))  
h = plot3(x,y,z,'-bo')  
set(h,'MarkerEdgeColor','r')
```

x	y	z
-4.0000	-4.0000	-0.5862
-3.5000	-3.5000	-0.9720
-3.0000	-3.0000	-0.8917
-2.5000	-2.5000	-0.3838
.....		
3.0000	3.0000	-0.8917
3.5000	3.5000	-0.9720





```
x = -4:0.5:4
```

```
y = ( 4:-0.5:-4 )'
```

```
X = repmat(x,length(y),1)
```

```
Y = repmat(y,1,length(x))
```

X =

-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000
...
-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000

Y =

4.0000	4.0000	...	4.0000	4.0000
3.5000	3.5000	...	3.5000	3.5000
...
-3.5000	-3.5000	...	-3.5000	-3.5000
-4.0000	-4.0000	...	-4.0000	-4.0000

[X, Y] = meshgrid(x, y)

2、二维数据网格： `meshgrid()`

`[X,Y] = meshgrid(x,y)`

由向量 x 和 y 生成二维数组 X 和 Y ，用来计算二元函数 $f(x,y)$ 的值 $Z = f(X,Y)$ 。二维数组 X ， Y ， Z 可用来绘制三维曲线、三维网格图、三维曲面图等。

输出数组 X 中的行向量相当于向量 x ，输出数组 Y 中的列向量相当于向量 y 。

`[X,Y] = meshgrid(x)`

等价于 `[X,Y] = meshgrid(x,x)`。

```
x = -4:0.5:4  
y = -4:0.5:4  
[X,Y] = meshgrid(x,y)  
Z = sin(sqrt(X.^2+Y.^2))
```

```
subplot(2,2,1)  
plot3(X,Y,Z,'bo')
```

```
subplot(2,2,2)  
plot3(X,Y,Z,'b')
```

```
subplot(2,2,3)  
plot3(X',Y',Z', 'b')
```

```
subplot(2,2,4)  
plot3(X,Y,Z,'b',X',Y',Z', 'b')
```

X =				
-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000
...
-4.0000	-3.5000	...	3.5000	4.0000
-4.0000	-3.5000	...	3.5000	4.0000

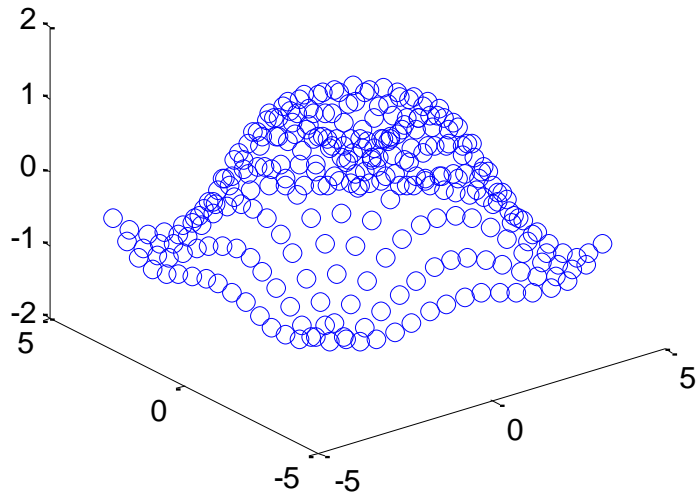
Y =				
-4.0000	-4.0000	...	-4.0000	-4.0000
-3.5000	-3.5000	...	-3.5000	-3.5000
...
3.5000	3.5000	...	3.5000	3.5000
4.0000	4.0000	...	4.0000	4.0000

Z =				
-0.5862	-0.8238	...	-0.8238	-0.5862
-0.8238	-0.9720	...	-0.9720	-0.8238
...
-0.8238	-0.9720	...	-0.9720	-0.8238
-0.5862	-0.8238	...	-0.8238	-0.5862

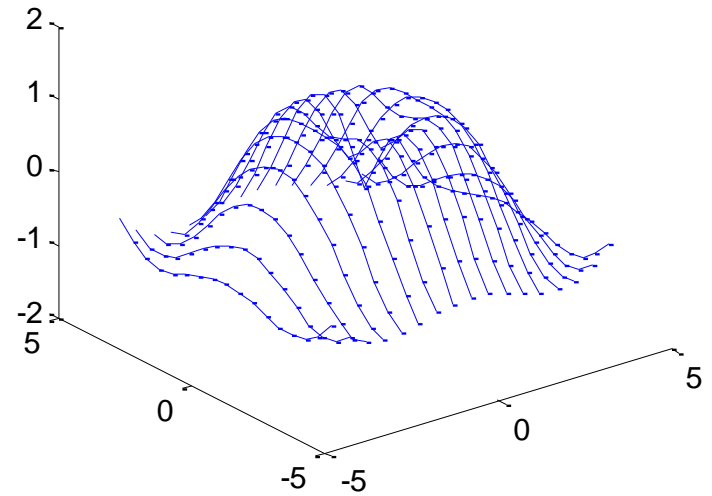
绘制第1条
3d曲线

绘制第2条
3d曲线

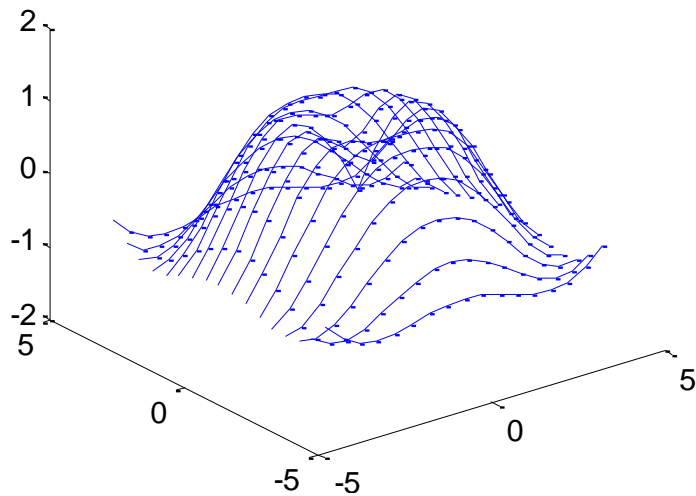
plot3(X,Y,Z,'bo')



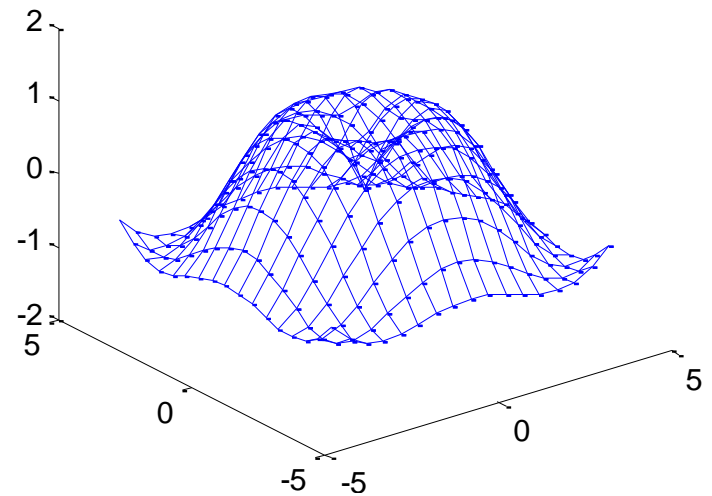
plot3(X,Y,Z,'b')



plot3(X',Y',Z','b')



plot3(X,Y,Z,'b',X',Y',Z','b')



3、三维网格图: `mesh()` / `meshc()` / `meshz()`

`mesh(X,Y,Z)`: 绘制由数组 `X,Y,Z` 所确定的曲面的网格图

`X, Y, Z` 都为二维数组时, 要求它们的维数相同。`X, Y` 也可以是向量, 但 `Z` 必须为二维数组, $[m,n] = \text{size}(Z)$, 此时必须满足: $\text{length}(X) = n$ 且 $\text{length}(Y) = m$ 。

`mesh(Z)`: 相当于 `X = 1:n`, `Y = 1:m`, 其中 $[m,n] = \text{size}(Z)$

`mesh(...,C)`: 二维数组 `C` 确定网格颜色, 省略 `C` 时相当于 `C=Z`

`mesh(...,'PropertyName',PropertyValue,...)`: 设置属性值

`mesh(axes_handles,...)`: 在指定的坐标轴绘图

`h = mesh(...)`: 返回句柄

$$z = \sin\left(\sqrt{x^2 + y^2}\right)$$

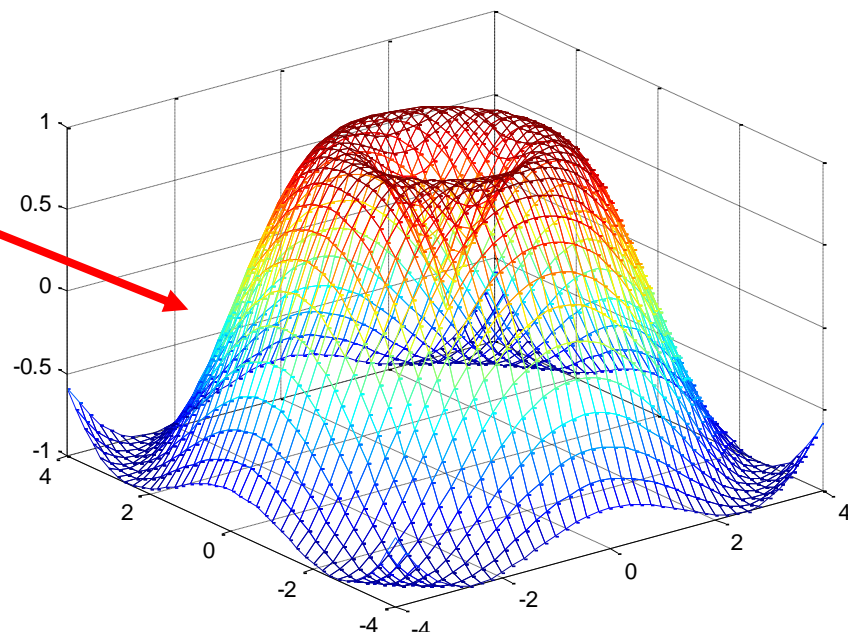
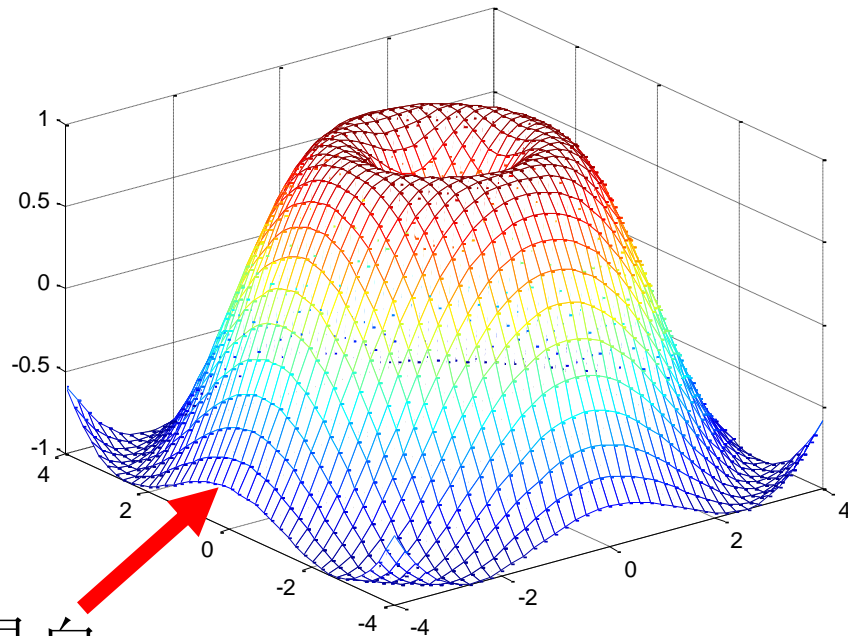
```
x = -4:0.2:4
[X,Y] = meshgrid(x)
Z = sin(sqrt(X.^2+Y.^2))
h = mesh(X,Y,Z)
c1 = get(h,'FaceColor')
```

默认情况下每个四边形区域填充的是白色，因此c1的值 [1,1,1]

hidden off

