一. 入门篇

MATLAB 有哪些主要功能？初学者应如何利用这一数学软件去解决自己的问题？要解决这些问题，应该尽快熟悉一些常用命令，了解它们的功能和使用格式。

1．MATLAB 的特点是什么？

MATLAB **是** ***Matrix Laboratory*** **的缩写，是 Mathworks 公司于 1984 年推出的一套科学计算软件，分为总包和若干工具箱。具有强大的矩阵计算和数据可视化能力。一方面可以实现数值分析、优化、统计、偏微分方程数值解、自动控制、信号处理、系统仿真等若干个领域的数学计算，另一方面可以实现二维、三维图形绘制、三维场景创建和渲染、科学计算可视化、图像处理、虚拟现实和地图制作等图形图象方面的处理。**

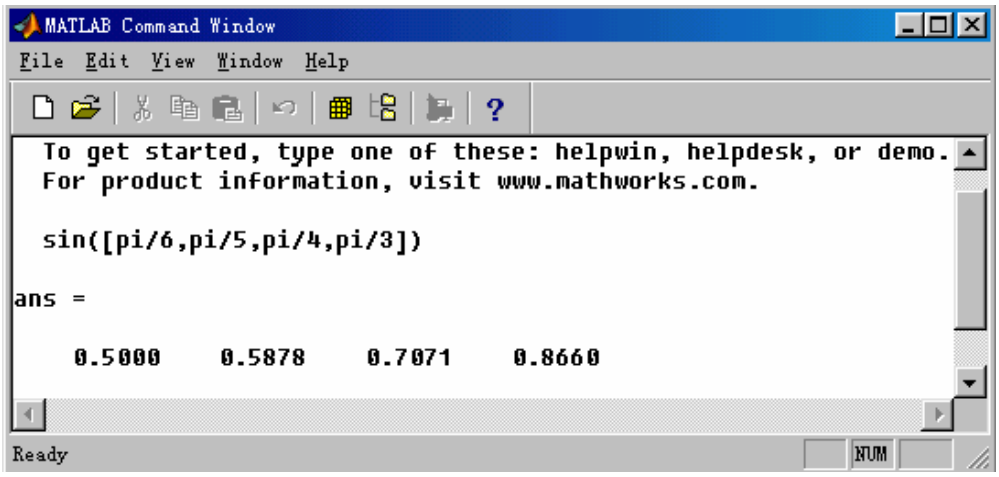
同时，MATLAB 是一种解释式语言. 简单易学、代码短小高效、计算功能强大、图形绘制和处理容易、可扩展性强. 其优势在于：

- ✓ **矩阵的数值运算、数值分析、模拟**
- ✓ **数据可视化、 2D/3D 的绘图**
- ✓ **可以与 FORTRAN、C/C++做数据链接**
- ✓ **几百个核心内部函数**
- ✓ **大量可选用的工具箱**

目前 MATLAB 的最新版本是 7.01 版，它包括了 MATLAB 的各种工具箱，功能强大，适合于较高配置的计算机；同学们在安装 MATLAB 的时候，应该根据自己的机器情况选用不同的版本。

2．如何启动 MATLAB ？

常用的进入 MATLAB 方法是鼠标双击 Windows 桌面上的 MATLAB 图标，以快捷方式进入（如果没有图标，可在桌面上新建“快捷方式”，将 MATLAB “图标”置于桌面）。下图展示了进入 MATLAB 后的工作桌面（窗口），例



如，用键盘输入正弦函数符号及其四个不同的自变量 π/6，π/5，π/4，π/3，

图 1-3 从开始菜单进入 MATLAB

可计算出对应的函数值。

第二种进入方法是鼠标点击 Windows 桌面左下角的“开始”按钮，选择“程序”，然后在第二级菜单中选择“MATLAB”，最后再从第三级菜单中选择“MATLAB6.5”进入了 MATLAB 系统，如下图所示。

在 **MATLAB** 的环境中，键入 **quit(**或 **exit)** 并回车，将退出 **MATLAB**，返回到 **Windows** 桌面。也可以用鼠标单击 **MATLAB** 命令窗口右上方的关闭按钮“×”退出 **MATLAB**。如果想用计算机做另外的工作而不退出 **MATLAB**，这时可以单击 **MATLAB** 命令窗口右上方的极小化按钮“__”，暂时退出（并没有真正退出）**MATLAB** 并保留了工作现场，随时可以单击 **Windows** 任务栏（屏幕下方）中的 **MATLAB** 标记以恢复命令窗口继续工作。

3．如何用 MATLAB 计算三角函数值

在 **MATLAB** 环境下，计算三角函数的一个值或一组值非常方便，只要给定自变量的数据并知道函数名就可以计算出对应函数值。常用的三角函数和反三角函数见表 **1-1**

表 **1-1** 常用三角函数

函数	名称	函数	名称
sin(x)	正弦函数	**asin(x)**	反正弦函数
cos(x)	余弦函数	**acos(x)**	反余弦函数
tan(x)	正切函数	**atan(x)**	反正切函数

通常 **MATLAB** 自变量采用弧度制，例如计算正弦函数在 45^0（即 $\pi/4$）处的值，只须在 **MATLAB** 环境下键入 **sin(pi/4)**，计算机屏幕将显示出计算结果

ans = 0.7071。

如果需计算出正弦函数 $\sin 30^0$，$\sin 45^0$，$\sin 60^0$ 的值，可键入

x=[pi/6，pi/4，pi/3]；sin(x)

计算机屏幕将显示计算结果

ans = 0.5000 0.7071 0.8660。

这说明 **MATLAB** 可以同时计算出某一函数在多个点处的值，而且所用的格式与数学书写格式几乎是完全一致的。

4. MATLAB 有哪些基本数学函数？

除了三角函数和反三角函数以外，**MATLAB** 的内部函数还包括基本初等函数在内的一些函数。这些函数的使用如同正弦函数一样，需要给定自变量数据，然后键入函数名、括号、自变量名并回车，便可得对应的函数值数据。表 **1-2**

中列出了部分数学函数

表 1-2 常用基本函数

函数	名称	函数	名称
abs(x)	绝对值	max(x)	最大值
min(x)	最小值	sum(x)	元素的总和
sqrt(x)	开平方	exp(x)	以 e 为底的指数
log(x)	自然对数	log10(x)	以 10 为底的对数
sign(x)	符号函数	fix(x)	取整

如果想进一步了解 MATLAB 有哪些函数，请键入

help matlab\elfun （回车）

计算机屏幕将显示出更多的细节，其中还有双曲函数和反双曲函数等。

注意，计算函数值时必须首先确定自变量的数据（一个或一组）。例如，随机抽取 10 名大学生的《高等数学》课程考试成绩，并统计他们中的最高分、最低分、以及这 10 人的平均分。可以利用表 1-2 中的函数 max、min、sum 来实现。首先在 MATLAB 环境下输入 10 个分数值并赋值给 f

f=[88，84，73，69，78，80，90，96，86，77]

然后分别计算这 10 个数据的最大值、最小值、平均值：

```
h=max(f)
l=min(f)
m=sum(f)/10
```

计算机屏幕显示最高分 h = 96；最低分 l = 69；平均分 m = 82.1000。

虽然 MATLAB 的内部函数已经相当多，但是还是不能完全满足人们的需求。如果需要计算一个复杂的函数值，我们必须利用字处理软件编写一个新的函数文件（参考下一部分基础篇），才能实现对函数值的计算。

5．如何用 *plot* 命令绘平面图形？

plot 是MATLAB的最基本的绘制二维图形指令。其主要功能是根据函数表

X	x_1	x_2	……	x_n
Y	y_1	y_2	……	y_n

绘制出函数的图形。用plot命令绘图必须要输入自变量的一组值，并计算出对应的函数值，其基本调用格式有下面几种：

(1) plot(X ,Y) 绘制出以 $X = [x_1\ x_2\ \cdots\ x_n]$为横坐标，以 $Y = [y_1\ y_2\ \cdots\ y_n]$为纵坐标的平面上点的连线图；

(2)plot(Y) 绘出以 **Y =[y_1 y_2 … y_n]**为纵坐标， **X =[1 2 … n]**为横坐标的二维图形；

(3) plot(X1,Y1,X2,Y2)同时绘出两个函数表 **(X1, Y1)**及 **(X2, Y2)**所描述的函数；

(4) plot(x,y,'s') 中的选项s可以控制图形的颜色及图形的线方式(或点方式)。

表1-3 图形控制选项列表

线方式	点方式	颜色
实线 -	点 .	红 r
虚线 --	加号 +	绿 g
冒号 :	星号 *	兰 b
横点 -.	圆 o	白 w
	叉 x	

例如，要绘制正弦函数在$[0,\ 2\pi]$内变化的曲线，可以先确定出 **0** 到2π之间，间隔为 **0.2** 的一组自变量数据，然后计算正弦函数在这些点处的函数值，最后根据所得数据绘制出函数图形。

***x*=0：0.2：2*pi；*y*=sin(*x*)；*plot*(*x*，*y*)**

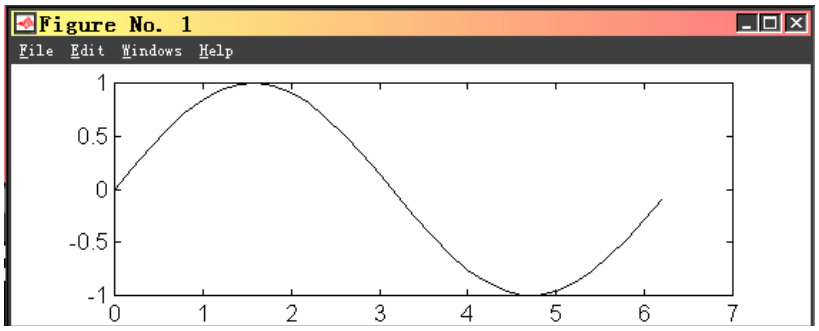


图 1- 5 正弦函数图形

6．如何在 MATLAB 中创建矩阵？

计算函数值须输入自变量数据，计算机程序也必须在数据集合上才能运行，所以初始数据的输入十分必要。在计算机程序设计中人们习惯将矩阵称为二维数组，将向量称为一维数组。向量本质上是一类特殊的矩阵，矩阵可以分解为一系列行向量或列向量，有限个同维行（列）向量也可以构成一个矩阵。

MATLAB 以矩阵为数据单元，数据的输入实际上是创建矩阵。

(1) 对于小型矩阵的创建可以用方括号方法。如：

A=[1 2 3；4 5 6；7 8 9];

就创建了一个 3×3 的矩阵。注意：矩阵中同一行的元素用“空隔”或“，”隔开，行与行之间用分号“；”隔开。也可以用回车换行来代替分号。如

```
A=[ 1  2  3
    4  5  6
    7  8  9 ];
```

仍然可以创建同样的 3×3 矩阵。

(2) 利用矩阵的剪裁和拼装创建新的矩阵。例如在 **MATLAB** 中键入命令

a=[1 2 3;4 5 6;7 8 9]；b=[1;1;1];

得到矩阵和向量

$$a=\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},\quad b=\begin{bmatrix}1\\1\\1\end{bmatrix}$$

命令 **A=[a b]**，将矩阵 **a** 和向量 **b** 拼装成新的矩阵

$$A=\begin{bmatrix}1&2&3&1\\4&5&6&1\\7&8&9&1\end{bmatrix}$$

而命令 ***a*1 = a(1:2,2:3)**将得到新的 2 阶矩阵

$$a1=\begin{bmatrix}2&3\\5&6\end{bmatrix}$$

(3) 对于大型矩阵的创建可以用数据文件的方法。

首先，用编辑器编辑一个数据文件，在这个文件中逐行录入矩阵的所有数据；然后，在 **MATLAB** 环境下用 **load** 指令将这一矩阵调入工作空间（计算机内存）。使用这种方法时须注意，文件名即是变量名。例如，某地区有 **12** 个气象观测站，记录了 **10** 年以来所测得的每年的年降雨量数据。每一个站点的十个数据构成了矩阵的一个行向量，**12** 个站点的数据就构成了一个 12×10 的矩阵。用编辑器如记事本编辑器编辑这个具有 **120** 个数据的文

件名为 a1.dat 的数据文件的步骤如下：

第一步：在 MATLAB 环境中键入 edit a1.dat （回车）进入编辑环境；

第二步：逐行录入矩阵的所有数据

```
276.2  251.6  192.7  246.2  291.7  466.5  258.6  453.4  158.5  324.8;
324.5  287.3  436.2  232.4  311.0  158.9  327.4  365.5  271.0  406.5;
158.6  349.5  289.9  243.7  502.4  223.5  432.1  357.6  410.2  235.7;
412.5  297.4  366.3  372.5  254.0  425.1  403.9  258.1  344.2  288.8;
292.8  227.8  466.2  460.2  245.6  251.4  256.6  278.8  250.0  192.6;
258.4  453.6  239.1  158.9  324.8  321.0  282.9  467.2  360.7  284.9;
334.1  321.5  357.4  298.7  401.0  315.4  389.7  355.2  396.4  290.5;
303.2  451.0  219.7  314.5  266.5  317.4  413.2  228.5  179.4  343.7;
292.9  466.2  245.7  256.6  251.3  246.6  466.5  453.6  159.2  283.4;
243.2  307.5  411.1  327.0  289.9  277.5  199.3  315.6  342.4  281.2;
159.7  421.2  357.1  296.5  255.4  304.2  282.1  456.3  331.2  243.7;
331.2  455.1  353.2  423.0  362.1  410.7  387.6  407.2  377.7  411.1
```

将文件存盘后退出 EDIT 环境，回到 MATLAB 环境中。

第三步：键入 load a1.dat （回车）便将数据调入内存。键入 a1 (回车)计算机屏幕将显示矩阵 a1 的 12×10 的全部数据。

7. 创建向量有哪些方法？

向量在计算机中称为一维数组，在计算某一函数的一组值时，需要给定一组自变量的值，即创建向量。

(1)创建向量的通用方法是冒号法。使用格式为

x = 初值：步长：终值

注意：当初值小于终值时，步长必须为正数；当初值大于终值时，步长必须为负数。当步长为 1 时，可以省略不写。如果初值为 1，步长为 1，终值为 10。则可以键入

x=1：10

则创建了向量 x=[1 2 3 4 5 6 7 8 9 10]。

如果初值为 1，步长为 0.5，终值为 10。则可以键入

x1=1：0.5：10

则创建具了向量 x1=[1 0.5 2 2.5 3 …… 8 8.5 9 9.5 10]。

(2)利用 linspace 和 logspace 指令创建向量。

命令 linspace(x1,x2)将产生一个具有 100 个元素的行向量，这个行向量的元素是介于

x1 和 x2 之间的有限等差数列，即行向量的第一个元素为 x1，最后一个元素是 x2。

命令 linspace(x1, x2, N) 将产生介于 x1 和 x2 之间的 N 个点形成的行向量，向量的第一个元素为 x1 最后一个元素为 x2，仍是等差数列。

例如 x=linspace(1,6,8)将得到介于 1 到 6 之间的八个元素的行向量

x =[1.0000 1.7143 2.4286 3.1429 3.8571 4.5714 5.2857 6.0000]

logspace(d1, d2)将产生具有 50 个对数等距点的行向量，其中向量的元素介于 10^{d1} 和 10^{d2} 之间。

logspace(d1, d2, N) 将产生 N 个对数等距点的行向量。

(3) 利用已有的矩阵剪裁方法创建向量。

例如，在 MATLAB 中用下面的命令

A=[1 2 3; 4 5 6; 7 8 9];

创建了矩阵

$$A=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$$

键入 p1 = A(1，:)，可得行向量 p1 = [1 2 3]。这是矩阵 A 的第一行行向量。

键入 p2 = A(:，1)，可得列向量 p2 = $[1 \quad 4 \quad 7]^T$。这是矩阵 A 的第一列列向量。

同理， A(2，:)、A(3，:) 分别得到矩阵 A 的第二行和第三行两个行向量；

A(:，2)、A(:，3) 分别得到矩阵 A 的第二列和第三列两个列向量。

8．特殊符号 “；”和“：”有何用处？

在 MATLAB 中，分号“；”的用处通常有两个，一是是用于矩阵数据输入时将相邻两行数据分隔开；二是将它用于一条 MATLAB 命令之后，使该命令被执行后所产生的数据结果（如果有数据结果）不显示在计算机屏幕上（因为计算机 CPU 运行速度远高于屏幕显示速度，在程序运行时不显示一些中间数据将节约不少时间）。

例如，在 MATLAB 中键入

y=sin(pi/5);

计算机将计算出正弦函数在 $\pi/5$ 处的函数值，并把计算所得的值赋值给 y，但是计算机屏幕上却不显示出任何数据。如果想知道 y 的值是多少，只须键入 y 并回车，屏幕上将显示出 y 的数据。

在 MATLAB 中冒号“：”的用处通常也有两个，一是用于循环，二是用于矩阵的裁剪。

如，语句 “x=1：6”将产生一个具有六个元素的向量

x = [1 2 3 4 5 6]

“x=100：-5：10”将产生一个具有十九个元素的向量

x = [100 95 90 … 20 15 10]

对于矩阵

$$A=\begin{bmatrix}1&2&3\\4&5&6\\7&8&0\end{bmatrix}$$

A(1 ，:)表示矩阵 A 的第一行之所有元素形成的行向量 1 2 3。A(: ，1)则表示矩阵 A 的第一列所有的元素 1 4 7 形成的列向量。而 A(1:2 , 1:2) 则表示矩阵 A 的前两行和前两列的元素形成的 2 阶矩阵

$$\begin{bmatrix}1&2\\4&5\end{bmatrix}$$

9．如何求解一元 *n* 次方程

求一元 *n* 次方程的根用命令 **roots**。我们都知道，一元二次方程 $ax^2+bx+c=0$ 的求根公式为

$$x_{1,\ 2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

例如对于方程式 $2x^2+5x+8=0$，首先输入系数，然后按公式计算两个根的值，在 MATLAB 中键入

```
a=2；b=5；c=8；
p=sqrt(b^2-4*a*c);
x1=(-b-p)/(2*a)
x2=(-b+p)/(2*a)
```

计算机屏幕将显示此方程的两个根：

x1 = -1.2500 - 1.5612i x2 = -1.2500 + 1.5612i

对于高次方程，比如 5 次以上的方程，我们无法用求根公式求解。但是用 MATLAB 的求多项式零点的命令可以求出高次方程的全部根。以上面的例子为例，只须键入

```
roots([2,5,8])
```

并回车，计算机将显示 ans =

-1.2500 + 1.5612i

-1.2500 - 1.5612i

这与前面计算结果相同，由此可知命令“**roots([2,5,8])**”求出了一元二次方程

$$2x^2+5x+8=0$$

的全部根。所以对于一个高次代数方程

$$a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n=0$$

用命令 **roots([a_0 a_1 … a_n])**可以求出该 ***n*** 次方程的全部根。

10. 如何求解线性方程组？

当一个方程组 ***Ax=b*** 有唯一解时可用指令 ***A\b*** 直接求解，其中，***A*** 是线性方程组的系数矩阵，***b***是方程组的右端向量。例如给定线性方程组

$$\begin{bmatrix}1 & 2 & 1 & -2\\ 2 & 5 & 3 & -2\\ -2 & -2 & 3 & 5\\ 1 & 3 & 2 & 3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}=\begin{bmatrix}4\\ 7\\ -1\\ 0\end{bmatrix}$$

可以先输入系数矩阵和右端向量，然后直接求解，在**MATLAB**中键入

***A*=[1 2 1 -2；2 5 3 -2；-2 -2 3 5；1 3 2 3];**

***b*=[4；7；-1；0];**

***x*=A\b**

计算机执行后，将显示数据结果

***x*= 2 -1 2 -1**

由此得知方程组的解为

$x_1=2$，$x_2=-1$，$x_3=2$，$x_4=-1$

上面这一种方法非常适用于方程组有唯一解的情形，在处理实际问题时，有些方程组有无穷多组解，这时可用另一条指令 **rref([*A b*])** 化简方程组的增广矩阵，然后利用线性代数的方法得出方程组的通解。当方程组有唯一解时，仍然可以用这一指令，例如求解上面例子

在**MATLAB**中键入

```
A=[1 2 1 -2；2 5 3 -2；-2 -2 3 5；1 3 2 3];
b=[4；7；-1；0];
rref([A b])
```

计算机执行后，屏幕将显示数据结果

```
ans =     1     0     0     0     2
          0     1     0     0    -1
          0     0     1     0     2
          0     0     0     1    -1
```

这一矩阵与增广矩阵等价(4 个非零行 5 列)，由此写出与原方程等价的方程组，即

$$x_1 = 2，\ x_2 = -1，\ x_3 = 2，\ x_4 = -1$$

如果用 **rref([A b])**命令得到最后的矩阵中非零行数小于列数减 **1**，则可求出线性方程组的通解。有关线性方程组的通解和基础解系的概念请参考线性代数教材。

11．如何求解矩阵方程？

在一个矩阵方程中，未知元素是一个矩阵。线性代数中的矩阵方程总可以写成如下两种形式中的一种

$$AX = B \quad 或 \quad XA = B$$

这里，X 是未知矩阵，而 A、B 则是已知的矩阵。这两种类型的矩阵方程的解分别为

$$X = A^{-1}B \quad 或 \quad X = BA^{-1}$$

在 **MATLAB** 环境中，可以分别用下面两条命令求解这两类方程

$$X = A\backslash B \quad 或 \quad X = B/A$$

注意，A 左除 B（用反斜杠）相当于用 A 的逆矩阵左乘矩阵 B；A 右除 B（用斜杠）相当于用 A 的逆矩阵右乘矩阵 B。

例如求矩阵方程

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & 2 \\ 1 & -1 & 0 \end{bmatrix} X = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}$$

可用如下命令

```
A = [1  1  -1; 0  2  2; 1  -1  0];
B = [1  -1; 1  1; 2  1];
X = A\B
```

计算机运行后，显示数据结果

```
X =
    1.8333    0.5000
```

-0.1667 -0.5000

0.6667 1.0000

这就是矩阵方程的解。显然，***X*** 是一个 **3×2** 阶的矩阵。

再例如，求矩阵方程

$$X\begin{bmatrix}1 & 1 & -1\\ 0 & 2 & 2\\ 1 & -1 & 0\end{bmatrix}=\begin{bmatrix}1 & -1 & 1\\ 1 & 1 & 0\end{bmatrix}$$

可用如下命令

A **= [1 1 -1；0 2 2；1 -1 0];**

B **= [1 -1 1；1 1 0];**

X **=** ***B*****/*****A***

计算机运行后，显示数据结果

X =

-0.3333 0.3333 1.3333

0.6667 0.3333 0.3333

显然，这一矩阵方程的解是一个 **2×3** 的矩阵。

12. 如何求一个 *n* 阶行列式的值？

如果已经输入了一个方阵 ***A*** 的全部数据，那么只须用命令 **det(*A*)** 就能求出 ***A*** 的行列式的值。对于一个特殊的 ***n*** 阶行列式的计算必须先分析行列式中数据分布的规律，用算法产生对应的矩阵 ***A***，然后用命令 **det(*A*)** 计算出具体的行列式的值。

例如对于 ***n*** 阶行列式

$$\begin{vmatrix}1 & 2 & 2 & \cdots & 2\\ 2 & 2 & 2 & \cdots & 2\\ 2 & 2 & 3 & \cdots & 2\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 2 & 2 & 2 & \cdots & n\end{vmatrix}$$

分析数据分布的规律可知，对应的矩阵主对角元素是 **1、2、3、…、*n***，而其余的元素均为 **2**。另外，当行列式的阶数不同时数据结果也不一样，所以 ***n*** 扮演参数的角色。请参考下面的程序段计算行列式的值

```
n=input('input  n= ?');
```

```
A=2*ones(n，n)；
for   k=1：n
      A(k，k)=k；
end
det(A)
```

这里，第一条语句是要求从键盘上输入一个 ***n*** 的值，第二条语句是创建全部元素为 2 的 ***n*** 阶方阵，接下来用 for …… end 语句将方阵的主对角元修改为 1、2、3、…、***n***，最后用求行列式命令计算并显示出该行列式的值。如果将 ***n*** 的值输入为 5，计算结果为 －12；如果将 ***n*** 的值输入为 6，则计算结果为 －48；……，从多次试验的数据结果可以猜测出本题答案为 －2（***n***－2）!。但是，这样多次使用这段程序计算必须将程序作为一个方件保存在磁盘上，关于 MATLAB 的文件操作请参考问题 22 。

13．如何使用 MATLAB 的在线帮助 ？

MATLAB 的命令非常多，即使是经常使用这一软件的人，也会忘记常用的 MATLAB 命令。对于记住的一些命令，在使用时又可能忘记了格式。这时，可以向 MATLAB 系统寻求帮助。获取 MATLAB 指令的帮助信息有多种方法，这里介绍常用的三种。

（1）知道命令名称，不熟悉使用格式，可以用 help 的命令获得帮助，格式如下

help <MATLAB 的具体指令> (回车) 。

例如：键入 help magic (回车) 屏幕将显示怎样使用 magic 命令求 n 阶幻方。

MAGIC(N) is an N-by-N matrix constructed from the integers 1 through N^2 with equal row and column sums.

