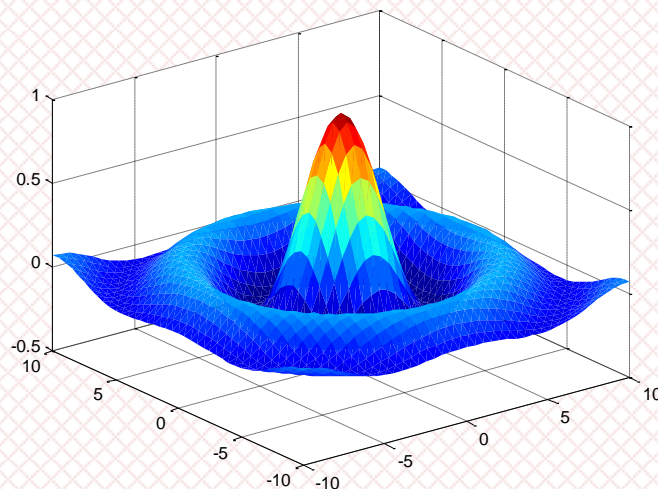




第二章 Matlab图形绘制



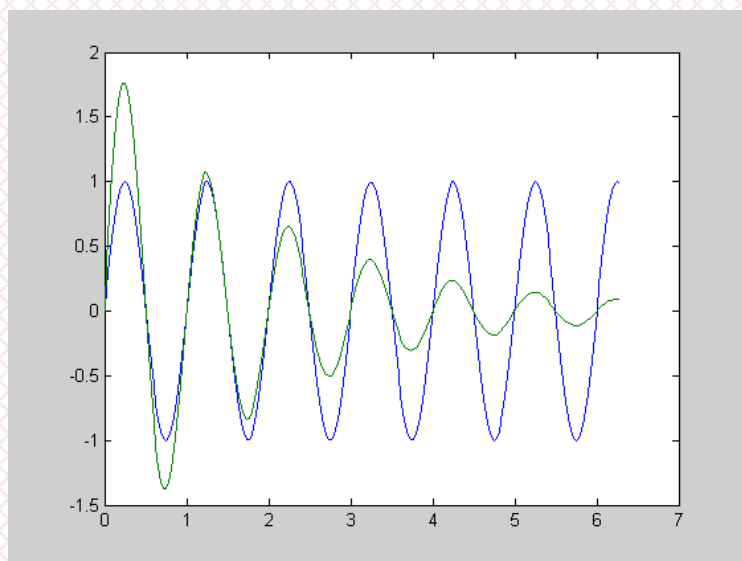


MATLAB图形绘制

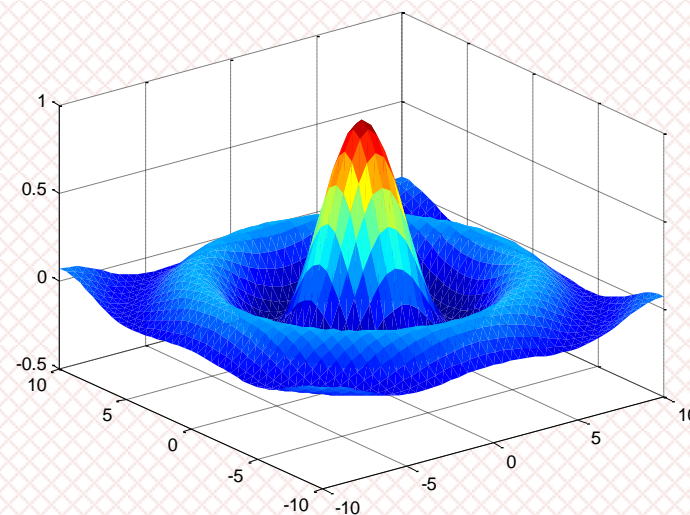
MATLAB提供多种图形功能，使数据或函数可视化，使数据不再是枯燥乏味的。使用MATLAB的图形函数，可以绘制二维或三维的数据图形和函数图形，如数据的散点图、直方图、茎干图、饼图、阶梯图和面积图等。

MATLAB图形绘制

1 二维图形



2 三维图形



两个例子

- 画出函数 $y = \sin t \cdot \sin(9t) \quad t \in [0, 2\pi]$ 的图形。
- 绘制表格数据

中美两国在**1950**年到**2015**年期间的年钢产量统计数据如下表所示，请进行对比说明。

1950	1955	1960	1965	1970	1975	1980	1985	1990	1995	2000	2005	2010	2015
61	285	1351	1223	1779	2390	3712	4679	6635	9536	12850	34940	63874	80380
8785	10617	9007	11926	11931	10582	10146	8006	8972	9359	10182	9490	8050	7880

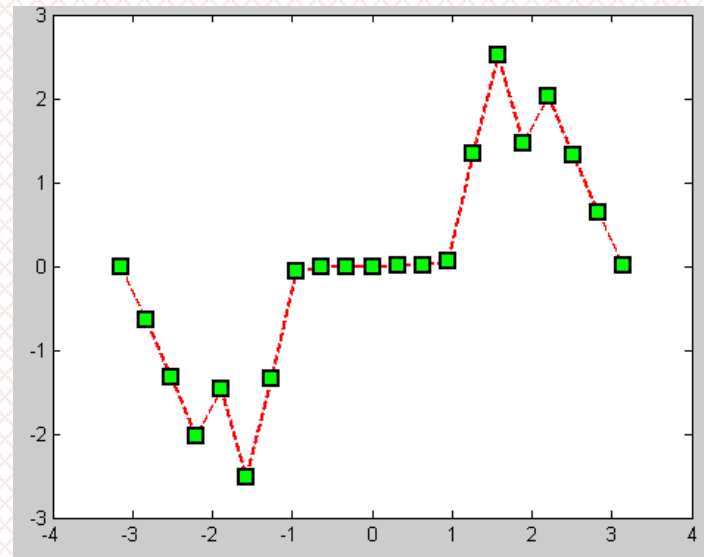
2.1 二维图形

数据可视化的基本步骤是：

- (1) 准备好数据；
- (2) 选择适用的绘制图形函数；
- (3) 选择窗口和位置；
- (4) 编辑图形标注和说明；
- (5) 输出或保存图形。

2.1 二维图形

- 1、绘制二维曲线的最基本函数
- 2、图形窗口 (**Figure**)
- 3、绘制二维图形的其它函数



1. 绘制二维曲线的最基本函数

(1) plot函数的基本用法

plot函数的基本调用格式为:

plot(x,y)

- 其中x和y为长度相同的向量，分别用于存储x坐标和y坐标数据。条件是元素个数能对应。
- 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

Matlab 作图的基本步骤

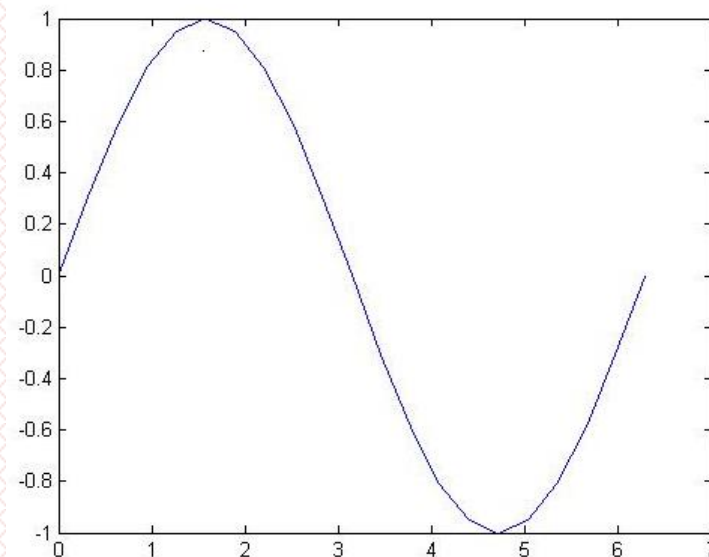
1、准备绘图数据:

