

# 萝卜分享会：简单但强大的 MATLAB 数据处理方式

许婷婷 SA18168132，章坤 SA18168226

杜沈达 SA18168163，刘淼 SA18168215

文献管理课第 33 组

---

## 简介

本次萝卜分享会，我们组准备的是 MATLAB 数据可视化，主要介绍三大方面，第一是基础的 MATLAB 数据操作，包括读写和运算；第二是在 MATLAB 内进行画图，主要是介绍一些常用的绘图指令；第三是数据分析，主要介绍几个统计学工具箱的使用，如曲线拟合，线性回归分析。

---

## 一、数据获取

### 1.1 常用的数据读取

#### 从 Excel 表格读入

只需要一条命令

```
data=xlsread('path\filename.xls(x)','Range');
```

这句命令说的意思就是把路径 path 下面的 filename 文件告诉 MATLAB，要用读取 Excel 的方式 xlsread 读范围 Range 的数值。Range 一般都写成 A1: G15 这样字的形式。

当然，除了读取还有写入，如果你做好了一个 data 矩阵的数据要写到 Excel 里面，只需要使用下面的命令，就可以把 data 里面的数据写到 matlab 里面了。

```
xlswrite('path\filename.xls(x)',data);
```

#### txt 文本读入数据

有些软件处理之后的数据都是 txt 文本格式的，这时候就需要用文本读写指令来操作，读取也是一个指令完成。

```
data=load('path\filename.txt')
```

这样子就可以加载一个文本文件的数据了，如果计算完数据要将其导出为文本文件数据，稍微有点麻烦。需要下面的指令完成。

```
[fid,errmsg]=fopen('txtTest.txt','wt');  
filename=fopen(fid);
```

```
fprintf(fid, '%f\t%f\n', data);  
fclose(fid);
```

这个命令说的是先打开 txtTest.txt 的文本，用写命令打开，然后 fid 和 errmsg 是函数的输出参数，fid 是一个整数，代表文件打开的状态，好像-1 是表示不成功，这种都可以通过 help 去查看函数的具体用法，errmsg 是代表函数的出错的信息，然后就把这个打开的文件名称告诉 filename，这主要是在工作区可以看到文件夹名字，然后是核心指令 fprintf 把 data 输出到 fid 也就是文件里面，data 的数据分两列写入文件，用 tab 隔开，之后再关闭文件。其实这个东西有点复杂了，一般把数据输出到 Excel 就够了可以发给别人看了，所以这个写入 txt 我也很少用到。值得说明一下的是，MATLAB 的写函数的方式也就是这样子，其中 function 是关键字，[] 里面的的是输出的参数，FuncName 是自定义函数的名称，() 里面的的是输入的参数。

```
function [outPara1,outPara2,...]=FuncName(inPara1,inPara2,...)  
end
```

## 1.2 总结

主要就是讲了两个读取数据的方式，其中与 Excel 联合使用应该会比较常用并且实用的，命令也比较简单强大，txt 一般在一些软件处理完输出一些参数可能会存入文本文件，这时候也需要用到，但是没事把数据写到 txt 文件再给别人看我还没有做过这样的事情。☹

## 二、数据计算和可视化

数据计算就通过几个例子来说怎么做的吧，正好算完的数据可以用来画图，这也是很顺便的一件事情。

### 计算器计算

要求：计算  $12.5 + \sqrt{3^2 + 4^2} + \cos(\frac{\pi}{3}) + 3^5 + 2^3$

代码：

```
1 y=12.5*2+sqrt(3^2+4^2)+cos(pi/3)+3^5+power(2,3);
```

分析：上面的代码也就跟普通的计算器差不多，唯一要说明的是两个函数，一个是 sqrt(x)，一个是 power(x,y)。sqrt(x) 的 x 也就是要计算的表达式，把它写进括号也就是写进根号里面的意思，power(x,y) 代表的是  $x^y$ ，也就是跟普通的写  $x^y$  没什么区别，一般都用  $x^y$  这样的形式，在帮助文档中可以看到说  $C = \text{power}(A,B)$  是执行  $A.^B$  的替代方法，

但很少使用，所以一般用  $x^y$  就够了。它可以启用类的运算符重载。还有一点需要说明的是，在  $\sin, \cos$  等三角函数进行运算的时候，用的都是弧度制。

## 函数画法

```
1 clear;clc;
2 x1=0:0.01:4*pi;
3 y1=sin(x1);
4 plot(x1,y1);
5 xlabel('x');ylabel('y');
6 figure(2)
7 x2=linspace(1,100,100);
8 y2=x2.^2-2*x2+sqrt(x2)-x2.^(1/3)-10*x2.^(1.5);
9 plot(x2,y2,'k');
10 xlabel('x');ylabel('y');
```

其中 `figure(2)` 是再建立一个窗口用来画图，`xlabel('x');``ylabel('y')` 用来对  $x, y$  轴打标注，`title('xxx')` 用来写标题。