根据帮助信息,在 MATLAB 环境下键入 magic(3) (回车)，屏幕上会出现 3 阶幻方矩阵:

$$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

这一矩阵的每一行元素之和均为 15，每一列元素之和也均为 15。由此可以体会到 magic 这条指令的使用方法以及它的具体功能。

（2）知道命令名称的第一个英文字母，可用鼠标选择 MATLAB 命令窗口上方菜单栏中的“help”选项，再从下拉菜单中选择“Index”。这时，屏幕上将出现 MATLAB 的帮助窗口，窗口首先出现 MATLAB 全部命令名称的索引，索引的命令名按英文字母顺序排列。找到你所需要的命令名，用鼠标单击它，将会得到该命令的英文帮助信息。

（3）如果忘记了所要使用的命令的名称，可以根据命令的功能属于哪一类来进行分类的查找。MATLAB 的所有命令和语句被划分为 30 类。在 MATLAB 环境下键入指令 help (回车) 屏幕将显示出关于 MATLAB 的 30 个类的名称。

toolbox\local	本地函数库（启动 MATLAB 时所需的文件）
matlab\datafun	数据分析和付里叶变换（统计计算、离散卷积、快速付里叶变换）
matlab\elfun	基本数学函数（三角函数、双曲函数、指数函数、对数函数等）
matlab\elmat	基本矩阵和矩阵操作（常用矩阵、特殊变量和常数、时钟函数）
matlab\funfun	函数操作——非线性数值方法（数值积分、常微分方程求解等）
matlab\general	常用操作命令（用于日常管理、窗口控制的命令和变量）
matlab\color	色彩控制和灯光效果（关于图形的色彩和灯光效果控制命令）
matlab\graphics	图形窗口控制及一般绘图命令（清除、保持图形、做动画等）
matlab\iofun	低级文件输入/输出命令（打开、关闭文件，读写二进制文件等）
matlab\lang	语言结构和调试（MATLAB 编程常用的一些命令和语句）
matlab\matfun	矩阵函数和数值线性代数（矩阵变换、矩阵分解、特征值等）
matlab\ops	运算符和特殊符号（算术运算、逻辑运算以及特殊符号）
matlab\plotxy	二维图形绘图命令（绘制平面图形的各种命令）
matlab\plotxyz	三维图形绘图命令（绘制空间图形的各种命令）
matlab\polyfun	多项式函数与插值函数（有关多项式运算以及代数插值、样条）
matlab\sounds	声音处理函数（声音的读写、变换等）
matlab\sparfun	稀疏矩阵函数（稀疏矩阵的创建、图论中的图的绘制等命令）
matlab\specfun	特殊数学函数（贝塞尔函数、误差函数、椭圆积分等）
matlab\specmat	特殊矩阵（希尔伯特矩阵、幻方矩阵、多项式的伴随友阵等）
matlab\strfun	字符串函数（关于字符串的操作，字符比较、字符变换等）
matlab\dde	动态数据交换工具箱
matlab\demos	MATLAB 自动演示系统所用的主程序和全部 M 文件

根据分类的信息，可以再次用“help”命令。例如键入

help elfun

将获得 MATLAB 中全部基本数学函数的清单。最后可以用“help”命令获取具体命令的帮助信息。

14. MATLAB 有哪些运算符号及特殊字符?

MATLAB 具有其它计算机高级语言（如 BASIC、FORTRAN、C）所具有的内部函数和用于算术运算和逻辑运算的运算符号和关系符号。

早期的 MATLAB 主要是针对矩阵进行运算（MATLAB 的名称来源于矩阵实验室），所以它还具有一些特殊的运算符号。如点乘、点除、点方幂等，正是这些特殊的功能使得这一软件具有高效的编程环境。无论是在 MATLAB 的交互式环境下使用各种指令，还是利用批处理进行编程，这些运算符号和特殊字符都是使用者所必须熟悉的。

+ 加法运算。适用于两个数相加或两个同阶矩阵相加。

- 减法运算。

* 乘法运算。适用于两个数相乘或两个矩阵相乘。

.* 点乘运算。适用于两个同阶矩阵对应元素相乘。例如:
[1 2 3].*[1 2 2] 其结果为:[1 4 6]。

./ 点除运算。例如:
[1 2 3]./[1 2 2] 其结果为:[1 1 1.5]。

^ 乘幂运算。

\ 反斜杠表示左除。如，$x = A \backslash B$ 可以得到矩阵方程 $Ax = B$ 的解。

pi 数学常数 π。即，3.1415926535897.....。

! 惊叹号用于后接 DOS 操作命令(不退出 MATLAB 执行 DOS 命令)。
如，使用 DOS 的编辑器 EDIT.EXE 编写 MATLAB 程序，可以键入
！EDIT <LAB1.M> (回车) 就进入编辑环境，编写文件名为 LAB1.M 的 MATLAB 程序。

== 双等号表示相等关系符号。另外，“<” 和 “>”分别表示小于和大于关系符号。

% 用于注释行开始。在程序中某行的第一列加上%，将不执行这一行。

& 逻辑与运算符号。另外，“|”和“~”分别表示逻辑或和逻辑非运算符号。(注意：“~” 这个符号在计算机键盘左上角，与“`”在同一键上)。

15．WHO 和 CLEAR 有什么用处？

这两条指令主要用于管理 MATLAB 的工作空间。计算机内存是非常宝贵的资源，如果连续使用 MATLAB 的时间较长，已经创建或产的很多变量，就会占用相当多的空间。而一些过时无用的变量长时间滞留于内存，对另外的一些数据产生干扰。这时，应当清理工作空间，删除一些过时不用的变量以释放计算机内存。这样就可以保持一个正常良好的工作环境，使工作可以不间断地进行下去。

指令 WHO 用于显示出当前工作空间中存在的所有变量列表，而另一指令 CLEAR 则可以用于清除多余的变量。 CLEAR x 仅仅清除变量 x。CLEAR x y z 将清除变量 x、 y、z。注意：单独使用 CLEAR 将清除掉所有工作空间的变量，这样等同于退出 MATLAB 再重新进入。

另外，指令 WHOS 不仅列出当前工作空间中的所有变量，还将每一变量所占用的空间大小列表显示出来。从所显示出的信息还可以得知这些变量中哪些

是矩阵，哪些是向量，哪些是单个的数据。

16．如何用 MATLAB 绘制标志的图形？

MATLAB 软件系统的图标的雏形是二元函数

$$f(x, y) = e^{-(x^2+y^2)} \qquad -2.3 \le x \le 0.8, \quad -1.5 \le y \le 1.5$$

所对应曲面图形。这一曲面图形和 **MATLAB** 的系统图标有一些微小的差别。

在 **MATLAB** 环境中键入如下指令

```
x= -2.3：0.1：0.8；y=x+0.8；          %确定离散点横坐标和纵坐标
[x，y] = meshgrid(x，y)；              %产生二元函数自变量域的离散点
z=exp(-(x.^2+y.^2));                  %计算各离散点处函数值
z(20:32,1:14)=zeros(13,14);           %修改部分函数值为零
mesh(z)                               %根据离散点处函数值绘图
```

这五行指令被计算机执行后，屏幕上将显图 **1-6** 中的图形。**Mesh** 是绘制曲面的命令，具体使用格式请参考问题 **47**。

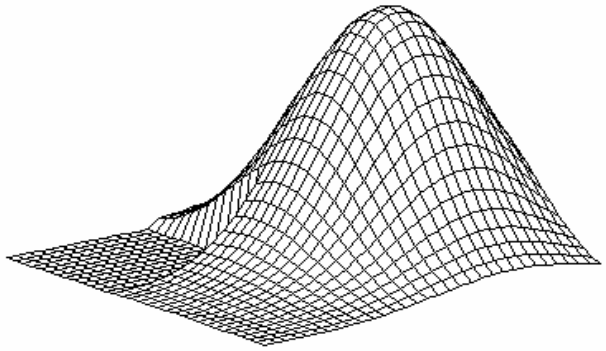


图 1-6 系统标志图形

17．如何保存计算所得的数据结果？

保存数据结果是非常有用的，例如第 16 问题中通过计算获得了图 1-6 中曲面上各点处的高度值，这些数据以变量 z 的名义保存在计算机内存中，一旦退出 MATLAB，变量 z 所含的数据就会丢失。在这种情况下可用命令 “save date1 z” 将变量 z 的数据保存在磁盘上，下一次使用 MATLAB 时用命令 “load date1” 将所保存的数据从磁盘上调入内存，可以再次使用。

这里，date1 是数据文件名。Save 命令的具体使用格式如下

save 文件名 变量名 1 变量名 2 … 变量名 *n* （回车）

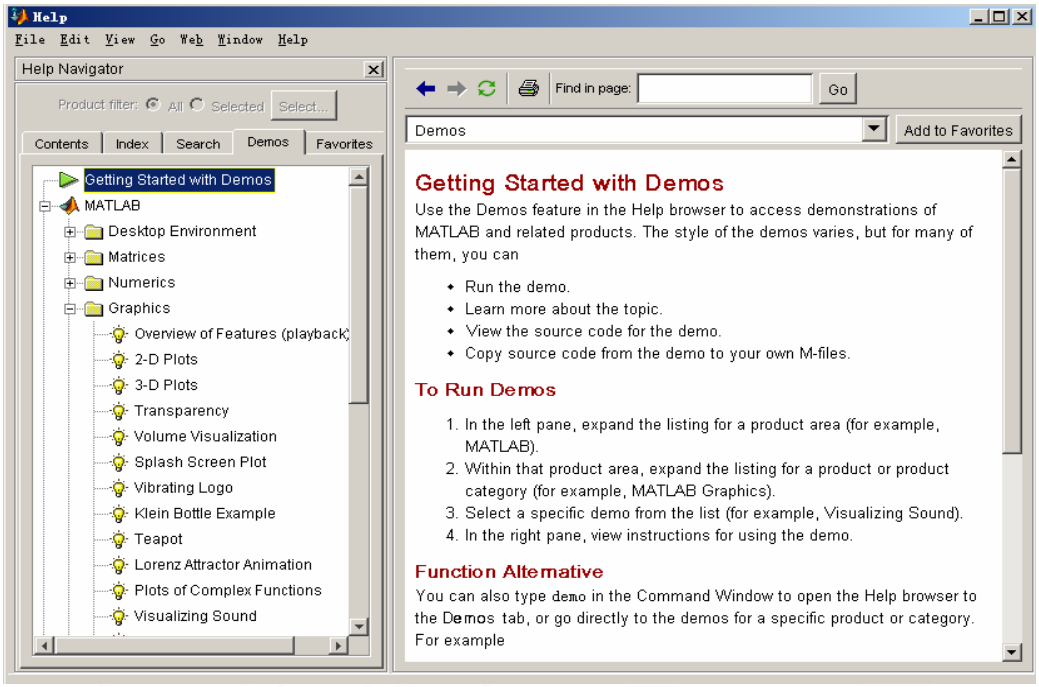
如果将文件名取为 date1，则 MATLAB 将在磁盘上产生一个文件，文件名为 date1.mat，这一文件保存了“变量 1 变量 2 … 变量 *n*”这 *n* 个变量的数据。使用这一命令时缺省了变量名，即只用命令“save date1”,则 MATLAB 会将当前工作空间中所有的变量（数据）保存在文件 date1.mat 中。

将数据恢复（即从磁盘上将数据读入内存）所用命令的格式如下

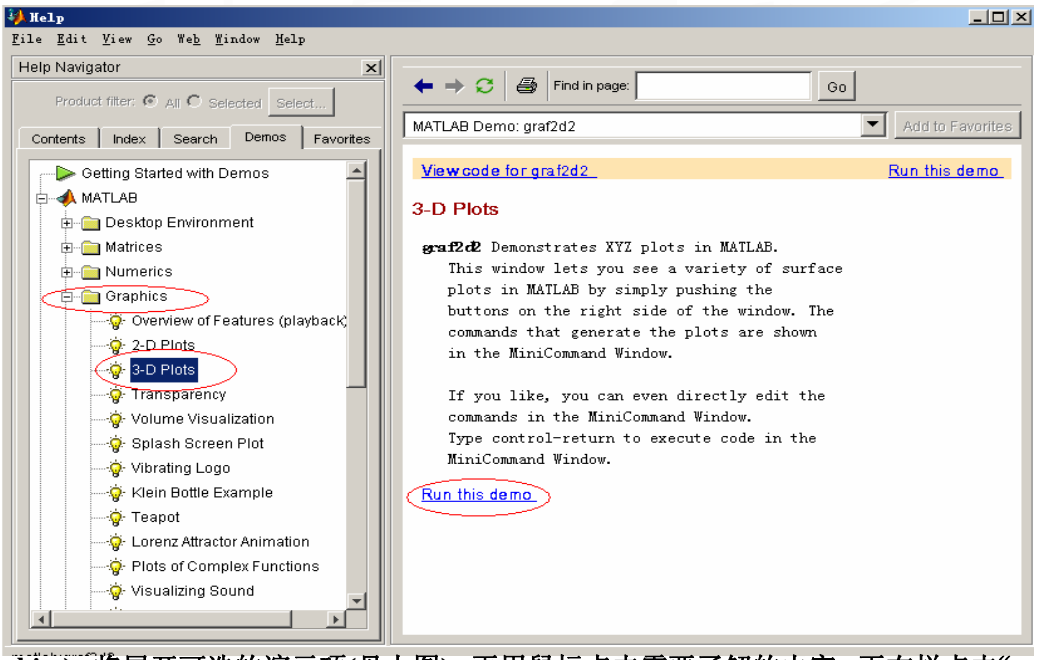
load 文件名（回车）

如果执行命令 load date1 ，MATLAB 会将 date1.mat 文件中所含变量（数据）调入工作空间。

18．如何观看 MATLAB 的入门演示 ？



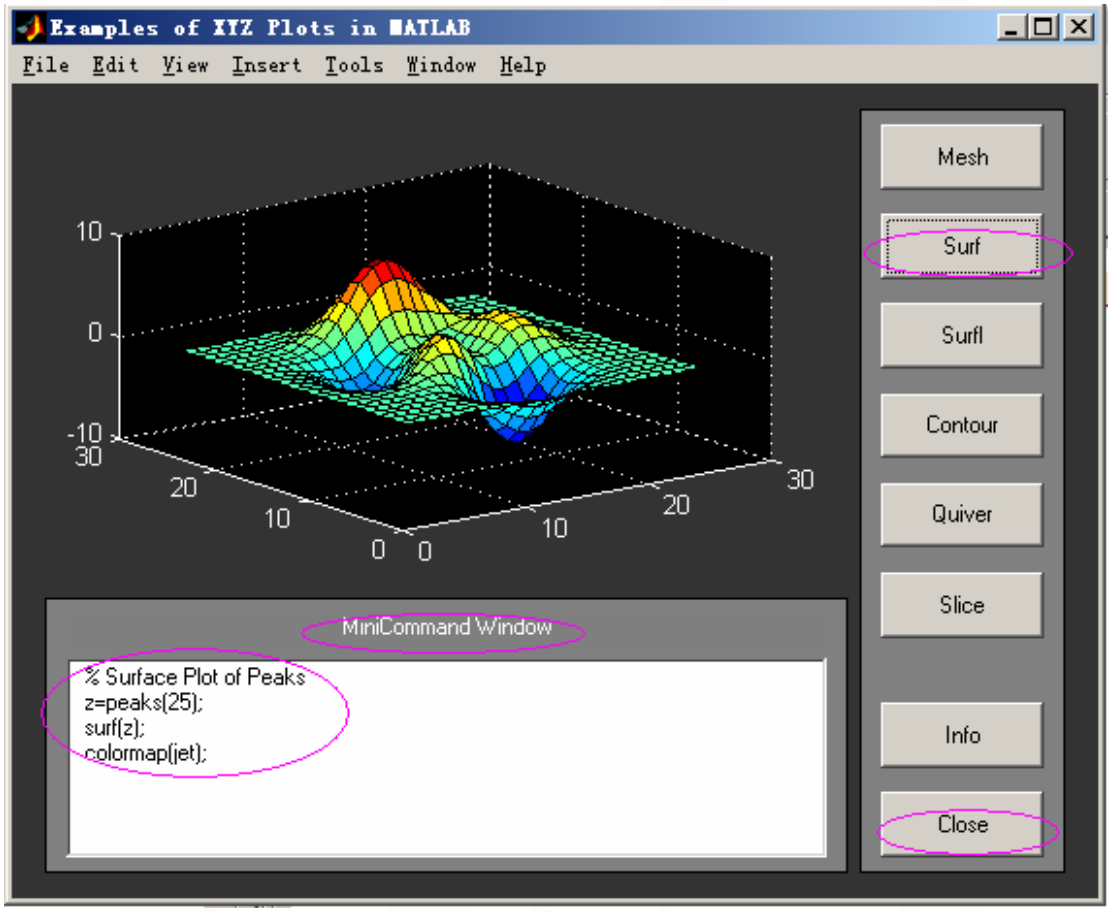
在 MATLAB 的命令窗口中键入“demos”并回车，将进入下图所示的 help 窗口。初学者在观看入门演示中将获得 MATLAB 的基础知识。用鼠标点击窗口左栏选项(如



Graphics)，将展开可选的演示项(见上图)．再用鼠标点击需要了解的内容，再右栏点击“run this demo”，将进入示例演示程序，同时 matlab 的基本命令及使用方法将显示在迷你命令

窗口里。下面，我们以 3-D Plots 为例演示操作步骤.

从这些基本指令的介绍中，你可以看到 MATLAB 用于数值计算和图形绘制的强大功能和易学易用的特点。Demo 中每一步的观看可用鼠标操作，初学者可以多做一些尝试以获得对数学软件 MATLAB 的初步印象，点击“close”会关闭演示.



19．如何将 MATLAB 计算结果保存到 WORD 文稿中？

WORD 是微软公司出品的一个优秀文字处理软件，除了可以方便输入中、英文文字和编辑排版外，它的另一优点就是能快速编辑数学公式。目前世界上很多科技工作者都用这一软件写作论文。为了保证数据结果的正确性，在 MATLAB 与 WORD 之间进行数据传送是很重要的。

数据传送应该是双向进行，传送的意义包括两方面。一是将 MATLAB 工作环境中计算获得的数据结果快速而且准确无误地保存到 WORD 文稿中。二是检验 WORD 文稿中某

一数据结果，如积分值、方程的根等是否正确。这两种传送都可以利用 Windows98 中剪贴板的“粘贴”功能实现。通常用得较多的是第一种数据传送，即由 MATLAB 到 WORD 的数据单向传送。

数据结果可以是单独一个的数据，也可以是多个数据(向量或矩阵)，具体的操作如下

(1) 在 MATLAB 命令窗口中，将数据结果选定；

(2) 在 MATLAB 命令窗口上方菜单栏中，用鼠标单击“EDIT”，在下拉菜单中选择“COPY”；

(3) 切换到 WORD 文稿的窗口，将光标移到数据应该出现的位置；

(4) 用鼠标单击 WORD 工具栏上的“粘贴”按钮。

例如，计算定积分 $\int_0^{\pi} \sin x\,dx$ 的值。在 MATLAB 窗口中键入 quat('sin',0,pi) ，并回车，得 ans = 2.0000。按上面四步操作将数据结果粘贴到 WORD 文稿中，这是单个数据的单向传送。

我们可以利用 WORD 所带的公式编辑器输入符号“ $\int_0^{\pi} \sin xdx =$ ”，然后将积分的数据结果从 MATLAB 命令窗口中粘贴到等号后面。得

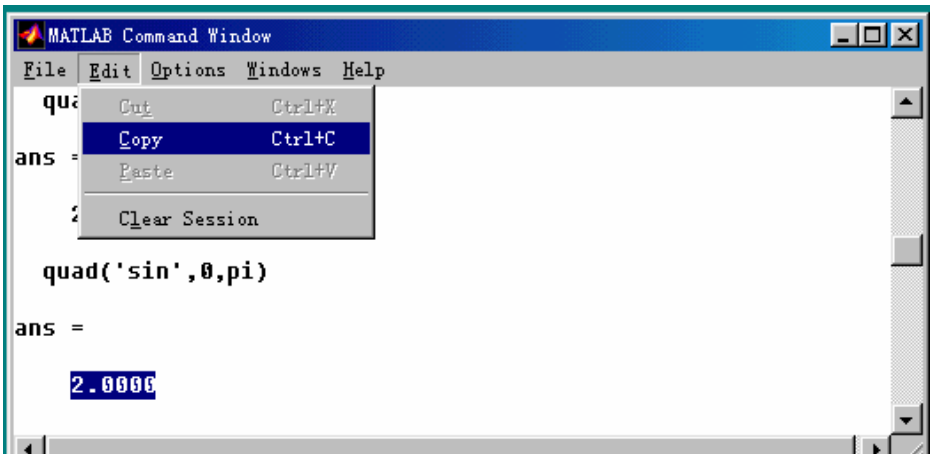


图 1-7 选定 MATLAB 命令窗口中的数据

$$\int_0^{\pi} \sin xdx = 2.0000$$

也可以按上面介绍的操作步骤，将 MATLAB 计算所得的多个数据结果“粘贴”至 WORD 文稿中的合适位置，然后用 WORD 的编辑功能将数据转换为表格。

在 WORD 工作窗口中，用鼠标将“粘贴”而来的多个数据选定，然后选择“表格”菜单中的“将文字转换成表格”即可完成这项工作。

例如计算下列函数值

$$f(x_j)=\int_0^{x_j}\frac{1}{\sqrt{2\pi}}\exp(-\frac{t^2}{2})dt\text{，}\quad(\ x_1=1，x_2=2，x_3=3\ )$$

可以用 MATLAB 计算出三个函数值并粘贴后，在 WORD 文稿中创建下面表格

表 1-5

x	1	2	3
$f(x)$	0.3413	0.4772	0.4987

表格的具体操作请参考 WORD 的有关书籍，如表格居中、表格内各单元的文字符号居中、表格中各单元的高度和宽度“设置值”的调整。结合这些命令可以制做出美观、大方的数据表格。

20．如何将 MATLAB 所绘图形置入 WORD 文稿

文字处理软件 WORD 的另一个突出的优点就是“图文混排”，我们可以在 WORD 文稿中置入图形、图片，使得 WORD 所编辑的文稿丰富多彩。而 MATLAB 的“绘制图形”功能可以很方便地绘出函数的图形，或由数据结果绘出所需要的空间曲线图形、统计直方图、复杂的二维曲面图等等。这些图形都显示在 MATLAB 的图形窗口中，我们可以通过 Windows98 的剪贴板功能将图形直接“粘贴”到 WORD 文稿中。

例如已知曲线的方程为

$$y=e^{-0.5x}\sin 5x$$

现绘制对应的曲线图形。在 MATLAB 命令窗口中键入

```
x=0:0.1:8;
y=exp(-0.5*x) .* sin(5*x);
plot(x,y)
```

MATLAB 的图形窗口中将绘出衰减振荡曲线。为了将图形置入 WORD 文稿中，可以按如下操作进行

(1) 在 MATLAB 的图形窗口中用鼠标单击菜单栏中的“Edit”，在下拉菜单中选择“Copy”；

(2) 切换到 WORD 文稿的编辑窗口，用鼠标单击菜单栏中的“插入”菜单，在下拉菜单中选择“文本框”，“横排”；

(3) 将鼠标光标（十字架）移到文稿中适当的位置处，按住鼠标左键向右下方拖动，出现文本框；

(4) 用鼠标单击 WORD 工具栏上的“粘贴”按钮。

此时，MATLAB 绘制出的图形就被置入 WORD 文稿的文本框中。注意：MATLAB 图形窗口背景是黑色，如果事先将曲线颜色设置为白色，图片粘贴到 WORD 文稿后自动反色，将得到白色背景的黑色曲线图形。

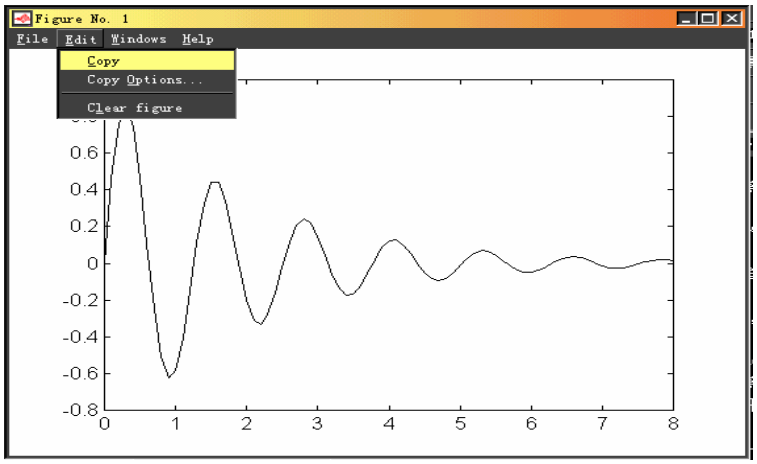


图 1-8 在 MATLAB 图形窗口中选择图形拷贝

将图形“粘贴”到 WORD 文稿的文本框内后，可以设置文本框格式，如文本框边框的颜色、文本框的环绕方式等，有关操作请参考有关 WORD 的书籍。

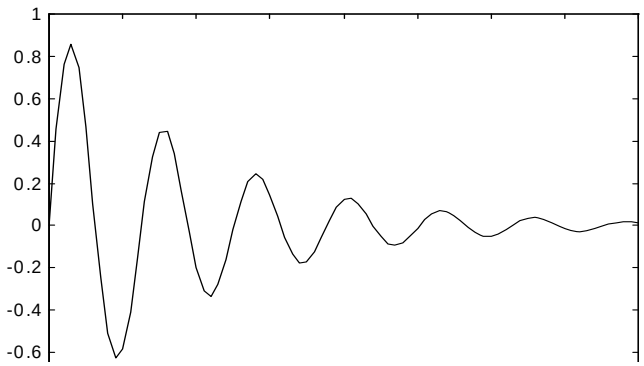


图 1-9 粘贴到 WORD 文稿中的图形

二．基础篇

用 MATLAB 解决一般的数学问题时应熟悉最基本的命令，如求积分、求函数极值、求解常微分方程初值问题以及绘制函数图形等等。有时，为了研究函数需要编辑函数文件，为了提高解决问题的效率需要编写程序，这些正是本篇要介绍的。

21．什么是 MATLAB 的命令操作方式？

所谓命令操作方式就是常用的简单操作方式。为了完成一个作业，用户在 MATLAB 的命令窗口中键入一条（或几条）命令，计算机执行后显示出计算结果。用户观察结果后，修改或重新输入命令，计算机再次执行，再次输出数据结果。这种操作方式也称为“交互式操作方式”或“人机对话”方式。它的特点是

(1) 计算机处理及时。当用户输入数据或键入 *MATLAB* 指令时，计算机将即时接受、及时处理，给出计算结果，并等待人工的下一步操作。

(2) 人工干预性强。完成作业过程中的每一步都需人工键入命令，即使数据有误或某一指令输入有错，也可以重新调出前次或再前次的操作指令进行修改后重新执行。

例如用海伦公式计算三角形面积。已知 ΔABC 的三条边长分别为 $a = 3, b = 5, c = 7$ 由海伦公式，三角形面积为

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

其中，$p=(a + b + c)/2$ 为三角形半周长。

在 MATLAB 命令窗口中键入

```
a=3, b=5, c=7                %输入三条边的长度
p=(a + b + c)/2              %计算三角形半周长
s=sqrt(p*(p-a)*(p-b)*(p-c))  %计算三角形面积
```

每键入一行命令，计算机都会显示出计算结果。p = 7.5000，s = 6.4952 这说明，边长为 $a = 3, b = 5, c = 7$ 的三角形的半周长为 7.5000，面积为 6.4952。

MATLAB 的命令窗口中命令的输入是行编辑方式，可以在一个命令行上输入一条命令，也可以在一个命令行上输入几条命令。当用户输入一条命令后再

击回车键，MATLAB 将执行这条命令，当命令正确无误将显示正确的结果，当命令有误将给出错误信息。如果在一个命令行上同时输入两条以上的命令，则应在指令和指令之间用逗号“，”(或分号“；”)隔开，输入完最后一条命令后击回车键，MATALAB 将按先后顺序逐条执行这一命令行上的每一命令。注意，如果在某一条指令结束处用了分号，则 MATLAB 执行该指令后计算所得的数据结果将不会在屏幕上显示出来。表 2-1 列出了 MATLAB 命令窗口中常用的一些热键，使用这些热键，可在 MATLAB 的环境中充分发挥行编辑功能，提高解决问题的效率。

表 2-1

热键	功能	热键	功能
↑	重调出前一命令	Home	光标移行首
↓	重调出下一命令	End	光标移行尾
←	光标左移一个字符	Esc	擦掉当前行
→	光标右移一个字符	Ins	在光标处插入
^→	右移一个字	Del	删除光标处字符
^←	左移一个字	Backspace	删除光标左字符

22．什么是 MATLAB 的文件操作方式？

文件操作方式主要是指编写 *MATLAB* 程序，然后在 MATLAB 环境中执行程序，如果程序正确，最后可获得计算的数据结果。程序是静态实体，以文件方式存入在磁盘上。

在 MATLAB 桌面上点击“File”按钮，在下拉菜单中选择“new”，从而新建 m 文件，进入 Editor 进行文件操作；也可以利用文字处理软件如 DOS 系统中的屏幕编辑软件 EDIT.EXE 或 Windows 中的记事本 NOTEPAD，在字处理软件的编辑窗口编写程序(输入 *MATLAB* 的命令)，当程序编写好之后将其保存在 *MATLAB* 的工作目录下（程序的文件名以“.m”为后缀）；然后切换到 *MATLAB* 的命令窗口执行程序。让 *MATLAB* 执行程序时，只须键入该程序的文件名即可。

这种文件操作方式又称为批处理操作方式，其特点有：

(1) 自动性。在顺利的情况下(机器运行正常、程序编写无误、……)，程序中的一批指令能自动地逐个被执行，无须人工干预。

(2) 顺序性。程序中的各条指令是按先后顺序被 ***MATLAB*** 所执行，各条指令执行的顺序与它们在程序中的顺序是完全相同的。

例如用海伦公式计算三角形面积的作业可以编写 ***MATLAB*** 的程序来完成。在 **MATLAB** 的命令窗口中键入： **edit herno.m** （回车）此时将进入文字编辑环境，在这一环境中输入以下几行命令

```
a=input('put a=');
b=input('put b=');
c=input('put c=');
p=(a+b+c)/2;
s=sqrt(p*(p-a)*(p-b)*(p-c))
```

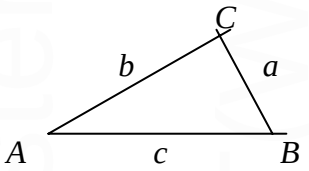


图 2-1

输入完成后，将这一文件（文件名为 **herno.m**）保存在当前目录下，退出文字编辑环境回到 **MATLAB** 的命令窗口，只须键入文件名 **herno**（不须后缀 **m**）并回车，**MATLAB** 将自动执行上面的五行命令。前三行命令是从键盘输入三角形的三条边长数据，最后一条命令是用边长和半周长计算三角形面积并将计算结果显示在计算机的屏幕上。根据屏幕出现提示信息，依次输入三个数据：

```
3(回车)
4(回车)
5(回车)
```

计算机运行后，显示所求三角形面积：**s = 6.** ***MATLAB*** 的程序实际上是一个有序的指令集合，所以这种操作方式又称为批处理方式。批处理方式实际上是从第一条指令开始，按顺序连续执行指令集合中的各条指令，直到最后一条指令执行完毕。如果程序中某一条指令有错，将输出出错信息，并停止程序的执行。 如果一个程序中有 ***n*** 条指令，***MATLAB*** 批处理操作方式的工作流程将如图 **2-2** 所示。

