**MATLAB的基本知识**

**1、基本运算与函数**

在MATLAB下进行基本数学运算，只需将运算式直接打入提示号（>>）之後，并按入**Enter**键即可。例如：

>> (5\*2+1.3-0.8)\*10/25

ans =4.2000

MATLAB会将运算结果直接存入一变数ans，代表MATLAB运算後的答案（Answer）并显示其数值於萤幕上。

小提示： ">>"是MATLAB的提示符号（Prompt），但在PC中文视窗系统下，由於编码方式不同，此提示符号常会消失不见，但这并不会影响到MATLAB的运算结果。

我们也可将上述运算式的结果设定给另一个变数x：

x = (5\*2+1.3-0.8)\*10^2/25

x = 42

此时MATLAB会直接显示x的值。由上例可知，MATLAB认识所有一般常用到的加（+）、减（-）、乘（\*）、除（/）的数学运算符号，以及幂次运算（^）。

小提示： MATLAB将所有变数均存成double的形式，所以不需经过变数宣告（Variable declaration）。MATLAB同时也会自动进行记忆体的使用和回收，而不必像C语言,必须由使用者一一指定.这些功能使的MATLAB易学易用，使用者可专心致力於撰写程式，而不必被软体枝节问题所干扰。

若不想让MATLAB每次都显示运算结果，只需在运算式最後加上分号（；）即可，如下例：

y = sin(10)\*exp(-0.3\*4^2);

若要显示变数y的值，直接键入y即可：

>>y

y =-0.0045

在上例中，sin是正弦函数，exp是指数函数，这些都是MATLAB常用到的数学函数。

下表即为MATLAB常用的基本数学函数及三角函数：

小整理：MATLAB常用的基本数学函数

abs(x)：纯量的绝对值或向量的长度

angle(z)：复 数z的相角(Phase angle)

sqrt(x)：开平方

real(z)：复数z的实部

imag(z)：复数z的虚 部

conj(z)：复数z的共轭复数

round(x)：四舍五入至最近整数

fix(x)：无论正负，舍去小数至最近整数

floor(x)：地板函数，即舍去正小数至最近整数

ceil(x)：天花板函数，即加入正小数至最近整数

rat(x)：将实数x化为分数表示

rats(x)：将实数x化为多项分数展开

sign(x)：符号函数 (Signum function)。

当x<0时，sign(x)=-1；

当x=0时，sign(x)=0;

当x>0时，sign(x)=1。

> 小整理：MATLAB常用的三角函数

sin(x)：正弦函数

cos(x)：馀弦函数

tan(x)：正切函数

asin(x)：反正弦函数

acos(x)：反馀弦函数

atan(x)：反正切函数

atan2(x,y)：四象限的反正切函数

sinh(x)：超越正弦函数

cosh(x)：超越馀弦函数

tanh(x)：超越正切函数

asinh(x)：反超越正弦函数

acosh(x)：反超越馀弦函数

atanh(x)：反超越正切函数

变数也可用来存放向量或矩阵，并进行各种运算，如下例的列向量（Row vector）运算：

x = [1 3 5 2];

y = 2\*x+1

y = 3 7 11 5

小提示：变数命名的规则

1.第一个字母必须是英文字母 2.字母间不可留空格 3.最多只能有19个字母，MATLAB会忽略多馀字母

我们可以随意更改、增加或删除向量的元素：

y(3) = 2 % 更改第三个元素

y =3 7 2 5

y(6) = 10 % 加入第六个元素

y = 3 7 2 5 0 10

y(4) = [] % 删除第四个元素，

y = 3 7 2 0 10

在上例中，MATLAB会忽略所有在百分比符号（%）之後的文字，因此百分比之後的文字均可视为程式的注解（Comments）。MATLAB亦可取出向量的一个元素或一部份来做运算：

x(2)\*3+y(4) % 取出x的第二个元素和y的第四个元素来做运算

ans = 9

y(2:4)-1 % 取出y的第二至第四个元素来做运算

ans = 6 1 -1

在上例中，2:4代表一个由2、3、4组成的向量

若对MATLAB函数用法有疑问，可随时使用help来寻求线上支援（on-line help）：help linspace

小整理：MATLAB的查询命令

help：用来查询已知命令的用法。例如已知inv是用来计算反矩阵，键入help inv即可得知有关inv命令的用法。（键入help help则显示help的用法，请试看看！） lookfor：用来寻找未知的命令。例如要寻找计算反矩阵的命令，可键入 lookfor inverse，MATLAB即会列出所有和关键字inverse相关的指令。找到所需的命令後 ，即可用help进一步找出其用法。（lookfor事实上是对所有在搜寻路径下的M档案进行关键字对第一注解行的比对，详见後叙。）