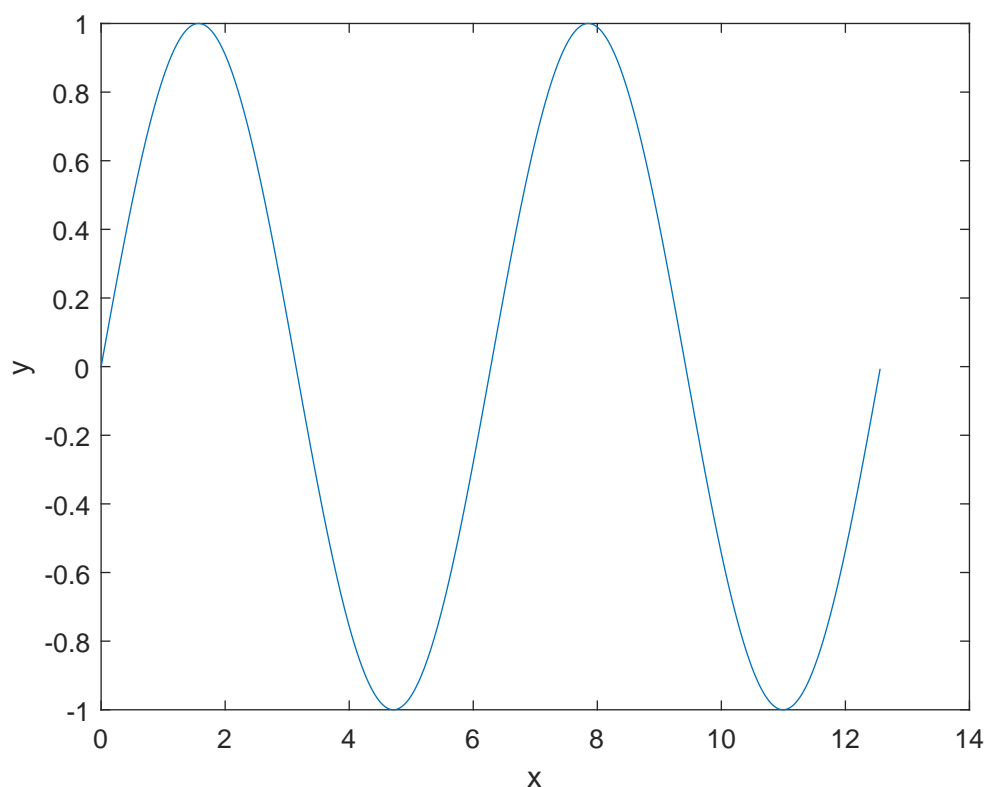


图 1:  $y = \sin(x)$

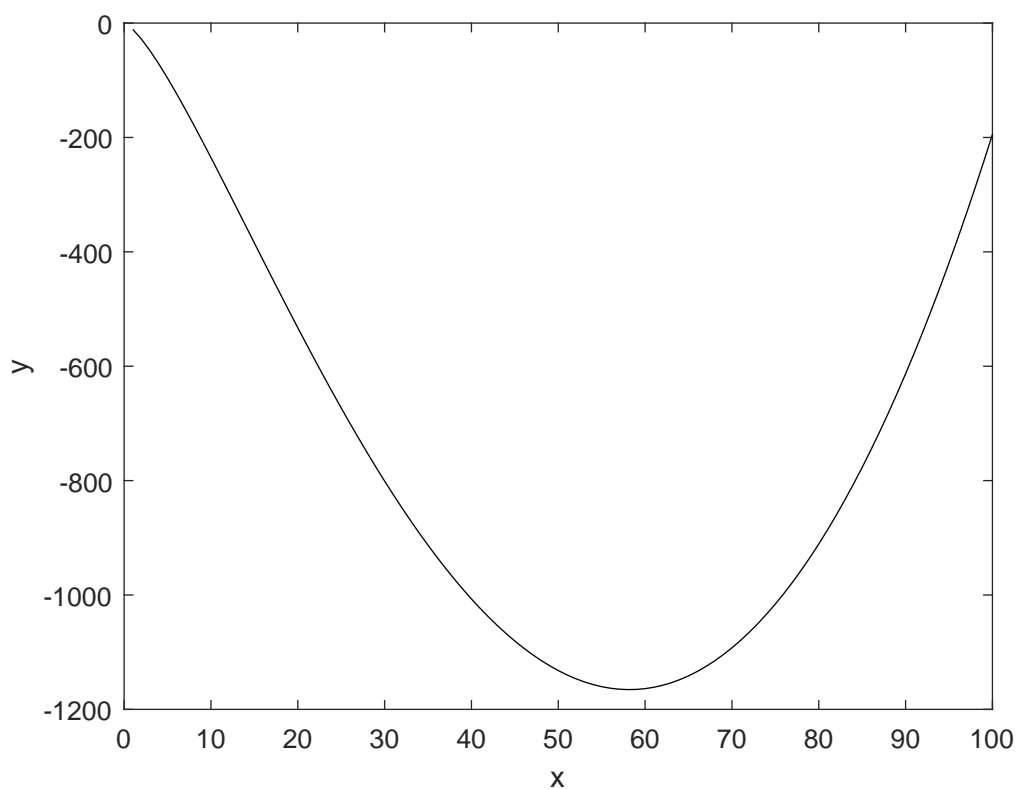


图 2:  $y = x^2 - 2 * x + \sqrt{x} - x^{\frac{1}{3}} - 10 * x^{1.5}$

## 隐函数画法

```

1 clear;clc;
2 k=0;a=1;
3 f = @(x,y) cos(x*a)+cos(y*a)-k;
4 fimplicit(f);
5 hold on
6 plot([-5,5],[0,0],'k');
7 hold on
8 plot([0,0],[-5,5],'k');
9 xlim([-4,4]);ylim([-4,4]);

```

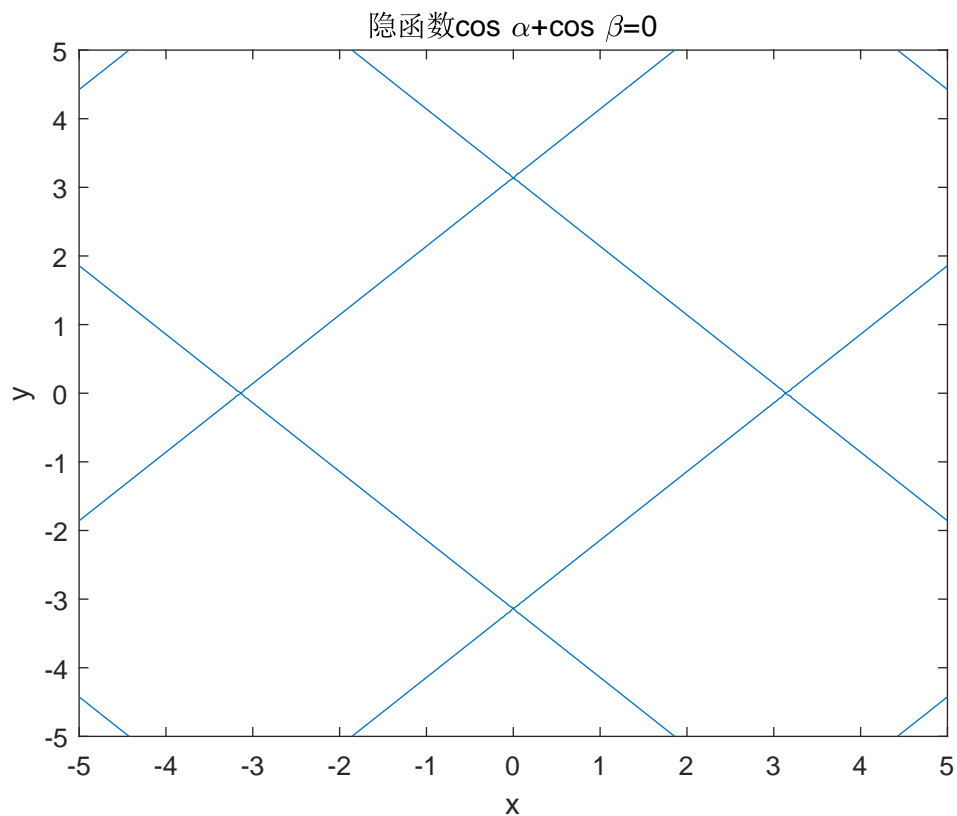


图 3: 隐函数  $\cos \alpha + \cos \beta = 0$

### 子图画法

```

1  clear;clc;
2  x1=0:0.01:4*pi;
3  subplot(2,2,1)
4  y1=sin(x1);
5  plot(x1,y1);
6  xlabel('x');ylabel('y');title('(1)');
7  subplot(2,2,2)
8  x2=linspace(1,100,100);
9  y2=x2.^2-2*x2+sqrt(x2)-x2.^(1/3)-10*x2.^(1.5);
10 plot(x2,y2,'k');
11 xlabel('x');ylabel('y');title('(2)');
12 subplot(2,2,3)
13 peaks(8);title('(3)');
14 subplot(2,2,4)
15 x4=linspace(0,7,90);
16 a1=2*exp(-0.2*x4);
17 b1=sin(8*x4);
18 y41=a1.*b1;
19 a2=3*exp(-0.3*x4);
20 b2=sin(4*x4);

```

```

21 y42=5+a2.*b2;
22 plot(x4,y41,'--',x4,y42,':');title('(4)');