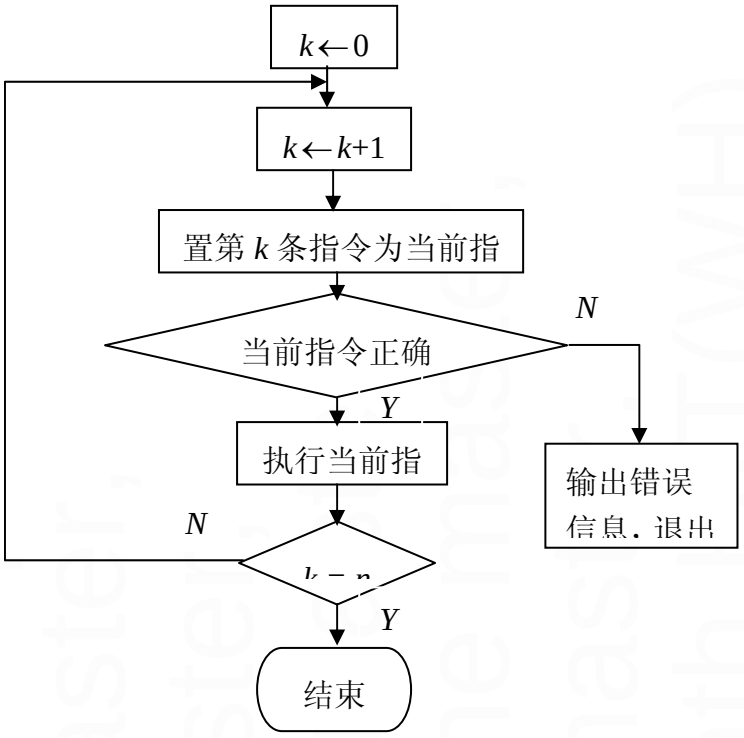


图 2-2 批处理方式的工作流程图

23．如何编辑 MATLAB 的函数文件

由于 **MATLAB** 内部函数有限，为了研究某一个函数的各种性态（极大、极小值、积分值等）需要为 **MATLAB** 定义新函数，为此必须编写函数文件。函数文件的文件名也是以“***.m***”为后缀，这类文件的第一行必须以字符 **function** 开始，格式如下

function 因变量名 = 函数名(自变量名)
函数体

MATLAB 函数文件第一行特殊字符及格式决定了它与一般 **MATLAB** 程序的区别，函数体主要是根据函数结构由形式自变量计算所得结果赋值给因变量。为了区别，**MATLAB** 的程序一般又称为非函数文件，尽管函数文件和非函数文件都是以“***.m***”为文件名的后缀，但两种文件不能混在一起，函数和程序不能编写在一个文件中。可以在命令窗口中或程序中调用已编辑好的函数。例如为了研究函数

$$f(x)=\sqrt{(x-20)^2+100^2}+\sqrt{(x-120)^2+120^2}$$

的极值点，首先要编写 MATLAB 能识别的函数文件。

在 Matlab 环境下用鼠标点击菜单栏上的”File”，并在下拉菜单中选择”new”|”M-file”将进入 Matlab 的 EDITOR 环境，在其中录入下列两行:

function yy=f2(x)

yy=sqrt((x-20).^2+100^2)+sqrt((x-120).^2+120^2);

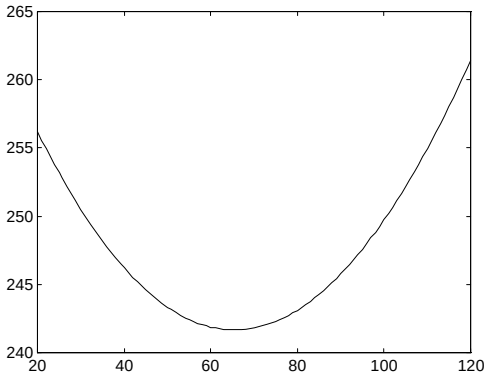


图 2-3

注意，函数名和文件名相同，均是 f2。另外，x 是函数中的自变量，而 yy 是函数因变量。函数定义中实际上是将自变量数据经过计算并赋值给因变量。因为自变量数据一般是数组，故涉及自变量的乘法和方幂均应该用“点乘”和“点方幂”。录入完毕后,将文件存盘并退出 EDIT，然后重新回到 MATLAB 环境下。这时，可用指令

x=20:120；y=f2(x)；

plot(x,y)

确定自变量值并计算对应的函数值数据，最后再绘出函数的图形（图 2-3），从图形可以得知，该函数有一个极小值点大约在 $x=65$ 处。也可以用指令

x=60：70；y=f2(x)

计算出当自变量取 **60，61，62，…，70** 时，对应的函数值数据为：

241.8674 241.7985 241.7436 241.7027 241.6756 241.6623

241.6630 241.6774 241.7057 241.7477 241.8034

显然，第六个数据 **241.6623** 最小。这正是该函数在 $x=65$ 处的函数值。

如果函数的结构比较简单，可以使用 **inline** 函数在内存中定义一个临时函数. 例如，我们计算一个半径为 **R** 的圆内正 **n** 边形面积. 取 $\alpha=2\pi/n$ **(见图 2-4)**，根据面积公式有

$$S(n)=R^2 n\sin\frac{\pi}{n}\cos\frac{\pi}{n}$$

令 **R=1**，在 **Matlab** 的命令窗口中键入

S=inline(‘n*sin(pi/n)*cos(pi/n)’,’n’)

[s(3),s(6),s(12),s(100),s(200),s(500),s(1000)]

可得下面一系列数据

1.2990 2.5981 3.0000 3.1395 3.1411 3.1415 3.1416

由此可见，多边形面积逐步逼近圆面积，即有

$$\lim_{n\to\infty} S(n) = \pi R^2$$

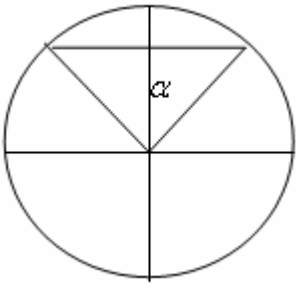


图 2-4

24. 什么是 MATLAB 中的“.”运算

为了使两个同维的矩阵（或向量）的每一对元素同时作相同的运算，可以用这种“.”运算，又称为按元素运算。在编写 MATLAB 的函数文件时，为了同时计算出函数在一组自变量数据处的值，一般都要用到这种“.”运算。如果在一个函数文件中，只用一般的乘、除、乘幂运算而不用点运算，这样的函数一次只能计算单个自变量数据的值。

例如，下面两个矩阵

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},\quad B=\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$

做点运算，可以分别考虑“点乘”、“点除”和“点乘幂”三种运算。

（1） 点乘“.*”（按元素乘）运算。***A .*B*** 实际所做运算为

$$\begin{bmatrix} 1\times1 & 2\times1 & 3\times1 \\ 4\times2 & 5\times2 & 6\times2 \\ 7\times3 & 8\times3 & 9\times3 \end{bmatrix}$$

计算结果为

```
ans =
     1     2     3
     8    10    12
    21    24    27
```

（2） 点除“./”或“.\”（按元素右除或按元素左除）运算。***A ./B*** 或 ***A .\B***

计算结果为

ans = **ans =**

1.0000 2.0000 3.0000 1.0000 0.5000 0.3333
2.0000 2.5000 3.0000 0.5000 0.4000 0.3333
2.3333 2.6667 3.0000 0.4286 0.3750 0.3333

（3） 点乘幂“.^”（按元素乘幂）运算。*A* .^*B*
计算结果为

ans =

1	**2**	**3**
16	**25**	**36**
343	**512**	**729**

如果欲求 ***B*** 矩阵的每个元素的倒数形成的同型矩阵，应该用命令“**1. /*B***”而不能用“**1 / *B***”。**1. / *B*** 的计算结果为

ans =

1	**1**	**1**
1/2	**1/2**	**1/2**
1/3	**1/3**	**1/3**

25．如何建立和使用自己的工作目录？

MATLAB 安装成功以后，在硬盘上将形成一个 **MATLAB** 目录（从 **WINDOWS** 桌面“我的电脑”进入 **C** 盘可看到一个“**MATLAB**”文件夹），在这一目录中包含了四个子目录，其中子目录 **\work** 非常重要，它包含了 **MATLAB** 系统运行文件、帮助文件以及一些必要的二进制文件。当我们进入 **MATLAB** 后，**C:\MATLAB\WORK** 将自动成为当前目录。用户如果不加修改，那么我们在 **MATLAB** 工作环境下所创建的所有函数、程序以及产生的数据文件等等，都将保存在这一当前目录下。

为了管理好自己的 **M** 文件和其它文件，用户应该建立和使用自己的工作目录。例如在 **C** 盘上建立一个名为 **USER** 的目录，将用户自己的文件都存放在这

一目录中将会大有好处。具体操作如下：

（1） 从 Windows 桌面“我的电脑”进入 C 盘；

（2） 用鼠标选择屏幕上方“文件”菜单中的“新建”| “文件夹”，C 盘上将出现一个名为“新建文件夹”的文件夹；

（3） 将“新建文件夹”改名为“USER”。

这时用户在 C 盘上有了自己的工作目录：USER。为了使用自己的工作目录，在进入 MATLAB 后，在 MATLAB 环境下键入以下命令：

CD C：\USER （回车）

就将用户目录指定为 MATLAB 当前工作目录。系统的当前目录不再是 C：\MATLAB\BIN 而是 C：\USER。以后用户在 MATLAB 环境下工作时所创建的所有函数、程序，保存的变量（数据）也都将存放在这一目录下。

26．如何绘制函数 $y(x)=\frac{\sin x}{x}$ 的图形?

考察微积分中重要的函数极限之一，即 $y(x)=\frac{\sin x}{x}$ 当 *x* 趋于零时的极限。绘制该函数图形的困难之处在于当 *x*=0 时，分母为零。由于该函数在零点的极限值为 1，故当 x 接近于 0 时，有 $y(x)\approx 1$。可补充定义函数如下

$$f(x)=\begin{cases}\sin x/x, & x\neq 0\\ 1, & x=0\end{cases}$$

为了避免分母为零，可作如下特殊处理，在操作时将 $x=0$ 处理为非常接近于零的数，使其误差可忽略不计。为了达到这一效果可以借用 MATLAB 中的一个特殊常数 $eps=10^{-16}$（浮点数的相对误差），将所有自变量的离散数据统一都加上 *eps*。这样既使得 *x* 在零点处的值只是机器数零，又使得 *x* 在其它点处的值不会与正确值有较大的误差。

可以用如下程序段绘图

```
x=-12:0.5:12;x=x+eps;          %取 x 的离散值并做特殊处理
y=sin(x)./x;                   %计算对应的函数值
plot(x,y)                      %根据函数的离散数据绘图
```

计算机运行后，MATLAB 的图形窗口将出现图 2-4 中的曲线图形

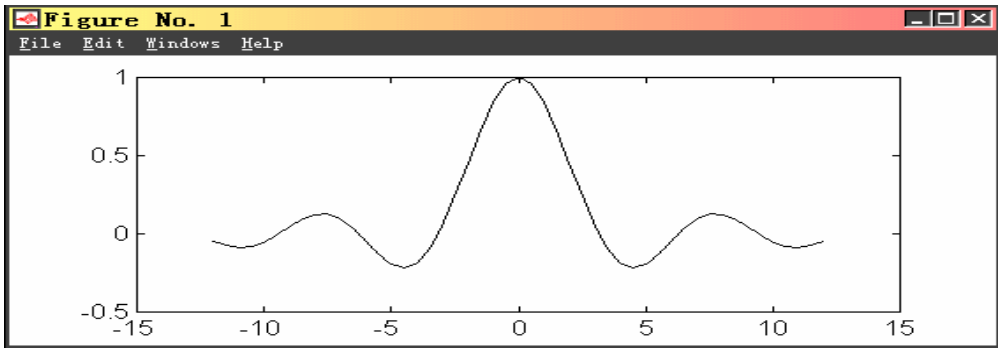


图 2-4　函数 y=sin x / x 的图形

27．如何绘制函数 $y(x)=(1+\frac{1}{x})^x$ 的图形?

另一个重要的函数极限是 $y(x)=(1+\frac{1}{x})^x$ 当 *x* 趋于无穷大时的极限，即

$$\lim_{x\to\infty}(1+\frac{1}{x})^x = e \approx \mathbf{2.71828046915643}$$

由于 $y(1000000)=$ **2.71828046915643**，故这一数据可以用作该极限的近似值。表 2-2 给出了五个点处的函数值数据，显然当 x 在 100 以内时，函数值变化很大；而当 x 在 100 到 100000 范围内时，函数值变化较小。所以在取有限个自变量数据时应选取自变量点的间隔为对数步长，即当 x 的值越大时离散点的间隔越大。

表 2-2

x	10	100	1000	10000	100000
y	2.5937	2.7048	2.7169	2.7181	2.7183

为了用有限个点的函数值绘出较为光滑的函数曲线图形，可利用 MATLAB 中的对数步长命令产生自变量的离散数据。用下面程序段绘图。

```
x=logspace(1,3);          %取 10 到 1000 之间有限个对数步长的自变量值
y=(1+ones(size(x))./x).^x;    %计算对应的函数值数据
plot(x,y)                 %绘图
```

计算机运行后，**MATLAB** 的图形窗口将出现图 **2-5** 中的图形

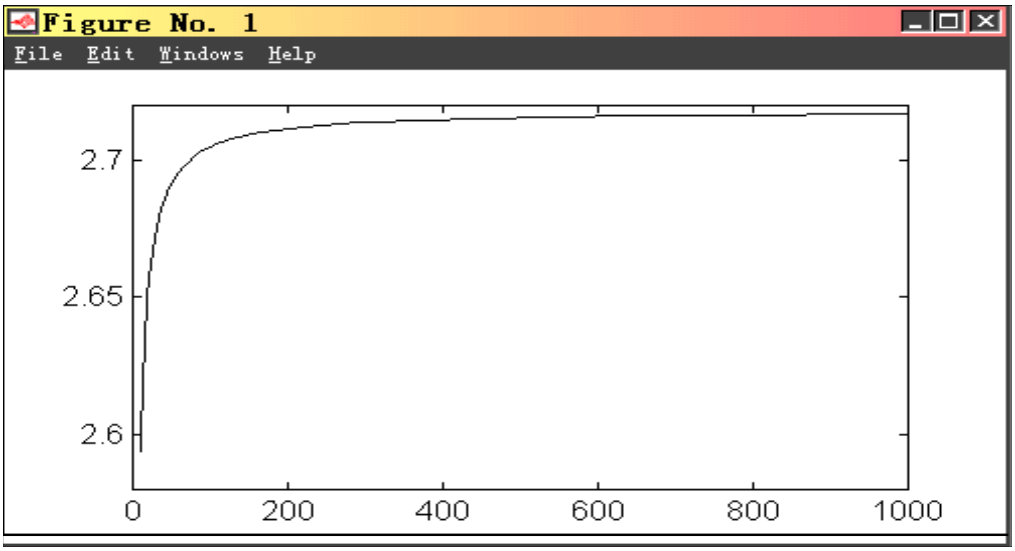


图 2-5　函数 $y=(1+1/x)^x$ 的图形

28．如何计算定积分 $\int_a^b f(x)dx$ 的值？

MATLAB 中求定积分的指令为 **quad**，具体使用这一命令的格式为 **quad('函数名', a, b)**。使用中，要用到被积函数的调用，也要注意给定积分上下限。例如求定积分

$$I=\int_0^{\pi}\sin x\,dx$$

的值，在 **MATLAB** 环境下直接键入下面指令

quad('sin',0,pi)

计算机运行后，屏幕将显示

ans =　　2.0000

这表明用指令 **quad** 直接计算出积分 $I=\int_0^{\pi}\sin x\,dx=2$。

在上面的计算中由于正弦函数是 **MATLAB** 的一个内部函数，所以可以直接调用，而对于任意一个连续函数的定积分计算，就必须先定义被积函数才能用 **quad** 指令求积分值。

例如求定积分

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}}dx$$

的值，必须先编辑被积函数的文件（函数文件名：**ff3.m**）如下：

```
function y=ff3(x)
y=exp(-x.^2/2)/sqrt(2*pi);
```

将这一函数文件保存在当前工作目录下后，可以直接调用函数 **ff3(x)**，在 **MATLAB** 环境下用有穷积分来求该积分的近似值，选取积分限为**-4** 到**+4** 积分：

```
quad('ff3',-4,4)
```

MATLAB 计算出积分的近似值为 **ans = 0.9999**

这实际上是用有限定积分

$$\frac{1}{\sqrt{2\pi}}\int_{-4}^{+4} e^{-\frac{x^2}{2}}dx$$

代替原来的无穷区间上积分的值。

29．如何求椭圆周长？

如果已知椭圆长半轴 a 和短半轴 b，则该椭圆的参数方程为

$$\begin{cases} x = a\cos t \\ y = b\sin t \end{cases}$$

根据求曲线弧长的定积分公式，椭圆周长为

$$S = 2\int_0^{\pi} \sqrt{a^2\sin^2 t + b^2\cos^2 t}\ dt$$

在上面的定积分中被积函数为

$$f(t) = 2\sqrt{a^2\sin^2 t + b^2\cos^2 t}$$

例如，求长半轴为 **20**，短半轴为 **10** 的椭圆周长。首先编辑如下函数文件：

```
function y=f3(t)
y=2*sqrt((20*sin(t)).^2+(10*cos(t)).^2);
```

将这一函数文件录入后存放在工作目录下，回到 **MATLAB** 环境中可以进接调用这一被积函数做定积分，键入如下命令：

```
quad('f3',0,pi)
```

计算机屏幕将显示

ans = 96.8853

这说明，所求的椭圆周长为 **96.8853**

30．怎样用 fplot 画一元函数的图形？

在 **MATLAB** 中绘制一元函数的图形（即平面曲线图形）除了用 **plot** 命令绘画外，还可以用 **fplot** 直接画一元函数图形。使用 **fplot** 这一指令绘图的最大特点是不用确定自变量的离散数据并计算对应的函数值。计算机在执行这一条指令时，将根据所给自变量取值区间来自动选取自变量结点（离散数据），并计算出对应的函数值数据，然后绘图。这对于一般用户来讲，更有利于提高工作效率。使用这一指令时，需要先定义函数，同时还要在指令中使用方括号，在方括号内给定四个数据作为自变量取值区间和函数值显示范围。

例如绘制正切函数的部分图形（图 **2-6**），可直接用下面指令

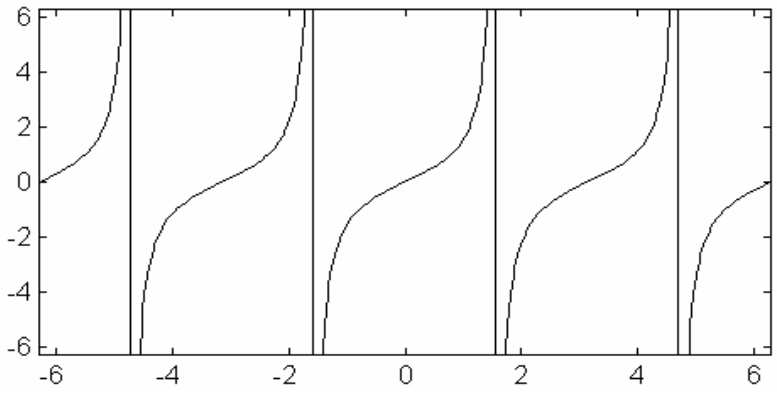


图 2-6 正切函数图形

fplot('tan',[-2*pi 2*pi -2*pi 2*pi])

由于正切函数是 **MATLAB** 的内部函数，所以不需要编缉函数文件定义函数而可以直接调用；方括号中的四个数据分别表明所绘图形在 **X** 和 **Y** 两个方向上均是由 -2π 到 2π 之间。如果要直接绘制出任意的一个函数图形，则必须事先编辑函数文件，然后再使用直接绘图命令。**fplot** 命令使用格式为

fplot(’函数名’，[Xmin Xmax Ymin Ymax])

或 fplot(’函数名’，[xmim xmax])

例如，绘制函数

$$f(x)=\frac{1}{(x-0.3)^2+0.01}+\frac{1}{(x-0.9)^2+0.04}-6$$

在区间[0，2]内的图形。首先编辑函数文件（文件名为 hh.m）

```
function    y=hh(x)
y = 1 ./ ((x-.3).^2 + .01) + 1 ./ ((x-.9).^2 + .04) - 6;
```

将这一文件保存在当前目录下，在 MATLAB 环境中用命令

```
fplot('hh',[0,2]);
```

将绘出该函数的图形如图　所示

如果用如下命令

```
[x    y] = fplot('hh',[0,2]);
```

图形窗口中将不绘出该函数的图形，而只是自动选取有限个自变量的离散点的值并计算出该函数在对应自变量离散点处的函数值。

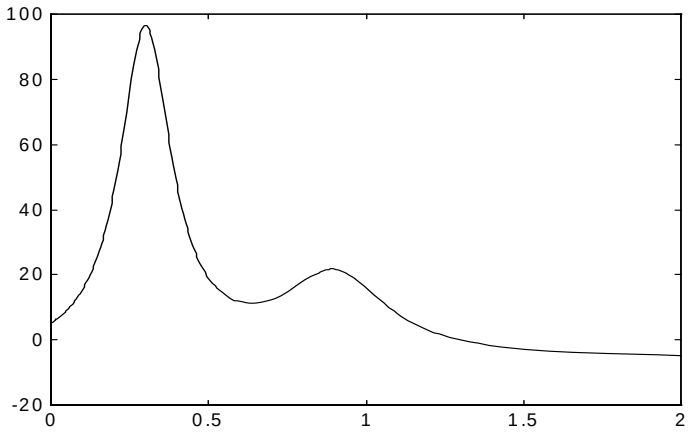


图 2-7

31．如何求一元函数的极小值？

求一元函数极小值用命令 ***fmin***。这一命令的使用格式为

***fmin*('文件名'，x1，x2)**

通常，要事先编写好函数文件用以定义所研究的函数 F，同时在使用这一命令时要给定极小值点存在的区间[x_1，x_2]。计算机执行这一指令后，将求出函数 F(x)在区间 (x1, x2) 内的局部极小值点。其中，F 是所考虑的函数的函数文件名。

例如，键入：

***fmin*('cos'，3，4)**　　（回车）

计算机将计算出余弦函数在区间（3, 4）内的局部极小值点

ans =　　3.1416

一般情况下，对所考虑的函数必须要事先编缉该函数的函数文件，然后再使用求函数极小值点的指令。而极小值点存在的范围也要人为给定，在对该函数

不太了解时不能给出合适的极值点范围，这时可以结合 ***fplot*** 指令直接绘出该函数的局部图形，通过观察函数曲线的变化情况，估计出极小值点的范围（极小值点存在的区间）。如果要求一个函数在某一区间内的极大值，只须在编写函数文件时，人为地将函数值取反号。在 **MATLAB** 命令窗口仍可以用 ***fmin*** 命令求极大值。

32．如何求一元函数的零点？

求一元函数零点用指令 ***fzero***，但是必须要事先给出一个初值 x_0（零点的初始近似值）。具体使用格式为

***fzero*('f name', x_0)**

由此可计算出函数名为 **fname.m** 的函数在点 x_0 点附近的零点.

例如，在 **MATLAB** 环境下输入

fzero('sin',3)

计算机运行后可计算出

ans = 3.14159265358979

这一零点的值实际上是 π 的值。注意：在一般情况下，要首先编辑函数文件以定义函数，并且大概估计出所研究的函数的零点的位置（初始点近似值），最后再使用 **fzero** 指令求函数的零点。本问题中正弦函数是 **MATLAB** 中的内部函数，所以不用编写函数文件。

33．如何求一阶常微分方程初值问题的数值解？

常微分方程初值问题的数学描述如下：

$$\begin{cases} \dfrac{dy}{dx} = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

这里，$f(x, y)$ 为常微分方程的右端函数，而 y_0 为所求未知函数的初始值。

求解常微分方程初值问题用指令 **ode23** 或 **ode45**。使用这两条命令中的任何一条都必须事先编写好函数文件并保存在工作目录下（如取文件名为 **yprime.m**）。命令的具体使用格式为

[x,y] = ode23('yprime', x_0, x_n, y_0)

其中，**yprime** 为描述常微分方程右端函数的函数文件名，而 x_0 和 x_n 分别为自变量的初始点和终值点，y_0 为未知函数的初值。

例如求一阶常微分方程

$$\begin{cases} \dfrac{dy}{dx} = x - y + 1, & x > 0 \\ y(0) = 1 \end{cases}$$

在（**0**，**1**）区间内的数值解，并与该初值问题的解析解

$$y(x) = x + e^{-x}$$

进行比较。

首先编缉两个函数文件，第一个用于描述微分方程右端函数(文件名：**ff1.m**)：

```
function   z=ff1(x,y)
z=x-y+1;
```

另一个用于描述微分方程的解析解（文件名：**ff2.m**）：

```
function   y=ff2(x)
y=x+exp(-x);
```

将这两个函数文件保存在工作目录下，然后求出初值问题的数值解以及微分方程解析解在对应自变量的离散点处的函数值，最后同时绘出两个函数的图形加以比较。在 **MATLAB** 环境中键入下列指令：

```
[x,y]=ode23('ff1',0,1,1);
y=ff2(x);
plot(x,y,'o',x,y)
```

计算机屏将显示出数值解（用小圆 **o** 表示）和解析解的图形如图 **2-8** 。

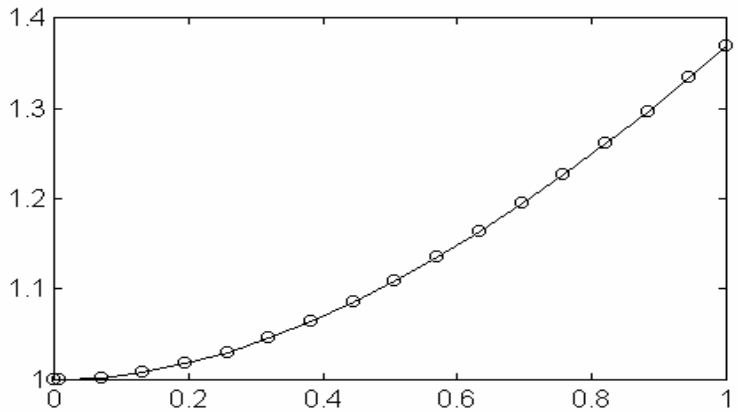


图 2-8 数值解与解析解的比较

34．如何求高阶常微分方程初值问题的数值解？

根据常微分方程求解理论，高阶常微分方程初值问题可以化为与之等价的一阶常微分方程组初值问题来求解。所以仍然可以用 **ode23** 指令求解高阶常微分方程初值问题。

例如，著名的追赶曲线问题：有动点 **Q** 从点 $\mathbf{Q_0(100, 0)}$ 出发，以 **1** 米/秒的速度平行于 **Y** 轴的正向方向运动，另一动点 **P** 从原点 **O(0, 0)**出发，以 **2** 米/秒的速度追赶 **Q** 点，**P** 在运动过程中方向总是指向 **Q** 点。已求得 **P** 点的轨迹满足的微分方程为

$$\begin{cases} y'' = \dfrac{\sqrt{1+y'^2}}{2(100-x)} & x \in (0\ ,\ 100) \\ y\big|_{x=0} = 0,\ \ y'\big|_{x=0} = 0. \end{cases}$$

求解该方程并画出追赶曲线。

首先,将问题转化为一阶常微分方程组的初值问题

$$\begin{cases} \dfrac{dy_1}{dx} = y_2\ , & y_1(0) = 0 \\ \dfrac{dy_2}{dx} = \dfrac{\sqrt{1+y_2{}^2}}{2(100-x)}, & y_2(0) = 0 \end{cases}$$

为了求解上面的一阶常微分方程初值问题，首先编辑描述常微分方程右端函数的函数文件（文件名：**rab.m**）

```
function z=rab(x,y)
z(1)=y(2);
z(2)=sqrt(1+y(2).^2)./(2*(100-x));
```

将这一函数文件保存在工作目录下，并在 **MATLAB** 环境中键入下面指令：

```
y0=[0 0];
[x,y]=ode23('rab',0,99.99,y0);
plot(x,y(:,1),'o')
```

