Matlab 引例

MATLAB 是由美国 Mathworks 公司于 1984 年推出的一套高性能的数值计算和可视化软件，在工程计算、系统实时仿真等领域都得到了广泛应用。它将计算、可视化和编程功能集成于一个使用非常方便的环境之中，而计算及结果用人们非常熟悉的数学符号表示，易于上手。目前最新的版本为 R2006a。

MATLAB 可以广泛用于数学计算、算法开发研究、建模、仿真和试制、数据分析处理及可视化、科学和工程作图、应用系统的开发，包括建立用户界面。

由于 MATLAB 拥有大量的常用数学函数、工具箱，基本包括了现今数学、物理及工程应用领域的常用函数，如能熟练掌握，无论在编程时间或程序量上，都远远低于 Fortran、C 等高级语言，尤其在处理矩阵和向量计算方面更具有无可比拟的优势。

我们在数学建模竞赛中，由于只有短短的三到四天，而论文的评判不仅注重计算的结果，更注重模型的创造性等很多方面，因此比赛中把大量的时间花费在编写和调试程序上只会喧宾夺主，是很不值得的，这时使用 MATLAB 可以很大程度上的方便计算、节省时间，使我们将精力更多的放在模型的完善上，所以是较为理想的。

这里我们仅举了一些运用 MATLAB 的例子，这些都是在数学建模中时常遇到的问题，希望能抛砖引玉，帮助同学们在最短的时间内方便、快捷的使用 MATLAB 解决数学建模中的问题，并善用这一工具。由于 MATLAB 本身库函数庞大、涉及领域众多，这里我们不能一一列举，仅在例子后列举了一些在数学建模中较常用的函数，对于 MATLAB 的详细使用方法及其它函数请同学们自行查找相关书籍或 MATLAB 自带的帮助文件。如，想知道 roots()函数的详细用法，可以在 MATLAB 主界面进行如下输入：

```
>>help roots
```

将得到如下的显示：

```
ROOTS    Find polynomial roots.
    ROOTS(C) computes the roots of the polynomial whose coefficients
    are the elements of the vector C. If C has N+1 components,
    the polynomial is C(1)*X^N + ... + C(N)*X + C(N+1).

    See also POLY, RESIDUE, FZERO.

 Overloaded methods
    help gf/roots.m
    help localpoly/roots.m
```

其中给出了 roots()函数的使用方法及与之相关的函数等。由于 MATLAB 本身语法和结构要求不严格，建议同学们在使用中尽可能的使用帮助文件，以便减少语法错误，以及由于不正确或不恰当地引用 MATLAB 函数而产生的错误。

附：常用控制命令：

```
clc        %清屏
clear      %清变量
save       %保存变量
load       %导入变量
```

例 1：利用公式直接进行赋值计算：

本金 P 以每年 n 次，每次 i%的增值率（n 与 i 的乘积为每年增值额的百分比）增加，当增加到 r×P 时所花费的时间为：

$$T = \frac{\ln r}{n\ln(1+0.01i)} \text{年} \quad (\text{令 } r=2,\ i=0.5,n=12)$$

求所花费的时间 T。

MATLAB 的表达形式及结果如下（%后的部分为注释，Matlab 会忽略%后的部分，这里的注释旨在帮助理解，计算时可以不必输入）：

```
>> r=2;i=0.5;n=12;        %变量赋值
>> T=log(r)/(n*log(1+0.01*i))
```

计算结果显示为：

T = 11.5813

即，所花费的时间为 T=11.5813 年。

分析：上面的问题是一个利用公式直接进行赋值计算问题，实际中我们用计算器也可以完成；若其中的某个变量改变，我们就重新计算；但若变量在某个范围变化取很多值时，使用计算器将是一个枯燥、易犯错的巨大工程；这时使用 MATLAB，你将倍感方便，它不仅能帮你轻松得到结果，其绘图功能还能将结果轻松的显示出来，变量之间的变化规律将一目了然。

若 r 在[1,9]变化，i 在[0.5,3.5]变化；我们将 MATLAB 的表达式作如下改动，结果如图 1。

```
r=1:0.5:9;
i=0.5:0.5:3.5;
n=12;
p=1./(n*log(1+0.01*i));
T=log(r')*p;
plot(r,T)
xlabel('r')       %给 x 轴加标题
ylabel('T')       %给 y 轴加标题
q=ones(1,length(i));
text(7*q-0.2,[T(14,1:5)+0.5,T(14,6)-0.1,T(14,7)-0.9],num2str(i'))
```

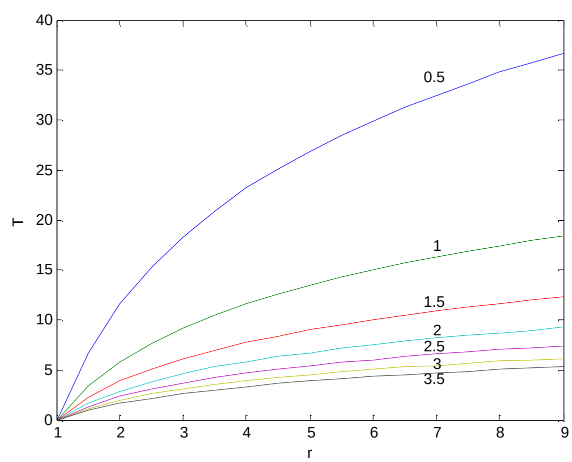


图 1

从图 1 中我们不仅可以看到 T 随 r 的变化规律，而且还能看到 i 的不同取值对 T—r 曲线的影响（图中的六条曲线分别代表 i 的不同取值）。

Matlab 的矩阵运算功能是其优势，只要结合线性代数的有关知识，并善加利用，就可以避免很多原本需要利用循环语句才能实现的计算。在近两年的全国竞赛中，出现的数据表格越来越多、数据量越来越大；处理这些数据，传统的计算器显然已不能满足，即使运用 Excel，其数据处理功能也十分有限；而 Matlab 不仅可以实现对整个矩阵的整体运算，还可以对其中的部分数据，如：行、列、对角线等，进行提取、求和等很多复杂计算，结合其绘图功能，使对数据间变化的分析更加直观、有效。让你轻松、快速的尝试更多的方法，从而找到其中最适合的。

附：结果的显示格式函数：

format long	%15 位数字
format bank	%小数点后 2 位
format +	%显示+、—或 0
format short e	%5 位科学计数法
format long e	%15 位科学计数法
format rat	%最接近的有理数
format hex	%16 进制
format short	%小数点后 4 位

特殊变量和常数：

pi	%圆周率 π
eps	%浮点数的精确度
inf	%无穷大的值，即一个数除以零的结果
NaN	%不定值，例如 0/0 或 inf/inf
i,j	%虚数单位

基本函数：

round	%四舍五入取整
floor	%向－∞方向取整
ceil	%向+∞方向取整
fix	%向 0 方向取整
real	%取实部
imag	%取虚部
conj	%复数共轭
angle	%取幅角
rats	%有理逼近
rem(x,y)	%求作除法后的剩余数
gcd(x,y)	%整数 x,y 的最大公约数
lcm(x,y)	%整数 x,y 的最小公倍数
exp(x)	% e^x （矩阵函数用 expm(x)）
sqrt(x)	% $\sqrt{x}$ （矩阵函数用 sqrtm(x)或 funm(x,’fn’)）
log(x)	%ln(x)或 $\log_e(x)$ （矩阵函数 logm(x)）
log10(x)	% $\log_{10}(x)$
abs(x)	%|x|或模
sign(x)	%符号函数
size(x)	%矩阵大小
length(x)	%向量长度
dot(x,y)	%向量点乘
cross(x,y)	%向量叉乘
rot90(x)	%将矩阵 x 逆时针方向旋转 90 °
fliplr(x)	%矩阵 x 左右翻转
flipud(x)	%矩阵 x 上下翻转
intersect(a,b)	%两个集合的交集
union(a,b)	%两个集合的并集
setfiff(a,b)	%两个集合的差集

gamma(n+1)　　%n!
polyarea　　%决定一闭合曲线内的面积

例 2：已知多项式求根：

已知多项式为：

$$h = x^6 - 10x^5 + 31x^4 - 10x^3 - 116x^2 + 200x - 96$$

求其根。

分析：对多项式求根问题，我们常用 roots()函数。MATLAB 的表达形式及结果如下：

```
>> h=roots([1 -10 31 -10 -116 200 -96])    %中括号内为多项式系数由高阶到常数
```

计算结果显示为（其中 i 为虚数单位）：

```
h =
   -2.0000
     4.0000
     3.0000
     2.0000 + 0.0000i
     2.0000 - 0.0000i
     1.0000
```