```
c2 = get(h,'FaceColor')
```

hidden off命令是使每个四边形区域不填充任何颜色，是空的，可以看到后面的图线。因此c2的值为 none



meshc(X,Y,Z)

调用方式与 mesh 相同，在 mesh 基础上增加等高线

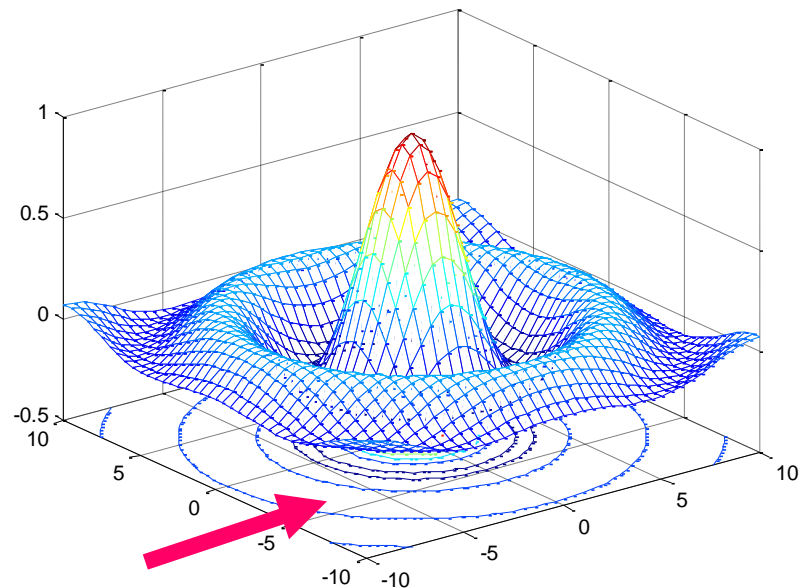
$$Z = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$$

墨西哥帽子

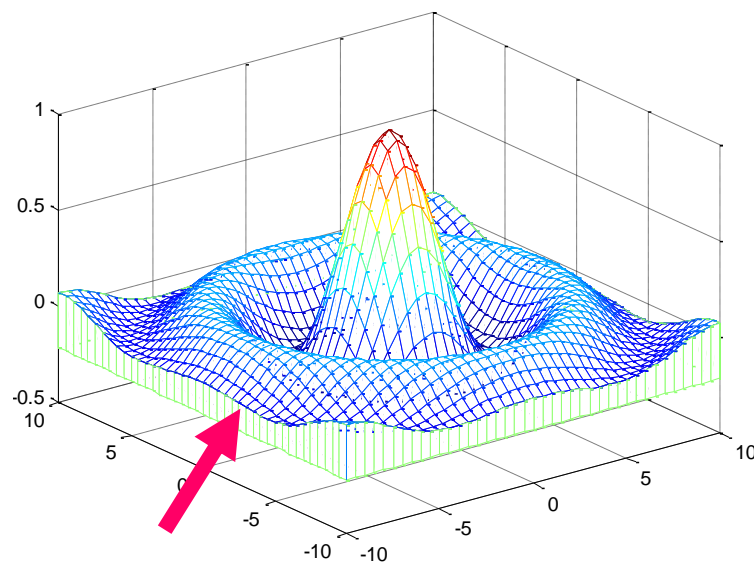
```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x);  
r = sqrt(X.^2+Y.^2)+eps  
Z = sin(r)./r  
meshc(X,Y,Z)
```

meshz(X,Y,Z)

调用方式与 mesh 相同，在 mesh 基础上屏蔽边界面



等高线



屏蔽边界面

4、三维表面图： `surf()` / `surfc()`

绘制由矩阵 `X,Y,Z` 所确定的表面图，参数含义同 `mesh`

`surf(Z)` : 相当于 `X = 1:n` , `Y = 1:m` , 其中 `[m,n] = size(Z)`

`surf(Z,C)` : 二维数组 `C` 确定网格颜色，省略 `C` 时相当于 `C=Z`

`surf(X,Y,Z)` : 绘制由数组 `X,Y,Z` 所确定的曲面图

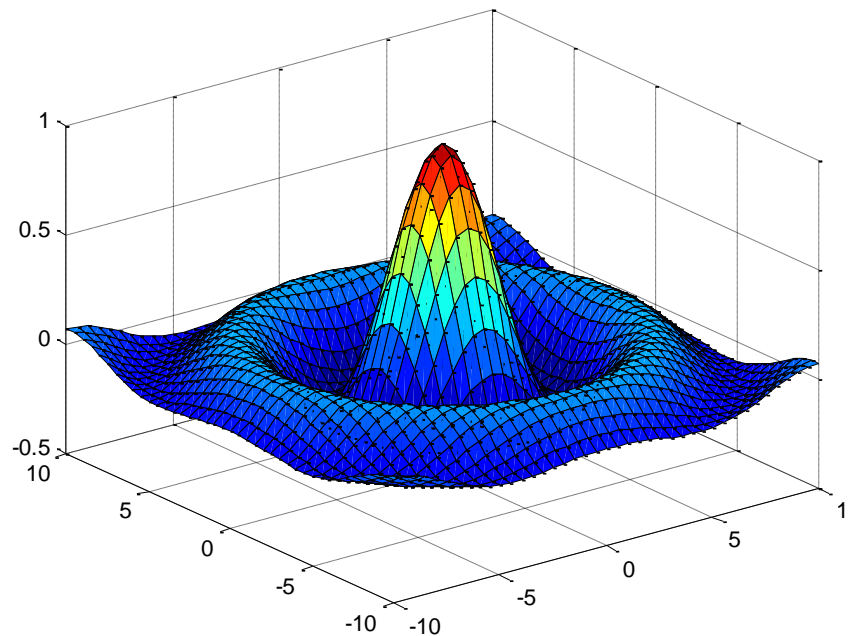
`surf(X,Y,Z,C)` :

`surf(...,'PropertyName',PropertyValue)` : 设置属性值

`surf(axes_handles,...)` : 在指定的坐标轴绘图

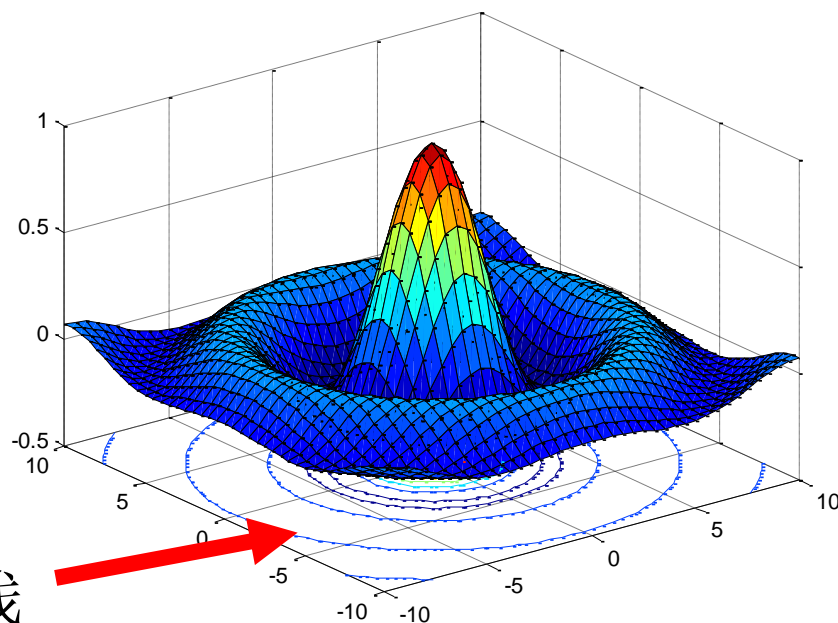
`h = surf(...)` : 返回句柄


