

《matlab初学者教程》

一、matlab介绍

1. 命令行窗口中的输入小注意

一行敲完了不要加分号。

一个语句在一行内太长了，需要在第一行末尾加上**完整省略号**（.....），然后开始第二行书写。

```
1 x1=1/2+1/3+1/4.....  
2 +1/5+1/6
```

2. 编辑调试器

编辑调试器一般用于创建M文件，或者修改已存在的M文件。

创建一个M文件：主页-新建脚本

打开一个M文件：主页-打开-文件夹中选择

在编辑器窗口输入M文件的语句，然后在命令行窗口输入M文件名就可以输出相应的结果。

在脚本文件中，每一行的末尾需要输入分号。



3.图像窗口

在脚本文件中可以进行相关图像的打印。

例如下面输出从0到6以0.1为间隔的sin(x)的图像。

```
1 % this m-file calculates and plots the function sin(x) for 0<=x<=6.  
2 x=0:0.1:6;  
3 y=sin(x);  
4 plot(x,y);
```

4. whos和clear

在命令行输入whos，可以查看在**当前工作区**内的所有变量和数组状况表（包括执行过的M文件中的变量和数组）。

命令行窗口

不熟悉 MATLAB? 请参阅有关[快速入门](#)的资源。

是不是想输入:

```
>> test1
```

the area of the circle is 19.635

```
>> test1
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
area	1x1	8	double	
areal	1x1	8	double	
radius	1x1	8	double	
string	1x32	64	char	
x	1x61	488	double	
x1	1x1	8	double	
y	1x61	488	double	

之前在工作区的变量

修改前使用过的test1.m中的变量

修改后的图像变量

 >>

可以在命令行输入之前使用过的变量（可以参考工作区），然后就可以输出对应的变量的值啦。

在命令行中使用clear可以清除本工作区的变量。

```
1 | clear var1, var2
```

然后就可以看到在工作区对应的变量var1, var2消失了。

二、matlab基础

1. 变量的初始化

数组的大小由数组的行数和列数共同决定，**注意行数在前**。

(1)用赋值语句初始化变量

```
1 | var=expression
```

var是变量名，expression可以是一个标量、一个数组或者常量、其他变量和数学运算符的联合。

数组变量也可以初始化。**用括号和分号建立数组**。所有元素按照行阶排序，每一行的值从左往右，顶部的行置于最前，底部的行置于最后。

```
1 | [3.4] %表示1*1数组，一个标量，此时数组可以省略
2 | [1 2 3] %创建了1*3数组，即一个一维行向量
3 | [1;2;3] %创建了3*1数组，即一个一维列向量
4 | [1,2,3;4,5,6] %创建了2*3数组
5 |
6 | [1,2,3
7 | 4,5,6] %创建了一个2*3数组
8 |
9 | [] %空数组，注意和全为0的数组的区别
```

数组创建时，**并非每个元素都需要定义**。没有定义的元素会被自动创建并初始化为0。

```
>> c(2,3)=5

c =

     0     0     0
     0     0     5

fx >>
```

在每个赋值语句末尾的分号有“停止重复”的作用。

重复可以方便快速查验代码的正确性和显示赋值语句的结果，但会降低工作速度。

```
>> c(2,3)=5
c =
     0     0     0
     0     0     5
>> b(2,3)=5;
fx >>
```

未加分号，再次输出矩阵c

加了分号，不在重复输出

(2)用捷径表达式赋值

克隆运算符：可以指定一系列的数值，它规定了初值、末值和步长。

```
1 first:step:last
```

当步长为1时，可以省略step，变成：

```
1 first:last
```

例如：下面的语句可以生成包含100个元素的数组 $[\frac{\pi}{100}, \frac{2\pi}{100}, \dots, \pi]$

```
1 angles=(.01:.01:1)*pi;
```

转置运算符：用行向量的转置初始化列向量或者更加复杂的矩阵。

```
1 >> f=[1:4] '
2
3 f =
4
5     1
6     2
7     3
8     4
```

```

9   >> t=[3:6];
10  >> g=[t' t']
11
12  g =
13
14      3      3
15      4      4
16      5      5
17      6      6
18

```

(3)用内置函数来初始化

用matlab内置函数来初始化数组。

```

1  zeros(n) %创建一个n*n零矩阵
2  zeros(n,m) %创建一个n*m零矩阵
3  zeros(size(arr)) %创建一个与数组arr大小相同的零矩阵
4  ones(n) %创建一个n*n的全一矩阵
5  ones(n,m) %创建一个n*m全一矩阵
6  eye(n) %创建一个n*n的单位矩阵
7  eye(n,m) %创建一个n*m的单位矩阵
8  length(arr) %返回一个向量的长度，或者返回二维数组中最长的哪一维的长度
9  size(arr) %返回指定数组的行数和列数

```

(4)使用关键字input初始化变量

在脚本文件中，可以用来提示使用者输入。显示提示语句，等待用户输入一个值。

```

1  my_val=input('Enter an input value:')

```

输入一个数：直接键入；

输入一个数组：必须加上中括号[]。

按下回车键时，窗口中输入的任何值都会被存储于变量my_val。

如果什么都不输入就按下回车键，那么它就是一个空矩阵。

如果input函数中含有字符's'作为它的第二个参数，那么输入的数据就被当作字符串。

```

1  >> my_val=input('Enter a variable:')
2  Enter a variable:[1,2,3;4,5,6]
3
4  my_val =
5
6      1      2      3
7      4      5      6
8
9  >> int1=input('Enter an input int:')
10 Enter an input int:1.23
11
12 int1 =
13
14      1.2300
15
16 >> int2=input('Enter an input int:','s')

```

```

17 Enter an input int:1.23
18
19 int2 =
20
21     '1.23'
22

```

2.多维数组

matlab允许创建多维数组，这些数组的每一维对应一个下标，每一个单个元素都可以通过它的下标被调用。

例如下面用两个语句创建2×3×2的数组：

```

1  >> c(:,:,1)=[1,2,3;4,5,6];
2  >> c(:,:,2)=[6,7,8;9,10,11];
3  >> whos c
4      Name      Size      Bytes  Class      Attributes
5
6      c          2x3x2          96  double
7
8  >> c
9
10 c(:,:,1) =
11
12      1      2      3
13      4      5      6
14
15
16 c(:,:,2) =
17
18      6      7      8
19      9     10     11
20

```

注意：上面用冒号:表示省略/指代。这个三维数组的显示方法和一般的大致相同。

访问多维数组时，最好写清楚访问的位置，即行数和列数要写清楚。

```

1  >> c(2,2,2)
2
3  ans =
4
5      10
6

```

3.特殊变量

matlab中有一些预先定义好的变量，可以随时调用。

```

1  pi %有15个有效值的圆周率
2  i,j %代表虚数
3  inf 或者 Inf %代表无穷大，一般是除以0产生的
4  NaN %表示没有这个数，一般是通过数学运算得到的，比如0/0
5  clock %这个特殊变量包含了当前的年、月、日、时、分、秒，是一个六元素行向量
6  date %当前的日期，使用字符形式，如30-Dec-2007

```

```

7  eps %变量名为epsilon (ε) 的简写。表示计算机能够识别的两个数之间的最小数
8  ans %用于存储表达式的结果，当一个结果没有明确的赋值给某个变量时
9
10
11 %测试如下：
12
13 >> clock
14
15 ans =
16
17     1.0e+03 *
18
19     2.0210     0.0030     0.0050     0.0150     0.0520     0.0283
20
21 >> date
22
23 ans =
24
25     '05-Mar-2021'
26
27 >> eps
28
29 ans =
30
31     2.2204e-16
32

```

4. 标量运算和数组运算

(1) 标量运算符

加法，减法，乘法，除法，乘方，括号

(2) 数组运算和矩阵运算

1. 数组运算

针对两数组相对应的运算使用的。**两个数组的行和列必须相同（即必须时方阵）**。也可以用于一个标量和数组的运算，**此时标量将和数组的每一个元素进行运算。**

```

1  >> a=[1,3;2,4];
2  >> b=[-1,3;-2,1];
3  >> a+b
4
5  ans =
6
7      0      6
8      0      5
9
10 >> a+4
11
12 ans =
13
14      5      7
15      6      8

```

2. 矩阵运算

遵守线性代数的一般规则，比如矩阵的乘法。

一般在进行矩阵的加减法时，数组和矩阵运算时没有区别的；但是在有些操作中，它们有一定的区别，这时我们在运算符前加一个点来表示这是一个数组运算。

表 2.6 常见的数组和矩阵运算

运算	MATLAB 形式	注释
数组加法	$A+B$	数组加法和矩阵加法相同
数组减法	$A-B$	数组减法和矩阵减法相同
数组乘法	$A.*B$	A 和 B 的元素逐个对应相乘.两数组之间必须有相同的形,或其中一个是标量.
矩阵乘法	$A*B$	A 和 B 的矩阵乘法.A 的列数必须和 B 的行数相同.
数组右除法	$A./B$	A 和 B 的元素逐个对应相除: $A(i,j)/B(i,j)$ 两数组之间必须有相同的形,或其中一个是标量.
数组左除法	$A.\backslash B$	A 和 B 的元素逐个对应相除: $B(i,j)/A(i,j)$ 两数组之间必须有相同的形,或其中一个是标量.
矩阵右除法	A/B	矩阵除法,等价于 $A*inv(B)$, $inv(B)$ 是 B 的逆阵
矩阵左除法	$A\backslash B$	矩阵除法,等价于 $inv(B)*A$, $inv(A)$ 是 A 的逆阵
数组指数运算	$A.^B$	AB 中的元素逐个进行如下运算 $A(i,j)^B(i,j)$, $A(i,j)/B(i,j)$ 两数组之间必须有相同的形,或其中一个是标量.

5.matlab的内建函数

(1)选择性结果

max函数可以返回两个参数，第一个参数是输入的向量中的最大值，第二参数是输入向量中的最大值在向量中的位置。（当然，也可以选择只返回一个最大值）

```
1  >> maxval=max([1 -5 7 3])
2
3  maxval =
4
5      7
6
7  >> [maxval2 index]=max([1 -5 7 3])
8
9  maxval2 =
10
11      7
12
13
14  index =
15
16      3
17
```

min函数也有这样的功能：

```
1  >> minval=min([1 -5 7 3])
2
3  minval =
4
```



```
5      -5
6
7  >> [minval2 index2]=min([1 -5 7 3])
8
9  minval2 =
10
11      -5
12
13
14  index2 =
15
16       2
17
```

(2)带数组输入的matlab函数的应用

如果一个函数的输入是一个数组，那么这个函数的输出也是一个数组。

