数学建模及其基础知识详解

王文波　编著

武 汉 大 学 出 版 社

内 容 简 介

本书详细、系统地介绍了数学建模中所用到的微积分、线性代数、常微分方程、概率论与数理统计、最优化和图论等知识，并重点讲解了这些知识在 Matlab 中的编程实现方法，书中给出了大量经典建模实例和模型在 Matlab 中的实现方法。

本书共分四个部分，第一部分是数学建模中所用到的数学知识的系统介绍以及它们在 Matlab 中的实现方法；第二部分是数学模型的实例的分析，精选了大量的经典例题和真题，详细地给出了每个例题的建模过程以及在 Matlab 中的实现方法；第三部分介绍数学软件 Matlab 的用法，包括 Matlab 的基础知识和 Matlab 中的高级图形编程知识；第四部分附录收集了历年全国大学生数学建模竞赛试题。

书中以微积分、线性代数、常微分方程、概率论与数理统计、图论为知识背景，以模型实例为载体，以数学软件 Matlab 为工具，将数学知识、数学建模与数学软件应用三者有机地结合起来。

前　言

全国大学生数模竞赛现在不论是参加的省区、学校的数目，还是参赛的队数、人数，都是目前全国规模最大的课外科技活动。很多不同专业的同学都对数学建模很感兴趣，积极踊跃地报名参加数模竞赛。

参加数模竞赛的同学需要具备下面两方面的能力：第一是数学知识的应用能力；第二是计算机应用能力。可是，很大一部分同学在学习建模的过程中都感到非常困惑和吃力：首先就是要学的数学基础知识这么多，该怎么学？学哪些？学到什么程度为止？其次，众多同学在对模型编程求解时感到很困难，想要学习这方面的知识却又感到无从下手，数学软件包括的内容那么多，哪些是建模中常用的，需要重点学习？哪些只需要了解就可以了？

编写本书的目的就是为了帮助参加数模竞赛的同学解决这两方面的问题。

第一，诚然，数学建模中所涉及的数学知识太多，根据历年比赛的试题可以看出，数模竞赛涉及的数学知识面十分宽广，包括数学分析、线性代数、概率论与数理统计、最优化理论、图论、微分方程求解及稳定性分析等几乎全部的数学基础知识。其中任何一门课如果想要透彻地学习，都需要花一年以上的时间。但对参加数模竞赛的同学来说，不可能有这么多的时间来系统地学习每门知识。在和众多的参赛同学和竞赛辅导老师进行探讨，并对历年竞赛的题目进行研究后，本书对参加数模竞赛需要掌握的知识点进行了归纳总结，对每个知识点给出了相应的理论、概念和计算实例，使读者对每科的知识结构有一个清晰的认识，便于学习和在竞赛中使用，帮助读者分清主次，而不是茫然、毫无头绪地去学习。

第二，在建模过程中，对所建立的模型求解时，用手工计算几乎是不可能的，基本上都要借助计算机来编程实现。当前，最流行的数学软件就是 Matlab 和 Mathmatical。Matlab 是 1984 年由美国 Math Works 公司推出的数学软件，其具有优秀的数值计算能力和数据可视化能力。该软件不但可以解决数学中的数值计算问题，还可以解决符号演算问题，并且能够方便地绘出各种函数图形。Matlab 提供的各种函数可以避免在解决问题中做繁琐的数学推导和计算。

只是 Matlab 本身的内容过于丰富复杂。而求解数学模型时，所需的知识又涉及 Matlab 的很多领域，对于一个利用业余时间参加数模竞赛的同学来说，对

Matlab 的每个领域都从头开始进行详细的学习,那也是不可能的。本书中针对建模中所常用的 Matlab 知识进行了系统的归纳、分类和总结,使得读者能在很短的时间内就可以掌握数学分析、线性代数、最优化、概率论、图论和微分方程等知识在 Matlab 中的求解方法,从而能够在 Matlab 中编程求解模型。

本书具有以下特点:

1. 详尽、清晰的基础知识归纳、分析和讲解。对建模中所有用到的每一科数学基础知识都进行了详尽、清晰的归纳和总结,给出了它们的概念、理论和算法推导。但本书并不能代替专业的数学书籍,在需要对问题进行更深入的探讨时,书中简略给出了探讨的方向及方法。

2. 数学知识与数学软件 Matlab 紧密的结合。对每一个数学知识点都详细地介绍了其在 Matlab 中的求解方法,并给出了大量的例子。书中也对 Matlab 中的符号计算、概率工具箱、优化工具箱、图像处理等进行了详尽的归纳总结,使读者易学易用。

3. 大量真题和经典习题的详细解答。书中选用的例题一部分是非常经典的数学建模例题,还有一部分是历年全国大学生数模竞赛中的真题。对每一个例题都详尽地给出了建模的过程、模型在 Matlab 中求解的分析过程和源代码以及计算结果,使读者更容易掌握模型在数学软件中的求解方法。

本书共分为四部分,第一部分是数学基础知识部分,详细地介绍了数学分析等知识点的理论和算法以及在 Matlab 中的实现方法;第二部分是数学建模和求解实例,精选了大量的真题及经典例题,给出了例题的建模方法以及在 Matlab 中的求解方法;第三部分是数学软件 Matlab 的基础知识和关于图形的高级编程知识;第四部分是附录,介绍了近些年来全国大学生数模竞赛的试题。

由于作者水平有限,书中难免有不足之处,尤其是在一些内容安排上,恐有偏颇之处,恳切希望读者批评指正。

作　者

2005 年 7 月于武汉珞珈山

目　录

第一部分　基础知识

第二部分 建模实例

第三部分 Matlab 基础和高级编程

第四部分　附　　录

第一部分　基础知识

第一章　微积分、线性代数的基础知识及其在 Matlab 中的实现

§1.1　数学建模中常用的微积分知识在 Matlab 中的实现

在数学建模问题的计算中，经常要用到极限、导数、微分和积分等基本运算．利用数学软件 Matlab 可以使复杂的微积分运算变得很容易．本节介绍微积分的基本运算在计算机上如何实现，包括：微分、积分、极限、级数求和、方程求根以及 Taylor 展开等功能的实现方法．

下面是 Matlab 中的基本符号运算函数（相关符号运算可以参考附录 A）：

1. x = sym ('x')

功能：创建一个符号变量 x.

2. syms　x　y　z

功能：创建多个符号变量 x，y，z.

3. r = collect (S,v)

功能：合并同类项，S 是符号表达式，v 是变量或表达式，r 是合并同类项后的结果．

4. factor (S)

功能：符号计算的因式分解，S 是待分解的符号多项式.

5. expand (S)

功能：对符号多项式或函数 S 进行展开．

6. r = simple (S) 或 r = simplify (S)

功能：对符号表达式 S 进行化简．

7. subs (S,old,new)

功能:把符号变量中的变量 old 用 new 代替,new 可以是一个符号,也可以是具体的数.

8. vpa (S)

功能:对符号表达式 S 计算其任意精度的数值.

9. eval (S)

功能:计算符号表达式(或字符串) S.

1.1.1 导数、极值和积分、Taylor 公式及在 Matlab 中的实现

1. **极限运算**

在 Matlab 中,计算极限采用如表 1.1 所示的命令.

表 1.1

命　　令	功　　能
limit (f,x,a)	计算 $\lim\limits_{x\to a} f(x)$
limit (f,x,inf)	计算 $\lim\limits_{x\to} f(x)$
limit (f,x,a,'right')	计算单侧极限 $\lim\limits_{x\to a^+} f(x)$
limit (f,x,a,'left')	计算单侧极限 $\lim\limits_{x\to a^-} f(x)$

注意　在左、右极限不相等或左、右极限有一个不存在时,Matlab 的默认状态是求右极限.

例 1. 求极限 $\lim\limits_{x\to 0}(1+4x)^{\frac{1}{x}}$ 与极限 $\lim\limits_{x\to 0}\dfrac{e^x-1}{x}$.

解　Matlab 命令为

```
syms x;
y1 = (1 + 4 * x) (1 / x) ; y2 = (exp (x) - 1) / x ;
limit (y1 , x , 0)
ans = exp (4)
limit (y2 , x , 0)
ans = 1.
```

例 2. 求极限 $\lim\limits_{x\to 0^+}\left(\sqrt{x}-2^{-\frac{1}{x}}\right)$.

解

```
syms x ; y = sqrt (x) - 2 (- 1 / x) ;
limit (y , x , 0 , 'right')
ans = 0.
```

2. 求导运算

导数是函数增量 Δy 与自变量增量 Δx 之比的极限（$\Delta x \to 0$），即 $f'(x) = \lim\limits_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$. 在 Matlab 中求函数的导数及其他一些类似计算均由 diff 命令来完成.

(1) 一元函数的求导

命令形式 1：diff (f)

功能：求函数 f 的一阶导数，其中 f 为符号函数.

命令形式 2：diff (f,n)

功能：求函数 f 的 n 阶导数，其中 f 为符号函数.

例 3. 求函数 $3x^3 + 5x + 1$ 的二阶导数.

解　Matlab 的命令为

```
syms x ;f = 3 * x 3 + 5 * x + 1 ;
diff (f,2) ;
ans = 18 * x.
```

例 4. 设 $y = 3x^2 - 2x + 1$，求 $y'|_{x=1}$.

解　

```
syms x ;y = 3 * x 2 - 2 * x + 1 ;
B = diff (y) , x = 1 ;
eval (B)
```

运行结果为

```
B = 6 * x - 2
ans = 4.
```

(2) 多元函数的偏导数

将函数 $z = f(x,y)$ 中的变量 y 看成常量而对变量 x 的导数称为二元函数 $f(x,y)$ 对变量 x 的偏导数，记为 $f_x(x,y)$. 若把 x 当做常量而对 y 求导的结果称为函数对 y 的偏导数，记为 $f_y(x,y)$. 求偏导数的方法和一元函数的求导方法一样，只要把另一个变量看成常量即可.

命令形式 1：diff (f,x_i)

功能：多元函数 f 对变量 x_i 的一阶偏导.

命令形式 2：diff (f,x_i,n)

功能：多元函数 f 对变量 x_i 的 n 阶偏导.

例 5. 求 $z = x^2 \sin 2y$ 关于 x 的偏导数.

解　

```
syms x y;
z = x 2 * sin (2 * y) ;B = diff (z' ,x') .
```

运行结果为 B = 2 * x * sin (2 * y) .

(3) 全微分、参数方程求导及隐函数求导

1) 若函数 z = f (x,y) 在某点 (x_0, y_0) 的两个偏导数存在且连续,则函数在该点的全微分为

$$dz = f_x(x_0, y_0)\,dx + f_y(x_0, y_0)\,dy$$

Matlab 中求函数全微分的命令为:diff (z,x) + diff (z,y) .