```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x) ;  
r = sqrt(X.^2+Y.^2)+eps  
Z = sin(r) ./r  
surf(X,Y,Z)
```



surf(X,Y,Z)

调用方式与 `surf` 相同，在 `surf` 基础上增加等高线



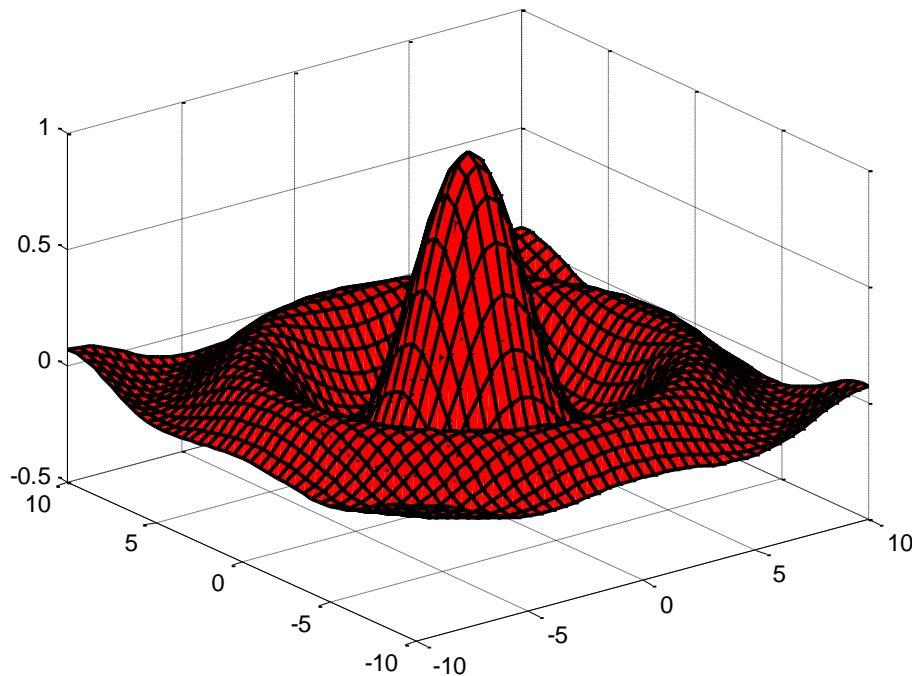
等高线

mesh() / surf()的一些常用属性

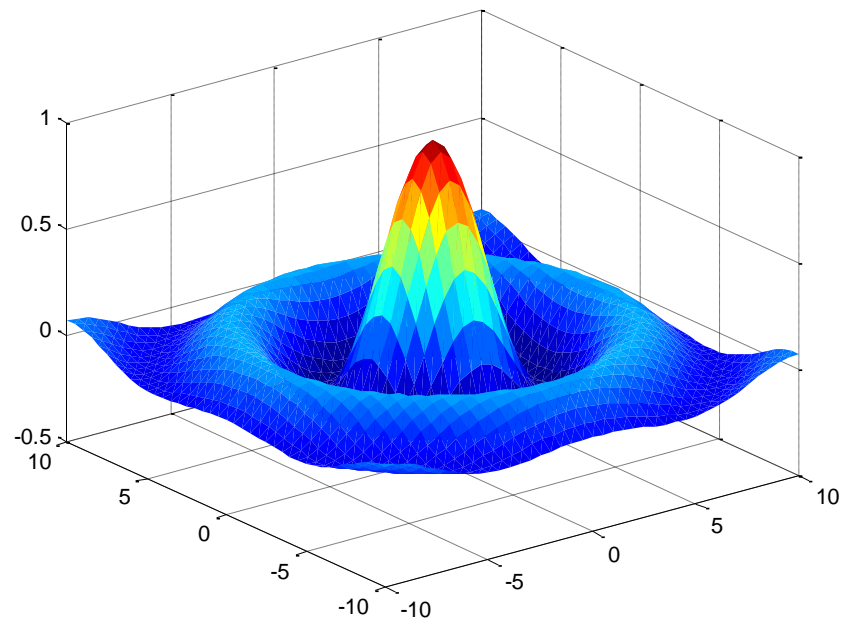
属性名称	意义	取值
EdgeColor	网格线颜色	{ColorSpec} none flat interp
FaceColor	四边形网格的填充颜色	ColorSpec none {flat} interp texturemap
LineStyle	网格线线型 :	{-} -- : -. none
LineWidth	网格线线宽	
Marker	标记点形状	none + o * . x s d p h
MarkerEdgeColor	标记点边界颜色	none {auto} flat ColorSpec
MarkerFaceColor	闭合的标记点填充颜色	{none} auto flat ColorSpec
MarkerSize	标记点大小	size in points
MeshStyle	网格类型	{both} row column

例： mesh() / surf()属性设置