如果已知多项式的根，求多项式，用 poly()函数。对上面得到的 h 的值求多项式，其 MATLAB 的表达形式及结果如下：

```
>>h=[-2.0000 4.0000 3.0000 2.0000+0.0000i 2.0000-0.0000i 1.0000];
>>c=poly(h)
```

计算结果显示为：

```
c =    1     -10      31     -10    -116     200     -96
```

常用函数求根：对型如：

$$f(x) = \cos(x)\cosh(x) - 1$$

的非线性函数求 $f(x) = 0$ 时，x 的值，MATLAB 中我们常用 fzero()函数。

MATLAB 的表达形式及结果如下：

```
>>qcc=inline('cos(x).*cosh(x)-1','x');
>>options=optimset('display','off');
>>w=fzero(qcc,4,options)
```

计算结果显示为：

```
w =    4.7300
```

上面的函数为多值函数，这里我们只求了 x 在 4 附近的根。下面的 MATLAB 的表达形式是求上面的函数最小的 5 个非零根。

```
qcc=inline('cos(x).*cosh(x)-1','x');
options=optimset('display','off');
%x=linspace(0,20);
%plot(x,qcc(x))
```

```
%axis([0 20 -10 10])
x0=[3 5];
for n=1:5
    q(n)=fzero(qcc,x0,options);
    x0=[1.05*q(n) q(n)+4];
end
disp(['Lowest five roots are:' num2str(q)])
```

MATLAB 显示的结果如下：

```
Lowest five roots are: 4.73004    7.8532    10.9956    14.137    17.2788
```

在实际的数学建模问题中，我们还常将数值型的问题转化为符号型的问题，以获得一般性的解；在后面的例子及函数中，我们将看到这种运用。

附：多项式计算常用函数：

```
conv(a,b)        %多项式乘法
deconv(a,b)      %多项式除法
polyder(x)       %多项式求导
```

例 3：方程组的求解：

求解下面的方程组：

$$
\begin{aligned}
8x_1 + x_2 + 6x_3 &= 7.5 \\
3x_1 + 5x_2 + 7x_3 &= 4 \\
4x_1 + 9x_2 + 2x_3 &= 12
\end{aligned}
$$

分析：对于线性方程组求解，常用线性代数的方法，把方程组转化为矩阵进行计算。

MATLAB 的表达形式及结果如下：

```
>> a=[8 1 6;3 5 7;4 9 2];    %建立矩阵
>> b=[7.5;4;12];
>> x=a\b        %求方程组的解
```

计算结果显示为：

```
x =    1.2931    0.8972    -0.6236
```

形如 AX=b 的线性方程组，还可以用以下几种形式求解，同学们可以自己试一试：

linsolve(A,b); sym(A)\sym(b); inv(A)*b; pinv(A)*b

附：常用矩阵函数：

```
det(x)       %行列式的值
inv(x)       %矩阵的逆
pinv(x)      %x 的广义逆矩阵
orth(x)      %正交化
poly(x)      %特征多项式
rank(x)      %矩阵的秩
```

trace(x) %矩阵的迹（对角元素之和）
norm(x) %矩阵的范数
x’ %矩阵转置或产生复数共轭转置（不进行共轭用“.’”）
solve(‘eqn1’,’eqn2’,’var1’,’var2’) %对变量 var1,var2 求解方程组 eqn1(var1)=0, eqn2(var2)=0
fzero(fun,x0) %非线性方程的解
fsolve(fun,x0) %非线性方程组的解

例 4：数值积分问题：

设 $y(t)=e^{-0.5t}\sin(t+\frac{\pi}{6})dt$，求 $S=\int_0^{3\pi} y(t)dt$。

分析：对于型如上面的数值积分问题，常用 quad()函数。MATLAB 求解的具体方法如下：

```
ff=inline('exp(-0.5*t).*sin(t+pi/6)','t');          %建立临时函数
S=quad(ff,0,3*pi)
```

计算结果显示为：

S = 0.9008

在数学建模的问题中，很多时候我们得到的积分函数是不可解的，这时就需要我们对函数进行转化，如进行泰勒展开等，然后再利用 quad()函数，以得到合适精度的近似解。或者直接利用一些数值计算的近似公式，通过编程取得需要的结果。

附：常用数值计算函数：

interp1 %一元插值
interp2 %二元插值
trapz %梯形法求数值积分
quad2dggen %任意区域上二元函数的数值积分
ode45 %常微分方程组初值问题的数值解

例 5：数据拟合与二维绘图：

浓度变化规律：

在化学反应中，为研究某化合物的浓度随时间的变化规律，测得一组数据如下表：

T（分）	1	2	3	4	5	6	7	8
y	4	6.4	8.0	8.4	9.28	9.5	9.7	9.86
T（分）	9	10	11	12	13	14	15	16
y	10	10.2	10.32	10.42	10.5	10.55	10.58	10.6

分析：在数学建模竞赛中，我们常会遇到这种数据表格问题，如果我们仅凭眼睛观察，很难看到其中的规律，也就更难写出有效的数学表达式从而建立数学模型。因此可以利用 MATLAB 的拟合函数，即 polyfit() 函数，并结合 MATLAB 的绘图功能(利用 plot()函数)，得到直观的表示。

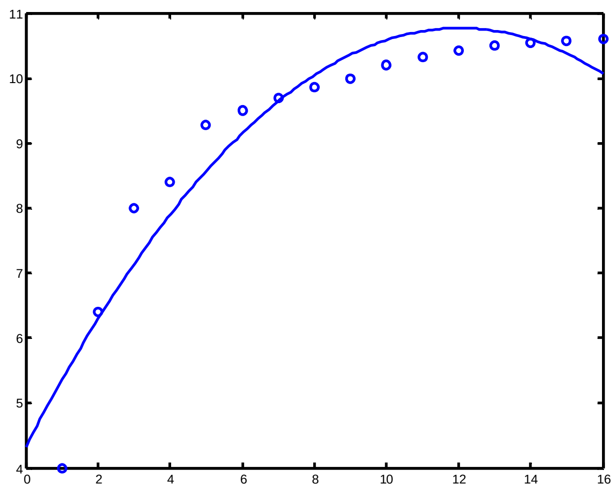


图 2 由程序得到的浓度 y 的拟合曲线与实测数据的比较

MATLAB 的表达形式如下（这里我们使用的是 MATLAB 的程序编辑器）：

```
t=[1:16];   %数据输入
y=[4 6.4 8 8.4 9.28 9.5 9.7 9.86 10 10.2 10.32 10.42 10.5 10.55 10.58 10.6];
plot(t,y,'o')   %画散点图
p=polyfit(t,y,2)   %二次多项式拟合
hold on
xi=linspace(0,16,160);   %在[0,16]等间距取 160 个点
yi=polyval(p,xi);   %由拟合得到的多项式及 xi，确定 yi
plot(xi,yi)   %画拟合曲线图
```

以上函数得到图 2；其中显示的结果

```
>> p =   -0.0445    1.0711    4.3252
```

p 的值表示二阶拟合得到的多项式为：

$$y = -0.0445t^2 + 1.0711t + 4.3253$$

下面是用 lsqcurvefit()函数，即**最小二乘拟合方法**的 Matlab 表达：

```
t=[1:16];
y=[4 6.4 8 8.4 9.28 9.5 9.7 9.86 10 10.2 10.32 10.42 10.5 10.55 10.58 10.6];
x0=[0.1,0.1,0.1];
zuixiao=inline('x(1)*t.^2+x(2)*t+x(3)','x','t');
x=lsqcurvefit(zuixiao,x0,t,y)   %利用最小二乘拟合
```

其显示的结果为：

```
x =   -0.0445    1.0711    4.3252
```

可以看出其得到的结果与 polyfit 函数的结果相同。这说明在多项式拟合问题上这两个函数的效果是相同的。下面的一个例子将体现 lsqcurvefit()函数的优势。

应力与应变关系：

在物理学中，为研究某种材料应力与应变的关系，测得一组数据如下表：

应力	925	1125	1625	2125	2625	3125	3625
应变	0.11	0.16	0.35	0.48	0.61	0.71	0.85

如果假定应力与应变有如下关系：(σ 为应力值， ε 为应变值)

$$\varepsilon = a + b\ln\sigma$$

试计算 a、b 的值。

MATLAB 的表达形式如下：

```
x=[925,1125,1625,2125,2625,3125,3625];
y=[0.11,0.16,0.35,0.48,0.61,0.71,0.85];
plot(x,y,'o')
[p,resid1]=polyfit(x,y,2)
hold on
xi=linspace(700,3700,3000);
yi=polyval(p,xi);
plot(xi,yi)
x0=[0.1,0.1];
fff=inline('a(1)+a(2)*log(x)','a','x');
[a,resid2]=lsqcurvefit(fff,x0,x,y)
plot(xi,fff(a,xi),'r')
```