```
1  >> x=[1,2,3,4,5,6,7,8,9,10];
2  >> y=sin(x)
3
4  y =
5
6      0.8415      0.9093      0.1411     -0.7568     -0.9589     -0.2794      0.6570
7      0.9894      0.4121     -0.5440
```

(3)matlab常用的函数

表 2.8 常见的 MATLAB 函数

函数	描述
数学函数	
abs(x)	计算 x 的绝对值
acos(x)	计算 x 的反余弦函数

angle(x)	计算复数 x 的幅角
asin(x)	计算 x 的反正弦函数值
atan(x)	计算 x 的反正切函数值
atan2(y,x)	$\tan^{-2}(y/x)$
cos(x)	cosx
exp(x)	e^x
log(x)	$\log_e x$
[value,index]=max(x)	返回 x 中的最大值, 和它所处的位置
[value,index]=min(x)	返回 x 中的最小值, 和它所处的位置
mod(x,y)	余数
sin(x)	sinx
sqrt(x)	x 的平方根
tan(x)	tanx
rounding(取整)函数	
ceil(x)	
fix(x)	
round(x)	
字符转换函数	
char(x)	将矩阵中的数转化为字符,矩阵中的元素就不大于 127
double(x)	将字符串转化为矩阵
int2str(x)	将整数 x 转化为字符串形式
num2str(x)	将带小数点的数转化为一个字符型数组
str2num(x)	将字符串转化为数

三、matlab画图

1. 画图入门

(1)最简单的xy图像

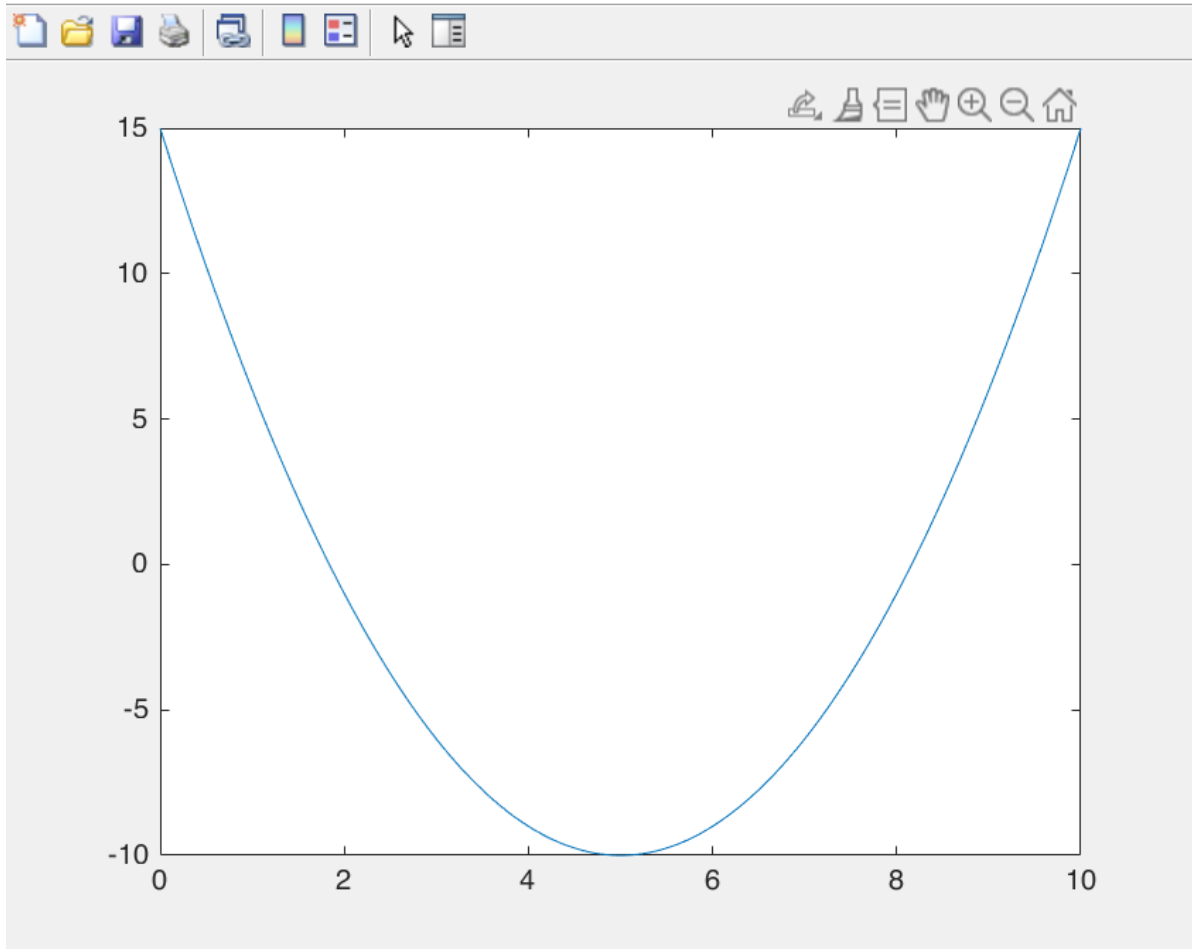
画一个数据图, 首先要创建两个向量, 由x,y构成, 然后使用plot函数。

```

1  >> x=0:0.05:10;
2  >> y=x.^2-10*x+15;
3  >> plot(x,y);

```

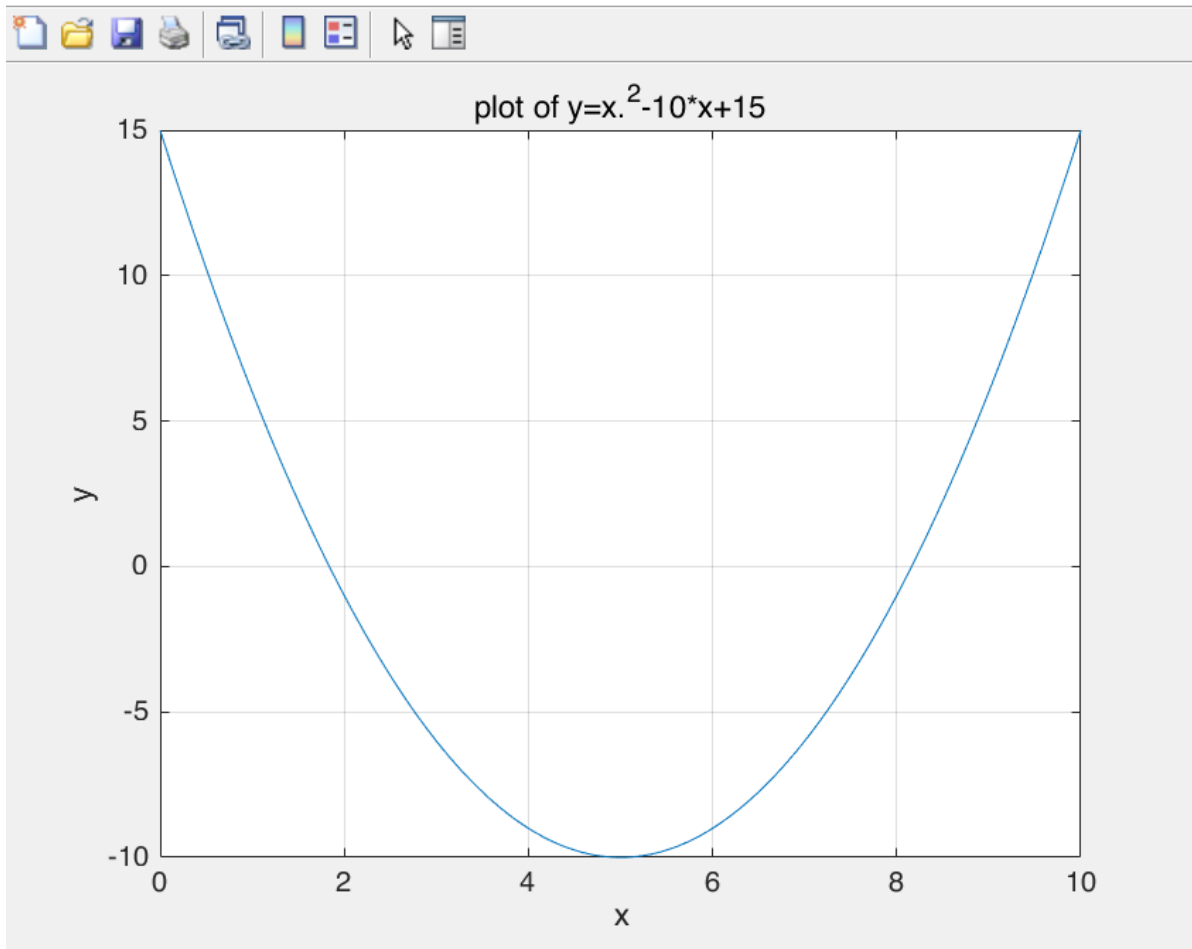
注意这里的乘方务必用数组计算方法! 必须要加一个小点“.”!



(2)标题、坐标轴标签和网格线

```
1 title('') %添加标题，在单引号中输入标题
2 xlabel('') %添加x轴轴标签
3 ylabel('') %添加y轴轴标签
4 grid on %在图像中出现网格线
5 grid off %在图像中去除网格线
```

```
1 >> x=0:0.01:10;
2 >> y=x.^2-10*x+15;
3 >> plot(x,y);
4 >> title('plot of y=x.^2-10*x+15');
5 >> xlabel('x');
6 >> ylabel('y');
7 >> grid on;
```



(3)打印图像

在图像窗口选择“另存为”即可，可以选择存储为.tif .png .svg .jpg等多种文件。

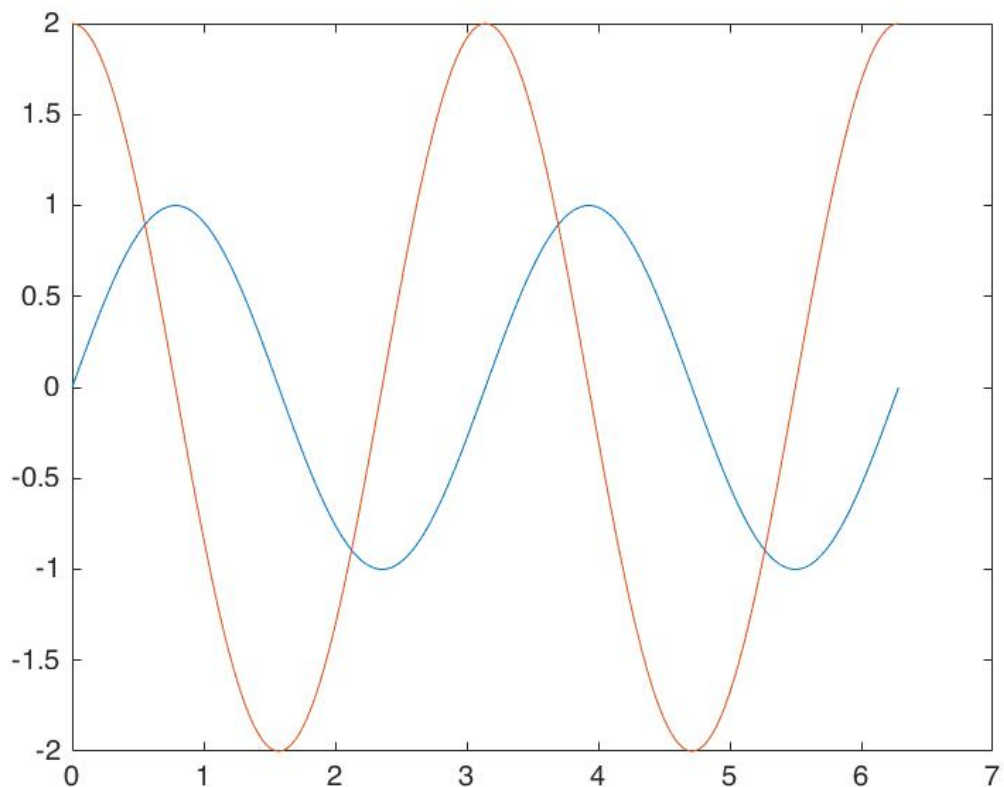
(4)联合作图

在同一个坐标系中做出多个函数的图像：

使用plot函数，必须将一系列的x值和每一个函数对应的y值相对应。

注意plot中使用逗号分开，不必用分号。