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

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

2、调用绘图函数作出图形,

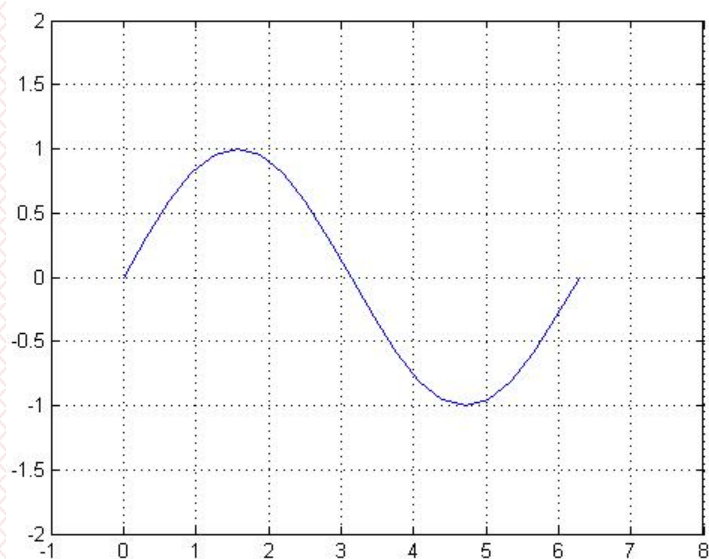
如: `plot(x,y)`



3、调用相关函数和命令调整图形特性, 如:

```
grid on
```

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



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：

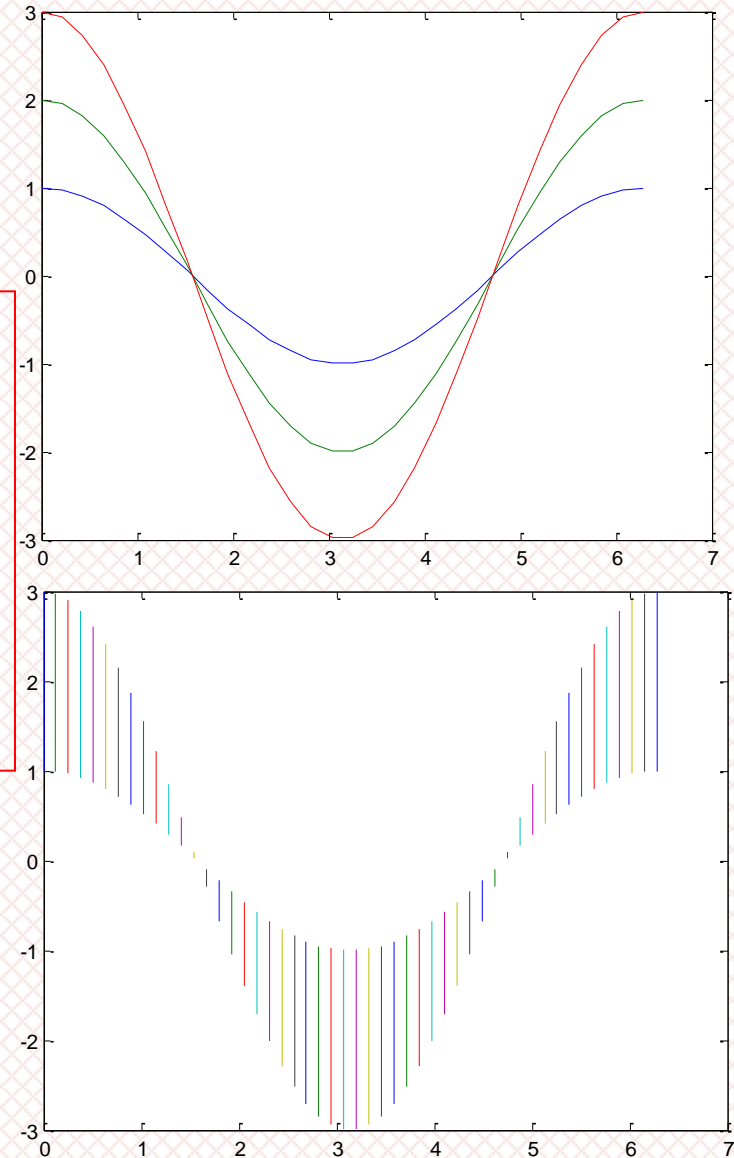
```
plot(real(Y),imag(Y))
```

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')
```

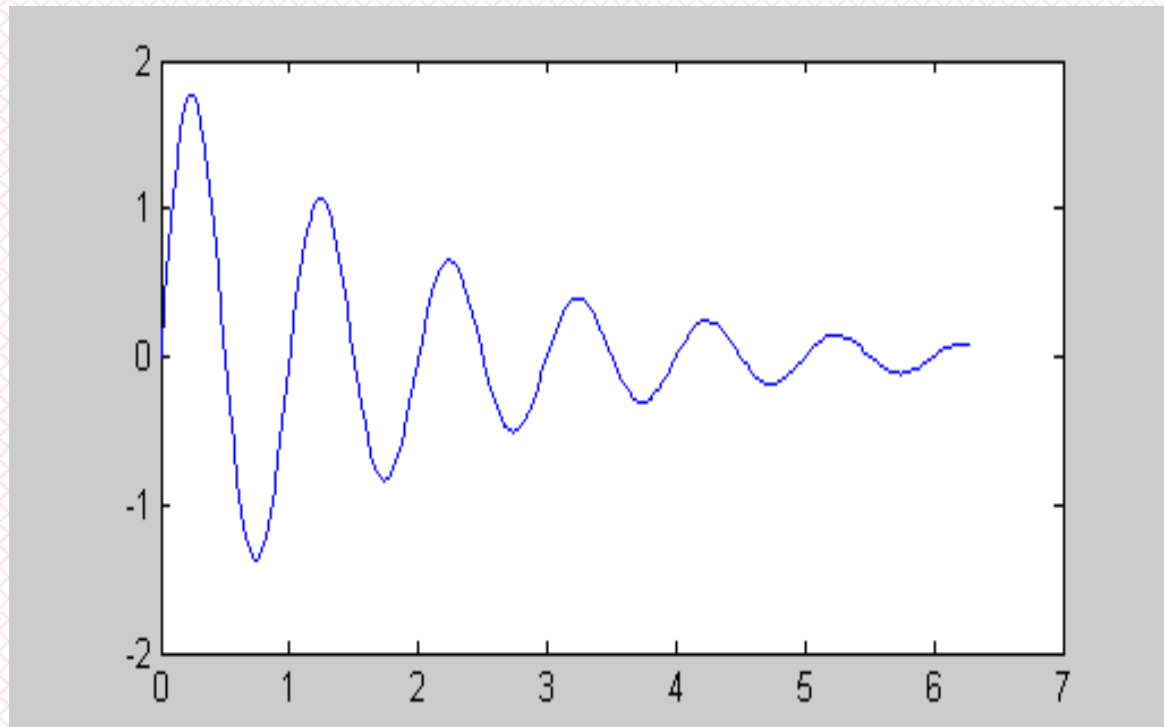


例1 在 $0 \leq x \leq 2\pi$ 区间内，绘制曲线

$$y = 2e^{-0.5x} \sin(2\pi x)$$

程序如下：

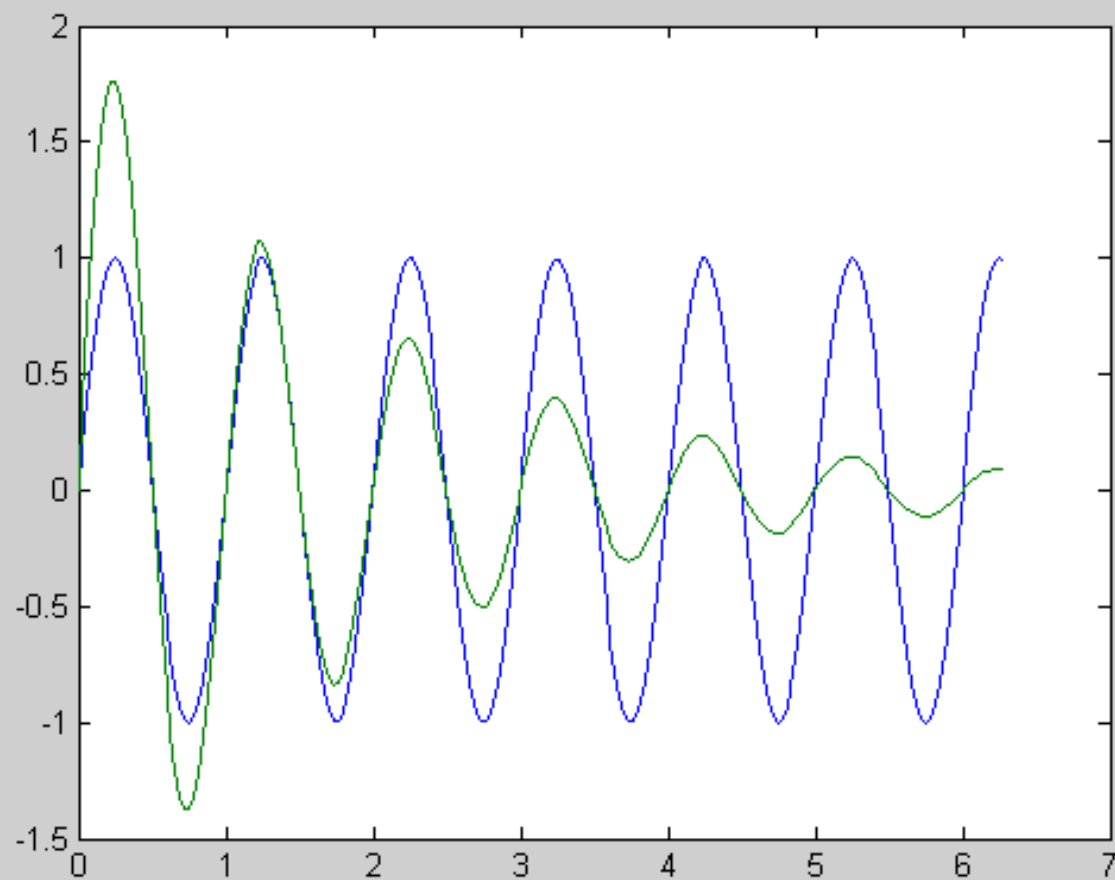
```
>> x=0:pi/100:2*pi;  
y=2*exp(-0.5*x).*sin(2*pi*x);  
plot(x,y)
```



说明:

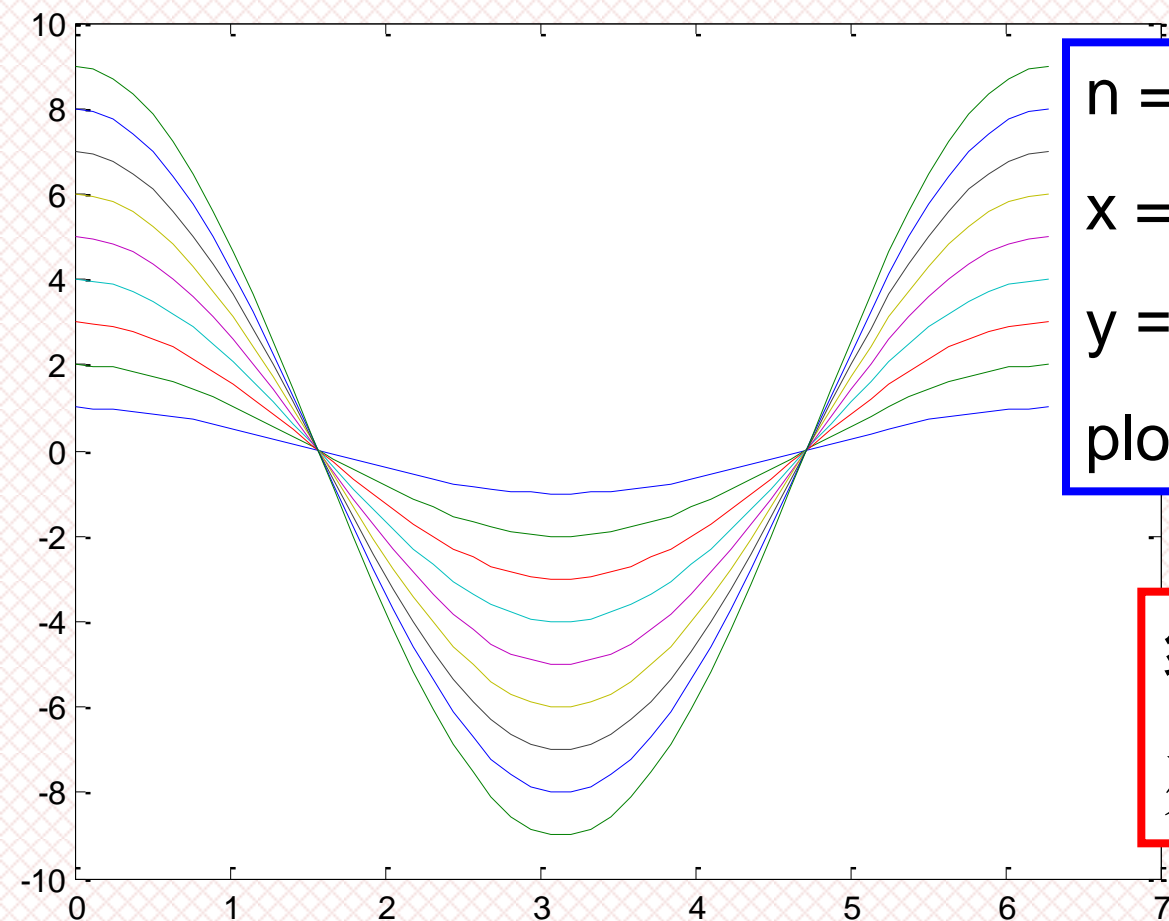
- (1) 当 x, y 是同维矩阵时，则以 x, y 对应列元素为横、纵坐标分别绘制曲线，曲线条数等于矩阵的列数。
- (2) 当 x 是向量， y 是有一维与 x 同维的矩阵时，则绘制出多根不同色彩的曲线。曲线条数等于 y 矩阵的另一维数， x 被作为这些曲线共同的横坐标。

```
>> x=0:pi/100:2*pi;  
    y=[ sin(2*pi*x); 2*exp(-0.5*x).*sin(2*pi*x)];  
    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：

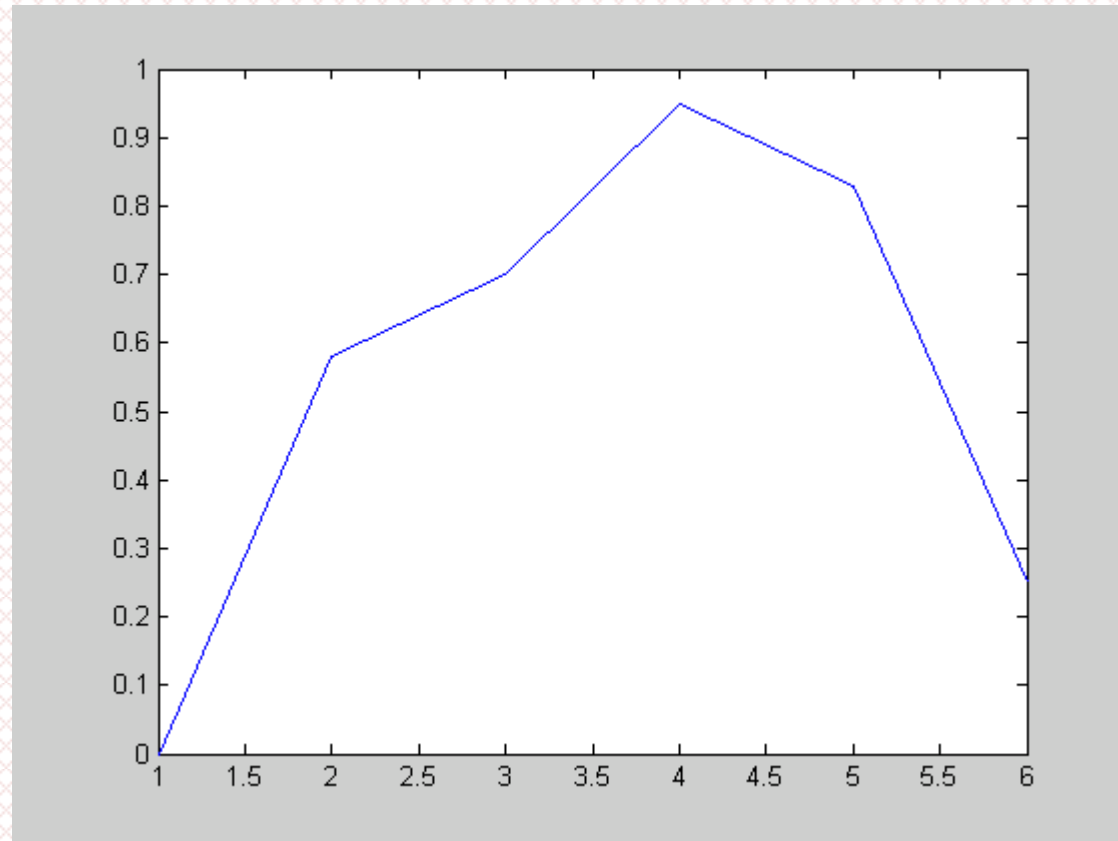
列 \times 行 = 矩阵

(3) **plot**函数最简单的调用格式是只包含一个输入参数:

plot(y)

即绘制以向量Y元素值为纵坐标（假如元素为实数），它的下标值为横坐标的线性图(即绘制Y的列向量对其坐标索引的图形)。

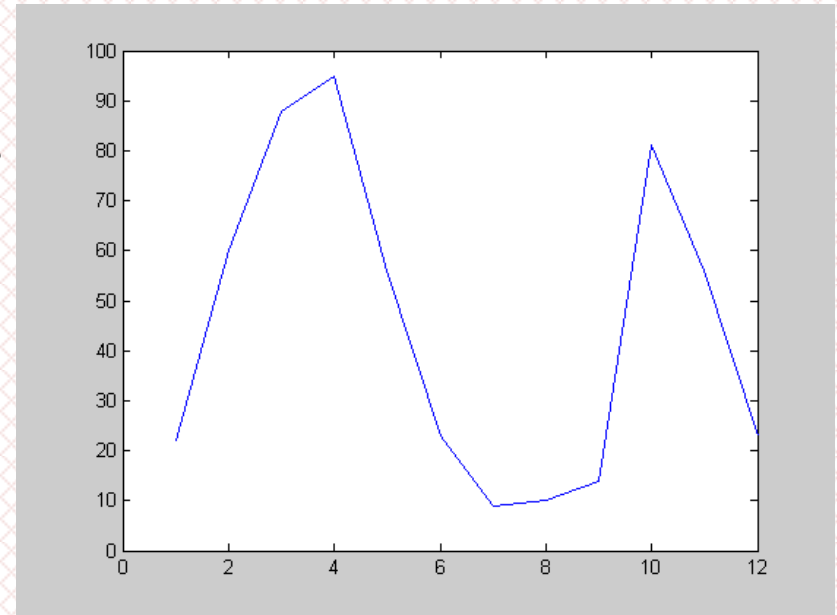

```
>> y=[0 0.58 0.7 0.95 0.83 0.25];  
>> plot(y); %实际上是画折线
```