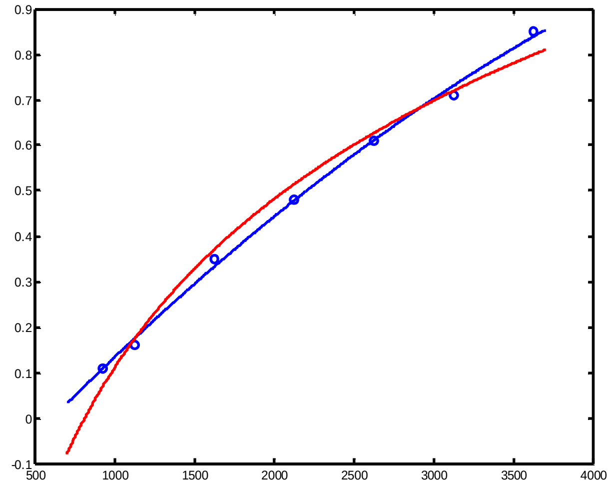


图 3 由程序得到的应力与应变关系曲线

以上函数得到图 3，图中蓝色曲线为利用 polyfit ()函数得到的曲线，红色曲线为利用 lsqcurvefit()函数得到的曲线；其显示的结果为：

```
>> p =     -0.0000      0.0004     -0.2266
resid1 = R: [3x3 double]
           df: 4
       normr: 0.0331
Optimization terminated successfully:
 First-order optimality less than OPTIONS.TolFun, and no negative/zero curvature detected
a =     -3.5810       0.5344
resid2 =    0.0064
```

其中 a 的值代表利用 lsqcurvefit()函数得到的关系为：

$$\varepsilon = -3.5810 + 0.5344\ln\sigma$$

resid1、resid2 分别代表运用 polyfit ()函数、lsqcurvefit()函数得到的残差。可以看出利用 lsqcurvefit()函数残差更小，即得到了更好的拟合效果。在数学建模的实际问题中，如果问题的机理不明，我们只能采用 polyfit ()函数，即多项式拟合的方法，以获得近似的数据描述函数；但如果通过分析，可以得到一些机理，那么采用最小二乘的方法将得到更好的效果，而且得到的拟合函数也更有意义。

极坐标图型：声源的远域辐射模式

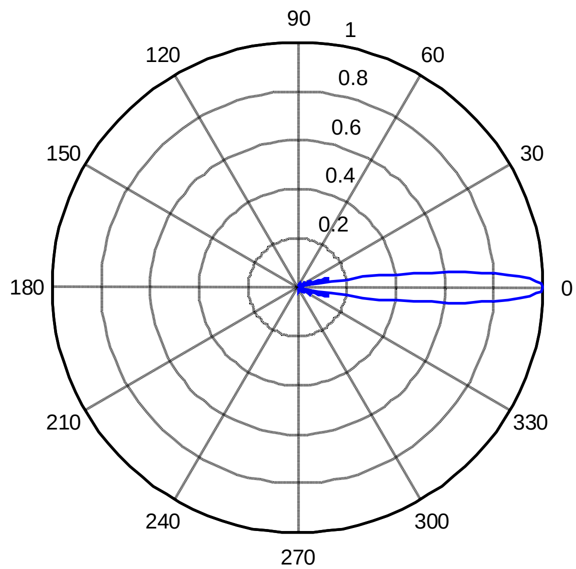
无限大挡板内的圆形活塞以频率 f 振动，距离其中心极远处的标准声压由下式给出：

$$p(r,\theta) = \left|\frac{J_1(ka\theta)}{ka\theta}\right| \qquad ka^2 << r \qquad 和 \qquad a << r$$

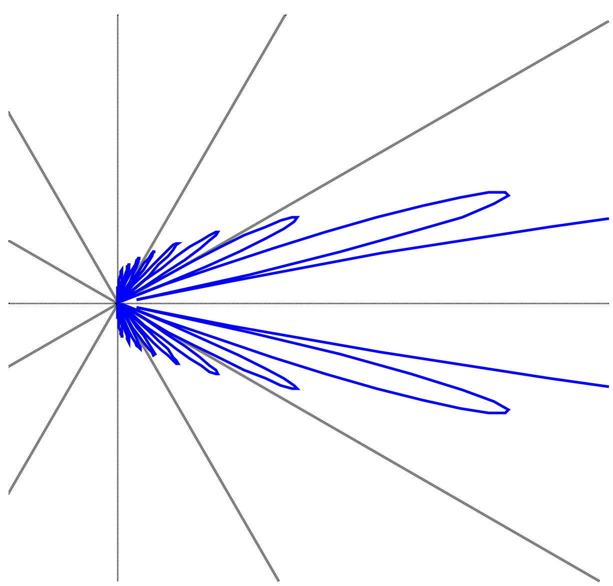
其中 r 为距离活塞圆心的极半径，θ 为 r 与挡板平面之间的角度，k 为波数，a 为活塞半径，$J_1(x)$ 为一类 1 阶贝塞尔函数。波数是频率为 f 的声波波长的倒数，因此 ka 是无量纲的。该模型是对扩音器声波角扩散的一个很好的近似。

创建一个标准辐射模型的极坐标，其中 $ka=6\pi$, $-\pi/2<\theta<\pi/2$。该例同时也可说明 zoom 函数的用法，正如下面所看到的，在 MATLAB 的极坐标图中，该辐射模型所展示的侧叶瓣不够清晰。程序如下：

```
theta=linspace(-pi/2,pi/2,300);
rad=abs(besselj(1,6*pi*theta)./(6*pi*theta));
polar(theta,rad/max(rad))
zoom on
```



(a) 辐射模型的极坐标表示



(b) 放大区域

图 4 辐射模型的极坐标图

不同 y 轴的多曲线绘制：

两个无量纲量分别是压力比和温度比，每一个都是高度比 h/h_L 的函数。比值由下式给出：

$$\frac{T}{T_o}=\left(1+\frac{h}{h_L}\right)^{1.5} \quad 和 \quad \frac{P}{P_o}=e^{-h/h_L}$$

下列程序说明了在同一标注图中如何用 plotyy 函数在 $0\le h/h_L\le 1$ 的范围内画出两条比率曲线的方法。程序执行结果如图 5 所示。

```
hhL=linspace(0,1,10);
tto=(1+hhL).^1.5;
ppo=exp(-hhL);
[ax,h1,h2]=plotyy(hhL,ppo,hhL,tto);
xlabel('h/h_L')
ylabel('P/P_o')
v=axis;
text(v(2)*1.06,v(3)+(v(4)-v(3))/2,'T/T_o','rotation',90)
text(v(1)+(v(2)-v(1))/5,v(3)+(v(4)-v(3))/1.6,'Pressure')
text(v(2)/1.6,v(4)/1.2,'Temperature')
set(h2,'Marker','s')
set(h1,'Marker','<')
```