将列向量转置（Transpose）後，即可得到行向量（Column vector）：

z = x'

z = 4.0000

5.2000

6.4000

7.6000

8.8000

10.0000

不论是行向量或列向量，我们均可用相同的函数找出其元素个数、最大值、最小值等：

length(z) % z的元素个数

ans = 6

max(z) % z的最大值

ans = 10

min(z) % z的最小值

ans = 4

小整理：适用於向量的常用函数有：

min(x): 向量x的元素的最小值

max(x): 向量x的元素的最大值

mean(x): 向量x的元素的平均值

median(x): 向量x的元素的中位数

std(x): 向量x的元素的标准差

diff(x): 向量x的相邻元素的差

sort(x): 对向量x的元素进行排序（Sorting）

length(x): 向量x的元素个数

norm(x): 向量x的欧氏（Euclidean）长度

sum(x): 向量x的元素总和

prod(x): 向量x的元素总乘积

cumsum(x): 向量x的累计元素总和

cumprod(x): 向量x的累计元素总乘积

dot(x, y): 向量x和y的内 积

cross(x, y): 向量x和y的外积 （大部份的向量函数也可适用於矩阵，详见下述。）

2.多项式运算

2.1．向量或数组的表达

向量输入格式：变量名=[ 向量数据 ]

输入向量方法：用键盘上方括号“ [ ] ”代替向量括号。向量数据在方括号内按行输入，相邻元素用空格（或逗号）隔开。

2.2．多项式的表达

(1) 多项式的向量表达

i）形如 的多项式，可以用向量来表示：

例如：就 可以表示为P=[1 0 -2 3 -4]。

ii）已知多项式的根为 ，则该多项式为：poly(A)

(2) 多项式的形式表达

i) 直接以符号的形式输出多项式，例如：

>> f1=sym('x^4-2\*x^2+3\*x-4')

f1 =

x^4-2\*x^2+3\*x-4

ii) 定义符号变量，再输出多项式的表达式，例如：

>> syms x

>> f1=x^4-2\*x^2+3\*x-4

f1 =

x^4-2\*x^2+3\*x-4

iii）已知多项式的向量表达形式，则多项式为：poly2str(A,’x’)，例如：

>> A=[1 0 -2 3 -4]; %多项式的向量表达式

>> f1=poly2str(A,'x') %多项式的形式表达式

f1 =

x^4 - 2 x^2 + 3 x - 4

2.3．多项式的运算

（1）向量表达的多项式运算

在进行加法与减法运算时，参加运算的多项式必须具有相同的阶数，如果阶数不同，则低阶的多项式必须补零。例如：

>> a=[1 0 2];b=[1 2 3 5 0]; %两个多项式

>> f1=poly2str(a,'x')

f1 =

x^2 + 2

>> f2=poly2str(b,'x')

f2 =

x^4 + 2 x^3 + 3 x^2 + 5 x

>> c=[0 0 a] + b %求和

c =

1 2 4 5 2

>> poly2str(c,'x')

ans =

x^4 + 2 x^3 + 4 x^2 + 5 x + 2

>> d=[0 0 ,a]-b %求差

d =

-1 -2 -2 -5 2

>> poly2str(d,'x')

ans =

-1 x^4 - 2 x^3 - 2 x^2 - 5 x + 2

多项式乘法：conv(x,y) 多项式x与y的乘积。例如：

>> e=conv(a,b) %求积

e =

1 2 5 9 6 10 0

>> poly2str(e,'x')

ans =

x^6 + 2 x^5 + 5 x^4 + 9 x^3 + 6 x^2 + 10 x

多项式带余除法：[q,r]=deconv(x,y) 多项式y被x除,q是商式，r是余式。

例如：

>> [q,r]=deconv(b,a) %带余除法

q =

1 2 1

r =

0 0 0 1 -2

>> q=poly2str(q,'x') %商式

q =

x^2 + 2 x + 1

>> r=poly2str(r,'x') %余式

r =

x - 2

多项式的根：roots(x)。例如matlab代码：

>> roots(a) %求多项式的根

运算结果为：

ans =

0 + 1.4142i

0 - 1.4142i

多项式的值：polyval(f,x) f(x)的值。例如：

>> polyval(a,-3) %计算f1(-3)

ans =

11

多项式函数求导运算：polyder(x) 多项式x的导函数。例如：

>> g=polyder(a) %求多项式f1的导函数

g =

2 0

>> poly2str(g,'x')

ans =

2 x

（2）符号形式的多项式运算

多项式的加法，减法，乘法，除法。例如：

>> clear

>> syms x

>> f1=x^2-1;f2=x^3+x^2-2; %输出多项式f1与f2

>> f1+f2 %求和

ans =

2\*x^2-3+x^3

>> f2-f1 %求差

ans =

x^3-1

>> f1\*f2 %求积

ans =

(x^2-1)\*(x^3+x^2-2)

>> f1/f2 %求商

ans =

(x^2-1)/(x^3+x^2-2)

多项式的幂：x^p或power(x,p) 其中x是多项式,p是正整数。

多项式展开：expand(x) 展开多项式x。

多项式因式分解：factor(x) 在有理数域上因式分解开多项式x。例如：

>> f3=f1^3 %多项式f1的三次方

f3 =

(x^2-1)^3

>> f4=power(f1,3) %多项式f1的三次方

f4 =

(x^2-1)^3

>> expand(f3) %展开多项式f3

ans =

x^6-3\*x^4+3\*x^2-1

>> f5=factor(f2) %将多项式f2因式分解

f5 =

(x-1)\*(x^2+2\*x+2)

最大公因式：gcd(x,y) x与y的最大公因式。

[g,c,d]=gcd(x,y) 其中g=cx+dy.