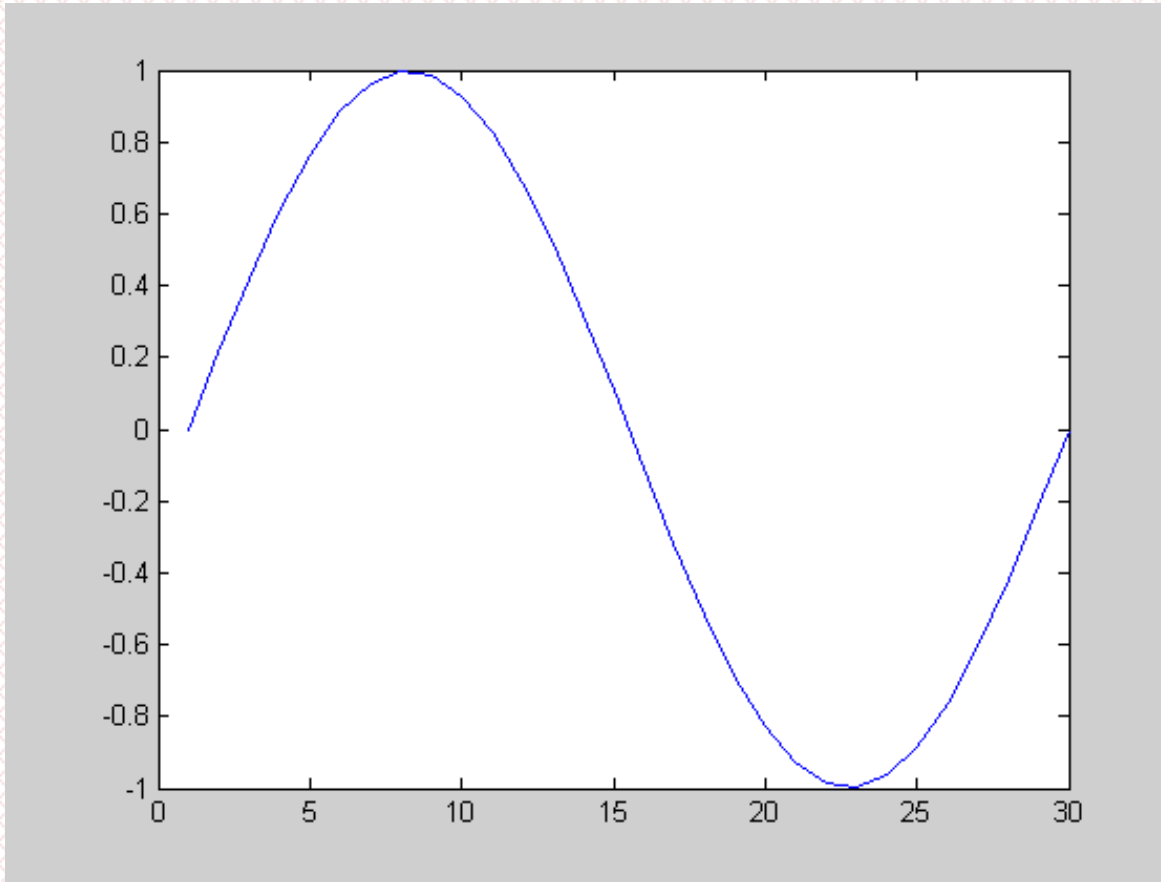
例2 某工厂2000年各月总产值（单位：万元）分别为22、60、88、95、56、23、9、10、14、81、56、23，试绘制折线图以显示出该厂总产值的变化情况。

程序如下：

```
>>p=[22,60,88,95,56,23,9,10,14,81,56,23];  
plot(p)
```



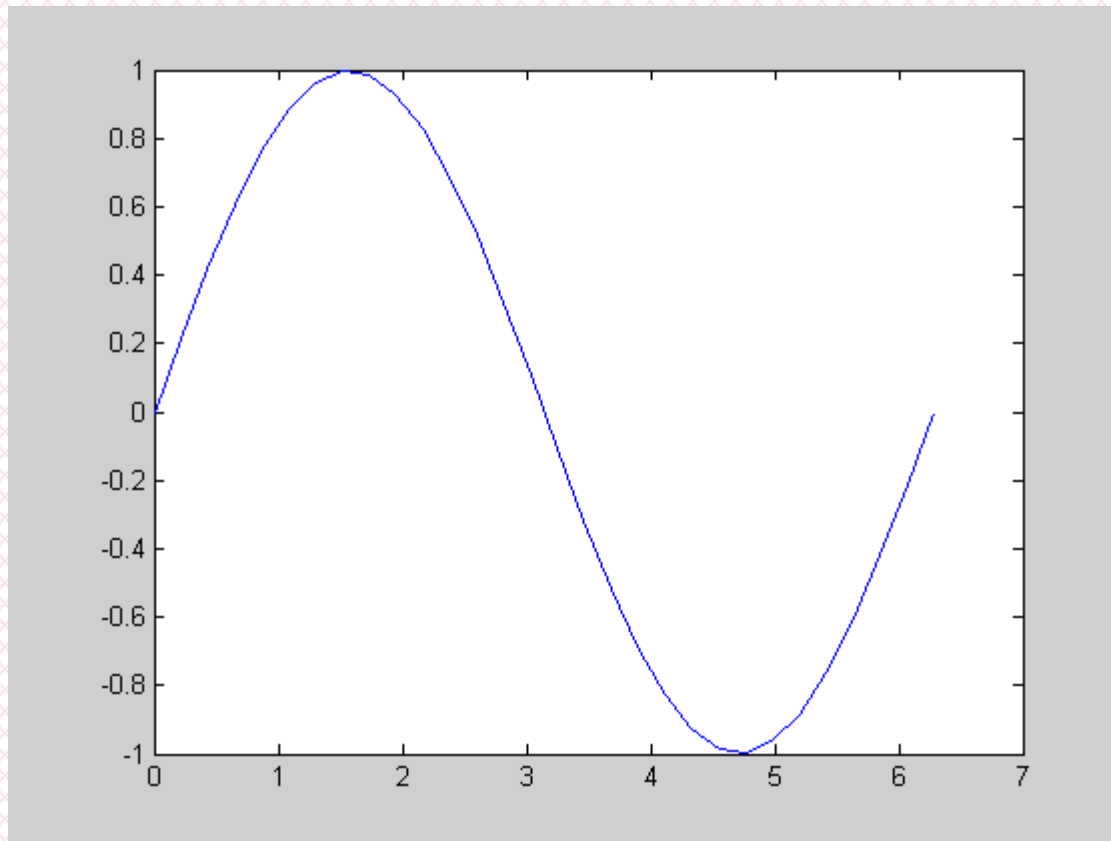
```
x=linspace(0,2*pi,30); y=sin(x); plot(y)
```



注意

1. 为29等分，30维向量，
2. 与 `plot(y)` 的区别.

```
>>x=linspace(0,2*pi,30); y=sin(x);  
>> plot(x,y)
```



(2) 含多个输入参数的plot函数

含多个输入参数的plot函数调用格式为:

plot(x1,y1,x2,y2,...,xn,yn)

含选项的plot函数调用格式为:

plot(x1,y1,选项1,x2,y2, 选项2,...,xn,yn,选项n)

选项参数option定义了图形曲线的颜色、线型及标示符号，它由一对单引号括起来。



线型

线方式: - 实线, : 点线, -. 虚点线, -- 波折线

标记点

. (圆点), +(加号), *(星号), x(叉号), o(圆点), s(方块), d(菱形), p(五角星), h(六角形)

线的颜色: y yellow, r red, g green, b blue,
w write, k black, c cyan(青色)

在不指定时, 默认实线方式, 颜色自动确定. 不同种类的选项可搭配使用, 如选项“ro”表示绘制红色的圆划线, “y-”表示黄色的实划线。

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

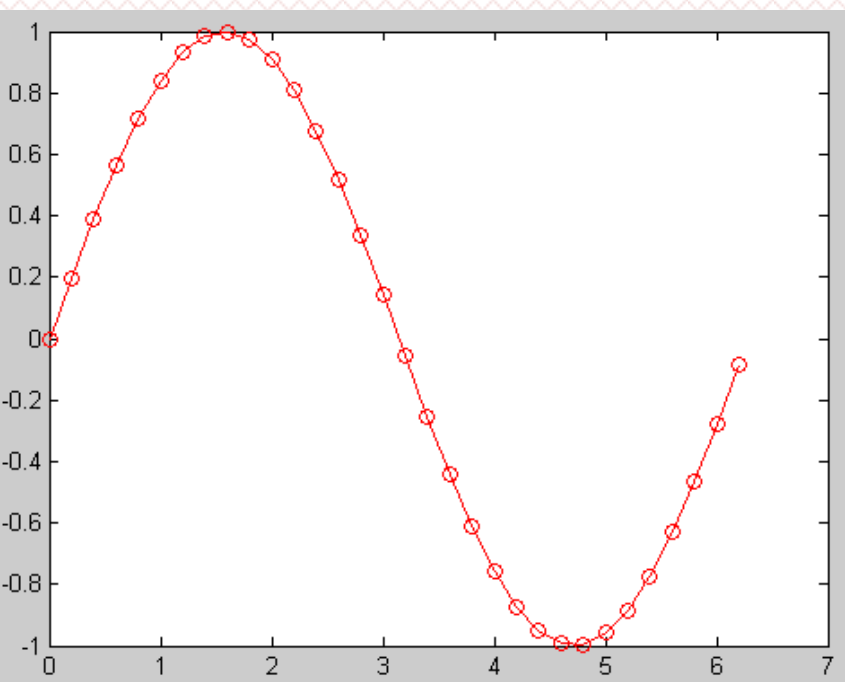
按照三个参数Xn, Yn, LineSpec画线，其中LineSpec指定了线型，点标记和画线的颜色，也可以混合使用三参数Xn,Yn,LineSpec和二参数Xn,Yn:

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			

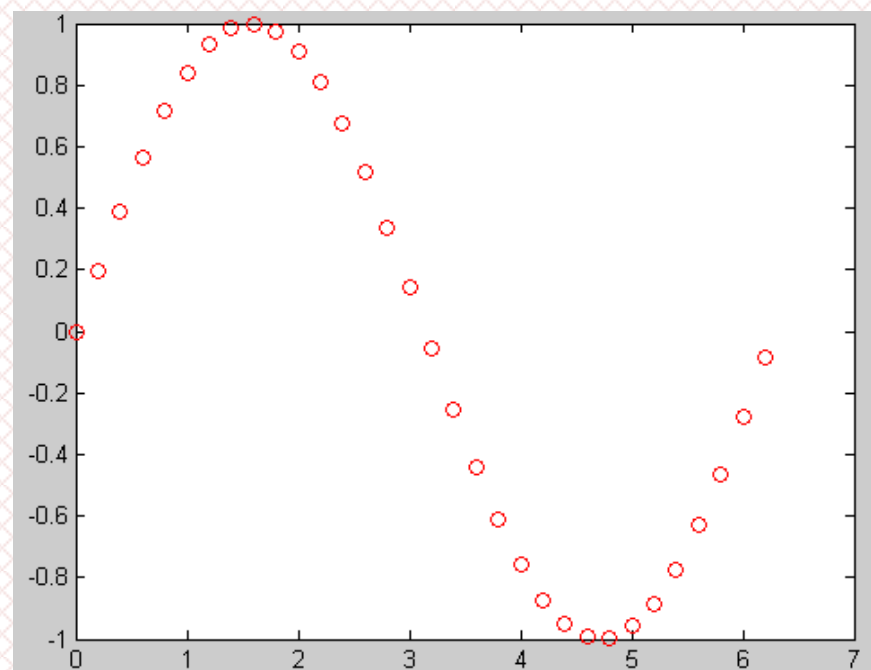
```
x = 0:0.2:2*pi;  
y = sin(x);  
plot(x,y,'r-o')
```



```
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

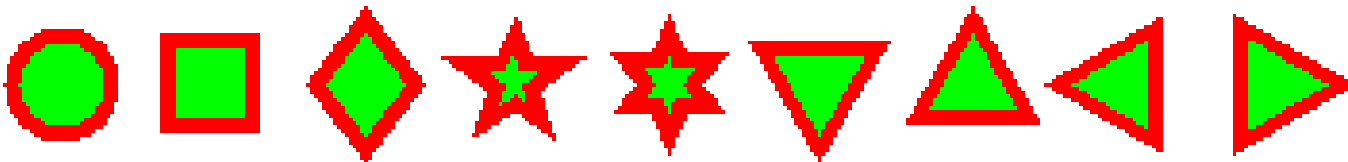
MarkerEdgeColor marker的颜色(或 **filled markers** 边的颜色)

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

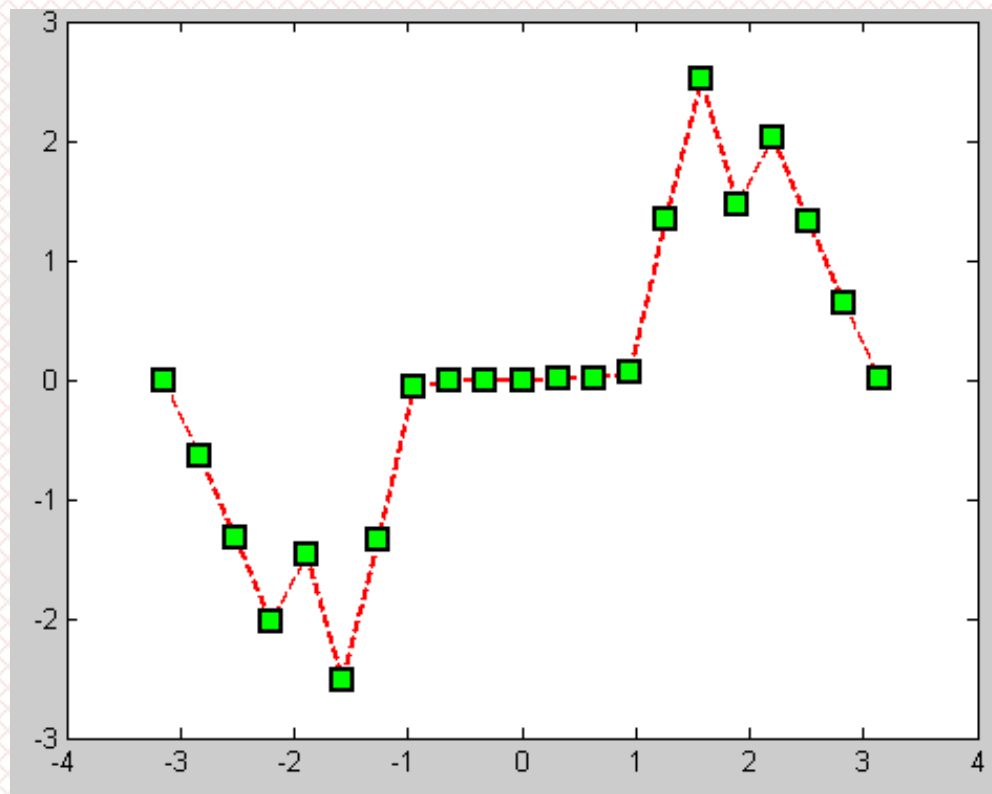
MarkerSize 点标记的大小

filled markers

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



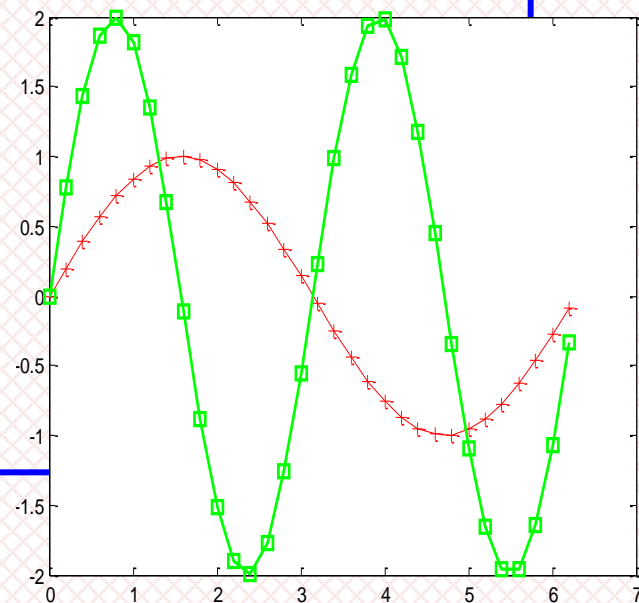
```
x = -pi:pi/10:pi;  
y = tan(sin(x)) - sin(tan(x));  
plot(x,y,'--rs','LineWidth',2,...  
      'MarkerEdgeColor','k',...  
      'MarkerFaceColor','g',...  
      'MarkerSize',10)
```



`h = plot(...)` 返回图形对象的句柄。图形中的每条曲线都返回自己的句柄（是一个双精度数）。如有多条曲线，则有多个句柄，以列向量的形式返回至 **`h`**。

句柄是图形对象的标识代码，标识代码含有图形对象的各种必要的属性信息，可用 **`get()`** 函数获取这些属性，用 **`set()`** 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')
```



