

4.1 二维曲线

- plot函数
- fplot函数

1. plot函数

(1) plot函数的基本用法

`plot(x, y)`

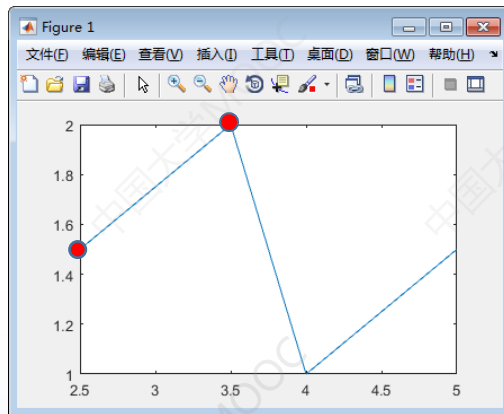
其中，x和y分别用于存储x坐标和y坐标数据。通常，x和y为长度相同的向量。

例1 绘制一条折线。

```
>> x=[2.5, 3.5, 4, 5];
```

```
>> y=[1.5, 2.0, 1, 1.5];
```

```
>> plot(x, y)
```



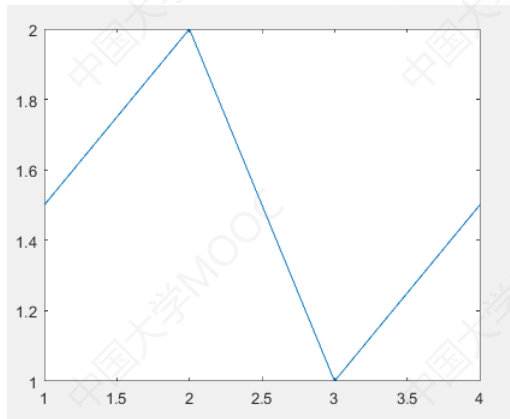
1. plot函数

(2) 最简单的plot函数调用格式

plot(x)

```
>> x=[1.5, 2, 1, 1.5];
```

```
>> plot(x)
```



1. plot函数

(2) 最简单的plot函数调用格式

`plot(x)`

当plot函数的参数x是复数向量时，则分别以该向量元素实部和虚部为横、纵坐标绘制出一条曲线。

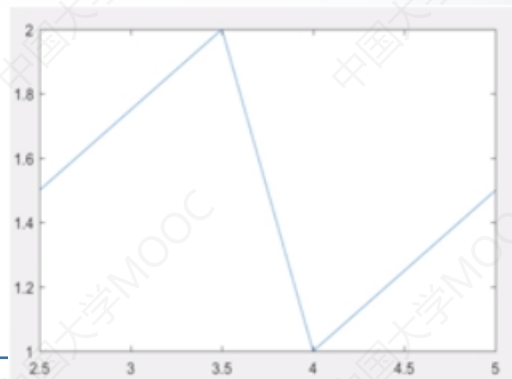
```
>> x=[2.5, 3.5, 4, 5];
```

```
>> y=[1.5, 2, 1, 1.5];
```

```
>> cx=x+y*i;
```

```
>> plot(cx)
```

复型变量也可以用complex函数构建
`cx = complex(x, y);`



1. plot函数

(3) plot(x, y)函数参数的变化形式

□ 当x是向量, y是矩阵时

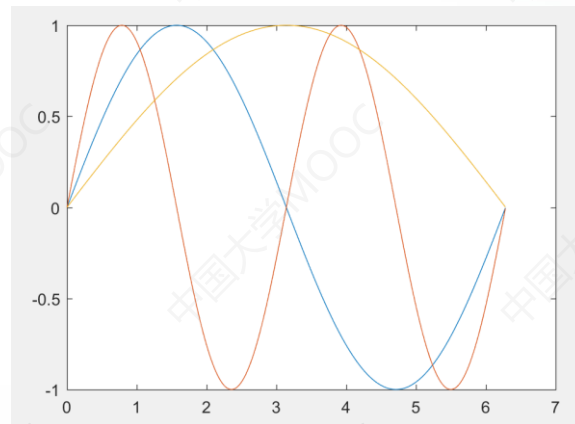
- 如果矩阵y的列数等于x的长度, 则以向量x为横坐标, 以y的每个行向量为纵坐标绘制曲线, 曲线的条数等于y的行数。
- 如果矩阵y的行数等于x的长度, 则以向量x为横坐标, 以y的每个列向量为纵坐标绘制曲线, 曲线的条数等于y的列数。

1. plot函数

(3) plot(x, y) 函数参数的变化形式

例2 绘制 $\sin(x)$ 、 $\sin(2x)$ 、 $\sin(x/2)$ 的函数曲线。

```
>> x=linspace(0,2*pi,100);  
>> y=[sin(x); sin(2*x); sin(0.5*x)];  
>> plot(x,y)
```



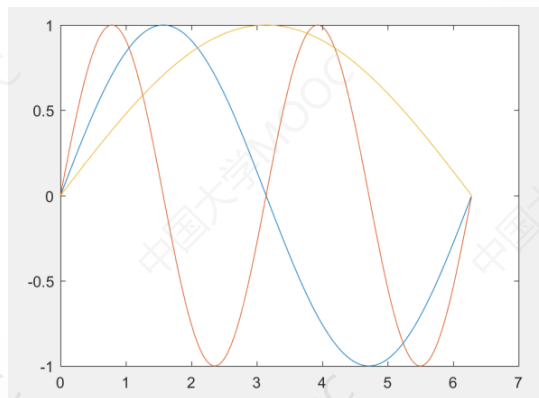
1. plot函数

(3) plot(x, y) 函数参数的变化形式

□ 当x、y是同型矩阵时

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。

```
>> t=0:0.01:2*pi;  
>> t1=t';  
>> x=[t1, t1, t1];  
>> y=[sin(t1), sin(2*t1), sin(0.5*t1)];  
>> plot(x,y)
```



1. plot函数

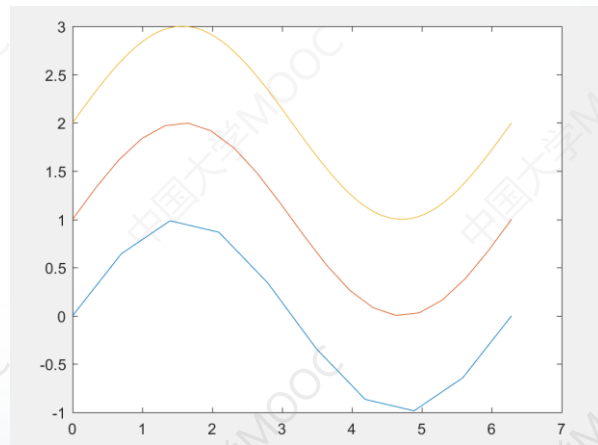
(4) 含多个输入参数的plot函数

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

其中，每一向量对构成一组数据点的横、纵坐标，绘制一条曲线。

例3 采用不同个数的数据点绘制正弦函数曲线，观察曲线形态。

```
>> t1=linspace(0, 2*pi, 10);  
>> t2=linspace(0, 2*pi, 20);  
>> t3=linspace(0, 2*pi, 100);  
>> plot(t1, sin(t1), t2, sin(t2)+1, ...  
t3, sin(t3)+2)
```



1. plot函数

(5) 含选项的plot函数

plot(x, y, 选项)

其中，选项用于指定曲线的线型、颜色和数据点标记。

线型

“—” : 实线
“:” : 虚线
“-.” : 点划线
“--” : 双划线

颜色

“r” : 红色
“g” : 绿色
“b” : 蓝色
“w” : 白色
“k” : 黑色

...

数据点标记

“*” : 星号
“o” : 圆圈
“s” : 方块
“p” : 五角星
“^” : 朝上三角符号

...



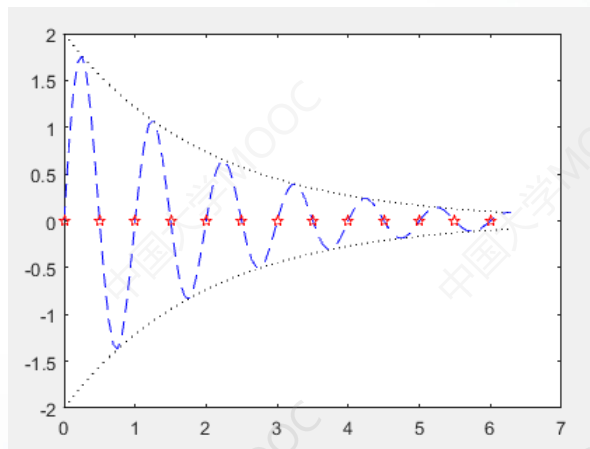
1. plot函数

(5) 含选项的plot函数

plot(x, y, 选项)

例4 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。

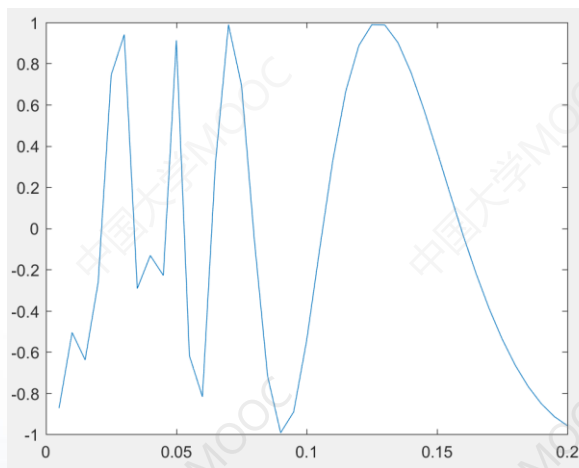
```
x=(0:pi/50:2*pi)';  
y1=2*exp(-0.5*x)*[1,-1];  
y2=2*exp(-0.5*x).*sin(2*pi*x);  
x1=0:0.5:6;  
y3=2*exp(-0.5*x1).*sin(2*pi*x1);  
plot(x,y1,'k:', x,y2,'b--', x1,y3,'rp')
```



使用plot函数绘图时，先要取得x、y坐标，然后再绘制曲线，x往往采取等间隔采样。在实际应用中，函数随着自变量的变化趋势未知，或者在不同区间函数频率特性差别大，此时使用plot函数绘制图形，如果自变量的采样间隔设置不合理，则无法反映函数的变化趋势。

例5 绘制函数 $\sin \frac{1}{x}$ 的图形。

```
>> x=0:0.005:0.2;  
>> y=sin(1./x);  
>> plot(x,y)
```



2. fplot函数

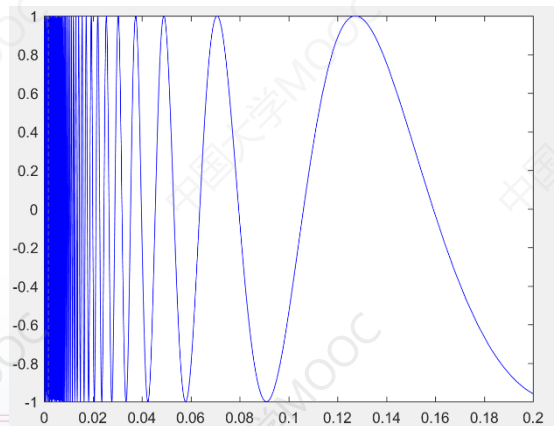
(1) fplot函数的基本用法

`fplot(f, lims, 选项)`

其中，`f`代表一个函数，通常采用函数句柄的形式。`lims`为`x`轴的取值范围，用二元向量`[xmin, xmax]`描述，默认值为`[-5, 5]`。选项定义与`plot`函数相同。

例6 采用`fplot`函数绘制函数 $\sin \frac{1}{x}$ 。

```
>> fplot(@(x) sin(1./x), [0, 0.2], 'b')
```



2. fplot函数

(2) 双输入函数参数的用法

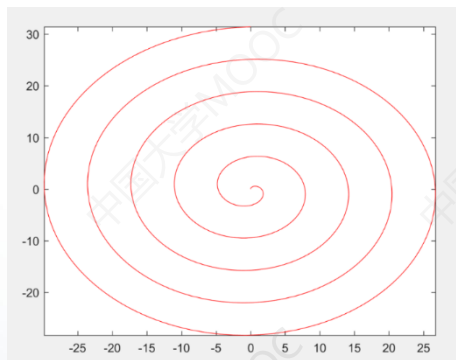
`fplot(funx, funy, tlims, 选项)`

其中，`funx`、`funy`代表函数，通常采用函数句柄的形式。`tlims`为参数函数`funx`和`funy`的自变量的取值范围，用二元向量`[tmin, tmax]`描述。

2. fplot函数

例7 已知螺旋线的参数方程 $\begin{cases} x = t \cdot \sin t \\ y = t \cdot \cos t \end{cases}$ ，绘制曲线。

```
>> fplot(@(t)t.*sin(t), @(t)t.*cos(t), [0, 10*pi], 'r')
```



4.2 绘制图形的辅助操作

- 给图形添加标注
- 坐标控制
- 图形保持
- 图形窗口的分割

1. 图形标注

- `title`(图形标题)
- `xlabel`(x轴说明)
- `ylabel`(y轴说明)
- `text`(x, y, 说明)
- `legend`(图例1, 图例2, ...)