```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x);  
r = sqrt(X.^2+Y.^2)+eps  
Z = sin(r)./r  
h = mesh(X,Y,Z,'EdgeColor','black')  
set(h,'FaceColor','r')  
set(h,'LineWidth',2)
```



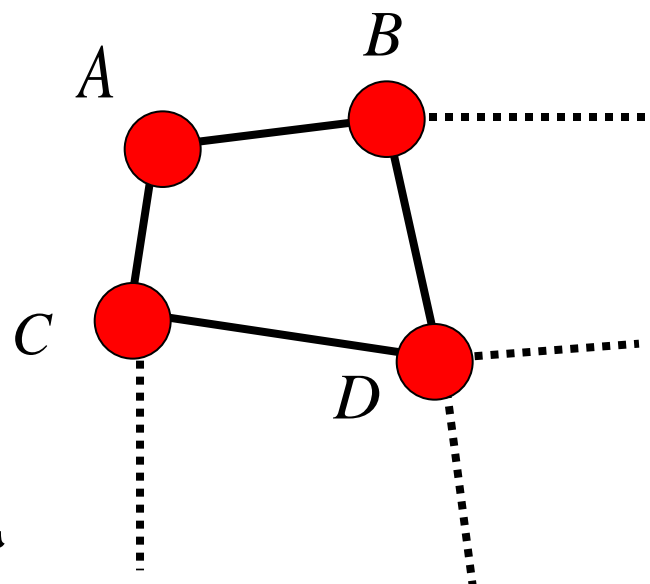
```
x = -10:0.5:10 ;  
[X,Y] = meshgrid(x);  
r = sqrt(X.^2+Y.^2)+eps  
Z = sin(r)./r  
surf(X,Y,Z,'EdgeColor','none')
```



5、利用surf()绘制一些常见的三维表面图

mesh()/surf()绘制三维曲面的方法:

- (1)先根据X,Y,Z数组确定网格点
- (2)用网格线连接在同一行中的网格点
- (3)用网格线连接在同一列中的网格点
- (4)用颜色数组C确定网格线（面）的颜色



$$X = \begin{bmatrix} A_x & B_x & \cdots \\ C_x & D_x & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$Y = \begin{bmatrix} A_y & B_y & \cdots \\ C_y & D_y & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$Z = \begin{bmatrix} A_z & B_z & \cdots \\ C_z & D_z & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

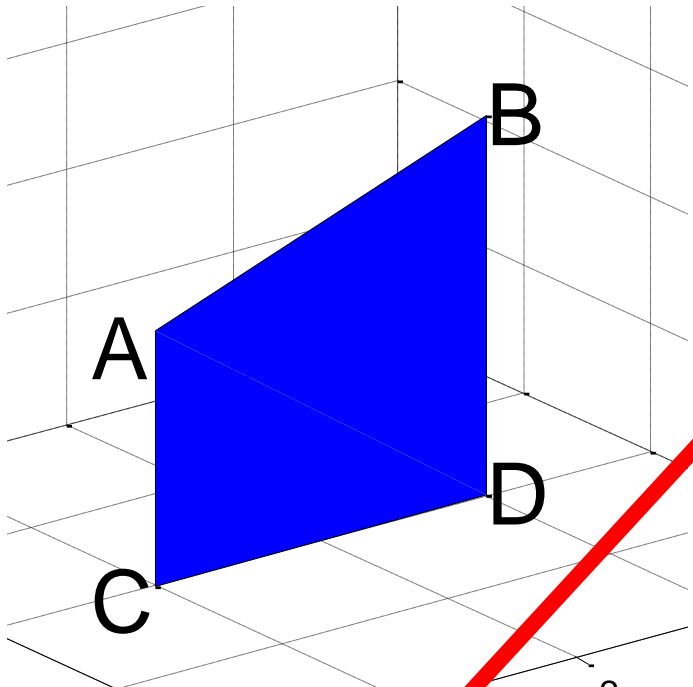
(1) 用surf()绘制四边形平面

绘图思路：

把四个顶点分成“2行2列”，将相应的坐标放进X,Y,Z数组即可绘图。

同理，对 $2n$ 边形，可将 $2n$ 个顶点分成“2行 n 列”或“ n 行2列”进行处理。

对凹多边形，这样处理可能会出错。

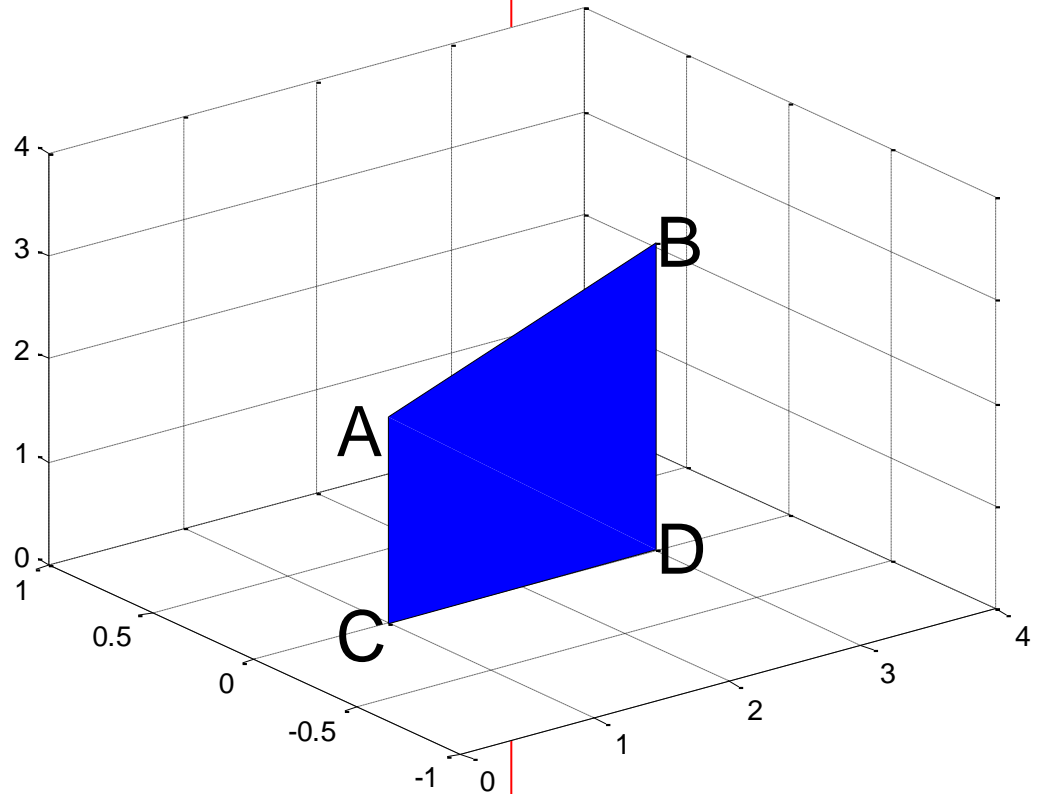

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

如上图，按这样的方式创建出X, Y, Z数组，即可绘制出四边形平面

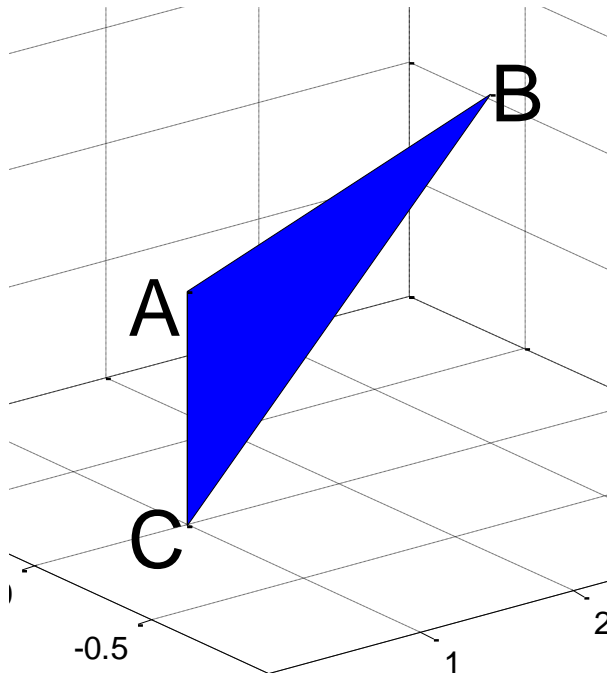
```
clc
clear all
close all
A = [1;0;2]
B = [3;0;3]
C = [1;0;0]
D = [3;0;0]