最小公倍式：lcm(x,y) x与y的最小公倍式。例如：

>> gcd(f1,f2) %求多项式f1与f2的最大公因式

ans =

x-1

>> [g,c,d]=gcd(f1,f2) %最大公因式的组合

g =

x-1

c =

-1-x

d =

1

>> lcm(f1,f2) %求多项式f1与f2的最小公倍式

ans =

(x+1)\*(x^3+x^2-2)

多项式的根：solve(x) 例如：

>> solve(f2) %求多项式f2的根

ans =

1

-1+i

-1-i

多项式函数求导运算：diff(x) 多项式x的导函数。例如：

>> g=diff(f2) %求多项式f2的导函数

g =

3\*x^2+2\*x

表2-1 多项式函数的一些相关命令

|  |  |  |  |
| --- | --- | --- | --- |
| conv(a, b) | 乘法 | x^p | 多项式幂运算 |
| [q, r]=deconv(a, b) | 带余除法 | power(x,p) | 多项式幂运算 |
| poly(r) | 用根构造多项式 | expand(x) | 多项式展开 |
| poly2str(f,'x') | 多项式符号表达式 | factor(x) | 多项式因式分解 |
| polyder(x) | 多项式求导 | roots(x) | 求多项式的根 |
| diff(x) | 符号多项式求导 | solve(x) | 求符号多项式的根 |
| polyval(f, x) | 计算f(x)的值 | gcd(x,y) | 求最大公因式 |
| lcm(x,y) | 求最小公倍式 | [g,c,d]=gcd(x,y) | 求最大公因式的组合 |

**3　作图**

**二维作图**

**3.1 基本形式**

如果y是一个向量，那么plot(y)绘制一个y中元素的线性图．假设我们希望画出

y=[0.,  0.48,  0.84,  1.,  0.91,  6.14 ]

则用命令：plot(y)

它相当于命令：plot(x, y)，其中x=[1,2,…,n]或x=[1;2;…;n]，即向量y的下标编号, n为向量y的长度

**Matlab**会产生一个图形窗口，显示如下图形，请注意：坐标x和y 是由计算机自动绘出的．



图4.1.1.1 plot([0.,0.48,0.84,1.,0.91,6.14])

上面的图形没有加上x轴和y轴的标注，也没有标题．用xlabel，ylabel，title命令可以加上．

如果x，y是同样长度的向量，plot(x,y)命令可画出相应的x元素与y元素的x-y坐标图．例：

x=0:0.05:4\*pi;  y=sin(x);  plot(x,y)

grid on, title(' y=sin( x ) 曲线图' )

xlabel(' x = 0 : 0.05 : 4Pi ')

结果见下图．

11

图4.1.1.2 y=sin(x)的图形

|  |  |
| --- | --- |
| title | 图形标题 |
| xlabel | x坐标轴标注 |
| ylabel | y坐标轴标注 |
| text | 标注数据点 |
| grid | 给图形加上网格 |
| hold | 保持图形窗口的图形 |

表4.1.1.1 Matlab图形命令

**§4.1.2 多重线**

在一个单线图上，绘制多重线有三种办法.

第一种方法是利用plot的多变量方式绘制：

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

x1,y1,x2,y2,...,xn,yn是成对的向量，每一对x, y在图上产生如上方式的单线．多变量方式绘图是允许不同长度的向量显示在同一图形上．

第二种方法也是利用plot绘制，但加上hold on/off命令的配合：

plot(x1,y1)

hold on

plot(x2,y2)

hold off

第三种方法还是利用plot绘制，但代入矩阵：

如果plot用于两个变量plot(x,y)，并且x，y是矩阵，则有以下情况：

（1）如果y是矩阵，x是向量，plot(x,y)用不同的画线形式绘出y的行或列及相应的x向量，y的行或列的方向与x向量元素的值选择是相同的．

（2）如果x是矩阵，y是向量，则除了x向量的线族及相应的y向量外，以上的规则也适用．

（3）如果x，y是同样大小的矩阵，plot(x,y)绘制x的列及y相应的列．

还有其它一些情况，请参见**Matlab**的帮助系统．

**3.2 线型和颜色的控制**

如果不指定划线方式和颜色，**Matlab**会自动为您选择点的表示方式及颜色．您也可以用不同的符号指定不同的曲线绘制方式．例如：

plot(x,y,'\*')                    用'\*'作为点绘制的图形．

plot(x1,y1,':',x2,y2,'+')    用':'画第一条线，用'+'画第二条线．

线型、点标记和颜色的取值有以下几种：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 线型 | | 点标记 | | 颜色 | |
| - | 实线 | . | 点 | y | 黄 |
| : | 虚线 | o | 小圆圈 | m | 棕色 |
| -. | 点划线 | x | 叉子符 | c | 青色 |
| -- | 间断线 | + | 加号 | r | 红色 |
|  |  | \* | 星号 | g | 绿色 |
|  |  | s | 方格 | b | 蓝色 |
|  |  | d | 菱形 | w | 白色 |
|  |  | ^ | 朝上三角 | k | 黑色 |
|  |  | v | 朝下三角 |  |  |
|  |  | > | 朝右三角 |  |  |
|  |  | < | 朝左三角 |  |  |
|  |  | p | 五角星 |  |  |
|  |  | h | 六角星 |  |  |