1. 图形标注

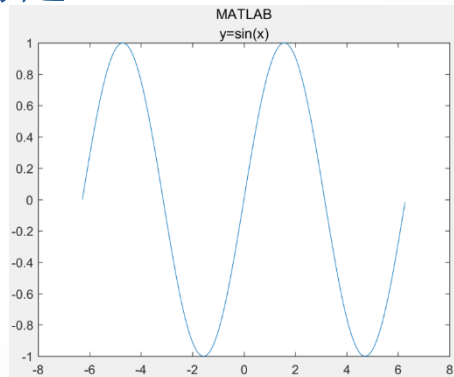
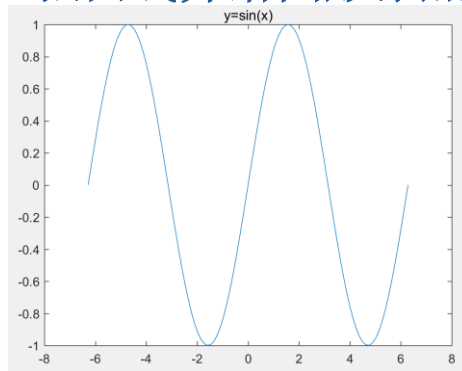
(1) title函数

①title函数的基本用法

title(图形标题)

例1 绘制 $[-2\pi, 2\pi]$ 区间的正弦曲线并给图形添加标题。

```
>> x=-2*pi:0.05:2*pi;  
>> y=sin(x);  
>> plot(x,y)  
>> title('y=sin(x)')
```



```
>> title({'MATLAB', 'y=sin(x)'})
```

1. 图形标注

(1) title函数

② 在图形标题中使用LaTeX格式控制符

受LaTeX格式控制的部分要用大括号括起来。

```
>> title('y=cos{\omega}t')
```

```
>> title('y=e^{\text{axt}}')
```

```
>> title('X_{1} {\geq} X_{2}')
```

```
>> title('{\bf y=cos{\omega}t + {\beta}}')
```

$y=\cos\omega t$

$y=e^{\text{axt}}$

$X_1 \geq X_2$

$y = \cos\omega t + \beta$

格式控制符

“\bf”：加粗

“\it”：斜体

“\rm”：正体

1. 图形标注

(1) title函数

③含属性设置的title函数

title(图形标题, 属性名, 属性值)

□ Color属性 : 用于设置图形标题文本的颜色。

```
>> title('y=cos{\omega}t', 'Color','r')
```

y=cos ω t

□ FontSize属性 : 用于设置标题文字的字号。

```
>> title('y=cos{\omega}t', 'FontSize', 24)
```

y=cos ω t

1. 图形标注

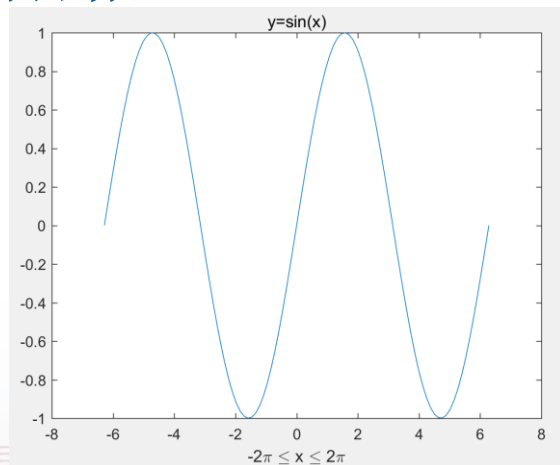
(2) xlabel函数和ylabel函数

xlabel(x轴说明)

ylabel(y轴说明)

绘制 $[-2\pi, 2\pi]$ 区间的正弦曲线并给x轴添加标签。

```
>> x=-2*pi:0.05:2*pi;  
>> y=sin(x);  
>> plot(x,y)  
>> title('y=sin(x)')  
>> xlabel('-2\pi \leq x \leq 2\pi')
```



1. 图形标注

(3) text函数和gtext函数

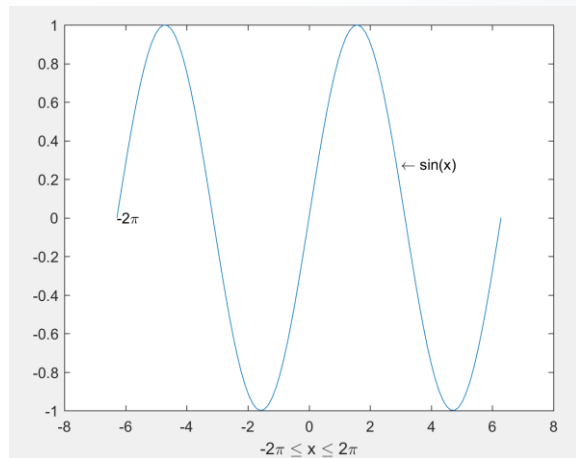
text(x, y, 说明)

gtext(说明)

在前面的图形中添加文字说明。

```
>> text(-2*pi, 0, '-2{\pi}')
```

```
>> text(3, 0.28, '\leftarrow sin(x)')
```



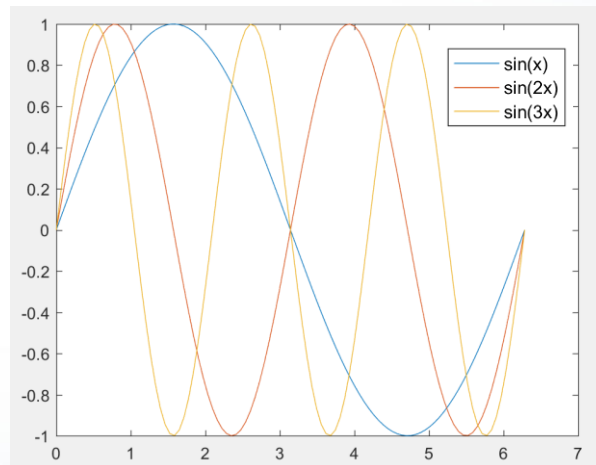
1. 图形标注

(4) legend函数

legend(图例1, 图例2, ...)

例2 绘制不同频率的正弦曲线并用图例标注曲线。

```
>> x = linspace(0, 2*pi, 100);  
>> plot(x, [sin(x); sin(2*x); sin(3*x)])  
>> legend('sin(x)', 'sin(2x)', 'sin(3x)')
```



2. 坐标控制

□ axis函数

□ grid函数

□ box函数

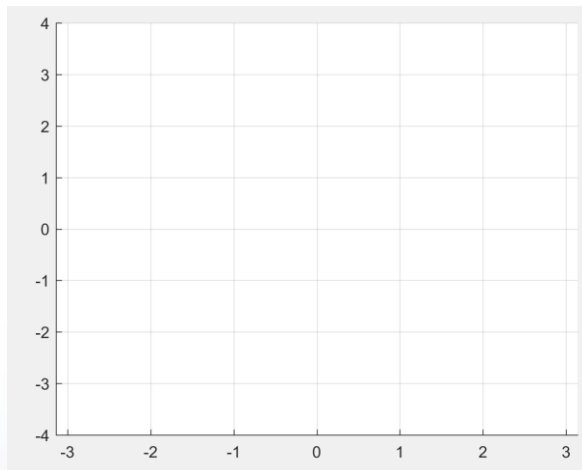
2. 坐标控制

(1) axis函数

□ axis函数的基本用法

`axis([xmin, xmax, ymin, ymax, zmin, zmax])`

`>> axis([-pi, pi, -4, 4])`



2. 坐标控制

(1) axis函数

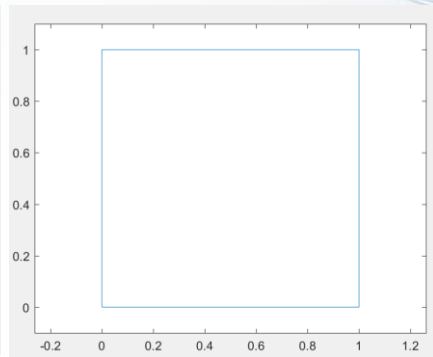
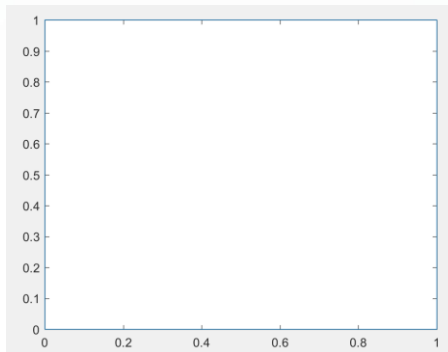
□ axis的其他用法

- axis equal: 纵、横坐标轴采用等长刻度
- axis square: 产生正方形坐标系(默认为矩形)
- axis auto: 使用默认设置
- axis off: 取消坐标轴
- axis on: 显示坐标轴



2. 坐标控制

```
>> x = [0, 1, 1, 0, 0];  
>> y = [0, 0, 1, 1, 0];  
>> plot(x,y)  
>> axis([-0.1, 1.1, -0.1, 1.1])  
>> axis equal;
```





2. 坐标控制

(2) 给坐标系加网格、边框

`grid on`

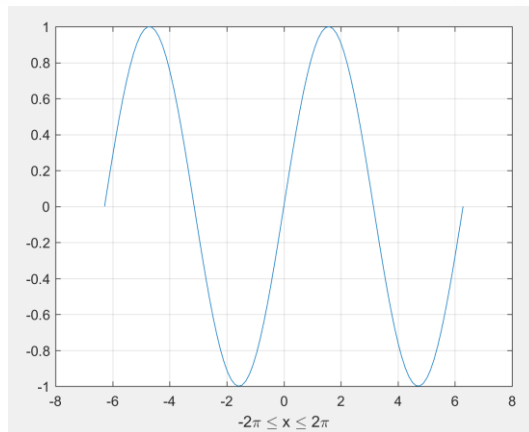
`grid off`

`grid`

`box on`

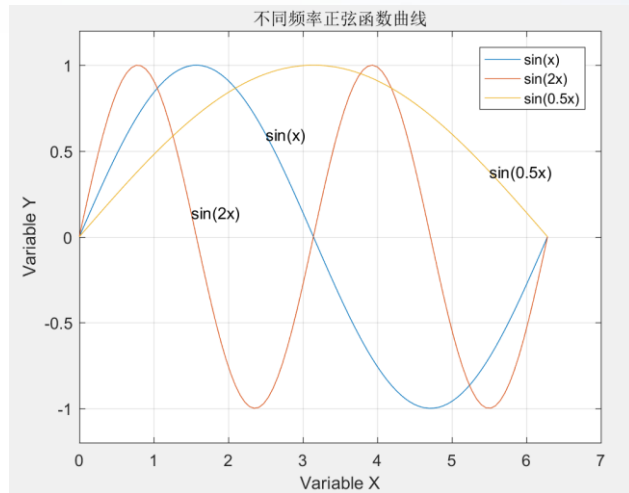
`box off`

`box`



例3 绘制 $\sin x$ 、 $\sin(2x)$ 、 $\sin(x/2)$ 的函数曲线并添加图形标注。

```
x=linspace(0, 2*pi, 100);  
y=[sin(x); sin(2*x); sin(0.5*x)];  
plot(x,y)  
axis([0 7 -1.2, 1.2])  
title('不同频率正弦函数曲线');  
xlabel('Variable X'); ylabel('Variable Y');  
text(2.5, sin(2.5), 'sin(x)');  
text(1.5, sin(2*1.5), 'sin(2x)');  
text(5.5, sin(0.5*5.5), 'sin(0.5x)');  
legend('sin(x)', 'sin(2x)', 'sin(0.5x)')  
grid on
```