P = [A,B;C,D]

X = P([1,4], :)
Y = P([2,5], :)
Z = P([3,6], :)
h = surf(X,Y,Z)
set(h, 'FaceColor', 'b')
axis([0,4,-1,1,0,4])
```



(2) 用surf()绘制三角形平面



绘图思路：

想象一下，有两个A点，只不过它们完全重合，这样就有四个顶点了，可以分成“2行2列”，将相应的坐标放进X，Y，Z数组即可绘图。

选取合适的顶点，这个想法对任意多边形都可以。

$$\begin{bmatrix} A & A \\ B & C \end{bmatrix}$$

按这样的方式创建出X，Y，Z数组，即可绘制出三角形平面

$$\begin{bmatrix} A & A & A \\ A & B & C \end{bmatrix}$$

这样更简单。第一行中的A点也可以是B点、C点或多边形内部的任意一个点。

例：绘制一个长方体表面图（共六个面）

```
L = rand(1);  
W = rand(1);  
H = rand(1);
```

```
A = rand(3,1);  
B = A + [L;0;0];  
C = B + [0;W;0];  
D = A + [0;W;0];
```

```
r1 = repmat(A,1,5);  
r2 = [A,B,C,D,A];  
r3 = r2 + repmat([0;0;H],1,5);  
r4 = repmat(r3(:,1),1,5);  
P=[r1;r2;r3;r4];
```

```
X = P(1:3:end,:);  
Y = P(2:3:end,:);  
Z = P(3:3:end,:);
```

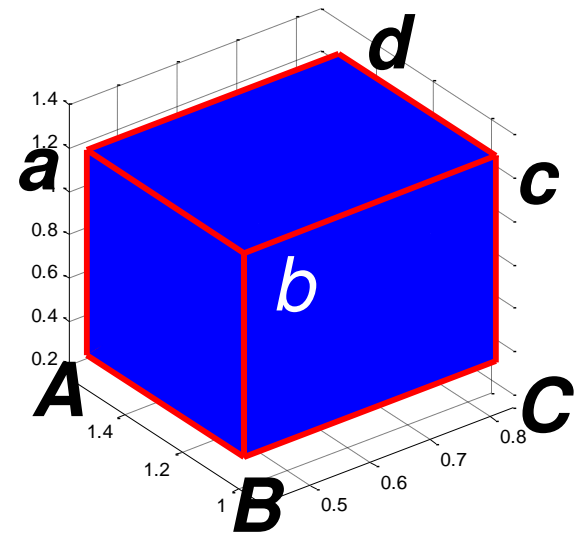
```
surf(X,Y,Z,'FaceColor','b','EdgeColor','none')  
axis vis3d  
hold on
```

```
x = X(2:3,:); % x,y,z用来绘制线框  
y = Y(2:3,:);  
z = Z(2:3,:);  
plot3(x,y,z,'r','LineWidth',3)  
plot3(x',y',z','r','LineWidth',3)
```

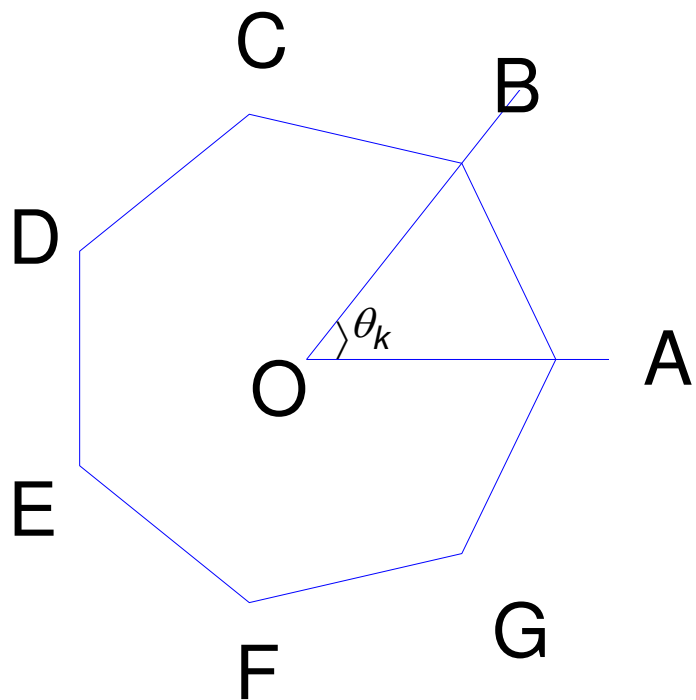
A	A	A	A	A
A	B	C	D	A
a	b	c	d	e
a	a	a	a	a



按这样的方式创建X, Y, Z数组,可以同时绘制出六个面。当然也可以用其它的方式进行。



(3) 用surf()绘制平行于XOY平面的正多边形平面



```
N = 5    %绘制正N边形
```

```
R = 2    %外接圆半径
```

```
z = 0
```

```
t = 0:2*pi/N:2*pi
```

```
%绘图数组
```

```
X = [R*cos(t); zeros(size(t))]
```

```
Y = [R*sin(t); zeros(size(t))]
```

```
Z = z*ones(size(X))
```

```
.....
```

```
[ A  B  C  ...  F  G  A ]  
[ O  O  O  ...  O  O  O ]
```

按这样的方式创建出X, Y, Z数组, 即可绘制出正多边形平面

N = 5 %绘制正N边形

R = 2 %外接圆半径

z = 0

t = 0:2*pi/N:2*pi

%侧面

X = [R*cos(t); r*cos(t)]

Y = [R*sin(t); r*sin(t)]

Z = [z1*ones(size(t)); z2*ones(size(t))]

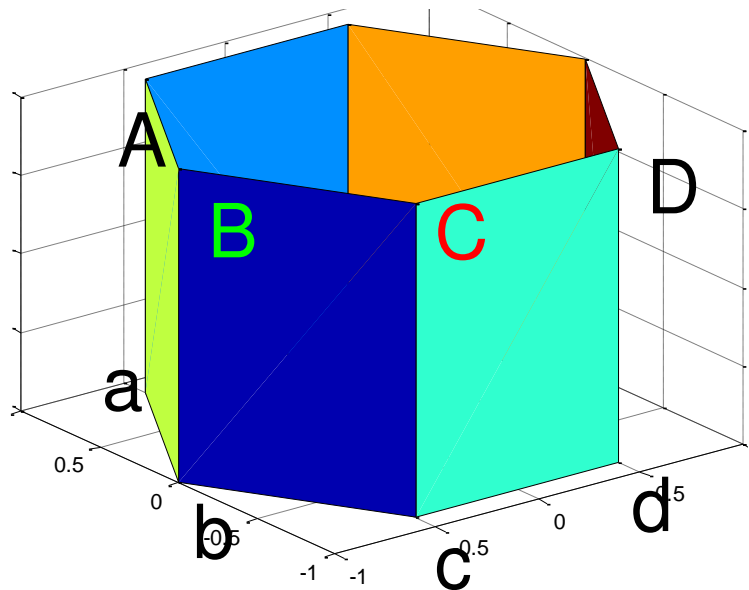
%绘图数组

X = [R*cos(t); zeros(size(t))]

Y = [R*sin(t); zeros(size(t))]

Z = z*ones(size(X))

(4) 用surf()绘制正棱柱（台、锥）面