```
1 >> x=0:pi/100:2*pi;  
2 >> y1=sin(2*x);  
3 >> y2=2*cos(2*x);  
4 >> plot(x,y1,x,y2);
```



(5)线的颜色、形式，符号形式

在x,y向量参数后带有相关属性的字符串的plot函数，可以选择轨迹颜色和形式和符号类型，主要包括三类：

- 轨迹的颜色
- 符号的类型（线上加圆圈、三角等）
- 线的类型（实线虚线、点画线等）

表 2.9 图象的颜色，标记（符号）类型，线型

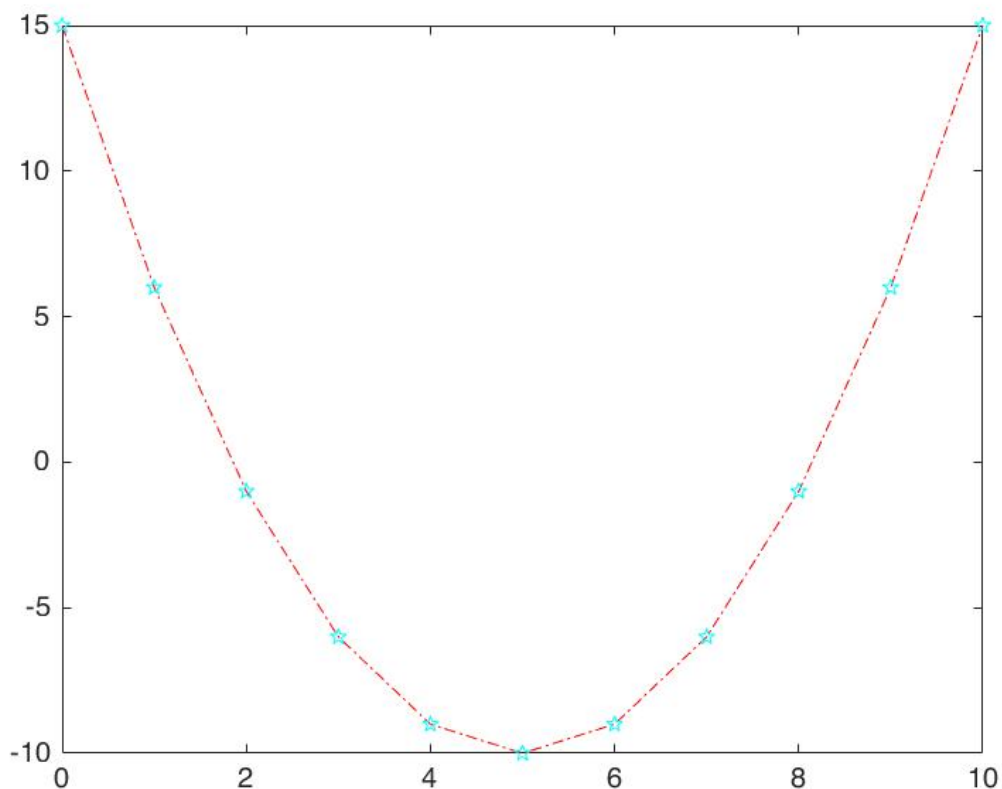
	颜色		标记类型		线型
y	黄色	.	点	-	实线
m	品红色	o	圈	:	点线
c	青绿色	x	×号	-.	画点线
r	红色	s	正方形	--	虚线
g	绿色	d	菱形	<none>	无
b	蓝色	v	倒三角		
w	白色	^	正三角		
k	黑色	>	三角(向右)		
		<	三角(向左)		
		p	五角星		
		h	六线形		
		<none>	无		

下面给出一个例子，注意plot的语法！

```

1  >> x = 0:1:10;
2  >> y = x.^2 -10.*x +15;
3  >> plot(x,y, 'r-.',x,y, 'cp');

```



(6)图例

用legend制作图例。其基本形式为：

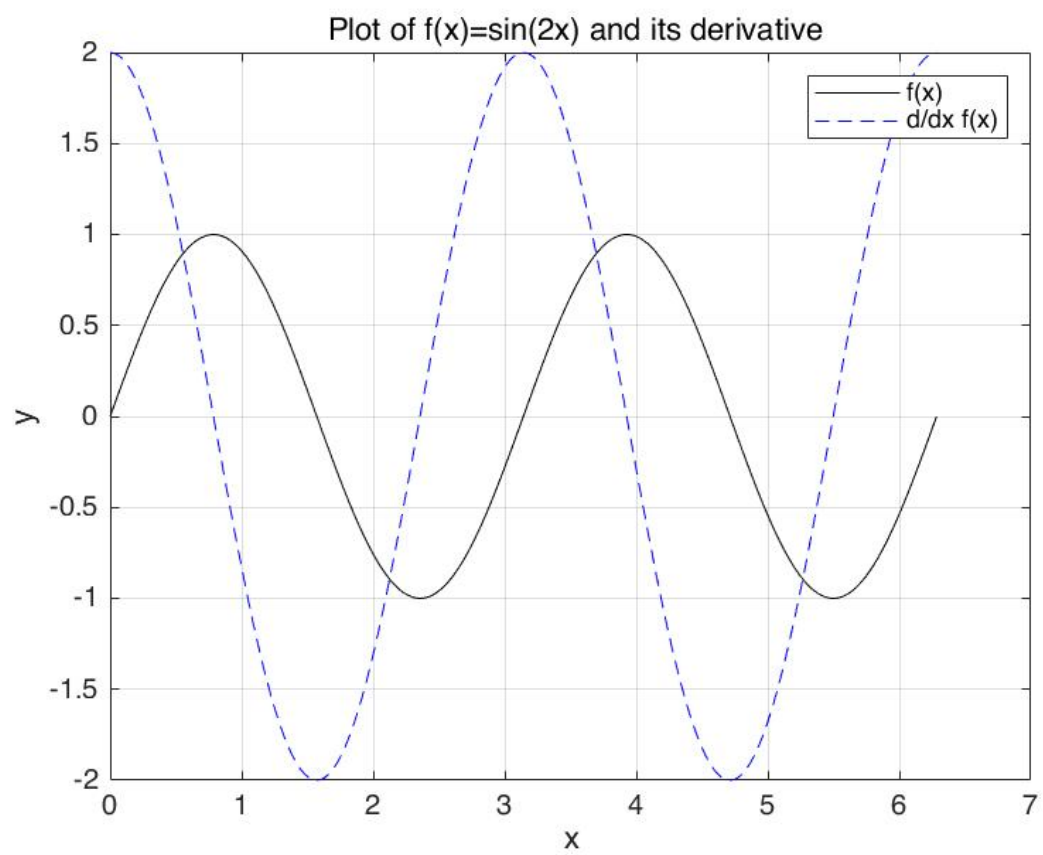
```
1 legend('string1','string2',...,'Location','pos')
```

其中string1, string2等是轨迹标签名，Location是一个必须写上去的（具体意义不明），pos是一个字符串，用于指定图例的位置。

```
1 legend off %去除多余的图例
```

下面给出一个完整的xy图像的例子，注意legend的语法。不写参数pos相当于默认在右上角。

```
1 x=0:pi/100:2*pi;
2 y1=sin(2*x);
3 y2=2*cos(2*x);
4 plot(x,y1,'k-',x,y2,'b--');
5 title(' Plot of f(x)=sin(2x) and its derivative');
6 xlabel('x');
7 ylabel('y');
8 legend('f(x)', 'd/dx f(x)') %这里加不加分号都可以正常运行
9 grid on;
```



legend的位置选择, 'pos'字符串的取值:

设置	位置
'North'	inside plot box near top
'South'	inside bottom
'East'	inside right
'West'	inside left
'NorthEast'	inside top right (default for 2-D plots)
'NorthWest'	inside top left
'SouthEast'	inside bottom right
'SouthWest'	inside bottom left
'NorthOutside'	outside plot box near top
'SouthOutside'	outside bottom
'EastOutside'	outside right
'WestOutside'	outside left
'NorthEastOutside'	outside top right (default for 3-D plots)
'NorthWestOutside'	outside top left
'SouthEastOutside'	outside bottom right
'SouthWestOutside'	outside bottom left
'Best'	least conflict with data in plot
'BestOutside'	least unused space outside plot

(7)对数尺度

打印数据既可以对数尺度，也可以线性尺度，于是就有4种组合，每种组合都有各自的意义。

```

1 plot %x,y都是线性尺度
2 semilogx %x轴是对数尺度，y轴是线性尺度
3 semilogy %x轴是线性尺度，y轴是对数尺度
4 loglog %x,y都是对数尺度

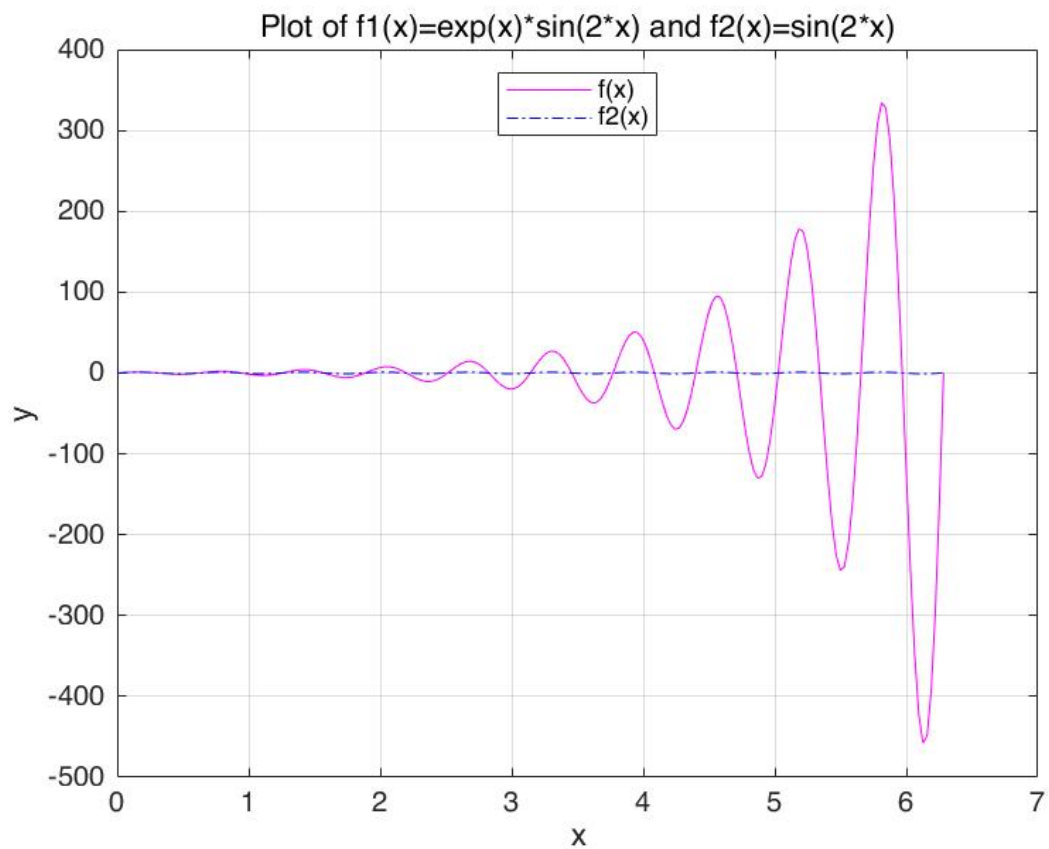
```

下面举出四种不同方式下的函数图像：

```

1 x=0:pi/100:2*pi;
2 y1=exp(x).*sin(10*x);
3 y2=sin(10*x);
4 plot(x,y1,'m-',x,y2,'b-.');
5 title(' Plot of f1(x)=exp(x)*sin(2*x) and f2(x)=sin(2*x) ');
6 xlabel('x');
7 ylabel('y');
8 legend('f(x)', 'f2(x)', 'Location', 'North');
9 grid on;