计算机屏幕将显示出追赶曲线的图形如图 **2-9**

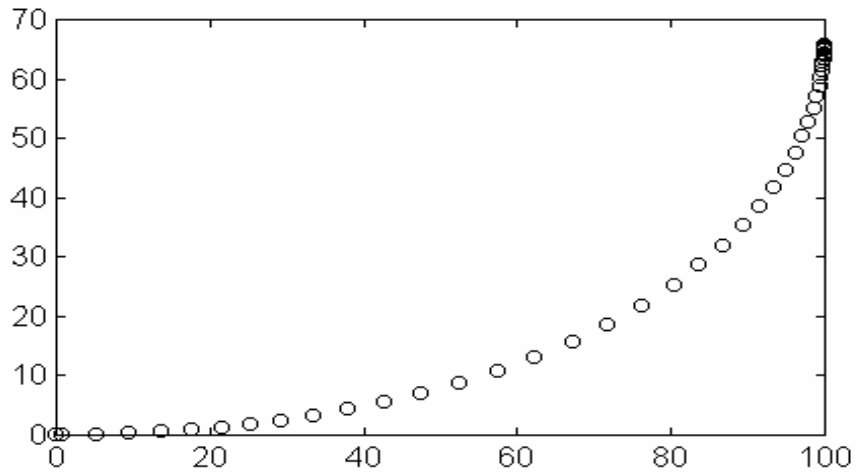


图 2-9 追赶曲线图形

35．MATLAB 的局部变量和全局变量有何区别？

MATLAB 的函数文件中除了自变量、因变量以外，在计算过程中如果用到了中间变量或其它的变量，这些都将是局部变量。局部变量只在这一函数中有效，在其它函数或程序中无效。例如，将三角形面积 **S** 视为三条边长 **a,b,c** 的函数，则可以编缉一个函数文件(文件名为：**heron.m**)来实现海伦公式的计算：

```
function   s=heron(a, b, c)
p=(a+b+c)/2;
s=sqrt(p*(p-a)*(p-b)*(p-c));
```

将这一函数文件保存在工作目录下，在 **MATLAB** 环境中，键入如下指令

```
heron(3, 4, 5)
```

计算机执行后，将显示出计算结果：**ans= 6**。在这一函数文件中，三角形的半周长 ***p*** 就是一个局部变量。调用这一函数时，计算机用实际的数据 **3**、**4**、**5** 代替形式变量 ***a***、***b***、***c*** 计算出变量 ***p*** 的值，然后再计算出 ***s*** 的值，最后将数据结果输出到屏幕上。但是只要计算出函数值，则自变量的实际数据和中间变量都将被自动删除，计算机内存中只保留了函数值变量（缺省变量名为 ***ans***）。

如果想保留中间变量 ***p*** 的值，就必须将它定义为全局变量。定义 ***p*** 为全局变量用命令

global p

注意：这一命令必须出现在两个地方，一是函数文件中第一次使用变量 ***p*** 之前；二是在命令窗口第一次调用这一函数文件之前。

当一个作业的初始数据较多或算法比较复杂时，用函数文件的方法来完成作业不太方便，这时常用非函数文件（即程序）来实现算法，这不但方便，更利于调试。

36. 多项式运算有哪些命令？

常用多项式运算命令有多项式的创建和多项式求值等。一个 ***n*** 次多项式函数

$$P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$$

的各项系数确定了一个 **(*n*+1)** 维向量

$$p=[a_0\ \ a_1\ \ a_2\ \ \cdots\ \ a_n]$$

反之，如果给定了一个（***n*+1**）维向量 $p=[a_0\ \ a_1\ \ a_2\ \ \cdots\ \ a_n]$，则可以将各元素作为多项式的系数按降幂形式写出一个多项式。

（**1**） 用创建向量的方法创建一个 ***n*** 次多项式

例如创建一个 **8** 次多项式

$p(x)=x^8-36x^7+546x^6-4536x^5+22449x^4-67284x^3+118124x^2-109584x+40320$

可以用命令

***p*=[1 -36 546 -4536 22449 -67284 118124 -109584 40320]**

（**2**） 用命令 ***p*=*poly*（*R*）**创建多项式

如果知道一个 ***n*** 次多项式的全部根

r_1、r_2、…… 、r_n

则可以用向量 $R=[r_1\ \ r_2\ \ \cdots\ \ r_n]$，结合命令 ***p*=*poly*（*R*）** 创建 ***n*** 次多项式

$$p(x)=x^n+a_1x^{n-1}+a_2x^{n-2}+\cdots\cdots+a_{n-1}x+a_n$$

即产生出该多项式的系数

$$p=[1 \quad a_1 \quad a_2 \quad \cdots \quad a_n]$$

显然，有

$$(x-r_1)(x-r_2)\cdots(x-r_n)=x^n+a_1x^{n-1}+a_2x^{n-2}+\cdots+a_{n-1}x+a_n$$

若 **R** 是 **n** 阶方阵，则 **p=poly(R)** 将获得 **R** 的特征多项式。

（3）求一个多项式的全部零点

多项式的零点（高次代数方程的根）用命令 ***roots(p)***

例如求 **8** 次代数方程

$x^8-36x^7+546x^6-4536x^5+22449x^4-67284x^3+118124x^2-109584x+40320=0$

的根，可以用MATLAB语句

```
p=[1  -36  546  -4536  22449  -67284  118124  -109584  40320];
roots(p)
```

得 8 个根分别为：8 7 6 5 4 3 2 1

如果修改 7 次幂的系数 -36 为 -37 再求新的 8 次方程的根，可用下面的命令

```
p(2)=-37;
roots(p)
```

得新的8次方程的根如下

```
ans =   16.1192      5.0351 + 5.1497i   5.0351 - 5.1497i   2.8210 + 1.7281i
        2.8210 - 1.7281i   2.0844 + 0.2494i   2.0844 - 0.2494i   0.9998
```

比较两次求根结果，发现多项式系数的微小变动会引起多项式的根的显著变化。

（4） 计算一个多项式在 $x=s$ 处的值用命令：***polyval(p, s)***

这里，***s*** 可以是自变量的某一个确定的数据，**s** 也可以是自变量的一组确定的数据。

37．如何求多元函数的极小值？

求多元函数的极小值用指令 **fzeros**，使用的具体格式为

fzeros('fname ',x_0)

使用这一指令时，必须事先编缉文件名为 **fname.m** 的函数文件(文件名可由操作者自己选定)，用以描述所研究的多元函数；同时还要给出极小值点的近似

值(初始值)x_0，对于多元函数而言，x_0 是一个常向量。

考虑三村短路问题(Steiner 问题)：已知 *A*、*B*、*C* 三地，*AB*=5(公里)，*BC*=6(公里)，*AC*=7(公里)。现要找一地 *H*，使 *AH*+*BH*+*CH* 为最短，并求出 *AH*+*BH*+*CH* 的最小值。

以 *A* 为原点建立平面直角坐标系，让 *X* 轴通过 *AC* 连线(如图所示)。设 *H* 点的坐标为 $(x,\ y)$，则 *A*(0,0),*C*(7,0),*B*(x_0，y_0)，(*B*点坐标可以通过计算获得)

于是问题转化为求二元函数

$$S(x,y)=\sqrt{x^2+y^2}+\sqrt{(x-7)^2+y^2}+\sqrt{(x-x_0)^2+(y-y_0)^2}$$

的最小值问题。

由于 AB=5，BC=6，于是B 点坐标$(x_0,\ y_0)$满面足方程组

$$\begin{cases} {x_0}^2+{y_0}^2=25 \\ (x_0-7)^2+{y_0}^2=36 \end{cases}$$

求解，得

x_0=2.7143，　y_0=4.1991

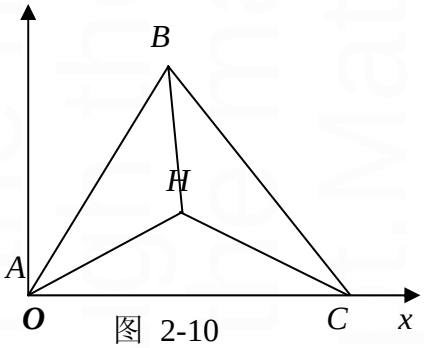


图 2-10

首先定义目标函数 S(x) ，注意，这时自变量 x 是含有两个元素的向量。建立文件名为 s.m 的函数文件如下：

```
function w=s(x)
x0=[2.7143 4.1991];x1=[7 0];
w=sqrt(x*x')+sqrt((x-x0)*(x-x0)')+sqrt((x-x1)*(x-x1)');
```

将上述函数文件存放于工作目录下。然后取初始点为 B 点的坐标，编辑如下文件名为 steiner.m 的程序如下：

```
x0=[2.7143 4.1991];x1=[7 0];          %输入B点和C点的坐标
h=fmins('s',x0)                       %求目标函数的最小值点并输出结果
d(1)=sqrt(h*h');
d(2)=sqrt((h-x0)*(h-x0)');
d(3)=sqrt((h-x1)*(h-x1)');
d                                     %输出AH,BH,CH 的(距离)数据
s=sum(d)                              %求目标函数最小值并输出数据结果
```

在MATLAB环境下运行程序 **steiner.m** ，（即直接键入文件名 **steiner** 并回车)屏幕将显示计算结果如下:

```
h=2.8847    1.9736                    % H 点的坐标
d=3.4952    2.2320    4.5641          % AH,BH,CH 的数据
s=10.2913                             % AH+BH+CH 的值
```

38. MATLAB 有哪些数据输出格式？

在 **MATLAB** 工作环境中，所有参加运算的数据都是以双精度存贮方式进行，即具有十进制的十五位有效数。但是，如果没有特殊的要求，数据只以五位十进制数被显示出来。例如，在 **MATLAB** 环境中键入 **pi** 并回车，屏幕将显示出数学常数 π 的值 **3.1416**。如果键入 **exp(1)** 并回车，屏幕将显示出数学常数 ***e*** 的值 **2.7183**。这两个数都是精度不太高的近似值，它们的位数只有五位，这种显示方法被称为短格式方式。如果我们要想看到π 的更精确的数据值，可以利用数据显示的长格式方式。键入指令 **format long** ，再键入 **pi** ，屏幕将显示出数据 **3.14159265358979**，这是具有十五位的数据。如果再键入 **format**，以后的数据显示方式将以短格式的方式进行。

选择数据输出格式的方法有两种：一是用键盘命令，二是用菜单选择。

(1) MATLAB 可以用键盘命令 **format** 加不同的“开关”使数据具有不同的输出格式。表 **2-3** 改变数据输出格式命令

命令	开关	功能
format		缺省时为默认短格式方式与 **format short** 相同
format	**short**	短格式方式，显示 **5** 位定点十进制数。
format	**long**	长格式方式，显示 **15** 位定点十进制数。
format	**hex**	十六进制格式方式。
format	**bank**	银行格式。按元、角、分（小数点后具有两位）的固定格式。
format	+	+格式，以＋，—和空格分别表示中的正数，负数和零元素
format	**short e**	短格式 **e** 方式，显示 **5** 位浮点十进制数
format	**long e**	长格式 **e** 方式，显示 **15** 位浮点十进制数。
format	**rat**	分数格式形式。用有理数逼近显示数据。如 **pi** 显示为

		355/113。
format	**loose**	**松散格式。数据之间有空行。**
format	**compact**	**紧凑格式。数据之间无空行。**

(2) 在 MATLAB 命令窗口中也可以用鼠标选择菜单“Options”。如图 2-11 所示

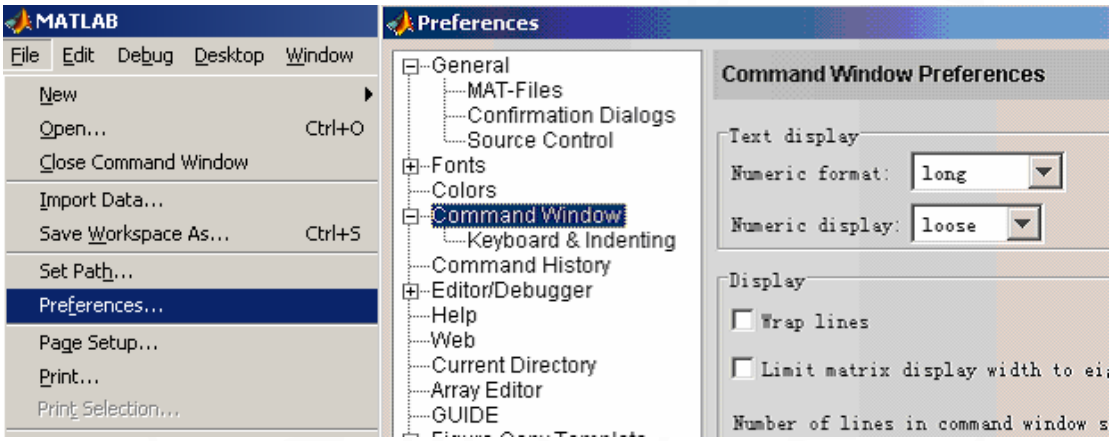


图 2-11 数据输出格式的菜单选择

在下拉菜单中选择第一条“Numeric Format”，会出现十种数据输出格式的选择菜单。再用鼠标单击即可。

39. 在 M ATLAB 中如何使用特殊矩阵？

MATLAB 中的特殊矩阵比较多，常用的有下面几类：

（1） 单位矩阵。产生 *n* 阶单位矩阵用命令

***eye*(*n*)**

例如，*eye*(4)将产生 4 阶单位矩阵

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

（2） 全“1”矩阵。产生 *m*×*n* 阶全“1”矩阵用命令

***ones*(*m*，*n*)**

例如，*ones*(3，4) 将产生 3×4 阶的全“1”矩阵

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

（3） 全零矩阵。产生 ***m*×*n*** 阶全“**0**”矩阵用命令

***zeros*(*m*，*n*)**

例如，***zeros*(3，4)** 将产生 **3×4** 阶的全“**0**”矩阵

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

（4） 对角矩阵。产生 ***n*** 阶对角矩阵用命令

$$diag([a_1 \quad a_2 \quad \cdots \quad a_n])$$

例如，***diag*([1 2 3 4])**将产生 **4** 阶对角矩阵

$$\begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 3 & \\ & & & 4 \end{bmatrix}$$

如果要提取一个矩阵 ***A*** 的主对角元素，可以用命令

***diag*(*A*)**

例如，在 **MATLAB** 工作环境中已经创建了一个 **3** 阶矩阵

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

则命令 ***diag*(*A*)** 将提取出矩阵 ***A*** 的主对角元素并形成一个列向量

$$ans = [\,1 \quad 5 \quad 9\,]^T$$

（5） 范德蒙矩阵。命令

***vander*([1 2 3])**

将产生一个 **3** 阶范德蒙矩阵

1 1 1
4 2 1
9 3 1

（6） 幻方矩阵。命令

***magic*(3)**

将产生一个 **3** 阶幻方矩阵（纵、横相加得数均相等）

8 1 6
3 5 7

4 9 2

（7） 希尔伯特矩阵。命令

***hilb*(3)**

将产生一个 3 阶希尔伯特矩阵

1	**1/2**	**1/3**
1/2	**1/3**	**1/4**
1/3	**1/4**	**1/5**

（这种输出格式是分数形式，在产生矩阵之前用了命令 ***format rat*** ）

40．如何求一个方阵的特征值与特征向量？

求一个方阵 ***A*** 的特征值用命令 ***eig*(*A*)**，求特征向量也用同一命令。即取如下格式

[*v d*] = *eig*(*A*)

例如，求矩阵

$$A=\begin{bmatrix} -1 & 1 & 0 \\ -4 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

的特征值和特征向量，可以用命令

[*v d*] = *eig*(*A*)

MATLAB 执行这条命令后，屏幕将显示出特征值和特征向量

***v* =**

0	**0.4082**	**-0.4082**
0	**0.8165**	**-0.8165**
1.0000	**-0.4082**	**0.4082**

***d* =**

2	**0**	**0**
0	**1**	**0**
0	**0**	**1**

这里，对角矩阵 ***d*** 中的三个对角元 **2**、**1** 、**1** 是 ***A*** 矩阵的三个特征值，即

$$\lambda_1=2，\ \lambda_2=1，\ \lambda_3=1$$

而矩阵 ***v*** 的三个列向量

$$\alpha_1=\begin{bmatrix}0\\0\\1\end{bmatrix},\quad \alpha_2=\begin{bmatrix}0.4082\\0.8165\\-0.4082\end{bmatrix},\quad \alpha_1=\begin{bmatrix}-0.4082\\-0.8165\\0.4082\end{bmatrix}$$

分别对应于上面三个特征值，即 α_1 是 $\lambda_1=2$ 的特征向量，α_2 是 $\lambda_2=1$ 的特征向量，α_3 是 $\lambda_3=1$ 的特征向量。（这种数据输出格式是默认的定点数短格式）

41．如何用正交变换将一个二次型化为标准形式？

利用正交变换化二次型为标准型，应首先写出二次型所对应的对称矩阵，然后用正交变换将这一对称矩阵化为对角矩阵。在 **MATLAB** 中用 **schur** 分解的指令可以求出所需对角矩阵和所用的正交矩阵，从而完成这一任务。例如，有下面的二次型

$$f(x_1,x_2,x_3)=x_1^{\ 2}+2x_2^{\ 2}+3x_3^{\ 2}+4x_1x_2-4x_2x_3$$

为了将其化为标准形，首先写出二次型所对应的矩阵

$$A=\begin{bmatrix}1&2&0\\2&2&-2\\0&-2&3\end{bmatrix}$$

现在对 **A** 作 **schur** 分解，可以获得二次型标准型所对应的对角阵以及用于正交变换的正交矩阵。为了适应于一般的线性代数习题，还需要将数据格式转换为有理数（分数）形式。

A=[1 2 0;2 2 -2;0 -2 3]; format rat

[u, t]=schur(A)

上面三行指令运行后，可由矩阵 **A** 分解出正交矩阵 **u** 和对角矩阵 **t**。所得数据结果为

$$u=\begin{bmatrix}\frac{2}{3}&-\frac{2}{3}&-\frac{1}{3}\\-\frac{2}{3}&-\frac{1}{3}&-\frac{2}{3}\\-\frac{1}{3}&-\frac{2}{3}&\frac{2}{3}\end{bmatrix},\qquad t=\begin{bmatrix}-1&0&0\\0&2&0\\0&0&5\end{bmatrix}$$

矩阵 **u** 正是用于将矩阵 **A** 正交对角化的正交矩阵。三个矩阵 **A**，**u**，**t** 满足如下关系

A = u*t*u'　　和　　u'*u = I

如果令 $[y_1 \quad y_2 \quad y_3]=[x_1 \quad x_2 \quad x_3]u$ ，则有

[x_1　x_2　x_3] = [y_1　y_2　y_3] u^{-1}

而 $u^T=u^{-1}$，由此可知，所用正交变换及所求标准二次型分别为：

$$\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}\frac{2}{3} & -\frac{2}{3} & -\frac{1}{3}\\ -\frac{2}{3} & -\frac{1}{3} & -\frac{2}{3}\\ -\frac{1}{3} & -\frac{2}{3} & \frac{2}{3}\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix},\qquad f(x_1,x_2,x_3)=-y_1^{\,2}+2y_2^{\,2}+5y_3^{\,2}$$

42．如何计算出 30 个概率积分值？

设随机变量 ***X*** 服从标准正态分布，即　***X*** ～ ***N*(0, 1)**。现分别取

u_α **= 0.1，0.2，0.3，…，2.9，3.0**

为了计算出 **30** 个概率积分值

$$P\{X>u_\alpha\}=\int_{u_\alpha}^{+\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}dx\approx\int_{u_\alpha}^{4}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}dx$$

可以选取定积分中的不同积分下限值，用 **MATLAB** 的求积分指令 **quad** 配合循环语句来达到所求的数据。

首先编辑描述被积函数的函数文件，取文件名为 **fai.m**

```
function y=fai(x)
y=exp(-x.^2/2)/sqrt(2*pi);
```

将这一文件保存在工作目录下，然后再编辑一个文件名为 **prop.m** 的程序如下：

```
a=0;h=.1;
for k=1:30
    a=a+h;
    p(k)=quad('fai',a,4);
end
```

将这一文件保存在工作目录下，回到 **MATLAB** 环境下，运行名为 **prop.m** 的程序，只需要键入

prop

并回车，就可以计算出 30 个概率积分值（保存在数组变量 p 中）。在 MATLAB 环境中直接键入 **p** 并回车，屏幕将显示出所计算出的数据结果如表 2-4 所示

表 2-4

编号	积分值	编号	积分值	编号	积分值
1	0.4601	11	0.1356	21	0.0178
2	0.4207	12	0.1150	22	0.0139
3	0.3821	13	0.0968	23	0.0107
4	0.3445	14	0.0807	24	0.0082
5	0.3085	15	0.0668	25	0.0062
6	0.2742	16	0.0548	26	0.0046
7	0.2419	17	0.0445	27	0.0034
8	0.2118	18	0.0359	28	0.0025
9	0.1840	19	0.0287	29	0.0018
10	0.1586	20	0.0227	30	0.0013

43．如何产生两类常用的随机数？

在 **MATLAB** 中有专用的随机数发生器，可以产生均匀分布的随机数以及标准正态分布的随机数。这两个命令分别为 ***rand*** 和 ***randn***。

rand*(*m*，*n*)** 将产生出一个 ***m*×*n 阶的矩阵，矩阵的每个元素均为区间 **(0，1)** 中的数。即 ***rand*** 产生服从均匀分布 ***U*(0，1)** 的随机变量 ***X*** 的伪随机数（由计算机模拟）。

若为了模拟某一随机事件，需要八个均匀随机数，用如下命令

***x* = *rand*(1，10)**

将得到八个元素的行向量，例如

***x* = [0.5269　0.0920　0.6539　0.4160　0.7012　0.9103　0.7622　0.2625]**

如果用这八个随机数模拟八次掷硬币的结果，并规定掷出硬币为正面的结果用随机数大于 **0.5** 表示。则上面的随机数依次代表了掷硬币结果为

正、反、正、反、正、正、正、反。

同理，命令 ***randn*(*m*，*n*)** 将产生出一个 ***m*×*n*** 阶的随机数矩阵，矩阵的每个元素均为区间（－3，3）中的数。由于标准正态随机变量其均值 μ =0，方差 σ =1，根据概率论中著名的 3σ 原则， ***randn*** 产生服从标准正态分布 ***N*(0，1)** 的随机变量 ***X*** 的伪随机数。

例如，为了模拟某一学生班 30 个同学的《高等数学》期末考试成绩，已知

最高分数不超过 100 分，最低分数不低于 40 分，平均分数为 70 分，为了模拟这一随机事件，可用如下命令

$$f = randn(30，1)*10 + 70$$

产生 30 个随机数的列向量，例如表 2-5

表 2-5

f_k	分数	f_k	分数	f_k	分数
f_1	74.3871	f_{11}	62.4030	f_{21}	60.9797
f_2	57.5266	f_{12}	63.2528	f_{22}	49.4674
f_3	73.2467	f_{13}	58.2831	f_{23}	70.8909
f_4	73.9007	f_{14}	90.3293	f_{24}	90.8710
f_5	65.9486	f_{15}	79.6848	f_{25}	73.6512
f_6	72.9231	f_{16}	76.7029	f_{26}	78.4611
f_7	95.6591	f_{17}	74.2015	f_{27}	68.1546
f_8	65.4218	f_{18}	41.2725	f_{28}	80.3071
f_9	53.8917	f_{19}	86.8587	f_{29}	54.7238
f_{10}	43.3048	f_{20}	70.2792	f_{30}	79.6494

由表 2-5 中数据可知，最高分为学号第 7 号的同学，分数为 95.6591，最低分为学号为 18 号的同学，分数为 41.2725。经计算可得，这 30 个同学的平均分数为 69.5545。

44.如何绘制极坐标所描述的曲线图形？

有时人们须用极坐标来描述平面曲线，MATLAB 可以通过曲线的极坐标方程的两个变量来绘制对应的曲线图形，
这时可以用 polar 指令。

具体使用格式为：

polar(THETA, RHO)

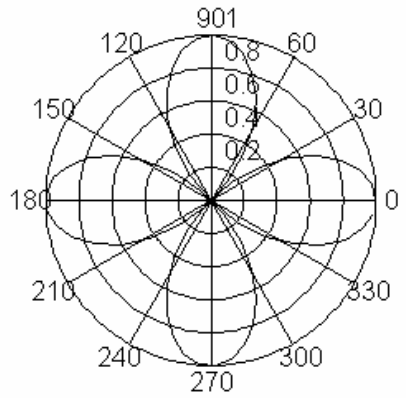
这里， THETA 为角度变量，而 RHO 为极径变量。

例如用极坐标绘图指令绘制四叶玫瑰线的图形，其极坐标方程为

$$r = \cos 2\theta \quad , \quad (0 \le \theta \le 2\pi)$$

在 MATLAB 环境中键入下面指令：

theta=0:0.1:2*pi;



```
r=cos(2*theta);
polar(theta,r)
```

运行上面三条语句，可得图 2-12 所示图形

图 2-12 四叶玫瑰线图形

45．如何绘制统计直方图？

统计直方图是人们根据已经得到的数据，将数据按不同的小区间分类，统计各小区间内数据出现的频率。由频率值画出的一种条形图。

画直方图指令用 **hist**。

已知某班 36 个大学生的单科成绩，根据这些分数的数据画统计直方图。

```
X=[84 82 60 75 66 92 92 80 86 89 …
   85 100 84 89 91 95 78 85 81 85 …
   91 83 82 87 72 91 81 63 88 88 …
   88 92 93 98 84 84];
hist(X)
```

上面第一条语句是给变量 X 赋值，X 代表 36 名大学生的成绩分数数据；第二条指令是用绘直方图指令 hist 绘制出十等分直方图。如果将 hist(X) 改为指令

```
hist(X,4)
```

则绘出的是 4 等分直方图，如图 2-13 所示，4 等分直方图表示 60～69，70～79，80～89，90～100 四个分数段内的分数出现的频率大小。由此可知，绘直方图指令中第一个参数代表数据集合，第二个参数控制直方图中的等分数，其默认值为 10，要改变默认值就必须给出具体的数据。

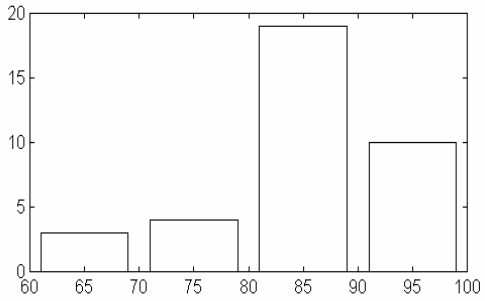


图 2-13 四等分直方图

46．如何绘制空间曲线的图形？

在 MATLAB 中绘制一条或多条空间曲线用命令 ***plot3***。它的使用方法类似于绘制平面曲线命令 ***plot*** 。设空间曲线 ***L*** 的表达式为

$$\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases} \qquad \alpha \le t \le \beta$$

首先确定出曲线中参数 ***t*** 的离散点数据，然后根据 ***t*** 的数据计算出曲线上坐标 ***x***，***y***，***z*** 的值（所得 ***x***，***y***，***z*** 是同维数的向量），最后按如下命令格式绘图

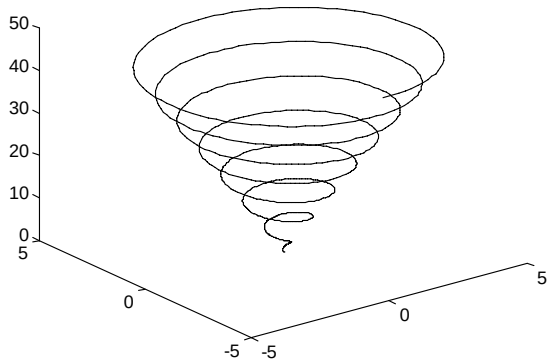
***plot*3(*x*，*y*，*z*)**

其中，*x*，*y*，*z* 为空间空间曲线上点的坐标。例如，用下面程序段

```
t = 0：pi /50：15* pi;
x = 0.1* t .* sin(t)；y = 0.1* t.* cos(t)；z = t;
plot3(x，y，z)
```

可以绘出曲线

$$\begin{cases} x = 0.1\times t\times \sin t \\ y = 0.1\times t\times \cos t \quad 0\le t\le 15\pi \\ z = t \end{cases}$$



的图形如下所示

图 2-14 空间曲线图形

如果 *x*，*y* ，*z* 是三个同型矩阵，则命令 ***plot*3(*x*，*y*，*z*)** 将同时绘出以三个矩阵的列向量为坐标的几条空间曲线图形。

另外，与二维曲线绘图命令一样，对曲线的颜色与画线方式也可以作不同选择。下面命令中

***plot*3(*x*，*y*，*z*，’*s*’)**

s 是一个或两个选择参数。对于不同的颜色选取和画线方式选取可以参考二维绘图命令 *plot* 中的参数列表。

47．如何用 MATLAB 绘制曲面的图形？

MATLAB 绘制三维曲面的网面图形指令用 **mesh**，在使用这一指令之前必须给出曲面在 **X-Y** 平面内的规则离散点所对应的函数值(曲面高度)数据。