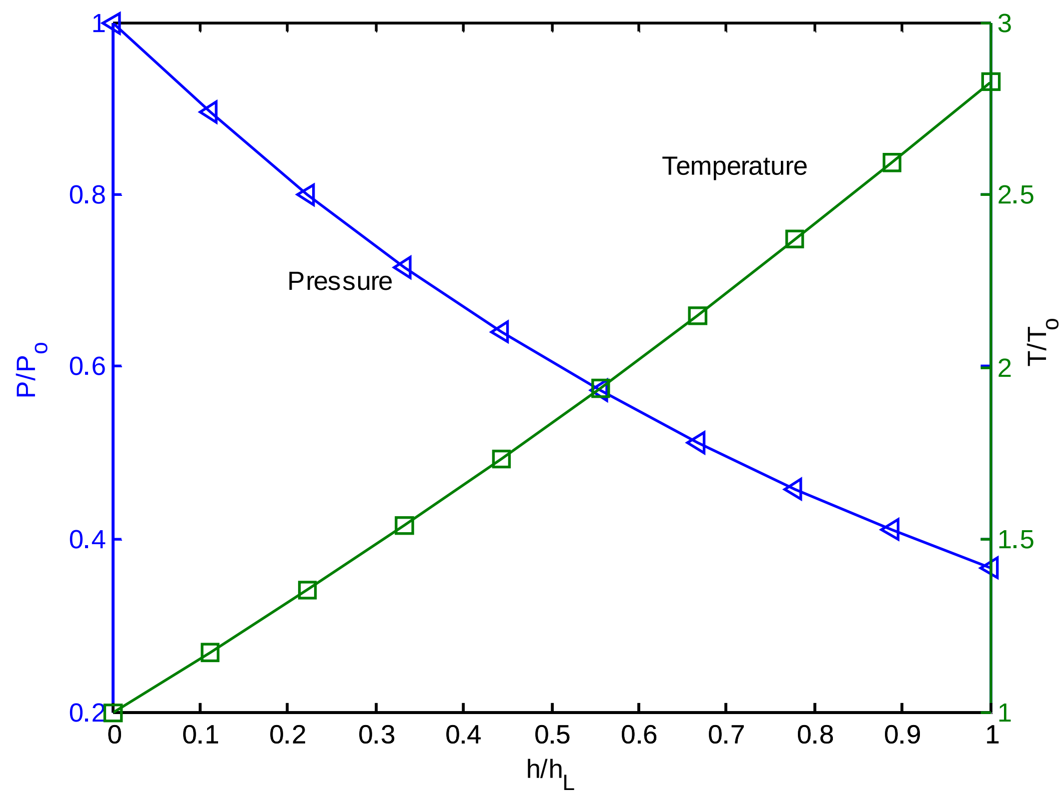


图 5

附：数组构造函数：

linspace(x,y,m)	%构造从 x 到 y，m 个线性分隔数组
logspace(x,y,m)	%构造从 10 的 x 次方到 10 的 y 次方，m 个对数分隔数组
zeros(m,n)	%生成 m×n 全零阵
ones(m,n)	%生成 m×n 全 1 阵
eye(m,n)	%生成 m×n 单位阵
rand(m,n)	%生成 m×n 均匀分布随机矩阵
randn(m,n)	%生成 m×n 正态分布随机矩阵
blkdiag(x)	%产生以输入元素为对角线元素的矩阵
y(x)	%按数组 x 的顺序提取数组 y 中的元素

例 6：隐函数的绘制：

上例中我们已经使用了 plot()函数用以绘制图形，但该函数只能绘制显函数，对于型如：

$$\frac{1}{y}-\ln(y)+\ln(-1+y)+x-\sin(x)=0$$

的复杂隐函数，很难转化为显函数并利用 plot()函数绘制图形，这时就可以用 ezplot()函数直接绘制其曲线。MATLAB 的表达形式如下，绘制的图形如图 6。

```
>> ezplot('1/y-log(y)+log(-1+y)+x-sin(x)')
```

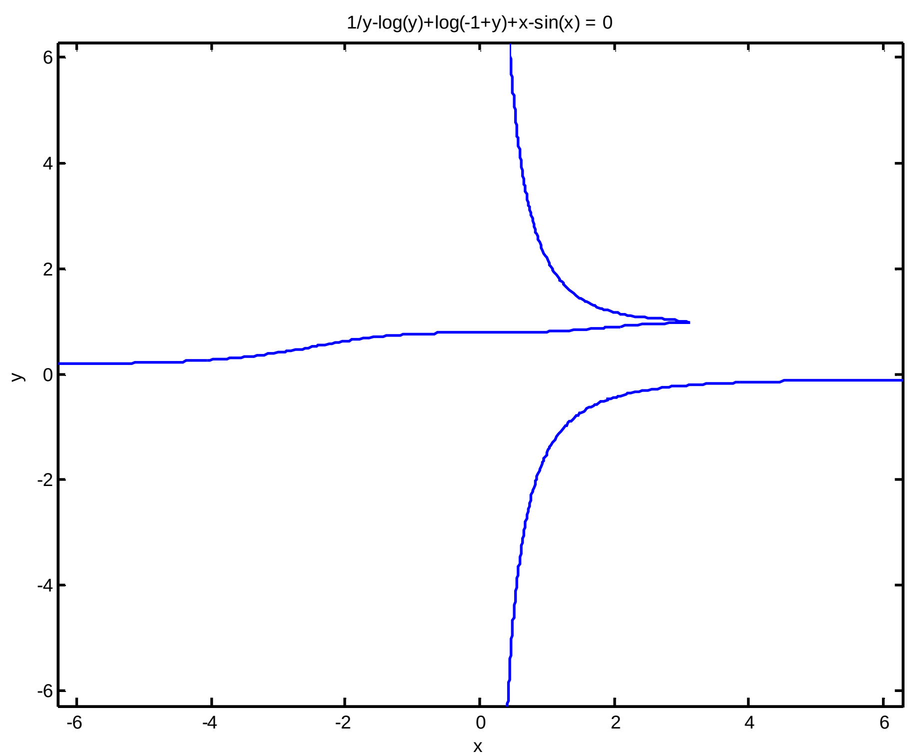


图 6 $\frac{1}{y}-\ln(y)+\ln(-1+y)+x-\sin(x)=0$ 曲线

如果是型如下面的**参数方程**：

$$x = \sin 3t \cos t\text{，}\quad y = \sin 3t \sin t\text{，}\quad t \in (0, \pi)$$

我们同样可以利用 ezplot()函数绘制其曲线。绘制的图形如图 7 所示。

MATLAB 的表达式形式如下：

```
>> ezplot('sin(3*t)*cos(t)','sin(3*t)*sin(t)',[0,pi])
```

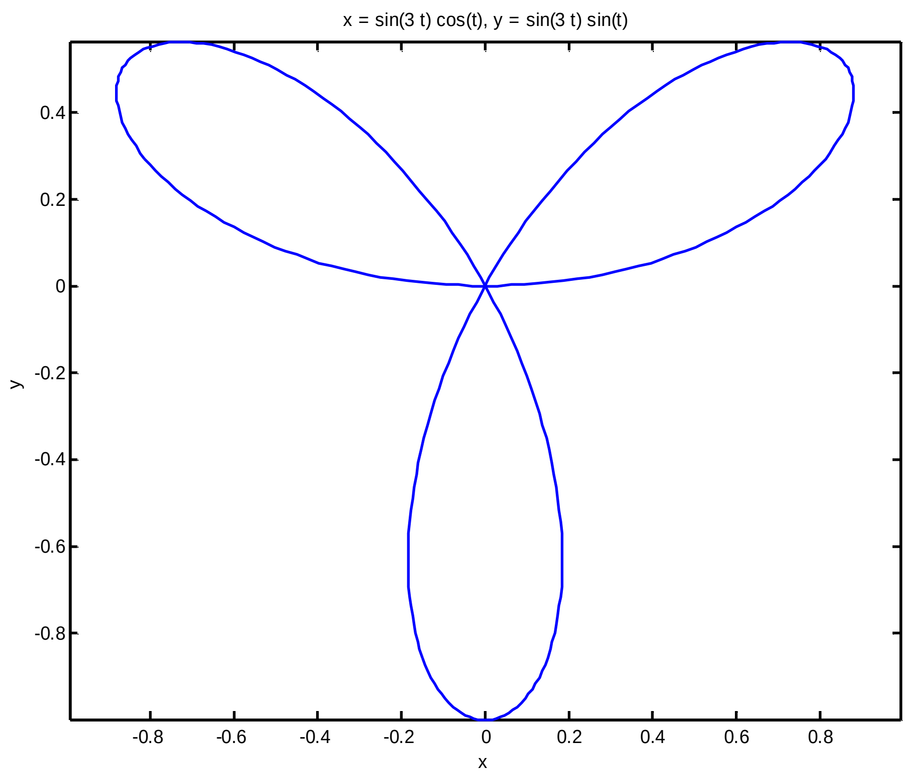


图 7　$x = \sin 3t \cos t$，　$y = \sin 3t \sin t$，　$t \in (0, \pi)$ 曲线

附：

常用绘图函数：

fplot()	%采用自适应步长绘制函数
bar()	%直方图
pie()	%饼图
polar()	%画极坐标图
feather()	%画速度向量图
fplot()	%一元函数绘图
loglog()	%双对数图
semilogx()	%x 轴对数图
semilogy()	%y 轴对数图
hist	%二维条形直方图
rose	%画角度直方图
figure(x)	%图像选择函数

subplot(m,n,p)	%将图形窗分割成 m×n 个子图，并选择第 p 个子图
hold on	%在原来的图上继续画图
grid on	%坐标网格线
axis()	%坐标轴取值范围
xlabel	%x 轴标记
ylabel	%y 轴标记
title	%标题
text()	%注释
legend	%添加图例
ezpolar()	%画符号函数极坐标图
ezcontour()	%画符号函数的等高线图
quiver()	%画矢量图
slice()	%画立体切片图