3. 图形保持

hold on

hold off

hold

例4 用图形保持功能绘制两个同心圆。

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

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

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

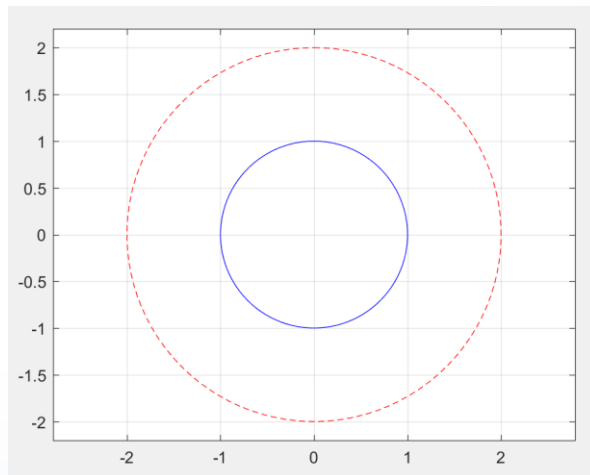
```
hold on;
```

```
plot(2*x, 2*y, 'r--')
```

```
grid on
```

```
axis([-2.2 2.2 -2.2 2.2])
```

```
axis equal
```



4. 图形窗口的分割

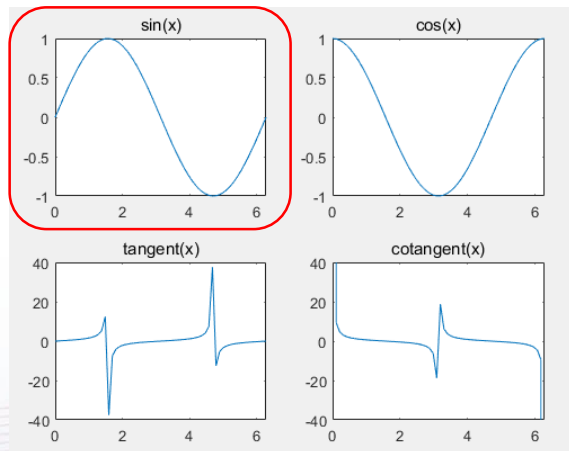
❑ 子图：同一图形窗口中的不同坐标系下的图形称为子图。

❑ subplot函数

`subplot(m, n, p)`

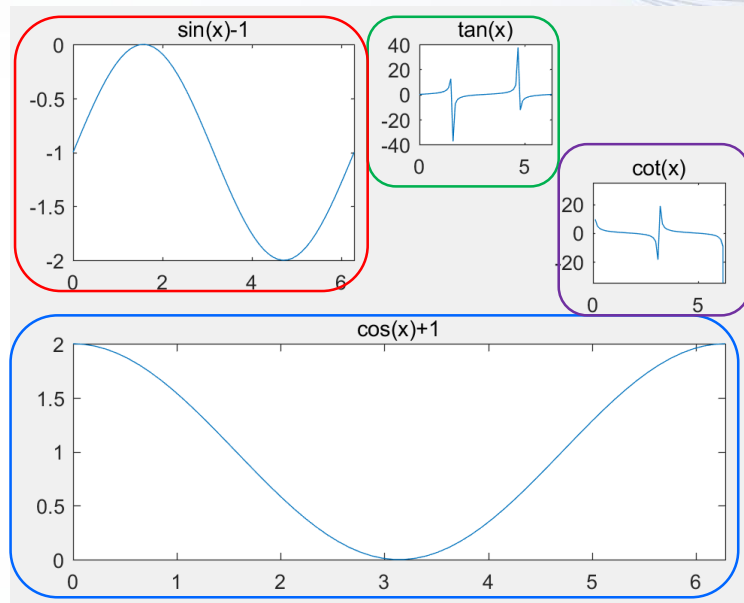
其中，`m`和`n`指定将图形窗口分成`m`×`n`个绘图区，`p`指定当前活动区。

```
>> subplot(2, 2, 1);  
>> x=linspace(0, 2*pi, 60);  
>> y=sin(x);  
>> plot(x, y);  
>> title('sin(x)');  
>> axis([0, 2*pi, -1, 1]);
```



4. 图形窗口的分割

```
x=linspace(0,2*pi,60);  
subplot(2,2,1)  
plot(x,sin(x)-1);  
title('sin(x)-1');axis([0,2*pi,-2,0])  
subplot(2,1,2)  
plot(x,cos(x)+1);  
title('cos(x)+1');axis([0,2*pi,0,2])  
subplot(4,4,3)  
plot(x,tan(x));  
title('tan(x)');axis([0,2*pi,-40,40])  
subplot(4,4,8)  
plot(x,cot(x));  
title('cot(x)');axis([0,2*pi,-35,35])
```



4.3 其他形式的二维曲线

- 其他坐标系下的二维曲线图
- 统计图
- 矢量图形

1. 其他坐标系下的二维曲线图

(1) 对数坐标图

`semilogx(x1, y1, 选项1, x2, y2, 选项2, ...)`

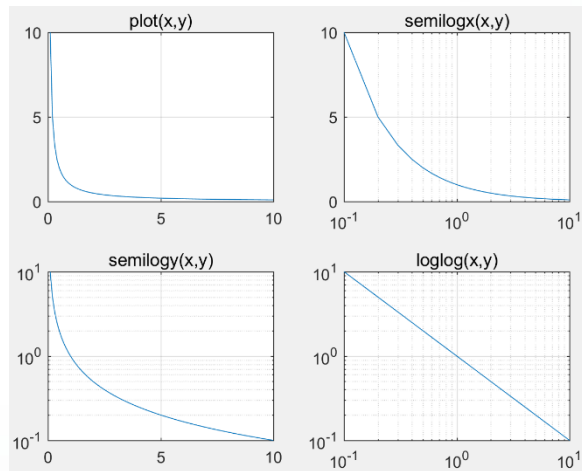
`semilogy(x1, y1, 选项1, x2, y2, 选项2, ...)`

`loglog(x1, y1, 选项1, x2, y2, 选项2, ...)`

其中，`semilogx`函数x轴为常用对数刻度，y轴为线性刻度；`semilogy`函数x轴为线性刻度，y轴为常用对数刻度；`loglog`函数x轴和y轴均采用常用对数刻度。

例1 绘制 $\frac{1}{x}$ 的直角线性坐标图和三种对数坐标图。

```
x=0:0.1:10;  
y=1./x;  
subplot(2,2,1)  
plot(x,y)  
title('plot(x,y)');grid on  
subplot(2,2,2)  
semilogx(x,y)  
title('semilogx(x,y)');grid on  
subplot(2,2,3)  
semilogy(x,y)  
title('semilogy(x,y)');grid on  
subplot(2,2,4)  
loglog(x,y)  
title('loglog(x,y)');grid on
```



1. 其他坐标系下的二维曲线图

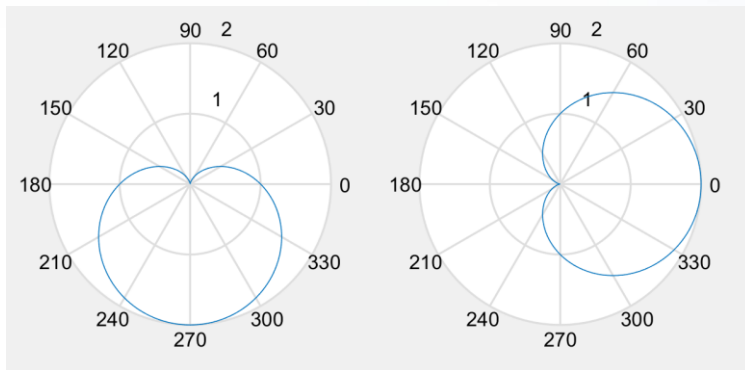
(2) 极坐标图

`polar(theta, rho, 选项)`

其中， θ 为极角， ρ 为极径，选项的内容与`plot`函数相同。

例2 按极坐标方程 $\rho = 1 - \sin\theta$ 绘制心形曲线。

```
t = 0:pi/100:2*pi;  
r = 1-sin(t);  
subplot(1,2,1)  
polar(t,r)  
subplot(1,2,2)  
t1 = t-pi/2;  
r1 = 1-sin(t1);  
polar(t,r1)
```





2. 统计图

□ 条形图

□ 直方图

□ 饼图

□ 散点图

2. 统计图

(1) 条形类图形

① 条形图

- `bar`函数：绘制垂直条形图。
- `barh`函数：绘制水平条形图。

2. 统计图

(1) 条形类图形

① 条形图

□ bar函数

`bar(y, style)`

其中，参数y是数据，选项style用于指定分组排列模式。

“grouped”：簇状分组
“stacked”：堆积分组

例3 绘制分组条形图。

```
y=[1, 2, 3, 4, 5; 1, 2, 1, 2, 1; 5, 4, 3, 2, 1];
```

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

```
bar(y)
```

```
title('Group')
```

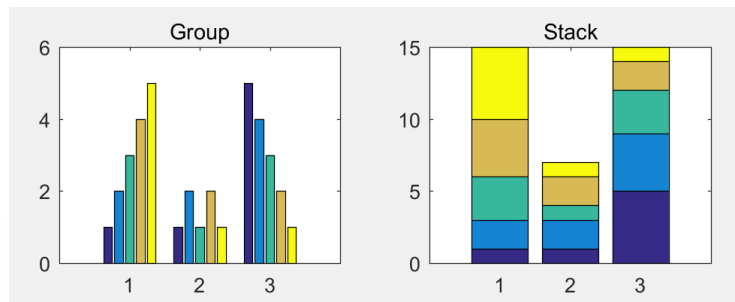
```
subplot(1,2,2)
```

```
bar(y, 'stacked')
```

```
title('Stack')
```

Y矩阵

1	2	3	4	5
1	2	1	2	1
5	4	3	2	1



2. 统计图

(1) 条形类图形

① 条形图

□ bar函数

`bar(x, y, style)`

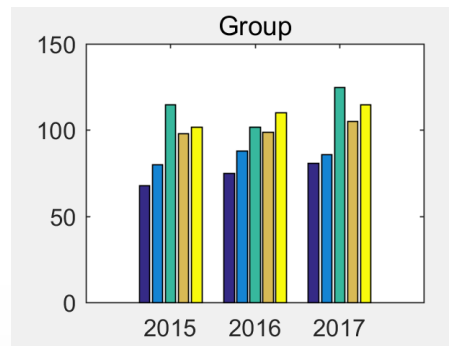
其中，`x`存储横坐标，`y`存储数据，`y`的行数必须与向量`x`的长度相同。选项`style`用于指定分组排列模式。

例4 下表是某公司2015~2017年家电类商品1月份的销售数据，绘制
条形图。

2015~2017年家电类1月份销售数据(单位: 万台)

年份 \ 商品	冰箱	空调	洗衣机	电视机	油烟机
2015	68	80	115	98	102
2016	75	88	102	99	110
2017	81	86	125	105	115

```
x=[2015, 2016, 2017];  
y=[68, 80, 115, 98, 102;  
75, 88, 102, 99, 110;  
81, 86, 125, 105, 115];  
bar(x, y)  
title('Group');
```



2. 统计图

(1) 条形类图形

② 直方图

- ❑ hist函数：绘制直角坐标系下的直方图。
- ❑ rose函数：绘制极坐标系下的直方图。

2. 统计图

(1) 条形类图形

② 直方图

□ hist函数

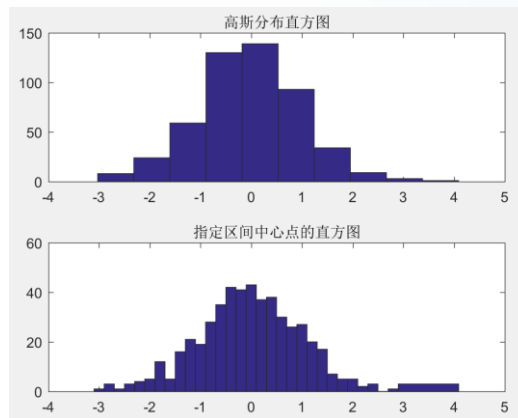
hist(y)

hist(y, x)

其中，y是要统计的数据，x用于指定区间的划分方式。若x是标量，则统计区间均分成x个小区间；若x是向量，则向量x中的每一个数指定分组中心值，元素的个数为数据分组数。x缺省时，默认按10个等分区间进行统计。

例5 绘制服从高斯分布的直方图。

```
y=randn(500,1);  
subplot(2,1,1);  
hist(y);  
title('高斯分布直方图');  
subplot(2,1,2);  
x=-3:0.2:3;  
hist(y,x);  
title('指定区间中心点的直方图'))
```