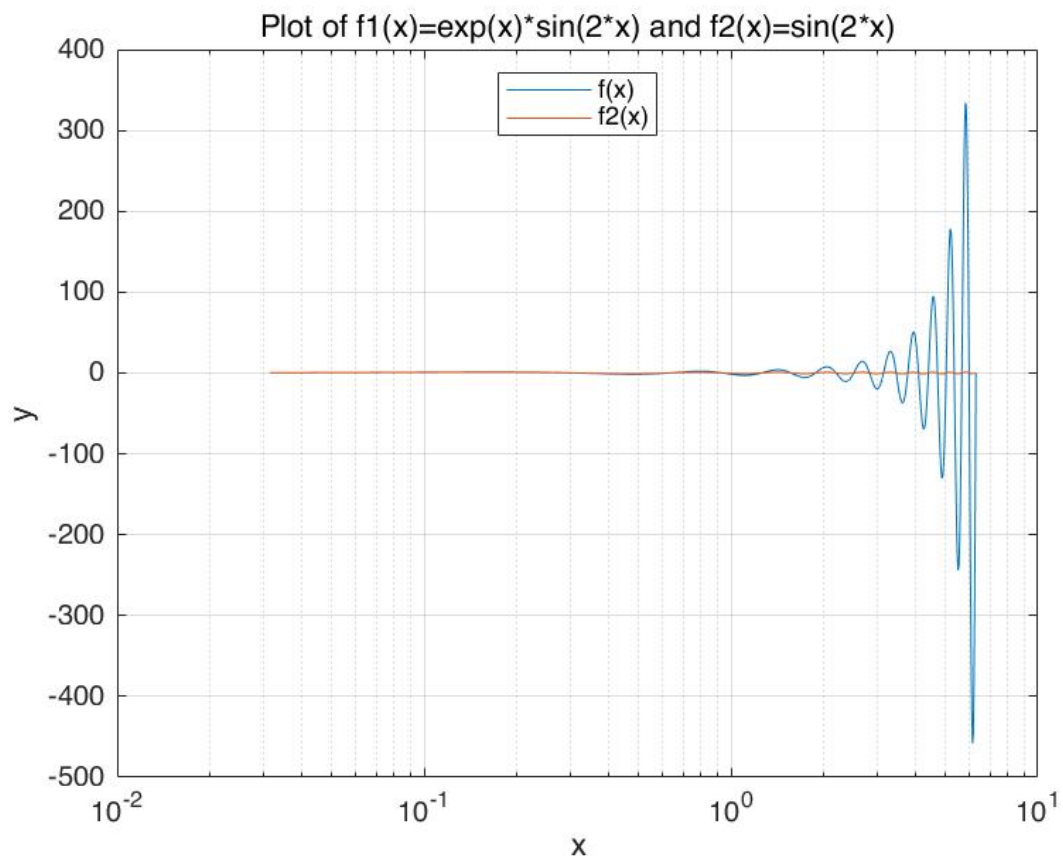
```

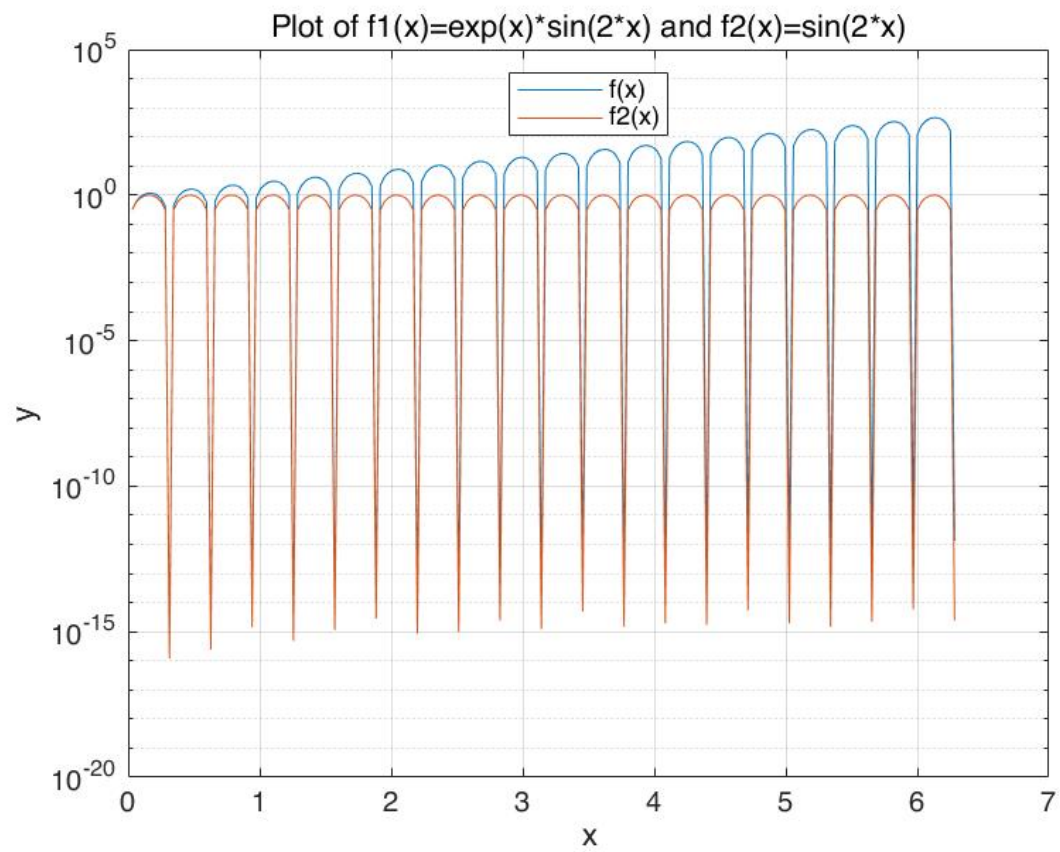
```

1 x=0:pi/100:2*pi;
2 y1=exp(x).*sin(10*x);
3 y2=sin(10*x);
4 semilogx(x,y1,x,y2);
5 title(' Plot of f1(x)=exp(x)*sin(2*x) and f2(x)=sin(2*x) ');
6 xlabel('x');
7 ylabel('y');
8 legend('f(x)', 'f2(x)', 'Location', 'North');
9 grid on;

```



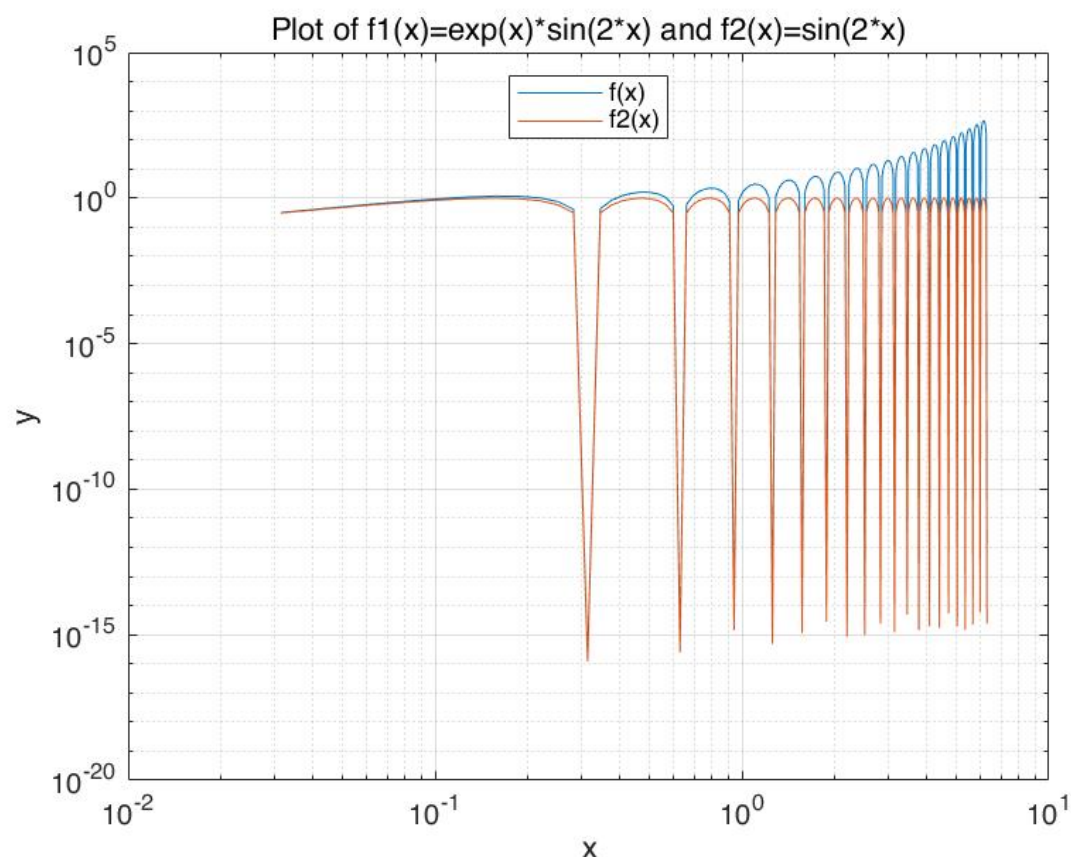
```
1 x=0:pi/100:2*pi;  
2 y1=exp(x).*abs(sin(10*x));  
3 y2=abs(sin(10*x));  
4 semilogy(x,y1,x,y2);  
5 title(' Plot of f1(x)=exp(x)*sin(2*x) and f2(x)=sin(2*x) ');  
6 xlabel('x');  
7 ylabel('y');  
8 legend('f(x)', 'f2(x)', 'Location', 'North');  
9 grid on;
```



```

1 x=0:pi/100:2*pi;
2 y1=exp(x).*abs(sin(10*x));
3 y2=abs(sin(10*x));
4 loglog(x,y1,x,y2);
5 title(' Plot of f1(x)=exp(x)*sin(2*pi*x) and f2(x)=sin(2*pi*x) ');
6 xlabel('x');
7 ylabel('y');
8 legend('f(x)', 'f2(x)', 'Location', 'North');
9 grid on;

```



2. 简单的二维图像

(1) 控制x,y轴的上下限

利用axis命令/函数可以实现设定和修改坐标的上下限。

通常预设值是范围足够显示输入值的每一个点。

表 3.5axis 函数/命令的形式

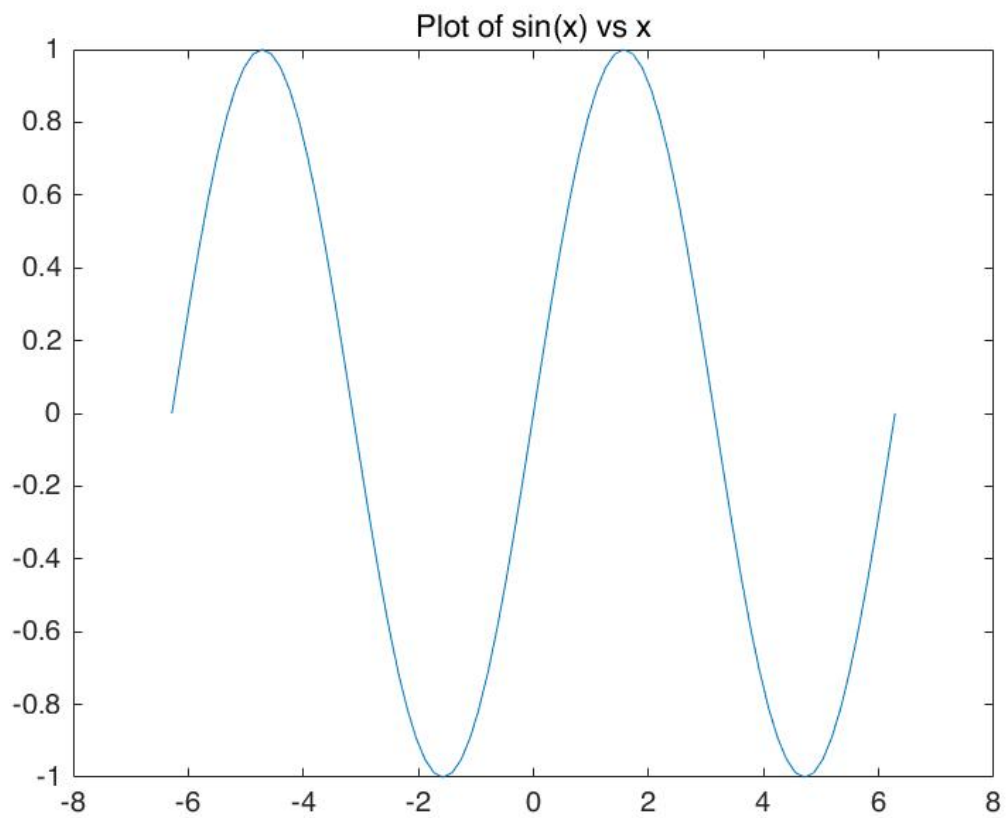
命令	功能
<code>v=axis</code>	此函数将会返回一个 4 元素行向量[xmin xmax ymin ymax]，其中 xmin xmax ymin ymax 代表 x, y 轴的上下限
<code>axis([xmin xmax ymin ymax])</code>	xmin xmax 设定横轴的下限及上限， ymin ymax 设定纵轴的下限及上限
<code>axis equal</code>	将横轴纵轴的尺度比例设成相同值
<code>axis square</code>	横轴及纵轴比例是 1:1
<code>axis normal</code>	以预设值画纵轴及横轴
<code>axis off</code>	将纵轴及横轴取消
<code>axis on</code>	这个命令打开所有的轴标签，核对符号，背景(默认情形)

matlab无论遇到什么命令，都会转化为一个函数进行操作。只有带有字符参数的函数才能当作命令，带有数字参数的函数只能被当作函数。所以axis有时当作命令，有时候当作函数。