例如，绘制二元函数

$$z=\frac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}$$

的三维网面图。这一函数的图形很类似一顶巴拿马草帽，如图 **2-16** 所示。下面的程序体现的主要算法是，先产生 **34** 个介于**-8** 到**+8** 之间的 **X** 坐标数据，以行向量 ***x*** 表示；再产生同样多的 **Y** 坐标数据以 ***y*** 表示；然后用命令 ***meshgrid*** 将这两个向量分别扩展为矩阵（第一个矩阵 ***x*** 每一列元素相同，第二个矩阵 ***y*** 的每一行元素相同）。这两个矩阵分别代表了 **X-Y** 平面上十分规则的网格结点的坐标，任取两个矩阵在某一位置的对应元素形成的数据对代表了平面上某个点处的坐标。根据坐标数据计算出对应的二元函值数据，最后绘出图 **2-15** 中图形。程序如下：

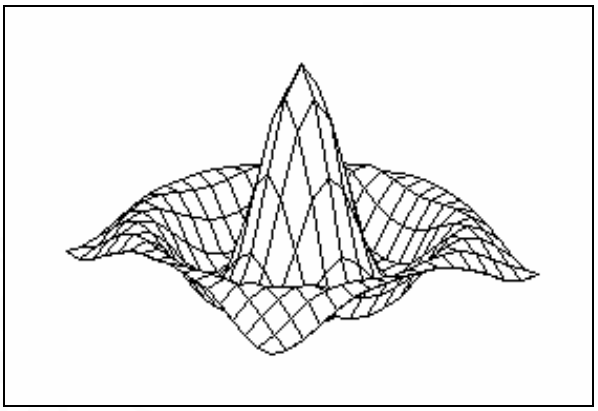


图 2-15 巴拿马草帽图形

```
x=-8:0.5:8;y=x;
[x, y]=meshgrid(x,y);
r=sqrt(x,^2+y.^2)+eps;
z=sin(r)./r;
mesh(z)
```

运行上面程序，**MATLAB** 的图形窗口将出现图 **2-15** 所显示的图形。

48．如何绘制圆域上的曲面图形？

图 **2-17** 中的空间曲面图形是由二元函数

$$f(x,\ y)=3(x^2+y^2)-x^3$$

在圆域

$$D=\{(x,\ y)\mid x^2+y^2\leqslant 4\}$$

上绘出的图形。

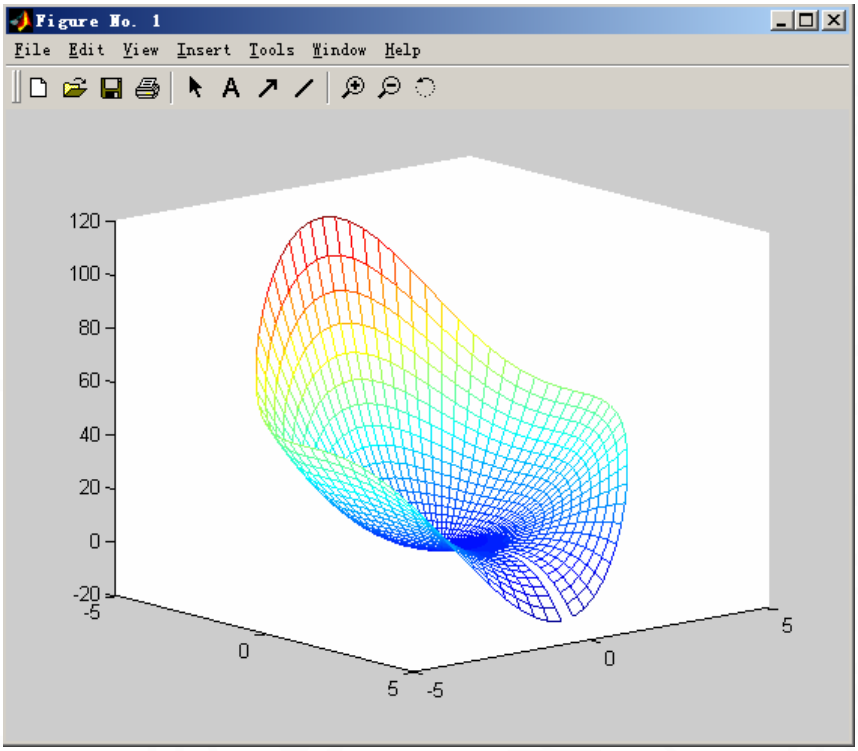


图 2-16 圆域上的曲面图形

绘制这一曲面需要用到 **MATLAB** 的命令 ***mesh*(*x*，*y*，*z*)**，使用这一命令时要求 ***x***，***y***，***z*** 是同型的矩阵。为了达到要求，可以作如下变换，令

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases} \quad 0 \le r \le 4,\ \ 0 \le \theta \le 2\pi$$

由于 **(*r*，** θ**)** 的变化范围是规则的矩形区域 ***G* = [0，4]×[0，2** π**]**，可以将 **[0，4]** 分为 **20** 等分，将 **[0，2** π**]**分为 **72** 等分。于是，由 ***G*** 上的离散点坐标所算出的 ***x*** 离散值和 ***y*** 的离散值就形成了同型矩阵，由这两个同型矩阵按“**.**”运算计算出 ***z* = *f*(*x*，*y*)** 函数值也是同型的矩阵。 用如下程序段绘出曲面图形

```
r=(1：20)/5；tha=(1：72)*pi/36；
x=r'*cos(tha)；y=r'*sin(tha)；
z=3*(x.^2+y.^2)-x.^3；
mesh(x，y，z)
```

为了从另一个角度来观察该图形，再加上命令 **view(50，15)** 可得到上面所示的图形。

49. 如何用从不同角度观察一个三维曲面图形？

MATLAB 允许用户从各种不同的角度去观察一个已存在的三维图形。从用户自己定义的视角观察三维图用命令 ***view***，这一命令的使用格式为

***view*(*az*，*el*)**

命令中的两个参数，***az*** 和 **el** 分别表示方位角和俯视角。方位角相当于球坐标中的经度，俯视角相当于球坐标中的纬度，这两个角度的基准面和方向如图 **2-17** 所示。

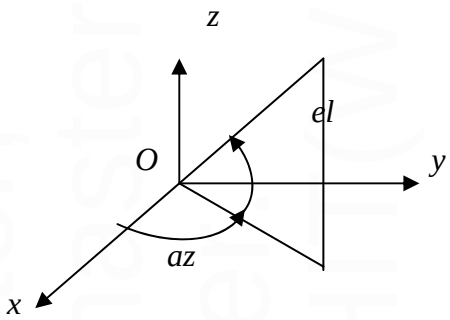


图 2-17 方位角和俯视角

一般用三维曲面绘图命令 ***mesh*** 所绘出的曲面图形默认视角为

***az* = −37.5 ^{0}**

***el* = 30 ^{0}**

一般用二维绘图命令例如 ***plot*** 所绘出的图形（在 ***y-z*** 平面上）默认视角为

az = 0 ^{0}

el = 90 ^{0}

例如，**48** 问题中的圆域上的曲面（形如座椅），用命令 ***view*(130，15)** 可得如下图形。

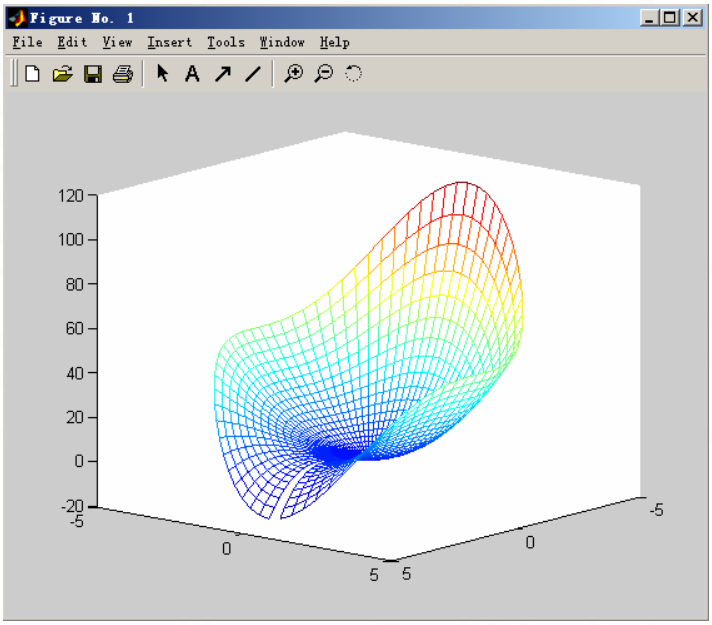


图 2-18 自选角度观看曲面图形

50．如何绘制填充图形？

绘填充图形用命令 *fill*。例如，用红色填充一个三角形，这个三角形的三个顶点坐标分别为 $P_1(0，0)$、$P_2(1，0)$、$P_3(1，1)$。于是三角形顶点的横坐标和纵坐标可以用两个行向量分别表示

横坐标数据： [0 1 1]

纵坐标数据： [0 0 1]

利用 *fill* 命令结合三角形顶点的横坐标向量、纵坐标向量，以及填充颜色可以绘出填充三角形。如下命令

fill([0 1 1]，[0 0 1]，’r’)

可绘出图 2-19 中图形。再例如，为了绘制出正弦函数 *sin x* 在区间 [0.5，2.5] 上的曲边梯形的填充图形，先令 *a* = 0.5，*b* = 2.5，然后选取有限个自变量的离散点数据

x = [*a* *a*+0.1 *a*+0.2 …… *b*]

并计算出对应的正弦函数值的离散数据

y = [*sin a* *sin* (*a*+0.1) …… *sin b*]

为了绘填充图形，必须由正弦曲线上的离散点加上 *x* 坐标轴上的点 (*a*，0) 和 (*b*，0)形成封闭的曲线，封闭曲线的离散点横坐标和纵坐标分别为

a *a* *a*+0.1 …… *b* *b*

和

0 *sin a* *sin* (*a*+0.1) …… *sin b* 0

利用这两组数据就可以绘曲边梯形的填充图了。为了使图形清楚，还可以事先画出正弦函数在一个包括[*a*，*b*]的大区间上的曲线图。用下面程序段可绘出图 2-20 中图形。

```
x=0：0.1：3.1；y = sin(x)；
plot(x，y)，hold
a = 0.5；b = 2.5；
x = a：0.1：b；y = sin(x)；
fill([a，x，b]，[0，y，0]，’r’)
```

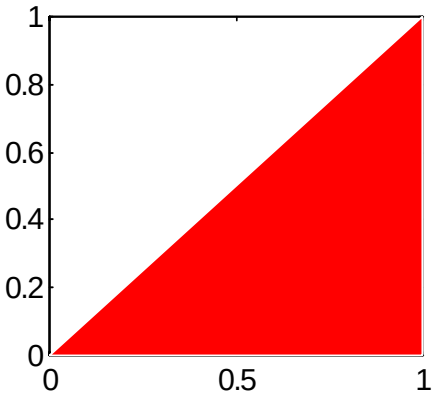


图 2-19 三角形填充

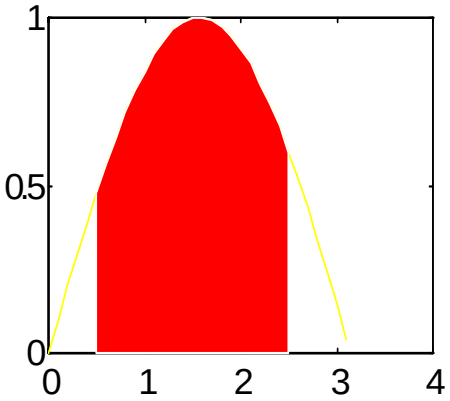


图 2-20 曲边梯形填充

三．提高篇

用 **MATLAB** 解决数学问题时，可用交互式（人机对话形式）的操作方式解决较简单的问题以完成作业，在解决一般的问题时，大多数情况下都需要设计程序，运行、调试程序。设计 **MATLAB** 的程序时，需要熟悉一些常用的 **MATLAB** 语句，了解它们的使用方法和技巧。程序绝不是语句、命令的简单组合。运行 **MATLAB** 程序时，如果出现错误信息，则需要对程序（文件）进行改写，如此反复多次才可能编写出优秀的 **MATLAB** 程序。

51．如何使用 *if* 条件语句？

在 **MATLAB** 中，可以利用条件语句来实现分支算法，最简单的条件语句是 **if** 语句。当条件满足时，执行条件后的指令；当条件不满足时，则不执行条件后的指令。

其具体格式为：

```
if  <条件>
    ……
    ……    语句块
end
```

在 **if** 语句的<条件>选项中至少应有一个关系（或逻辑）表达式，当表达式的逻辑值为真时，则执行<条件>后面的语句块包含的指令；当<条件>中的表达式的逻辑值为假时，则不执行<条件>后面的语句块。表达式通常由变量、数据、关系运算符和逻辑运算符组成，如当 ***a*=2**，***b*=3** 时，表达式

$$a + b > 10$$

为假（不成立）。在建立表达式时，常用的一些符号列出如下(表 3-1)

表 3-1　常用关系符

符号	名称	意义
= =	双等号	等于
<	小于号	小于
>	大于号	大于
<=	小于与等于号	小于等于
>=	大于与等于号	大于等于
~=	上波号与等号	不等于
&	与号	逻辑与
|	竖杠	逻辑或
~	上波号	逻辑非

例如，计算函数 $f(x)=\dfrac{\sin x}{x}$ 的值。当自变量 $x \neq 0$ 时，按函数的表达式计算函数值；当 ***x* = 0** 时，定义函数值为 **1**。求该函数值的程序用到了分支算法。用于分支算法的 **if** 语句，其格式为:

```
if  < 条件 1 >
    ..........
    ..........      语句块 1
else
    ..........
    ..........      语句块 2
end
```

```
程序
x=input('input
x:   ');
if   x = = 0
     y=1;
else
     y=sin(x)/x;
end
```

上面语句的算法结构是：当条件 1 满足时，执行语句块 1；当条件不满足时，则执行语句块 2。当条件较多时必须用多分支算法。用于多分支算法的 **if** 语句，其格式为:

```
if < 条件 1 >
   ……
   ……        语句块 1
elseif < 条件 2 >
   ……
   ……        语句块 2

else
   ……
   ……        语句块 3-
end
```

52．如何使用 *for* 循环语句？

循环算法的实现有两种形式：一是数控制的循环，另一种是条件控制的循环。前一种形式严格规定循环应进行的次数，达到循环次数后则终止循环；后一种规定了循环应满足的条件，满足条件则进入循环，不满足条件则不进入循环。**MATLAB** 中的 **for** 循环语句用于数控制的循环的实现，通过规定循环变量的初值和终值以及步长的设定来达到控制循环次数的目的。从循环初值开始，重复执行某些操作，直至循环变量达到终值为止。具体格式为：

```
for <循环变量>=<初值>:[步长:]<终值>
   ……
   ……        循环体
   ……
end
```

for 循环语句中，当初值小于终值时，步长为正数；当初值大于终值时，步

长应为负数。当步长被省略时，**MATLAB** 默认值为 **1**。

例如，著名的裴波拉奇数列的递推公式为

$$f_n = f_{n-1} + f_{n-2}，\ (n = 3, 4, \cdots)$$

它的第一项和第二项的值都是 **1**，即 f_1**=1**，f_2**=1**。我们可以通过数列的前两项初值，利用递推公式计算出裴波拉奇数列第三项到第十项中各项的值。注意 **MATLAB** 不允许数组的下标为零或负值，所以数组或矩阵的下标只能用正整数。递推公式的计算用“***for*** 循环语句”来实现比较方便。程序如下

```
f(1)=1; f(2)=1;
for   k=3: 10
        f(k)=f(k-1)+f(k-2);
end
f
```

计算机运行后，屏幕显示

```
f =    1    1    2    3    5    8    13    21    34    55
```

这表明计算机计算出裴波拉奇数列第三项到第十项的值分别为

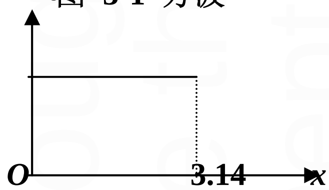
$$f_3 = 2、f_4 = 3、f_5 = 5、\cdots、f_{10} = 55。$$

53. 如何说明方波是奇谐波的叠加

如下定义的分段函数

$$f(t) = \begin{cases} 1 , & t \in [0, \pi] \\ 0 , & t \notin [0, \pi] \end{cases}$$

图 3-1 方波



的图形被称为方波，如右图所示。

考虑奇次谐波的叠加，令

$$y_1(t) = \sin t，\ y_2(t) = \sin t + \frac{1}{3}\sin 3t，$$

$$y_3(t) = \sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t$$

$$y_4(t) = \sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t + \frac{1}{7}\sin 7t，$$

$$y_5(t) = \sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t + \frac{1}{7}\sin 7t + \frac{1}{9}\sin 9t$$

用如下程序段绘出 $y_1(x)$、$y_2(x)$、$y_3(x)$、$y_4(x)$、$y_5(x)$ 的图形

```
t=0: 0.02: 3.14; x=zeros(size(t));
for   k = 1: 5
```

```
    j = 2*k-1;
    x = x + sin(j*t)/j;
    y (k，：) = x;
end
plot(t，y)
```

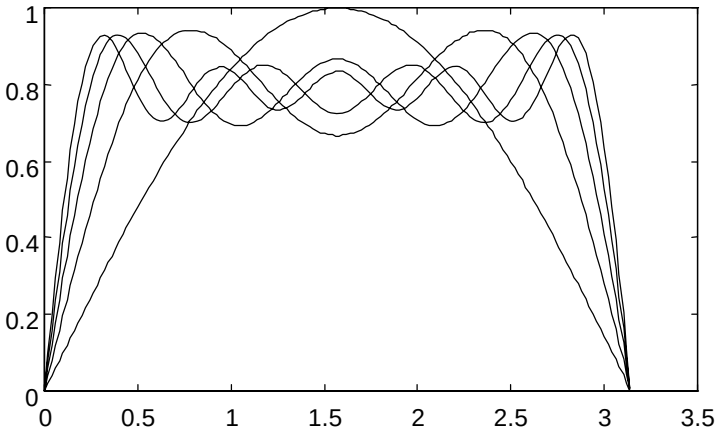


图 3-2 方波是奇次谐波的叠加

从图 3-2 中五条曲线可见，随着奇次谐波叠加的项数增多，图形越来越逼近于方波的图形。

54.如何使用 *while* 循环语句？

由条件控制的循环算法的实现可以分为两种：一种是将条件的判定放于循环体开始之前，满足条件则进入循环，条件不满足时则不进入循环；另一种是将条件的判定放于循环体后面，当条件满足则退出循环，条件不满足则继续循环。 **while** 循环语句适用于由条件控制的循环算法的前一种的实现。反复执行某些操作，直至条件不满足为止。其具体格式为:

```
while <条件>,
    ……
    ……    循环体
    ……
end
```

例如产生裴波拉奇数列的有限项，要求数列中最后一个数不超过 1000。 数列满足的递推公式为：$f_{k+2}= f_k + f_{k+1}$ (k=1,2,……)， f_1 =1，f_2=1。

程序 3.4

```
f=[1 1]；i=1；
while  f(i)+f(i+1)<1000，
       f(i+2)=f(i)+f(i+1)；
       i=i+1；
end
```

MATLAB 中的 *for* 循环语句主要用于实现数控制的循环算法，而 while 循环语句则用于条件控制的循环算法。在数值计算中 *for* 循环语句用得较多，这是因为很多数值计算问题的递推公式是可以用数进行控制的。而 *while* 循环语句由于用条件控制，比较灵活，对于编程者来说如果对问题的条件理解不够，则可能将条件语句设置不好，使得在程序运行时出现死循环（即循环一直进行下去不能自动终止）。这两种循环语句有时可以相互替代，在解决问题时常常将两种语句结合起来应用。

55．如何实现十进制数与二进制数的相互转换

十进制数是人们最熟悉且常用的数，二进制数则是比较陌生的一类数。在用计算机解决实际问题时，常用到二进制数以及二进制数与十进制数的相互转换。如求解“0—1”规划问题，用遗传算法解优化问题等。一个二进制数中每一个位都是“0”或“1”，故可用以“0”或“1”为元素组成的 n 维向量表示二进制数，在 MATLAB 中实现二进制数到十进制数的转换非常容易。例如，用向量

$$x= [\,1 \quad 1 \quad 1\,]$$

表示一个三位二进制数 $(111)_2$，与之对应的十进制数是 7。按公式

$$(111)_2 = 1\times2^2 + 1\times2^1 + 1\times2^0$$

计算，便得十进制数 7。实现这一转换可用 MATLAB 中多项式求值命令 *polyval*(*x*，2)。

再例如，为了将五位二进制数 $(11001)_2$ 转换为十进制数，用命令

```
y=[1 1 0 0 1]；polyval(y,2)
```

得到

```
ans =      25
```

这说明，（11001）$_2$ 对应于十进制数 25。

如果要将一个十进制数转换为二进制数，可以用算法来实现。由

$$25 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$$

$$=2\,(\,2\,(\,2\,(\,2+1\,)+0\,)+0\,)+1$$

知，如果用 2 连续除这个十进制数，直到除数为 0 为止，会依次得余数 1，0，0，1，1。再这 5 个数颠倒秩序，便得到对应的二进制数（11001）$_2$，按这一方法编写下面程序段

```
x=input('input   x=   ');
n=fix(log(x)/log(2))+1;
k=n+1;
while x>0
    k=k-1;
    b(k)=rem(x,2);
    x=fix(x/2);
end
b
```

如果将这段程序写入一个文件名为 ***bio.m*** 的文件，运行这段程序时，在 MATLAB 命令窗口中键入文件名 ***bio*** 并回车，输入十进制数 25，将得到

```
b =     1     1     0     0     1
```

56．如何用 *break* 语句中断循环？

为了使循环算法更灵活，在实现循环算法时常用到控制转移语句，以达到在循环计算时中断循环的目的。 MATLAB 中的转移语句 ***break*** 的使用格式为

if <条件> ***break*** ， ***end***

例 计算无穷级数 $\sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{n!}$ 的近似值。要求误差不超过 10^{-10}。

由于究竟应该计算多少项，在程序开始之前无法确定，故不能用数控制的循环来实现算法。可以考虑用条件控制的循环算法来实现这一问题的计算，在计算过程中当误差达到要求时则跳出循环，并输出数据结果。用 while 语句编程如下

```
an=1;s=1;f=1;n=2;
while n>1
```

```
an=an/n;
if  an<1e-010，break，end
f = -f; s = s + f*an; n=n+1;
end
```

这段程序执行的结果, **s = 0.6321,** 误差界限不超过 **an = 1.1471e-011** (即1.1471×10^{-11})。如果没有转移语句，程序将按照公式求和，并一直进行下去永不休止，形成死循环。

操作者可以考虑用数控制的循环实现，计算上面的无穷级数到 **100** 项。并将数据结果与上面的结果作比较。

57. 怎样求解水手、猴子和椰子问题？

问题如下：有五个水手带了一只猴子来到南太平洋的一个荒岛上，发现那里有一大堆椰子。由于旅途的颠簸，大家都很疲倦，很快就入睡了。第一个水手醒来后，把椰子平分成五堆，并将多余的一个椰子给了猴子，他私藏了一堆后便又去睡了。第二、第三、第四、第五个水手也陆续起来，和第个水手一样，把椰子平分成五堆后，将恰好多的一个也给猴子，并私藏了一堆，再去入睡。天亮以后，大家把余下的椰子重新等分成五堆，每人分一堆，正好余一个再给猴子。试问原先共有几个椰子？

求解这一问题可以用递推算法。首先分析椰子数目的变化规律，设最初的椰子数为 p_0，即第一个水手所处理之前的椰子数，用 p_1、p_2、p_3、p_4、p_5 分别表示五个水手对椰子动了手脚以后剩余的椰子数目，则根据问题有

$$p_{k+1}=\frac{4}{5}(p_k-1),\qquad (k=0,1,2,3,4)$$

再用 x 表示最后每个水手平分得到的椰子数，于是有

$$x=\frac{1}{5}(p_5-1)$$

所以

$$p_5 = 5x+1$$

利用逆向递推的方法，有

$$p_k=\frac{5}{4}p_{k+1}+1,\qquad (k=4,3,2,1,0)$$

有了逆向递推关系式，求解这一问题似乎很简单，但由于椰子数为一正整数，用任意的 x 作为初值递推出的 p_0 数据不一定是合适的。

这里用 *for* 循环语句结合 *break* 语句来寻找合适的 x 和 p_0，对任意的 x 递推计算出 p_0，当计算结果为正整数时，结果正确，否则选取另外的 x 再次重新递推计算，直到计算出的结果 p_0 为正整数为止。程序如下

```
n=input('input n:');
for x=1:n
    p=5*x+1;
    for k=1:5
        p=5*p/4+1;
    end
    if p==fix(p), break, end
end
disp([x, p])
```

运行这段程序后，屏幕出现要求从键盘输入 x 数据的信息 *input n*，输入 1200 后，MATLAB 计算出合适的 x 和 p_0 的值为

1023 15621

为了验证结论的正确性，可从理论上来作一番分析。由于

$$p_0=\frac{5}{4}p_1+1=\frac{5}{4}(\frac{5}{4}(\frac{5}{4}(\frac{5}{4}(\frac{5}{4}(5x+1)+1)+1)+1)+1)+1$$

$$=\frac{5^6}{4^5}x+1+\frac{5}{4}+(\frac{5}{4})^2+(\frac{5}{4})^3+(\frac{5}{4})^4+(\frac{5}{4})^5=\frac{5^6}{4^5}x+\frac{5^6}{4^5}-4$$

所以

$$p_0=\frac{5^6}{4^5}(x+1)-4$$

要使得最初的椰子数 p_0 为整数，必须取 $(x+1)$ 为 $4^5(=1024)$的倍数，一种简单的处理可取 $x=1023$。

58．如何绘出多条抛射曲线以及它们的包络线？

抛射物体的运动可描述为平面上一个动点的轨迹，即抛射曲线，其参数方程为

$$\begin{cases} x = v_0 \cos\alpha \times t \\ y = v_0 \sin\alpha \times t - \dfrac{1}{2} g t^2 \end{cases}$$

其中 ***g*** 是重力加速度，动点初始速度为 v_0，发射角度为 α。

当发射角度在区间 $[0,\ \pi/2]$ 内变化时，不同发射角便形成不同曲线。由

$$v_0 \sin\alpha \times t - \frac{1}{2} g t^2 = 0$$

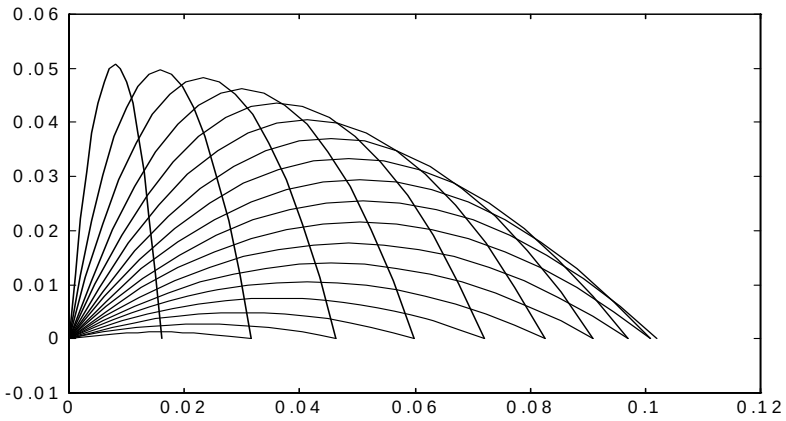
解之，得弹落点所对应的参数值

$$t_0 = \frac{2v_0 \sin\alpha}{g}$$

所以，对发射角 α，参数 ***t*** 的变化范围为**[0，t_0]**。为了简化问题，取 $v_0 = 1$。下面程序段可绘制曲线簇中的 **n-2** 条曲线

```
n=input('input  n：');                输入数据 n，确定所绘曲线簇曲线数
alpha=(2：n-1)*pi/(2*n);              确定不同曲线所对应的发射角
for  k=1：n-2                         开始计算 n-2 条曲线上的离散点数据
  a=alpha(k);                         选取α的值
  v1=cos(a)；v2=sin(a);               计算初始速度分量
  t0=v2/4.9；
  t=(0：16)*t0/16；                   确定参数值
  x(k，：)=v1*t；y(k，：)=v2*t-4.9*t.^2；  确定曲线上离散点坐标数据
end
plot(x'，y')                          同时绘出曲线簇中 n-2 条曲线
```

运行上面程序，输入 ***n*** = **20** 则可以绘出图 **3-3** 中的 **18** 条曲线。



不同发射角所形成的抛射线构成一曲线簇，如果存在一条曲线 ***L***，曲线簇中每一曲线都与 ***L*** 相切，则称 ***L*** 为该曲线簇的包络。对于参数方程，曲线族的包络曲线由

$$x = x(t,\alpha), \quad y = y(t,\alpha), \quad \frac{\partial x}{\partial t}\frac{\partial y}{\partial \alpha} - \frac{\partial y}{\partial t}\frac{\partial x}{\partial \alpha} = 0$$