符号函数：

syms x y z	%定义 x,y,z 为符号变量
compose(x,y)	%x,y 的复合函数
finverse(x)	%x 的逆函数
expand(x)	%多项式展开
collect(S,v)	%对变量 v 合并同类项，S 为多项式
factor(x)	%符号因式分解
simplify(x)	%符号表达式的化简
poly2sym	%将多项式系数向量转化为带符号变量的多项式
limit(F,x,a)	%计算 x→a 时 F=F(x)的极限值
diff(S,’v’,n)	%微分 S，v 为变量，n 阶数
int(S,v,a,b)	%积分 S，v 为变量，a、b 为上下限
dsolve	%常微分方程的符号解
fourier()	%傅立叶积分变换
ifourier()	%逆傅立叶积分变换
laplace()	%拉普拉斯变换
ilaplace()	%逆拉普拉斯变换
taylor()	%泰勒级数展开

例 7：三维图形绘制：

假设有一个时间向量 t，对该向量进行下列运算则可以构成三个坐标值向量

$$x = \sin(t)\text{，}\quad y = \cos(t)\text{，}\quad z = t$$

对于上面的方程可以利用 ezplot3()函数或 plot3()函数绘制三维曲线。这里仅列举了 ezplot3()函数的使用，plot3()函数的使用同学们可以自己尝试。

MATLAB 的表达形式如下：

```
>> ezplot3('sin(t)','cos(t)','t',[0,6*pi])
```

得到的曲线为图 8：

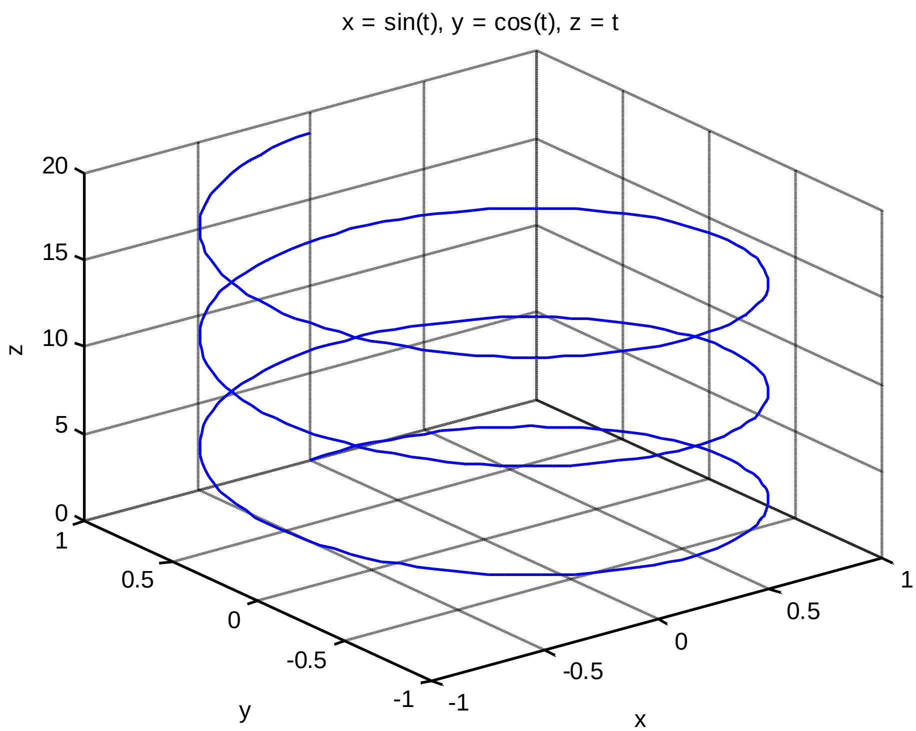


图 8 $x = \sin(t)$， $y = \cos(t)$， $z = t$ 曲线

试绘制下述**曲面**：

$$z(r,\theta) = r^3 \cos(3\theta)$$

其中$0 \le r \le 1$， $0 \le \theta \le 2\pi$。

MATLAB 的表达形式如下：

```
nr=12;nth=50;
r=linspace(0,1,nr);
theta=linspace(0,2*pi,nth);
[R,T]=meshgrid(r,theta)
x=cos(theta')*r;
y=sin(theta')*r;
surf(x,y,R.^3.*cos(3*T))
```

得到的曲面为图 9：

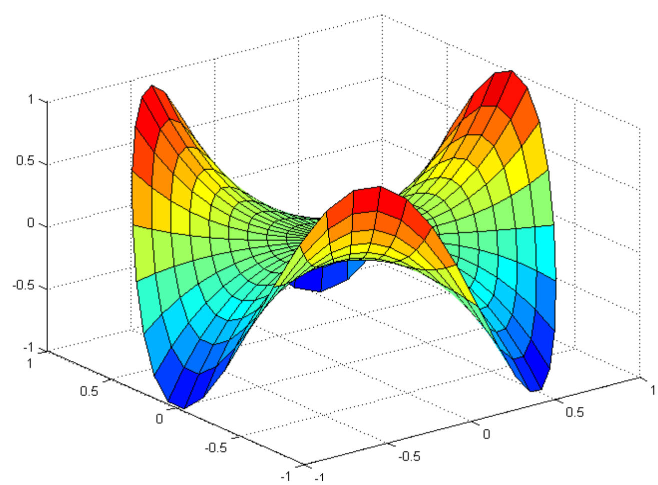


图 9 $z(r,\theta) = r^3 \cos(3\theta)$ 曲面

除了 surf()函数可以绘制曲面外，还有 surfc()、surfl()、mesh()、waterfall()函数也用于曲面的绘制，具体效果如图 10 所示，同学们可以针对自己的需要选取适合的曲面绘制函数。

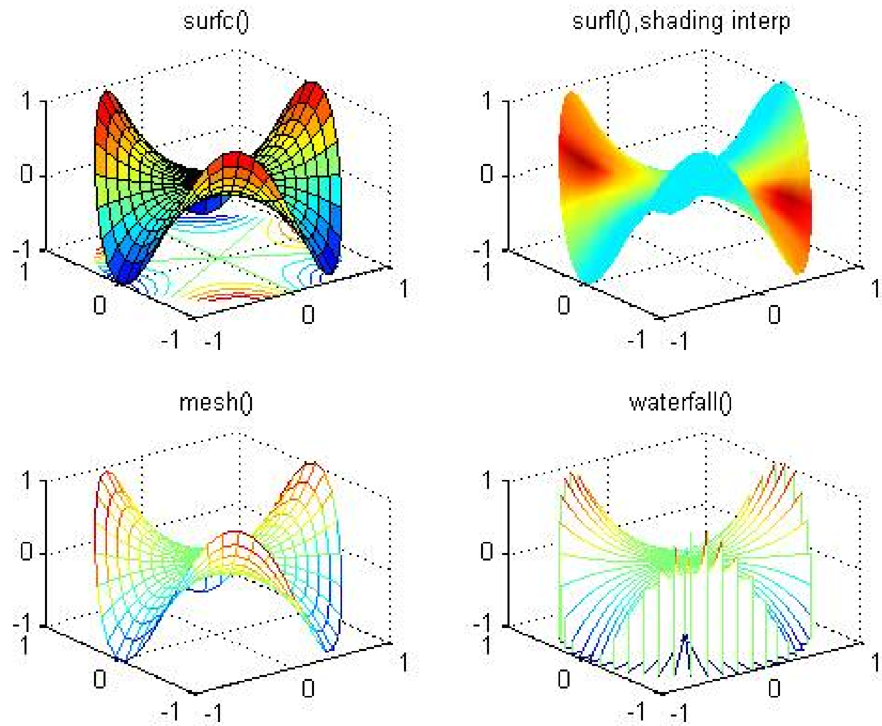


图 10

例 8：线性规划问题：

生产安排问题：

某工厂生产甲、乙两种产品，生产每件产品需要原材料、能源消耗、劳动力及所获利润如下表所示：

品种	原材料（千克）	能源消耗（百元）	劳动力（人）	利润（千元）
甲	2	1	4	5
乙	3	6	2	6

现有库存原材料 1400 千克；能源消耗总额不超过 2400 百元；全厂劳动力满员为 2000 人，试安排生产任务（生产甲、乙产品各多少件），使获得利润最大，并求出最大利润。

解：设安排生产甲产品 x 件，乙产品 y 件，相应的利润为 S。则此问题的数学模型为：

$$\max S = 5x + 6y$$
$$s.t. 2x + 3y \le 1400$$
$$x + 6y \le 2400$$
$$4x + 2y \le 2000$$
$$x \ge 0, y \ge 0, x, y \in Z$$