```

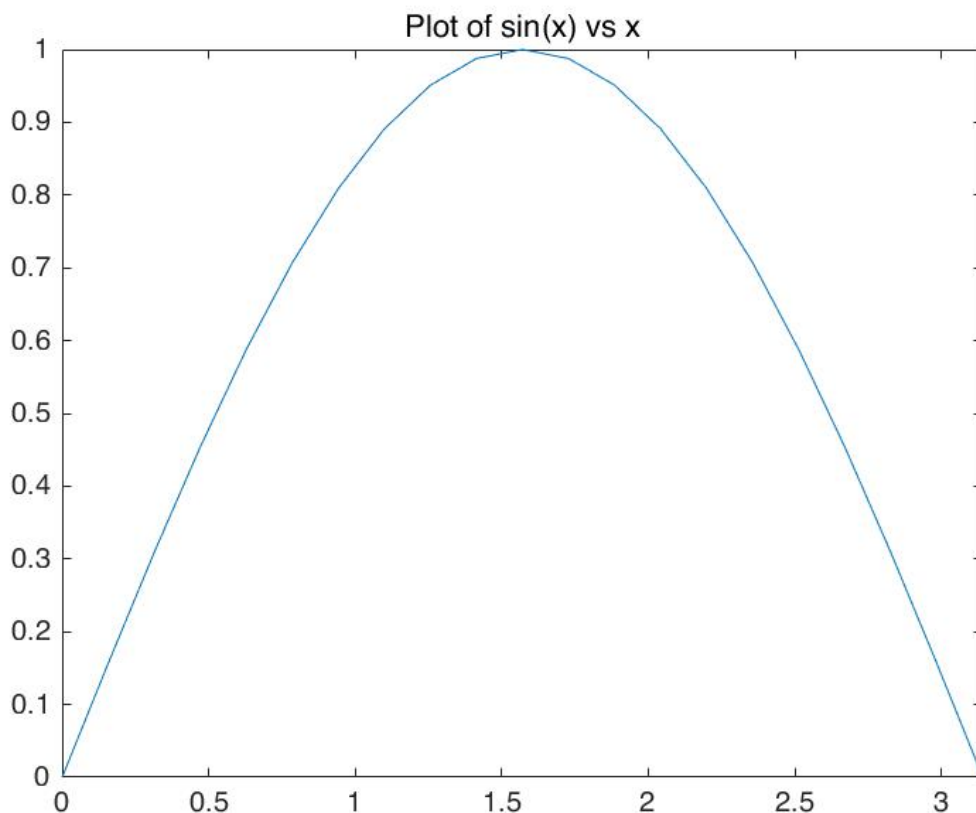
1 x=-2*pi:pi/20:2*pi;
2 y=sin(x);
3 plot(x,y);
4 title('Plot of sin(x) vs x');

```



当限定范围后:

```
1 x=-2*pi:pi/20:2*pi;  
2 y=sin(x);  
3 plot(x,y);  
4 title('Plot of sin(x) vs x');  
5 axis([0 pi 0 1]);
```



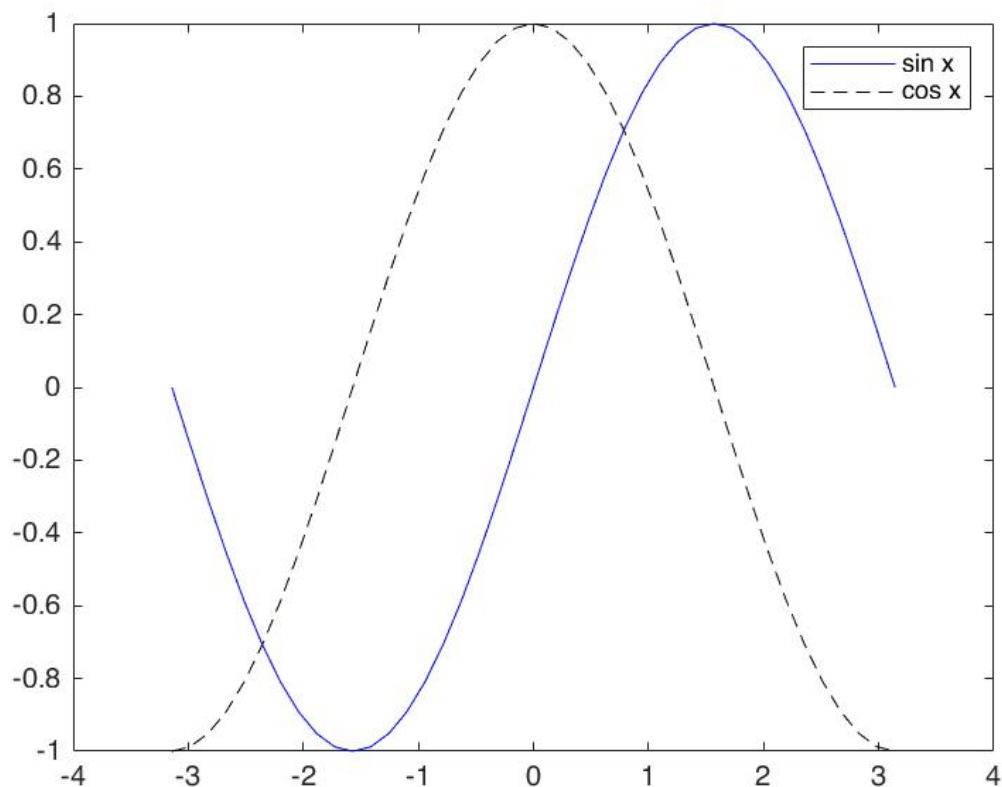
(2) 在同一坐标系内画多个图像

一般情况下，创建一个新的图像就要用到一个plot命令，**前面的数据就会自动消失**。可以采用hold命令加以修改。

- 1 `hold on` %所有新的图像都会叠加在原来存在的图像上
- 2 `hold off` %可以恢复默认情况，用新的图像来替代原来的图像

例如，在同一坐标系种画出sin x和cos x的图像

```
1 x = -pi:pi/20:pi;  
2 y1 = sin(x);  
3 y2 = cos(x);  
4 plot(x,y1,'b-');  
5 hold on;  
6 plot(x,y2,'k--');  
7 hold off;  
8 legend('sin x','cos x','Location','Northeast');
```



没有hold on语句则只有sin x的图像，没有cos x的图像；没有hold off对图像没有影响，但是可能会对之后的操作造成干扰。

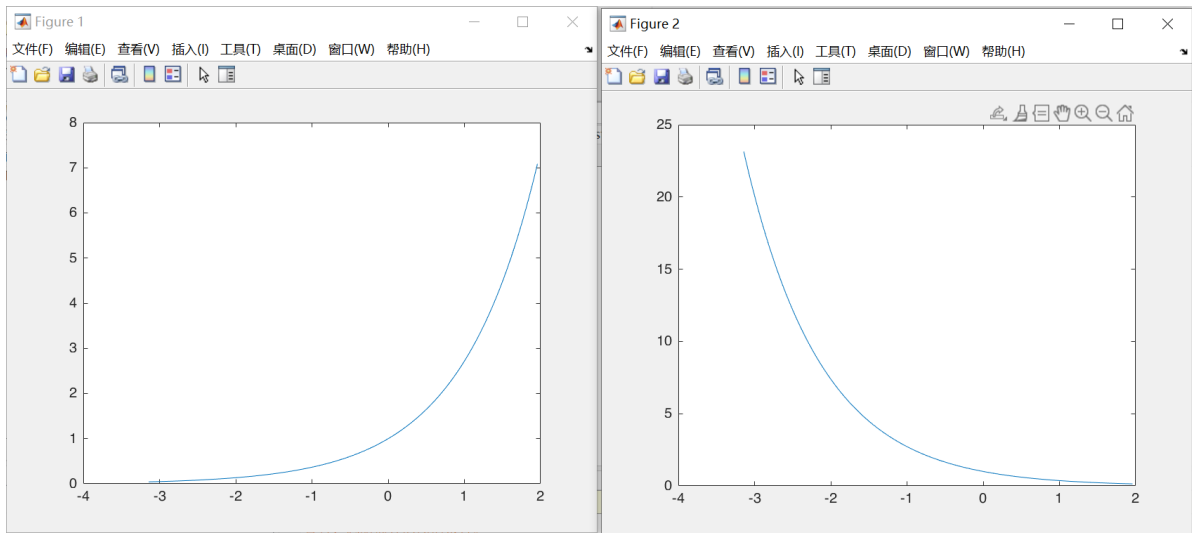
(3) 创建多个图像窗口

matlab可以创建多个图像窗口，每个窗口都有不同的数据。用图象数（整数）加以区分。这些窗口中的一个成为当前图像窗口，所有新的图画命令都将会展示在那个窗口中。

我们用**figure**函数来选择当前窗口。函数形式为**figure(n)**，其中n代表图象数。当这个函数被执行时，图n为成为当前图象，执行所有的画图命令。窗口不存在时，matlab将自动创建。

gef函数用于返回当前图象数。

```
1 figure(1);
2 x = x:0.05:2;
3 y1 = exp(x);
4 plot(x,y1);
5 figure(2);
6 y2 = exp(-x);
7 plot(x,y2);
```



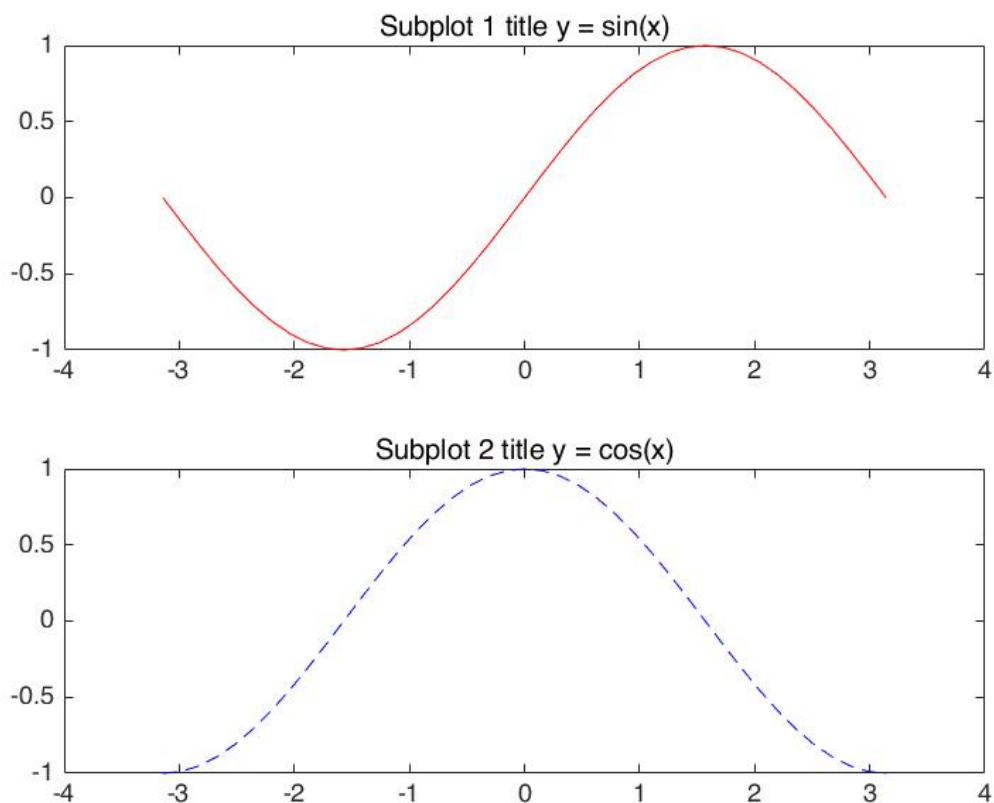
(4) 子图象

在一个图像窗口中有一系列的坐标系，创建出多个子图象。