消去参变量 λ 而得到。在上面抛射线族的包络曲线中

$$\frac{\partial x}{\partial t} = v_0 \cos\alpha\,, \quad \frac{\partial x}{\partial \alpha} = -v_0 \sin\alpha \cdot t$$

$$\frac{\partial y}{\partial t} = v_0 \sin\alpha - gt\,, \quad \frac{\partial y}{\partial \alpha} = v_0 \cos\alpha \cdot t$$

由

$$\frac{\partial x}{\partial t}\frac{\partial y}{\partial \alpha} - \frac{\partial y}{\partial t}\frac{\partial x}{\partial \alpha} = 0$$

即

$-\sin\alpha \cdot t[\sin\alpha - gt] - \cos\alpha \cdot \cos\alpha \cdot t = 0$。求解得

$$\sin\alpha = \frac{1}{gt}$$

代入曲线族的参数方程，便得包络曲线的参数方程为

图 **3-3** 不同发射角形成的抛射线簇

$$\begin{cases} x = \sqrt{t^2 - \dfrac{1}{g^2}} \\ y = \dfrac{1}{g} - \dfrac{1}{2}gt^2 \end{cases} \quad \frac{1}{g} \le t \le \frac{\sqrt{2}}{g}$$

下面程序段将绘制出曲线簇的包络曲线（又称为安全抛物线）。

```
g=9.8;
t=1/g:.001:sqrt(2)/g;
x=sqrt(t.^2-1/g^2);
y=1/g-.5*g*t.^2;
plot(x,y)
```

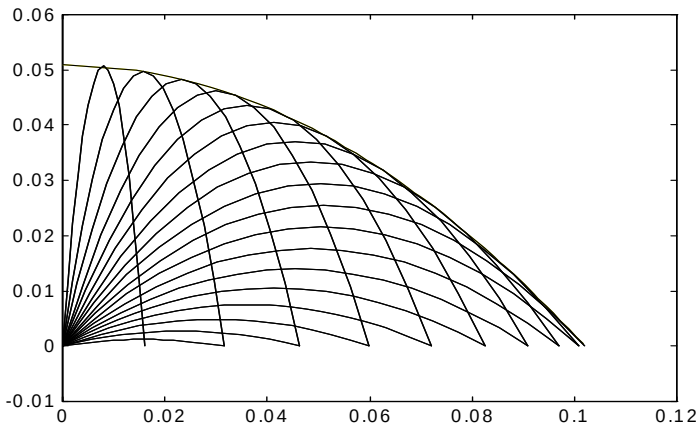


图 3-4　曲线簇及其包络

59．如何输入大型稀疏矩阵的数据？

当一个矩阵中只含一部分非零元素，而其余均为“0”元素时，我们称这一类矩阵为稀疏矩阵。在实际问题中，相当一部分的线性方程组的系数矩阵是大型稀疏矩阵，而且非零元素在矩阵中的位置表现得很有规律。为了提高工作效率，MATLAB 提供了稀疏矩阵的创建命令和稀疏矩阵的存储方式。

创建一个稀疏矩阵常用命令为 ***sparse***。使用格式如下

***A* = *sparse*(I，J，S，m，n，nzmax)**

其中，S 是稀疏矩阵中所有非零元素组成的列向量；I，J 分别为非零元素的行

下标和列下标构成的列向量；**m**，**n** 表明 ***A*** 是 **m×n** 阶矩阵（可省略）；**nzmax** 用于指定 ***A*** 中非零元素所用存储空间大小（可省略）。最简单的使用方式是只输入非零元的数据以及各非零元的行指标和列指标。例如，创建一个只有三个非零元的 **4×5** 阶矩阵

$$A=\begin{bmatrix}6&0&0&0&0\\0&0&7&0&0\\0&0&0&0&0\\0&0&0&0&8\end{bmatrix}$$

可用下面命令

***i*=[1 2 4]；*j*=[1 3 5]；*s* = [6 7 8]；**

***A* = *sparse*(*i*，*j*，*s*)**

MATLAB 执行后，将显示出

***A* =**

(1,1) 6

(2,3) 7

(4,5) 8

这是特殊的稀疏矩阵存储方式，它的特点是所占内存少，运算速度快。如果想得到矩阵的全元素存储方式，可用下面命令

***B* = *full*(*A*)**

计算机运行后，将显示出

***B* =**

6 0 0 0 0

0 0 7 0 0

0 0 0 0 0

0 0 0 0 8

另一个创建稀疏矩阵命令是 ***spdiags***。它主要用于创建非零元素位于矩阵的对角线上的情况。例如，创建一个三对角的 **10×10** 矩阵

$$A=\begin{bmatrix}4&1&&&\\1&4&1&&\\&\ddots&\ddots&\ddots&\\&&1&4&1\\&&&1&4\end{bmatrix}_{10\times10}$$

可用下面命令

e = *ones* (10，1);

A = *spdiags* ([*e*，4**e*，*e*]，[-1，0，1]，10，10);

B = *full* (*A*)

这里，*A* 是稀疏矩阵存储的方式，*B* 是全元素存储的方式

B =

4	1	0	0	0	0	0	0	0	0
1	4	1	0	0	0	0	0	0	0
0	1	4	1	0	0	0	0	0	0
0	0	1	4	1	0	0	0	0	0
0	0	0	1	4	1	0	0	0	0
0	0	0	0	1	4	1	0	0	0
0	0	0	0	0	1	4	1	0	0
0	0	0	0	0	0	1	4	1	0
0	0	0	0	0	0	0	1	4	1
0	0	0	0	0	0	0	0	1	4

60．如何绘制出图论中的图形?

绘制一个图论中的图形用命令 *gplot*。在图论中，一个图 *G* 是指一个点的集合及其连线的集合所组成的图形。点的集合中，点的编号为 1，2，……，*n*；点之间的连结情况可以用一个称为邻接矩阵的 *n* 阶矩阵 *A* 来描述，*A* 中的元素 a_{ij} 不为零时表示点 *i* 和点 *j* 是相连的。而点的集合可以用一个坐标数组 xy 来描述，这个数组的第 *i* 行两个元素表明第 *i* 个点的坐标，即 xy(*i*，：)=[x_i ，y_i] ， (*i* = 1，2，……，*n*)。绘制图论图形命令的使用格式为

gplot (*A*，xy)

这里，*A* 是邻接矩阵，xy 则是点的坐标数组。

例如，为了绘出一个邻接矩阵为

$$A=\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

的图，图中四个点的坐标分别为：P_1(0，1)、P_2(1，1)、P_3(0，0)、P_4(1，0)。用

如下的程序可绘出图 3-5 中图形。

```
A = [0  1  1  0; 1  0  0  1; 1  0  0  1; 0  1  1  0];
xy =[0  1; 1  1; 0  0; 1  0];
gplot(A, xy);  axis('off')
```

如果加上下面命令可得有点编号的图 3-6 中的图形。

```
x=xy(:, 1); y=xy(:, 2);
for  i = 1: 4, text(x(i), y(i), int2str(i)); end
```

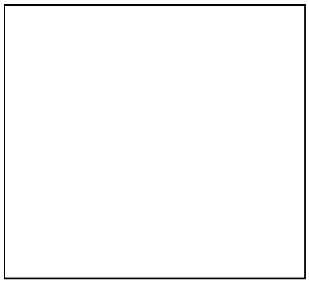


图 3-5

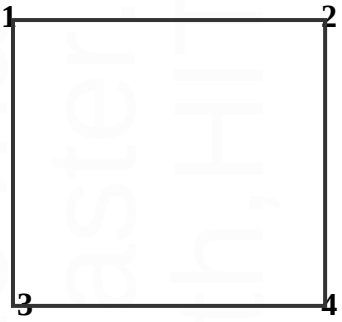


图 3-6

61．如何求多项式拟合函数?

如果已经获得如下的一组实验数据

X	x_1	x_2	……	x_m
Y	y_1	y_2	……	y_m

可以用多项式曲线拟合函数命令 *polyfit*，来获得一个反映数据规律的多项式函数

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$$

这一命令的使用格式为

polyfit(*x*，*y*，*n*)

其中，*n* 是所求拟合多项式的阶数，可以人为确定。拟合多项式在拟合点处的函数值与被拟合的数据值之间满足 $p(x_i) \approx y_i$ ，这是最小二乘意义下的多项式拟合。

例如，炼钢厂出钢时所用的盛钢水的钢包，在使用过程中由于钢液及炉渣对包衬耐火材料的侵蚀，使其容积不断增大，经过试验，钢包的容积与相应的使用次数的数据如下表

表 3-2

使用次数 *x*	容积 *y*	使用次数 *x*	容积 *y*
2	106.42	11	110.59
3	108.26	14	110.60
4	109.58	15	110.90
5	109.50	16	110.76
7	110.00	18	111.00
8	109.93	19	111.20
10	110.49		

下面程序用三次多项式拟合上面表中数据，并画出拟合曲线及散点图。

```
x=[2 3 4 5 7 8 10 11 14 15 16 18 19];
y=[106.42 108.26 109.58 109.5 110 109.93 110.49 110.59 110.6 110.9 110.76 111
111.2];
v=polyfit(x，y，3);
t=1：0.5：19；u=polyval(v，t);
plot(t，u，x，y，’*’)
```

程序运行后，MATLAB将绘出离散点及拟合曲线如图3-7。

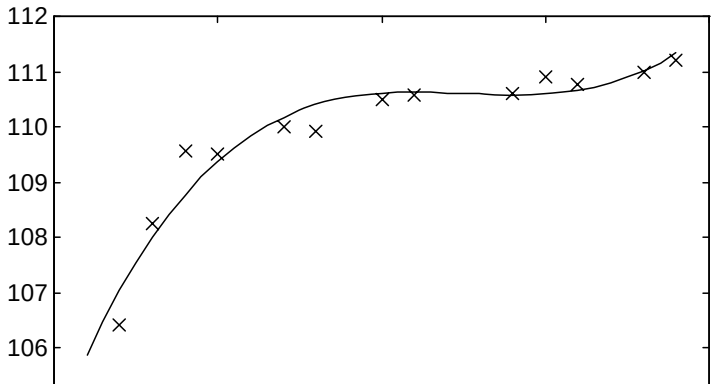


图 3-7　离散点及拟合曲线

而 *polyfit* 的执行结果为 *v* 的值（a_0、a_1、a_2、a_3）

0.0033　-0.1224　1.5113　104.4824

62．如何使用多项式插值命令

MATLAB 中常用的数据插值命令列表如下

表 3-3

interp1	一元函数插值
interp2	二元函数插值
interpft	用快速付氏变换的一元插值
griddata	二元函数离散点插值

一元函数插值处理这样一类的问题：如果已知函数 $y = f(x)$ 在点 x_1，x_2，……，x_n 处的函数值，即已知表

X	x_1	x_2	……	x_n
Y	y_1	y_2	……	y_n

中的数据，现在要通过满足条件 $P(x_j) = y_j$ （$j = 1$，2，…，n）的插值函数 $P(x)$ 计算出自变量在离散点 t_1，t_2，……，t_m（$m>n$）处的函数值 。

如果记 $X=[x_1 \quad x_2 \quad \cdots \quad x_n]$，$Y=[y_1 \quad y_2 \quad \cdots \quad y_n]$，$XI=[t_1 \quad t_2 \quad \cdots \quad t_m]$，则一元函数插值命令

$$YI = interp1(X，Y，XI ，’ method ’)$$

将得出向量 YI，这一向量的元素就是对应于 XI 中每一元素的插值函数值

$$u_1 = P(t_1)，u_2 = P(t_2)，……，u_m = P(t_m)。$$

在实现这一计算的过程中，构造几种不同插值函数的依据是向量 X 和 Y。如果省略了参数 *method*，将由分段线性插值的方法计算出数据结果。这一参数的选择有下面三个

表 3-4

'linear'	分段线性插值方法
'spline'	三次样条插值方法
'cubic'	分段三次插值方法

无能哪一 方法都要求 X 的元素按单调排列；而分段三次插值方法要求 X 的元素按等距离分布。例如，取正弦曲线上 11 个点的自变量和函数值点，再选取

41 个自变量点，计算确定插值函数的值。分别用三种方法计算，程序如下

```
x = 0:10; y = sin(x);
xi = 0:.25:10;
y1 = interp1(x,y,xi);
y2=interp1(x,y,xi,'cubic');
y3=interp1(x,y,xi,'spline');
plot(x,y,'o',xi,y1,xi,y2,'-.',xi,y3)
```

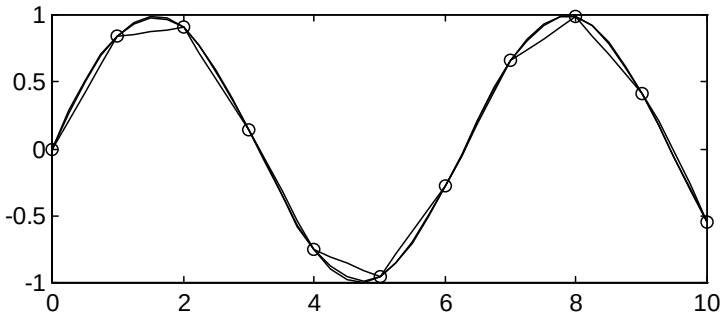


图 3-8　三种插值方法比较

从图 3-8 可以看出，样条插值和三次分段插值效果较好，而分段线性插值则较差。

63．如何对数据作样条插值处理?

实现数据的三次样条插值用命令 ***spline***。这一命令的功能是根据函数表

X	x_1	x_2	……	x_n
Y	y_1	y_2	……	y_n

中的数据，以及新的自变量离散数据 t_1，t_2，……，t_m ($m>n$)。用三次样条插值函数计算出对应的函数值 $u_1 = f(t_1)$，$u_2 = f(t_2)$，……，$u_m = f(t_m)$。

如果记 $T=[t_1, t_2, \cdots\cdots, t_m]$，则样条插值命令使用格式如下

$$U = spline(X, Y, T)$$

这里，U 是经计算得出的对应于 T 中自变量数据的函数值。

例如，下面程序段是先取自变量的离散数据 $x = [0 \quad 1 \quad 2 \quad \cdots \quad 10]$，并计算出对应的正弦函数值；然后再取自变量的数据 $xi = [0 \quad 0.25 \quad 0.5 \quad \cdots\cdots \quad 9.75 \quad 10]$，最后用命令 ***spline*** 计算出新的自变量离散数据点处的函数值并根据数据绘出图形。

$x = 0:10; \ y = sin(x);$

***xi* = 0：0.25：10；**

***yi* = *spline*(*x*，*y*，*xi*)；**

***plot*(*x*，*y*，’o’，*xi*，*yi*)**

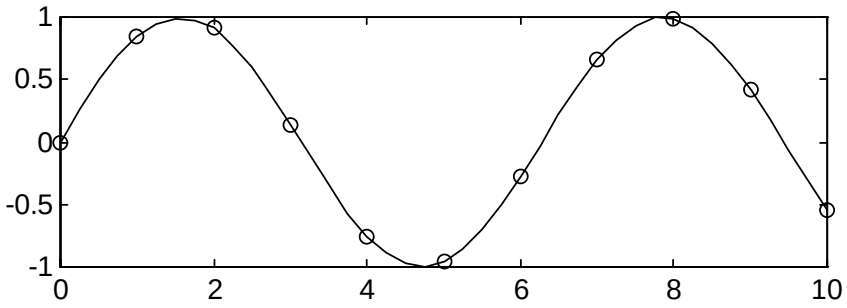


图 3-9　三次样条插值曲线

由图 3-9 可见，尽管只取了正弦函数 11 个点处的数据值，但三次样条插值计算出的数据所绘图形却与正弦曲线非常相似。

64．如何实现数据滤波？

在信号处理中，对于带有噪声的数据需要进行滤波。MATLAB 提供了一个滤波器用以实现数据滤波。设被滤波的数据向量为

$$X = [\, x_1 \quad x_2 \quad \cdots\cdots \quad x_N \,]$$

滤波器分母、分子向量为

$$A = [a_1 \quad a_2 \quad \cdots\cdots \quad a_{na}\,], \quad B = [b_1 \quad b_2 \quad \cdots\cdots \quad b_{nb}\,]$$

数据滤波命令的使用格式为

***Y* = *filter* (*B*，*A*，*X*)**

下面的程序是做一个滤波实验，带噪声的输入信号经滤波后得输出信号。

```
t=linspace(0,10,100);                    %定义时间轴
s=sin(2*pi/5*t);                         %原始信号
noise=0.2*rand(size(t));                 %定义噪声
x=s+noise;                               %带噪声的输入信号
y=zeros(size(x));
A=[1 -0.9];
B=[0.05 0.06];
y=filter(B,A,x);                         %使用滤波器
```

plot(t,x,'b',t,y,'r') **%作图**

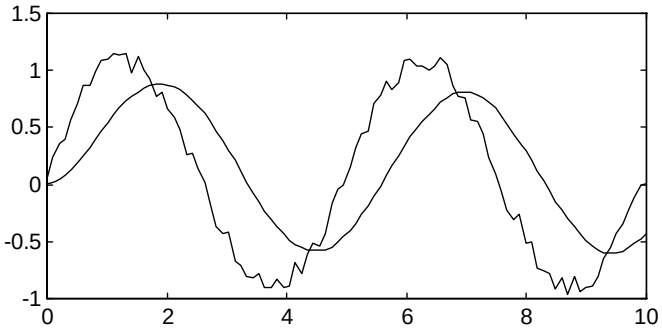


图 3-10 滤波实验

65. 如何用蒙特卡罗法计算定积分和重积分?

设函数 ***f(x)*** 的值介于 **0** 和 **1** 之间，定积分 $\int_0^1 f(x)dx$ 的几何意义是曲边梯形面积，此曲边梯形位于正方形

$$D=\{(x,y)\,|\,0\le x\le 1,\ 0\le y\le 1\}$$

的内部。显然，***D*** 的面积为 **1**， 用随机投点的方法在区域 **D** 内产生充分多的均匀分布的点(至少 **2000** 个点)。设随机点总数为 ***N***，这些点随机地落入 ***D*** 中任何一处，于是，落入曲边梯形内点的数目 ***m*** 与 ***N*** 之比反映了曲边梯形面积与正方形 ***D*** 的面积之比。由此计算曲边梯形面积近似值。

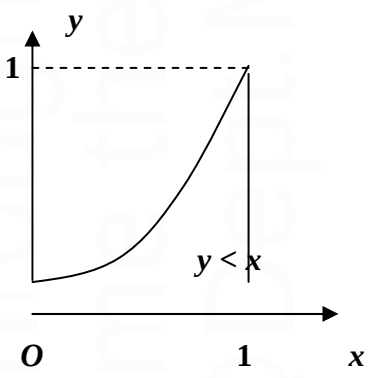


图 3-11

$$\int_0^1 f(x)dx\approx\frac{m}{N}\times(D\text{的面积})=\frac{m}{N}$$

这种计算定积分的方法被称为蒙特卡罗方法。例如，为了计算定积分 $\int_0^1 x^2dx$ ，将如下程序

```
x=rand(10000,2);
s=sum(x(:,1).^2-x(:,2)>=0)/10000;
```

重复计算 **6** 次，得结果

表 3-5

第 k 次	1	2	3	4	5	6
数据	0.3396	0.3335	0.3363	0.3339	0.3416	0.3309

由表中数据可知，每次计算的结果都接近于定积分准确值 1/3。 由于随机因素的影响，计算机模拟的结果可能不一样，但是各次计算所得数据总是在准确值附近摆动。

考虑重积分的情形，例如，图 3-12 中的锥形冰淇淋图形。它的体积是由锥面与球面所围空间区域的大小。可通过二重积分

$$\iint_D [(1+\sqrt{1-x^2-y^2})-\sqrt{x^2+y^2}]dxdy$$

求得。其中 $D=\{(x,y)\,|\,x^2+y^2\le 1\}$。所求的锥形体可表示为

$$G=\{(x,y,z)\,|\,x^2+y^2\le z(2-z),\ x^2+y^2\le z^2\}$$

它位于长方体 $\Omega=\{(x,y,z)\,|\,-1\le x\le 1,-1\le y\le 1,0\le z\le 2\}$ 的内部，所以可用蒙特卡罗方法计算 G 的体积，其思路是在区域 Ω 内产生 N 个随机点，由于这 N 个点是均匀分布的，故落入区域 G 内的点的数目与总的随机点的数目 N 之比反映了空间区域 G 与空间区域Ω的体积之比。若在长方体内部随机投点 N = 10000 个，然后统计锥形体内的随机点数 m，则可按公式

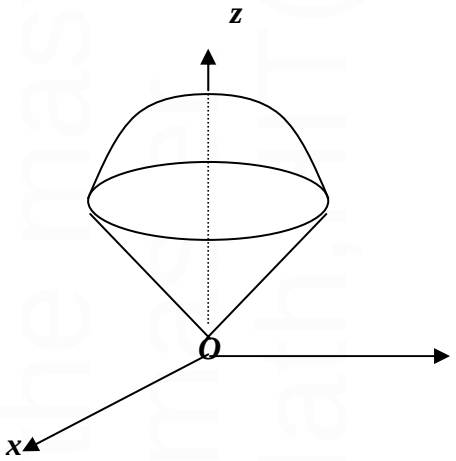


图 3-12 锥形冰淇淋图形

$$V\approx 8\times\frac{m}{10000}$$

计算出重积分的近似值。所用程序如下

```
for k=1:10
r=rand(10000,3);x=2*r(:,1)-1;y=2*r(:,2)-1;z=2*r(:,3);
f1=x.^2+y.^2;
p(k)=8*sum(f1-z.*z<=0&f1-z.*(2-z)<=0)/10000;
end
p
```

程序运行后，十次计算数值为

p = 3.2032 3.1736 3.1040 3.1368 3.2024 3.1016 3.1712

3.1304 3.1928 3.1840

它们都很接近这一积分的准确值 π.

66. 如何用赛德尔迭代法求解线性方程组？

线性方程组的一般形式为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

或写成紧凑格式

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad (\ i = 1,\ 2,\ \cdots\cdots,\ n\ )$$

赛德尔迭代法计算格式为

$$x_i^{(k+1)} = \frac{1}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}], \quad (\ i = 1,\ 2,\ \cdots,\ n\ )$$

其中，k 是迭代次数。例如，用赛德尔迭代法求解下列方程组

$$\begin{cases} 5x_1 - x_2 - x_3 - x_4 = -4, \\ -x_1 + 10x_2 - x_3 - x_4 = 12, \\ -x_1 - x_2 + 5x_3 - x_4 = 8, \\ -x_1 - x_2 - x_3 + 10x_4 = 34 \end{cases}$$

该方程组的准确解为 $x_1 = 1$，$x_2 = 2$，$x_3 = 3$，$x_4 = 4$。

赛德尔迭代法求解方程组 MATLAB 程序如下

```
a=[5 -1 -1 -1;-1 10 -1 -1;-1 -1 5 -1;-1 -1 -1 10];
b=[-4;12;8;34];x=[0;0;0;0];
for k=1:8
    eorr=0;
    for i=1:4
```

```
            s=x(i);x(i)=0;
            x(i)=(b(i)-a(i,:)*x)/a(i,i);
            eorr=max(abs(s-x(i)),eorr);
        end
    x',pause，eorr
    end
```

程序运行后，将显示出每一次迭代计算的解的近似向量和误差估计值

表 3-6

第k次	x_1	x_2	x_3	x_4
6	1.0004546181818 0	1.9995243397529 0	3.0005562619742 4	3.9998479579789 6
7	0.9998919306931 0	2.0001306701269 8	2.9998576817169 2	4.0000160423086 5
8	1.0000226684579 9	1.9999614330726 3	3.0000284981780 4	3.9999983035033 1

误差估计为

$$\left\| X^{(7)} - X^{(8)} \right\|_{\infty} = 1.708164611193830 \times 10^{-4}$$

这说明随着迭代次数的增加，迭代向量将不断逼近方程组的准确解。

67．如何计算二项分布随机变量的概率值？

二项分布是常用的离散随机变量的分布类型。当随机变量的取值范围很大时，人们常用泊松分布的公式来做近似计算。

例如，已知随机变量 $X \sim B(6, 0.4)$。由二项分布随机变量的分布律知

$$P\{X = k\} = C_6^k (0.4)^k (0.6)^{6-k}, \quad (k = 0, 1, 2, 3, 4, 5, 6)$$

为了计算出七个概率值，需要对组合数

$$C_6^k = \frac{6!}{(6-k)!\,k!} = \frac{6 \times 5 \times \cdots \times (6-k+1)}{1 \times 2 \times \cdots \times k}$$

进行计算，组合数中的分子与分母是有限个数的连乘，在 MATLAB 中可以用命令 *prod* 来实现。下面程序可计算出随机变量 $X \sim B(n, p)$ 的分布律中的不同概率值

```
p=input('input   p=   ');
```

```
n=input('input   n=   ');
q=1-p;
for k=1:n
    m=prod(n-k+1:n);
    mm=prod(1:k);
    b(k+1)=p^k*q^(n-k)*m/mm;
end
b(1)=q^n
```

对于 $X \sim B(6, 0.4)$，由于 $n = 6$，$p = 0.4$。当程序运行时，屏幕出现要求输入参数的信息“input p= ”和 “input n= 6 ”，此时，分别输入：0.4 和 6 ，MATLAB 将计算出 7 个概率值数据如下

b = 0.0467 0.1866 0.3110 0.2765 0.1382 0.0369 0.0041

考虑数字通讯中误码概率的计算。某一条通讯线路以第秒 512×10^3 个 0 或 1 的速度传输信息，已知误码率为 $p = 10^{-7}$，求 10 秒钟内出现一个误码的概率。由二项分布的分布律知，概率计算公式为

$$P = C_{512\times10^4}^{1} \times 10^{-7} \times (1-10^{-7})^{512\times10^4-1}$$

对于这样的计算，可以在 MATLAB 的命令窗口中键入命令

```
(5120000/10000000)*(1-0.0000001)^(5120000-1)
```

可得概率值 ans = 0.3068。

如果用泊松分布做近似计算，则有

$$P(X = k) \approx \frac{\lambda^k}{k!} e^{-\lambda}$$

其中，$\lambda = n\times p$。由于 $n = 5120000$，$p = 10^{-7}$，所以用下列命令

```
lamda =5120000/10000000
lamda*exp(-lamda)
```

计算出概率值为 ans = 0.3068。这一结论与前面是一致的。

68．如何产生两点分布的随机数?