例 用不同线型和颜色在同一坐标内绘制曲线 $y=\sin x$, $y=\cos x$ 的图像。

程序如下：

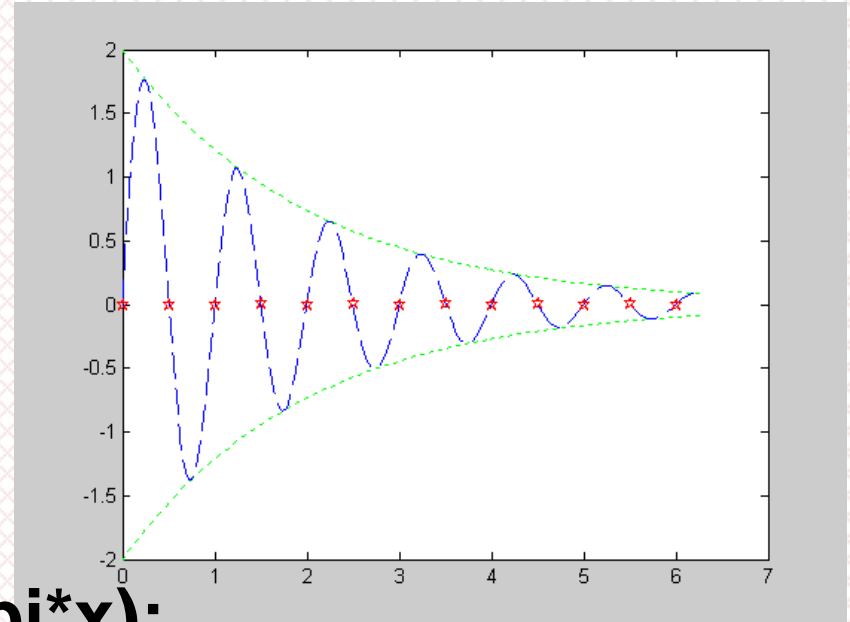
```
x=linspace(0,2*pi,100);
```

```
plot(x,sin(x),'kh',x,cos(x),'gp') %正、余弦曲线  
分别用黑色六角形、绿色五角星表示
```

例3 用不同线型和颜色在同一坐标内绘制曲线 $y=2e^{-0.5x}\sin(2\pi x)$ 及其包络线。

程序如下：

```
x=(0:pi/100:2*pi)';  
y1=2*exp(-0.5*x)*[1,-1];  
y2=2*exp(-0.5*x).*sin(2*pi*x);  
x1=(0:12)/2;  
y3=2*exp(-0.5*x1).*sin(2*pi*x1);  
plot(x,y1,'g:',x,y2,'b--',x1,y3,'rp');
```



有关图形标注函数的调用格式为：

`title(‘字符串’)`：

在所画图形的最上端显示说明该图形标题的字符串。

`xlabel(‘字符串’)`，`ylabel(‘字符串’)`：

分别为x，y坐标轴加上注解和说明，其中`ylabel`会自动旋转90°显示。

`text(x, y, ‘字符串’，‘属性名’，值)`

在图形的指定坐标位置(x, y)处，标示单引号括起来的字符串是要说明的内容，属性名及其定义字符串的显示格式。字符对象的常用属性如下：

Color 属性：1*3的颜色向量

FontAngle属性：字体倾斜形式

FontName属性：字体名称

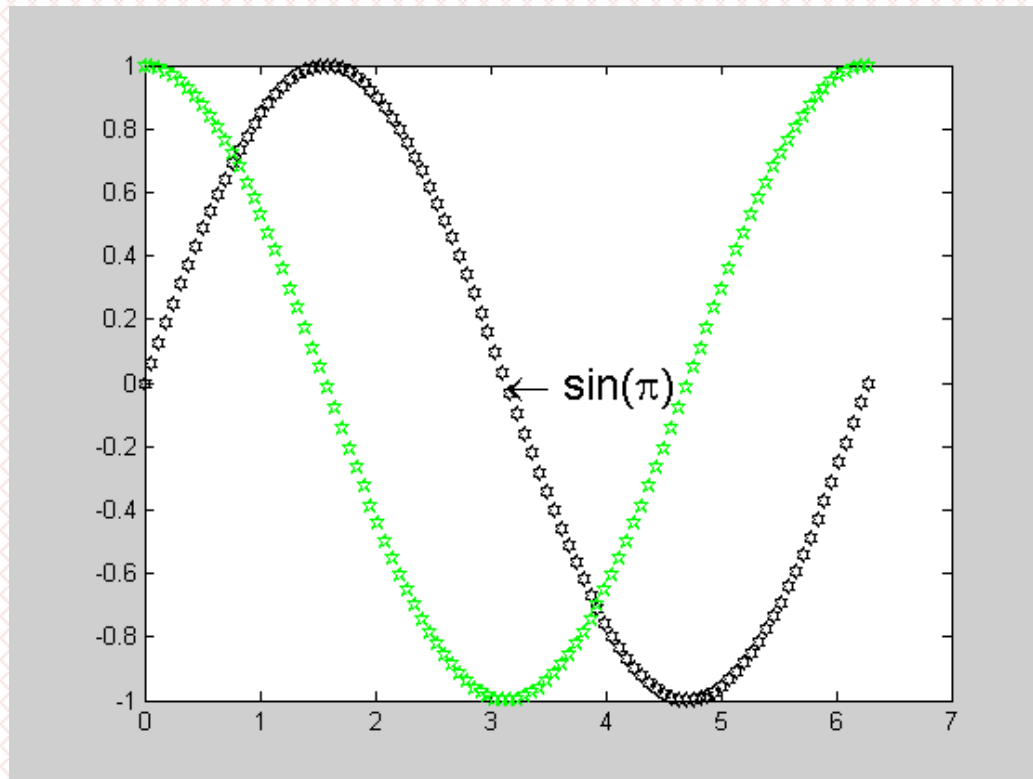
.....

`text(2.5, 0.7, ‘sinx’)` 在图中(2.5, 0.7)处加字符串
‘sinx’，或 `gtext(‘sinx’)`，用鼠标光标定位

```
x=linspace(0,2*pi,100);
```

```
plot(x,sin(x),'kh',x,cos(x),'gp')
```

```
text(pi,0,'\leftarrow sin(\pi)','fontSize',18)
```



legend('字符串1','字符串2',...,'字符串n')

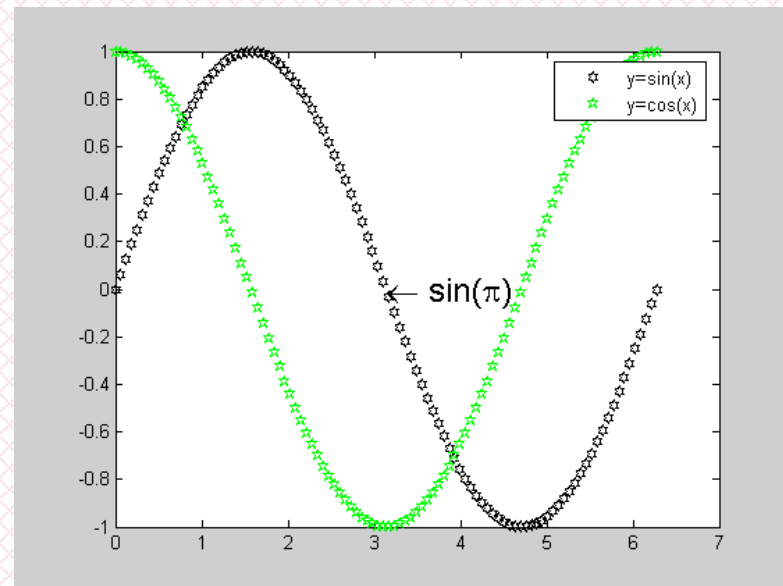
在屏幕上开启一个小视窗，然后依据绘图命令的先后次序，用对应的字符串区分图形上的线。

```
x=linspace(0,2*pi,100);  
plot(x,sin(x),'kh',x,cos(x),'gp')  
text(pi,0,'\leftarrow sin(\pi)','fontSize',18)  
legend('y=sin(x)','y=cos(x)');
```

legend('s1','s2',...,'Location',pos)

位置参数pos常取以下字符串：

- 'northeast', 图例在图形的右上角（默认位置）
- 'northwest', 图例在图形的左上角。
- 'southeast', 图例在图形的右下角。
- 'southwest', 图例在图形的左下角。

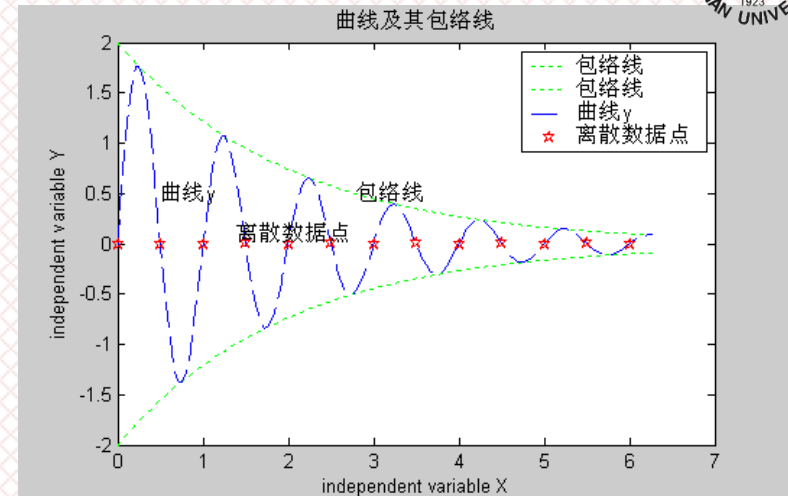


例5 给图形添加图形标注。 程序如下：

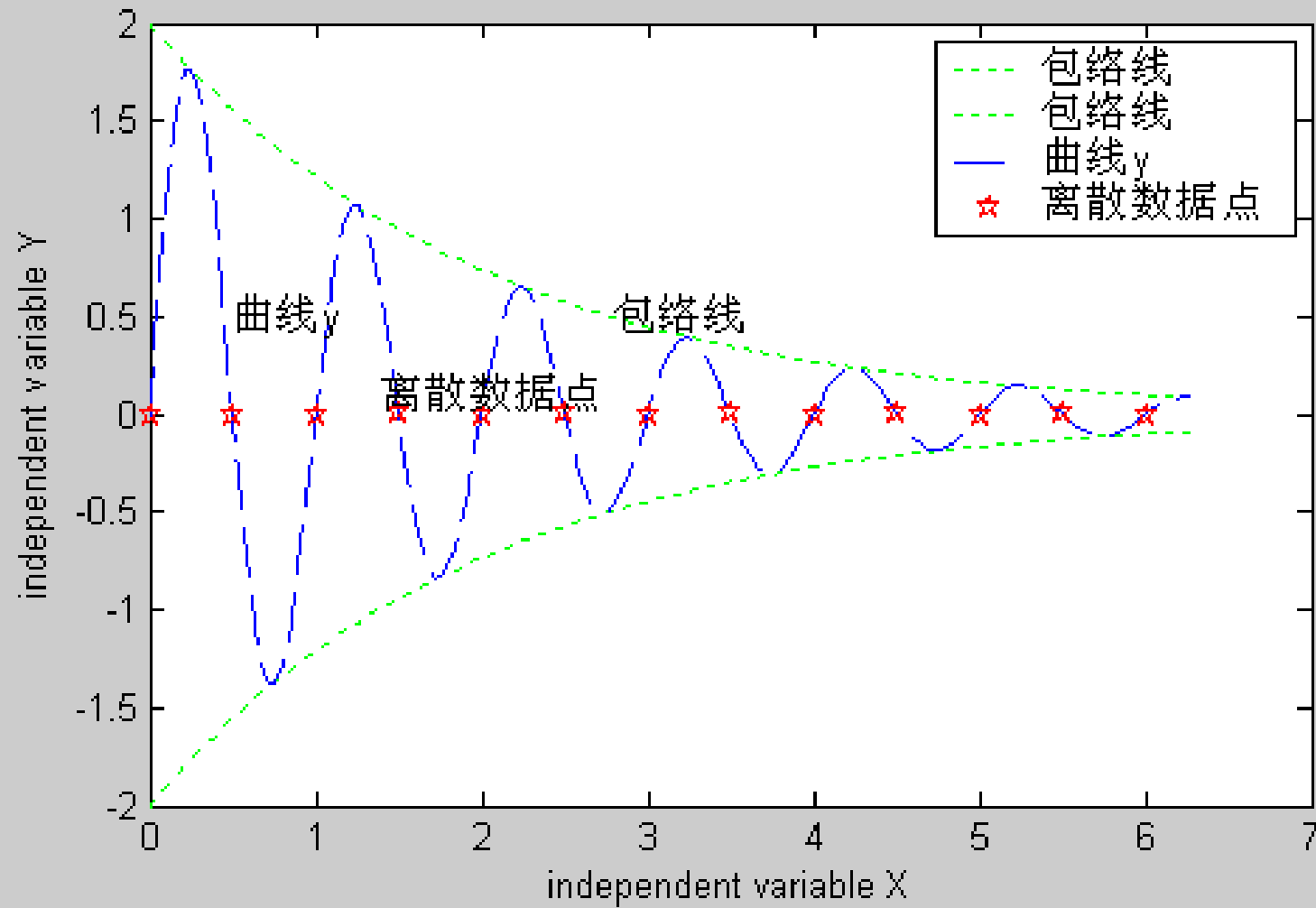
```

x=(0:pi/100:2*pi)';
y1=2*exp(-0.5*x)*[1,-1];
y2=2*exp(-0.5*x).*sin(2*pi*x);
x1=(0:12)/2;
y3=2*exp(-0.5*x1).*sin(2*pi*x1);
plot(x,y1,'g:',x,y2,'b--',x1,y3,'rp');
title('曲线及其包络线');           %加图形标题
xlabel('independent variable X');   %加X轴说明
ylabel('independent variable Y');   %加Y轴说明
gtext('包络线' );                  %在鼠标光标定位置添加图形说明
gtext('包络线'); gtext('曲线y');
gtext('离散数据点');
legend('包络线','包络线','曲线y','离散数据点') %加图例