表4.1.3.1线型和颜色控制符

如果你的计算机系统不支持彩色显示，**Matlab**将把颜色符号解释为线型符号，用不同的线型表示不同的颜色．颜色与线型也可以一起给出，即同时指定曲线的颜色和线型．

例如：     t=-3.14:0.2:3.14;

x=sin(t);  y=cos(t);

plot(t,x, '+r',t,y, '-b')



图4.1.3.1不同线型、颜色的sin,cos图形

在一个图形上可以加网格、标题、x轴标记、y轴标记，用下列命令完成这些工作。

>> x=linspace(0,2\*pi,30); y=sin(x); z=cos(x);

>> plot(x,y,x,z)

>> grid

>> xlabel(‘Independent Variable X’)

>> ylabel(‘Dependent Variables Y and Z’)

>> title(‘Sine and Cosine Curves’)

它们产生图5-5：

a2

图5-5

也可以在图形的任何位置加上一个字符串，如用：

>> text(2.5,0.7,’sinx’)

表示在坐标x=2.5, y=0.7处加上字符串sinx。更方便的是用鼠标来确定字符串的位置，方法是输入命令：

>> gtext(‘sinx’)

在图形窗口十字线的交点是字符串的位置，用鼠标点一下就可以将字符串放在那里。

**3.3 坐标系的控制**

在缺省情况下MATLAB自动选择图形的横、纵坐标的比例，如果你对这个比例不满意，可以用axis命令控制，常用的有：

axis([xmin xmax ymin ymax]) [ ]中分别给出x轴和y轴的最大值、最小值

axis equal 或 axis(‘equal’) x轴和y轴的单位长度相同

axis square 或 axis(‘square’) 图框呈方形

axis off 或 axis(‘off’) 清除坐标刻度

还有axis auto axis image axis xy axis ij axis normal axis on axis(axis)

用法可参考在线帮助系统。

**3.4 多幅图形**

可以在同一个画面上建立几个坐标系, 用subplot(m,n,p)命令；把一个画面分成m×n个图形区域, p代表当前的区域号，在每个区域中分别画一个图,如

>> x=linspace(0,2\*pi,30); y=sin(x); z=cos(x);

>> u=2\*sin(x)**.**\*cos(x); v=sin(x)**.**/cos(x);

>> subplot(2,2,1),plot(x,y),axis([0 2\*pi –1 1]),title(‘sin(x)’)

>> subplot(2,2,2),plot(x,z),axis([0 2\*pi –1 1]),title(‘cos(x)’)

>> subplot(2,2,3),plot(x,u),axis([0 2\*pi –1 1]),title(‘2sin(x)cos(x)’)

>> subplot(2,2,4),plot(x,v),axis([0 2\*pi –20 20]),title(‘sin(x)/cos(x)’)

共得到4幅图形,见图5-6.

a3

图5-6

**三维作图**

**3.5 mesh(Z)语句**

mesh(Z)语句可以给出矩阵Z元素的三维消隐图，网络表面由Z坐标点定义，与前面叙述的x-y平面的线格相同，图形由邻近的点连接而成．它可用来显示用其它方式难以输出的包含大量数据的大型矩阵，也可用来绘制Z变量函数．

显示两变量的函数Z=f(x,y)，第一步需产生特定的行和列的x-y矩阵．然后计算函数在各网格点上的值．最后用mesh函数输出．

下面我们绘制sin(r)/r函数的图形．建立图形用以下方法：

x=-8:.5:8;

y=x';

x=ones(size(y))\*x;

y=y\*ones(size(y))';

R=sqrt(x.^2+y.^2)+eps;

z=sin(R)./R;

mesh(z)       %%  试运行 mesh(x,y,z)，看看与mesh(z)有什么不同之处？

各语句的意义是：首先建立行向量x，列向量y；然后按向量的长度建立1-矩阵；用向量乘以产生的1-矩阵，生成网格矩阵，它们的值对应于x-y坐标平面；接下来计算各网格点的半径；最后计算函数值矩阵Z．用mesh函数即可以得到图形．

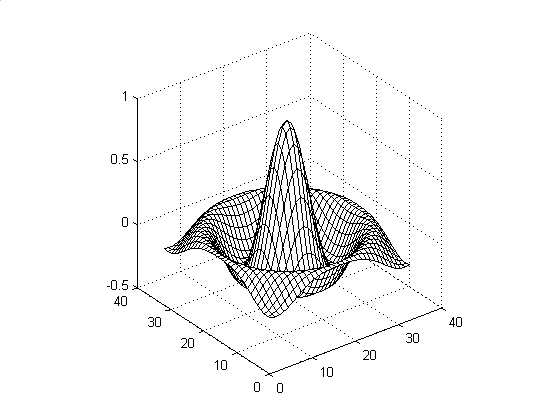


图4.2.1三维消隐图

第一条语句x的赋值为定义域，在其上估计函数；第三条语句建立一个重复行的x矩阵，第四条语句产生y的响应，第五条语句产生矩阵R（其元素为各网格点到原点的距离）．用mesh方法结果如上．