```
1 subplot(m,n,p) %m行，n列，第p个图象
```

这个命令在当前图象窗口创建了 $m \times n$ 个子图象，按照 m 行、 n 列排列，并选择子图象 p 来接受当前所有画图命令。这些子图象按照从左到右、从上到下编号。如果subplot命令创建的新坐标系和原来的坐标系相冲突，那么原来的坐标系会被自动删除。

```
1 figure(1);
2 subplot(2,1,1);
3 x = -pi:pi/20:pi;
4 y = sin(x);
5 plot(x,y,'r-');
6 title('Subplot 1 title y = sin(x)');
7 subplot(2,1,2);
8 x = -pi:pi/20:pi;
9 y = cos(x);
10 plot(x,y,'b--');
11 title('Subplot 2 title y = cos(x)');
```

(5)对画线的增强控制

设置线宽、标识颜色、标识填充色、标识的大小。

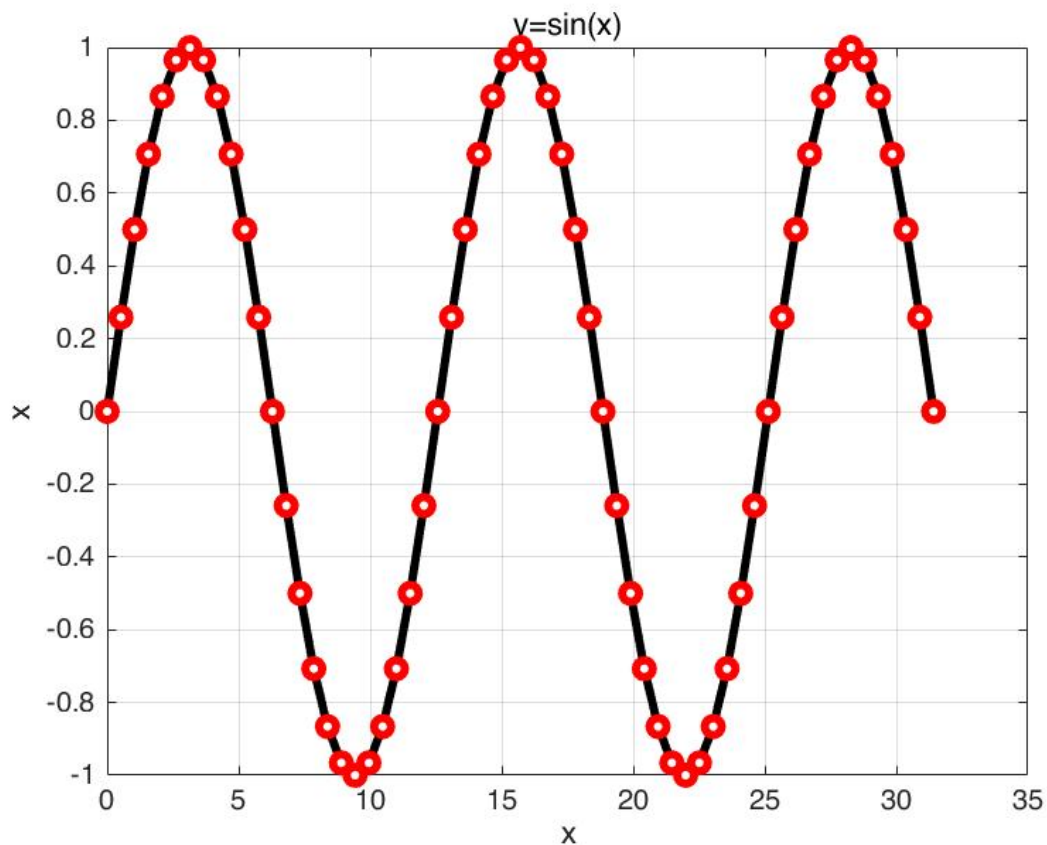
属性	说明
LineWidth	用来指定线的宽度
MarkerEdgeColor	用来指定标识表面的颜色
MarkerFaceColor	填充标识的颜色
MarkerSize	指定标识的大小

都可以在 `plot` 命令中调节：

```
1 | plot(x,y,'PropertyName',value,...)
```

画出一个图象，轨迹的宽度为3，颜色为黑色，圆圈标识的宽度为6，每个标识为红色边缘和绿色内核。

```
1 | x=0:pi/6:10*pi;
2 | y=sin(x/2);
3 | plot(x,y,'k-
   | ',x,y,'wo','LineWidth',3,'MarkerSize',6,'MarkerEdgeColor','r','MarkerFaceColor','g','w');
4 | title('y=sin(x)');
5 | xlabel('x');
6 | ylabel('y');
7 | grid on;
```



(6) 文本字符串的高级控制

使用的时候貌似容易出现一些问题。

画图中可能会用到文本字符串（标签、坐标轴标签），可以用黑体、斜体来格式化，也包括特殊的希腊字母或数字符号。

文本的字体可以通过**stream modifiers**修改。一个stream modifiers是一个特殊的字符序列，用于告诉编译器改变它的行为。

常用的是：

stream modifiers	含义
\bf	黑体
\it	斜体
\rm	恢复正常字体
\fontname	字体的名字
\fontsize	字体的大小
_{xxx}	xxx作为某字符的上标
^{\xxx}	xxx作为某字符的下标

一旦一个stream modifier插入到一个文本字符串中，它将持续发挥作用，知道这个字符串结束或者消失。如果一个modifier后再跟着一个{}，那么只对{}中的文本起作用。

如果需要插入特殊的希腊字母或者数字符号，可以参考latex转义序列。即matlab中的这些语法和latex是一样的。

(7)极坐标图象

！！注意！！matlab官网推荐的是使用 `polarplot` 函数在极坐标中绘制线条。

```
1 polarplot(theta,rho) %theta是辐角，rho是长度
2 polarplot(theta1,rho1,theta2,rho2,...) %绘制多个对组
```

函数 `polar` 用于在极坐标系中画图。

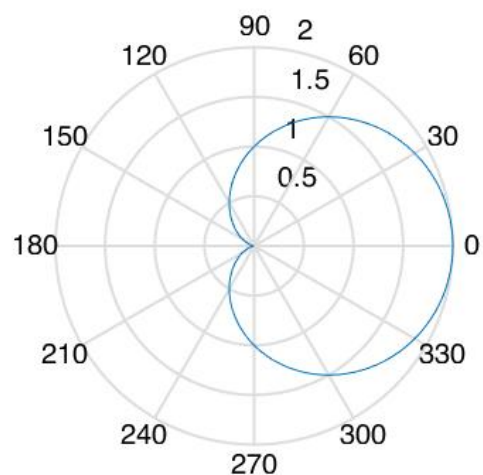
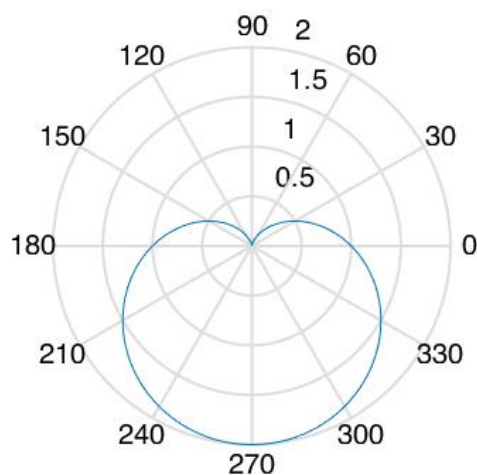
```
1 polar(theta,r)
```

`theta`代表一个弧度角数组，`r`代表一个距离数组。

它用来画以角度为自变量的函数的极坐标图是很有用的。

例如用`polar`函数在matlab中画出心型曲线。

```
1 t = 0:pi/100:2*pi;
2 r = 1-sin(t);
3 subplot(1,2,1);
4 polar(t, r);
5 subplot(1,2,2);
6 t1 = t-pi/2;
7 r1 = 1-sin(t1);
8 polar(t, r1);
```



3. 二维作图的补充说明

(1) 二维作图的附加类型

除了前面提到的简单的xy图象，还有针头图(stem plots)、阶梯图(stair plots)、条形图、饼图(pie plots)、罗盘图(compass plots)。

函数	描述
bar(x, y)	这个函数用于创建一个水平的条形图，x 代表第一个 X 轴的取值，y 代表对应于 Y 的取值
barh(x, y)	这个函数用于创建一个竖直的条形图，x 代表第一个 X 轴的取值，y 代表对应于 Y 的取值
compass(x, y)	这个函数用于创建一个极坐标图，它的每一个值都用箭头表示，从原点指向 (x, y)，注意：(x, y) 是直角坐标系中的坐标。
pie(x) pie(x, explode)	这个函数用来创建一个饼状图，x 代表占总数的百分数。explode 用来判断是否还有剩余的百分数
stairs(x, y)	用来创建一个阶梯图，每一个阶梯的中心为点(x, y)
stem(x, y)	这个函数可以创建一个针头图，它的取值为(x,y)

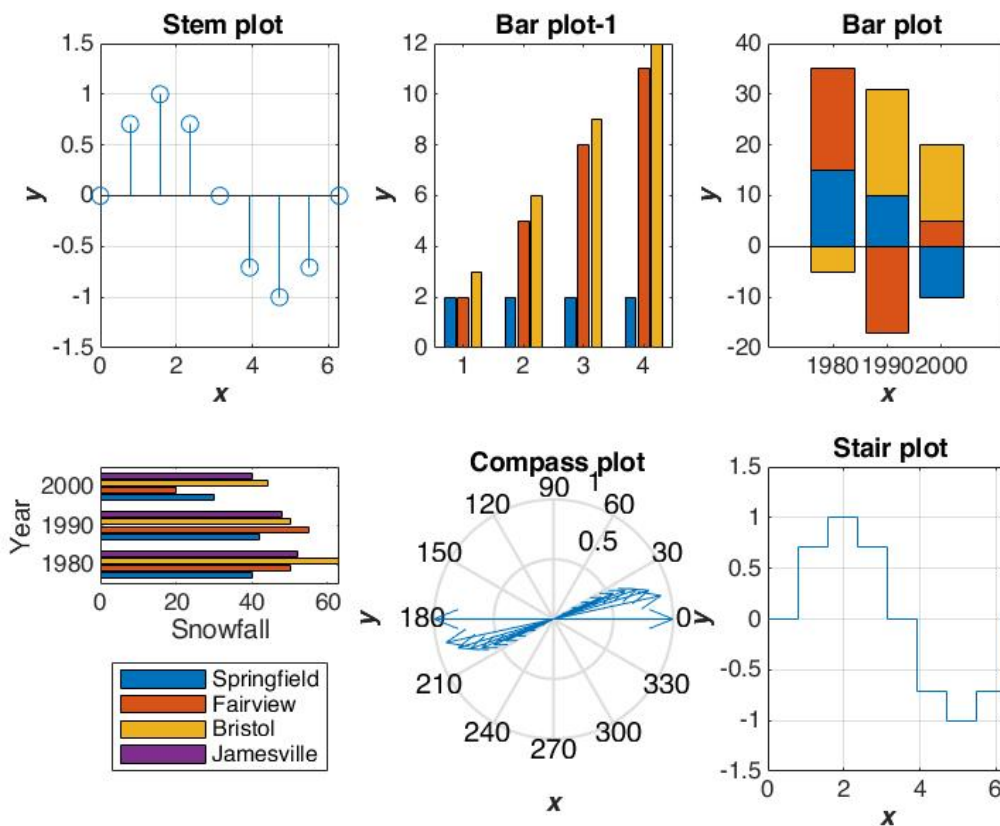
针头图、条形图、罗盘图、阶梯图都可以参考下面的代码（用子图的形式，分别做出几个图象）：