```
z1 = 0 %底面所在的平面（平行于xoy）  
z2 = 5 %顶面面所在的平面（平行于xoy）  
t = 0:2*pi/N:2*pi
```

%侧面

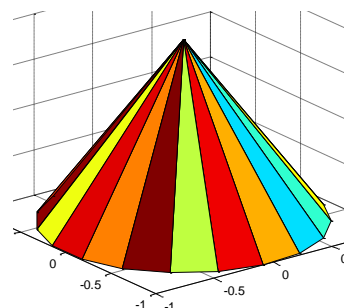
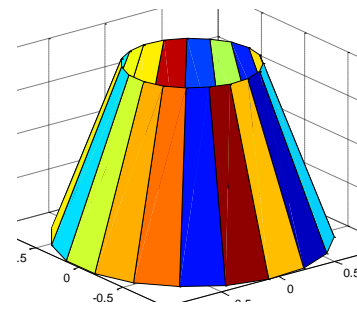
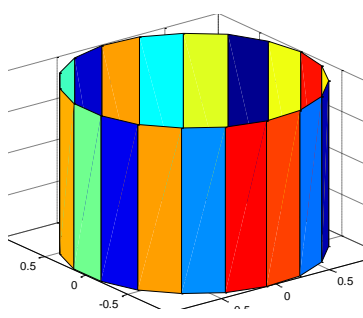
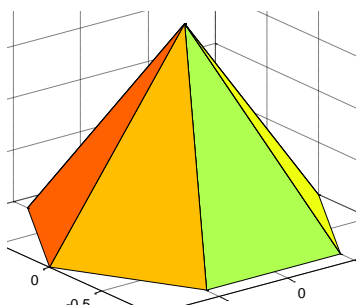
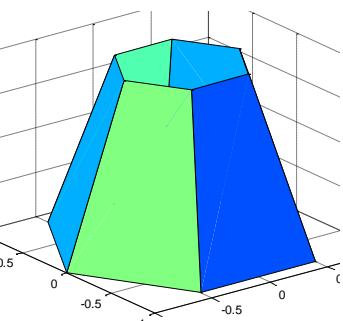
```
X = [R*cos(t); r*cos(t)]
```

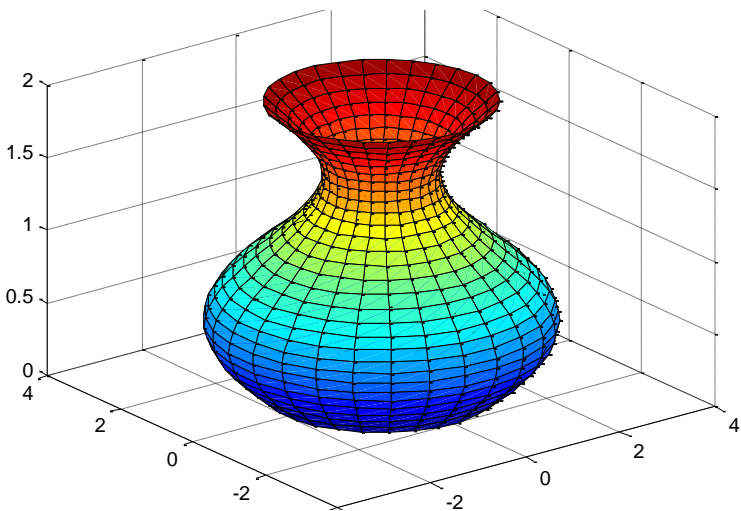
```
Y = [R*sin(t); r*sin(t)]
```

```
Z = [z1*ones(size(t)); z2*ones(size(t))]
```

$$\begin{bmatrix} A & B & C & \dots & A \\ a & b & c & \dots & a \end{bmatrix}$$

按这样的方式创建出X，Y，Z数组，即可绘制出正棱柱的侧面





设底面中心在(0,0,0)

$z1 = 0$ %底面所在的平面

$z2 = 2$ %顶面面所在的平面

$M = 20$ % 纬线数目

$N = 20$ % 经线数目

横截面半径变化规律: $r = 2 + \sin \theta$

$$X = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} \begin{bmatrix} \cos t_1 & \cos t_2 & \cdots & \cos t_n \end{bmatrix}$$

$$= \begin{bmatrix} r_1 \cos t_1 & r_1 \cos t_2 & \cdots & r_1 \cos t_n \\ r_2 \cos t_1 & r_2 \cos t_2 & \cdots & r_2 \cos t_n \\ \vdots & \vdots & \cdots & \vdots \\ r_m \cos t_1 & r_m \cos t_2 & \cdots & r_m \cos t_n \end{bmatrix}$$

```
t = linspace(0,2*pi,N)
```

```
s = linspace(0,2*pi, M)'
```

```
r = (2 + sin(s))
```

```
h = linspace(z1,z2, M)'
```

```
X = r*cos(t);
```

```
Y = r*sin(t);
```

```
Z = h*ones(size(t))
```

```
surf(X,Y,Z)
```

列×行 与 meshgrid() 的联系

$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}, \quad t = [t_1 \quad t_2 \quad \cdots \quad t_n]$$

$$[T, R] = \text{meshgrid}(t, r)$$

$$T = \begin{bmatrix} t_1 & t_2 & \cdots & t_n \\ t_1 & t_2 & \cdots & t_n \\ \vdots & \vdots & \ddots & \vdots \\ t_1 & t_2 & \cdots & t_n \end{bmatrix}$$

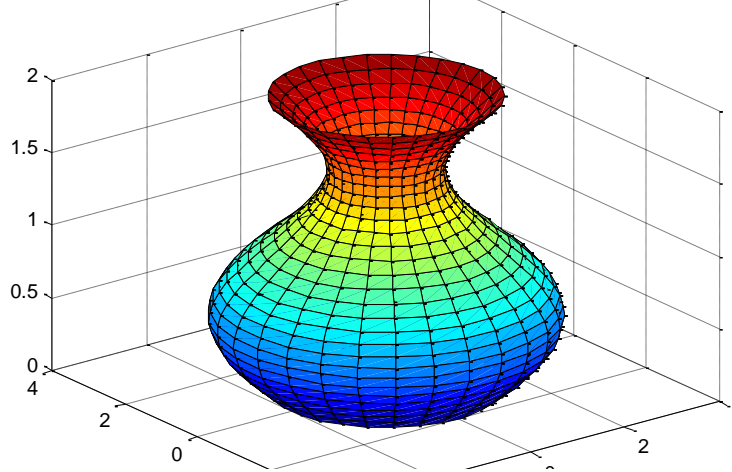
$$R = \begin{bmatrix} r_1 & r_1 & \cdots & r_1 \\ r_2 & r_2 & \cdots & r_2 \\ \vdots & \vdots & \ddots & \vdots \\ r_m & r_m & \cdots & r_m \end{bmatrix}$$

$$X = r * \cos(t)$$

$$= \begin{bmatrix} r_1 \cos t_1 & r_1 \cos t_2 & \cdots & r_1 \cos t_n \\ r_2 \cos t_1 & r_2 \cos t_2 & \cdots & r_2 \cos t_n \\ \vdots & \vdots & \ddots & \vdots \\ r_m \cos t_1 & r_m \cos t_2 & \cdots & r_m \cos t_n \end{bmatrix}$$

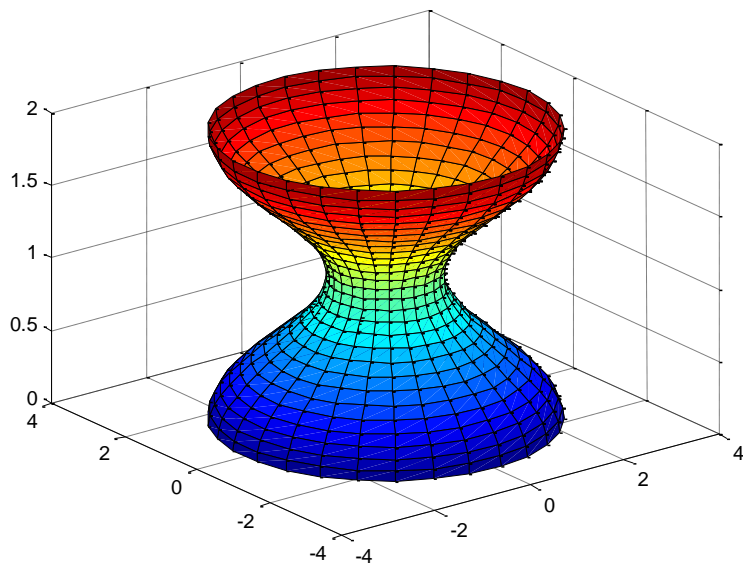
$$= \begin{bmatrix} r_1 & r_1 & \cdots & r_1 \\ r_2 & r_2 & \cdots & r_2 \\ \vdots & \vdots & \ddots & \vdots \\ r_m & r_m & \cdots & r_m \end{bmatrix} \bullet * \cos \left(\begin{bmatrix} t_1 & t_2 & \cdots & t_n \\ t_1 & t_2 & \cdots & t_n \\ \vdots & \vdots & \ddots & \vdots \\ t_1 & t_2 & \cdots & t_n \end{bmatrix} \right)$$

$$= R \bullet * \cos(T)$$