2) 对参数方程 $\begin{cases} x = x(t) \\ y = y(t) \end{cases}$ 所确定的函数 y = f (x) ,根据公式 $\frac{dy}{dx} = \frac{dy/dt}{dx/dt}$,连续两次利用 diff (f) 命令就可以求出结果 .

3) 隐函数求导

方程 F (x,y) = 0 所确定的隐函数 y = y (x) ,其导数为 $\frac{dy}{dx} = -\frac{F_x}{F_y}$;方程 F (x,y,z) = 0 确定的隐函数 z = z (x,y) ,其导数为 $\frac{\partial z}{\partial x} = -\frac{F_x}{F_z}, \frac{\partial z}{\partial y} = -\frac{F_y}{F_z}$. 在 Matlab 中按照上述公式,分别求出函数的偏导数再相除就可以得到隐函数的导数 .

3. 积分运算

(1) 一元函数的不定积分

命令形式 1:int (f)

功能:求函数 f 对默认变量的不定积分,用于函数中只有一个变量的情况 .

命令形式 2:int (f,v)

功能:求符号函数 f 对变量 v 的不定积分 .

例 6. 计算 $\int \frac{1}{\sin^2 x \cos^2 x} dx$.

解 Matlab 命令为

```
syms x ;
y = 1 / (sin (x) 2 * cos (x) 2) ;
int (y) ;
ans = 1 / (sin (x) * cos (x)) - 2 * cos (x) / sin (x) .
pretty (int (y) )     % 把 int (y) 化简为常用的数学形式的表达式. 结果为
```

$$\frac{1}{\sin(x)\cos(x)} - \frac{2\cos x}{\sin x}.$$

例 7. 求 $\int \frac{x}{1 + z^2} dz$.

解 Matlab 的命令为

```
syms x z ;B = int (x/ (1 + z 2) ,z) ;
```

B = x * atan (z)

(2) 一元函数的定积分

命令形式：int (f,x,a,b)

功能：用微积分基本公式计算定积分 $\int_a^b f(x)\,dx$.

例 8. 求 $\int_{\frac{1}{2}}^{2}\left(1+x-\frac{1}{x}\right)^{\frac{1}{2}}dx$.

解　Matlab 的命令为

```
syms x ;
t = 1 + x - 1/x ; y = exp (x + 1/x) ;
f = t * y ;
int (f,x,1/2,2) ;
ans = 3/2 * exp (5/2) .
```

(3) 多重积分运算

多重积分运算的命令与功能如表 1.2 所示.

表 1.2

命　　令	功　　能
int (int (f,y) ,x)	计算不定积分 $\int dx\int f(x,y)\,dy$
int (int (f,y,c,d) ,x,a,b)	计算定积分 $\int_a^b dx\int_c^d f(x,y)\,dy$

注意　对于三重积分的运算和二重积分的运算形式一致.

例 9. 计算 $A=\int_0^1\int_x^{x+1}(x^2+y^2+1)\,dxdy$.

解　Matlab 的命令为

```
syms x y ;
    A = int (int (x 2 + y 2 + 1,y,x,x + 1) ,x,0,1) % 求二重积分
                                A = 5/2.
```

4. 函数的 Taylor 展开

命令形式 1：taylor (f)

功能：将函数 f 展开成默认变量的 6 阶麦克劳林 (Maclaurin) 公式.

命令形式 2：taylor (f, n)

功能:将函数 f 展开成默认变量的 n 阶麦克劳林(Maclaurin)公式.

命令形式 3:taylor(f, n, v, a)

功能:将函数 f(v)在 v = a 处展开成 n 阶 Taylor 公式.

例 10. 将函数 $f(x) = x\arctan x - \ln\sqrt{1+x^2}$ 展开为 x 的 6 阶麦克劳林(Maclaurin)公式.

解 Matlab 命令为

```
syms x;
f = x * atan(x) - log(sqrt(1 + x 2));
taylor(f);
ans = 1/2 * x 2 - 1/12 * x 4.
```

例 11. 将函数 $f(x) = \frac{1}{x^2}$ 展开为关于(x - 2)的最高次为 4 的幂级数.

解 Matlab 命令为

```
syms x;
f = 1/x 2 ;
taylor(f, 4, x, 2);pretty(taylor(f, 4, x, 2));
```

$$\frac{3}{4} - \frac{1}{4}(x-2) + \frac{3}{16}(x-2)^2 - \frac{1}{8}(x-2)^3.$$

1.1.2 数值微分与数值积分在 Matlab 中的实现

1. 数值微分

数值微分是用离散的方法近似地计算函数 y = f(x)在某点 x = a 处的导数值,通常仅当函数以离散数值形式给出时才有这种必要. 根据导数的定义,可以用差商近似导数,有

$$f'(a) \approx \frac{f(a+h) - f(a)}{h} \tag{1.1}$$

$$f'(a) \approx \frac{f(a) - f(a-h)}{h} \tag{1.2}$$

其中 h > 0 为小的增量. 式(1.1)和式(1.2)分别称为前差公式和后差公式,如果将两者平均一下,得

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h} \tag{1.3}$$

式(1.3)称为中心差商.

Matlab 中可以用 diff 命令实现前差公式(式(1.1))的数值微分:

命令形式:diff(x)

功能：输入 x = [x(1),x(2),…,x(n)] 是 n 维数组，输出为 n - 1 维数组 [x(2) - x(1),x(3) - x(2),…,x(n) - x(n - 1)].

2. 用数值方法计算定积分

用定积分的符号解法求定积分有时会失效，此时可以用数值方法来计算定积分的值. Matlab 提供了如下一些计算定积分值的数值方法.

(1) 复合梯形公式

复合梯形公式用小梯形面积代替小曲边梯形的面积，然后求和以获得定积分的近似值.

命令形式：trapz(x,y)

功能：用复合梯形公式计算定积分，变量 x 是积分变量在被积区间上的分点向量，y 为被积函数在 x 处对应的函数值向量.

(2) 复合辛普生公式

该方法用抛物线代替小曲边梯形的曲边计算小曲边梯形的面积，然后求和以获得定积分的近似值.

命令形式 1：quad('fun',a,b,tol,trace)

命令形式 2：quadl('fun',a,b,tol,trace)

式中 fun 是被积函数表达式字符串或者是 M 函数文件名，a,b 表示积分的下限与上限，tol 代表精度，可以缺省；缺省时，tol = 0.001，trace = 1 时用图形展示积分过程，trace = 0 时无图形，默认值为 0. 命令形式 2 比命令形式 1 精度高.

注意：

① Matlab 早期版本中用的命令形式是 quad 和 quad8，现在比较新的版本中，命令 quad8 已逐渐被 quadl 代替.

② 被积函数 fun，可以是字符串、内联函数或 M 函数文件名. 在命令中，前三个输入量是必须的，后面的输入变量可以缺省.

例 12. 用复合梯形公式和复合辛普生公式求 $\int_2^5 \frac{\ln x}{x^2}dx$ 的积分值.

解　Matlab 命令为

```
syms x ;
x = 2:0.1:5 ;
y = log(x)./(x.2) ;
t = trapz(x,y) ;
ff = inline('log(x)./(x. 2)','x') ;
q = quad(ff,2,5) ;
disp([blanks(3)'梯形法求积分'blanks(3)'辛普生法求积分'])，[t,d] 梯形
```

法求积分

梯形法求积分	辛普生法求积分
0.3247	0.3247

说明：inline 表示内联函数.

例 13. 设 $s(x)=\int_0^x y(t)\,dt$，其中 $y(t)=e^{-0.8t|\sin t|}$，求 $s(10)$.

解 建立 M 命令文件

```
clf
dt = 0.1 ;t = 0 :dt :10 ;
y = exp (- 0.8 * t. * abs (sin (t))) ;
st10 = trapz (t,y) ;
ff = inline ('exp (- 0.8 * t. * abs (sin (t)))','t') ;
q = quad (ff,0,10) ;
q8 = quad8 (ff,0,10) ;
disp ([blank (6),'trapz',blanks (5),'quad',
blanks (5),'quad8'])
disp ([st10,q,q8])
```

运行结果得

trapz	quad	quad8
2.6576	2.6597	2.6597

(3) 用数值方法计算二重积分

命令形式：dblquad ('fun', xmin, xmax, ymin, ymax)

功能：计算二重积分 $\int_{xmin}^{xmax}dx\int_{ymin}^{ymax}f(x,y)\,dy$，其中 xmin,,xmax,ymin,ymax 表示积分限.

注：用数值方法计算三重积分的命令为

triplequad (fun,xmin,xmax,ymin,ymax,zmin,zmax)

其中被积函数 fun 可以是字符串、内联函数或 M 函数文件名.

例 14. 计算 $\iint_D xy\,dx\,dy$，其中 D 是由 $y=1$，$x=4$，$x=0$，$y=0$ 所围成的区域.

解 (1) 建立 M 函数文件

```
function z = ff (x,y)
z = x * y
```

(2) Matlab 命令为

dblquad (ff, 0.4 ,0.1)

ans = 4.

例 15. **计算** $\int_0^1 dy\int_0^1 (x^2 + y)\,dx$.

解　Matlab **命令**

```
ff = inline('x 2 + y','x','y')
dblquad(ff,0,1,0,1)
ans = 0.8333.
```

内积分限为函数的二重积分可以编写下列 M **函数文件,利用** quad8 **进行两次积分求值. 程序代码如下:**

```
[double_int.m]
function ss = double_int(fun,innlow,innhi,outlow,outhi)
y1 = outlow;y2 = outhi;x1 = innlow;x2 = innhi;f_p = fun;
ss = quad8('G_yi',y1,y2,[],[],x1,x2,f_p);

[G_yi.m]
function f = G_yi(y,x1,x2,f_p)
y = y(:);n = length(y);
if ischar(x1) == 1
xx1 = feval(x1,y);
else
xx1 = x1*ones(size(y));
end
if ischar(x2) == 1
xx2 = feval(x2,y);
else
xx2 = x2*ones(size(y));
end

for i = 1:n
f(i) = quad8(f_p,xx1(i),xx2(I),[],[],y(i));
end

f = f(:).
```

例 16. **计算** $I = \int_1^4 \left[\int_{\sqrt{y}}^2 (x^2 + y^2)\, dx \right] dy.$

解 **根据上述** M **函数文件** double_int. m **和** G_yi. m **后,再编写内积分区间上、下限的** M **函数文件** x_low. m **并保存**.

[x_low. m]

```
function    f = x_low (y)
f = sqrt (y)
```

被积函数用内联函数表示时,运行以下指令,即得结果

```
ff = inline ('x . 2 + y . 2', 'x', 'y')
ss = double _int (ff, 'x _ low', 2, 1, 4)
```

运行结果为

```
ss = 9. 5810.
```

1. 1. 3 线性方程和非线性方程在 Matlab 中的各种求解方法

1. **求多项式方程的根**

n **次多项式的一般形式为**

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = 0 \tag{1.4}$$

式 (1. 4) **中的** $a_0, a_1, a_2, \cdots, a_n$ **为常数**. **理论上已经知道,**n **次多项式方程有** n **个根,且对于** $n \leqslant 4$ **的多项式方程,其根可以用公式表示,而次数** $n > 5$ **的多项式方程,其根一般不能用解析式表示**. **因此,在** Matlab **中,对于次数** $n \leqslant 4$ **的多项式,可以快速求出所有根的准确形式,而对于** $n > 4$ **的多项式方程,不一定能求出所有根的准确形式,但可以求出所有根的近似形式**.