```

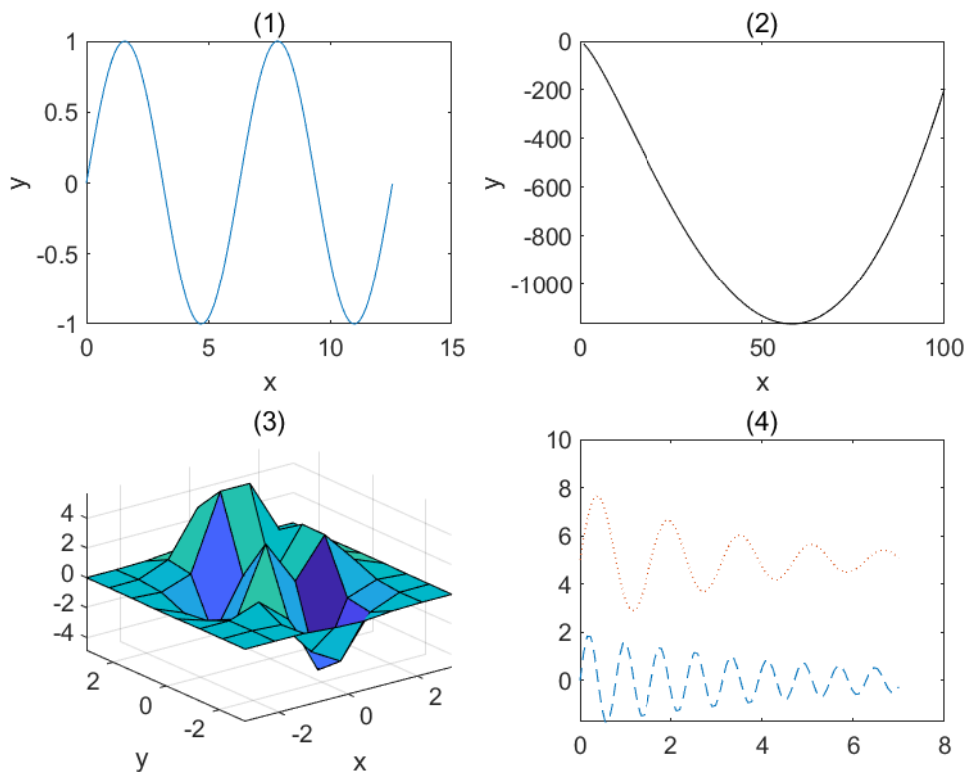


图 4: 子图画法

(1)  $y = \sin x$

(2)  $y = x^2 - 2 * x + \sqrt{x} - x^{\frac{1}{3}} - 10 * x^{1.5}$

(3)  $peaks$  函数

(4)  $y = 2 * e^{-0.2x} \sin 8x$  和  $y = 5 + 3e^{-0.3x} \sin 4x$

## 茎叶图和标注

```

1 x=[-0.3,0.5,1,-2,3,-4]; %读入信号
2 xDigital=round(x/0.15); %四舍五入，量化成二进制数
3 y=xDigital*0.15; %计算数字化后的信号
4 Error=y-x; %计算模拟信号和数字信号之间的误差
5 figure(1) %画两信号图
6 stem(x);
7 hold on
8 stem(y);
9 xlim([0 6]);
10 xticks(0:1:6);
11 legend('Origin Signal','Digital Signal');

```

```

12 xlabel('Signal');ylabel('Alputium');
13 title('Siginal and Alputium');
14 figure(2)           %画绝对误差图
15 stem(Error);
16 xticks(0:1:6);
17 xlabel('Signal');ylabel('Error');
18 title('Absolute Error');
19 figure(3)           %画相对误差图
20 stem(Error./x);
21 xlabel('Signal');ylabel('Error');
22 title('Relative Error');

```

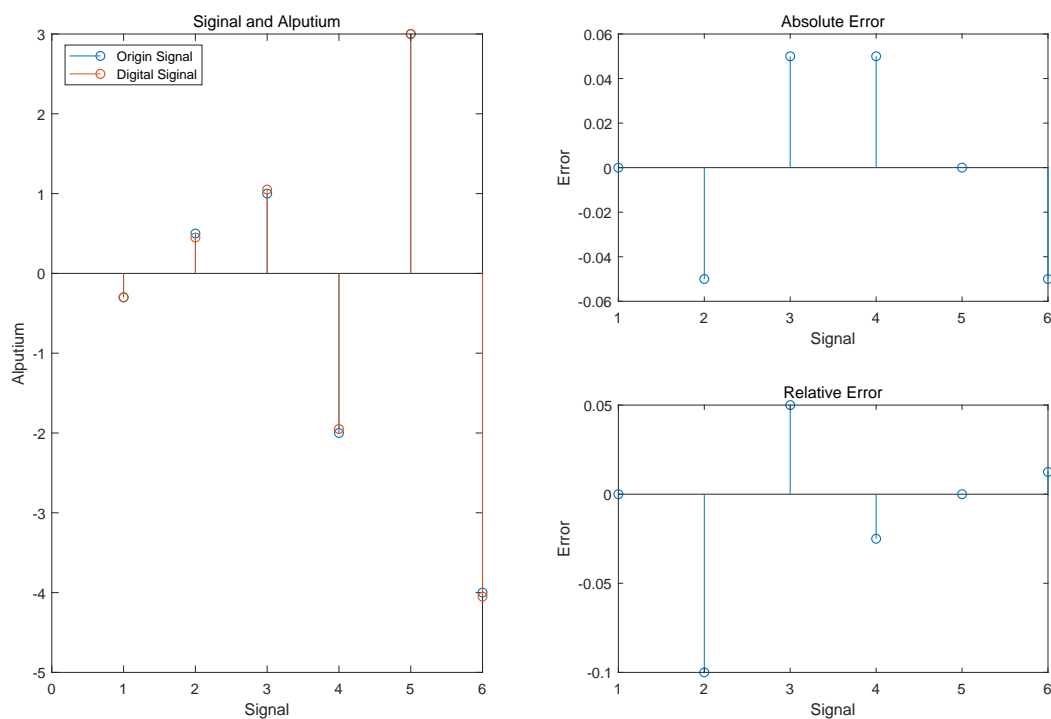


图 5: 茎叶图和标注

## 双对数坐标轴

```

1 clear;clc;
2 x=[1e1,1e2,1e3,1e4,1e5,1e6,1e7,1e8,1e9];
3 y1=x.^2+2*x+4;
4 y2=x.^3+3*x+5;
5 subplot(2,2,[1,3])
6 loglog(x,y1,x,y2,'k:');
7 texy1=texlabel('y_1=x^2+2x+4');texy2=texlabel('y_2=x^3+3x+5');
8 legend(texy1,texy2);
9 axis([1,1e10,1,1e30]);

```

```

10 box off
11 subplot(2,2,2)
12 semilogy(x,y1,x,y2,'k:');
13 subplot(2,2,4)
14 semilogx(x,y1,'r',x,y2);
15 box off