另外，上述命令系列中的前4行可用以下一条命令替代：

[x, y]=meshgrid(-8:0.5:8)

[返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/mat4.htm#实验#实验)

**3.6与mesh相关的几个函数**

(1) meshc与函数mesh的调用方式相同，只是该函数在mesh的基础上又增加了绘制相应等高线的功能．下面来看一个meshc的例子：

[x,y]=meshgrid([-4:.5:4]);

z=sqrt(x.^2+y.^2);

meshc(z)       %%  试运行 meshc(x,y,z)，看看与meshc(z)有什么不同之处？

我们可以得到图形：

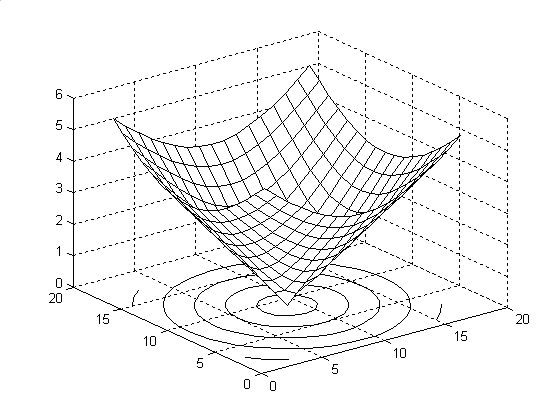


图4.2.2.1  meshc 图

地面上的圆圈就是上面图形的等高线．

(2) 函数meshz与mesh的调用方式也相同，不同的是该函数在mesh函数的作用之上增加了屏蔽作用，即增加了边界面屏蔽．例如：

[x,y]=meshgrid([-4:.5:4]);

z=sqrt(x.^2+y.^2);

meshz(z)       %%  试运行 meshz(x,y,z)，看看与meshz(z)有什么不同之处？

我们得到图形：

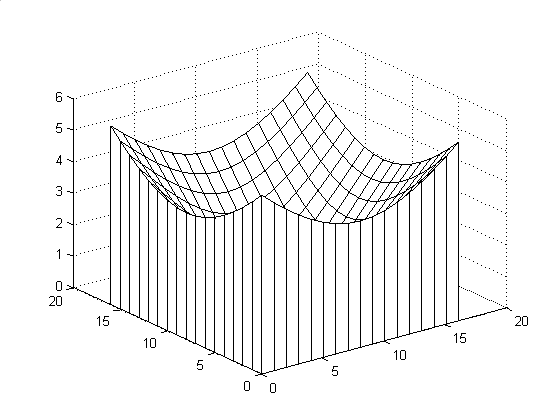


图4.2.2.2  meshz 图

[返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/mat4.htm#实验#实验)

**3.7 其它的几个三维绘图函数**

(1) 在**Matlab**中有一个专门绘制圆球体的函数sphere，其调用格式如下：

[x,y,z]=sphere(n)

此函数生成三个(n+1)×(n+1)阶的矩阵，再利用函数surf(x,y,z)可生成单位球面．

[x,y,z]=sphere   此形式使用了默认值n=20

sphere(n)   只绘制球面图，不返回值．

运行下面程序：

sphere(30);

axis square;

我们得到球体图形：

a7

图4.2.3.1  球面图

若只输入sphere画图，则是默认了n=20的情况．

(2) surf函数也是**Matlab**中常用的三维绘图函数．其调用格式如下：

surf(x,y,z,c)

输入参数的设置与mesh相同，不同的是mesh函数绘制的是一网格图，而surf绘制的是着色的三维表面．**Matlab**语言对表面进行着色的方法是，在得到相应网格后，对每一网格依据该网格所代表的节点的色值（由变量c控制），来定义这一网格的颜色．若不输入c，则默认为c=z．

我们看下面的例子：

%绘制地球表面的气温分布示意图．

[a,b,c]=sphere(40);

t=abs(c);    %求绝对值

surf(a,b,c,t);

axis equal

colormap('hot')

我们可以得到图形如下：

a8

图4.2.3.2  等温线示意图

[返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/mat4.htm#实验#实验)

**4. 特殊函数与图形**

[**一、问题背景与实验目的**](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#一、问题背景与实验目的)

[**二、相关函数（命令）及简介**](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#二、相关函数（命令）及简介)

[**三、实验内容**](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#三、实验内容)

[**四、自己动手**](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#四、自己动手)

**一、问题背景与实验目的**

著名的Riemann函数大家都很熟悉了，但是关于它的图像你是否清楚呢？除了最上面那几点，其他都很难画吧？你想不想看看下面那些“挤在一起”的点是怎样分布的呢？还有几何中的马鞍面、单叶双曲面等是怎样由直线生成的，是不是也想目睹一下呢？这些，都离不开绘图．

实际上绘图一直是数学中的一种重要手段，借助图形，往往可以化繁为简，使抽象的对象得到明白直观的体现．比如函数的基本性质，一个图形常可以使之一目了然，非常有效．它虽不能代替严格的分析与证明，但在问题的研究过程中，可以帮助研究人员节约相当一部分精力．此外，它还可以使计算、证明、建模等的结果得到更明白易懂的表现，有时，这比科学论证更有说服力．