```



曲线及其包络线



(3) 选择图像

`figure (1) ; figure (2) ; ...; figure(n)`

打开不同的图形窗口，以便绘制不同的图形。

(4) `grid on`: 在所画出的图形坐标中加入栅格

`grid off`: 除去图形坐标中的栅格

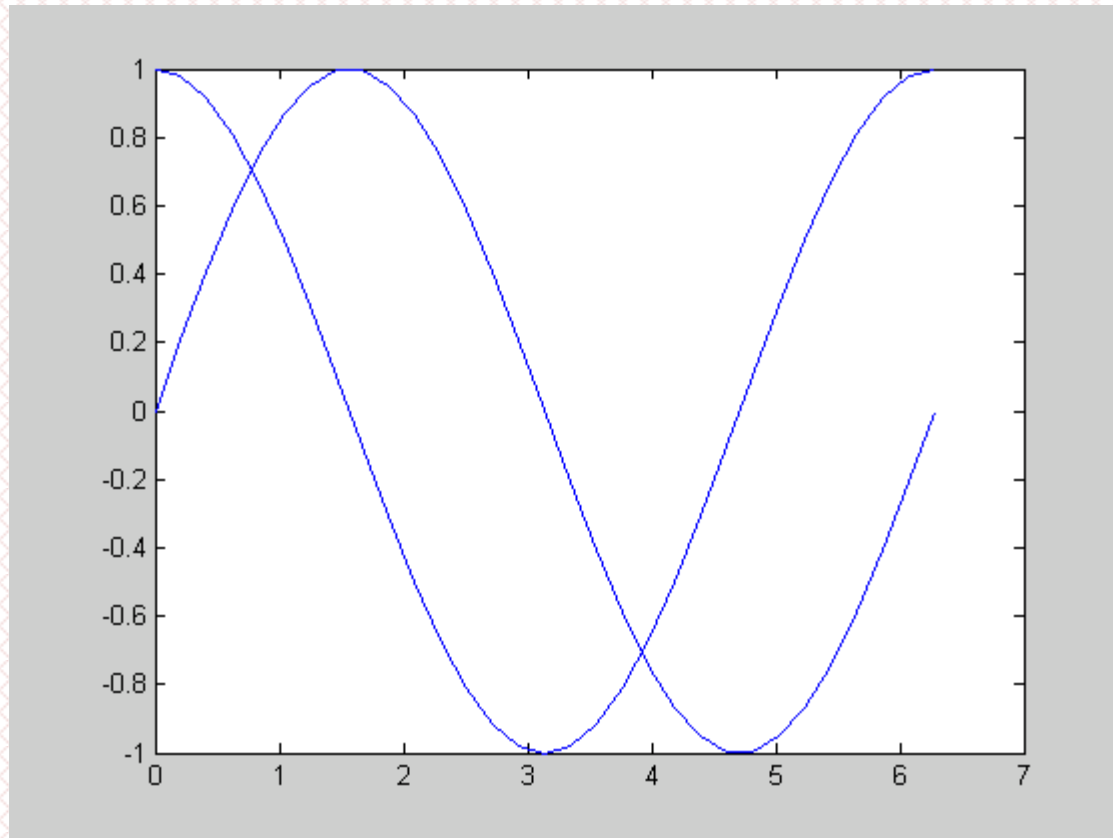
不带参数的`grid`命令在两种状态之间进行切换。

(5) `hold on`: 把当前图形保持在屏幕上不变，同时
允许在这个坐标内绘制另外一个图形。

`hold off`: 使新图覆盖旧的图形

(6) `box on/off`命令控制是加还是不加边框线，不带
参数的`box`命令在两种状态之间进行切换。

```
>> x=0:pi/15:2*pi; y=sin(x); plot(x,y),  
>> hold on, z=cos(x); plot(x,z), hold off
```

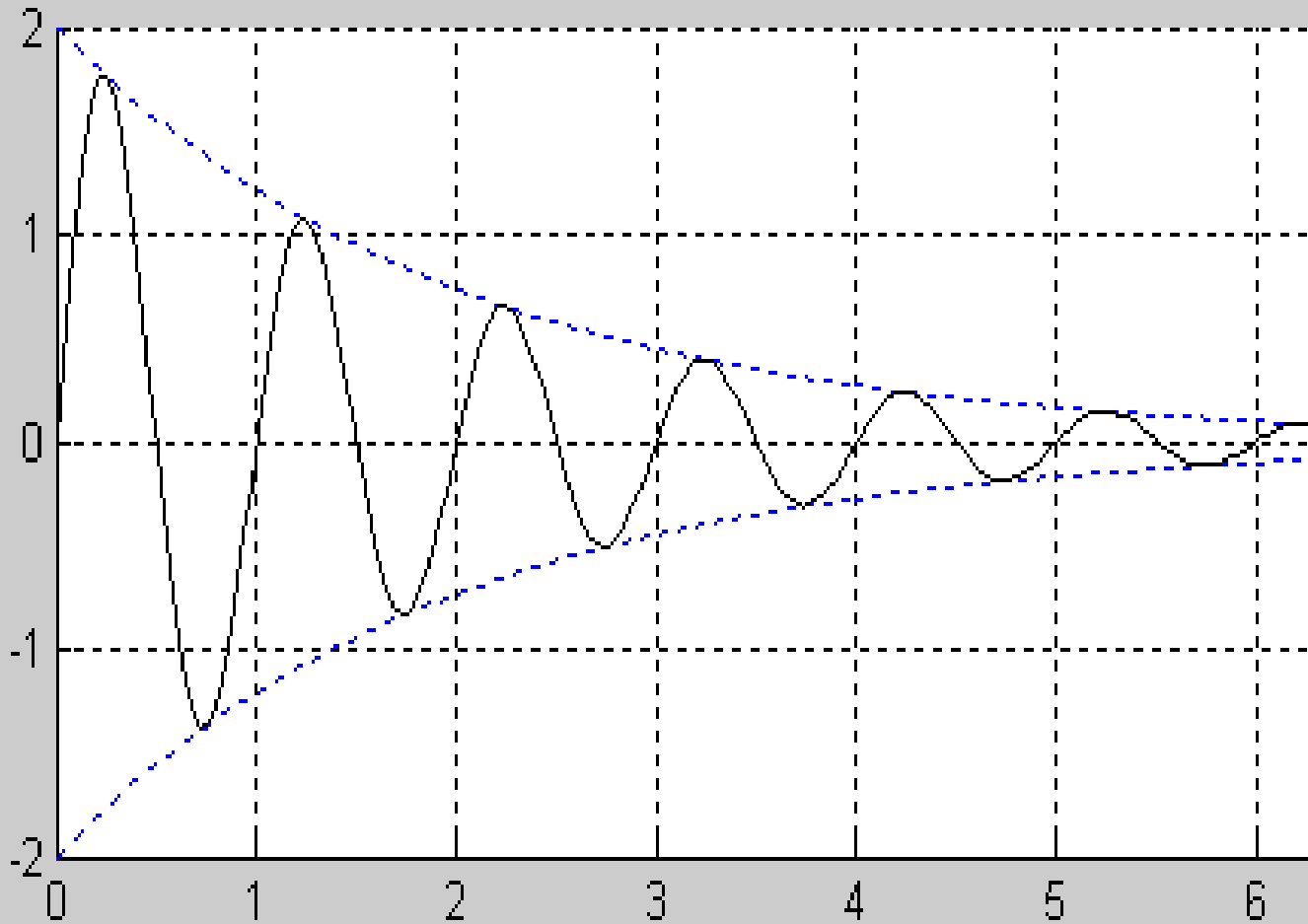


例6 用图形保持功能在同一坐标内绘制曲线

$y=2e^{-0.5x}\sin(2\pi x)$ 及其包络线，并加网格线。

程序如下：

```
x=(0:pi/100:2*pi)';  
y1=2*exp(-0.5*x)*[1,-1];y2=2*exp(-0.5*x).*sin(2*pi*x);  
plot(x,y1,'b:');  
axis([0,2*pi,-2,2]);           %设置坐标  
hold on;                       %设置图形保持状态  
plot(x,y2,'k');  
grid on;                       %加网格线  
box off;                       %不加坐标边框  
hold off;                      %关闭图形保持
```



2、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**

close 删除当前**figure**, 相当于**close(gcf)**

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

close name 删除指定名称的**figure**

close all 删除所有句柄没有隐藏的**figure**

close all hidden 删除所有**figure**, 包括句柄隐藏的。

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**

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。

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。

clf(fig) 或 **clf(fig, 'reset')** 清除由句柄为 **fig** 的图形窗口中的内容。

figure_handle = clf(...) 返回图形窗口的句柄,

图形窗口的分割 **subplot()**

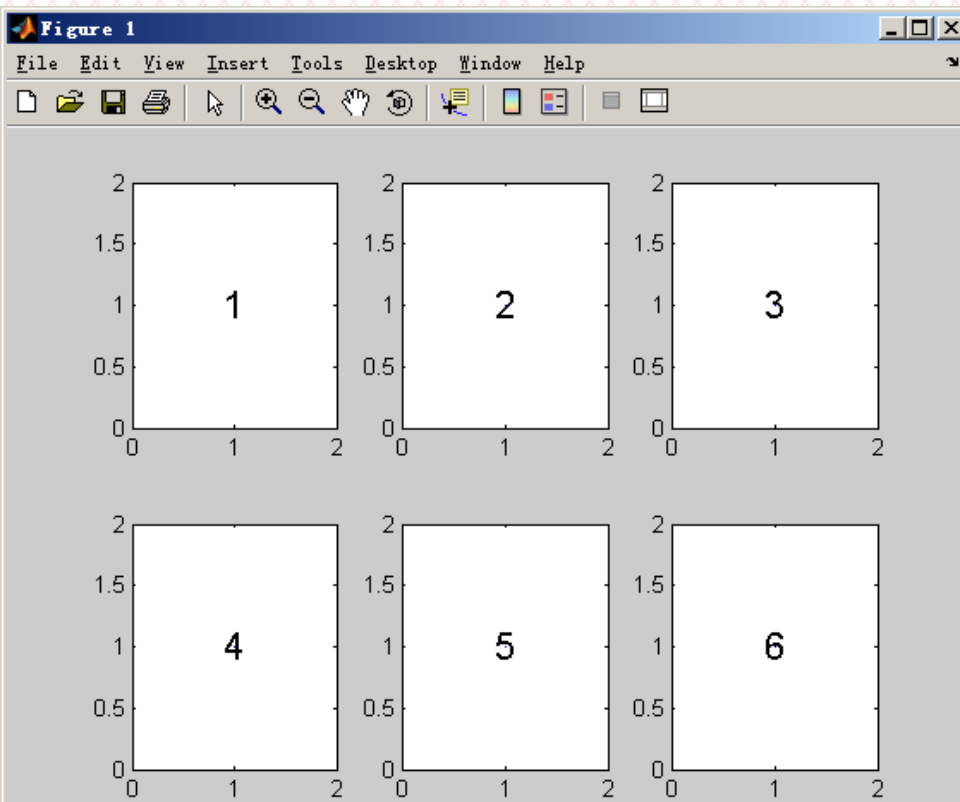
subplot函数的调用格式为:

subplot(m,n,p)

——把一个画面分割成 $m*n$ 个矩阵块区域，**p**代表当前的区域号，在每个区域中分别画一个图

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。

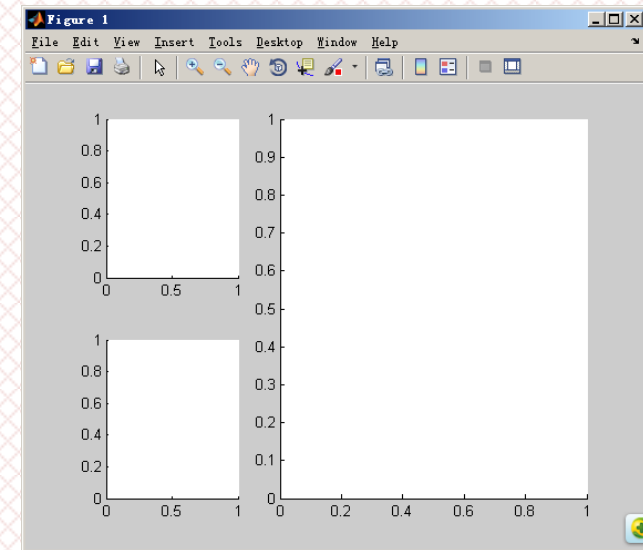
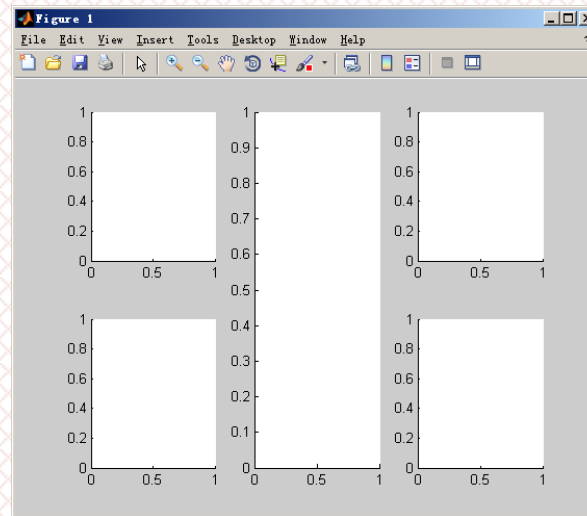
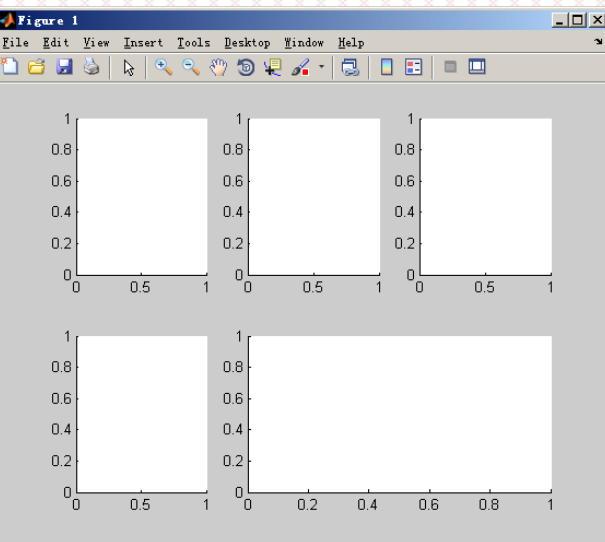
subplot(m,n,P)