```
1 subplot(2,3,1);
2 x = 0:pi/4:2*pi;
3 y = sin(x);
4 stem(x,y);
5 title('\bf Stem plot');
6 xlabel('\bf\itx');
7 ylabel('\bf\ity');
8 axis([0 2*pi -1.5 1.5]);
9 grid on;
10
11 subplot(2,3,2);
12 y = [2 2 3; 2 5 6; 2 8 9; 2 11 12];
13 bar(y);
14 title('\bf Bar plot-1');
15 ylabel('\bf\ity');
16
17 subplot(2,3,3);
18 x = [1980 1990 2000];
19 y = [15 20 -5; 10 -17 21; -10 5 15];
20 bar(x,y, 'stacked'); %可以实现堆叠
21 title('\bf Bar plot');
22 xlabel('\bf\itx');
23 ylabel('\bf\ity');
24
25 subplot(2,3,4);
26 x = [1980 1990 2000];
27 y = [40 50 63 52; 42 55 50 48; 30 20 44 40];
28 barh(x,y)
29 xlabel('Snowfall')
30 ylabel('Year')
31 legend({'Springfield', 'Fairview', 'Bristol', 'Jamesville'}, 'Location', 'Southou
tside')
32
33 subplot(2,3,5);
34 x = -1:0.1:1;
```

```

35 y = ( x / 2 ) .* sqrt(1-x.^2);
36 compass(x,y);
37 title('\bf Compass plot');
38 xlabel('\bf\itx');
39 ylabel('\bf\ity');
40
41 subplot(2,3,6);
42 x = 0:pi/4:2*pi;
43 y = sin(x);
44 stairs(x,y);
45 title('\bf Stair plot');
46 xlabel('\bf\itx');
47 ylabel('\bf\ity');
48 axis([0 2*pi -1.5 1.5]);
49 grid on;

```



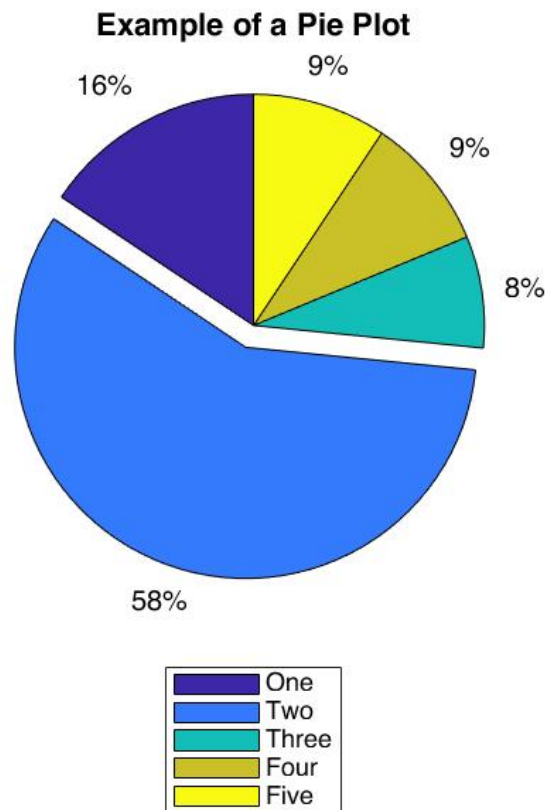
饼状图和前面的图有一点点不同。把数组x传递给函数pie，函数计算出每个元素占全部元素和的百分比。

pie 也支持选择性参数 `explode`，它是一个逻辑数组，包含元素1和0。当 `explode=1`时，它对应的扇区从整体中分离出来。

```

1 data = [10 37 5 6 6];
2 explode = [0 1 0 0 0];
3 pie(data, explode);
4 title('\bfExample of a Pie Plot');
5 legend('One', 'Two', 'Three', 'Four', 'Five', 'Location', 'SouthOutside');

```



(2) 作图函数

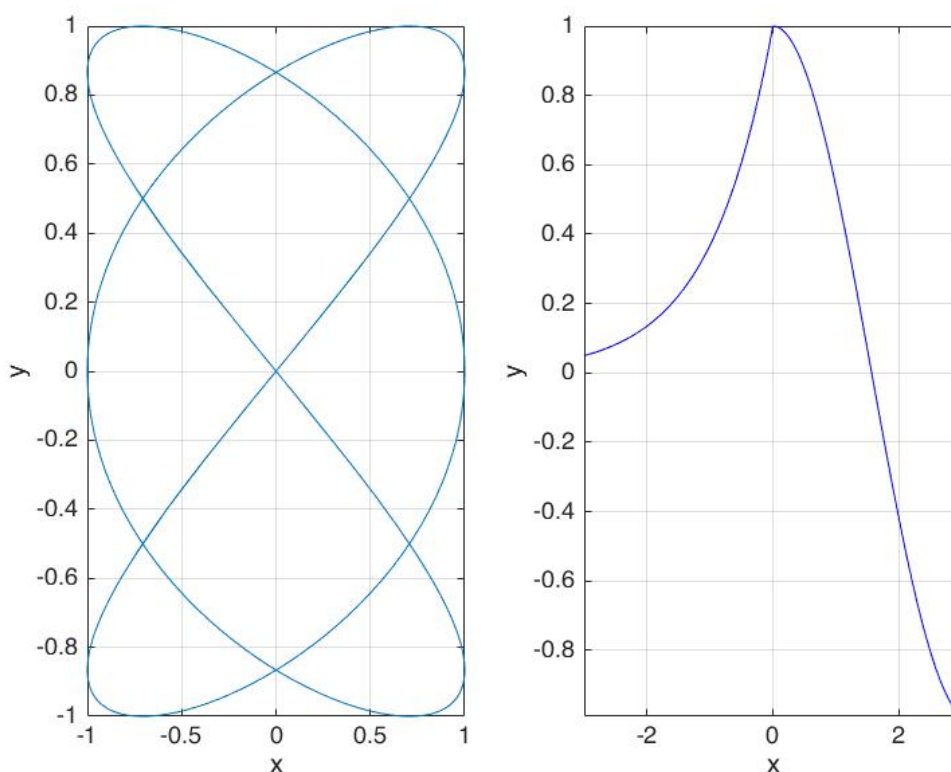
前面提到的作图，都必须要先创建数组。

直接做出图象，而不需要创建中间数据数组的函数：`ezplot`、`fplot`

!!! 注意：matlab官网中更为推荐 `fplot` 函数!!!

```
1 fplot(f) % 在默认区间[-5,5]（对于x）绘制由函数y=f(x)定义的曲线
2 fplot(f,xinterval) % 在指定区间绘图：xinterval是区间，应当指定为[xmin,xmax]形式的二元素向量
3 fplot(funx,funy) % 在默认区间[-5,5]（对于t）绘制由x=funx(t)和y=funy(t)这样的参数方程定义的曲线。
4 fplot(funx,funy,tinterval) % 将在指定区间绘图。将区间指定为 [tmin tmax] 形式的二元素向量。
```

```
1 subplot(1,2,1); % 参数方程的图象
2 xt = @(t) cos(3*t);
3 yt = @(t) sin(2*t);
4 fplot(xt,yt);
5 xlabel('x');
6 ylabel('y');
7 grid on;
8
9 subplot(1,2,2); % 分段函数
10 fplot(@(x) exp(x),[-3 0],'b');
11 hold on;
12 fplot(@(x) cos(x),[0 3],'b');
13 hold off;
14 xlabel('x');
15 ylabel('y');
16 grid on;
```



(3) 柱状图

标准的matlab柱状图函数为hist。

不推荐使用 `hist`。请改用 `histogram`。

直方图属于数值数据的条形图类型，将数据分组为 bin。创建 `Histogram` 对象后，可以通过更改直方图的属性值修改它的各个方面。这对快速修改 bin 属性或更改显示特别有用。

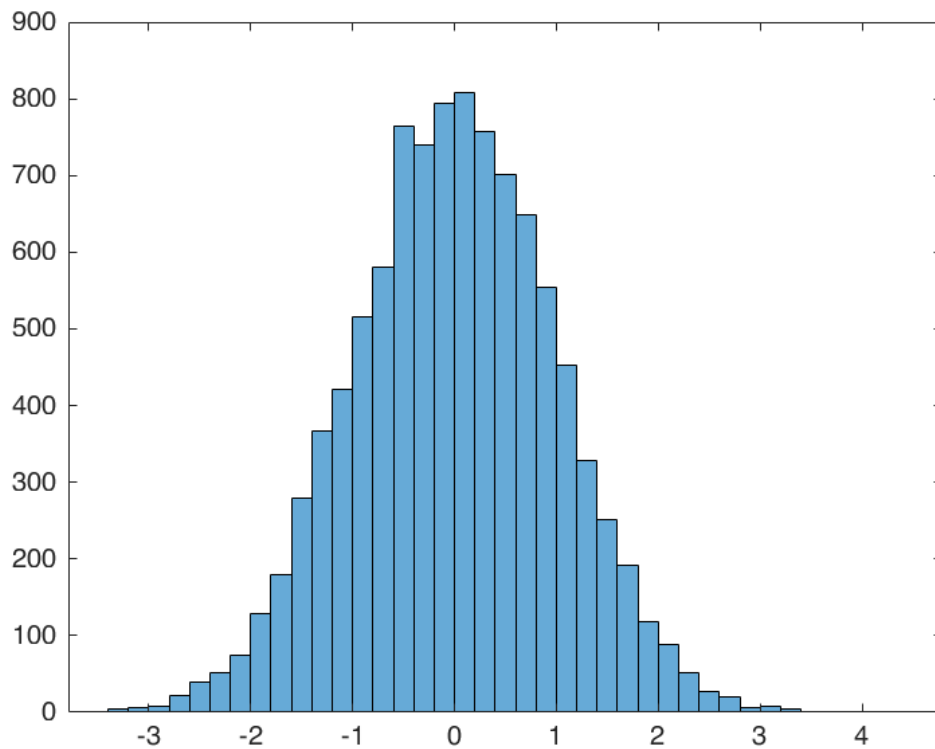
函数 `histogram` 的用法：

- 1 `histogram(X)` % 基于 `X` 创建直方图。`histogram` 函数使用自动 bin 划分算法，然后返回均匀宽度的 bin，这些 bin 可涵盖 `X` 中的元素范围并显示分布的基本形状。
- 2 % 参数 `X` 为要分布到各 bin 的数据，指定为向量、矩阵或多维数组。如果 `X` 不是向量，则 `histogram` 将它视作单列向量 `X(:)` 并绘制一个直方图。
- 3 `histogram(X,nbins)` % 创建以 `nbins` 为宽度的柱状图
- 4 % 使用标量 `nbins` 指定的 bin 数量。
- 5 % bin 数量，指定为正整数。如果不指定 `nbins`，则 `histogram` 基于 `X` 中的值自动计算将使用多少个 bin。示例：`histogram(X,15)` 创建一个带 15 个 bin 的直方图。
- 6

示例：