```

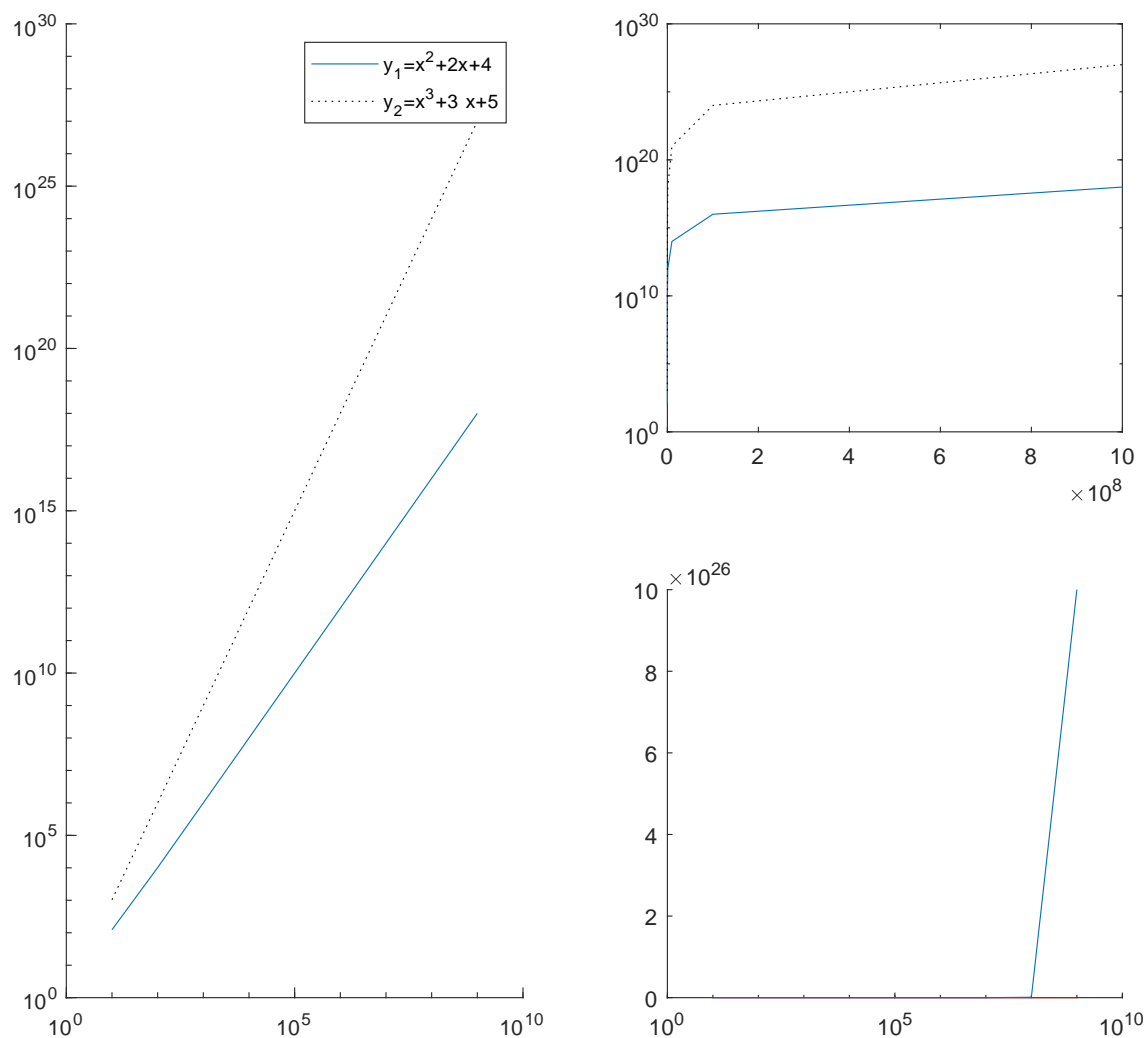


图 6: 双对数坐标绘图

### 三、函数表达式画图

之前的 `plot` 命令都是根据点连线画出的曲线，如果不知道  $x$  的取值，那怎么画图？就用 `fplot` 命令即可



## fplot 绘图

```
1 clear;clc
2 syms x
3 y=sin(x)/x;
4 fplot(y);
5 tex=texlabel('y=sin alpha/alpha');
6 text(3,0.8,tex);
7 grid on;box off
```

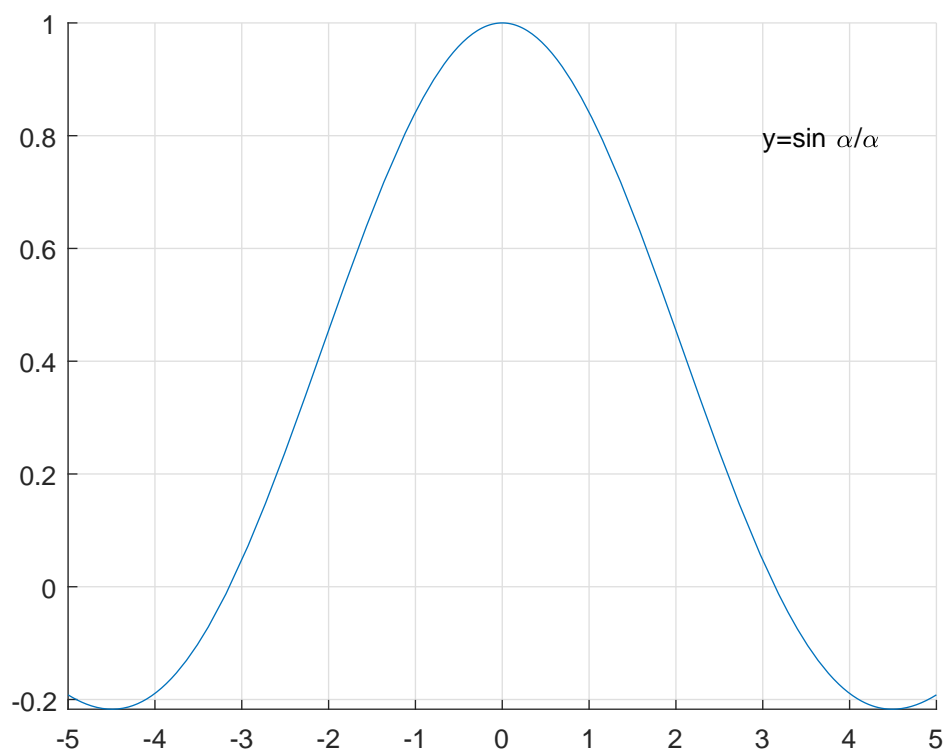


图 7: text 打标记和 fplot 画图

## 交互绘图

这个看图是看不出来什么的，要把代码放进 MATLAB 里面跑一下会比较好，大概适用于画出了某些函数，然后要写个标注，但是现在的标注很多都在图像窗口可以完成了，所以可能有些鸡肋吧。

```
1 clear;clc;
2 axis([0,20,0,20]);
3 grid on
4 [x,y] = ginput(6);
5 plot(x,y,'k.-');
6 axis([0,20,0,20]);
```

```

7 gtext('USTC');
8 axis([0,20,0,20]);
9 grid on;box off;

```

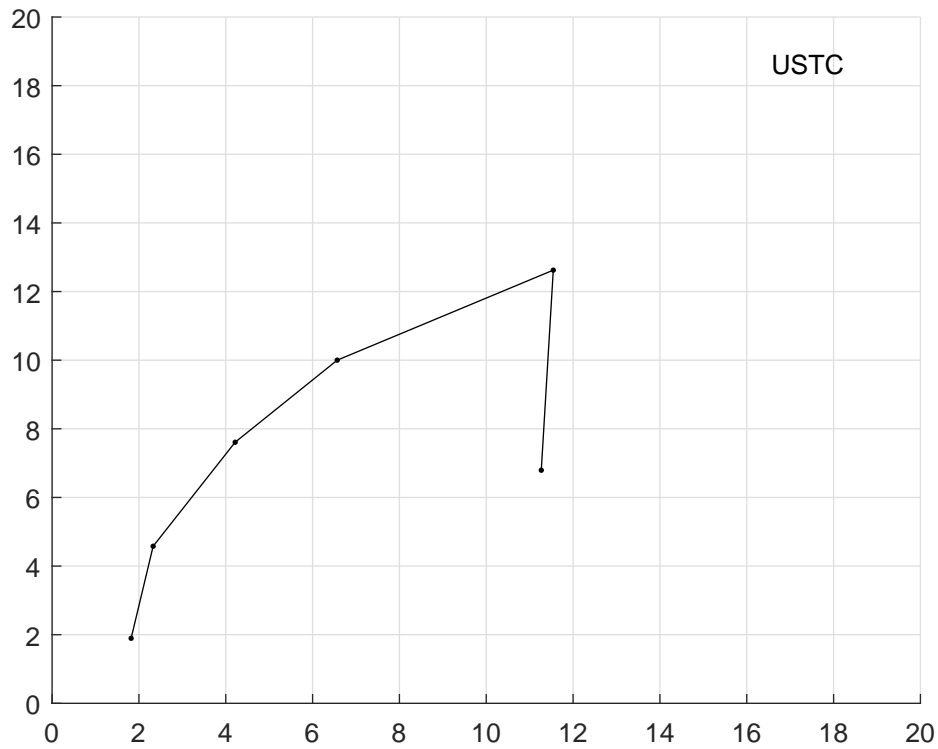


图 8: 交互绘图

#### 四、常用作图

##### 柱状图

```

1 clear;clc;
2 x=linspace(0,2*pi,18);
3 x=x';
4 y1=10*sin(x);
5 y2=1.2*y1;
6 y=[y1,y2];
7 barh(x,y);
8 hold on
9 plot(y,x,'o-');
10 axis([-12,12,0,7]);
11 box off