分析：对于型如上面的线形规划问题，可以用 linprog()函数求解。
MATLAB 的表达形式如下：

```
f=[-5,-6];     %目标函数系数；在 max 时，目标函数需乘-1；min 时，为原系数
A=[2,3;1,6;4,2];   %不等式约束条件系数
b=[1400;2400;2000];   %不等式约束条件常数
LBnd=[0,0];   %两变量取值范围
[x,f]=linprog(f,A,b,[],[],LBnd)        %两个[]位置分别用来代表等式约束条件系数和等式约束条件常数，该题没有等式约束，所以用空矩阵表示
```

计算结果显示为：

```
>> Optimization terminated successfully.
x =
   400.0000
   200.0000
f =    -3.2000e+003
```

即，甲产品 400 件，乙产品 200 件，得到的最大利润为 S=3200（千元）

下面是一个**有约束的多元函数优化**的一个例子。
求下面问题在初始点 x=(10, 10, 10)处的最优解。

$$\min f(x) = -x_1 x_2 x_3$$
$$s.t. 0 \le x_1 + 2x_2 + 2x_3 \le 72$$

解：约束条件的标准形式为

$$-x_1 - 2x_2 - 2x_3 \le 0$$

$$x_1 + 2x_2 + 2x_3 \le 72$$

```
>> fun= '-x(1)*x(2)*x(3)';
>> x0=[10,10,10];
>> A=[-1 -2 -2;1 2 2];
>> b=[0;72];
>> [x,fval]=fmincon(fun,x0,A,b)
```

结果为：

```
x =    24.0000     12.0000     12.0000
fval =        -3456
```

附：
基本统计函数：

mean(x)	%求 x 各列的算术平均值
median(x)	%找 x 各列的中位元素
geomean(x)	%求 x 各列的几何平均值
harmmean(x)	%求 x 各列的调和平均值
max(x)	%找 x 各列的最大元素
min(x)	%找 x 各列的最小元素
range(x)	%极差
var(x)	%求 x 各列的方差

std(x) %求 x 各列的标准差
cov(x) %求 x 的协方差
sum(x) %求 x 各列的元素之和
prod(x) %求 x 各列的元素之积
cumsum(x) %求 x 各列的元素累计和
cumprod(x) %求 x 各列的元素累计积
sort(x) %使 x 各列的元素按递增排序
find() %按条件检索，返回位置矩阵

常用优化函数：

fminbnd(fun,x1,x2) %有约束的一元函数的最小值
fminsearch(fun,x0) %无约束的多元函数的最小值（不连续时）
fminunc(fun,x0) %无约束的多元函数的最小值（阶数大于 2 时）
fmincon(fun,x0,A,b,Aeq,beq,lb,ub) %有约束的多元函数的最小值
lsqcurvefit %非线性最小二乘曲线拟合
lsqin %约束线性最小二乘
lsqnonlin %非线性最小二乘
lsqnonneg %非负线性最小二乘
quadprog(H,fun,A,b,Aeq,beq,lb,ub) %二次规划
fseminf %半无穷约束优化
fminimax(fun,x0,A,b,Aeq,beq,lb,ub) %极小化极大优化
fgoalattain %多目标优化

例 9：图论问题：

最优联网问题：

某乡的乡政府Ｓ与它的几个村Ａ、Ｂ、Ｃ、Ｄ、Ｅ、Ｆ进行信息联网，已测得各村及乡政府间的联网费用如下表（单位：千元）

	S	A	B	C	D	E	F
S		7	8	6	3	4	5
A	7		2	_	_	_	2
B	8	2		5	3.6	_	4
C	6	_	5		5	_	_
D	3	_	3.6	5		3	5.6
E	4	_	_	_	3		4
F	5	2	4	_	5.6	4	

“—”表示两村不能直接联网.

请设计一个最优联网方案，使各村之间与乡政府都能连通，而联网费用又最小（图示联网方案），并求出最小联网费用。

分析：这里我们采用了 Kruskal 算法，编制的 MATLAB 程序如下（关于 Kruskal 算法的详细解释请同学们自行查找图论的相关书籍，在遇到相类似的实际问题时，可以改变变量 b 和 n 的值并直接套用下面的程序即可）：

```
b=[1 1 1 1 1 1 2 2 3 3 3 4 5 5 6;2 3 4 5 6 7 3 7 4 5 7 5 6 7 7;7 8 6 3 4 5 2 2 5 3.6 4 5 3 5.6 4];
%第一个分号前为起点，第二个分号前为终点，后面为起点与终点之间的权。
%这里定义 S 代表 1，A 代表 2，依次类推。没有权值的不输入。
[B,i]=sortrows(b',3);B=B'; m=size(b,2);
n=7;                                    %节点数
t=1:n;k=0;T=[];c=0;
for i=1:m
    if t(B(1,i))~=t(B(2,i))
        k=k+1;
        T(k,1:2)=B(1:2,i)',c=c+B(3,i);
        tmin=min(t(B(1,i)),t(B(2,i)));
        tmax=max(t(B(1,i)),t(B(2,i)));
        for j=1:n
            if t(j)==tmax
                t(j)=tmin;
            end
        end
    end
    if k==n-1
        break;
    end
end
T,c
```

程序运行结果为：

```
>>T =
     2     3
     2     7
     1     5
     5     6
     3     5
     3     4
c =18.6000
```

即最小联网费用为 18.6（千元）。

例 10：二项分布的使用：

飞机成功起飞的概率问题：

由 16 架飞机组成的空军飞行中队要求做好立即起飞的准备，其中一架飞机不能立即起飞的概率为 20%，重新起飞需几分钟的时间，因此一架飞机立刻起飞的概率为 0.80。12 架飞机能够成功起飞的概率为？

分析：这是一个概率中的二项分布问题，常用 binopdf()函数。

```
h=binopdf(12,16,0.80)      %二项分布函数的概率值
```

计算结果为 h=0.2001。

另一方面，至少有 14 架飞机立刻成功起飞的概率为：

```
h=1-binocdf(13,16,0.80)    % 或 h=sum(binopdf(14:16,16,0.80)),
                           其中 binocdf()为二项分布的累积概率值
```

计算结果为 h=0.3518。

在实际的数学建模竞赛中，仅罗列一个一个的数据是枯燥而又不直观的，很难吸引人们的注意，也不容易打动评委们的心；因此，结合数值计算结果，并合理的利用 MATLAB 的绘图功能会起到事半功倍的效果。甚至一些用言语难以表达的结果，用图示也不言自明了。

下面的程序为运行结果的绘图（图 11）表示，希望其中的方法对同学们有所借鉴。

```
n=1:16;
h=binopdf(n,16,0.80);
plot([n;n],[zeros(1,16);h],'k')      %二维绘图函数
text(8-.7:16-.7,h(8:16)+.005,num2str(h(8:16)',3))      %在图中进行注释函数
axis([0 17 0 0.27])                  %坐标轴取值范围函数
xlabel('Number of aircraft launched on time')      %给 x 轴加标题
ylabel('probability')            %给 y 轴加标题
set(gca,'XTick',0:2:16)
set(gca,'XTickLabel',{'0 架','2 架','4 架','6 架','8 架','10 架','12 架','14 架','16 架'})
```