求多项式方程的根有如下一些命令:

命令形式 1:roots (p)

功能:求多项式 p **的所有根,注意这里的** p **只能是多项式方程**.

命令形式 2:solve (s)

功能:对一个方程 s **的默认变量求解,这里的方程** s **可以是多项式方程,也可以是一般的任意方程**.

命令形式 3:solve (s,v)

功能:对一个方程指定的变量 v **求解**.

命令形式 4:solve (s1,s2,…,sn,v1,v2,…,vn)

功能:对 n **个方程的指定变量** v1,v2,…,vn **求解**.

命令形式 5:[x1,x2,x3,…,xn] = solve (s1,s2,…,sn,v1,v2,…,vn)

功能:将 n **个方程的指定变量** v1,v2,…,vn **求解的结果赋给** x1,x2,x3,…,

xn.

例 17. **求方程** $x^3 - 4x^2 + 9x - 10 = 0$ **的所有根**.

解　Matlab **的命令为**

```
p = [1  -4  9  -10];
r = roots(p);
r =
    1.0000 + 2.0000i
    1.0000 - 2.0000i
    2.0000
```

或

```
s1 = sym('x3 - 4*x2 + 9*x - 10');
solve(s1).
```

例 18. **求方程** $x^2 - ax - 4b = 0$ **的所有根,其中** a,b **为常数**.

解　Matlab **的命令为**

```
s1 = sym('x2 - a*x - 4*b = 0');
solve(s1,'x');
ams = [1/2*a + 1/2*(a2 + 16*b)(1/2)]
      [1/2*a - 1/2*(a2 + 16*b)2(1/2)].
```

2. **求超越方程的根**

超越方程是指除了多项式方程之外的函数方程,这类方程通常不容易求得全部根和确切根,而往往是采用数值方法求近似根,对于某些方程组可能连近似根也求不出,因为非线性方程组的求解还有很多问题没有解决. 在 Matlab **中,求超越方程的根可以用** solve **命令**.

例 19. **求方程** psinx = r **的根,其中** p,r **为常数**.

解　Matlab **的命令为**

```
ff = syms('p*sin(x) = r');
solve(ff,'x');
ans = asin(r/p).
```

solve **命令还可以用来求解方程组**.

例 20. **求方程组** $\begin{cases} x + y = 1 \\ x - 11y = 5 \end{cases}$ **的根**.

解　Matlab **的命令为**

```
[x,y] = solve('x + y = 1','x - 11*y = 5','x','y');
```

x = 4 /3 ;

y = - 1 /3.

求一元函数超越方程的数值解可以用 fzero 命令：

命令形式：z = fzero ('fname' ,x0 ,tol ,trace)

功能：求一元函数的零点．其中 fname 是待求零点的函数文件名，或是待求方程．x0 是预定待搜索零点的大致位置．tol 是精度，可以默认为 eps，trace 表示是否显示迭代步骤，可以默认为不显示．

求多元函数的数值解可以使用命令 fsolve：

命令形式：x = fsolve (fun , x0)

功能：求多元函数 fun 在点 x0 处的零点，其中 x0 为一向量．

例 21. 求方程 $x = (\cos x)^2$ 在 1 附近的根．

解　Matlab 命令为

x = fzero ('x - (cos (x)) 2' ,1)

x = 0. 6417

或采用 M 函数文件的形式：

function　y = ff (t)

y = t - (cos (t)) . 2 ;

x = fzero ('ff' ,1) ;

x = 0. 6417.

例 22. 求方程组 $\begin{cases} x = y^2 \\ y = \cos x \end{cases}$ 在 (1 ,2) 附近的根．

解　Matlab 命令为

fun = ' [x - y 2 ,y - cos (x)] '

f = fsolve (fun , [1 ,2])

f = 0. 6417 ,0. 8011.

1. 1. 4　Matlab 中求和及求极值方法

1. 求和

(1) 向量或矩阵的求和

命令形式：sum (x)

功能：求向量 x 的和或者是矩阵每一列向量的和．

例 23. a = 1: 5 ; A = [1 2 3 ; 2 3 4 ; 7 8 9] ;

解　sum (a)

ans = 15

```
sum (A)
ans = 10,13,16.
```

(2) 级数求和

级数求和也是微积分中比较常见的运算,在 Matlab 中,采用函数 symsum 来对符号表达式求和.

命令形式:symsum (s,v,a,b)

功能:对表达式 s 的符号变量 v 从 v = a 到 v = b 进行求和.

例 24. 求 $\sum_{k=0}^{n-1} k^3$.

解　Matlab 命令为

```
syms k n;
f = k 3;symsum (f,k,0,n - 1)
ans = 1/4*n 4 - 1/2*n 3 + 1/4*n 2
```

例 25. 求 $\sum_{k=0}^{+} \frac{x^k}{k!}$

解　Matlab 的命令为

```
syms x k ;
symsum (x k/sym ('k!'), k, 0, inf)
ans = exp (x)
```

2. 求函数的极值点

(1) 求一元函数的极值问题

命令形式 1:fmin (fun,x1,x2)

功能:在区间 [x1,x2] 内求函数 fun 的极小值点.

命令形式 2:fminbnd (fun,x1,x2)

功能:在区间 [x1,x2] 内求函数 fun 的极小值点.

两个函数的功能相同,命令 1 在 Matlab 早期版本中使用. 在命令 fmin 中函数 fun 的创建一般有三种方法:(1) 用字符串创建;(2) 用内联函数命令 inline 创建;(3) 建立一个 M 函数命令文件. 下面例子中对所述三种方法分别都有说明.

例 26. 求 $f(x) = x + 3*(x^2 + \cos x)$ 在区间 [- 1,1] 内的最小值.

解　(1) 建立 M 函数命令文件

```
functiony = gg2 (x)
y = 3*2. (5*x) .*(x . 2 + cos (x)) - 40.
```

(2) 建立 M 函数命令文件

```
clf
```

```
x = - 2 : 0.1 : 2
xmin = fmin ('gg2', - 1,1) .
```

(3) 运行 M 函数命令文件得 xmin = - 2.7756e - 017

或 ff = 'x + 3 * (x 2 + cos (x))'

```
xmin = fmin (ff, - 1,1)
xmin = - 2.7756e - 017.
```

(2) 求多元函数极值问题

求多元函数极小值常用的方法有单纯形法和拟牛顿法,主要命令有下面两个:

命令形式 1:fminsearch (fun,x0)

功能:用单纯形法求多元函数 fun 在 x0 附近的极值点.

命令形式 2:fminunc (fun,x0)

功能:用拟牛顿法求多元函数 fun 在 x0 附近的极值点.

例 27. 求函数 $f(x,y) = 100(y - x^2)^2 + (1 - x)^2$ 的极小值点.

解 Matlab 的命令为

```
ff = inline ('100 * (y - x 2) 2 + (1 - x) 2')
x0 = [- 1.2, 1]
x = fminunc (ff,x0)
x = 1.0,1.000,2.0.
```

例 28. 求函数 $f(x,y,z) = x^4 + \sin y - \cos z$ 在点 (0,5,4) 附近的极小值.

解

```
syms x y z
x0 = [0,5,4]; ff = inline ('x 4 + sin (y) - cos (z) ')
[xmin, fval] = fminsearch (ff,x0)
x = - 0.0021, 4.7124, 6.2832
fval = - 2.000.
```

1.1.5 函数插值与曲线的拟合

1. 函数插值

插值法是由实验或测量的方法得到所求函数 $y = f(x)$ 在互异点 $x_0, x_1, \cdots, x_n$ 处的值 $y_0, y_1, \cdots, y_n$,构造一个简单函数 $\varphi(x)$ 作为函数 $y = f(x)$ 的近似表达式:$y = f(x) \approx \varphi(x)$,使

$$\varphi(x_0) = y_0, \varphi(x_1) = y_1, \cdots, \varphi(x_n) = y_n$$

$\varphi(x)$ 称为插值函数,该函数常取多项式或分段多项式.

(1) 一维插值

一维插值的命令格式为：

Y1 = interp1 (x,y,X1,'method')

功能：根据已知的数据 (x,y),用 method 方法进行插值,然后计算 X1 对应的函数值 Y1.

说明：x,y 是已知的数据向量,其中 x 应以升序或降序来排；X1 是插值点的自变量坐标向量；'method'是用来选择插值算法的,它可以取：'linear'(线性插值)、'cubic'(三次多项式插值)、'nearst'(最近插值)、'spline'(三次样条插值).

例 29. 对 $y = \frac{1}{(1 + x^2)}$, $-5 \leqslant x \leqslant 5$,用 11 个节点作三种插值,并比较其结果.

解　(1) 在 Matlab 中程序如下：

```
x0 = -5:0.5:5;  y0 = 1./(1+x0.2);
x = -5:0.2:5;
y1 = interp1(x0,y0,x,'linear');y2 = interp1(x0,y0,x,'spline');
y3 = interp1(x0,y0,x,'nearst');
subplot(2,2,1), plot(x0,y0,'r-p');title('y =1/(1+x2)');
subplot(2,2,2), plot(x0,y0,'r-',x,y1), title('linear');
subplot(2,2,3), plot(x0,y0,'r-',x,y2), title('spline');
subplot(2,2,4), plot(x0,y0,'r-',x,y3), title('nearst');
```

运行结果如图 1.1 所示.

(2) 二维插值

二维插值的命令格式为：

Z1 = interp2 (x,y,z,X1,Y1,'method')

功能：根据已知的数据 (x,y,z),用 method 方法进行插值,然后计算 (X1,Y1) 对应的值 Z1.

说明：x,y 是已知的原始数据,z 是函数值；X1,Y1 是插值点的自变量坐标向量；'method'是用来选择插值算法的,它可以取："linear'(双线性插值)、'cubic'(三次插值)、'nearst'(最近插值).

例 30. 利用二维插值对 peak 函数进行插值.