同时，数学的教学与学习过程也离不开绘图．借助直观的图形，常可以使初学者更容易接受新知识．如数学分析中有不少函数，其解析式着实让人望而生畏，即使对其性质作了详尽的分析，还是感到难明就里；但如果能看到它的图形，再配合理论分析，则问题可以迎刃而解．又如在几何的学习中，会遇到大量的曲线与曲面，也离不开图形的配合．

传统的手工作图，往往费力耗时，效果也不尽理想．计算机恰恰弥补了这个不足，使你可以方便地指定各种视角、比例、明暗，从各个角度进行观察．

本实验通过对函数的图形表示和几个曲面（线）图形的介绍，一方面展示它们的特点，另一方面，也将就Matlab软件的作图功能作一个简单介绍．大家将会看到，Matlab 的作图功能非常强大．[返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#实验一__特殊函数与图形)

**二、相关函数（命令）及简介**

1．平面作图函数：plot，其基本调用形式：

plot(x,y,s)

以x作为横坐标，y作为纵坐标．s是图形显示属性的设置选项．例如：

x=-pi:pi/10:pi;

y=sin(x);

plot(x,y,'--rh','linewidth',2,'markeredgecolor','b','markerfacecolor','g')

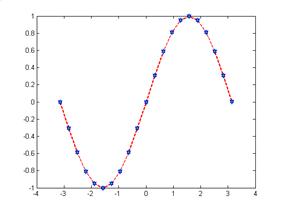


图1

在使用函数plot时，应当注意到当两个输入量同为向量时，向量x与y必须维数相同，而且必须同是行向量或者同是列向量．

绘图时，可以制定标记的颜色和大小，也可以用图形属性制定其他线条特征，这些属性包括：

linewidth            指定线条的粗细．

markeredgecolor      指定标记的边缘色

markerfacecolor      指定标记表面的颜色．

markersize           指定标记的大小．

若在一个坐标系中画几个函数，则plot的调用格式如下：

plot(x1,y1,s1,x2,y2,s2,……)

2．空间曲线作图函数：plot3，它与plot相比，只是多了一个维数而已．其调用格式如下：

plot3(x,y,z,s)．例如：

x=0:pi/30:20\*pi;

y=sin(x);z=cos(x);

plot3(x,y,z)

得到三维螺旋线：

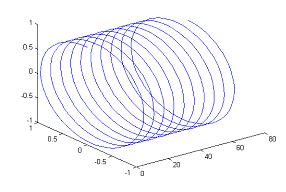


图2

3．空间曲面作图函数：

（1）mesh函数．绘制彩色网格面图形．调用格式：

mesh(z)，mesh(x,y,z)和mesh(x,y,z,c)．

其中，mesh(x,y,z,c)画出颜色由c指定的三维网格图．若x、y均为向量，则length(x)=n，length(y)=m，[m,n]=size(z)．

（2）surf在矩形区域内显示三维带阴影曲面图．调用格式与mesh类似．

（3）ezmesh用符号函数作三维曲面网格图．

调用格式：

ezmesh(x,y,z)

其中x = x(s,t), y = y(s,t),z = z(s,t)．画图区域默认为： -2\*pi < s < 2\*pi 且-2\*pi < t < 2\*pi.

或者用格式：

ezmesh(x,y,z,[smin,smax,tmin,tmax])

（4）ezsurf用符号函数作三维曲面图．调用格式与ezmesh类似．

（5）sphere画球体命令．

4．meshgrid，调用格式：

[x,y]=meshgrid(m,n)，

这里的m，n为给定的向量，可以定义网格划分区域和划分方法．矩阵x和矩阵y是网格划分后的数据矩阵．

5．图像的修饰与其他函数：

（1）axis equal 控制各个坐标轴的分度，使其相等；

（2）colormap设置绘图颜色．

调用格式：

colormap([r g b])

其中r,g,b都是0-1之间的数．

或者用格式：

colormap(s)

s为颜色映像．下面举几个常用的例子：

|  |  |  |  |
| --- | --- | --- | --- |
| 颜色映像 | 相应的颜色系 | 颜色映像 | 相应的颜色系 |
| autumn | 红黄色系 | hsv | 色调饱和色系 |
| gray | 线性灰色系 | hot | 黑红黄白色系 |
| cool | 青和洋红色系 | pink | 柔和色系 |

（3）grid网格函数   grid on添加网格．grid off取消网格．

（4）find找出符合条件的元素在数组中的位置．调用格式：

y=find(条件)

例如：输入：

a=[4  5  78  121  4  665  225  4  1];

b=find(a>7)

   输出：b =3     4     6     7                                             [返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#实验一__特殊函数与图形)

**三、实验内容**

数学分析中，特别是积分部分，我们接触了不少有趣的函数，由于其中有的不是一一对应的，用上面的方法无法画出它们的图像，这时就只能用参数了．

此外还有些图形只能用参数来画，比如空间曲线，在计算机上不接受“两个曲面的交线”这种表示，所以也只能用参数来实现．

用参数方式作图的关键在于找出合适的参数表示，尤其是不能有奇点，最好也不要用到开方．所以要找的参数最好是有几何意义的．当然这也不可一概而论，需要多积累经验．

1．利用函数plot在一个坐标系中画以下几个函数图像，要求采用不同颜色、不同线形、不同的符号标记．函数为：http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image004.gif．