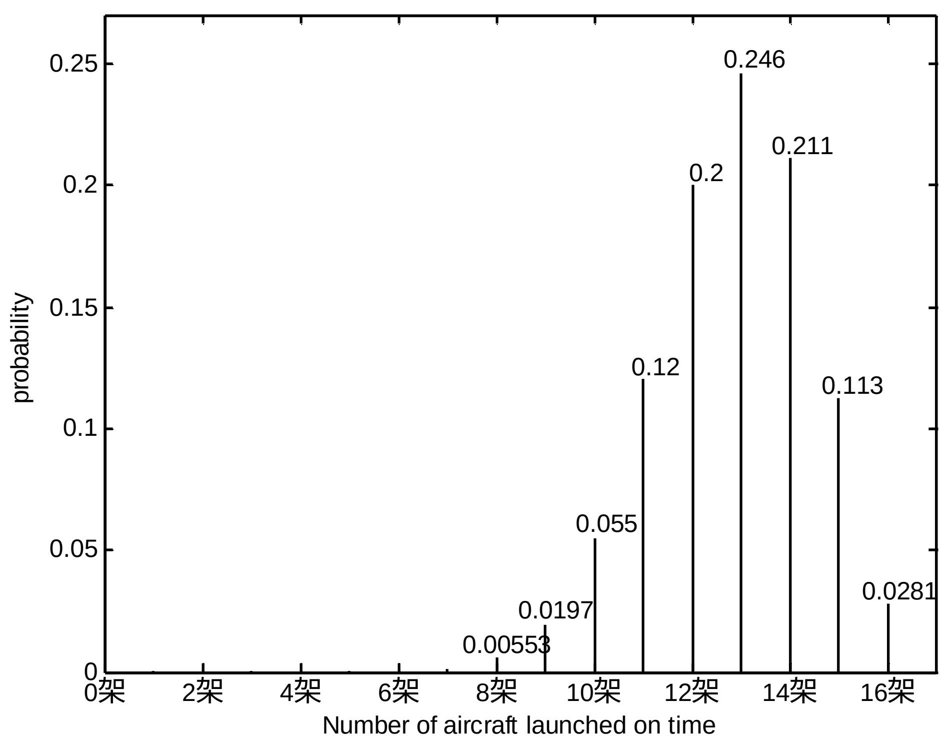


图 11

下面是一个关于**逆累积分布函数**的一个例子。

公共汽车门的高度是按成年男子与车门顶碰头的机会不超过 1%设计的。设男子身高 X（单位：cm）服从正态分布 N（175，36），求车门的最低高度。

解：设 h 为车门高度，X 为身高

求满足条件 $P\{X > h\} \le 0.01$ 的 h，即 $P\{X < h\} \ge 0.99$。

Matlab 的表达式如下：

```
>>h=norminv(0.99, 175, 6)
```

结果为：h = 188.9581

下面我们从另一个角度来计算这个问题。这里如果我们先随机产生 1000 个男子的身高，让它们近似服从正态分布 N（175，36），再求车门的高度，使男子与车门顶碰头的机会不超过 1%。

Matlab 函数表达如下：

```
p=normrnd(175,6,1,1000);      %产生 1000 个服从正态分布 N（175，36）的男子身高
histfit(p)                    %画统计直方图并画出正态分布参考线，如图 14
colormap([1 1 1])             %设置直方图颜色为白色
hh=norminv(0.99, mean(p), std(p))
```

结果显示为：hh =   189.2753

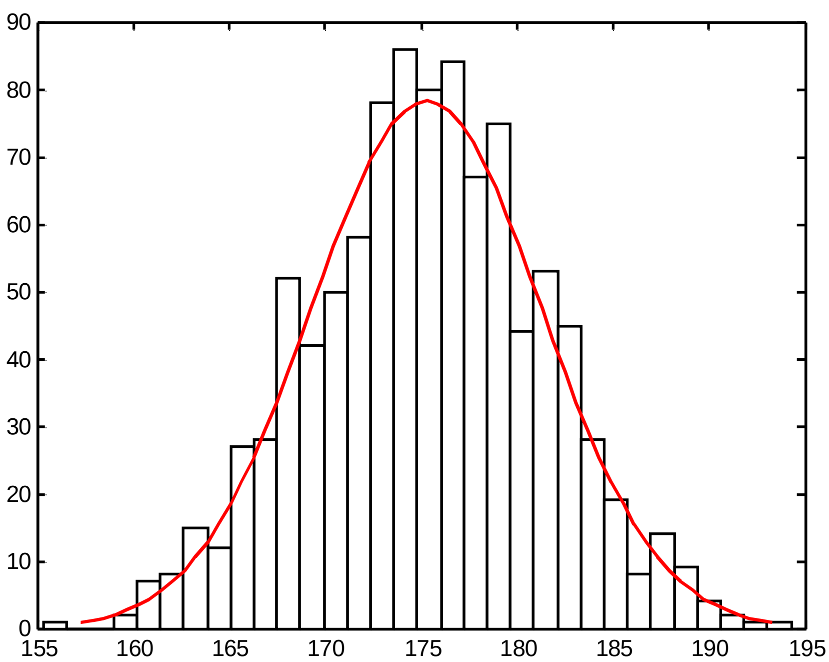


图 14

由于每次都是随机产生 1000 个身高值，得到的图 12 及结果 hh 的值都略有不同。我们可以看到 hh 值与直接计算的 h 值也存在一定的偏差，这种偏差将随着选取随机数的个数的增加而减少，也可以说直接计算的 h 值是 hh 的值的极限情况。

附：常用概率函数：

binornd()	%产生二项分布的随机数
unifrnd()	%产生均匀分布的随机数（连续）
unidrnd()	%产生均匀分布的随机数（离散）
normrnd()	%产生正态分布的随机数
poissrnd()	%产生泊松分布的随机数
exprnd()	%产生指数分布的随机数
geornd()	%产生几何分布的随机数
binopdf()	%二项分布的概率值
unifpdf()	%均匀分布的随机数（连续）的概率值
unidpdf()	%均匀分布的随机数（离散）的概率值
normpdf()	%正态分布的概率值
poisspdf()	%泊松分布的概率值
exppdf()	%指数分布的随机数的概率值
geopdf()	%几何分布的随机数的概率值
chi2pdf()	%卡方分布的随机数的概率值
fpdf()	%F 分布的随机数的概率值
binocdf()	%二项分布的累积概率值
normcdf()	%正态分布的累积概率值
binoinv()	%二项分布的逆累积分布函数
norminv()	%正态分布的逆累积分布函数
chi2inv()	%卡方分布的逆累积分布函数
finv()	%F 分布的逆累积分布函数
binostat()	%二项分布的均值和方差

例 11：线性回归：

一元线性回归：

回归分析统计是在分析两变量或多变量间关系并确定其模型时采用的技术。一元线性回归模型只有一个独立变量。下面的例子是一元线性回归的简单应用，同学们可以结合前面的例子及其函数理解它的用法。

根据例 5 中的数据，用线性回归的方法画出 95%的置信区间。

程序如下，结果如图 12 所示。

```
x=[1:16];   %数据输入
y=[4 6.4 8 8.4 9.28 9.5 9.7 9.86 10 10.2 10.32 10.42 10.5 10.55 10.58 10.6];
[p,resid1]=polyfit(x,y,2) ;
[yhat,w]=polyconf(p,x,resid1,0.05);
plot(x,yhat,'k-',x,yhat-w,'k--',x,yhat+w,'k--',x,y,'ks',[x;x],[yhat;y],'k-')
```

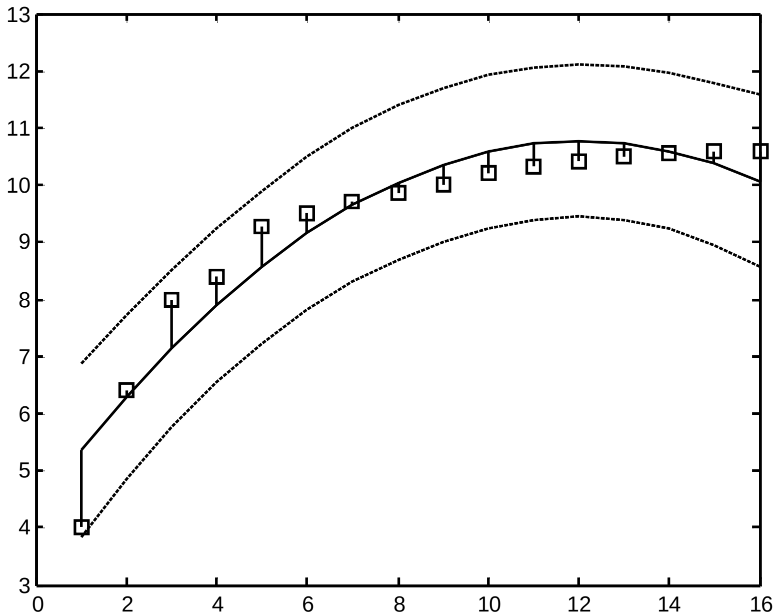


图 12

残差的进一步分析：首先计算残差值然后用函数 normplot 将其画在图 13 中，以观察残差是否服从正态分布。程序如下：

```
x=[1:16];   %数据输入
y=[4 6.4 8 8.4 9.28 9.5 9.7 9.86 10 10.2 10.32 10.42 10.5 10.55 10.58 10.6];
[p,resid1]=polyfit(x,y,2)
normplot(y-polyval(p,x))
whitebg('white')           %设置背景为白色
```