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])
```

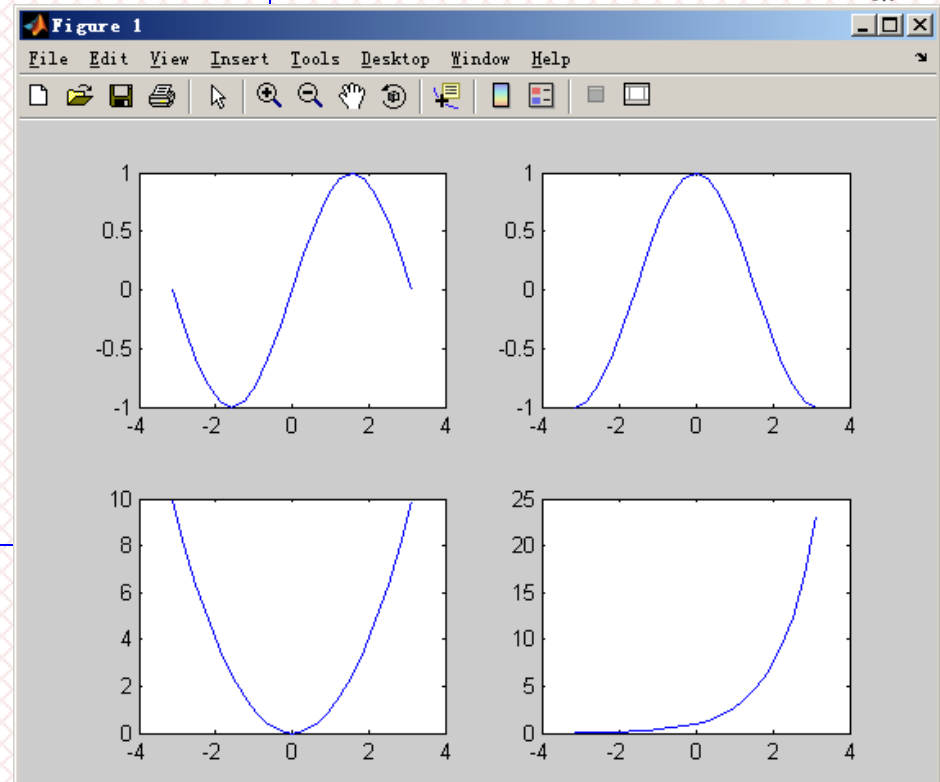
```
subplot(2,3,1)
subplot(2,3,4)
subplot(2,3,[2,3,5,6])
```



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。

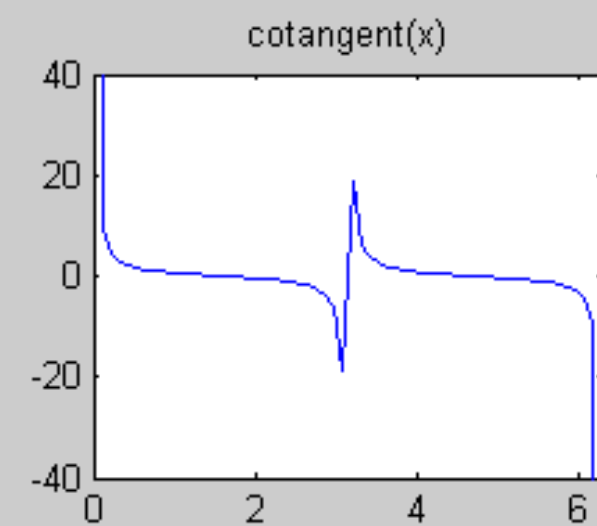
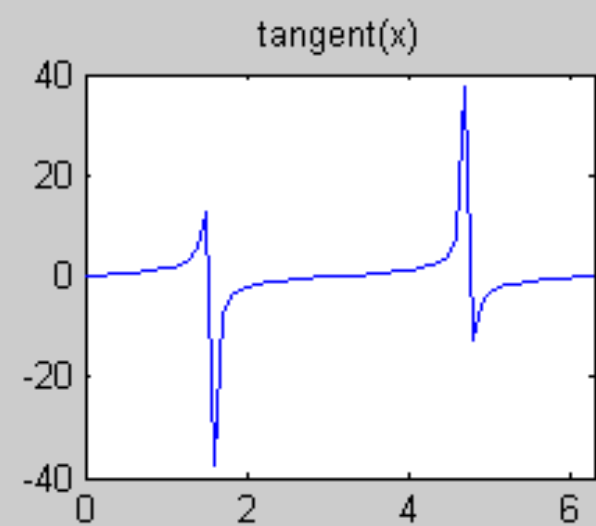
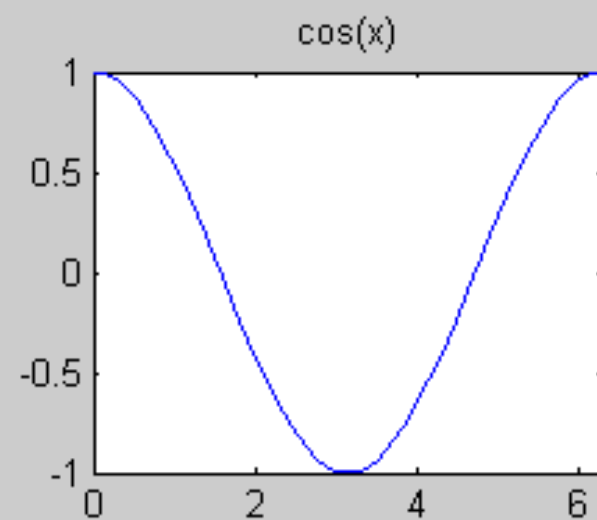
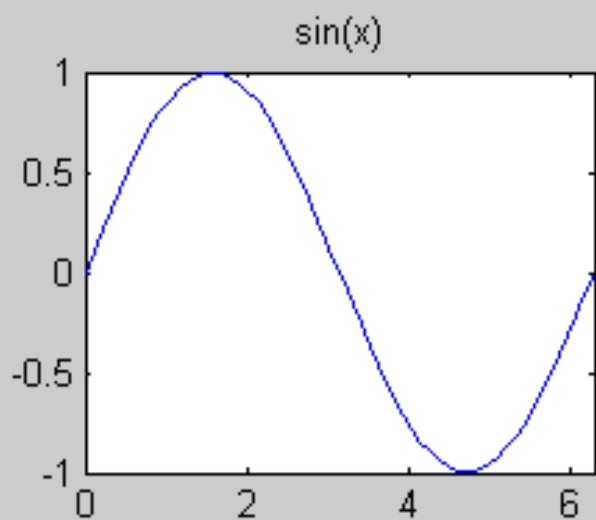


被激活的绘图区域在用户输入另一个subplot或者figure命令之前会一直保持被激活状态。

例7 在一个图形窗口中以子图形式同时绘制正弦、余弦、正切、余切曲线。

程序如下：

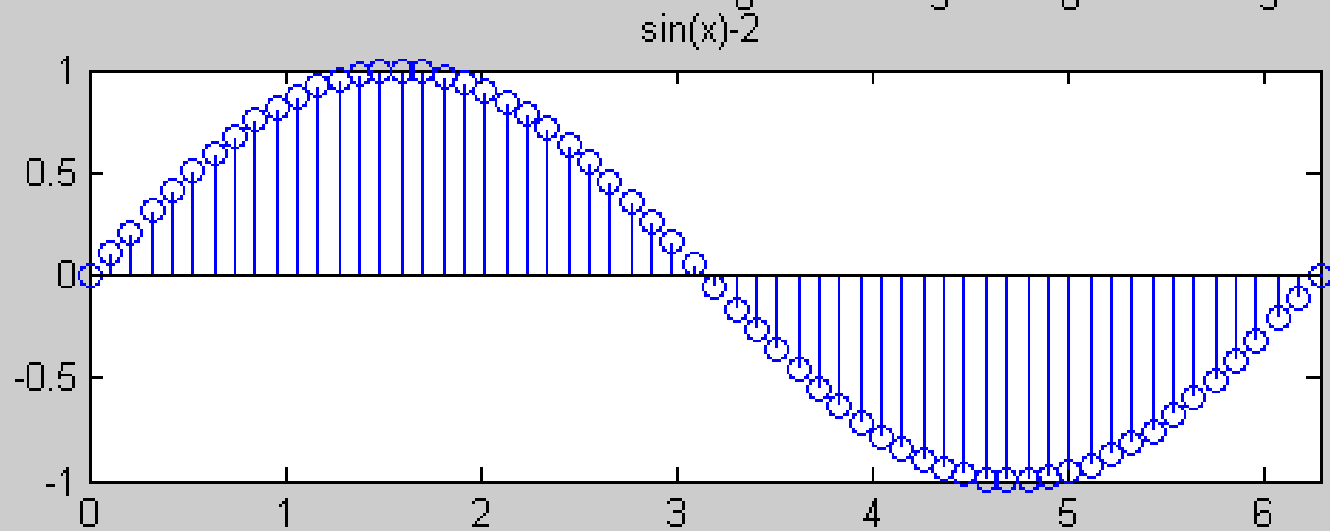
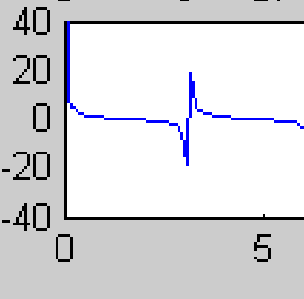
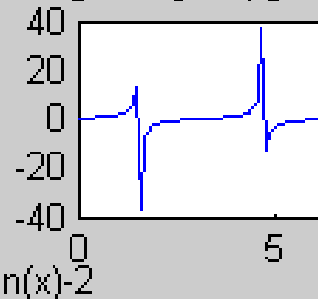
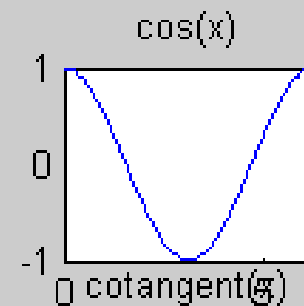
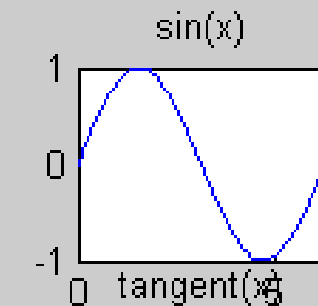
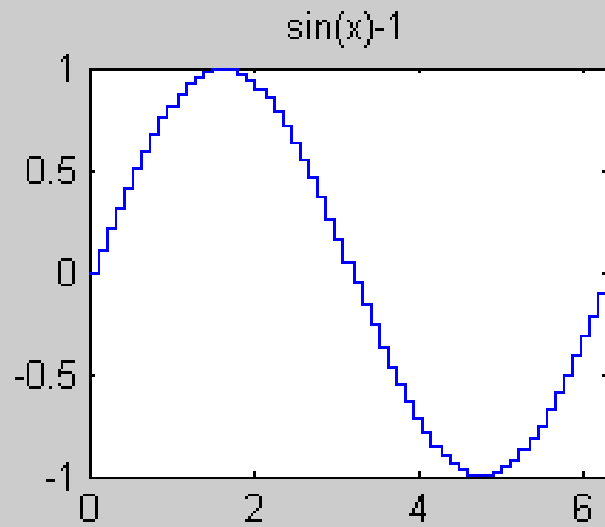
```
x=linspace(0,2*pi,60);  
y=sin(x);z=cos(x);  
t=sin(x)./(cos(x)+eps); ct=cos(x)./(sin(x)+eps);  
subplot(2,2,1);  
plot(x,y);title('sin(x)');axis ([0,2*pi,-1,1]);  
subplot(2,2,2);  
plot(x,z);title('cos(x)');axis ([0,2*pi, -1,1]);  
subplot(2,2,3);  
plot(x,t);title('tangent(x)');axis ([0,2*pi,-40,40]);  
subplot(2,2,4);  
plot(x,ct);title('cotangent(x)');axis ([0,2*pi,-40,40]);
```



对图形窗口灵活分割，请看下面的程序。