程序如下：

t=0:pi/20:2\*pi;

x=sin(t);

y=cos(t);

z=sin(2\*t);

plot(t, x, '--k\*', t, y, '-rs', t, z, ':bo')

图像如下：

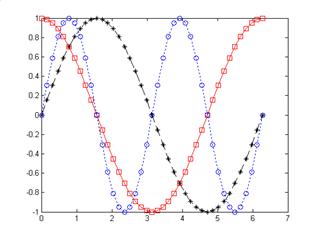


图3

2．绘制类似田螺线的一条三维螺线(方程自己设计)．

程序如下：

t=0:.1:30;

x=2\*(cos(t)+t.\*sin(t));

y=2\*(sin(t)-t.\*cos(t));

z=1.5\*t;

plot3(x,y,-z)  %取 –z 主要是为了画图看起来更清楚

axis equal

图像如下：

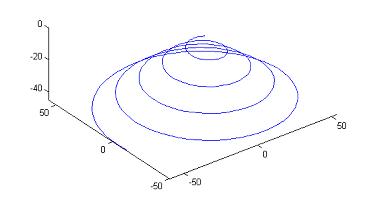


图4

3．利用函数http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image007.gif，绘制一个墨西哥帽子的图形．

程序如下：

[a,b]=meshgrid(-8:.5:8);     %先生成一个网格

c=sqrt(a.^2+b.^2)+eps;

z=sin(c)./c;

mesh(a,b,z)

axis square

图像如下：

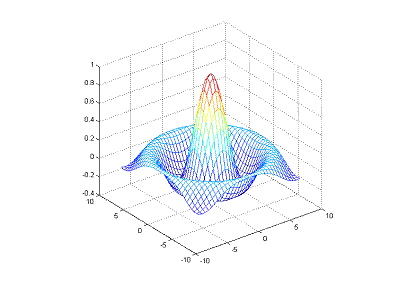


图5

思考：这里的 eps 是什么？其作用是什么？

4．利用surf绘制马鞍面图形（函数为：http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image009.gif）．

程序如下：

 [x,y]=meshgrid(-25:1:25,-25:1:25);

z=x.^2/9-y.^2/4;

surf(x,y,z)

title('马鞍面')

grid off

图像如下：

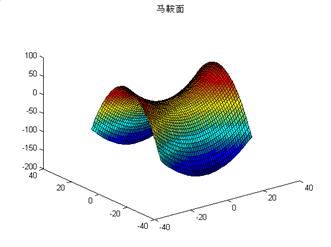
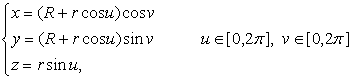


图6

5．分别用ezmesh和ezsurf各绘制一个圆环面，尝试将两个圆环面放在一个图形界面内，观察它们有什么不同之处．

       提示：圆环面的方程为： http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image011.gif，而圆环面的参数方程为：



程序参见附录1．

图像如下：

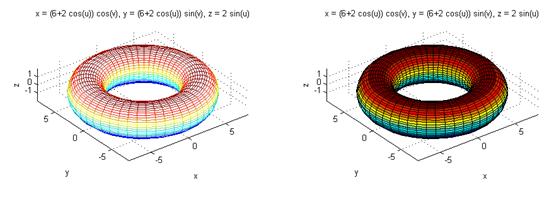
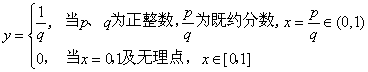


图7

6．绘制黎曼函数图形，加深对黎曼函数的理解．

说明：黎曼函数的定义为



程序参见附录2．

图像如下：

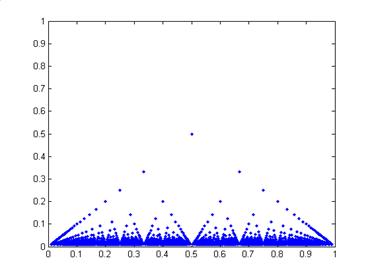


图8

[返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#实验一__特殊函数与图形)

**四、自己动手**

1．作出下图所示的三维图形：

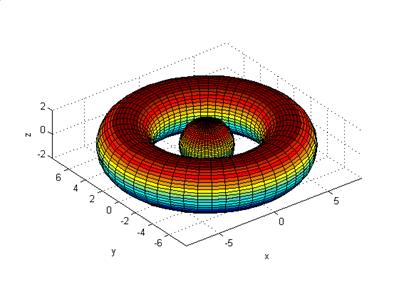


图9

提示：图形为圆环面和球面的组合.