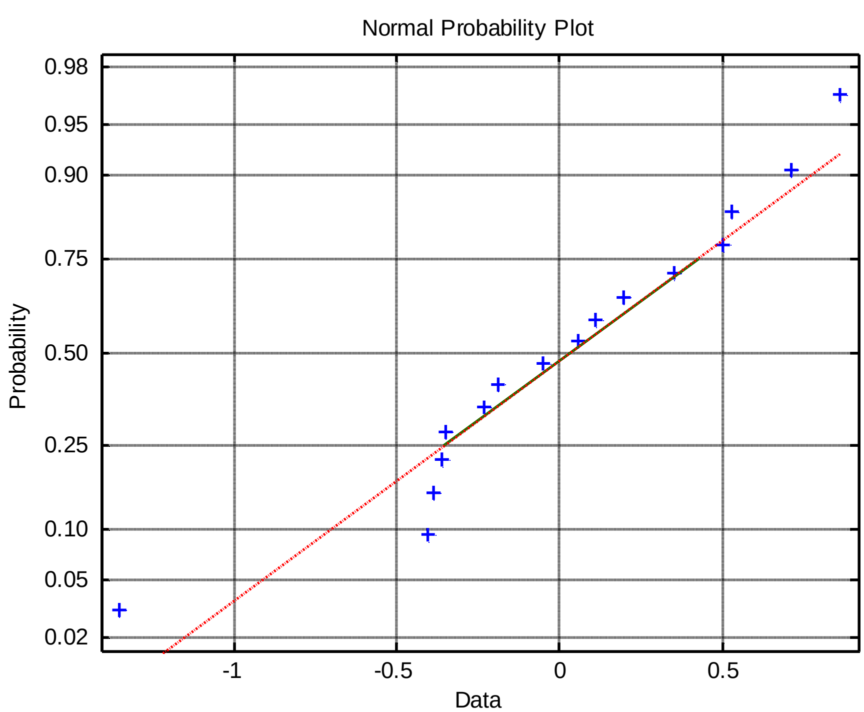


图 13

由图可以看出这些残差点与直线非常接近，由此得出结论，这些残差值非常接近于正态分布，从而证明选择的模型是合适的。

在近几年的数学建模竞赛中，由于各个学校的水平都普遍的提高，竞赛的论文如果仅列出结果，但没有进行适当的误差分析或结果讨论，即使模型很优秀也很难取得理想的成绩。下面我们列举了一个多元回归分析的例子，希望能对同学们在这部分的计算和处理有所帮助。

多元回归分析：有一组数据如下表所示，现用下述方程模拟这些数据：

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3 + \beta_7 x_1^2 + \beta_8 x_2^2 + \beta_9 x_3^2$$

计算出系数的估计值及其在 95%置信度上的置信区间，绘出残差值，以确定残差是否服从正态分布。

y	x_1	x_2	x_3	y	x_1	x_2	x_3
0.22200	7.3	0.0	0.0	0.10100	7.3	2.5	6.8
0.39500	8.7	0.0	0.3	0.23200	8.5	2.0	6.6
0.42200	8.8	0.7	1.0	0.30600	9.5	2.5	5.0
0.43700	8.1	4.0	0.2	0.09230	7.4	2.8	7.8
0.42800	9.0	0.5	1.0	0.11600	7.8	2.8	7.7
0.46700	8.7	1.5	2.8	0.07640	7.7	3.0	8.0
0.44400	9.3	2.1	1.0	0.43900	10.3	1.7	4.2
0.37800	7.6	5.1	3.4	0.09440	7.8	3.3	8.5
0.49400	10.0	0.0	0.3	0.11700	7.1	3.9	6.6
0.45600	8.4	3.7	4.1	0.07260	7.7	4.3	9.5
0.45200	9.3	3.6	2.0	0.04120	7.4	6.0	10.9
0.11200	7.7	2.8	7.1	0.25100	7.3	2.0	5.2
0.43200	9.8	4.2	2.0	0.00002	7.6	7.8	20.7

Matlab 的程序如下：

```
y=[0.222 0.395 0.422 0.437 0.428 0.467 0.444 0.378 0.494...
    0.456 0.452 0.112 0.432 0.101 0.232 0.306 0.0923 0.116...
    0.0764 0.439 0.0944 0.117 0.0726 0.0412 0.251 0.00002]';
x1=[7.3 8.7 8.8 8.1 9.0 8.7 9.3 7.6 10.0 8.4 9.3 7.7 9.8...
    7.3 8.5 9.5 7.4 7.8 7.7 10.3 7.8 7.1 7.7 7.4 7.3 7.6]';
x2=[0.0 0.0 0.7 4.0 0.5 1.5 2.1 5.1 0.0 3.7 3.6 2.8 4.2...
    2.5 2.0 2.5 2.8 2.8 3.0 1.7 3.3 3.9 4.3 6.0 2.0 7.8]';
x3=[0.0 0.3 1.0 0.2 1.0 2.8 1.0 3.4 0.3 4.1 2.0 7.1 2.0...
    6.8 6.6 5.0 7.8 7.7 8.0 4.2 8.5 6.6 9.5 10.9 5.2 20.7]';
X=[ones(length(y),1) x1 x2 x3 x1.*x2 x1.*x3 x2.*x3 x1.^2 x2.^2 x3.^2];
[b,bcl,e,ecl]=regress(y,X,0.05);
lenb=length(b);
disp('Regression coefficients and their confidence limits')
disp([num2str(bcl(:,1)) repmat('< = beta(',lenb,1) num2str((0:lenb-1)')...
    repmat(')=',lenb,1) num2str(b) repmat('< =',lenb,1) num2str(bcl(:,2))])
normplot(e)
whitebg('white')
```

计算结果显示为：

Regression coefficients and their confidence limits

```
    -4.4976< = beta(0)=     -1.7694< =      0.9589
   -0.20282< = beta(1)=      0.4208< =      1.0444
  -0.054708< = beta(2)=     0.22245< =     0.49961
   -0.27691< = beta(3)=      -0.128< =    0.020918
  -0.045395< = beta(4)=   -0.019876< =   0.0056419
 -0.0070049< = beta(5)=   0.0091515< =    0.025308
  -0.012346< = beta(6)=   0.0025762< =    0.017499
  -0.054932< = beta(7)=   -0.019325< =    0.016283
  -0.032989< = beta(8)=  -0.0074485< =    0.018092
  -0.002231< = beta(9)=  0.00082397< =    0.003879
```

在上面的 Matlab 语句中，变量 b,bcl,e,ecl 的含义如下：

$b=[\beta_0,\beta_1,\beta_2,\cdots,\beta_k]$；

bcl 为存放置信下限和置信上限值的数组；

e 为残差；

ecl 为残差的置信区间；

由图 15 可以看出大部分的点都落在正态分布直线的附近，因此题设的模型是合适的。

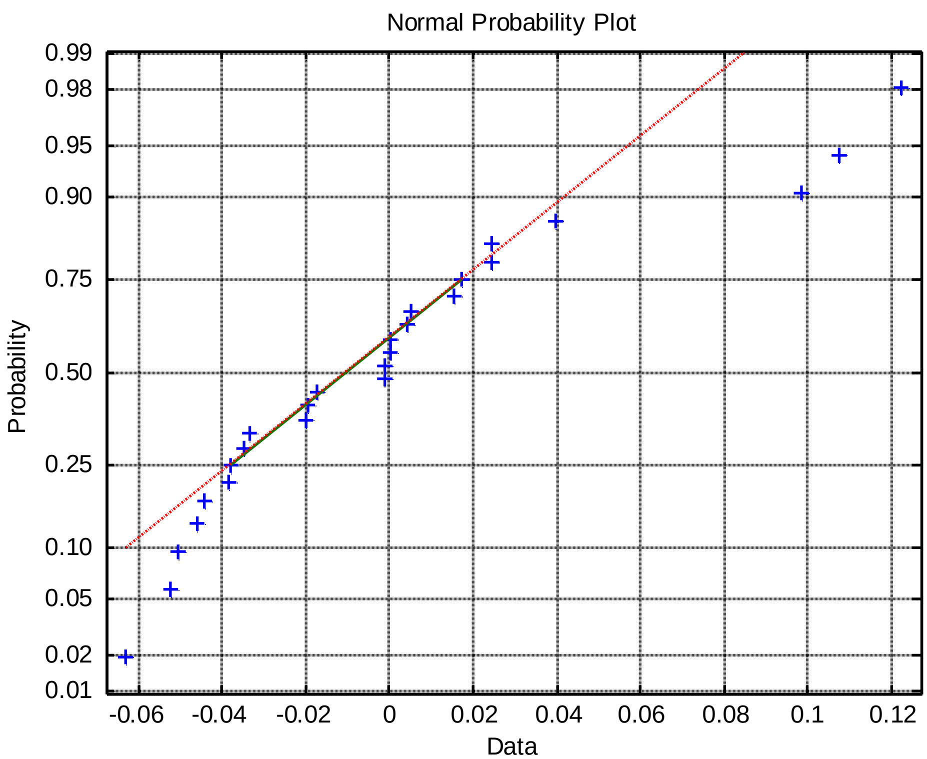


图 15 残差正态累积分布图