```

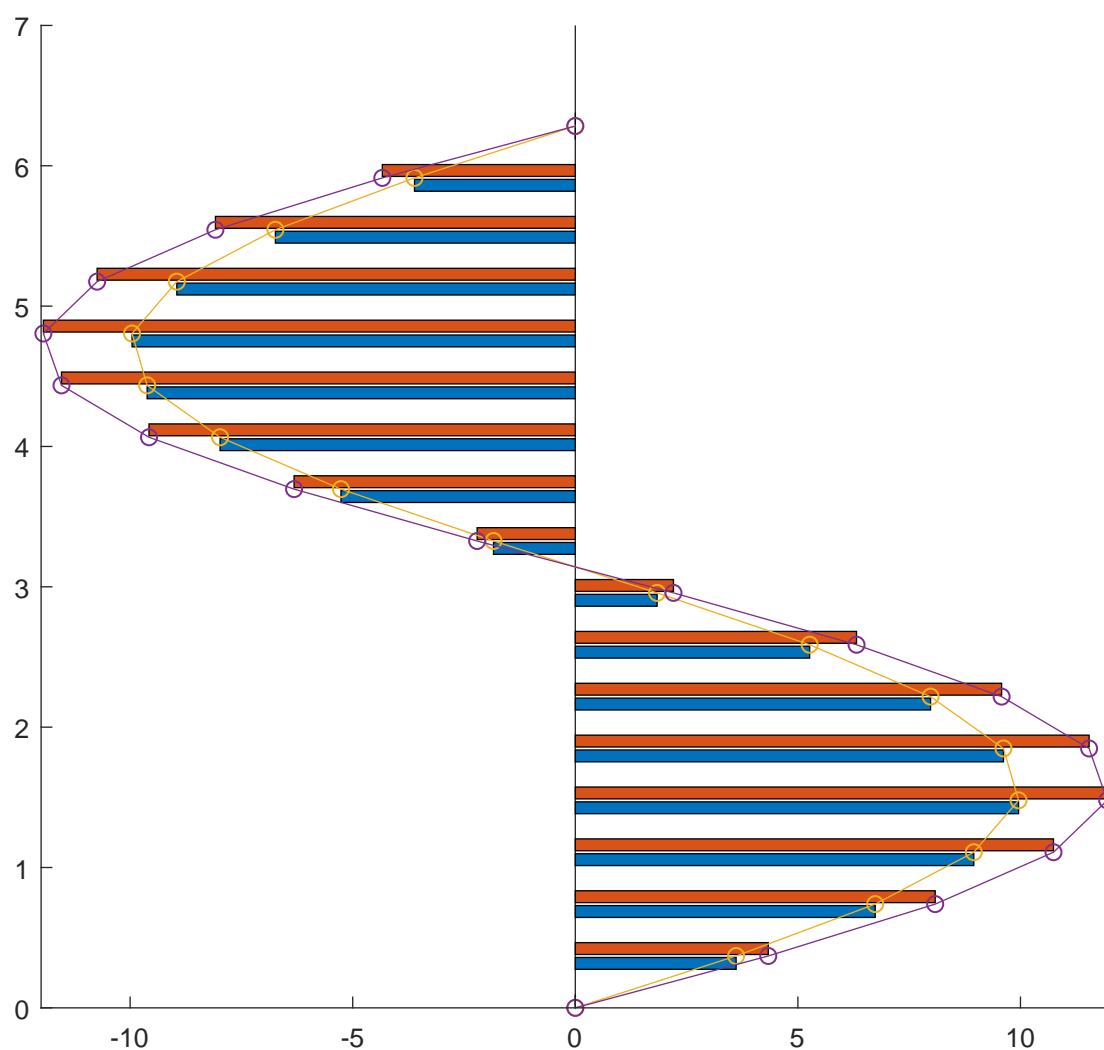


图 9: 水平柱状图绘制

## 饼图绘制

```

1 clear;clc;
2 x=[1,2;2,4];
3 y=[1,0;0,0];
4 pie(x,y);
5 legend('p','p','p','p','p');
6 title('c k%± ');

```

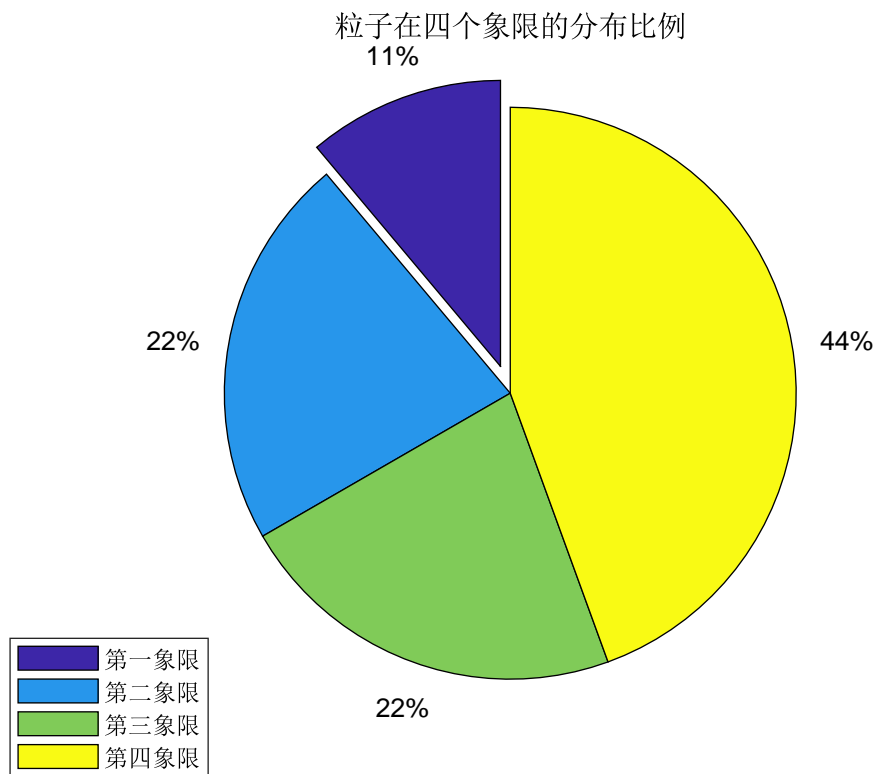


图 10: 饼图绘制

## 彗星图

这个也就是那种可以根据  $x$  取点一点一点走的图, emm, 在这里还展示不了, MATLAB 导出动图和 pdf 插入动图都比较麻烦, 可以在 MATLAB 上面跑一下代码一下子就出来了, 那个小圈就是轨迹出来的点。

```
1 clear;clc;
2 x=linspace(0,8*pi,3600);
3 y=exp(-0.1*x).*sin(x);
4 comet(x,y);
```

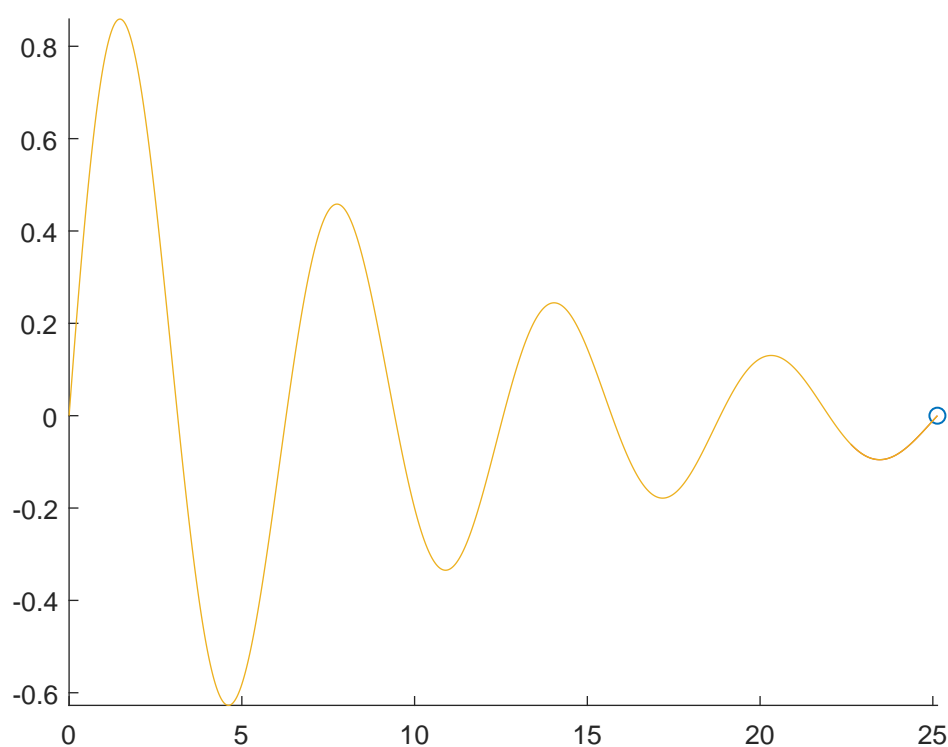


图 11: 彗星图 (这里是静态的, 要在 MATLAB 里面跑)