2. 统计图

(1) 条形类图形

② 直方图

□ rose函数

`rose(theta[, x])`

其中，参数theta用于确定每一区间与原点的角度，选项x用于指定区间的划分方式。

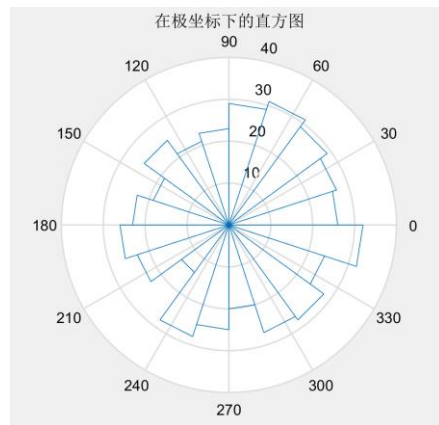
例6 绘制高斯分布数据在极坐标下的直方图。

```
y=randn(500,1);
```

```
theta=y*pi;
```

```
rose(theta)
```

```
title('在极坐标下的直方图')
```



2. 统计图

(2) 面积类图形

① 扇形图

□ pie函数

② 面积图

□ area函数



2. 统计图

(2) 面积类图形

① pie函数

`pie(x, explode)`

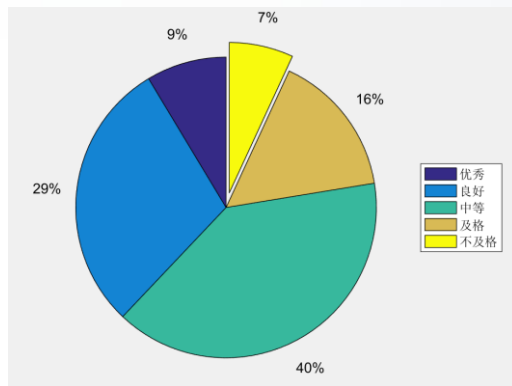
其中，参数x存储待统计数据，选项explode控制图块的显示模式。

例7 某次考试优秀、良好、中等、及格、不及格的人数分别为：5、17、23、9、4，试用扇形统计图作成绩统计分析。

```
score = [5, 17, 23, 9, 4];  
ex = [0, 0, 0, 0, 1];  
pie(score, ex)  
legend('优秀', '良好', '中等', '及格', ...  
      '不及格', 'location', 'eastoutside')
```

说明：

'location' 用于指定图例位置，'eastoutside' 表示图例放在绘图区域右边的外侧。



2. 统计图

(3) 散点类图形

□ scatter函数：散点图

□ stairs函数：阶梯图

□ stem函数：杆图

2. 统计图

(3) 散点类图形

■ scatter函数

`scatter(x, y, 选项, 'filled')`

其中， x 、 y 用于定位数据点，选项用于指定线型、颜色、数据点标记。如果数据点标记是封闭图形，可以用选项‘filled’指定填充数据点标记。该选项省略时，数据点是空心的。

例8 以散点图形式绘制桃心曲线，曲线的参数方程如下：

$$\begin{cases} x = 16\sin^3 t \\ y = 13\cos t - 5\cos(2t) - 2\cos(3t) - \cos(4t) \end{cases}$$

```
t = 0:pi/50:2*pi;
```

```
x = 16*sin(t).^3;
```

```
y = 13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t);
```

```
scatter(x,y,'rd','filled')
```



3. 矢量类图形

□ compass函数：罗盘图

□ feather函数：羽毛图

□ quiver函数：箭头图

3. 矢量类图形

□ quiver函数

quiver函数调用格式:

`quiver(x, y, u, v)`

其中, (x, y) 指定矢量起点, (u, v) 指定矢量终点。x、y、u、v 是同样大小的向量或同型矩阵, 若省略x、y, 则在x-y平面上均匀取若干个点作为起点。

例9 已知向量A、B，求A+B，并用矢量图表示。

$A=[4, 5]$; $B=[-10, 0]$; $C=A+B$;

hold on;

quiver(0, 0, A(1), A(2));

quiver(0, 0, B(1), B(2));

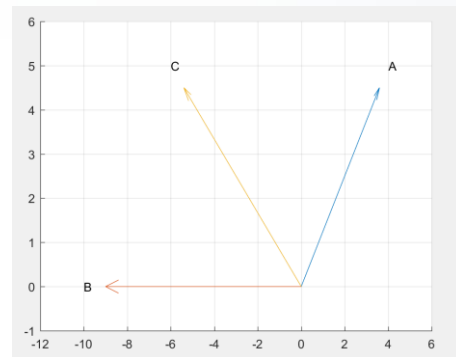
quiver(0, 0, C(1), C(2));

text(A(1),A(2),'A');text(B(1),B(2),'B');

text(C(1),C(2),'C');

axis ([-12, 6, -1, 6])

grid on



4.3 其他形式的二维曲线

- ❑ 其他坐标系下的二维曲线图
- ❑ 统计图
- ❑ 矢量图形

1. 其他坐标系下的二维曲线图

(1) 对数坐标图

`semilogx(x1, y1, 选项1, x2, y2, 选项2, ...)`

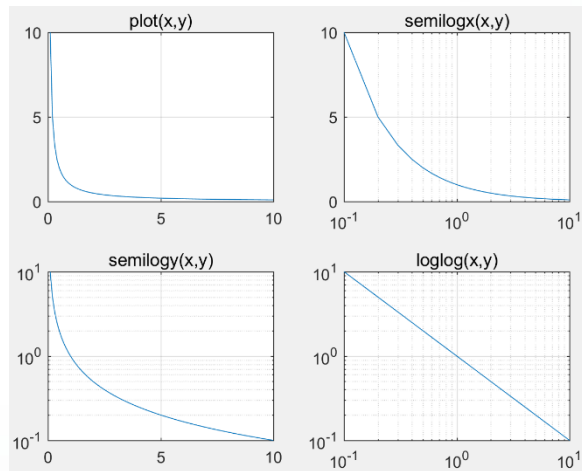
`semilogy(x1, y1, 选项1, x2, y2, 选项2, ...)`

`loglog(x1, y1, 选项1, x2, y2, 选项2, ...)`

其中，`semilogx`函数x轴为常用对数刻度，y轴为线性刻度；`semilogy`函数x轴为线性刻度，y轴为常用对数刻度；`loglog`函数x轴和y轴均采用常用对数刻度。

例1 绘制 $\frac{1}{x}$ 的直角线性坐标图和三种对数坐标图。

```
x=0:0.1:10;  
y=1./x;  
subplot(2,2,1)  
plot(x,y)  
title('plot(x,y)');grid on  
subplot(2,2,2)  
semilogx(x,y)  
title('semilogx(x,y)');grid on  
subplot(2,2,3)  
semilogy(x,y)  
title('semilogy(x,y)');grid on  
subplot(2,2,4)  
loglog(x,y)  
title('loglog(x,y)');grid on
```



1. 其他坐标系下的二维曲线图

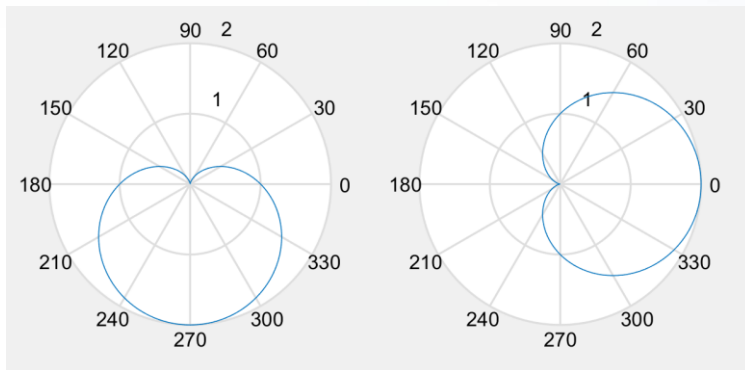
(2) 极坐标图

`polar(theta, rho, 选项)`

其中， θ 为极角， ρ 为极径，选项的内容与`plot`函数相同。

例2 按极坐标方程 $\rho = 1 - \sin\theta$ 绘制心形曲线。

```
t = 0:pi/100:2*pi;  
r = 1-sin(t);  
subplot(1,2,1)  
polar(t,r)  
subplot(1,2,2)  
t1 = t-pi/2;  
r1 = 1-sin(t1);  
polar(t,r1)
```



2. 统计图

□ 条形图

□ 直方图

□ 饼图

□ 散点图

2. 统计图

(1) 条形类图形

① 条形图

- `bar`函数：绘制垂直条形图。
- `barh`函数：绘制水平条形图。

2. 统计图

(1) 条形类图形

① 条形图

□ bar函数

`bar(y, style)`

其中，参数y是数据，选项style用于指定分组排列模式。

“grouped”：簇状分组
“stacked”：堆积分组

例3 绘制分组条形图。

```
y=[1, 2, 3, 4, 5; 1, 2, 1, 2, 1; 5, 4, 3, 2, 1];
```

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

```
bar(y)
```

```
title('Group')
```

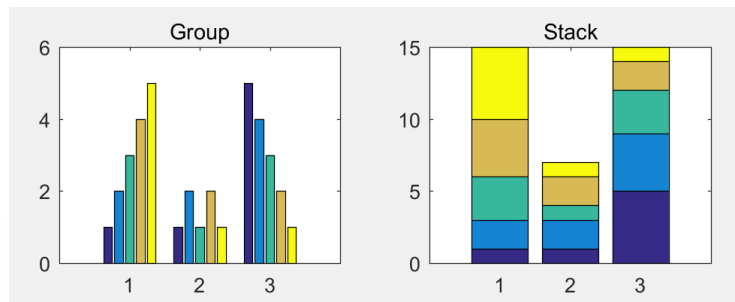
```
subplot(1,2,2)
```

```
bar(y, 'stacked')
```

```
title('Stack')
```

Y矩阵

1	2	3	4	5
1	2	1	2	1
5	4	3	2	1



2. 统计图

(1) 条形类图形

① 条形图

□ bar函数

`bar(x, y, style)`

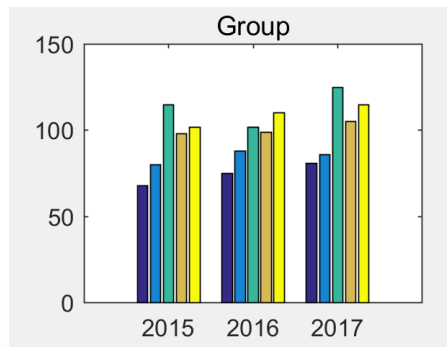
其中，`x`存储横坐标，`y`存储数据，`y`的行数必须与向量`x`的长度相同。选项`style`用于指定分组排列模式。

例4 下表是某公司2015~2017年家电类商品1月份的销售数据，绘制
条形图。

2015~2017年家电类1月份销售数据(单位: 万台)

年份 \ 商品	冰箱	空调	洗衣机	电视机	油烟机
2015	68	80	115	98	102
2016	75	88	102	99	110
2017	81	86	125	105	115

```
x=[2015, 2016, 2017];  
y=[68, 80, 115, 98, 102;  
75, 88, 102, 99, 110;  
81, 86, 125, 105, 115];  
bar(x, y)  
title('Group');
```



2. 统计图

(1) 条形类图形

② 直方图

- ❑ hist函数：绘制直角坐标系下的直方图。
- ❑ rose函数：绘制极坐标系下的直方图。

2. 统计图

(1) 条形类图形

② 直方图

□ hist函数

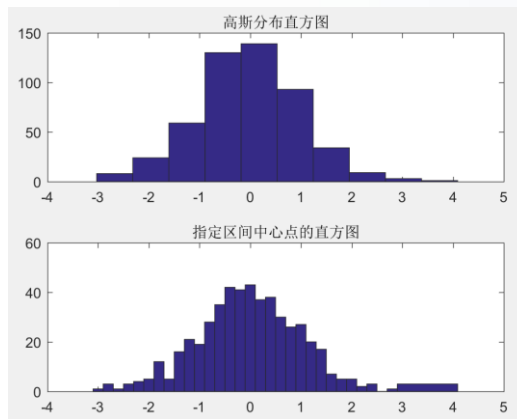
hist(y)

hist(y, x)

其中，y是要统计的数据，x用于指定区间的划分方式。若x是标量，则统计区间均分成x个小区间；若x是向量，则向量x中的每一个数指定分组中心值，元素的个数为数据分组数。x缺省时，默认按10个等分区间进行统计。

例5 绘制服从高斯分布的直方图。

```
y=randn(500,1);  
subplot(2,1,1);  
hist(y);  
title('高斯分布直方图');  
subplot(2,1,2);  
x=-3:0.2:3;  
hist(y,x);  
title('指定区间中心点的直方图'))
```