解　Matlab 的命令为

```
[x,y] = meshgrid(-3:0.25:3);
z = peaks(x,y);
[x1,y1] = meshgrid(-3:0.125:3);
z1 = interp2(x,y,z,x1,y1);
```

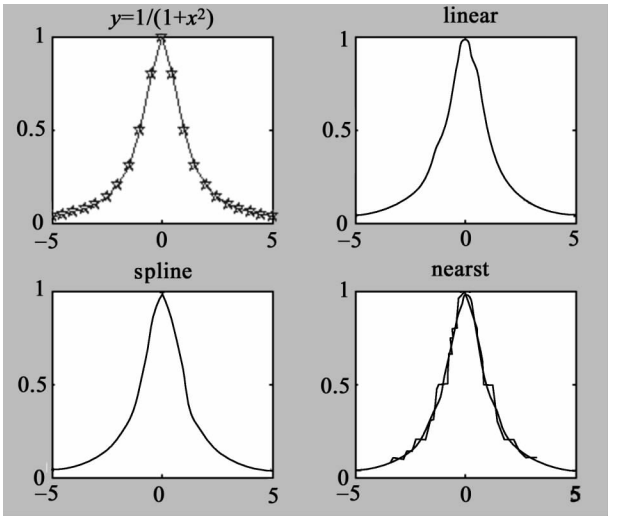


图 1.1　一维插值函数的图像

mesh (x1 ,y1 ,z1) ;

运行结果如图 1.2 所示 .

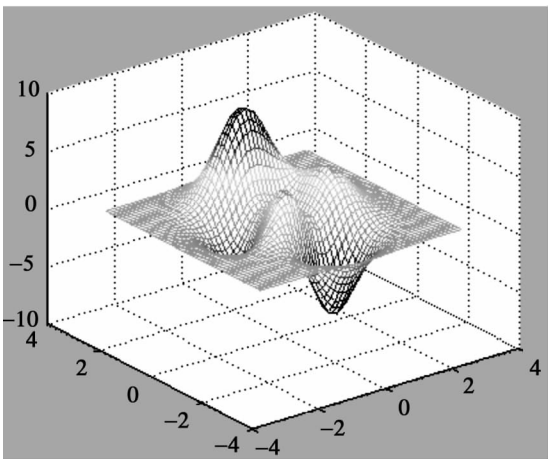


图 1.2　二维插值函数的图像

2. 曲线的拟合

(1) 多项式拟合

多项式拟合的命令格式为：

[p,s] = polyfit (x,y,n)

功能：对于已知的数据组 x、y 进行多项式拟合，拟合的多项式的阶数为 n，其中 p 为多项式的系数矩阵，s 为预测误差估计值的矩阵.

例 31.　x 取 0 至 1 之间的数，间隔为 0.1；y 为 2.3，2.5，2.1，2.5，3.2，3.6，3.0，3.1，4.1，5.1，3.8. 分别用二次、三次和七次拟合曲线来拟合这组数据，试观察三组拟合曲线哪个效果最好.

解　Matlab 命令如下：

```
clf;
x = 0:0.1:1;
y = [2.3,2.5,2.1,2.5,3.2,3.6,3.0,3.1,4.1,5.1,3.8];
p2 = polyfit (x,y,2);
p3 = polyfit (x,y,3);
p7 = polyfit (x,y,7).
disp ('二次拟合曲线'), p2
disp ('三次拟合曲线'), p3
disp ('七次拟合曲线'), p7
x1 = 0:0.1:1;
y2 = polyval (p2,x1);
y3 = polyval (p3,x1);
y7 = polyval (p7,x1);
plot (x,y,'rp',x1,y2,'--',x1,y3,'k-.',x1,y7).
legend ('拟合点','二次拟合','三次拟合','七次拟合')
```

运行结果如图 1.3 所示.

二次拟合曲线

p2 = 0.6410，1.6226，2.1734

三次拟合曲线

p3 = -4.9728，8.1002，-1.2218，2.3524

七次拟合曲线

p7 = 1.0e+003 *

Columns 1 through 6

1.0563，-4.5980，7.6095，-6.0779，2.4241，-0.4399

Columns 7 through 8

0.0275，0.0023.

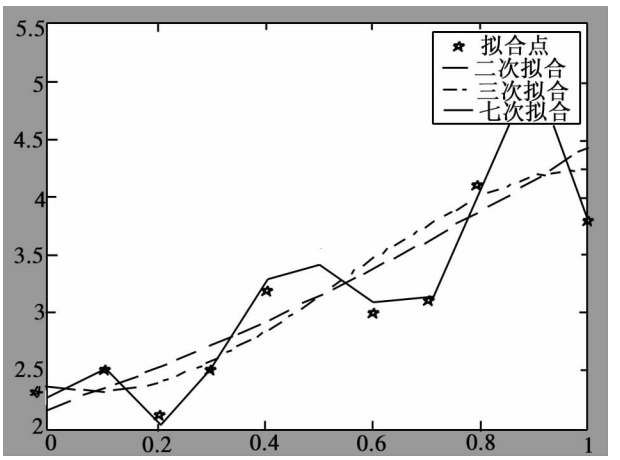


图 1.3　多项式拟合函数的图像

(2) 非线性最小二乘拟合

命令形式：leastsq ('f',x0)

功能：作非线性最小二乘拟合，其中 f 是 M 函数文件.

例 32. **用表** 1.3 **中的一组数据拟合** $c(t) = re^{(-kt)}$ **中系数** r,k,**并绘出图像**.

表 1.3

t	0.25	0.5	1	1.5	2	3	4	6	8
c	19.21	18.15	15.36	14.10	12.98	9.32	7.45	5.24	3.01

解　(1) 建立函数文件 ct.m

```
function    y = ct (x)
t = [0.25    0.5    1    1.5    2    3    4    6    8]
c = [19.21   18.15   15.36   14.10   12.98   9.32   7.45   5.24   3.01]
y = c - x (1) * exp (- x (2) * t) .
```

(2) 建立命令文件

```
x0 = [10,0.5]
x = leastsq ('ct',x0)
tt = 0:0.2:8
yy = x (1) . * exp (- x (2) . * tt)
```

plot (tt,yy,' - rp')

运行结果如图 1.4 所示.

x = 20.2413, 0.2420

即 r = 20.2413, k = 0.2410.

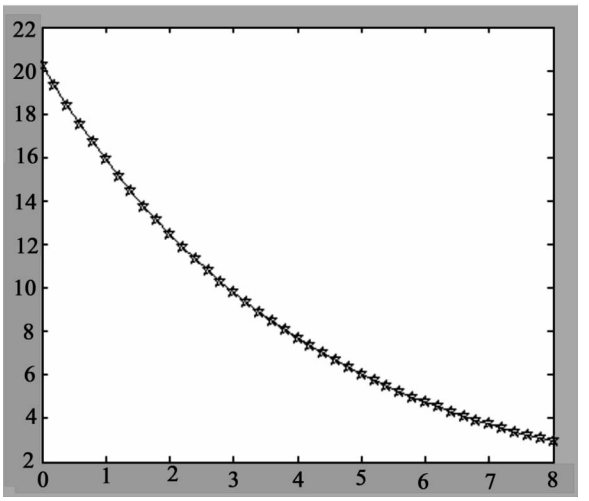


图 1.4　最小二乘拟合的图像

习　题　1

1. 在 Matlab 中求下列函数的导数

(1) 求 $x^{10}+10^{x}+\log_{x}10$ 的一阶导数;

(2) 求 $\ln(2-x^{2})$ 的二阶导数;

(3) 求 $z=e^{2x}(x+y^{2}+2y)$ 分别对 x 和 y 的一阶偏导数;

(4) 求 $z=\cos^{2}(2x+y)$ 分别对 x 和 y 的二阶偏导数;

(5) 求 $z=\ln(x^{2}+y^{2}+v^{2})$ 的全微分;

(6) 已知 $y=\sqrt{x\sin\sqrt{3^{e^{x}-\ln x}}}$,求 y''.

2. 在 Matlab 中求下列极限

(1) $\lim\limits_{x\to 0}\dfrac{2\sin x}{x^{3}+3x}$;　　(2) $\lim\limits_{x\to 1}\dfrac{\sqrt{5x-4}-\sqrt{x}}{x-1}$;

(3) $\lim\limits_{x\to}\left(1-\dfrac{2}{x}\right)^{2x}$;　　(4) $\lim\limits_{n\to}\sqrt{n+\sqrt{n}}-\sqrt{n}$;

(5) $\lim\limits_{x\to}\left(\sin\frac{1}{x}+\cos\frac{1}{x}\right)^{x}$.

3. 在 Matlab 中用符号表达式计算下列定积分的值

(1) $\int_0^1 e^{-x^2}dx$；　　(2) $\int_0^2 \frac{t^2+4}{t^2+2}dt$；

(3) $\int_0^{\pi}\sqrt{\sin^3x-\sin^5x}dx$；　　(4) $\int_0^{\frac{\pi}{6}}\frac{1}{1+8\sin^2x}dx$.

4. 在 Matlab 中计算下列不定积分

(1) $\int\frac{x^2}{\sqrt{2-x^2}}dx$；　　(2) $\int\sin2x\sin3xdx$；

(3) $\int\frac{dx}{x(\sqrt{\ln x+a}+\sqrt{\ln x+b})}$，其中 $a\neq b$；

(4) $\int\begin{pmatrix}\sin x & x^3\\ xe^x & \tan x\end{pmatrix}dx$.

5. 在 Matlab 中利用数值积分求下列表达式的值

(1) $\int_6^e x^2\ln xdx$；　　(2) $\int_0^{\pi}\sqrt{\sin x-\sin^3x}dx$；

(3) $\int_0^{\pi}\int_0^{\sin y}xdxdy$；　　(4) $\int_{-1}^{2}dx\int_{x^2}^{x+2}xydy$；

(5) $\iint\limits_D\frac{y^2}{x^2}dxdy$，其中区域 $D:\frac{1}{2}\leqslant x\leqslant 2,1\leqslant y\leqslant 2$；

(6) 计算二重积分 $\iint\limits_D\frac{1}{1+x^2+y^2}dxdy$，其中区域 $D:x^2+y^2\leqslant 1$.

6. 计算下列二重积分的值

(1) $\int_0^{\pi}d\theta\int_0^1\sqrt{1+r\sin(\theta)}dr$；

(2) $\iint\limits_D e^{-(x^2+y^2)}d\sigma$，其中 D 为圆周 $D:x^2+y^2=1$ 所围成的区域.

7. 在 Matlab 中求解下列微分方程

(1) $(1+x)\frac{d^2y}{dx^2}+\frac{dy}{dx}=0$；　　(2) $\frac{d^2y}{dx^2}=\frac{dy}{dx}-3x^2+6x$；

(3) $y''-y=x-2,y(0)=2,y(1)=1$；

(4) $(1+x)y''=2y-4,y(0)=0,y(1)-2y'(1)=0$.

8. 求下列函数的 6 阶 Taylor 展开式

(1) $f(x)=\sin x$；　　(2) $f(x)=\frac{\sin x}{x}$；

(3) $f(x) = e^{2x}$；　　(4) $f(x) = xe^{x}$.

9. 在 Matlab 中求方程 $x^2 + bx + c = 0$ 的解.

10. 求解多项式方程 $x^9 + x^8 + 1 = 0$.

11. 试求方程 $x^6 - x^2 + 2x - 3 = 0$ 的所有根.