```
t = linspace(0,2*pi,N)
s = linspace(0,2*pi, M)'
r = (2 + sin(s))
h = linspace(z1,z2, M)'
X = r*cos(t);
Y = r*sin(t);
Z = h*ones(size(t))
surf(X,Y,Z)
```

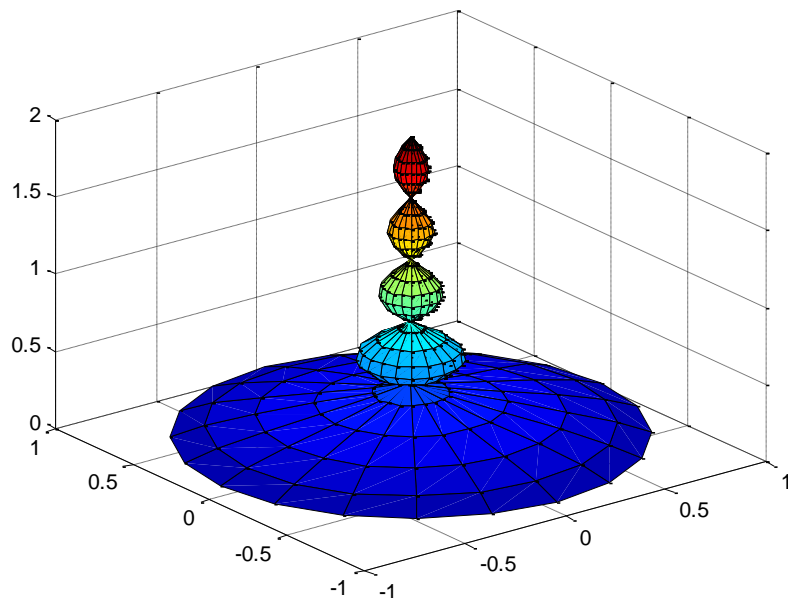
```
t = linspace(0,2*pi,N)
s = linspace(0,2*pi, M)'
r = (2 + sin(s))
h = linspace(z1,z2, M)'
[T,R] = meshgrid(t,r)
[T,H] = meshgrid(t,h)
X = R.*cos(T);
Y = R.*sin(T);
Z = H
surf(X,Y,Z)
```



横截面半径变化规律:

$$r = 2 + \cos \theta$$

$$\theta \in [0, 2\pi]$$



横截面半径变化规律:

$$r = \frac{\sin \theta}{\theta}$$

$$\theta \in [0, 5\pi]$$

(5) 用surf()绘制球面

方法1：用 矩阵乘法 生成二维数组

```
r = 2    %球半径
```

```
M = 30   %经线数
```

```
N = 30   %纬线数
```

```
phi = 0:2*pi/M:2*pi
```

```
theta = linspace(0,pi,N)'
```

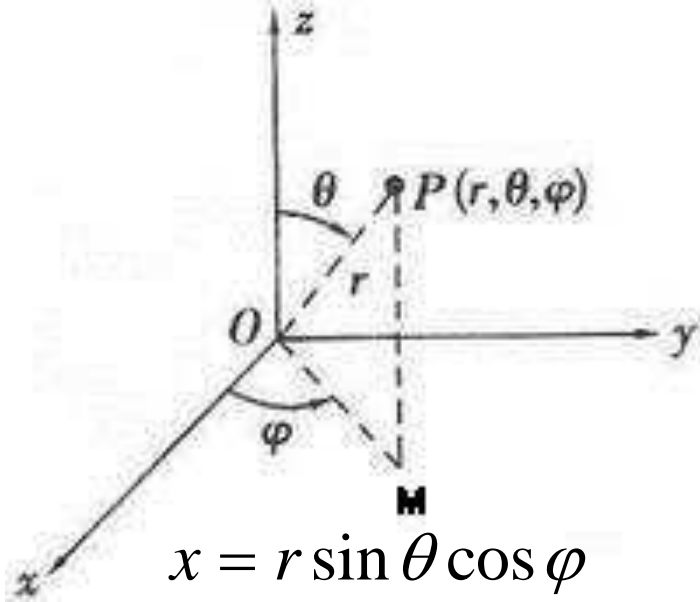
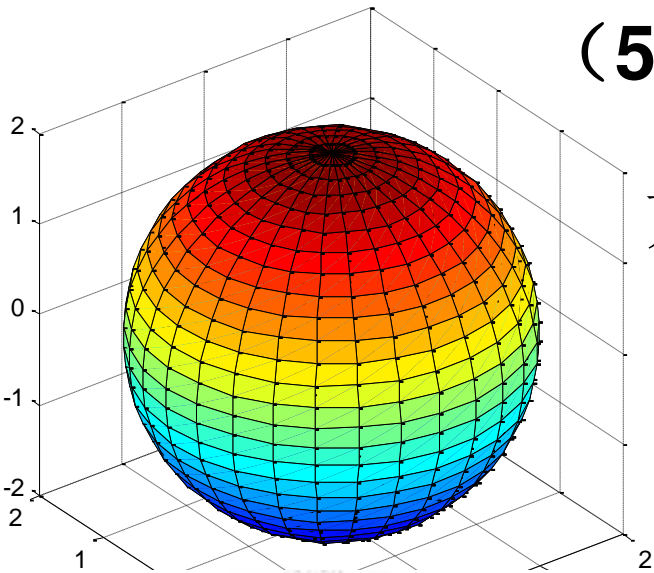
```
X = r*sin(theta)*cos(phi);
```

```
Y = r*sin(theta)*sin(phi);
```

```
Z = r*cos(theta)*ones(size(phi))
```

```
surf(X,Y,Z)
```

```
axis square
```



$$x = r \sin \theta \cos \varphi$$

$$y = r \sin \theta \sin \varphi$$

$$z = r \cos \theta$$

方法2: 用 meshgrid 生成二维数组

$$x = r \sin \theta \cos \varphi$$

$$y = r \sin \theta \sin \varphi$$

$$z = r \cos \theta$$