2. 统计图

(1) 条形类图形

② 直方图

□ rose函数

`rose(theta[, x])`

其中，参数theta用于确定每一区间与原点的角度，选项x用于指定区间的划分方式。

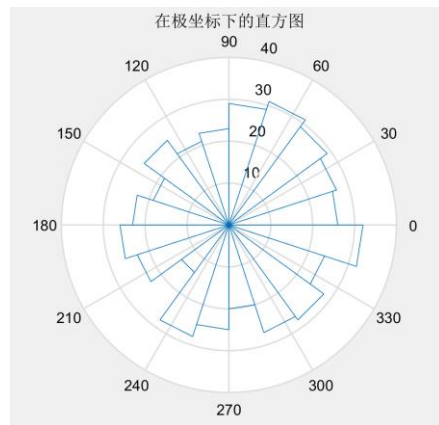
例6 绘制高斯分布数据在极坐标下的直方图。

```
y=randn(500,1);
```

```
theta=y*pi;
```

```
rose(theta)
```

```
title('在极坐标下的直方图')
```



2. 统计图

(2) 面积类图形

① 扇形图

□ pie函数

② 面积图

□ area函数

2. 统计图

(2) 面积类图形

① pie函数

`pie(x, explode)`

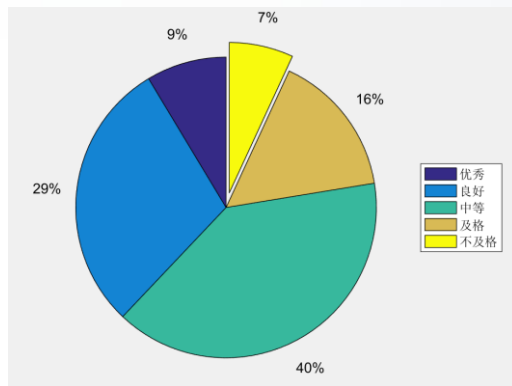
其中，参数x存储待统计数据，选项explode控制图块的显示模式。

例7 某次考试优秀、良好、中等、及格、不及格的人数分别为：5、17、23、9、4，试用扇形统计图作成绩统计分析。

```
score = [5, 17, 23, 9, 4];  
ex = [0, 0, 0, 0, 1];  
pie(score, ex)  
legend('优秀', '良好', '中等', '及格', ...  
      '不及格', 'location', 'eastoutside')
```

说明：

'location' 用于指定图例位置，'eastoutside' 表示图例放在绘图区域右边的外侧。



2. 统计图

(3) 散点类图形

□ scatter函数：散点图

□ stairs函数：阶梯图

□ stem函数：杆图

2. 统计图

(3) 散点类图形

■ scatter函数

`scatter(x, y, 选项, 'filled')`

其中， x 、 y 用于定位数据点，选项用于指定线型、颜色、数据点标记。如果数据点标记是封闭图形，可以用选项‘filled’指定填充数据点标记。该选项省略时，数据点是空心的。

例8 以散点图形式绘制桃心曲线，曲线的参数方程如下：

$$\begin{cases} x = 16\sin^3 t \\ y = 13\cos t - 5\cos(2t) - 2\cos(3t) - \cos(4t) \end{cases}$$

```
t = 0:pi/50:2*pi;
```

```
x = 16*sin(t).^3;
```

```
y = 13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t);
```

```
scatter(x,y,'rd','filled')
```



3. 矢量类图形

□ compass函数：罗盘图

□ feather函数：羽毛图

□ quiver函数：箭头图

3. 矢量类图形

□ quiver函数

quiver函数调用格式:

`quiver(x, y, u, v)`

其中, (x, y) 指定矢量起点, (u, v) 指定矢量终点。x、y、u、v 是同样大小的向量或同型矩阵, 若省略x、y, 则在x-y平面上均匀取若干个点作为起点。

例9 已知向量A、B，求A+B，并用矢量图表示。

$A=[4, 5]$; $B=[-10, 0]$; $C=A+B$;

hold on;

quiver(0, 0, A(1), A(2));

quiver(0, 0, B(1), B(2));

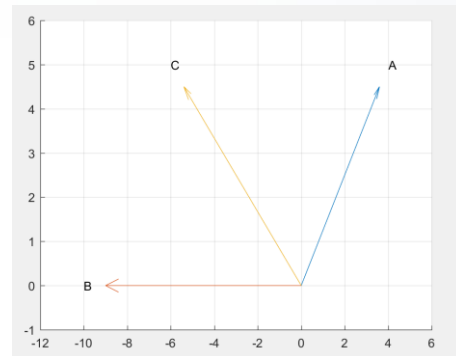
quiver(0, 0, C(1), C(2));

text(A(1),A(2),'A');text(B(1),B(2),'B');

text(C(1),C(2),'C');

axis ([-12, 6, -1, 6])

grid on

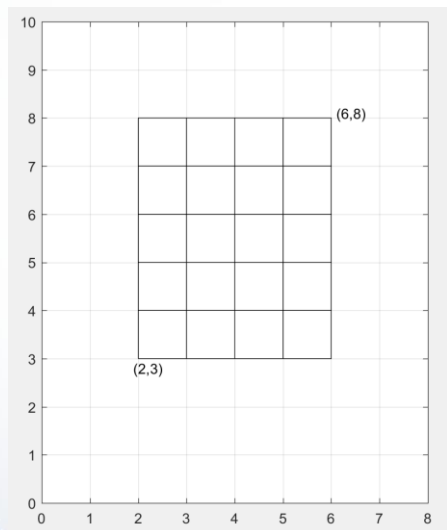


• 4.5 三维曲面

- 平面网格数据的生成
- 绘制三维曲面的mesh函数和surf函数
- fmesh函数和fsurf函数

1. 平面网格数据的生成

用矩阵X、Y分别存储每一个小矩形顶点的x坐标与y坐标，
矩阵X、Y就是该矩形区域的xy平面网格坐标矩阵。



X						
6x5 double						
	1	2	3	4	5	
1	2	3	4	5	6	^
2	2	3	4	5	6	
3	2	3	4	5	6	
4	2	3	4	5	6	
5	2	3	4	5	6	
6	2	3	4	5	6	v

Y						
6x5 double						
	1	2	3	4	5	
1	3	3	3	3	3	^
2	4	4	4	4	4	
3	5	5	5	5	5	
4	6	6	6	6	6	
5	7	7	7	7	7	
6	8	8	8	8	8	v

1. 平面网格数据的生成

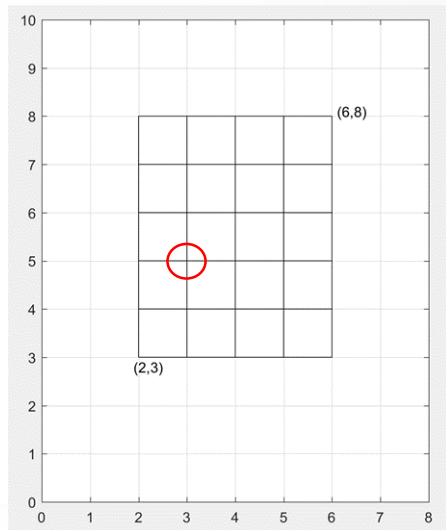
在MATLAB中，产生平面区域内的网格坐标矩阵有两种方法。

(1) 利用矩阵运算生成。

```
>> x = 2:6;  
>> y = (3:8)';  
>> X = ones(size(y))*x;  
>> Y = y*ones(size(x));
```

X						
6x5 double						
	1	2	3	4	5	
1		2	3	4	5	6
2		2	3	4	5	6
3		2	3	4	5	6
4		2	3	4	5	6
5		2	3	4	5	6
6		2	3	4	5	6

Y						
6x5 double						
	1	2	3	4	5	
1		3	3	3	3	3
2		4	4	4	4	4
3		5	5	5	5	5
4		6	6	6	6	6
5		7	7	7	7	7
6		8	8	8	8	8



X和Y相同位置上的元素，如 X_{32} 、 Y_{32} 是区域的第3行第2列网格点的坐标(3, 5)。

1. 平面网格数据的生成

(2) 利用meshgrid函数生成。

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

其中，参数x、y为向量，存储网格点坐标的X、Y为矩阵。

```
>> x = 2:1:6;
```

```
>> y = (3:1:8)';
```

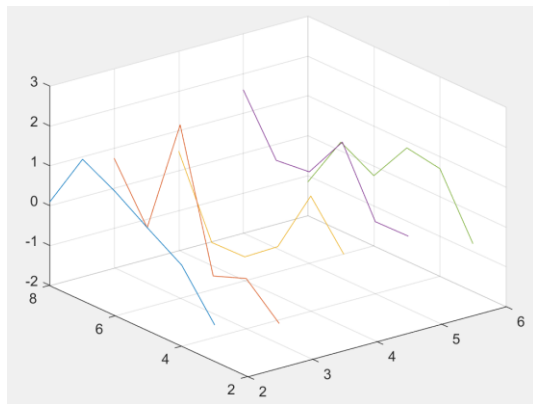
```
>> [X, Y] = meshgrid(x, y);
```

第3行命令生成的网格坐标矩阵X、Y与方法(1)得到的相同。

若根据每一个网格点上的 x 、 y 坐标求函数值 z ，则得到函数值矩阵 Z 。矩阵 X 、 Y 、 Z 中的各个列向量，对应于一条曲线数据点的坐标。

例1 绘制空间曲线。

```
x = 2:6;  
y = (3:8)';  
[X, Y] = meshgrid(x, y);  
Z = randn(size(X));  
plot3(X, Y, Z)  
grid on;
```



2. 绘制三维曲面的函数

□ mesh函数和surf函数的调用格式:

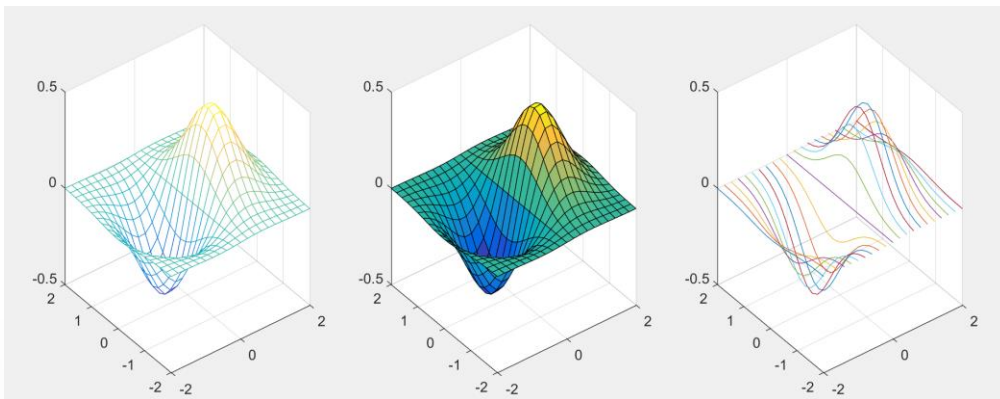
`mesh(x, y, z, c)`

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

其中, x 、 y 是网格坐标矩阵, z 是网格点上的高度矩阵, c 用于指定在不同高度下的曲面颜色。 c 省略时, 颜色的设定正比于图形的高度。

例2 绘制三维曲面图 $z = xe^{-x^2 - y^2}$ 。

```
t = -2:0.2:2;  
[X, Y] = meshgrid(t);  
Z = X .* exp(-X.^2 - Y.^2);  
subplot(1, 3, 1)  
mesh(X, Y, Z);  
subplot(1, 3, 2)  
surf(X, Y, Z);  
subplot(1, 3, 3)  
plot3(X, Y, Z);  
grid on
```



2. 绘制三维曲面的函数

□ mesh函数和surf函数的其他调用格式:

`mesh(z, c)`

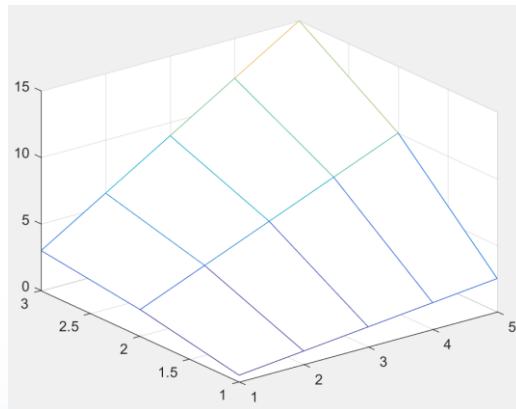
`surf(z, c)`

当x、y省略时，z矩阵的第2维下标当作x轴坐标，z矩阵的第1维下标当作y轴坐标。

```
>> t = 1:5;
```

```
>> z = [0.5*t; 2*t; 3*t];
```

```
>> mesh(z);
```