```
x=linspace(0,2*pi,60);  
y=sin(x);z=cos(x);  
t=sin(x)./(cos(x)+eps); ct=cos(x)./(sin(x)+eps);  
subplot(2,2,1);      %选择2×2个区中的1号区  
stairs(x,y);title('sin(x)-1');axis ([0,2*pi,-1,1]);  
subplot(2,1,2);      %选择2×1个区中的2号区  
stem(x,y);title('sin(x)-2');axis ([0,2*pi,-1,1]);  
subplot(4,4,3);      %选择4×4个区中的3号区  
plot(x,y);title('sin(x)');axis ([0,2*pi,-1,1]);  
subplot(4,4,4);      %选择4×4个区中的4号区  
plot(x,z);title('cos(x)');axis ([0,2*pi,-1,1]);  
subplot(4,4,7);      %选择4×4个区中的7号区  
plot(x,t);title('tangent(x)');axis ([0,2*pi,-40,40]);  
subplot(4,4,8);      %选择4×4个区中的8号区  
plot(x,ct);title('cotangent(x)');axis ([0,2*pi,-40,40]);
```



3. 绘制二维图形的其它函数

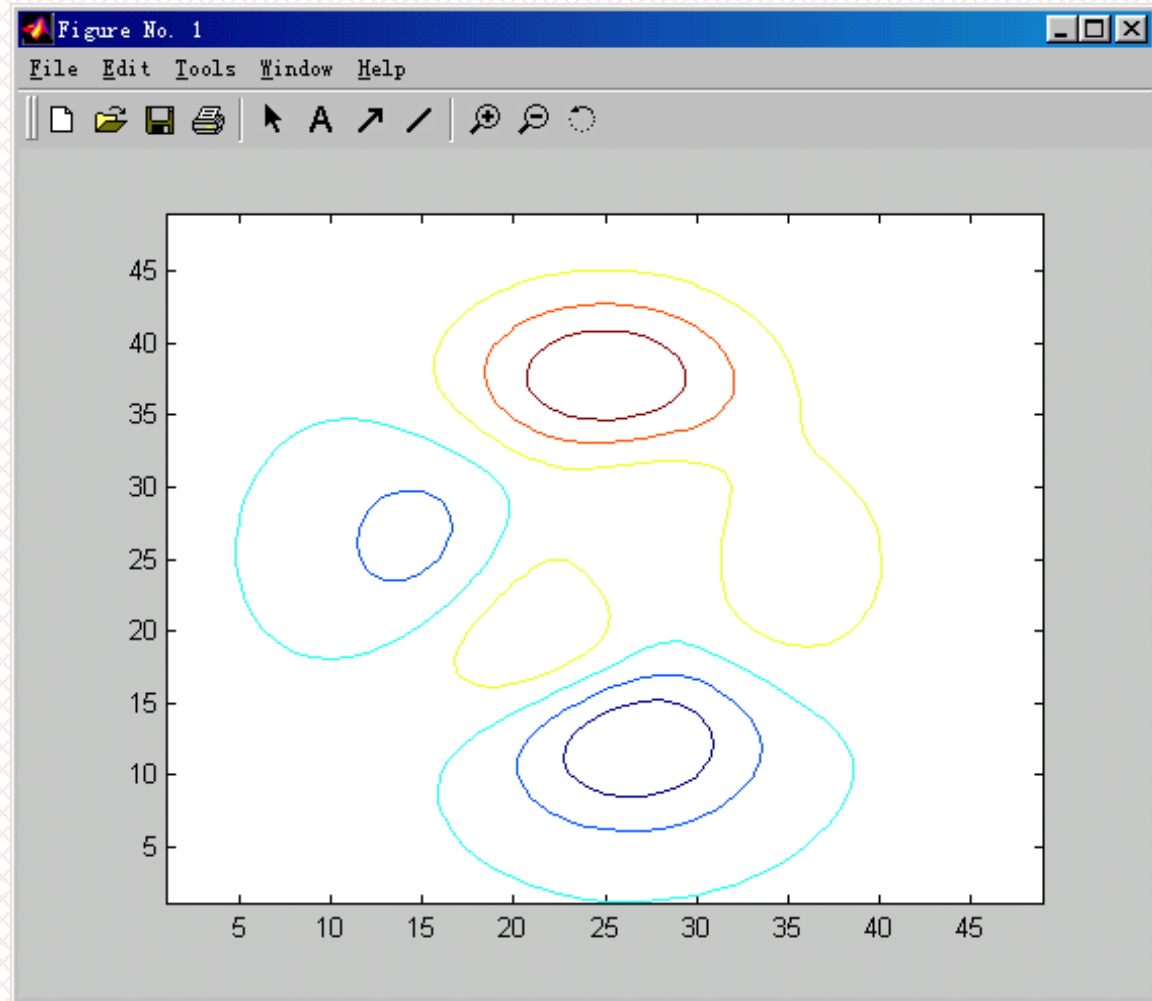


函数名	功能	函数名	功能
area	填充面积图	pie	圆饼图
bar	条形图	plotmatrix	绘矩阵点图
barh	水平柱图	ribbon	以三维带形式画二维线
comet	慧星形轨线	stem	火柴杆图
errorbar	误差条形图	stairs	台阶图
gplot	以图论方式绘图	contour	等高线图
feather	箭头图	contourf	填充等高线图
fill	填充二维多边形	clabel	等高线图仰角标签
pcolor	伪色图	rose	扇形统计图
quiver	场图	hist	直方统计图
voronoi	Voronoi图	pareto	Pareto图表
polar	极坐标图		

例：在平面上绘制**peaks**矩阵的等值线。

```
z=peaks;  
contour(z,6);
```

MATLAB提供了一个**peaks**函数，可产生一个凹凸有致的曲面，包含了三个局部极大点及三个局部极小点



其它形式的线性直角坐标图

在线性直角坐标系中，其他形式的图形有条形图、阶梯图、杆图和填充图等，所采用的函数分别是：

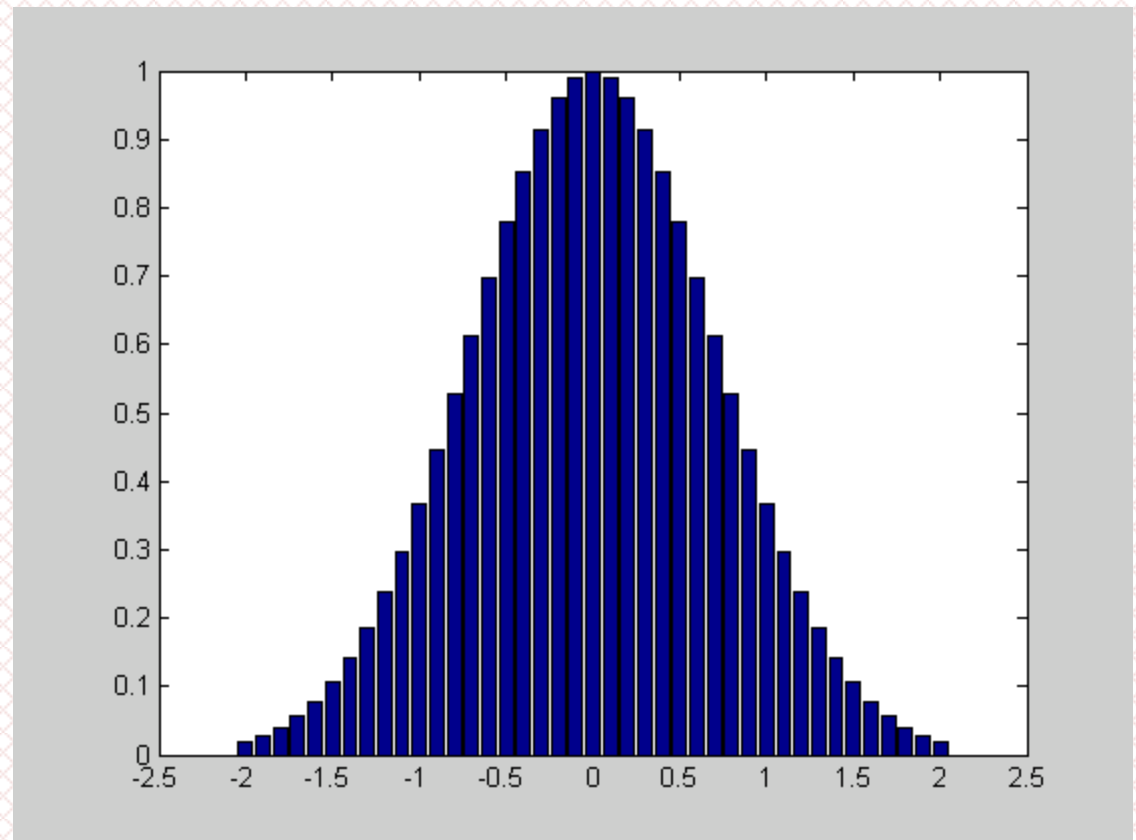
bar(x,y,选项) —— 条形图

stairs(x,y,选项) —— 阶梯图

stem(x,y,选项)

fill(x1,y1,选项1,x2,y2,选项2,...)

例：画一个钟形曲线
bar(randn(1,1000))
x=-2:0.1:2;
y=exp(-x.*x);
bar(x,y)



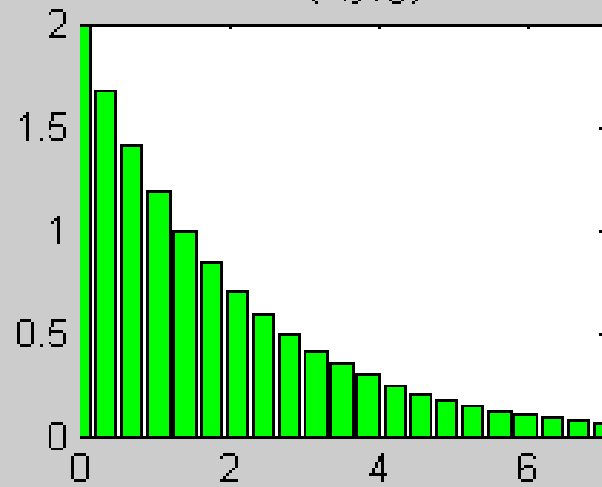


例8 分别以条形图、填充图、阶梯图和杆图形式绘制曲线 $y=2e^{-0.5x}$ 。

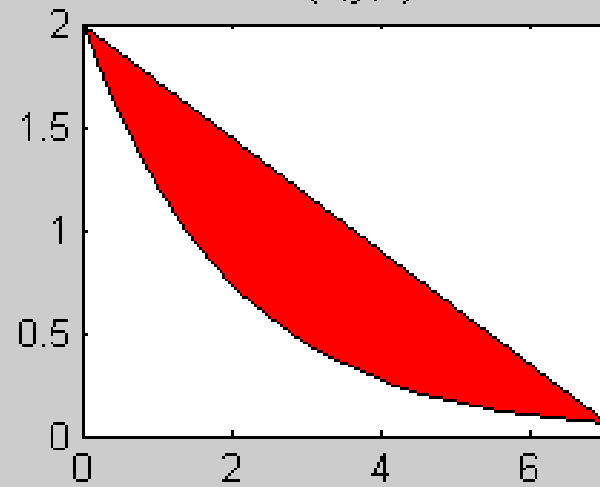
程序如下：

```
x=0:0.35:7;  
y=2*exp(-0.5*x);  
subplot(2,2,1);bar(x,y,'g');  
title('bar(x,y,"g")');axis([0,7,0,2]);  
subplot(2,2,2);fill(x,y,'r');  
title('fill(x,y,"r")');axis([0,7,0,2]);  
subplot(2,2,3);stairs(x,y,'b');  
title('stairs(x,y,"b")');axis([0,7,0,2]);  
subplot(2,2,4);stem(x,y,'k');  
title('stem(x,y,"k")');axis([0,7,0,2]);
```

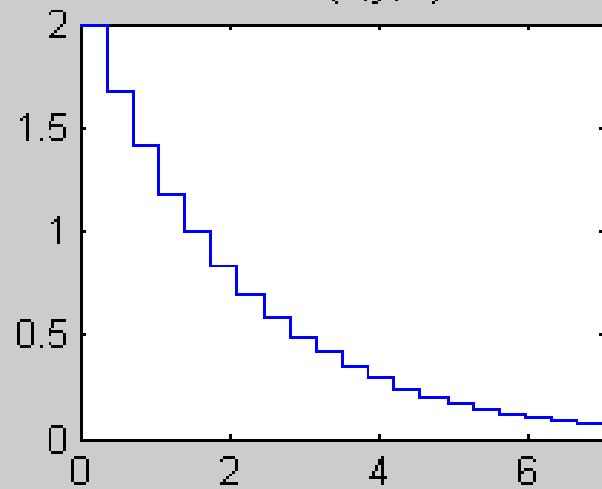

$\text{bar}(x,y,'g')$



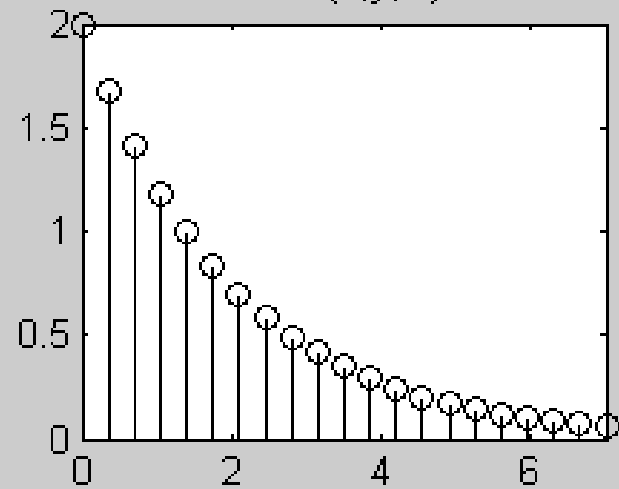
$\text{fill}(x,y,'r')$



$\text{stairs}(x,y,'b')$



$\text{stem}(x,y,'k')$



例9 bar()的用法举例