```
r = 2;  
theta = linspace(0,pi,20);  
phi = linspace(0,2*pi,21);  
[T,P]=meshgrid(theta,phi);  
X = r.*sin(T).*cos(P);  
Y = r.*sin(T).*sin(P);  
Z = r.*cos(T);  
surf(X,Y,Z)  
axis square
```

r 确定的情况下, X, Y, Z 都是自变量 θ 和 φ 的二元函数, 可以利用 *meshgrid*() 生成二个自变量 θ 和 φ 的二维数据网格, 然后计算 X, Y, Z 的值。

练习：用surf()绘制椭球面

中心在原点，三条半主轴分别在X、Y、Z轴的椭球面在直角坐标系中的方程是：

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

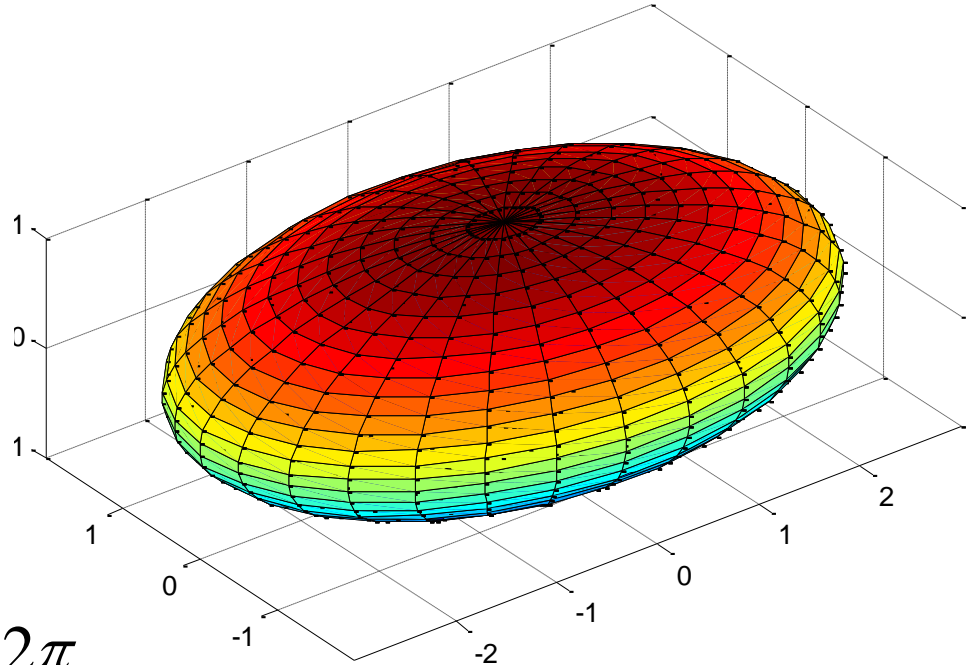
使用球坐标系：

$$x = a \sin \theta \cos \varphi$$

$$y = b \sin \theta \sin \varphi$$

$$z = c \cos \theta$$

其中 $0 \leq \theta \leq \pi$, $0 \leq \varphi \leq 2\pi$



试在Matlab中绘制一椭球面

$$a = 3, b = 2, c = 1$$

练习：用surf()绘制圆环面

如右下图所示，中心在 $(0,0,0)$ 平行于 XOY 平面的圆环面上任意点的直角坐标可表示为：

$$x = (R + r \sin \theta) \cos \varphi$$

$$y = (R + r \sin \theta) \sin \varphi$$

$$z = r \cos \theta$$

其中 $0 \leq \theta \leq 2\pi$ ， $0 \leq \varphi \leq 2\pi$

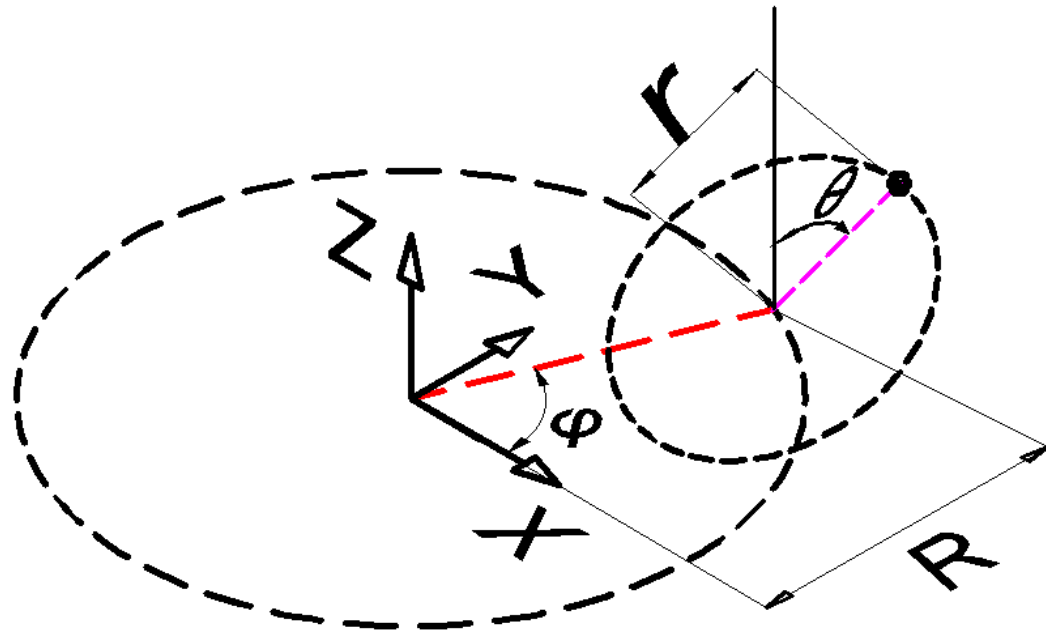
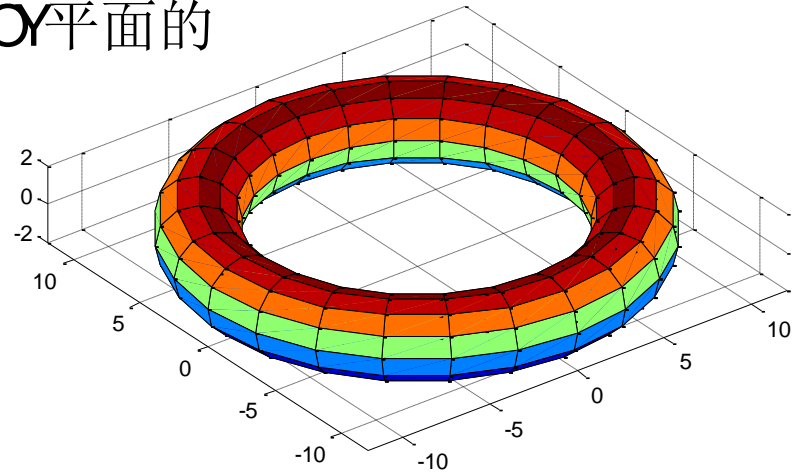
试在Matlab中绘制一圆环面：

$$R = 10$$

$$r = 2,$$

$$\theta = 0 : \frac{\pi}{24} : 2\pi$$

$$\varphi = 0 : \frac{\pi}{12} : 2\pi$$



Matlab提供的绘制柱面的函数cylinder

[X,Y,Z]=cylinder 返回一半径为1、高度为1的圆柱面的x-, y-, z-轴的坐标值, 圆柱面的圆周有20个距离相同的点。

[X,Y,Z]=cylinder(r) 返回一半径为r、高度为1的柱面的x-, y-, z-轴的坐标值, 柱面的圆周有20个距离相同的点。

[X,Y,Z]=cylinder(r,n) 返回一半径为r、高度为1的柱面的x-, y-, z-轴的坐标值, 圆柱面的圆周有指定的n个距离相同的点

cylinder(...) 画出柱面。

```
t = 0:pi/10:2*pi;  
[X,Y,Z] = cylinder(2+cos(t));  
surf(X,Y,Z)  
axis square
```

Matlab提供的绘制球面的函数sphere

sphere 生成三维直角坐标系中的单位球体。该单位球体有 20×20 个面。

sphere(n) 在当前坐标系中画出有 $n \times n$ 个面的球体

[X,Y,Z] = sphere(n)

返回三个阶数为 $(n+1) \times (n+1)$ 的直角坐标系中的二维坐标数组。该命令没有画图，只是返回矩阵。用户可以用命令 **surf(X, Y, Z)**或**mesh(X, Y, Z)**画出单位球体球体也可以直接用**sphere(n)**直接画出球体

```
sphere  
axis equal
```

Matlab提供的绘制椭球面的函数`ellipsoid`

`[x,y,z] = ellipsoid(xc,yc,zc, a,b,c,n)`

返回绘图数据，`x`，`y`，`z`均为 $(n+1) \times (n+1)$ 的二维数组

`[x,y,z] = ellipsoid(xc,yc,zc,a,b,c)`

返回绘图数据，`n = 20`

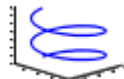
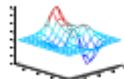

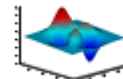
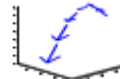
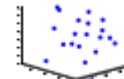

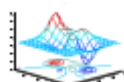

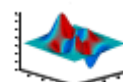
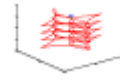
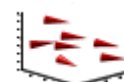



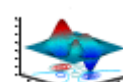
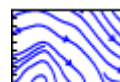
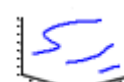

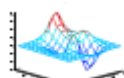

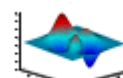

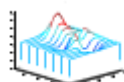
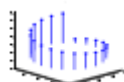

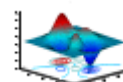
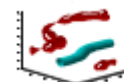
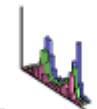

`ellipsoid(axes_handle,...)`在指定坐标轴画出椭球面

`ellipsoid(...)` 画出椭球面

$$\frac{(x - xc)^2}{a^2} + \frac{(y - yc)^2}{b^2} + \frac{(z - zc)^2}{c^2} = 1$$

```
[x, y, z] = ellipsoid(0,0,0,3,2,1,30);  
surf(x, y, z)  
axis equal
```

6、三维绘图函数汇总

Line Graphs	Mesh Graphs and Bar Graphs	Area Graphs and Constructive Objects	Surface Graphs	Direction Graphs	Volumetric Graphs
plot3 	mesh 	pie3 	surf 	quiver3 	scatter3 
contour3 	meshc 	fill3 	surfl 	comet3 	coneplot 
contourslice 	meshz 	patch 	surfz 	streamslice 	streamline 
ezplot3 	ezmesh 	cylinder 	ezsurf 	streamribbon 	
waterfall 	stem3 	ellipsoid 	ezsurfz 	streamtube 	
	bar3 	sphere 			
	bar3h 