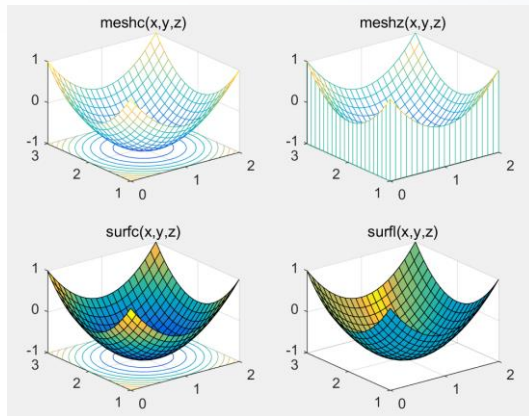


2. 绘制三维曲面的函数

- ❑ 带等高线的三维网格曲面函数`meshc`
- ❑ 带底座的三维网格曲面函数`meshz`
- ❑ 具有等高线的曲面函数`surf`
- ❑ 具有光照效果的曲面函数`surf1`

例3 用4种方式绘制函数 $z = (x - 1)^2 + (y - 2)^2 - 1$ 的曲面图。
其中, $x \in [0, 2]$, $y \in [1, 3]$ 。

```
[x,y]=meshgrid(0:0.1:2,1:0.1:3);  
z=(x-1).^2+(y-2).^2-1;  
subplot(2,2,1);  
meshc(x,y,z);title('meshc(x,y,z)')  
subplot(2,2,2);  
meshz(x,y,z);title('meshz(x,y,z)')  
subplot(2,2,3);  
surfc(x,y,z);title('surfc(x,y,z)')  
subplot(2,2,4);  
surfl(x,y,z);title('surfl(x,y,z)')
```



3. 标准三维曲面

(1) sphere函数

$$[x, y, z] = \text{sphere}(n)$$

产生3个 $(n+1)$ 阶的方阵，采用这3个矩阵可以绘制出圆心位于原点、半径为1的单位球体。

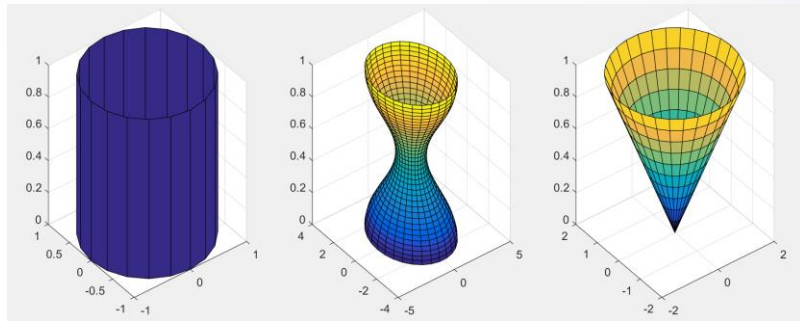
(2) cylinder函数

$$[x, y, z] = \text{cylinder}(R, n)$$

其中，参数 R 是一个向量，存放柱面各个等间隔高度上的半径， n 表示在圆柱圆周上有 n 个间隔点，默认有20个间隔点。

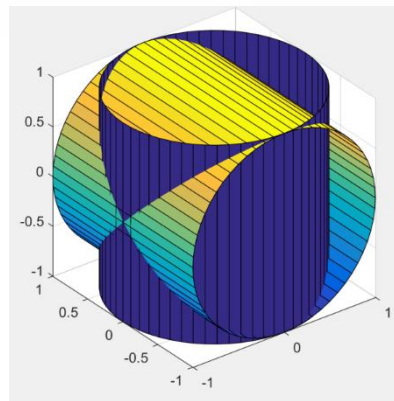
例4 用cylinder函数分别绘制柱面、花瓶和圆锥面。

```
subplot(1, 3, 1);  
[x, y, z]=cylinder;  
surf(x, y, z);  
subplot(1, 3, 2);  
t=linspace(0, 2*pi, 40);  
[x, y, z]= cylinder(2+cos(t), 30);  
surf(x, y, z);  
subplot(1, 3, 3);  
[x, y, z]= cylinder(0:0.2:2, 30);  
surf(x, y, z);
```



例5 用cylinder函数绘制两个相互垂直且直径相等的圆柱面的相交图形。

```
[x, y, z]= cylinder(1, 60);  
z=[-1*z(2, :); z(2, :)];  
surf(x, y, z)  
hold on  
surf(y, z, x)  
axis equal
```



3. 标准三维曲面

(3) peaks 函数

peaks函数的调用格式:

peaks(n)

```
>> p1=peaks(10);
```

peaks(V)

```
>> p2=peaks;
```

peaks(x, y)

```
>> p3=peaks(-3:0.2:3);
```

peaks

```
>> [x, y]=meshgrid(-2:0.1:2, 0:0.1:5);
```

```
>> p4=peaks(x, y);
```


4. fmesh函数和fsurf函数

用于绘制参数方程定义的曲面

```
fsurf(funx, funy, funz, uvlims)
```

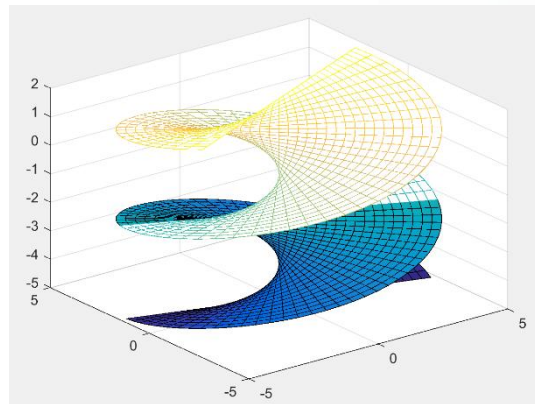
```
fmesh(funx, funy, funz, uvlims)
```

其中，funx、funy、funz代表定义曲面x、y、z坐标的函数，通常采用函数句柄的形式。uvlims为funx、funy和funz的自变量的取值范围，用4元向量[umin, umax, vmin, vmax]描述，默认为[-5, 5, -5, 5]。

例6 绘制螺旋曲面。

$$\begin{cases} x = u \sin v \\ y = -u \cos v \\ z = v \end{cases} \quad -5 < u < 5, \quad -5 < v < 2$$

```
funx = @(u,v) u.*sin(v);  
funy = @(u,v) -u.*cos(v);  
funz = @(u,v) v;  
fsurf(funx,funy,funz,[-5 5 -5 -2])  
hold on  
fmesh(funx,funy,funz,[-5 5 -2 2])  
hold off
```

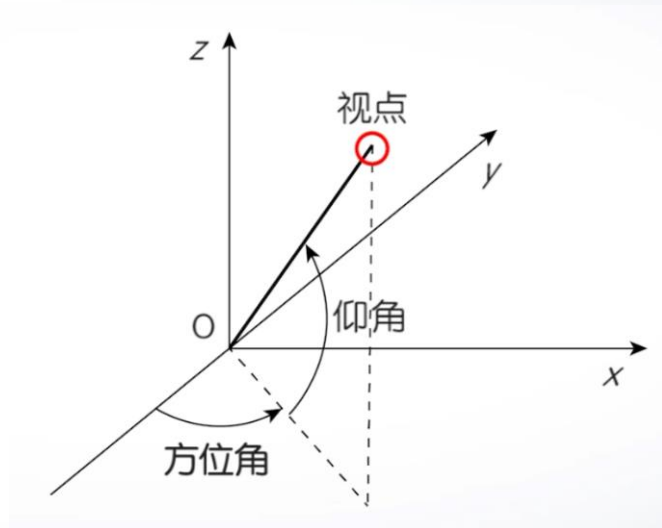


• 4.6 图形修饰处理

- 视点处理
- 色彩处理
- 图形的裁剪处理

1. 视点处理

- 方位角：视点与原点连线在 xy 平面上的投影与 y 轴负方向形成的角度，正值表示逆时针，负值表示顺时针。
- 仰角：视点与原点连线与 xy 平面的夹角，正值表示视点在 xy 平面上方，负值表示视点在 xy 平面下方。



1. 视点处理

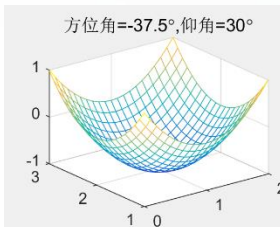
(1) view函数的基本用法

`view(az, el)`

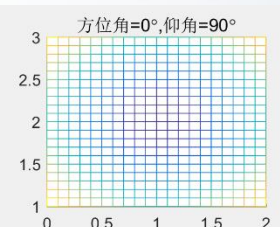
其中，az为方位角，el为仰角。系统默认的视点定义为方位角 -37.5° ，仰角 30° 。

例1 绘制函数 $z = (x - 1)^2 + (y - 2)^2 - 1$ 曲面，并从不同视点展示曲面。

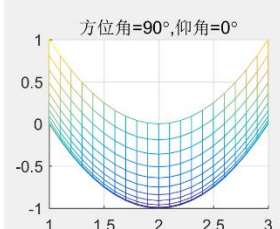
```
[x, y]=meshgrid(0:0.1:2, 1:0.1:3);  
z=(x-1).^2+(y-2).^2-1;  
subplot(2, 2, 1); mesh(x, y, z)  
title('方位角=-37.5{\circ}, 仰角=30{\circ}')
```



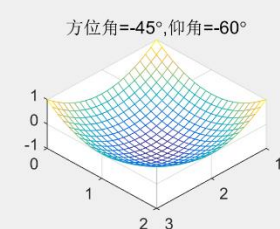
```
subplot(2, 2, 2); mesh(x, y, z)  
view(0, 90); title('方位角=0{\circ}, 仰角=90{\circ}')
```



```
subplot(2, 2, 3); mesh(x, y, z)  
view(90, 0); title('方位角=90{\circ}, 仰角=0{\circ}')
```



```
subplot(2, 2, 4); mesh(x, y, z)  
view(-45, -60); title('方位角=-45{\circ}, 仰角=-60{\circ}')
```



1. 视点处理

(2) view函数的其他用法

`view(x, y, z)`

`view(2)`

`view(3)`

2. 色彩处理

(1) 颜色的向量表示

向量元素在 $[0, 1]$ 范围内取值，3个元素依次表示红、绿、蓝3种颜色的相对亮度，称为RGB三元组。

[R	G	B]
↑	↑	↑
红	绿	蓝

[0	0	1]	: 蓝色
[1	0	0]	: 红色
[0	1	0]	: 绿色
[1	1	1]	: 白色
[0	0	0]	: 黑色

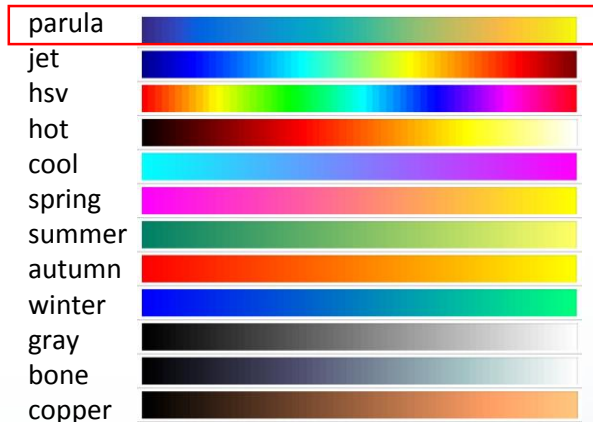
2. 色彩处理

(2) 色图 (Colormap)

□ 色图矩阵

```
>> cmap = colormap(parula(5))  
cmap =  
    0.2081    0.1663    0.5292  
    0.0795    0.5159    0.8328  
    0.1986    0.7214    0.6310  
    0.8266    0.7320    0.3464  
    0.9763    0.9831    0.0538
```

□ 内建色图



2. 色彩处理

(2) 色图 (Colormap)

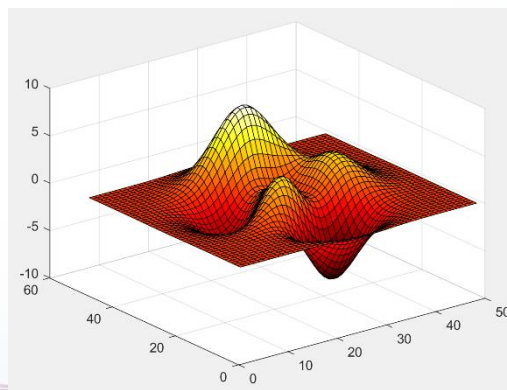
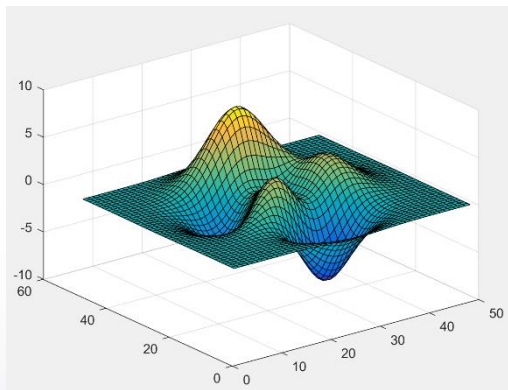
- 指定当前图形使用的色图