2．作出下图所示的墨西哥帽子及其剪裁图形：

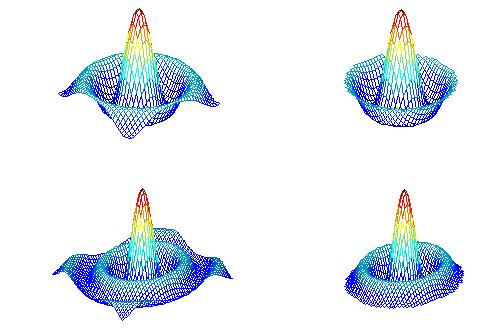


图10

3．画出球面、椭球面、双叶双曲面、单叶双曲面．

4．若要求田螺线的一条轴截面的曲边是一条抛物线：http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image018.gif时http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image019.gif．试重新设计田螺线的参数方程，并画出该田螺线．

5．作出下图所示的马鞍面（颜色为灰色，并有一个标题：“马鞍面”）：

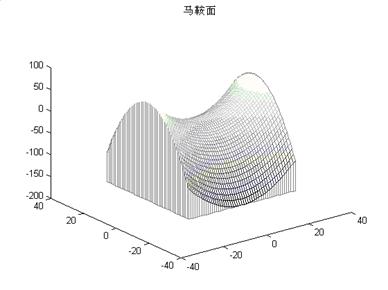


图11

6．绘制图8所示的黎曼函数图形，要求分母的最大值http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.files/image021.gif的数值由键盘输入（提示：使用input语句）．[返回](http://www.math.ecnu.edu.cn/sxsykc/jxnr/sy1.htm#实验一__特殊函数与图形)

**5、搜寻路径**

在前一节中，test.m所在的目录是d:\mlbook。如果不先进入这个目录，MATLAB就找不到你要执行的M档案。如果希望MATLAB不论在何处都能执行test.m，那麽就必须将d:\mlbook加入MATLAB的搜寻路径（Search path）上。要检视MATLAB的搜寻路径，键入path即可：

path

MATLABPATH

d:\matlab5\toolbox\matlab\general

d:\matlab5\toolbox\matlab\ops

d:\matlab5\toolbox\matlab\lang

d:\matlab5\toolbox\matlab\elmat

d:\matlab5\toolbox\matlab\elfun

d:\matlab5\toolbox\matlab\specfun

d:\matlab5\toolbox\matlab\matfun

d:\matlab5\toolbox\matlab\datafun

d:\matlab5\toolbox\matlab\polyfun

d:\matlab5\toolbox\matlab\funfun

d:\matlab5\toolbox\matlab\sparfun

d:\matlab5\toolbox\matlab\graph2d

d:\matlab5\toolbox\matlab\graph3d

d:\matlab5\toolbox\matlab\specgraph

d:\matlab5\toolbox\matlab\graphics

d:\matlab5\toolbox\matlab\uitools

d:\matlab5\toolbox\matlab\strfun

d:\matlab5\toolbox\matlab\iofun

d:\matlab5\toolbox\matlab\timefun

d:\matlab5\toolbox\matlab\datatypes

d:\matlab5\toolbox\matlab\dde

d:\matlab5\toolbox\matlab\demos

d:\matlab5\toolbox\tour

d:\matlab5\toolbox\simulink\simulink

d:\matlab5\toolbox\simulink\blocks

d:\matlab5\toolbox\simulink\simdemos

d:\matlab5\toolbox\simulink\dee

d:\matlab5\toolbox\local

此搜寻路径会依已安装的工具箱（Toolboxes）不同而有所不同。要查询某一命令是在搜寻路径的何处，可用which命令：

which expo

d:\matlab5\toolbox\matlab\demos\expo.m

很显然c:\data\mlbook并不在MATLAB的搜寻路径中，因此MATLAB找不到test.m这个M档案：

which test

c:\data\mlbook\test.m

要将d:\mlbook加入MATLAB的搜寻路径，还是使用path命令：

path(path, 'c:\data\mlbook');

此时d:\mlbook已加入MATLAB搜寻路径（键入path试看看），因此MATLAB已经"看"得到

test.m:

which test

c:\data\mlbook\test.m

现在我们就可以直接键入test，而不必先进入test.m所在的目录。

小提示：如何在其启动MATLAB时，自动设定所需的搜寻路径？ 如果在每一次启动MATLAB後都要设定所需的搜寻路径，将是一件很麻烦的事。有两种方法，可以使MATLAB启动後 ，即可载入使用者定义的搜寻路径：

1.MATLAB的预设搜寻路径是定义在matlabrc.m（在c:\matlab之下，或是其他安装MATLAB 的主目录下），MATLAB每次启动後，即自动执行此档案。因此你可以直接修改matlabrc.m ，以加入新的目录於搜寻路径之中。

2.MATLAB在执行matlabrc.m时，同时也会在预设搜寻路径中寻找startup.m，若此档案存在，则执行其所含的命令。因此我们可将所有在MATLAB启动时必须执行的命令（包含更改搜寻路径的命令），放在此档案中。

每次MATLAB遇到一个命令（例如test）时，其处置程序为：

1.将test视为使用者定义的变数。

2.若test不是使用者定义的变数，将其视为永久常数 。

3.若test不是永久常数，检查其是否为目前工作目录下的M档案。

4.若不是，则由搜寻路径寻找是否有test.m的档案。

5.若在搜寻路径中找不到，则MATLAB会发出哔哔声并印出错误讯息。

以下介绍与MATLAB搜寻路径相关的各项命令。