## 其他绘图

```

1  clear;clc;
2  t=linspace(0,2*pi,9);
3  t2=0:0.001:2*pi;
4  x=sin(t);
5  y=cos(t);
6  rho=sin(t2).*cos(t2);
7  subplot(2,2,1)
8  polar(t2,rho);
9  tex1=texlabel('rho=rho(theta)');
10 title([tex1,' $\mu \ll \pi$ ']);
11 subplot(2,2,2)
12 compass(x,y);
13 title(' ');
14 subplot(2,2,3)
15 scatter(x,y);
16 title('  $\mu$  ');
17 subplot(2,2,4)
18 u=gradient(t);
19 v=gradient(y);
20 quiver(t,y,u,v);
21 title(' ');

```

22 `grid on;box off`

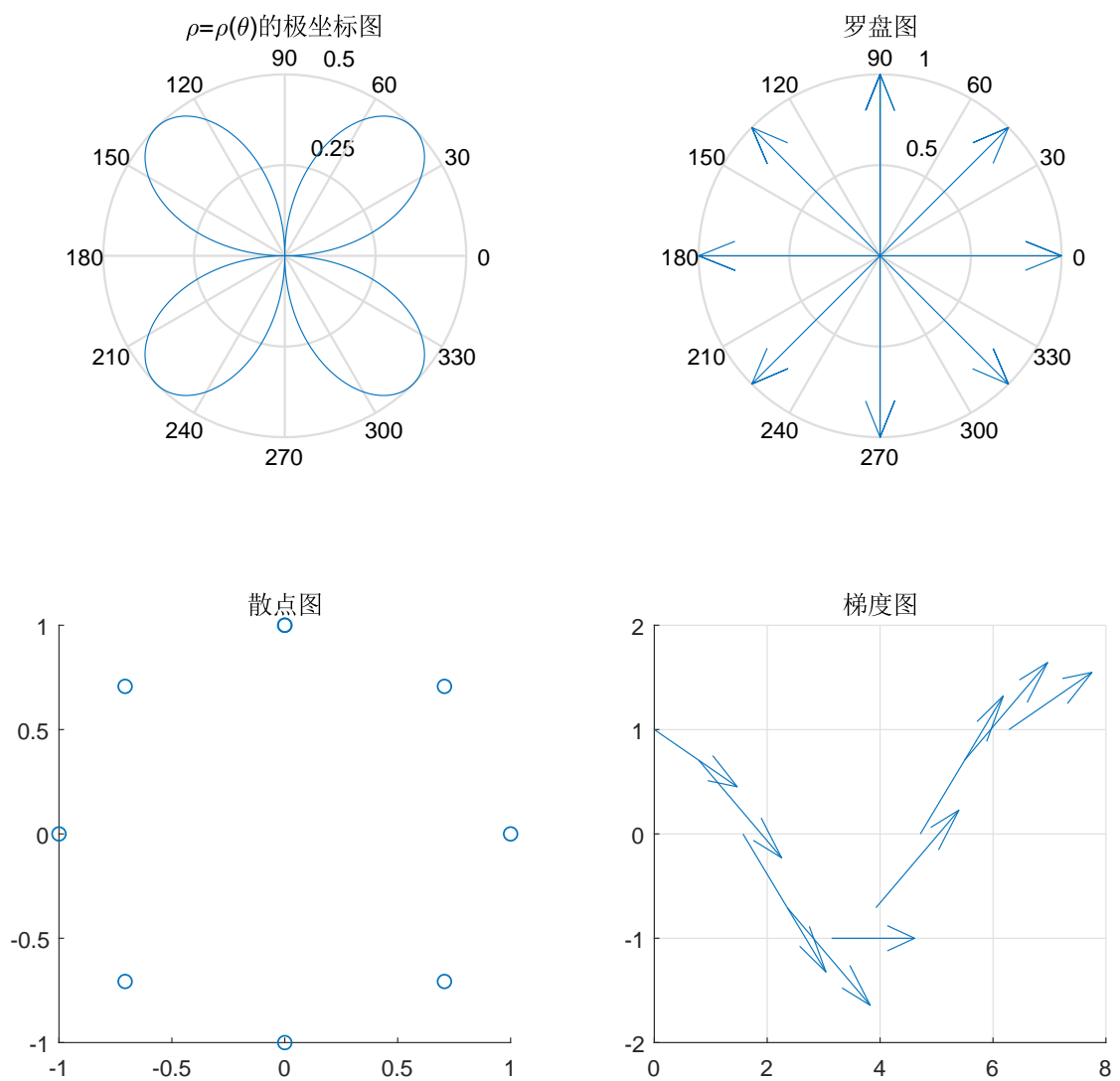


图 12: (1) 极坐标;(2) 罗盘图;(3) 散点图;(4) 梯度图

## 五、三维图

### 空间曲线

```
1 clear;clc;
2 t=linspace(0,12*pi,500);
3 x=cos(t);
4 y=cos(t).*sin(t);
5 z=t.^2;
6 plot3(x,y,z);
```

```

7 xlabel('cos t');ylabel('cos t*sin t');zlabel('t^2');
8 grid on

```

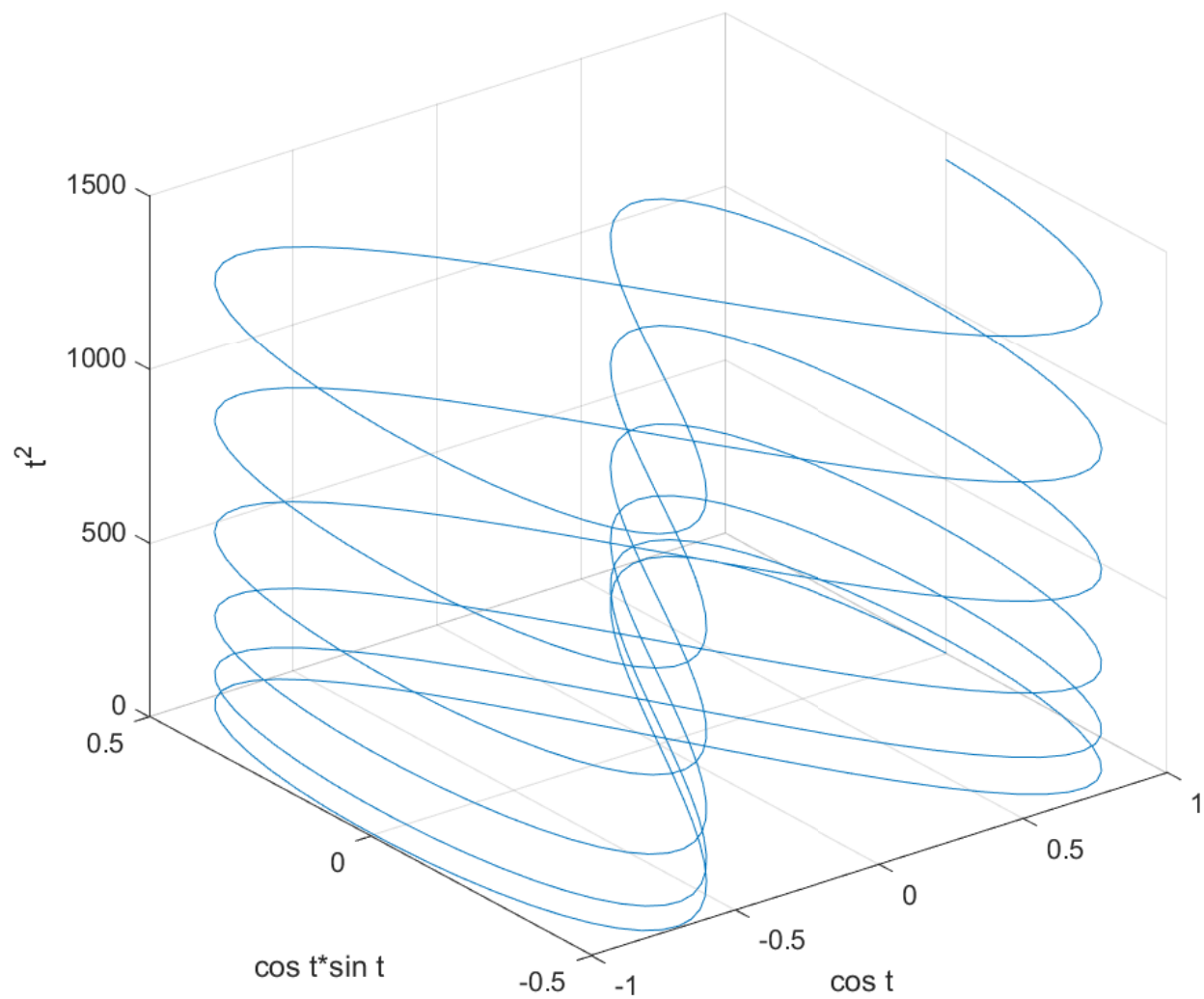


图 13: 空间曲线

### 空间曲线切线

```

1 clear;clc;
2 t=linspace(0,6*pi,30);
3 x=10*sin(t);
4 y=10*cos(t);
5 z=t;
6 u=gradient(x);
7 v=gradient(y);
8 w=gradient(z);