12. 试求方程 $\sin x - \ln(x + 0.1) = 0$ 在 1 附近的近似根.

13. 求下列方程组的解

(1) 试求方程组 $\begin{cases} x + 3y = 0 \\ x^2 + y^2 = 1 \end{cases}$ 的所有根；

(2) 试求方程组 $\begin{cases} x = y^2 \\ y = \cos x \end{cases}$ 在 (1,2) 附近的根.(0.6417,0.8011)；

(3) 试求方程组 $\begin{cases} \sin x + y^2 + \ln z - 7 = 0 \\ 3x + 2^y - z^3 + 1 = 0 \\ x + y + z - 5 = 0 \end{cases}$ 在 (1,1,1) 附近的根.(0.5991,2.3959,2.0050).

14. 已知数据 x = [1.2,1.4,1.8,2.1,2.4,2.6,3.0,3.3],y = [4.85,5.2,5.6,6.2,6.5,7.0,7.5,8.0],试求对 x 与 y 进行一次、二次拟合的拟合系数.

15. 试分别用 2,3,4,6 阶多项式拟合函数 $y = \cos x$,并做出拟合曲线与函数曲线 $y = \cos x$ 进行比较.

16. 已知 x = [0.1,0.8,1.3,1.9,2.5,3.1],y = [1.2,1.6,2.7,2.0,1.3,0.5],试用不同的插值方法求 x = 2 处的插值,并比较其结果.

17. 对函数 $y = \dfrac{1}{1 + x^2}$, $-5 \leqslant x \leqslant 5$,用 11 个节点作三种插值,并比较其结果.

§1.2　数学建模中常用的线性代数基础知识在 Matlab 中的实现

线性代数中常用的工具是矩阵、向量和行列式. 用这些工具可以简单且准确地把所要研究的问题描述出来,以提高研究的效率. 在数学建模中经常用到大量的线性代数知识,用 Matlab 可以很方便地求出矩阵的对角化、二次型化标准行以及方程组的解.

1.2.1　Matlab 中向量和矩阵的基本运算

设 A,B 为两个矩阵,Matlab 中矩阵的基本运算命令有：

1. **命令形式：**A + B，A - B

功能：矩阵的加法和减法．将两个同型矩阵相加或相减．

2. **命令形式：**k * A

功能：数乘．将数与矩阵做乘法，其中 k 为一个数．

3. **命令形式：**A * B

功能：矩阵的乘法．将两个矩阵进行相乘，A 的列数要与 B 的行数相同．

4. **命令形式：**A hB

功能：矩阵的左除．计算 $A^{-1}B$，A 必须为方阵．

5. **命令形式：**A / B

功能：矩阵的右除．计算 AB^{-1}，B 必须为方阵．

6. **命令形式：**det (A)

功能：求矩阵行列式．A 必须为方阵．

7. **命令形式：**inv (A) 或 (A^{-1})

功能：求方阵的逆．

8. **命令形式：**A n

功能：矩阵乘幂．计算 A^n.

9. **命令形式：**A' 或 transpose (A)

功能：计算矩阵的转置．

10. **命令形式：**rank (A)

功能：求矩阵的秩．

11. **命令形式：**rref (A)

功能：矩阵行变化化简．求矩阵 A 阶梯形的行最简形式．

例 33. **计算** $\begin{pmatrix} 1 & 3 & 7 \\ -3 & 9 & -1 \end{pmatrix} + \begin{pmatrix} 2 & 3 & 2 \\ -1 & 6 & -7 \end{pmatrix}$.

解 Matlab 的命令为

```
A = [1,3,7;-3,9,-1];B = [2,3,-2;-1,6,7];
A + B
ans =    3    6    5
        -4   15   -8
```

例 34. **求向量** (a,b,c) **与矩阵** $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ **的乘积**．

解 Matlab 的命令为

```
syms  a  b  c;
```

v = [a　b　c]；A1 = [1　2；3　4；5　6]；

v * A1

ans = [a + 3 * b + 5 * c,2 * a + 4 * b + 6 * c].

例 35.　**求矩阵** $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ **的转置、逆和行列式**.

解　syms　a　b　c　d；

A = [a　b；c　d]；

A'

ans = [a　c；b　d]；

Inv (A)

ans = d / (a * d - b * c)　- b / (a * d - b * c)

　　- c / (a * d - b * c)　a / (a * d - b * c)

% **矩阵的行列式**

det (A)

ans = a * d - b * c.

例 36. **求矩阵** $\begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix}$ **的** 6 **次幂**.

解　A = [1　3；2　1]；

A　6

ans = 847　1026

　　684　847.

例 37. **求矩阵** $\begin{pmatrix} 4 & 1 & 2 & 4 \\ 1 & 2 & 0 & 2 \\ 10 & 5 & 2 & 0 \\ 0 & 1 & 1 & 7 \end{pmatrix}$ **的秩与行最简形**.

解　A = [4 ,1 ,2 ,4；1 ,2 ,0 ,2；10, 5 ,2 ,0；0 ,1 ,1, 7]；

rank (A)

ans = 3；

rref (A)

$$\text{ans} = \begin{matrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0. \end{matrix}$$

1.2.2 矩阵的变换与分解及其在 Matlab 中的实现

在线性代数的计算中,经常要用到对矩阵进行分解变形. 在解线性方程组的时候常常要用到矩阵的 LU、QR 和 Cholesky 等分解,在 Matlab 中可以很方便地对矩阵进行分解和变形.

1. 矩阵的对角元素

(1) 函数 diag 可以将一个矩阵的对角线元素提取出来

命令形式:diag (A)

功能:由矩阵 A 的对角线元素得到一个列向量.

例 38. A = pascal (3).

解 A = 1 1 1
1 2 3
1 3 6

diag (A) % 提取对角线元素

ans = 1
2
6

(2) 用该函数来产生第 k 阶对角线上的元素

命令形式:diag (A,k)

其中:k = 0 表示主对角线;k > 0 表示在主对角线以上;k < 0 表示在主对角线以下.

例 39. a = [1 2 3].

解 A = diag (a,0)

A = 1 0 0
0 2 0
0 0 3

B = diag (a,1)

B = 0 1 0 0
0 0 2 0
0 0 0 3
0 0 0 0

C = diag (a, - 1)

C = 0 0 0 0
1 0 0 0
0 2 0 0
0 0 3 0

(3) 函数 blkdiag 可以根据输入创建一个分块对角矩阵

命令形式：S = blkdiag (a,b,c,…)

功能：根据输入的 a,b,c 等参数构造一个分块对角矩阵.

例 40. a = 1;b = [1 2;3 4];c = 9;d = [1 -1;2 -2;3 -3].

```
S = blkdiag (a,b,c,d) % 构造分块对角矩阵

S = 1   0   0   0   0   0
    0   1   2   0   0   0
    0   3   4   0   0   0
    0   0   0   9   0   0
    0   0   0   0   1  -1
    0   0   0   0   2  -2
    0   0   0   0   3  -3
```

2. 矩阵的分解

(1) 矩阵的奇异值分解

奇异值分解在矩阵分析中具有极为重要的作用. 在 Matlab 中,矩阵的奇异值分解有三种命令形式：

命令形式 1：s = svd (A)

功能：对矩阵 A 进行奇异值分解,s 表示奇异值分解后的对角矩阵.

命令形式 2：[U,S,V] = svd (A)

功能：对矩阵 A 进行普通奇异值分解,U,S,V 分别表示分解后的矩阵.

命令形式 3：[U,S,V] = svd (A,0)

功能：对矩阵 A 进行简洁形式的奇异值分解.

例 41. B = [1, 2, 8;3, -2, 5;4, 6, 1;0, 9, 2],

[U,S,V] = svd (B).

```
U =  -0.6592    0.4219    0.4154   -0.4636
     -0.1089    0.7549   -0.3308    0.5496
     -0.4415   -0.2067   -0.8213   -0.2963
     -0.5989   -0.4501    0.2084    0.6287

S = 14.2081    0         0
    0          7.6552    0
    0          0         0

V =  -0.1937    0.2447   -0.9501
     -0.8288   -0.5589    0.0250
     -0.5249    0.7923    0.3111
```

(2) 矩阵的 LU 分解

普通方阵的 LU 分解又叫 Gauss 消元法,可以把任意方阵分解成下三角矩阵的基本变换形式(行交换)和上三角矩阵的乘积. LU 分解的数学表达式为 A = LU,这里 L 为下三角矩阵的基本变换形式,U 为上三角矩阵.

LU 分解在 Matlab 中的命令函数为 lu:

命令形式:[L,U] = lu (A)

功能:对矩阵 A 进行 LU 分解.

(3) 矩阵的 QR 分解

QR 分解也叫长方阵的正交分解,QR 分解可以把任意长方阵分解为正交矩阵和上三角矩阵初等变换形式的乘积. QR 分解在 Matlab 中的命令函数为 qr:

命令形式:[Q,R] = qr (A).

(4) 矩阵的 Cholesky 分解

Cholesky 分解是针对对称、正定矩阵的分解. Cholesky 分解把矩阵分解成上三角矩阵和其转置的乘积,数学表达示为:A = C'C,这里 C 为上三角矩阵. 如果复数矩阵是 Hermite 正定矩阵,则也有 Cholesky 分解. Cholesky 分解在 Matlab 中的命令函数为 chol:

命令形式:C = chol (A).

1.2.3 Matlab 中矩阵特征值和特征向量的求解方法

1. 矩阵的特征值与特征向量

定义 设 A 是 n 阶方阵,λ 是一个数,如果存在非零的列向量 X,使得 $AX = \lambda X$ 成立,则称数 λ 为方阵 A 的特征值,非零列向量 X 称为 A 的特征向量.

用 Matlab 中的命令 eig 可以求出矩阵 A 的特征值和特征向量. 命令 eig 的使用方法有两种:

命令形式 1:d = eig (A)

功能:返回方阵 A 的全部特征值组成的特征向量 d.

命令形式 2:[V, D] = eig (A)

功能:返回矩阵 A 的特征值矩阵 D 与特征向量矩阵 V,满足 AV = VD.

求矩阵的特征多项式用命令 poly.

命令形式:poly (A)

功能:求矩阵 A 的特征多项式.

注意 这三条命令求出的都是数值解,并不是解析解.

例 42. **求矩阵** $A=\begin{pmatrix}4 & 6 & 0\\ -3 & -5 & 0\\ -3 & -6 & 1\end{pmatrix}$**的特征多项式,特征值和特征向量**.

解　Matlab **的命令为**

```
A = [4  6  0; -3  -5  0; -3  -6  1]
P = poly (A)
P = 1  0  -3  2
[V, D] = eig (A)
```

$$V=\begin{pmatrix}0 & 0.5774 & -0.8944\\ 0 & -0.5774 & 0.4472\\ 1.0000 & -0.5774 & 0\end{pmatrix}$$

$$D=\begin{pmatrix}1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & 1\end{pmatrix}$$

所以矩阵 A **的特征多项式为** $p=x^3-3x+2$**,矩阵** D **的对角元素分别为矩阵** A **的三个特征值,矩阵** V **的三个列向量表示与三个特征值相对应的三个特征向量**.