```
Y = round(rand(5,3)*10);
```

```
subplot(2,2,1)
```

```
bar(Y,'group')
```

```
title 'Group'
```

```
subplot(2,2,2)
```

```
bar(Y,'stack')
```

```
title 'Stack'
```

```
subplot(2,2,3)
```

```
barh(Y,'stack')
```

```
title 'Stack'
```

```
subplot(2,2,4)
```

```
bar(Y,1.5)
```

```
title 'Width = 1.5'
```

极坐标图

polar函数用来绘制极坐标图，其调用格式为：

polar(theta,rho,选项)

其中**theta**为极坐标极角，**rho**为极坐标矢径，选项的内容与**plot**函数相似。

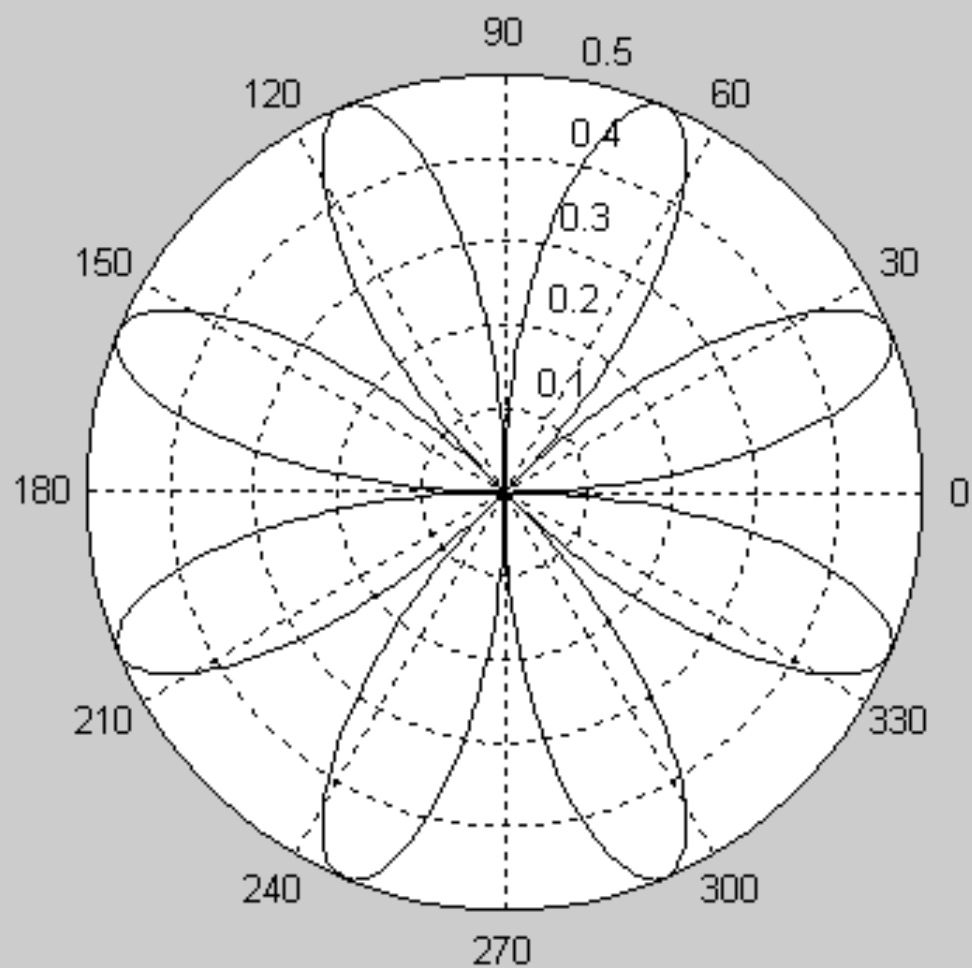
例：绘制 $\rho=\sin(2\theta)\cos(2\theta)$ 的极坐标图。

程序如下：

```
theta=0:0.01:2*pi;
```

```
rho=sin(2*theta).*cos(2*theta);
```

```
polar(theta,rho,'k');
```



其它形式的图形

MATLAB提供的绘图函数还有很多，例如，用来表示各元素占总和的百分比的饼图（用**pie**函数）、复数的相量图（用**compass**函数绘制从原点向外发射的向量图）等等。

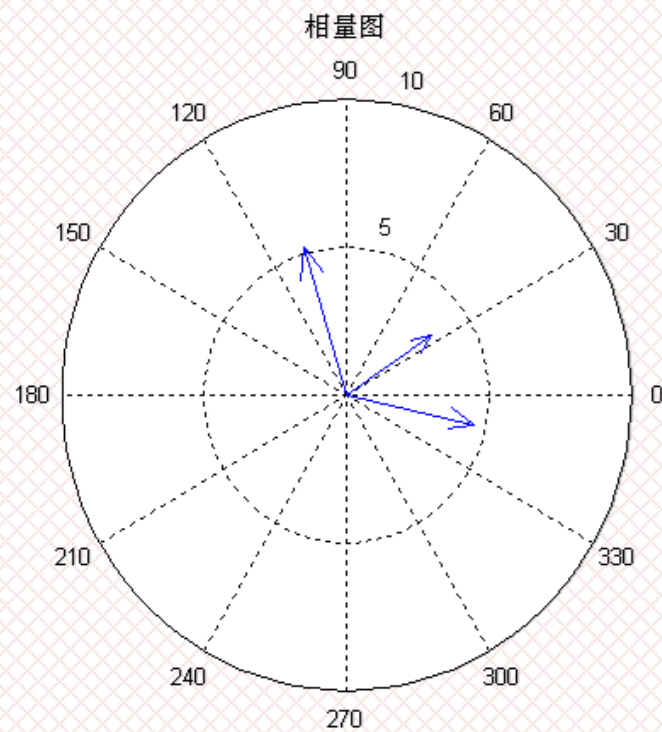
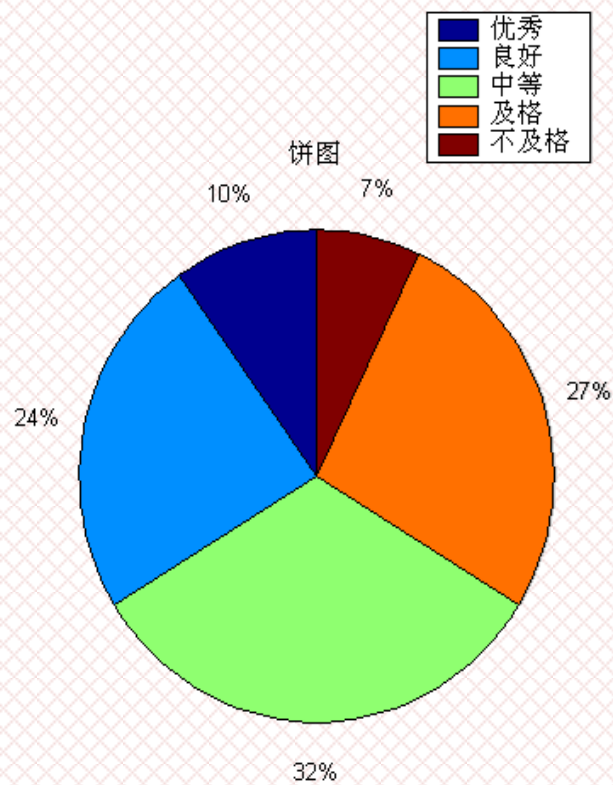
例12 绘制图形:

(1)某次考试优秀、良好、中等、及格、不及格的人数分别为: 7,17,23,19,5, 试用饼图作成绩统计分析。

(2)绘制复数的相量图: $3+2i$ 、 $4.5-i$ 和 $-1.5+5i$ 。

程序如下:

```
subplot(1,2,1);  
pie([7,17,23,19,5]); legend('优秀','良好','中等','及格','不及格');  
title('饼图');  
subplot(1,2,2);  
compass([3+2i,4.5-i,-1.5+5i]);title('相量图');
```



练习题

1. 已知椭圆的长、短轴 $a = 4, b = 2$, 用“小红点线”画如图 p5-1 所示的椭圆 $\begin{cases} x = a \cos t \\ y = b \sin t \end{cases}$ 。

(提示: 参量 t ; 点的大小; axis equal) ↵

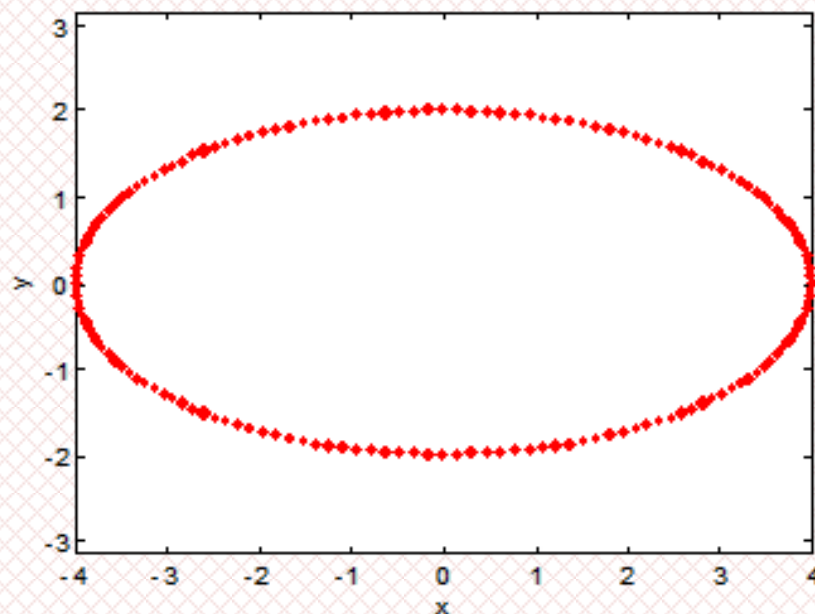


图 p5-1 ↵

练习题

2. 根据表达式 $\rho = 1 - \cos \theta$ 绘制如图 p5-2 的心脏线。(提示: polar; 注意 title 中特殊字符; 线宽; axis square。可以用 plot 试试。)

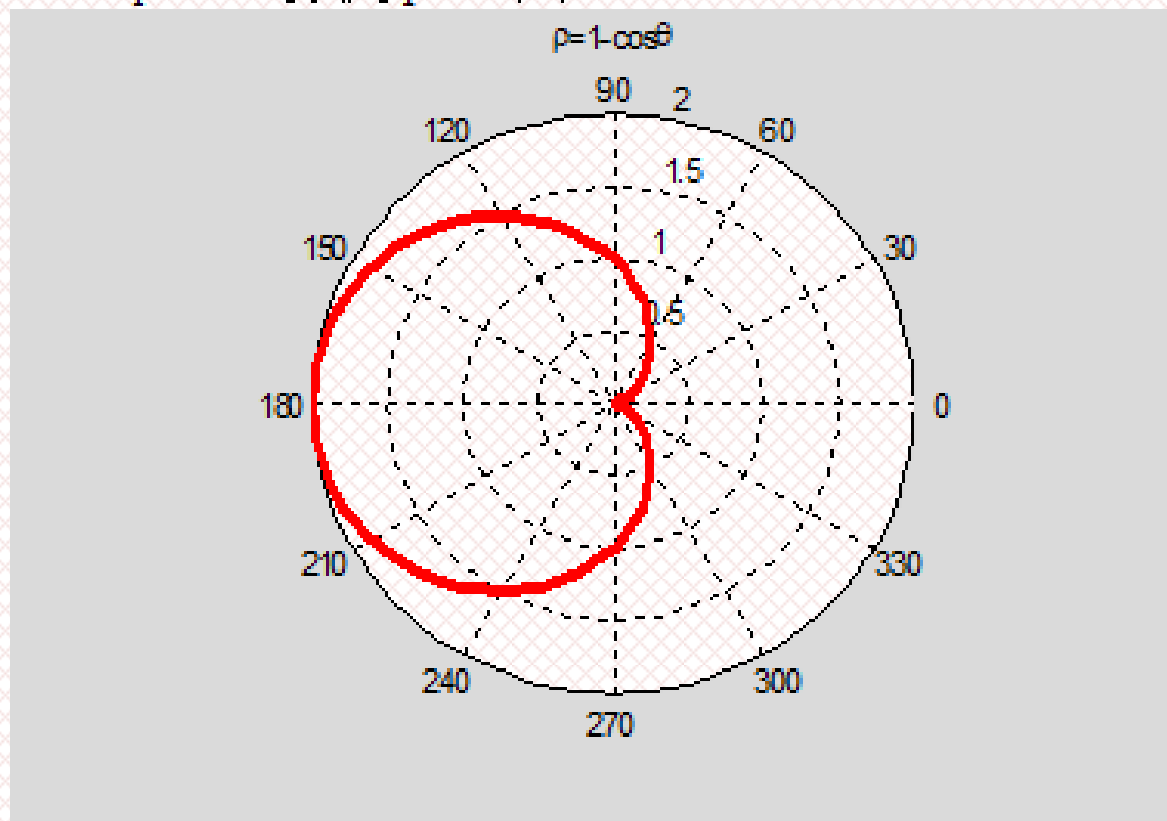


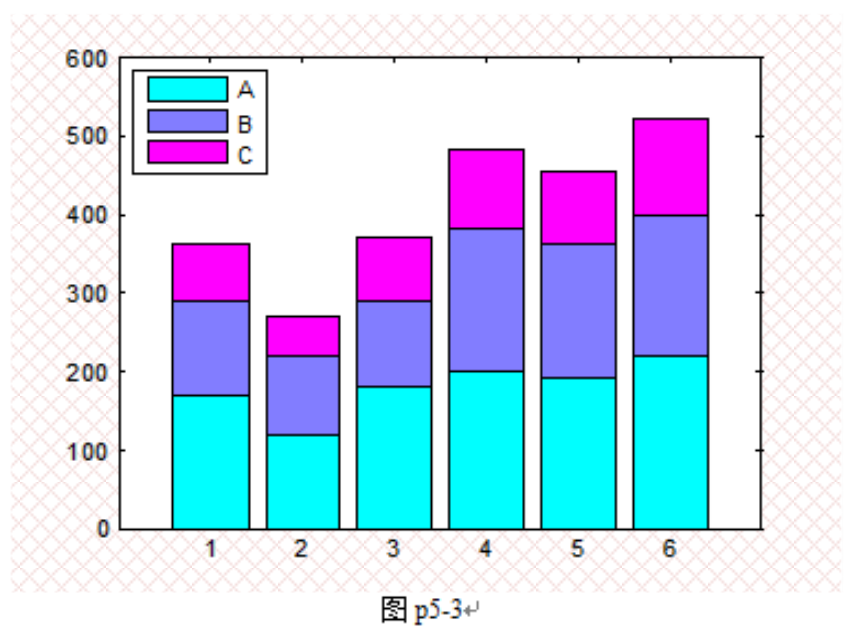
图 p5-2

练习题

3. A,B,C三个城市上半年每个月的国民生产总值表 p5.1。试画出如图 p5-3 所示的三城市上半年每月生产总值的累计直方图。（提示：`bar(x,Y,'style');` `colormap(cool); legend。`）

表 p5.1 各城市生产总值数据（单位：亿元）

城市	1 月	2 月	3 月	4 月	5 月	6 月
A	170	120	180	200	190	220
B	120	100	110	180	170	180
C	70	50	80	100	95	120



练习题

4、编写程序，绘制下列函数在区间[-6, 6]中的图形。

$$y(x) = \begin{cases} \sin x, & x \leq 0 \\ x, & 0 < x \leq 3 \\ -x + 6, & x > 3 \end{cases}$$

Q & A

- 有什么问题吗？