```

```

9 quiver3(x,y,z,u,v,w);
10 hold on
11 t=linspace(0,6*pi,3000);
12 x=10*sin(t);
13 y=10*cos(t);
14 z=t;
15 plot3(x,y,z);
16 box off

```

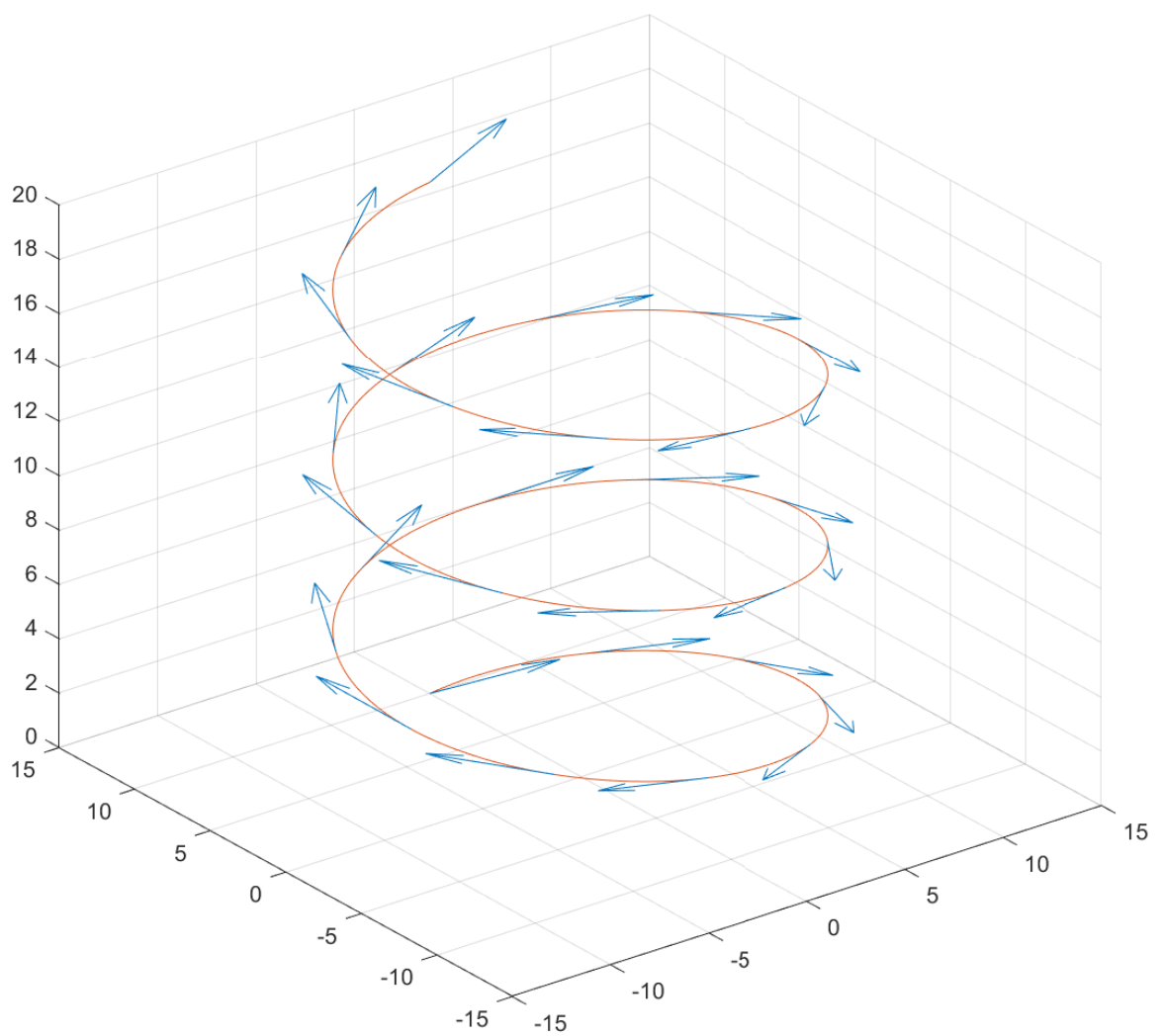


图 14: 空间曲线切线

### 三维曲面

```

1 [X,Y] = meshgrid(-8:.5:8);
2 R = sqrt(X.^2 + Y.^2) + eps;

```



```

3  Z = sin(R) ./ R;
4  C = del2(Z);
5  subplot(2,2,1)
6  mesh(X,Y,Z,C,'FaceLighting','gouraud','LineWidth',0.3);
7  subplot(2,2,2)
8  surf(X,Y,Z);
9  subplot(2,2,3)
10 meshc(X,Y,Z);colorbar;
11 subplot(2,2,4)
12 surfc(X,Y,Z);colorbar;