MATLAB 所提供的随机数发生器，只能产生均匀分布随机数和标准正态分布的随机数两种，其它分布的随机数的产生可以借助于均匀分布随机数根据分布律构造产生出来。

两点分布随机数描述了一个随机事件要么发生要么不发生两种可能结果。

设某一事件 ***A*** 发生的概率为 ***p***，则事件 ***A*** 不发生的概率为 **(1—*p*)**。由于 ***p*** 是介于 **0，1** 之间的数，所以 ***p*** 将 **[0，1]** 区间分为两部分，即 **（0，*p*]** 和（***p***，**1**）。当一个均匀分布的随机数落入第一个区间时，可以认为事件 ***A*** 发生；反之则事件 ***A*** 不发生。所以，由 **MATLAB** 产生一个均匀分布随机数 ***x*** ，再由此构成一个两点分布的随机数 ***y***

$$y=\begin{cases}1, & x\in(0,\ p]\\ 0, & x\in(p,\ 1)\end{cases}\quad \text{或}\quad y=\begin{cases}1, & x\in(0,\ p)\\ 0, & x\in[p,\ 1)\end{cases}$$

例如，**100** 个产品中有 **10** 个次品，则可以认为随机抽取一个产品是次品的概率 ***p*=10/100**。为了模拟抽查 **8** 个产品中次品数目的实际情况，用如下程序段

```
x=rand(1,8)
y=1-fix(x+.9)
```

上面程序中，第一行产生 **8** 个均匀分布的随机数向量 **x**，第二行中 **fix(x+0.9)** 将 **x** 中大于等于 **0.1** 元素变为 **1**，将小于 **0.1** 的元素变为 **0**，所以第二行命令最后将小于 **0.1** 的元素变为 **1**，而其余的元素变为 **0**。下面是模拟三次的结果

表 **3-7**

x	**0.7012**	**0.9103**	**0.7622**	**0.2625**	**0.0475**	**0.7361**	**0.3282**	**0.6326**
y	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
x	**0.7564**	**0.9910**	**0.3653**	**0.2470**	**0.9826**	**0.7227**	**0.7534**	**0.6515**
y	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
x	**0.0727**	**0.6316**	**0.8847**	**0.2727**	**0.4364**	**0.7665**	**0.4777**	**0.2378**
y	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

69．如何用计算机模拟 Galton 板试验？

六级 Galton 板试验如图 3-13 所示。当小球从顶部向落下时，遇到第一层隔板，此时小球向左落下或向右落下的可能性各占一半，即小球往左或往右的概率各为 0.5；当小球继续下落遇到第二层隔板时，小球往左或往右的概率仍然是各占二分之一；以后每一层情况都是如此，最后到了第六层底部，小球将落入底部七个槽中的其中一个。但是小球究竟落入哪一个小槽内的概率是不一样的。如果将这七个槽编号为 0，1，2，3，4，5，6，将小球最终落入某一确定槽的编号记为随机变量 X，则 X 的取值为：

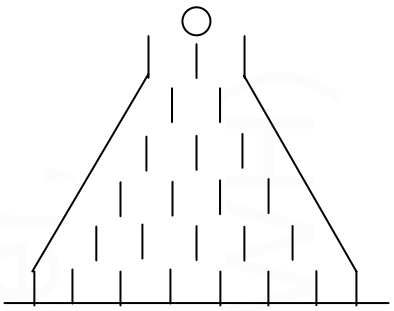


图 3-13　Galton 板试验

0，1，2，3，4，5，6

这是一个二项分布的古典概率模型。

考虑小球在第 j 层落下时，可能往左，也可能往右，用 X_j 表示，则 X_j (j=1，2，…，6)服从两点概率分布，当 X_j 取值为 0 时，表示小球向左落下；当 X_j 取值为 1 时，表示小球向右落下。其分布律为

X_j	0	1
P	0.5	0.5

令

$$X=X_1+X_2+X_3+X_4+X_5+X_6$$

则有

$$X=\sum_{j=1}^{6}X_j \sim B(6，0.5)$$

根据二项分布随机变量的分布律

$$P\{X=k\}=C_6^k(0.5)^k(0.5)^{6-k}，\quad （k = 0，1，2，3，4，5，6）$$

将分布律写成表格形式，需要计算上面的七个概率值。在计算过程中，涉及到组合数

$$C_6^k=\frac{6!}{(6-k)!\,k!}=\frac{6\times5\times\cdots\times(k+1)}{1\times2\times\cdots\times(6-k)}$$

在组合数的计算中，要用到有限个数的连乘，在 MATLAB 中计算连乘可以用指令 prod 来实现。下面程序可计算出上面二项分布的 7 个概率值

```
t=.5^6;
for k=1:6
m=prod(k+1:6);n=prod(1:6-k);
```

```
p(k+1)=t*m/n;
end
p(1)=t
```

在 MATLAB 环境中运行上面的程序，计算机可计算出 7 个概率值如下：

表 3-8

X	0	1	2	3	4	5	6
P	0.0156	0.0938	0.2344	0.3125	0.2344	0.0938	0.0156

或者将概率写做分数形式

表 3-9

X	0	1	2	3	4	5	6
P	1/64	3/32	15/64	5/16	15/64	3/32	1/64

考虑用计算机模拟小球落下的过程，在每一层，当小球往左或往右落下时用计算机模拟产生一个二进制数“0”或“1”。当小球落入最底层时，已经有了六个二进制数，它所落入的小槽的号码应该是将这六个二进制数按十进制数加法法则相加的结果。将这一小球下落的模拟过程重复 1000 次，可以统计出小球落入各小槽内的次数（即频数）。也可以画出七个频数数据的直方图（图 3-14）。

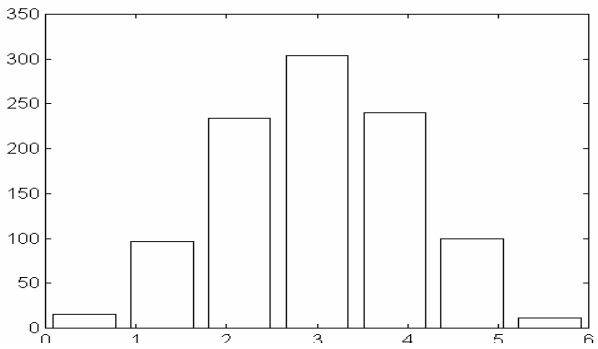


图 3-14　频数直方图

```
p=rand(6,1000);
q=sum(fix(p+.5));
hist(q,7),pause
s=[0 0 0 0 0 0 0];q=q+1;
for k=1:1000
    j=q(k);s(j)=s(j)+1;
end
s
```

程序运行结果为

```
15   96   234   304   240   100    11
```

这说明，在做 1000 次实验后，小球落入各个小槽内的频数分别为上面的 7 个数据

表 3-10

X	0	1	2	3	4	5	6
F	15	96	234	304	240	100	11

由表 2-7 可以计算出小球落入各小槽内的频率如下

0.0150　　0.0960　　0.2340　　0.3040　　0.2400　　0.1000　　0.0110

70．如何用计算机模拟追赶曲线？

狼追兔子问题是欧洲文艺复兴时代著名人物 达·芬奇提出的一个有趣的问题。当一只兔子正在它的洞穴南面60码处觅食时，一只饿狼出现在兔子正东的100码处.兔子急忙奔向自己的洞穴，狼立即以快于兔子一倍的速度紧追兔子不放.问狼是否会追赶上兔子？

设兔子所在位置为动点 Q，狼所在位置为动点 P。在时刻 t_k ，两个动点的坐标分别为：$Q(u_k, v_k)$，　$P(x_k, y_k)$，动点 P 的轨迹就是追赶曲线。在 t_k 时刻到 t_{k+1} 时刻这个时段，P 点的运动方向可以用单位向量描述：

$$\vec{e} = \frac{1}{\sqrt{x_k{}^2 + (t_k - y_k)^2}} \begin{pmatrix} -x_k \\ t_k - y_k \end{pmatrix}$$

显然，$u_k = 0$，$v_k = t_k$ 。根据题设在初始时刻兔子和狼的位置分别为

$Q(0，0)$　，$P(100，0)$

初始时刻的狼、兔距离为 100 码，我们不妨规定当狼、兔距离小于 0.5 码时，兔子被狼追上，结束追赶。下面 MATLAB 程序可计算并绘制追赶曲线（图 3-15）。

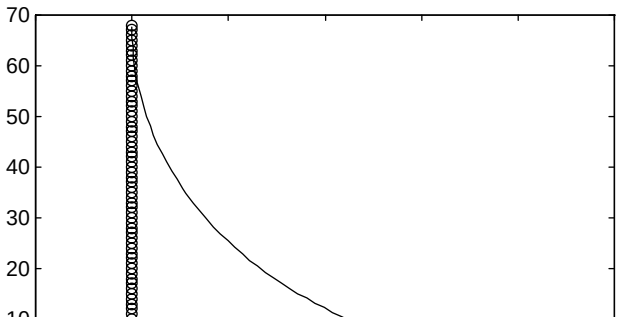
```
x(1)=100;y(1)=0;u(1)=0;v(1)=0;
t=1;d=100;e=[-1 0];
while d>0.5

x(t+1)=x(t)+2*e(1);

y(t+1)=y(t)+2*e(2);

t=t+1;u(t)=0;v(t)=t;
   e=[-x(t) t-y(t)];

d=sqrt(e(1)^2+e(2)^
2);
   e=e/d;
end
```



```
plot(u,v,'o',x,y)
```

图 3-15　追赶曲线

71. 如何用贪婪算法求背包问题的近似解

讨论如下问题，有旅行者要从 n 种物品中选取不超过 b 公斤的物品放入背包随身携带，要求总价值最大。已知第 j 种物品的重量为 a_j，价值为 c_j（j = 1，2，……，n）。这就是著名的背包问题。为了确定选入背包的物品，首先定义变量：$x_1, x_2, \cdots, x_n$，当选取第 j 种物品放入背包中时取 x_j=1，否则取 x_j=0。于是所有选入背包的物品总价值为：$c_1x_1+c_2x_2+ \cdots +c_nx_n$，总的重量为：$a_1x_1 + a_2x_2+ \cdots+a_nx_n$。背包问题的数学模型可以描述为如下的0—1规划问题

$$\max \quad z = c_1x_1+c_2x_2+ \cdots +c_nx_n$$

约束条件：　$a_1x_1 + a_2x_2+ \cdots+a_nx_n \leqslant b$

$$x_j=1 \text{ 或 } x_j=0 \quad (j = 1，2，\cdots\cdots，n)$$

贪婪算法是求解这一类问题的简便算法。具体方案是首先计算出所有物品的价值密度：

$$p_1 = c_1 / a_1，\quad p_2 = c_2 / a_2，\quad \cdots\cdots，\quad p_n = c_n / a_n$$

然后，将价值密度按由大到小的次序排列为：$p_{k_1} \geq p_{k_2} \geq \cdots \geq p_{k_n}$。

选取第 k_1 件物品，判断背包是否会超载。如果不超载，则将其放入背包，并选取第 k_2件物品再判断；如果第 k_1 件物品超载，则放弃第 k_1 件选取第 k_2 件物品，重复以上的操作，直到背包不能放入余下的任何一件物品为止。最后输出放入背包的所有物品的总重量、总价值以及各物品的编号。

例如，有 8 件物品它们的价值分别为：10、20、30、32、40、45、50、60(元)；而重量分别为：1、10、20、22、30、40、45、55(公斤)；背包最大载重量为：110(公斤)。为了选择装入背包的物品，用贪婪算法求解，程序如下

```
c=[10 20 30 32 40 50 55 60];          %输入物品价值数据
a=[1 10 20 22 30 40 45 55];           %输入物品重量数据
b=110;                                %输入背包最大载重量
p=c./a; n=length(c);                  %计算各物品的价值密度
[p,l]=sort(p);                        %将各物品按价值密度大小排序
```

```
    s0=0；p0=0；                          %选价值密度最大的物品并记录重量和价值
    for i=1：n
        k=l(n-i+1);s=s0+a(k);p=p0+c(k);    %按价值密度大小选物品并累计重量和价值
        if s<=b                           %判断当前所选物品放入背包后是否超载
            x(k)=1;s0=s;p0=p；
        else
            x(k)=0;
        end
    end
    x
    disp([p0,s0])                         %显示选中物品编号及总重量,总价值
```

程序运行后显示数据结果为

x = 1 1 1 1 1 0 0 0

p0= 132； s0= 83.

所以应选取第一、第二、第三、第四、第五个物品放入背包，这一方案所选物品的总价值：132 (元)，总重量：83 (公斤)。由于背包可以载重 110 公斤，还剩余 27 公斤可以装物品但却没有合适的物品放入。显然这不是总体最优，只是局部最优。

72. 如何知道一段程序运行多长时间？

MATLAB 提供了用于记录解决问题过程所用时间的计时器。使用方法是在程序第一行前用命令 ***tic*** 启动计时器，在程序的最后一行后面用命令 ***toc*** 关闭计时器。如此处理后，当这段程序运行结束时屏幕将自动显示出所花费的时间。

例如用穷举算法求解背包问题，可以求出这一问题的最优解，但是所用的时间将随问题中物品个数的增多而迅速增大。程序如下

```
tic                                   %启动计时器
c=[10 20 30 32 40 50 55 60];
a=[1 10 20 22 30 40 45 55];
m=110;
n=length(a);k=0;x=[];
for j=1:2^n-1
```

```
        i=n+1;s=j;
        while s>0
                i=i-1;
                t(i)=rem(s,2);
                s=fix(s/2);
        end
        if a*t'<=m,k=k+1;x(k,:)=t;end
    end
    p=x*c';[p0,i]=max(p);
    s=x(i,:)
    q0=a*s';
    disp([p0,q0])
    toc                                          %关闭计时器，并显示所用时间
```

计算机运行程序后，将显示问题答案

```
s =     1     1     0     1     1     0     1     0
        157   108
```

以及程序运行所用时间

```
elapsed_time =     0.2200
```

结果说明，运行这段程序用了 0.22 秒钟。这一背包问题的答案为选择第一、第二、第四、第五、第七个物品，它们的总价值为 157 元，总重量为 108 公斤。这是最优结果，但是用的时间比贪婪算法多。

如果将问题的规模扩大，修改原问题的 8 个物品为 10 个物品，在程序中物品价值和重量数据被修改为

```
c=[80 90 10 20 30 32 40 50 55 60];
a=[30 40 1 10 20 22 30 40 45 55];
```

程序中的其它语句不变，重新运行穷举法程序，屏幕显示

```
s =     1     1     1     1     0     1     0     0     0     0
        232   103
elapsed_time =     0.6600
```

这说明解决具有 10 个物品的背包问题计算机将花费 0.66 秒的时间。

四．应用篇

MATLAB 为我们提供了高效的编程环境，在解决实际问题和求解数学模型时，不必象使用一般的计算机语言那样将算法的每一个细节都考虑进去。用一系列的 MAATLAB 的命令解决问题时，实际上是将已有的程序模块进行组合拼装。在这样的高层次上考虑算法设计时，自然就节约了工作时间并避免了低层次的重复劳动。

73. 罗伦兹吸引子的空间曲线是如何绘制的？

罗伦兹微分方程组的解曲线——**Lorenz** 吸引子是三维空间中的一条曲线，如图 **4-1** 所示这条曲线相互缠绕而互不相交。如果将这条曲线视为某一动点的轨迹，这个动点将随自变量 ***t*** 的增大，在空间中的两个定点附近作环绕运动。

罗伦兹常微分方程组为

$$\begin{cases} \dfrac{dx}{dt} = -\beta x + yz \\ \dfrac{dy}{dt} = -\sigma(y - z) \\ \dfrac{dz}{dt} = -xy + \rho y - z \end{cases}$$

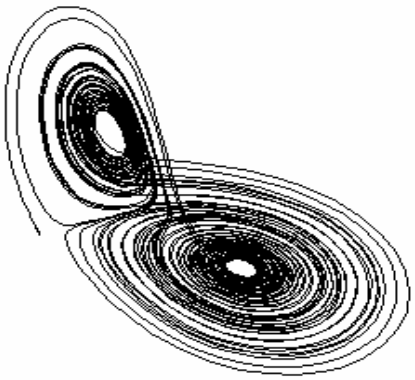


图 **4-1** 罗伦兹吸引子

给定初值条件

$$\begin{cases} x(0) = 0 \\ y(0) = 0 \\ z(0) = \varepsilon \end{cases}$$

取 $\beta = 8/3,\ \rho = 28,\ \sigma = 10$ ，$\varepsilon = \mathbf{2.2204\times 10^{-16}}$，则得微分方程组

$$\begin{cases} \dfrac{dx}{dt} = -8x/3 + yz \\ \dfrac{dy}{dt} = -10y + 10z \\ \dfrac{dz}{dt} = -xy + 28y - z \end{cases}$$

将三个方程的右端函数写成向量形式，得

$$\vec{f}(t,x,y,z) = \begin{bmatrix} -8x/3 + yz \\ -10y + 10z \\ -xy + 28y - z \end{bmatrix} = \begin{bmatrix} -8/3 & 0 & y \\ 0 & -10 & 10 \\ -y & 28 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

由于 ***MATLAB*** 中有常数 ***eps*** **= 2.2204×10^{-16}**，初始条件可以用列向量 **[0 0 eps]**^{T} 表示。首先建立描述微分方程组右端函数的函数文件(文件名为：**flo.m**)：

```
function z=flo(t,y)
```

A=[-8./3 0 y(2)；0 -10. 10.；-y(2) 28. -1]；

z=A*y；

将这一文件保存在 **MATLAB** 的工作目录下，然后在 **MATLAB** 环境中键入如下指令：

[t,y]=ode23('flo',0,80,[0 0 eps]');

u=y(:,1);v=y(:,2);w=y(:,3);plot3(u,v,w)

MATLAB 的图形窗口将显示 **Lorenz** 吸引子的图形如图 **4-1** 所示。另外，还可以绘制动画，如果关闭 ***MATLAB*** 的图形窗口，并在命令窗口中再键入

comet3(u,v,w)

便可以观察到 **Lorenz** 吸引子的图形生成过程的动态演示。命令 **comet3** 的使用格式和上面的 **plot3** 的使用格式相同，不同的是绘图效果增加了动感，其功能是在绘图时以动点模拟慧星运行并带有一条尾巴。

74．如何为足球队排名次？

已知我国 **8** 支足球队在 **1988～1989** 年全国足球甲级队联赛中的成绩，按国际足联公布的鼓励进攻的三分制计算比赛积分（胜—得 **3** 分；平—得 **1** 分；负—得 **0** 分）有下面的得分表

表 **4-1**

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
T_1	×	4	4	9	0	3	6	2
T_2	4	×	3	5	2	2	4	1
T_3	4	6	×	5	3	3	3	3
T_4	0	2	2	×	0	3	1	1
T_5	6	2	3	6	×	7	9	7
T_6	3	2	3	3	1	×	3	4
T_7	0	1	3	3	0	6	×	7
T_8	2	4	3	4	1	4	1	×

表中第 ***k*** 行数据表示第 ***k*** 队与其它各队比赛所得分数。为了依据现有的数据，对各足球队的实力给予科学客观的评价，可以用不同的方法排名次。第一种方案是依照各队的总分排名，下面的程序就是根据上表中的数据建立得分矩阵，将每一行求和得各队总分，然后根据总分排名次。

```
a=[0 4 4 9 0 3 6 2;4 0 3 5 2 2 4 1;4 6 0 5 3 3 3 3;0 2 2 0 0 3 1 1;
   6 2 3 6 0 7 9 7;3 2 3 3 1 0 3 4;0 1 3 3 0 6 0 7;2 4 3 4 1 4 1 0];
f=sum(a')
[ff l]=sort(-f)
```

运行程序后，屏幕显示

```
ff =    -40    -28    -27    -21    -20    -19    -19     -9
l =       5      1      3      2      7      6      8      4
```

由于 **MATLAB** 中的排序命令 ***sort*** 将数组由小到大排序，而我们需要将各队总分数按递减排序，所以将所有分数取负号后再使用排序命令。上面结果列表如下

表 4-2 总分排序数据结果

名次	1	2	3	4	5	6	7	8
队号	5	1	3	2	7	6	8	4
总分	40	28	27	21	20	19	19	9

第二种方案是根据得分矩阵的正特征值所对应的特征向量排名。利用上面程序中已经建立的得分矩阵求矩阵的特征值，用命令 **eig(a)** 可得

```
ans =
  20.3443
  -2.8104 + 5.1609i
  -2.8104 - 5.1609i
  -1.9756 + 2.1810i
  -1.9756 - 2.1810i
  -3.5585 + 0.8359i
  -3.5585 - 0.8359i
  -3.6552
```

观察数据结果知，第一个特征值为正，取这一正特征值对应的特征向量用命令

```
[x,d]=eig(a);g=x(:,1)'
```

得

```
-0.3636   -0.3224   -0.4133   -0.1438   -0.5680   -0.2938   -0.2837
                                -0.2880
```

这一向量每一元素均为负数，由特征向量的性质将其反号仍为特征向量。用排序命令

```
[f，l]=sort(g)
```

将特征向量元素排序，得

f = -0.5680 -0.4133 -0.3636 -0.3224 -0.2938 -0.2880 -0.2837 -0.1438

l = 5 3 1 2 6 8 7 4

由排序结果列表，得

表 4-3 特征向量方法排序结果

名次	1	2	3	4	5	6	7	8
队号	5	3	1	2	6	8	7	4
元素	0.5680	0.4133	0.3636	0.3224	0.2938	0.2880	0.2837	0.1438

比较两种排名方案结果，尽管结果不完全一致，但第一名和最后一名是一致的。这说明，分数是反映各队实力的一个方面，但并不是全面。最强的队与各队比赛总会得分，而弱队与其它队比赛总不会得分。在比赛中，一个实力中等的球队遇上强队时不会得分，遇上弱队时会得分，所以总分排名次的方法不是很公平的，特征向量法则比较公平。当特征向量中元素均为正时，各元素的大小就反映了各球队实力的强弱。

75. 怎样研究概率论中的生日问题？

假设每个人的生日在一年 **365** 天中的任意一天都是等可能性的，那么随机找 **n** 个人（不超过 **365** 人）。这 **n** 个人生日各不相同的概率（可能性）可用公式

$$P_1 = \frac{365 \times 364 \times \cdots \times (365 - n + 1)}{365^n}$$

计算出。因而， **n** 个人中至少有两人生日相同这一随机事件发生的概率为

$$P(n) = 1 - \frac{365 \times 364 \times \cdots \times (365 - n + 1)}{365^n}$$

首先利用上面的公式，用计算机分别计算出一个团体的人数分别为 n=1,2,…,100 时的 100 个概率值。用下面程序段

```
m=365;n=m;
for  k=1:100
  p(k)=1-n/m;
  m=m*365;n=n*(365-k);
end
```

使用指令 p(1)，p(2)，p(3)，…，p(100)可得出其相应的概率值。

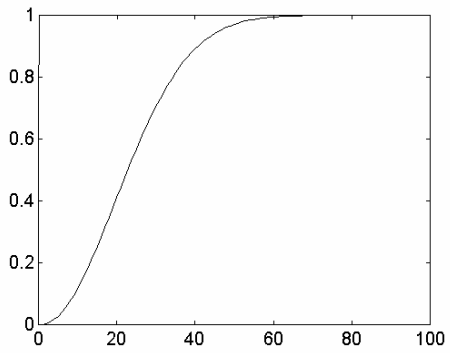


图 4-2 概率曲线

用指令 plot(p) 绘制出图形（图 4-2）,可以直观地了解概率值随团体人数变化的规律。

现在考虑几个特殊概率值，一个有三十个同学的学生班上，至少有两个同学“生日相同”的概率为 p(30)=0.7063 。五十个人的团体中，至少有两人生日在同一天（只考虑月日，不考虑年的不同）的概率为 p(50)=0.9704。在七十个人的团体中，p(70)=0.9992。

考虑团体总人数对概率值的影响，为了求在某团体中保证“至少有两人生日相同”的概率大于 99% 时的总人数。 键入下列命令

```
find(p>0.99)    (回车)
```

计算机屏幕将显示

```
ans=    57  58  59  60  61  ……  97  98  99  100
```

所以，团体人数若超过 57 人，则这个团体中至少有两人生日相同的概率将大于 99%。

最后，考虑用计算机数值模拟。随机产生 30 个正整数，介于 1 到 365 之间（代表 30 个同学的生日），然后统计数据，观察是否有两人以上的人生日相同。当 30 个人中有两人生日相同时，计算机输出为“1”，否则输出为“0”。如此重复观察 200 次，计算出这一事件发生的频率 f_{200}。下面由 MATLAB 程序做计算机模拟 200 次重复观察：

```
n=0;
for  m=1:200                          %做 200 次随机试验
y=0; x=1+fix(365*rand(1,30));         %产生 30 个随机数
  for i=1:29
```

```
    for j=i+1:30
        if x(i)= =x(j),y=1;break,end          %寻找 30 个随机数中是否
有相同数
    end
  end
n=n+y;                                        %累计有两人人生日相同
的试验次数
end
f=n/m                                         %计算频率
```

将程序重复运行三次后，数据结果为

f_{200}= 0.7350、 0.7150、 0.7250

比较前面所计算的 p(30)=0.7063，说明概率值和频率值比较接近，但是存在着差异。

76．怎样用动态规划算法求最短路径问题？

动态规划是用于分析一类多阶段决策过程的最优化方法。它解决问题的思路是将一个比较复杂的问题分解为多个同一类型的子问题，然后将这些子问题按整体最优的思想逐个地求最优解，最后再求出整个问题的最优解。

一个简单的最短路径问题描述如下：图 4-3 中 20 个结点表示 20 台不同地理位置上的计算机，为了将数据文件从编号为 20 的结点通过中间结点传输到编号为 1 的结点。求最短路径（数据在各相邻结点间传送的时间如图中所示）。

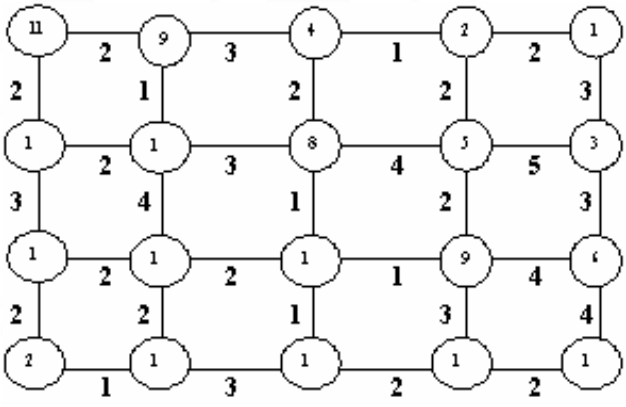


图 4-3 最短路问题

被传送的数据文件从编号为 20 的结点出发，最终到达编号为 1 的结点。无论哪一种方案都需要水平方向走 4 步，垂直方向走 3 步。从第 20 号结点到 1 号结点的所有路径数为

$$C_7^4 = \frac{7!}{4!(7-4)!} = \frac{7\times6\times5}{3\times2} = 35$$

用动态规划算法解决问题的理论依据是最优性原理，最优性原理可简述为：不论前面的状态和策略如何，以后的最优策略只取决于最初策略所确定的当前状态。具体求解问题时要用到递归方程，递归方程所起的作用是将一个子问题的解与下一个子问题相联系。由最优性原理可知，在从编号为 20 的结点到编号为 1 的结点的最短路径上，各结点到编号为 1 的结点的的路径也是各结点到编号为 1 的结点的最短路径。

我们称编号为 1 的结点为目标结点，用 d_k 表示编号为 k 的结点到目标结点的最短路径长度。由最优性原理可得

d_{20} = min{2+d_{18}，1+d_{19}}，d_{18} = min{3+d_{15}，2+d_{16}}，d_{19} = min{2+d_{16}，3+d_{17}}，
d_{15} = min{2+d_{11}，2+d_{12}}，d_{16} = min{4+d_{12}，2+d_{13}}，d_{17} = min{1+d_{13}，2+d_{14}}，
d_{11} = 2+d_7，d_{12} = min{1+d_7，3+d_8}，d_{13} = min{1+d_8，1+d_9}，d_{14} = min{3+d_9，3+d_{10}}，
d_7 =3+d_4，d_8 = min{2+d_4，4+d_5}，d_9 = min{2+d_5，4+d_6}，d_{10} = 4+d_6，
d_4 = 1+d_2，d_5 = min{2+d_2，5+d_3}，d_6 =3+d_3，d_2 = 2+ d_1，d_3 = 3+ d_1，d_1 = 0。

由 d_2 = 2，d_3 = 3，应用逆向递推方式，得

d_4 = 1+d_2 = 3，d_5 = min{2+d_2，5+d_3}=4，d_6 =3+d_3=6，……

d_{20} = min{2+d_{18}，1+d_{19}}=11。

最后再根据逆向递推的各式选择写出最短路径的路线。

用动态规划算法求解最短路径问题的程序如下：

```
x=[2 2 1;3 3 1;4 1 2;5 2 2;5 5 3;
    6 3 3;7 3 4;8 2 4;8 4 5;9 2 5;
    9 4 6;10 4 6;11 2 7;12 1 7;12 3 8;
    13 1 8;13 1 9;14 3 9;14 3 10;15 2 11;
    15 2 12;16 4 12;16 2 13;17 1 13;17 2 14;
    18 3 15;18 2 16;19 2 16;19 3 17;20 2 18;
    20 1 19];
x1=x;y=[1 1 0;2 1 2;3 1 3];
for k=4:20
    l=find(x1(:,1)= =k);
    d=x1(l,2);m=x1(l,3);
    mk=length(m);
```

```
        for i=1:mk
           j=m(i);
           mm(i,:)=find(y(:,1)= =j);
        end
        [v,n]=min(d+y(mm,3));
        y(k,:)=[k m(n) v];
    end
    z(1)=20;l=find(y(:,1)= =20);m=y(l,2);
    k=2;z(k)=m;
    while m>1
        k=k+1;
        l=find(y(:,1)= =m);
        m=y(l,2);z(k)=m;
    end
    rood=z
    Length=y(20,3)
```

将程序运行后，得数据结果：

```
rood =    20    19    16    13     8     4     2     1
Length =    11
```

结果表明，由动态规划算法求得的最短路径长度为 11，图 4-4 中箭头标出了最优解路径上的结点编号依次为

20→19→16→13→8→4→2→1

上面程序中，使用了一个 31×3 的矩阵。由于数据传输方向是由左向右，由下向上。故可将图作为有向图，每两个相邻结点的编号以及它们之间的一段路径长度形成一个三元数，即

[父结点号　路径长　子结点号]