注意　**可以先用** poly **命令求出矩阵的特征多项式** P**,再用多项式求根命令** roots **来求矩阵的特征根,即矩阵** A **的特征根为** D = roots (poly (A)) .

2. **矩阵的相似对角化**

如果三阶矩阵 A **有三个线性无关的特征向量** $\alpha_1,\alpha_2,\alpha_3$**,对应的特征值为** $\lambda_1,\lambda_2,\lambda_3$**,设**

$$P=[\alpha_1\quad \alpha_2\quad \alpha_3]$$

$$\boldsymbol{\Lambda}=\begin{pmatrix}\lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3\end{pmatrix},$$

由矩阵特征值和特征向量的定义知,有下面等式成立

$$AP=P\boldsymbol{\Lambda},\ \text{也即}\ A=P\boldsymbol{\Lambda}P^{-1}$$

上式说明矩阵 A **与对角矩阵** $\boldsymbol{\Lambda}$ **相似**. **由上述分析可知,利用特征向量和特征值可以求得矩阵** A **的相似对角矩阵**.

对于一般情况的矩阵对角化有以下结果:

(1) **如果** n **阶矩阵** A **的** n **个特征向量线性无关,则** A **与对角阵相似**.

(2) **如果** n **阶矩阵** A **的** n **个特征值互不相等,则** A **与对角阵相似**.

(3) 如果矩阵 A 是实对称矩阵,则必有正交矩阵 P,使 $PAP^{-1}=\Lambda$,其中 Λ 是以 A 的 n 个特征值为对角元素的对角矩阵.

利用上述的这些结论,就可以用 Matlab 命令处理矩阵对角化问题.

例 43. 化方阵 $A=\begin{pmatrix}4 & 6 & 0\\ -3 & -5 & 0\\ -3 & -6 & 1\end{pmatrix}$为对角阵.

解　用命令 eig 求出方阵 A 的特征向量并判断其相关性,Matlab 中的命令为

```
A = [4 6 0; -3 -5 0; -3 -6 1]
[V d] = eig (A)
```

$$V=\begin{pmatrix}0 & 0.5774 & -0.8944\\ 0 & -0.5774 & 0.4472\\ 1.0000 & -0.5774 & 0\end{pmatrix}$$

$$D=\begin{pmatrix}1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & 1\end{pmatrix}$$

这里 1, -2, 1 是方阵的特征值,V 是对应的特征向量组成的相似变换矩阵,D 即为对应于 V 的方阵 A 的对角矩阵. 即有 $D=V^{-1}AV$.

例 44. 求一个正交变换将二次型

$$f=x_1^2+x_2^2+x_3^2+x_4^2+2x_1x_2-2x_1x_4-2x_2x_3+2x_3x_4$$

化为标准形.

解　二次形对应的矩阵为

$$A=\begin{pmatrix}1 & 1 & 0 & -1\\ 1 & 1 & -1 & 0\\ 0 & -1 & 1 & 1\\ -1 & 0 & 1 & 1\end{pmatrix},$$

把二次型化为标准形就相当于矩阵 A 对角化. 在 Matlab 中解题的命令为

```
A = [1 1 0 -1;1 1 -1 0;0 -1 1 1; -1 0 1 1]
[P, D] = eig (A)
```

$$P=\begin{pmatrix}-0.5000 & 0.7071 & 0.0000 & 0.5000\\ 0.5000 & -0.0000 & 0.7071 & 0.5000\\ 0.5000 & 0.7071 & 0.0000 & -0.5000\\ -0.5000 & 0 & 0.7071 & -0.5000\end{pmatrix}$$

$$D = \begin{pmatrix} -1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 3.0000 \end{pmatrix}$$

```
syms x1    x2    x3    x4;
X = [x1;x2;x3;x4];Y = P*X
```

所以,可以得到所求的正交变换为 $Y = PX$,而矩阵 A 的标准形为

$$f = y_1^2 + y_2^2 + 3y_3^2 - y_4^2.$$

1.2.4　范数、条件数和方程解的精度

在 Matlab 中,数与数之间的最小分辨率用命令 eps 表示,该命令使关系式 $1 + eps1 \neq 1$成立. 因此,表达任何数的相对误差都不可能小于该 eps. 在求解方程组的过程中,不可避免地使这个最小误差被放大. 比如,被一个很小的数除、两个十分相近的数相减等运算都会对计算结果的精度产生很坏的影响. 矩阵的范数和条件数对方程组求解过程中误差放大现象进行定量描述具有重要作用.

1. 矩阵的范数

这里只介绍矩阵的 2-范数. 矩阵的 2-范数确定了该矩阵相乘向量长度的最大可能的放大倍数,其数学定义为

$$\| A \| = \max_{\forall x}\left[\frac{Ax}{|x|}\right]$$

在 Matlab 中,计算该范数的命令是 norm:

命令形式:norm (A)

功能:求矩阵 A 的 2-范数.

2. 矩阵的条件数

描述方程 $Ax = b$ 的解对 b 中误差或不确定性灵敏度的最好度量是矩阵 A 的条件数,该条件数的数学定义是

$$K = \| A^{-1} \| \cdot \| A \|$$

条件数总是大于或等于 1. 正交矩阵的条件数是 1,奇异矩阵的条件数是 . 而病态矩阵的条件数也非常大. Matlab 计算条件数的命令是 cond.

命令形式:cond (A)

功能:求矩阵 A 的条件数.

3. 方程解的精度

利用条件数,方程解的相对误差可以由下列不等式进行估计

$$\frac{1}{K}\left(\frac{|\delta b|}{|b|}\right)\leqslant\left(\frac{|\delta x|}{|x|}\right)\leqslant K\left(\frac{|\delta b|}{|b|}\right)$$

由于 eps 是机器精度,所以可以用 K^{*} eps 的大小粗略地判断所得解是否可靠.

1.2.5 线性方程组的直接求解法在 Matlab 中的实现

1. 齐次线性方程组的直接求解方法

齐次线性方程组的矩阵形式为 $AX=0$,其中,A 是 $m\times n$ 阶矩阵;X 是未知向量. 显然,n 维0向量是齐次方程组的解,当齐次线性方程组有惟一解时,解就是0向量. 若方程组 $AX=0$ 只有惟一0解且 $m=n$,则此时系数矩阵 A 的行列式非 0.

当 $m=n$ 时,如果 A 的行列式为 0,即 $|A|=0$,则方程组 $AX=0$ 有非零解. 非零解由齐次方程组的基础解系表示.

齐次线性方程组的基础解系有如下特点:

(1) 如果矩阵 A 的秩为 $r(r<n)$,则基础解系含有 $n-r$ 个向量;

(2) 基础解系 $\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_{n-r}$ 是一组线性无关的向量组;

(3) 基础解系 $\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_{n-r}$ 中的每一个向量都是该齐次方程组的非零解;

(4) 齐次方程组 $AX=0$ 的任一解向量 X 均可由基础解系的线性组合表示.

由基础解系的特点可知,齐次方程组 $AX=0$ 的通解可以表示为基础解系的线性组合,即

$$X=k_1\varepsilon_1+k_2\varepsilon_2+\cdots+k_{n-r}\varepsilon_{n-r}$$

上式就是齐次方程组的解结构.

求解齐次线性方程组常用的 Matlab 命令为:

rank:求矩阵的秩.

rref:通过行变换把矩阵化为上三角形阵.

还有一个求矩阵的基础解系的命令:

命令形式:B = null (A,'r')

功能:求出矩阵 A 的基础解系,B 是由 A 的基础解系构成的矩阵.

例 45. 用基础解系表示齐次方程组

$$\begin{cases}x_1+x_2+x_3+x_4+x_5=0\\3x_1+2x_2+x_3+x_4-3x_5=0\\x_1+2x_3+2x_4+6x_5=0\\5x_1+4x_2+3x_3+3x_4-x_5=0\end{cases}$$

的通解.

解 Matlab 的命令为

A = [1 1 1 1 1;3 2 1 1 -3;0 1 2 2 6;5 4 3 3 -1]

Format rat %指定有理格式输出

B = null (A,'r')

$$B = \begin{pmatrix} 1 & 1 & 5 \\ -2 & -2 & -6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

syms k_1 k_2 k_3% 定义符号参数

x = k_1 * B (:,1) + k_2 * B (:,2) + k_3 * B (:,3)

x =

[$k_1 + k_2 + 5 * k_3$]

[$-2 * k_1 - 2 * k_2 - 6 * k_3$]

[k_1]

[k_2]

[k_3]

即方程组的通解为 $X = k_1\begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \\ 0 \end{pmatrix} + k_2\begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \\ 0 \end{pmatrix} + k_3\begin{pmatrix} 5 \\ 6 \\ 0 \\ 0 \\ 1 \end{pmatrix}$,其中 k_1, k_2, k_3 为任意实数.

2. 非齐次线性方程组的直接求解方法

记非齐次线性方程组为 AX = B,其中,A 是 m × n 阶矩阵;X 是未知向量;B 是 m 维已知向量 ($B \neq 0$). 非齐次方程组分为有解和无解两大类;当方程组有解时又分为有惟一解和有无穷解两类.

若 A 可逆,则方程组 AX = b 就只有惟一解 $X = A^{-1}B$. 当非齐次方程组有无穷解时,有如下定理:

定理　设非齐次线性方程组 AX = B 有无穷多组解,若已知一个特解为 $\boldsymbol{\eta}$,而对应的齐次方程组 AX = 0 的基础解系为 $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_{n-r}$,则非齐次线性方程组 AX = B 的通解为

$$X = \boldsymbol{\eta} + k_1\varepsilon_1 + k_2\varepsilon_2 + \cdots + k_{n-r}\varepsilon_{n-r}$$

上式说明,非齐次方程组的通解由非齐次线性方程组的一个特解和对应的齐次方程组的通解叠加而成,这就是非齐次线性方程组的解结构.

在 Matlab 中求解非齐次线性方程组 AX = B 主要有三种方法:求逆法、左除

与右除法和初等变换法.

(1) 求逆法

对于线性方程组 AX = B,如果系数矩阵 A 是可逆矩阵,即行列式 $|A| \neq 0$,则解由 inv(A) * b 获得.

例 46. 求方程组 $\begin{cases} 2x+3y=4 \\ x-y=1 \end{cases}$ 的解.

解 Matlab 命令为

A = [2, 3;1 -1];B = [4,1];

X = inv(A) * B' % 因为 B 是行向量,故要用转置运算以满足矩阵相乘的要求

X = 1.4000

0.4000

例 47. 求解矩阵方程 $\begin{pmatrix} 1 & 4 \\ -1 & 2 \end{pmatrix} X \begin{pmatrix} 2 & 0 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 0 & -1 \end{pmatrix}$.