`colormap cmapname`

`colormap(cmap)`

```
>> surf(peaks)
```

```
>> colormap hot
```



2. 色彩处理

(2) 色图 (Colormap)

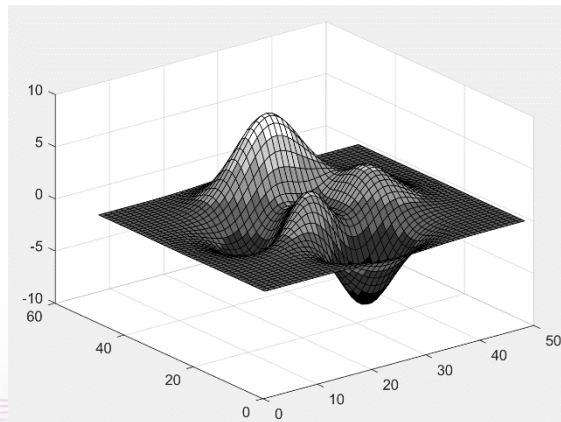
□ 创建色图矩阵

色图矩阵的每一行是RGB三元组。可以自定义色图矩阵，也可以调用MATLAB提供的函数来定义色图矩阵。

例2 创建一个灰色系列色图矩阵。

```
c = [0, 0.2, 0.4, 0.6, 0.8, 1.0]';  
cmap = [c, c, c];  
surf(peaks)  
colormap(cmap)
```

`cmap=gray(6);`



2. 色彩处理

(3) 三维图形表面的着色

可以用shading函数来改变着色方式。

`shading faceted`

每个网格片用其高度对应的颜色进行着色，网格线是黑色。这是默认着色方式。

`shading flat`

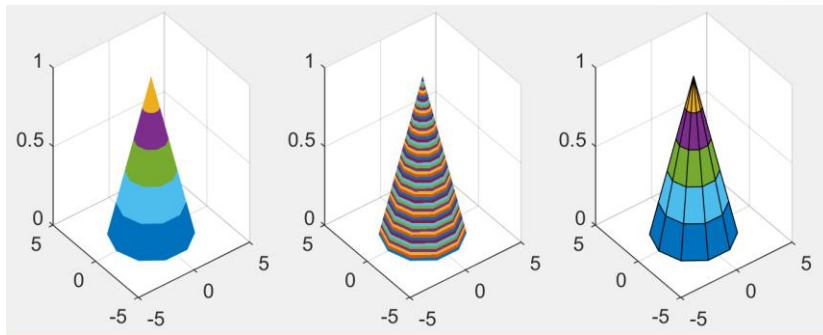
每个网格片用同一个颜色进行着色，且网格线也用相应的颜色。

`shading interp`

网格片内采用颜色插值处理。

例3 使用同一色图，以不同着色方式绘制圆锥体。

```
[x,y,z]= cylinder(pi:-pi/5:0,10);  
colormap(lines);  
subplot(1,3,1);  
surf(x,y,z); shading flat  
subplot(1,3,2);  
surf(x,y,z); shading interp  
subplot(1,3,3);  
surf(x,y,z);
```

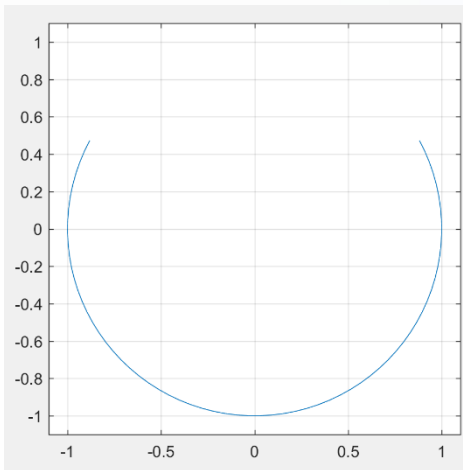


3. 图形的裁剪处理

将图形中需要裁剪部分对应的函数值设置成NaN，这样在绘制图形时，函数值为NaN的部分将不显示出来，从而达到对图形进行裁剪的目的。

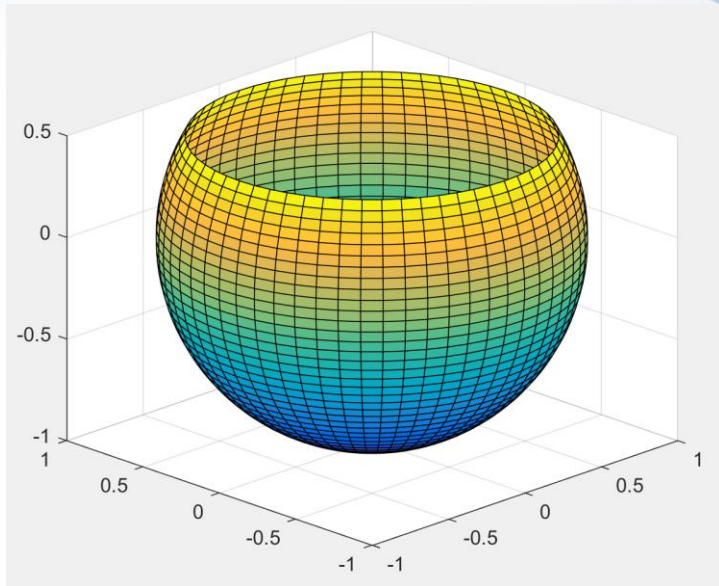
例4 绘制3/4圆。

```
t = linspace(0, 2*pi, 100);  
x = sin(t);  
y = cos(t);  
p = y > 0.5;  
y(p) = NaN;  
plot(x, y)  
axis([-1.1, 1.1, -1.1, 1.1])  
axis square  
grid on
```



例5 绘制3/4球面。

```
[X, Y, Z] = sphere(60);  
p = Z>0.5;  
Z(p) = NaN;  
surf(X, Y, Z)  
axis([-1, 1, -1, 1, -1, 1])  
axis equal  
view(-45, 20)
```



• 4.7 交互式绘图工具

- “绘图”选项卡
- 图形窗口绘图工具
- 图形窗口菜单和工具栏

1. “绘图”选项卡



“绘图”选项卡的工具条提供了绘制图形的基本命令。

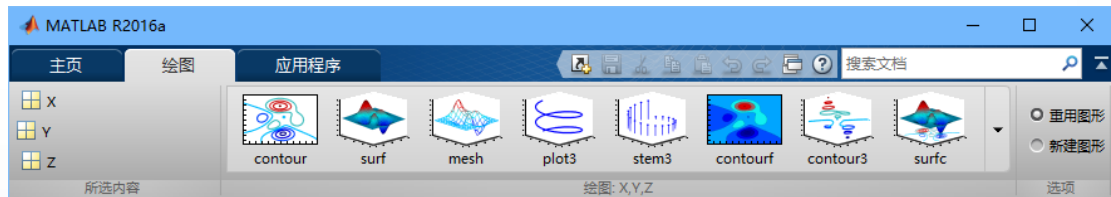
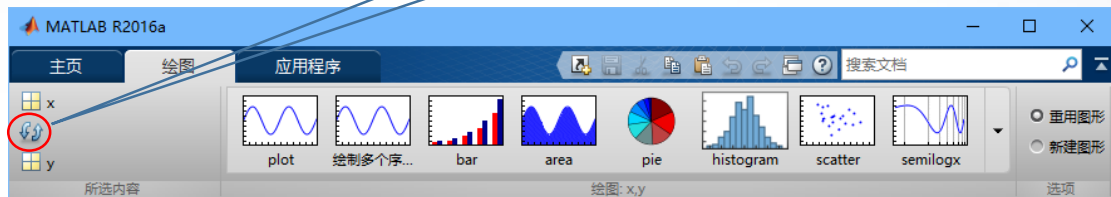
- ❑ “所选内容”命令组：用于显示已选中用于绘图的变量；
- ❑ “绘图”命令组：提供了绘制各种图形的命令；
- ❑ “选项”命令组：用于设置绘图时是否新建图形窗口。

例1 选择已有变量绘制图形。

名称	值
h	21x21 double
u	21x21 double
v	21x21 double
x	1x100 double
X	11x31 double
y	1x100 double
Y	11x31 double
y1	1x100 double
y2	1x100 double
y3	1x100 double
Z	11x31 double

名称	值
h	21x21 double
u	21x21 double
v	21x21 double
x	1x100 double
X	11x31 double
y	1x100 double
Y	11x31 double
y1	1x100 double
y2	1x100 double
y3	1x100 double
Z	11x31 double