所以，图中共 31 条路段的数据形成了

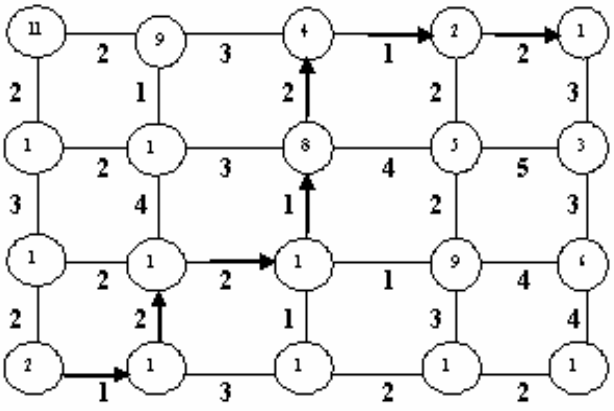


图 4-4　最短路问题的解

一个 31×3 的矩阵。以此为依据，从第 4 号结点到 20 号结点，逐步寻找它们到 1 号结点的最短路长度，同时记录每一结点到 1 号结点的最短路径上的子结点。形成一个三元数

[父结点　子结点　局部最短路径长]

最后，形成一个 20×3 的矩阵。再以此为依据，按秩序找出全局最短路径上的结点编号。

77．如何用 MATLAB 解决蠓虫分类问题？

生物学家试图对两类蠓虫(Af 与 Apf)进行鉴别,依据的资料是蠓虫的触角和翅膀的长度,已经测得 9 只 Af 和 6 只 Apf 的数据,(触角长度用 x 表示,翅膀长度用 y 表示)具体数据为:

表4-4 Af 类触角和翅膀长度

x	1.24	1.36	1.38	1.38	1.38	1.40	1.48	1.54	1.56
y	1.27	1.74	1.64	1.82	1.90	1.70	1.82	1.82	2.08

表4-5　Apf 类触角和翅膀长度数据

x	1.14	1.18	1.20	1.26	1.28	1.30
y	1.78	1.96	1.86	2.00	2.00	1.96

现需要解决三个问题:（1）如何凭借原始资料(15 对数据,被称之为学习样本)制定一种方法,正确区分两类蠓虫;（2）依据确立的方法,对题目提供的三个样本：(1.24,1.80)，(1.28,1.84)，(1.40,2.04)加以识别;（3）设Af是宝贵的传粉益虫,Apf是某种疾病的载体,是否应该修改分类方法。

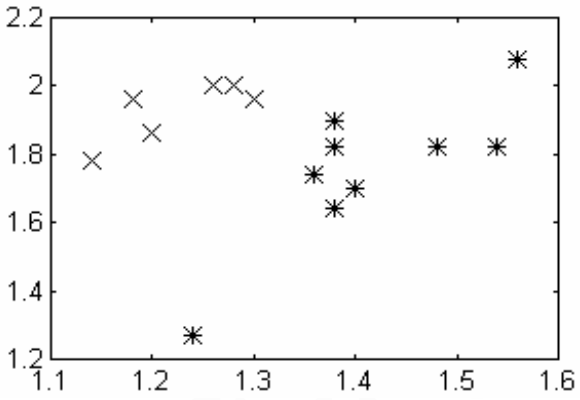


图 4-5 由触角和翅膀数据画的离散点图

首先画出15对数据的散点图，其中，Af 用 * 标记，Apf 用 × 标记。观察图4-5 可以发现，

代表 **Af** 的点位于图中偏右，而代表 **Apf** 的点位于图中偏左。应该存在一条位于两类点之间的直线 ***L***， 作为 **Af** 和 **Apf** 分界线，这条直线 ***L*** 应依据问题所给的数据，即学习样本来确定。设这条直线的方程为

$$w_1x + w_2y + w_3 = 0$$

对于平面上任意一点 $P(x, y)$，如果该点在直线上，将其坐标代入直线方程则使方程成为恒等式，即使方程左端恒为零；如果点 $P(x, y)$ 不在直线上，将其坐标代入直线方程，则方程左端不为零。由于 **Af** 和 **Apf** 的散点都不在所求的直线上，故将问题所提供的数据代入直线方程左端所得到的表达式的值应有大于零或者小于零两种不同的结果。

建立一个判别系统，引入判别函数 $g(x, y)$，当 (x, y) 代表 **Af** 类时，令 $g(x, y)>0$，否则 $g(x, y)<0$。

为了对判别系统引入学习机制，在学习过程中将两种不同的状态，以“**1**”和“**－1**”表示。取

$$g(x, y) = w_1x + w_2y + w_3$$

则由所给数据形成约束条件，这是关于判别函数中的三个待定系数值 w_0, w_1, w_2 的线性方程组：

$$\begin{cases} w_1x_j + w_2y_j + w_3 = 1, \ (j = 1,2,\cdots,9) \\ w_1x_j + w_2y_j + w_3 = -1, \ (j = 10,\cdots,15) \end{cases}$$

此为包括三个未知数共15个方程的超定方程组，可以求方程组的最小二乘解。

下列程序用于求超定方程组的最小二乘解，并绘制出分类边界曲线的图形。

```
xy=[1.24 1.27;1.36 1.74;1.38 1.64;1.38 1.82;1.38 1.90;
    1.40 1.70;1.48 1.82;1.54 1.82;1.56 2.08;1.14 1.78;
    1.18 1.96;1.20 1.86;1.26 2.00;1.28 2.00;1.30 1.96];      %学习样本数据
z=[1;1;1;1;1;1;1;1;1;-1;-1;-1;-1;-1;-1];
x=xy(:,1);y=xy(:,2);x1=x(1:9);y1=y(1:9);x2=x(10:15);y2=y(10:15);
plot(x1,y1,'*',x2,y2,'x'),pause                              %绘制原始数据散点图
X=[x y ones(x)];
A=X'*X;B=X'*z;w=A\B;                                         %求解正规方程组
a=－w(1)/w(2);b=－w(3)/w(2);
```

```
t=1.10:0.02:1.60;u=a*x+b      ;                    %确定分类直线数据
plot(x1,y1,'*',x2,y2,'x',t,u)                      %在散点图中画分类直线
```

运行上面程序可求出超定方程组的最小二乘解并画出分类边界曲线。为了由所给数据用判别函数判别三个新蠓虫的类属，即当 $g(x,y)>0$ 时，判为 **Af** 类；当 $g(x,y)<0$ 时，判为 **Apf** 类。运行上面程序后,键入下面命令

```
xx=[1.24 1.80 1;1.28 1.84 1;1.40 2.04 1];
xx*w
plot(t，u，xx(:,1)，xx(:,2)，'o')
```

求得

```
ans =    -0.3877
          -0.2384
          -0.0235
```

这说明，所给数据反映出三个蠓虫均属于**Apf**类。

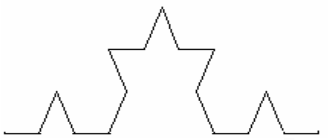
另外，上面的解决问题方案是在学习过程中用 **+1** 和 **–1** 分别代表正数和负数来完成的。这只是一种人为的规定，并不是一成不变的。当**Apf**是害虫时，可以修改超定方程组的右端项中“**-1**”为“**-0.6**”，或者将 **–1** 改为其它的负数以重新求超定方程组的最小二乘解获得分类边界直线的方程。这样将与前面所求分类边界直线的方程不一样，当然对新给定的蠓虫的翅膀和触角长度数据来做判断其结果也是不同的。

78．如何绘制分形曲线图形？

从一条直线段开始，将线段中间的三分之一部分用一个等边三角形的另两条边代替，形成具有 **5** 个结点的新的图形(图 **4-6**)；在新的图形中，又将图中每一直线段中间的三分之一部分都用一个等边三角形的另两条边代替，再次形成新的图形(图 **4-7**)，这时，图形中共有 **17** 个结点。

图 **4-6** 第一次迭代

图 **4-7** 第二次迭代



这种迭代继续进行下去可以形成 **Koch** 分形曲线。在迭代过程中，图形中的点将越来越多，而曲线最终显示细节的多少将取决于所进行的迭代次数和显示系统的分辨率。**Koch** 分形曲线的定义归结于算法和计算机实现。

考虑由直线段(2 个点)产生第一个图形(5 个点)的过程。设 P_1 和 P_2 分别为原始直线段的两个端点。现在需要在直线段的中间依次插入三个点 Q_1, Q_2, Q_3 产生第一次迭代的图形。显然，Q_1 位于 P_1 点右端直线段的三分之一处，Q_3 位于 P_1 点右端直线段的三分之二处；而 Q_2 点的位置可以认为是，由 Q_3 点绕 Q_1 旋转 60 度(逆时针方向)而得到的，故可以处理为向量 Q_1Q_3 经正交变换而得到向量 Q_1Q_2。形成算法如下：

(1) $Q_1 = P_1 + (P_2 - P_1)/3$;

(2) $Q_3 = P_1 + 2(P_2 - P_1)/3$;

(3) $Q_2 = Q_1 + (Q_3 - Q_1) \times A'$;

(4) $P_5 = P_2$; $P_2 = Q_1$; $P_3 = Q_2$; $P_4 = Q_3$.

在算法的第三步中，***A*** 为正交矩阵：

$$A = \begin{bmatrix} \cos\frac{\pi}{3} & -\sin\frac{\pi}{3} \\ \sin\frac{\pi}{3} & \cos\frac{\pi}{3} \end{bmatrix}$$

这一算法的结果将由初始数据 P_1 和 P_2 点的坐标，产生 5 个结点的坐标。这 5 个结点的坐标数组，组成一个 5×2 矩阵。这一矩阵的第一行为 P_1 的坐标，第二行为 P_2 的坐标，…，第五行为 P_5 的坐标。矩阵的第一列元素分别为 5 个结点的 **X** 坐标，第二列元素分别为 5 个结点的 **Y** 坐标。

首先考虑在 **Koch** 分形曲线的形成过程中结点数目的变化规律。设第 **k** 次迭代产生结点数为 n_k，第 **k+1** 次迭代产生结点数为 n_{k+1}，n_k 和 n_{k+1} 之间的递推关系式如下

$$n_{k+1} = 4n_k - 3$$

由第 **k** 次迭代的 n_k 个结点的结点坐标数组，产生第 **k+1** 次迭代的 n_{k+1} 个结点的结点坐标数组的算法可参考上面两点到五点的算法进行设计。

根据算法编写下面迭代五次的程序并绘制 **Koch** 分形曲线

```
p=[0    0;10    0];                          %给出初始数据两个点的
坐标
a=[cos(pi/3)    -sin(pi/3);sin(pi/3)    cos(pi/3)];      %设置用于正交变化的正
交矩阵
```

```
for   k=1:5                                    %开始执行第一到第五次迭代
    n=max(size(p));d=diff(p)/3;                %统计前一轮迭代的结点数及形成结点向量
    q=p(1:n-1,:);p(5:4:4*n-3,:)=p(2:n,:);      %保护前一轮的结点坐标数组
        p(2:4:4*n-6,:)=q+d;                    %插入第一组新结点
    p(3:4:4*n-5,:)=p(2:4:4*n-6,:)+d*a';        %用正交变换计算第二组新结点
    p(4:4:4*n-4,:)=q+2*d;                      %插入第三组新结点
end
plot(p(:,1),p(:,2))                            %根据结点坐标绘图
```

运行上面程序后，MATLAB 的图形窗口将显示图 4-8 中的分形图形。

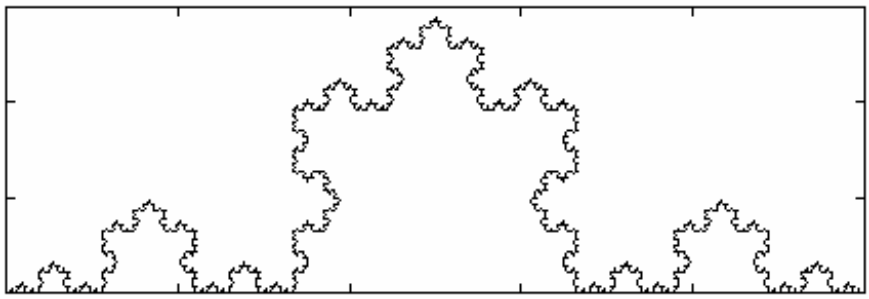


图 4-8　Koch 分形曲线

分形曲线的绘制可以有很多丰富多彩的变化。例如将上面分形曲线中正交变换的逆时针 60^0 旋转修改为 120^0 旋转，并将两点连线变成一个四边形，以四边形的四个顶点坐标为初始数据绘制第五次迭代产生的分形曲线图形，将大为改观。

修改上面程序，首先，将初始数据　p=[0　0;10　0] 修改为

p=[0 0;5 -5;10 0;5 5;0 0]

并将语句 a=[cos(pi/3) -sin(pi/3);sin(pi/3) cos(pi/3)]中的旋转达角度由 $\pi/3$ 改为 $2\pi/3$，即修改为

a=[cos(2*pi/3) -sin(2*pi/3);sin(2*pi/3) cos(2*pi/3)]

其它语句不变，运行修改后的程序可得图 4-9 中的分形图形。

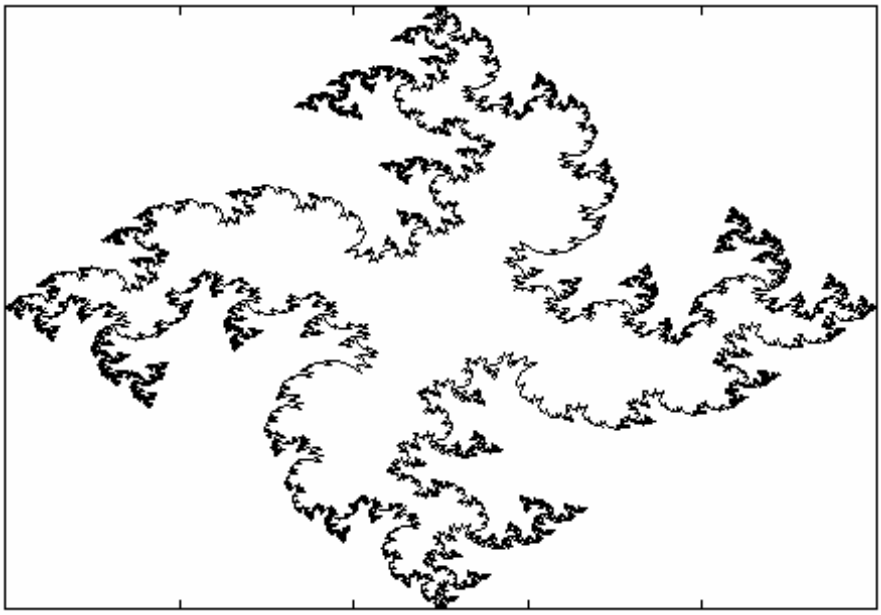


图 4-9　由四边形不断迭代计算所得分形曲线图

79．如何用聚类算法研究气象观测站问题

某地区内有 12 个气象观测站，10 年来各站测得的年降雨量如下表 4-6 所示。为了节省开支，有关部门想要减少四个气象观测站。如果不考虑各观测站的地理位置，试根据表中的数据推算出 10 年来降雨量数据相近的站点，以供参考。

表 4-6 气象观测站 10 年降雨量数据

	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
x_1	276.2	251.6	192.7	246.2	291.7	466.5	258.6	453.4	158.5	324.8
x_2	324.5	287.3	436.2	232.4	311.0	158.9	327.4	365.5	271.0	406.5
x_3	158.6	349.5	289.9	243.7	502.4	223.5	432.1	357.6	410.2	235.7
x_4	412.5	297.4	366.3	372.5	254.0	425.1	403.9	258.1	344.2	288.8
x_5	292.8	227.8	466.2	460.4	245.6	251.4	256.6	278.8	250.0	192.6
x_6	258.4	453.6	239.1	158.9	324.8	321.0	282.9	467.2	360.7	284.9
x_7	334.1	321.5	357.4	298.7	401.0	315.4	389.7	355.2	376.4	290.5
x_8	303.2	451.0	219.7	314.5	266.5	317.4	413.2	228.5	179.4	343.7
x_9	292.9	466.2	245.7	256.6	251.3	246.2	466.5	453.6	159.2	283.4

x_{10}	243.2	307.5	411.1	327.0	289.9	277.5	199.3	315.6	342.4	281.2
x_{11}	159.7	421.1	357.0	296.5	255.4	304.2	282.1	456.3	331.2	243.7
x_{12}	331.2	455.1	353.2	423.0	362.1	410.7	387.6	407.2	377.7	411.1

由于要从 12 个站点中减少四个气象观测点，所以应保留八个站点．每一站点在 1981 年至 1990 年的十年中所记录的年降雨量数据在数学上可视为一向量。现在需要用聚类算法将 12 个向量划分为 8 个组，在聚类过程中应记录下每一类所含站点的数目以及每一站点属于哪一类。聚类算法的思路是，在开始时每一个站点算一类，总共 12 类。考虑数据最接近的两个站点，将它们聚为一类，并计算这两个站点的降雨量数据的平均值，作为新一类的数据。数据块化为 11 个类，重复这一过程，直到减少至 8 类为止。

程序如下

```
n1=8;    x=[
276.2  251.6  192.7  246.2  291.7  466.5  258.6  453.4  158.5  324.8;
324.5  287.3  436.2  232.4  311.0  158.9  327.4  365.5  271.0  406.5;
158.6  349.5  289.9  243.7  502.4  223.5  432.1  357.6  410.2  235.7;
412.5  297.4  366.3  372.5  254.0  425.1  403.9  258.1  344.2  288.8;
292.8  227.8  466.2  460.4  245.6  251.4  256.6  278.8  250.0  192.6;
258.4  453.6  239.1  158.9  324.8  321.0  282.9  467.2  360.7  284.9;
334.1  321.5  357.4  298.7  401.0  315.4  389.7  355.2  376.4  290.5;
303.2  451.0  219.7  314.5  266.5  317.4  413.2  228.5  179.4  343.7;
292.9  466.2  245.7  256.6  251.3  246.2  466.5  453.6  159.2  283.4;
243.2  307.5  411.1  327.0  289.9  277.5  199.3  315.6  342.4  281.2;
159.7  421.1  357.0  296.5  255.4  304.2  282.1  456.3  331.2  243.7;
331.2  455.1  353.2  423.0  362.1  410.7  387.6  407.2  377.7  411.1];
x0=x;n=max(size(x));                          %统计数据块大小
p=[1:n]';q=ones(n,1);                         %置 12 个类为每类一个站点
for k=1:n-n1
    nk=max(size(x(:,1)));d0=1000000;
    for i=1:nk-1
        xi=x(i,:);
        for j=i+1:nk
            xj=x(j,:);
```

```
                d1=norm(xi-xj);
                if d1<d0,d0=d1;i0=i;j0=j;end                    %寻找距离最短的
两个类
            end
        end
        i=i0;j=j0;
        x(i,:)=q(i)*x(i,:)+q(j)*x(j,:);
        q(i)=q(i)+q(j);x(i,:)=x(i,:)/q(i);                      %计算新一类的数据
均值
        x(j,:)=[];q(j)=[];                                      %删除旧的类
        lj=find(p==j);p(lj)=i*ones(size(p(lj)));
        lj=find(p>j);p(lj)=p(lj)-1;                             %重新定义各站点的
类属
    end
    disp(q)                                                     %输出分类结果
    pp=[1:12];disp([pp;p])
```

程序运行后计算机屏幕显示各气象站的数据分类结果如下：

```
1     1     1     3     2     2     1     1
1     2     3     4     5     6     7     8     9     10    11    12
1     2     3     4     5     6     4     7     8     5     6     4
```

第一行 8 个数据表明了 8 个组各含有几个气象站，第二行为各气象站编号，第三行数据表明 12 个观测站分别被聚类到 8 个组中的哪一个组，即

第一组：{ x_1 }；　　第二组：{ x_2 }；

第三组：{ x_3 }；　　第四组：{ x_4 ， x_7 x_{12}}；

第五组：{ x_5 ， x_{10} }；　第六组：{ x_6 ， x_{11} }；

第七组：{ x_8 }；　　第八组：{ x_9 }

其中有五组每组只含一个气象站，另外三组每组含至少有两个气象站。它们之中可以考虑每组减去一个或两个气象观测站。

80．如何用穷举法求解截断切割问题？

这一问题选自 1997 年全国大学生数学建模竞赛 B 题。问题简单叙述如下：

某些工业部门（如贵重石材加工等）采用截断切割的加工方式从一个长方体中加工出一个已知尺寸、位置预定的长方体（这两个长方体的对应表面是平行的），通常要经过六次截断切割. 已知待加工长方体和成品长方体的长、宽、高分别为 **10**、**14.5**、**19** 和 **3**、**2**、**4,** 二者左侧面、正面、底面之间的距离分别为 **6**、**7**、**9 (**单位均为厘米**)**。 若切割费用为每平方厘米 **1** 元。 试为这些部门设计一种安排各面加工次序的方案，使加工费用最少。

首先考虑各种不同切割方式的数学描述. 将左、右、前、后、上、下相应的切割面编号为 **1**、**2**、**3**、**4**、**5**、**6**。于是，一个切割方式就是各加工面 **{ 1, 2, 3, 4, 5, 6 }**的一个全排列，记为：$\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5\sigma_6$。将所有不同的切割方式组成的集合记为

$$\Sigma = \{(\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5\sigma_6) \mid \sigma_i = 1,2,3,4,5,6\ (i = 1,2,3,4,5,6); \sigma_i \neq \sigma_j\ (i \neq j)\}$$

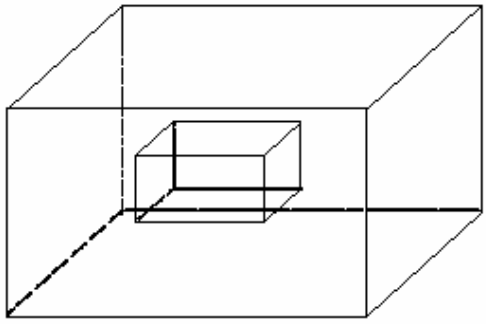


图 4-10 待加工长方体及成品长方体图形

现在考虑使用任一种切割方式 $\sigma = (\sigma_1\ \sigma_2\ \sigma_3\ \sigma_4\ \sigma_5\ \sigma_6)$ 的费用的描述。 由于这种切割方式在操作过程中总共要进行六次切割，记六次切割的面积依次为

$$S(\sigma_1), S(\sigma_1\sigma_2), S(\sigma_1\sigma_2\sigma_3), S(\sigma_1\sigma_2\sigma_3\sigma_4), S(\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5),$$

$$S(\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5\sigma_6)$$

则，六次切割的总费用为

$$f(\sigma) = \sum_{i=1}^{6} S(\sigma_1 \cdots \sigma_i)$$

由此可知，截断切割问题实际上是在可行域 Σ 上求目标函数 $f(x)$ 的最小

值。 由于这一问题规模不是很大，可以用穷举法求解。

设计算法如下

第一步：列出所有可能的切割方案，即 720 个操作数(720×6 阶的矩阵)；

第二步：计算每一种切割方案的总费用(720 个费用数据)；

第三步：从 720 个费用数据中选出最小值；

第四步：列出费用最少的全部操作方案，结束。

在这一算法中，比较困难的是第二步的算法实现. 有关细节考虑如下：

1. 待加工长方体的长、宽、高为 10、14.5、19 用向量记为

$$l_0 = (10 \quad 14.5 \quad 19)$$

2. 成品长方体的长、宽、高为 3、2、4，它距待加工长方体左侧面、正面、底面之间的距离分别为 6、7、9。由此我们可以计算出切割左，右，前，后，上，下各面的切割厚度数据，用向量记为

$$h = (6 \quad 1 \quad 7 \quad 5.5 \quad 6 \quad 9)$$

3. h 有六个数据，在切割过程中，将待加工的长方体切割为成品长方体要经历六种状态，每一状态均为从前一状态长方体的长、宽、高数据中的某一个减去对应的切割厚度数据而形成的新长方体。被减数只有三个而减数有六个，其.对应关系列表为

表 4-7

	对应	对应	对应
被减数指标	1	2	3
减数指标	1，2	3，4	5，6

例如当切割左面或切割右面时，都应该从长、宽、高数据中的第一个数减去切割厚度数据。

4. 切割面积的计算。每一次切割面的面积实际上是本次切割状态中半成品长、宽、高数据中不改变的两个数据的乘积。

程序如下

```
l1=[1 2 3 4 5 6];k=0;                          %设置 l1 = (1, 2, 3, 4, 5, 6)
for j1=1:6,v1=l1(j1);l2=l1;l2(j1)=[];          %从 l1 中选取 v1 并设置 l2
    for j2=1:5,v2=l2(j2);l3=l2;l3(j2)=[];      %从 l2 中选取 v2 并设置 l3
        for j3=1:4,v3=l3(j3);l4=l3;l4(j3)=[];  %从 l3 中选取 v3 并设置 l4
            for j4=1:3,v4=l4(j4);l5=l4;l5(j4)=[];  %从 l4 中选取 v4 并
```

设置l_5

```
                for j5=1:2,v5=l5(j5);v6=l5(3-j5);          %从 l5 中选取 v5 与
v6
                k=k+1;p(k,:)=[v1 v2 v3 v4 v5 v6];          %将[v1 v2 v3 v4 v5 v6]
赋于 P
    end,end,end,end,end
    clear l1 l2 l3 l4 l5 j1 j2 j3 j4 j5 v1 v2 v3 v4 v5 v6
    l0=[10 14.5 19];h=[6 1 7 5.5 6 9];u=[1 1 2 2 3 3];
    for k=1:720
    v=p(k,:);l=l0;s=0;                                     %提取 P 的第 k 行数据
    for j=1:6
        i=v(j);l(u(i))=l(u(i))-h(i);                       %模拟截割第 i 面并减去
相应的厚度
        ll=l;ll(u(i))=[];s=s+ll(1)*ll(2);                  %累加计算本次截割面的
费用
    end
    q(k)=s;                                                %将六次截割的总面积赋值给 Q
    end
    clear h j i k l l0 ll s u v
    an=min(q);find(q==an);p(ans,:),                        %寻求 Q 中的最小者
    an
```

运行程序后，操作数有两组

6 3 1 5 4 2

6 3 5 1 4 2

切割费用为： **374**， 达到最小值。 对应的切割方案如下

表 4-8

费用最少的切割方案	所需切割费用
底面，前面，左面，上面，后面，右面	**374**
底面，前面，上面，左面，后面，右面	**374**

81．如何进行文本操作

如果数据已经保存在文本文件里，我们可以用读取文件的方法来创建数据文件。例如，某地区有 12 个气象观测站，记录了 10 年以来所测得的每年的年降雨量数据(保存在一个名为”data.txt”的文件里)。每一个站点的十个数据构成了矩阵的一个行向量，12 个站点的数据就构成了一个 12×10 的矩阵。用 matlab 读取这个具有 120 个数据的文件名为 data.txt 的数据文件的步骤如下：

第一步：创建一个 M 文件；

第二步：在 M 文件中键入如下命令.

```
fid=fopen('data.txt');
    mat=[];
    while 1
        tline = fgetl(fid);
        disp(tline)
        if ~ischar(tline), break, end
        pos=find(tline==';')
        if length(pos)>0,
            tline(pos)=[];
        end
        vec = str2num(tline);
        mat=[mat;vec];
    end
    fclose(fid);
    mat
```

这样屏幕上就显示:

```
mat =

  276.2000  251.6000  192.7000  246.2000  291.7000  466.5000  258.6000  453.4000  158.5000
324.8000

  324.5000  287.3000  436.2000  232.4000  311.0000  158.9000  327.4000  365.5000  271.0000
406.5000

  158.6000  349.5000  289.9000  243.7000  502.4000  223.5000  432.1000  357.6000  410.2000
235.7000
```

412.5000 297.4000 366.3000 372.5000 254.0000 425.1000 403.9000 258.1000 344.2000
288.8000

292.8000 227.8000 466.2000 460.2000 245.6000 251.4000 256.6000 278.8000 250.0000
192.6000

258.4000 453.6000 239.1000 158.9000 324.8000 321.0000 282.9000 467.2000 360.7000
284.9000

334.1000 321.5000 357.4000 298.7000 401.0000 315.4000 389.7000 355.2000 396.4000
290.5000

303.2000 451.0000 219.7000 314.5000 266.5000 317.4000 413.2000 228.5000 179.4000
343.7000

292.9000 466.2000 245.7000 256.6000 251.3000 246.6000 466.5000 453.6000 159.2000
283.4000

243.2000 307.5000 411.1000 327.0000 289.9000 277.5000 199.3000 315.6000 342.4000
281.2000

159.7000 421.2000 357.1000 296.5000 255.4000 304.2000 282.1000 456.3000 331.2000
243.7000

331.2000 455.1000 353.2000 423.0000 362.1000 410.7000 387.6000 407.2000 377.7000
411.10007.