解 Matlab 的命令为

A_1 = [1 4;-1 2];A_2 = [2,0;-1,1];B = [3,1;0,-1];

X = inv(A_1) * B * inv(A_2)

X = 1.0000 1.0000

0.2500 0

(2) 左除法与右除法

当 X 和 B 都是矩阵而不是向量时,如果系数矩阵 A 是可逆矩阵,即行列式 $|A| \neq 0$,线性方程组 XA = B 的解为 $X = A^{-1}B$,同理,对线性方程组 AX = B 的解为 $X = BA^{-1}$,因此由 Matlab 的左除和右除运算可以方便地求出解,即:

线性方程组 AX = B 的求解命令为:X = A\B

线性方程组 XA = B 的求解命令为:X = B/A

左除法和右除法比求逆法用的时间少,且精度比求逆法高.

例 48. 求解方程 $X\begin{pmatrix} 2 & 1 & -1 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 3 \\ 4 & 3 & 2 \end{pmatrix}$.

解 Matlab 命令为

A = [2 1 -1;2 1 0;1 -1 1];B = [1 -1 3;4 3 2]

X = B/A

X = -2 2 1

-8/3 5 -2/3

例 49. **设** $A=\begin{pmatrix}4 & 2 & 3\\ 1 & 1 & 0\\ -1 & 2 & 3\end{pmatrix}$,$AB=A+2B$,**求** B.

解　$AB=A+2B\Rightarrow(A-2E)B=A$,**所以在** Matlab **中的求解命令应为**

```
A = [4  3  2;1  1  0;-1  2  3]
A! = A - 2*eye(3)
B = A! hA
```

$$B=\begin{pmatrix}5/3 & -2/3 & -4/3\\ 2/3 & -5/3 & -4.3\\ -2/3 & 14/3 & 13/3\end{pmatrix}$$

(3) 初等变换法

在线性代数中用消元法求非齐次线性方程组的通解的具体过程为:首先用行初等变换化线性方程组为阶梯形方程组,把最后的恒等式“0 = 0”(如果出现)去掉.

如果剩下的方程当中最后一个等式是 0 等于一个非 0 的数,那么方程组无解;否则方程组有解.

在有解的情况下,如果阶梯形方程组中方程的个数 r 等于未知数的个数,那么方程组有惟一解;如果阶梯形方程组中方程的个数 r 小于未知数的个数,那么方程组就有无穷多个解.

在 Matlab 中,对于线性方程组 AX = B,利用命令 rref(A) 可以方便地求出线性方程组系数矩阵、增广矩阵阶梯形的最简形式,并写出线性方程组的通解.

注意如果方程组 AX = B 有无穷多个解,也可以通过命令 null(A,'r') 方便地求出齐次方程 AX = 0 的基础解系,然后再求出 AX = B 的通解.

例 50. **求非齐次方程组**$\begin{cases}4x_1+2x_2-x_3=2\\ 3x_1-x_2+2x_3=10\\ 11x_1+3x_2=8\end{cases}$**的解.**

解　Matlab **的命令为**

```
A = [4 2 -1;3 -1 2;11 3 0];b = [2;10;8]
B = ([A, b]);rref(B)
ans = 1      0      3/10     0
      0      1      -11/10   0
      0      0      0        1
```

显然,系数矩阵 A 的秩 R(A) = 2,而增广矩阵 B 的秩 R(B) = 3,所以方程组无解.

例 51. **求非齐次线性方程组**$\begin{cases}2x+3y+z=4\\x-2y+4z=-5\\3x+8y-2z=13\\4x-y+9z=-6\end{cases}$**的通解**.

解 Matlab **的命令为**

```
A = [2 3 1;1 -2 4;3 8 -2;4 -1 9]
B = [4;-5;13;-6]
B = ([A,B]);rref(B)
ans = 1    0    2    -1
      0    1    -1   2
      0    0    0    0
      0    0    0    0
```

即原方程组可以化为

$\begin{cases}x=-2z-1\\y=z+2\\z=z\end{cases}$**,取** $z=0$**,显然**$(-1,2,0)$**是原方程组的一个特解**. **而方程组的通解为**$\begin{pmatrix}x\\y\\z\end{pmatrix}=k\begin{pmatrix}-2\\1\\1\end{pmatrix}+\begin{pmatrix}-1\\2\\0\end{pmatrix}$.

上面所讲的线性方程组 AX = B **的求解方法中,系数矩阵** A **都是实数矩阵,下面介绍符号方程组的求解方法**.

3. **符号方程组的求解方法**

(1) **线性方程组** AX = B **的符号解**

命令形式:X = linesolve (A,B)

功能:**求出线性方程组的符号解,但要注意,该命令只给出特解**.

例 52. **求非齐次方程组**$\begin{pmatrix}a&0&0\\0&b&0\end{pmatrix}X=\begin{pmatrix}1\\c\end{pmatrix}$**的解**.

解 Matlab **的命令为**

```
syms a b c;
A = [a 0 0;0 b 0];B = [1;c]
X = linesolve(A,B)
X = [1/a]
    [1/b*c]
    [0]
```

说明：只给出了方程组的一个特解.

(2) 非线性方程组的符号解

命令形式：$[x_1, x_2, x_3, \cdots] = \text{solve}(e_1, e_2, e_3, \cdots)$

功能：该命令给出非线性方程组的符号解. 其中 $e_1, e_2, e_3, \cdots$ 是符号方程，$x_1, x_2, x_3, \cdots$ 是要求的未知量.

例 53. 解非线性方程组

$$\begin{cases} a + b + x = y \\ 2ax - by = -1 \\ 2(a + b) = x + y \\ ay + bx = 4 \end{cases}$$

解　Matlab 的命令为

```
e1 = sym('a + b + x = y');
e2 = sym('2*a*x - b*y = -1');
e3 = sym('2*(a + b) = x + y');
e4 = sym('a*y + b*x = 4');
[a,b,x,y] = solve(e1,e2,e3,e4,a,b,x,y)
```

a = [1, -1],　b = [1, -1],　x = [1, -1],　y = [3, -3].

1.2.6　线性方程组的迭代求解法在 Matlab 中的实现方法

线性方程组的直接求解法一般适合于矩阵 A 为低阶稠密矩阵的情况，而在工程技术和科学计算中常常会遇到大型稀疏矩阵的方程组，此时用直接法求解有很大的困难，而迭代法在计算和存储两方面都适合后一种情况. 常用的迭代法有雅可比迭代法和高斯-塞德尔迭代法.

1. 雅可比迭代法

将 A 分解为 $A = D - L - U$，其中

$$D = \text{diag}(a_{11}, a_{22}, \cdots, a_{nn}),$$

$$L = -\begin{pmatrix} 0 & & & \\ a_{21} & 0 & & \\ M & & 0 & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \quad U = -\begin{pmatrix} 0 & a_{12} & \cdots & 0 \\ & 0 & 0 & M \\ & & 0 & a_{n-1,n} \\ & & & 0 \end{pmatrix}$$

若对角阵 D 非奇异（即 $a_{ii} \neq 0, i = 1, 2, \cdots, n$），则 $AX = B$ 化为

$$X = D^{-1}(L + U)X + D^{-1}B \tag{1.5}$$

若记

$$B_1 = D^{-1}(L+U),\ F_1 = D^{-1}B \tag{1.6}$$

则原方程组的迭代形式可以写成

$$X^{(k+1)} = B_1 x^{(k)} + F_1,\ (k = 0,1,2,\cdots) \tag{1.7}$$

如果序列 $\{X^{(k)}\}$ 收敛于 X^*，则 X^* 必是方程 (1.5) 的解，因而也是 $AX = B$ 的解．迭代式 (1.6)、式 (1.7) 称为雅可比迭代式．

2. 高斯—塞德尔迭代

在雅可比迭代过程中，计算 $x_i^{(k+1)}$ 时 $x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$ 已经得到，不必再用 $x_1^k, x_2^k, \cdots, x_{i-1}^k$，即原来的迭代公式

$$DX^k = LX^{(k+1)} + UX^{(k)} + B$$

可以改进为

$$DX^{(k+1)} = LX^{(k+1)} + UX^{(k)} + B$$

在 D 非奇异的假设下，(D - L) 可逆，于是得到

$$B_2 = (D-L)^{-1}U,\ F_2 = (D-L)^{-1}B$$

$$X^{(k+1)} = B_2 X^{(k)} + F_2$$

上面两式称为高斯—塞德尔迭代公式，这两式与雅可比迭代相比有精度高、收敛速度快的优点．

注意　在用迭代法求解 $AX = B$ 时，可以根据 A 来判断迭代的收敛性，常用的判断收敛条件有：

(1) 若 A 是严格对角占优的，即 $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$，$i = 1,2,\cdots,n$，则雅可比迭代和高斯—塞德尔迭代均收敛．

(2) 若 A 对称正定，则高斯—塞德尔迭代收敛．

3. 迭代法在 Matlab 中的实现

在 Matlab 中用迭代法求解方程组，主要就是将 A 分解为 $A = D - L - U$，即提取 (产生) 对角阵和提取 (产生) 上、下三角阵．

(1) 提取 (产生) 对角阵

命令形式：V = diag (X)

功能：若输入向量 X，则输出 V 是以 X 为对角元素的对角阵；若输入矩阵 X，则输出 V 是 X 的对角元素构成的向量．

命令形式：V = diag (diag (X))

功能：输入矩阵 X，输出 V 是 X 的对角元素构成的对角阵．该命令可以用于迭代法中从 A 中提取 D.

(2) 提取 (产生) 上 (下) 三角阵

命令形式：V = triu (X)

功能：输入矩阵 X，输出 V 是 X 的上三角阵.

命令形式：V = tril (X)

功能：输入矩阵 X，输出 V 是 X 的下三角阵.

命令形式：V = triu (X,1)

功能：输入矩阵 X，输出 V 是 X 的上三角阵，但对角元素为 0. 该命令可以用于迭代法中从 A 中提取 U.

命令形式：V = tril (X, - 1)

功能：输入矩阵 X，输出 V 是 X 的下三角阵，但对角元素为 0. 该命令可以用于迭代法中从 A 中提取 L.

习　题　2

1. 在 Matlab 中分别利用矩阵的初等变换及函数 rank 求下列矩阵的秩

(1) $A=\begin{pmatrix}1 & -6 & 3 & 2\\ 3 & -5 & 4 & 0\\ -1 & -11 & 2 & 4\end{pmatrix}$；　　(2) $A=\begin{pmatrix}2 & 4 & 5 & -4\\ 3 & 6 & -1 & 6\\ -2 & 2 & 0 & -3\\ 5 & 16 & 9 & -5\end{pmatrix}$

2. 在 Matlab 中求下列矩阵的逆矩阵

(1) $A=\begin{pmatrix}1 & 1 & -3\\ 2 & 3 & -8\\ 1 & 1 & -4\end{pmatrix}$；　　(2) $A=\begin{pmatrix}1 & 2 & 1 & 0\\ 6 & 2 & 4 & 1\\ 0 & 2 & 1 & 0\\ 3 & 1 & 4 & 1\end{pmatrix}$.