```

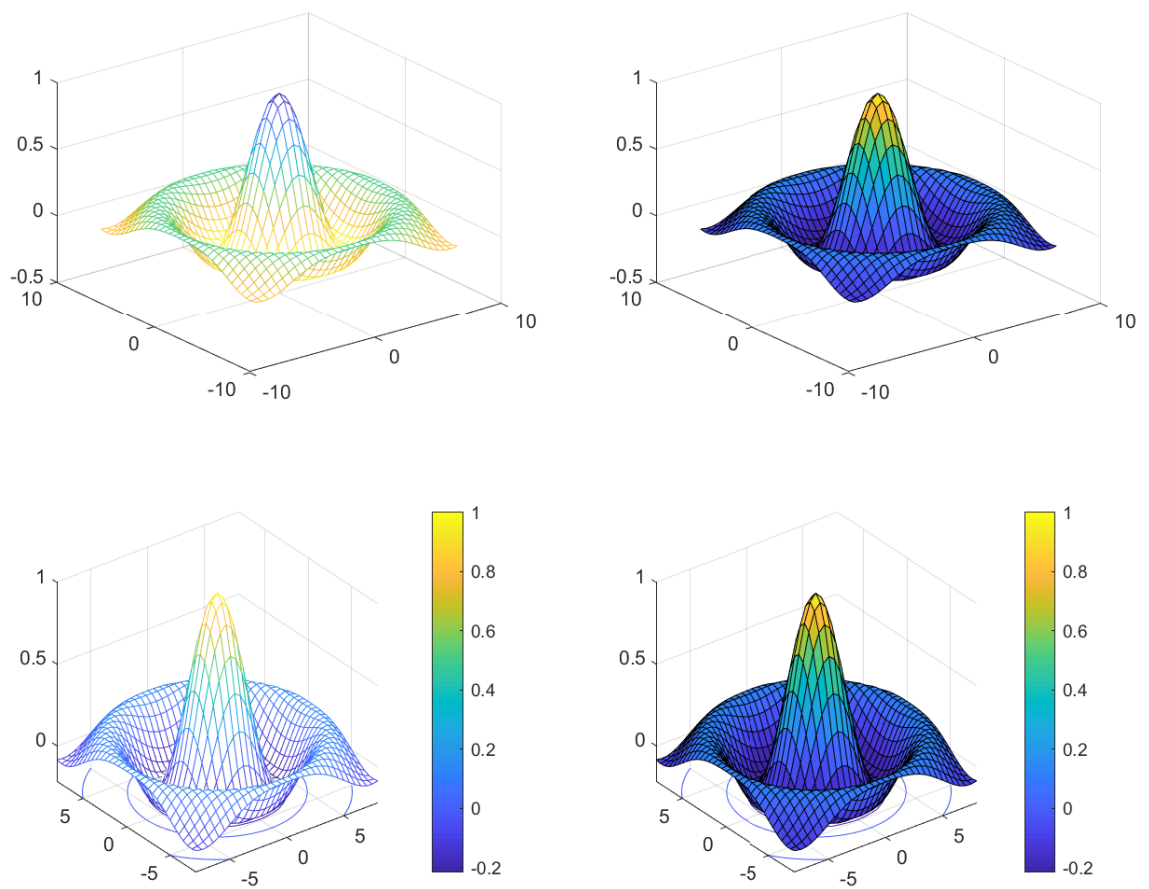


图 15: 三维曲面

### 几种其他的绘图命令

```

1  clear;clc
2  clf,x=linspace(-2,2,40);
3  y=x;z=x;
4  [X,Y,Z]=meshgrid(x,y,z);

```

```

5  w=X.^2+Y.^2+Z.^2;
6  figure(1)
7  hFig=gcf;
8  set(hFig, 'Position', [100,100,1200,800]);
9  subplot(2,2,1)
10 slice(X,Y,Z,w,[1,0,-1],[-1,0,1],[-1,0,1]);
11 xlabel('x');ylabel('y');zlabel('z');
12 colormap(gca,'spring')
13 hcb1=colorbar;
14 title('slice');
15 subplot(2,2,2)
16 contour(peaks);
17 colormap(gca,'gray')
18 hcb2=colorbar;
19 subplot(2,2,3)
20 sphere;
21 colormap(gca,'autumn')
22 hcb3=colorbar;
23 title('sphere');
24 shading flat
25 subplot(2,2,4)
26 surfc(peaks);
27 colormap(gca,'pink')
28 hcb4=colorbar;
29 shading interp

```

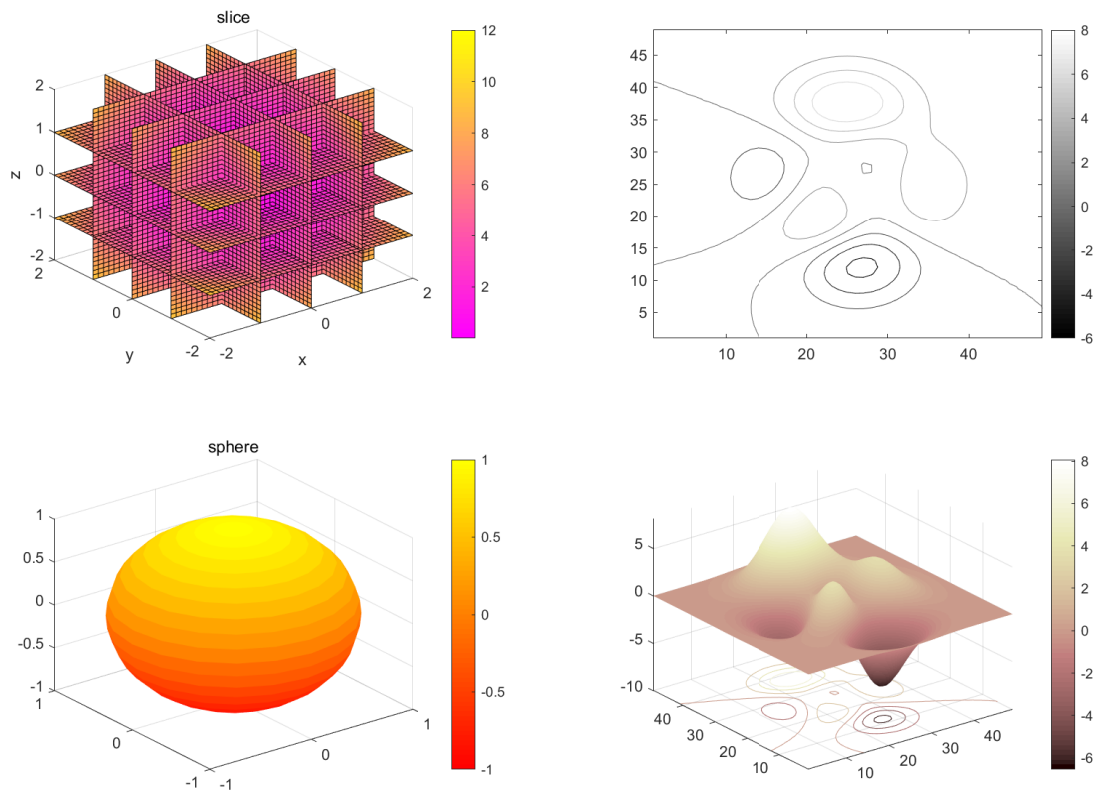


图 16: (1)slice;(2)contour;(3)colorbar;(4)surf

## 六、几个有趣的例子

### 莫比乌斯环

```

1  clc; clear all; close all;
2  t=linspace(-pi, pi); v=linspace(-1,1);
3  a=20; b=8; [T,V]=meshgrid(t,v);
4  r=a+b*cos(T/2); x=r*cos(T);
5  y=r*sin(T); z=b*cos(T/2);
6  surf(x,y,z);
7  shading interp; axis off;
8  grid off; axis equal;
9  % set(gcf, 'color', 'k', 'menubar', 'none', 'units', 'normalized', 'position', [0 0 1 1]);
10 % cameratoolbar;

```

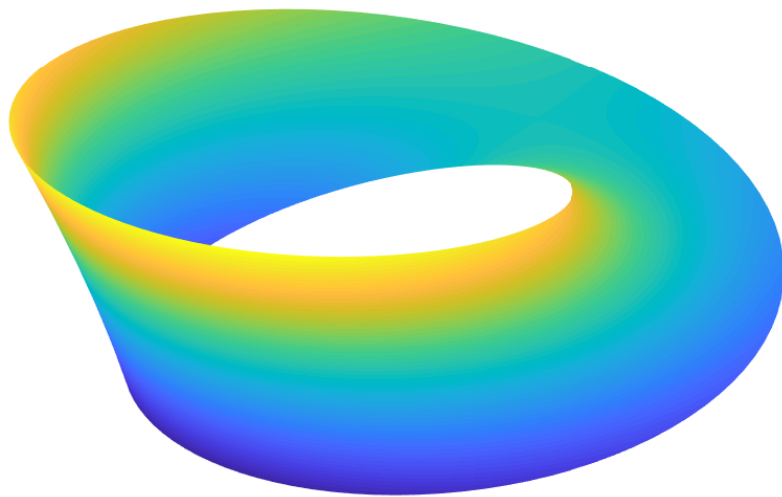


图 17: 莫比乌斯环

画一朵花

```

1  clc; clear all; close all;
2  t=linspace(-pi, pi); v=linspace(-1,1);
3  a=20; b=8; [T,V]=meshgrid(t,v);
4  r=a+b*V.*cos(T/2); x=r.*cos(T);
5  y=r.*sin(T); z=b*V.*sin(T/2);
6  surf(x,y,z);
7  shading interp; axis off;
8  grid off; axis equal;
9  % set(gcf, 'color', 'k', 'menubar', 'none', 'units', 'normalized', 'position', [0 0 1 1]);
10 % cameratoolbar;

```



图 18: 花

## 七、曲线拟合

待填坑...

## 八、回归分析

待填坑...

## 九、小技巧

- 开始学的时候可以用实时脚本文件做编程，“所见即所得”，这是很好的方式；
- *help* 很强大，一定要善于使用；
- **MATLAB** 很强大也很简单，有用到就去学，不必一个一个对着书上敲代码把例子一个一个过，要是想学可以看一下用了什么指令，调用格式是什么，然后自己写一个，这样子应该会学的快一点。