```
1 x = randn(10000,1);
2 h = histogram(x)
```

绘图：



```
1 h =  
2  
3 Histogram - 属性:  
4  
5     Data: [10000x1 double]  
6     Values: [1x39 double]  
7     NumBins: 39  
8     BinEdges: [1x40 double]  
9     BinWidth: 0.2000  
10    BinLimits: [-3.4000 4.4000]  
11    Normalization: 'count'  
12    FaceColor: 'auto'  
13    EdgeColor: [0 0 0]
```

4. 三维作图

一般情况下，三维图象常用于显示以下两类数据。

1. 两个变量是同一自变量的函数，当你希望显示自变量重要性时，你可以用三维作图表示。
2. 一个变量是另外两个变量的函数。

(1) 三维曲线作图

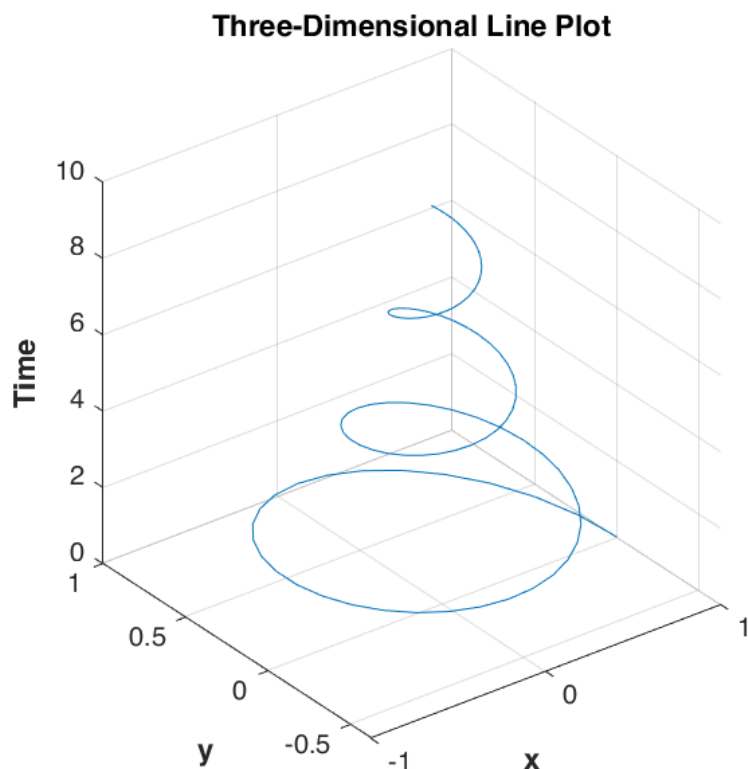
三维曲线作图最简单的函数是：

```
1 plot(x,y,z);
```

示例：


```
1 t = 0:0.1:10;
2 x = exp(-0.2*t) .* cos(2*t);
3 y = exp(-0.2*t) .* sin(2*t);
4 plot3(x,y,t);
5 title('\bfThree-Dimensional Line Plot');
6 xlabel('\bfx');
7 ylabel('\bfy');
8 zlabel('\bfTime');
9 axis square;
10 grid on;
```

图像：



(2) 三维表面，网格，等高线图像

表面，网格，等高线图像是非常简单的方法来表示两变量的函数。

数组x, y, z分别表示每个点的坐标值，函数分别如下：

表 6.4 表面，网格，等高线图像函数

函数	描述
mesh(x, y, z)	这个函数创建一个三维网格图像。数组 x 包括要画得每一点的 x 值，数组 y 包括要画得每一点的 y 值，数组 z 包括要画得每一点的 z 值。
surf(x, y, z)	这个函数创建一个三维表面图像。x, y, z 代表的意义与上式相同。
contour(x, y, z)	这个函数创建一个三维等高线图像。x, y, z 代表的意义与上式相同。

counter 函数显示的是x,y二维平面的等高线显示。

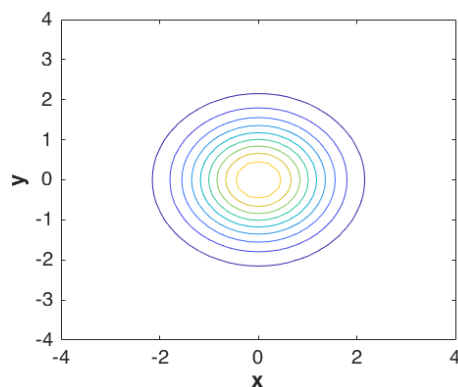
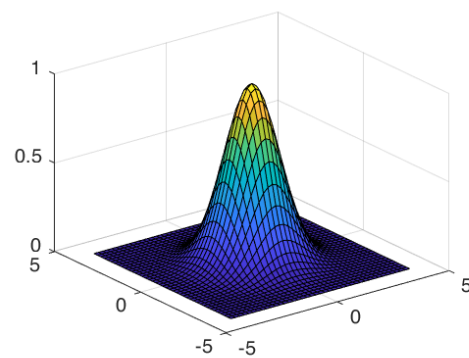
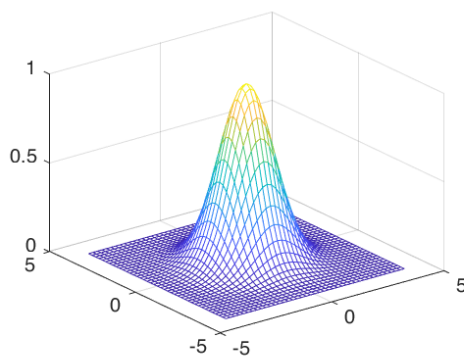
matlab中 meshgrid 函数使得函数图像数组x,y的创建十分容易。

- 1 `[X,Y] = meshgrid(x,y)` % 基于向量 `x` 和 `y` 中包含的坐标返回二维网格坐标。
- 2 % `x` 是一个矩阵，每一行是 `x` 的一个副本；`Y` 也是一个矩阵，每一列是 `y` 的一个副本。坐标 `x` 和 `Y` 表示的网格有 `length(y)` 个行和 `length(x)` 个列。

示例：

```
1 [x,y] = meshgrid(-4:0.2:4,-4:0.2:4);
2 z = exp(-0.5*(x.^2+y.^2));
3 subplot(2,2,1);
4 mesh(x,y,z);
5 subplot(2,2,2);
6 surf(x,y,z);
7 subplot(2,2,3);
8 contour(x,y,z);
9 xlabel('\bf{x}');
10 ylabel('\bf{y}');
11 zlabel('\bf{z}');
```

函数图像：



作业总结

1. 点电荷的电位分布

(1) 函数总结

meshgrid——二维或三维网格，取坐标

[meshgrid 的语法](#)

- 1 `[X,Y] = meshgrid(x,y)` %基于向量 `x` 和 `y` 中包含的坐标返回二维网格坐标。`X` 是一个矩阵，每一行是 `x` 的一个副本；`Y` 也是一个矩阵，每一列是 `y` 的一个副本。坐标 `X` 和 `Y` 表示的网格有 `length(y)` 个行和 `length(x)` 个列。
- 2 `[X,Y,Z] = meshgrid(x,y,z)` %返回由向量 `x`、`y` 和 `z` 定义的三维网格坐标。`X`、`Y` 和 `Z` 表示的网格的大小为 `length(y)×length(x)×length(z)`。

可以利用向量`x`定义的`x`坐标和向量`y`定义的`y`坐标创建二维网格坐标`[X,Y]`。

然后在二维网格上计算关于`x,y`的表达式时，就可以直接用`X,Y`。

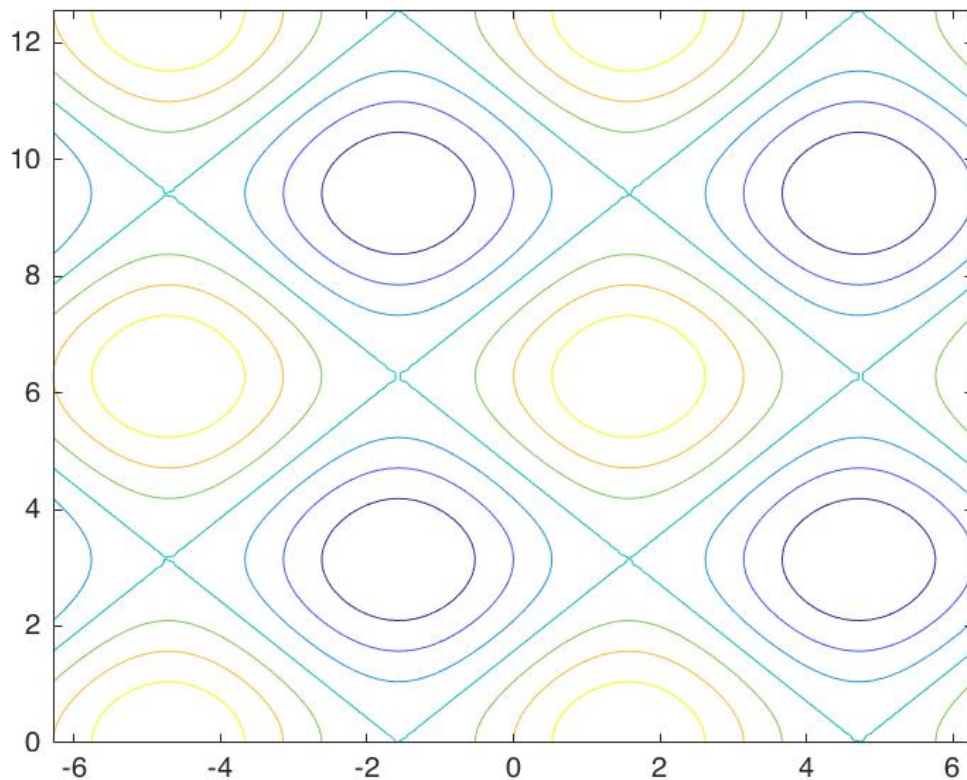
```
1  >> x = 1:3;
2  y = 1:5;
3  [X,Y] = meshgrid(x,y)
4
5  X =
6
7      1      2      3
8      1      2      3
9      1      2      3
10     1      2      3
11     1      2      3
12
13
14  Y =
15
16     1     1     1
17     2     2     2
18     3     3     3
19     4     4     4
20     5     5     5
21
22  >> X.^2 + Y.^2
23
24  ans =
25
26     2     5    10
27     5     8    13
28    10    13    18
29    17    20    25
30    26    29    34
```

contour——矩阵的等高线

[contour的语法](#)

- 1 `contour(Z)` %创建一个包含矩阵 `Z` 的等值线的等高线图，其中 `Z` 包含 `x-y` 平面上的高度值。
`MATLAB®` 会自动选择要显示的等高线。`Z` 的列和行索引分别是平面中的 `x` 和 `y` 坐标。
- 2 `contour(X,Y,Z)` %指定 `Z` 中各值的 `x` 和 `y` 坐标。
- 3 `contour(____,levels)` %将要显示的等高线指定为上述任一语法中的最后一个参数。将 `levels` 指定为标量值 `n`，以在 `n` 个自动选择的层级（高度）上显示等高线。要在某些特定高度绘制等高线，请将 `levels` 指定为单调递增值的向量。要在一个高度（`k`）绘制等高线，请将 `levels` 指定为二元素行向量 `[k k]`。
- 4 `contour(____,Name,value)` %使用一个或多个名称-值对组参数指定等高线图的其他选项。请在所有其他输入参数之后指定这些选项。有关属性列表，请参阅 `Contour` 属性。

```
1 x = linspace(-2*pi,2*pi);
2 y = linspace(0,4*pi);
3 [X,Y] = meshgrid(x,y);
4 Z = sin(X)+cos(Y);
5 contour(X,Y,Z)
```



linspace——生成线性间距向量

`linspace` 类似于冒号运算符“`:`”，但可以直接控制点数并始终包括端点。“`linspace`”名称中的“`lin`”指示生成线性间距值而不是同级函数 `logspace`，后者会生成对数间距值。

- 1 `y = linspace(x1,x2)` %返回包含 `x1` 和 `x2` 之间的 100 个等间距点的行向量。
- 2 `y = linspace(x1,x2,n)` %生成 `n` 个点。这些点的间距为 $(x2-x1)/(n-1)$ 。

(2) matlab代码

```
1 clear;
2 x=-100:100;
3 y=-100:100;
4 [X,Y]=meshgrid(x,y);
5 r1=sqrt((X+10.5).^2+Y.^2);
6 r2=sqrt((X-10.5).^2+Y.^2);
7 U=2./r1-1./r2;
8 u=-0.5:0.002:0.5;
9 figure(1);
10 contour(X,Y,U,u);
11 hold on;
12 plot(-10.5,0,'.','MarkerEdgeColor','r','MarkerSize',15);
13 plot(10.5,0,'.','MarkerEdgeColor','b','MarkerSize',15);
14 axis equal;
15 grid on;
16 title('正负点电荷电量比为2的等势线','FontSize',15,'FontName','msyh');
17 xlabel('x','FontSize',16);
18 ylabel('y','FontSize',16)
```