切换变量顺序



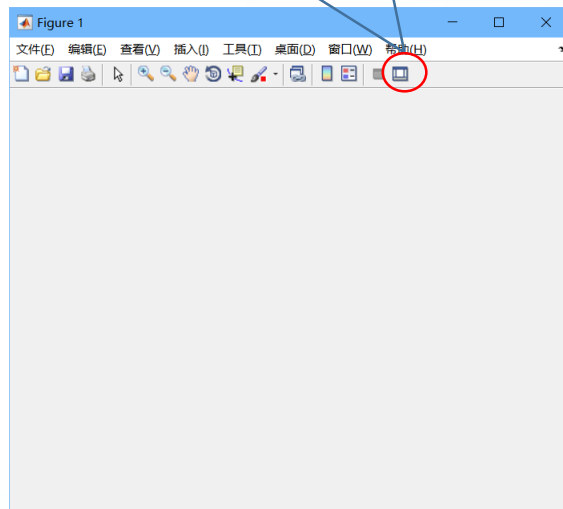
2. 绘图工具

(1) 显示绘图工具

- ❑ “显示绘图工具和停靠图形”按钮
- ❑ 命令行窗口中输入命令

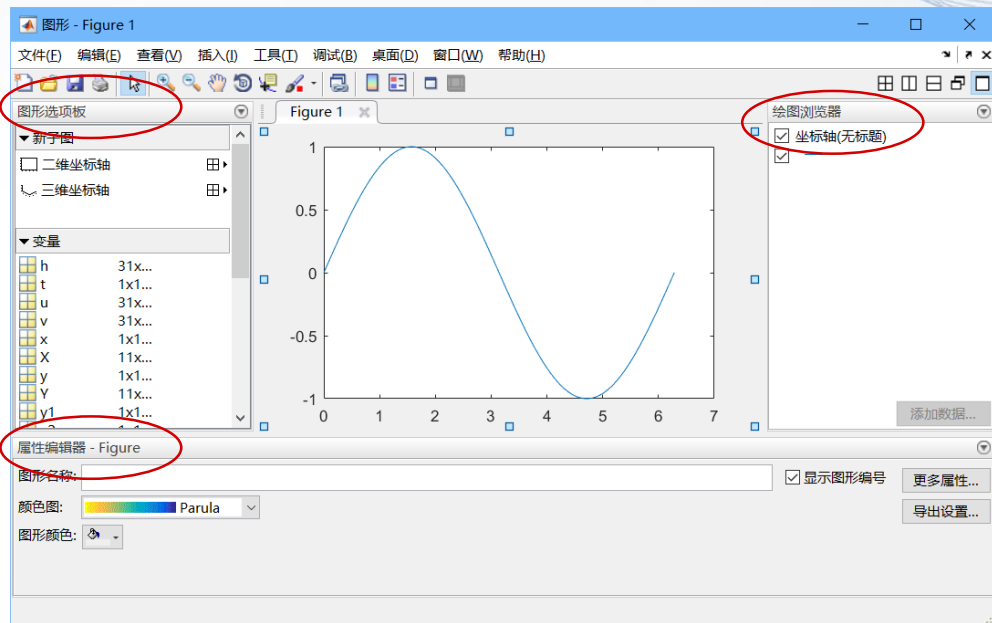
>> `plottools`

“显示绘图工具和停靠图形”
按钮



(2) 绘图工具的组成

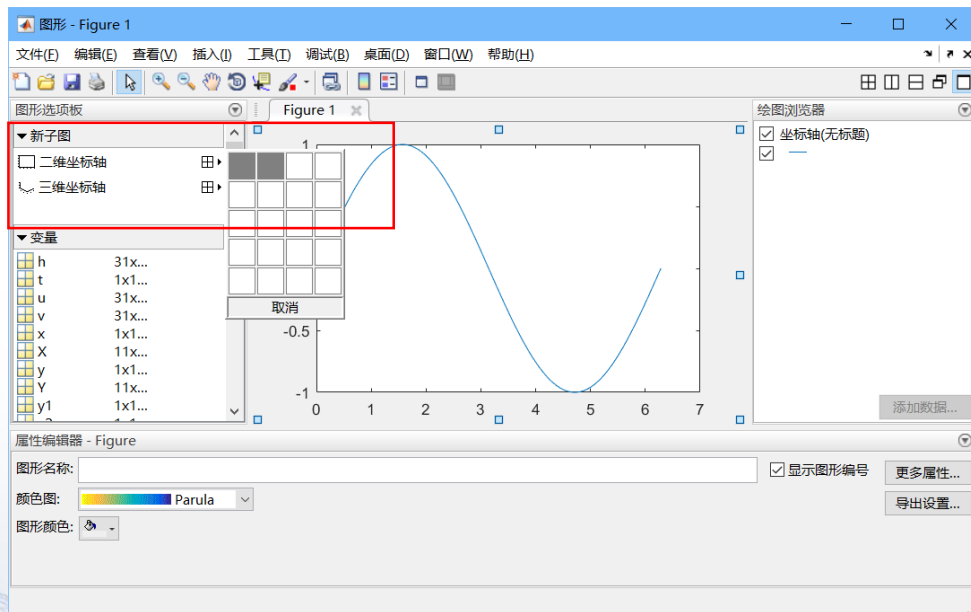
- 图形选项板
- 绘图浏览器
- 属性编辑器



图形选项板

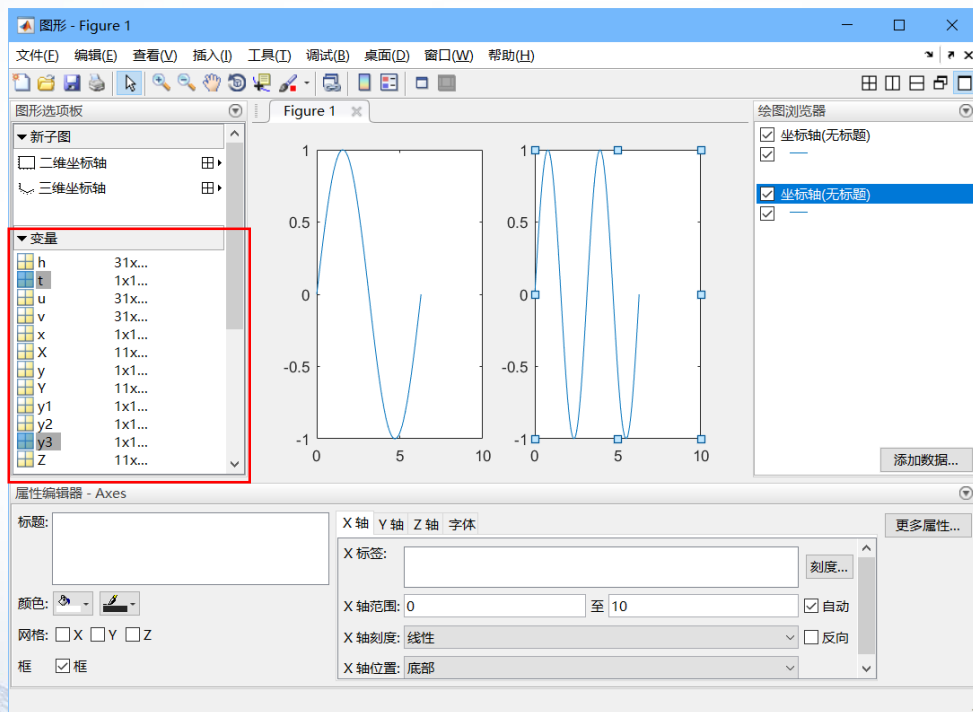
用于在图形窗口中添加和排列子图，观察和选择绘图数据以及添加图形标注。

“新子图”面板



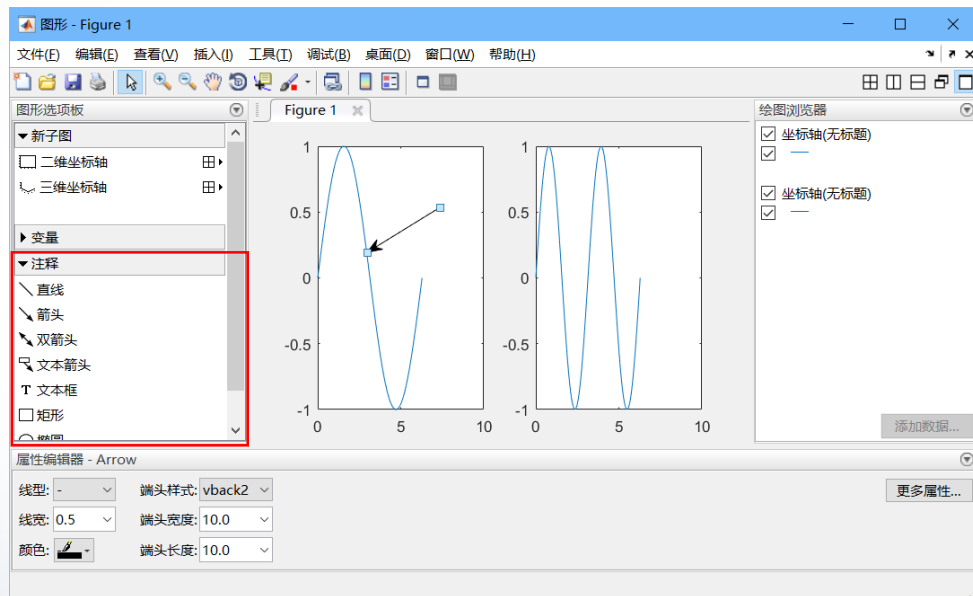
图形选项板

“变量”面板



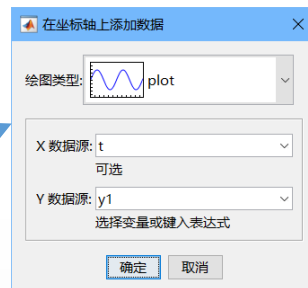
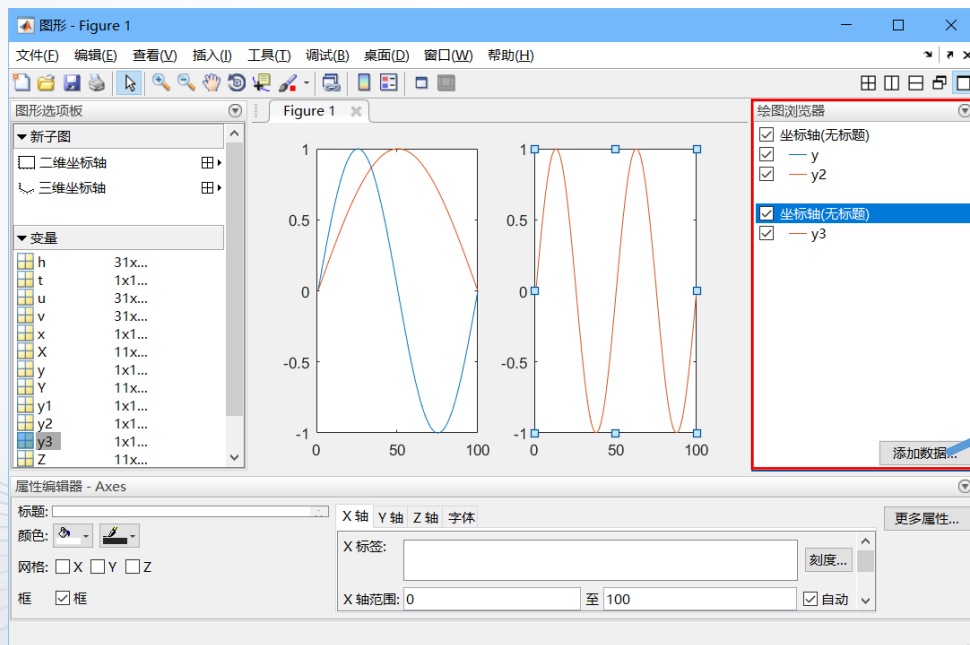
□ 图形选项板

• “注释”面板



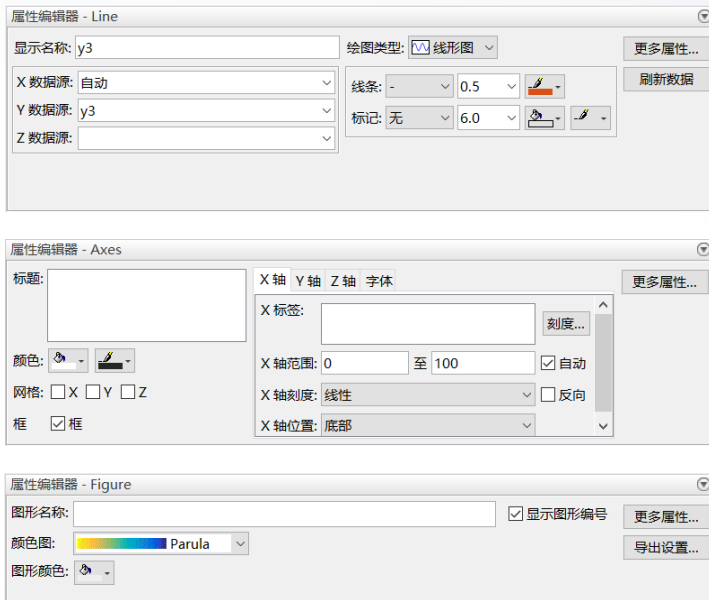
□ 绘图浏览器

以图例的方式列出图形中的元素。

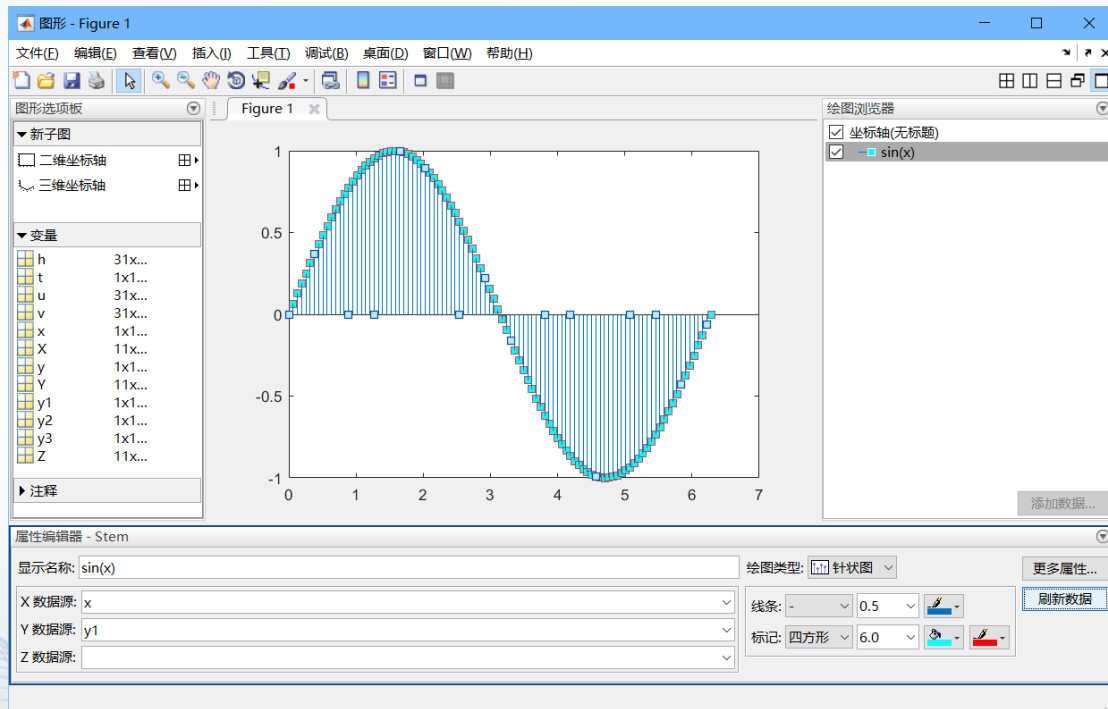


属性编辑器

用于观测和设置所选对象的名称、颜色、填充方法等参数。不同类型的对象，属性编辑器中的内容不同。



例2 用工作区的变量绘制曲线，改变曲线的类型、颜色、线型等。



3. 图形窗口菜单和工具栏

□ 工具栏



图形文件操作

图形操作

添加颜色栏、图例

图形窗口的显示模式

图形绘制完成后，可以用“文件”菜单中的“生成代码”命令，将实施在图形上的这些操作命令输出成脚本。也可以用“保存”命令将图形窗口内容保存为fig文件。