3. 在 Matlab 中判断下列向量是否线性相关，并找出向量中的最大无关组

(1) $a_1=(1\quad 1\quad 3\quad 2)$，$a_2=(-1\quad 1\quad -1\quad 3)$，$a_3=(5\quad -2\quad 8\quad 9)$，$a_4=(-1\quad 3\quad 1\quad 7)$；

(2) $a_1=(1\quad 1\quad 2\quad 3)$，$a_2=(1\quad -1\quad 1\quad 1)$，$a_3=(1\quad 3\quad 3\quad 5)$，$a_4=(4\quad -2\quad 5\quad 6)$，$a_5=(-3\quad -1\quad -5\quad -7)$；

(3) $a_1=(1\quad 1\quad 0)$，$a_2=(0\quad 2\quad 0)$，$a_3=(0\quad 0\quad 3)$.

4. 在 Matlab 中判断下列方程组解的情况，若有多个解，试写出通解.

(1) $\begin{cases}x_1-2x_2+3x_3-x_4=1\\ 3x_1-x_2+5x_3-3x_4=2\\ 2x_1+x_2+2x_3-2x_4=3\end{cases}$；　　(2) $\begin{cases}x_1-x_2+4x_3-2x_4=0\\ x_1-x_2-x_3+2x_4=0\\ 3x_1+x_2+7x_3-2x_4=0\\ x_1-3x_2-12x_3+6x_4=0\end{cases}$；

(3) $\begin{cases} 2x_1+3x_2+x_3=4 \\ x_1-2x_2+4x_3=-5 \\ 3x_1+8x_2-2x_3=13 \\ 4x_1-x_2+9x_3=-6 \end{cases}$；　　(4) $\begin{cases} x_1-x_2+x_3-x_4=0 \\ x_1-x_2-x_3+x_4=0 \\ x_1-x_2-2x_3+2x_4=0 \end{cases}$.

5. 在 Matlab 中求下列矩阵的特征向量，并判断能否对角化，若能则将其对角化

(1) $A=\begin{pmatrix} -1 & 2 & 0 \\ -2 & 3 & 0 \\ 3 & 0 & 2 \end{pmatrix}$；　(2) $A=\begin{pmatrix} -2 & 1 & 1 \\ 0 & 2 & 0 \\ -4 & 1 & 3 \end{pmatrix}$；　(3) $A=\begin{pmatrix} 5 & 4 & -2 \\ 4 & 5 & 2 \\ -2 & 2 & 8 \end{pmatrix}$.

6. 将下列二次型化为标准型

(1) $f(x_1,x_2,x_3)=x_1^2+2x_2^2+3x_3^2-4x_1x_2-4x_2x_3$；

(2) $f(x_1,x_2,x_3,x_4)=2x_1x_2+3x_2x_3$；

(3) $f(x_1,x_2,x_3,x_4)=x_1^2+x_2^2+x_3^2+x_4^2+2x_1x_2+2x_1x_3+2x_1x_4+2x_2x_3+2x_2x_4+2x_3x_4$；

7. 营养学家配制一种具有 1200 卡，30g 蛋白质及 300mg 维生素 C 的配餐，有 3 种食物可供选用：果冻，鲜鱼和牛肉．这 3 种食物每 28.35g 的营养含量如表 1.4 所示．

表 1.4

食　品	果　冻	鲜　鱼	牛　肉
热量/K	20	100	200
蛋白质/g	1	3	2
维生素 C/mg	30	30	10

试计算所需果冻、鲜鱼、牛肉的数量．

8. 对城乡人口流动作年度调查，发现有一个稳定的朝向城镇流动的趋势，每年农村居民的 5% 移居城镇而城镇居民的 1% 迁出，现在总人口的 20% 位于城镇．假如城乡总人口保持不变，并且人口流动的这种趋势继续下去，那么一年以后在城镇人口所占比例是多少？十年以后呢？若干年以后呢？

第二章　微分方程在 Matlab 中的求解方法

§2.1　微分方程的数值求解方法

在微分方程的教程中,介绍了微分方程的基本知识及求解方法,但主要都是解析方法,这些方法能够求解一些比较特殊类型的微分方程,而对于大量较为复杂的微分方程,却难以求出解析解,这就需要使用数值解法. 在实际问题的求解中,一般都是用数值解法来求微分方程的解,微分方程的数值解法主要有欧拉方法和龙格—库塔方法.

2.1.1　欧拉方法

设待求解的微分方程为

$$\frac{dy}{dx}=f(x,y),\quad y(x_0)=y_0 \tag{2.1}$$

欧拉(Euler)方法是一种最简单的数值解法. 该方法的基本思路是在小区间 $[x_n,x_{n+1}]$ 上用差商代替导数 y',即 $\frac{y(x_{n+1})-y(x_n)}{h}\Rightarrow y'$,而方程右端函数 $f(x,y)$ 中的 x 在小区间 $[x_n,x_{n+1}]$ 上取左端点 x_n,由式(2.1)可得近似的表达式

$$y(x_{n+1})\approx y(x_n)+hf(x_n,y(x_n))$$

假定 $y(x_n)\approx y_n$,则 $y(x_{n+1})$ 的近似值为 y_{n+1},且

$$y_{n+1}=y_n+hf(x_n,y_n) \tag{2.2}$$

其中初始点为 (x_0,y_0). 式(2.2)被称为显式欧拉公式,也称为向前欧拉公式.

若在小区间 $[x_n,x_{n+1}]$ 上用差商代替导数 y',方程右端函数 $f(x,y)$ 中的 x 在小区间 $[x_n,x_{n+1}]$ 上取右端点 x_{n+1},同理 $y(x_n)\approx y_n$,可得到另一个欧拉公式

$$y_{n+1}=y_n+hf(x_{n+1},y_{n+1}) \tag{2.3}$$

称为隐式欧拉公式,也称为向后欧拉公式.

向前欧拉公式简单,易于计算,但精度不高,收敛速度慢;向后欧拉公式计算起来比较困难. 如果将向前欧拉公式和向后欧拉公式加以平均,得到

$$y_{n+1} = y_n + \frac{1}{2}h\,[f(x_n, y_n) + f(x_{n+1}, y_{n+1})],\ n = 0, 1, \cdots \tag{2.4}$$

式(2.4)被称为梯形公式. 该方法的计算精度比向前欧拉法和向后欧拉法都要高,而且收敛速度快,但迭代计算与向后欧拉法一样很复杂.

下面介绍改进的欧拉方法,改进的欧拉方法是:将迭代过程简化为两步,先由向前欧拉公式计算出 y_{n+1} 的预测值 $\bar{y}_{n+1}$,再将 $\bar{y}_{n+1}$ 代入梯形式(2.4)的右端,作为校正,即

$$\bar{y}_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

$$y_{n+1} = y_n + \frac{1}{2}h\,[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})] \tag{2.5}$$

称式(2.5)为改进欧拉公式,该式还可以写成

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(k_1 + k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_{n+1}, y_n + hk_1) \end{cases}$$

一般常用的是向前欧拉公式(2.2)和改进欧拉公式(2.5).

在科学研究中会经常遇到常微分方程. 只含有一个自变量的微分方程称为常微分方程. 一般地,有些常微分方程可以找到解析解,但有些常微分方程或者没有解析解,或者求解析解的过程极其繁琐,或者只有数值解等. 在 Matlab 中,对常微分方程的解法一般有两种:符号解(解析解)和数值解.

2.1.2 龙格—库塔方法

利用泰勒(Taylor)公式将 $y(x+h)$ 在 x 处展开,并取前面若干项来近似 $y(x+h)$ 而得到公式

$$y(x+h) \approx y(x) + h\varphi(x, y(x), h)$$

由上式产生的迭代公式

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h), \quad n = 0, 1, \cdots$$

称为龙格—库塔(Runge-Kutta)方法.

若 $y(x+h) - [y(x) + h\varphi(x, y(x), h)] = O(h^{p+1})$,则称以上公式为 p 阶公式,p 的大小反映了截断误差的高低,高阶高精度. 要得到一个 p 阶公式,关键在于如何选取 $\varphi(x_n, y_n, h)$,使之满足阶的要求,常用的龙格—库塔公式如下所述.

1. 二阶公式

中点公式

$$k_1 = hf(x_n, y_n), k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$$

$$y_{n+1} = y_n + k_2$$

改进欧拉公式

$$k_1 = hf(x_n, y_n), k_2 = hf(x_n + h, y_n + k_1)$$

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

2. **三阶公式**

$$k_1 = hf(x_n, y_n), k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$$

$$k_3 = hf(x_n + h, y_n - k_1 + 2k_2)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 4k_2 + k_3)$$

Matlab **中的求微分方程数值解的函数其实现原理就是龙格—库塔方法**.

§2.2　数学建模中常用微分方程基础知识在 Matlab 中的实现

2.2.1　Matlab 中常微分方程的符号求解法

求微分方程或微分方程组的解析解的 Matlab **命令为：**

命令形式 1：r = dsolve（‘eqn1’,‘eqn2’,…,‘var’）

或 r = dsolve（‘eqn1,eqn2,…’,‘var’）

功能：求微分方程或微分方程组的通解. 其中 eqni **表示第** i **个常微分方程,** var **表示微分方程（组）中的自变量,默认时自变量为** t. **通解** r **中包含积分常数** C_1, C_2, C_3 **等**.

命令形式 2：r = dsolve（‘eqn1’,‘eqn2’,…,‘cond1’,‘cond2’,…,‘var’）

或 r = dsolve（‘eqn1,eqn2,…’,‘cond1,cond2,…’,‘var’）

功能：求微分方程（组）满足初始条件的特解. 其中 eqni **表示第** i **个微分方程,**condi **表示第** i **个初始条件,**var **表示微分方程（组）中的自变量,默认时自变量为** t.

在 Matlab **中,约定** D1 **表示一次微分,**D2 **表示二次微分,**D3 **表示三次微分,依此类推,**Dn **表示** n **次微分,符号** Dy **相当于** Dy/Dt. **函数** dsolve **把** D **后面的变量当做因变量,并且默认这些因变量是对自变量** t **求导,当然也可以指定其他的自变量**.

例 1. 求解一阶微分方程$\frac{dy}{dx}=1+y^2$的通解及 $x=0$ 时的特解.

解 在 Matlab 中的命令为

dsolve('Dy = 1 + y 2','x') % 求微分方程通解.

ans = tan (x + C_1)

dsolve('Dy = 1 + y 2','y (0) = 1','x') % 求微分方程特解.

ans = tan (x + 1 /4 * pi).

例 2. 求解二阶微分方程 $x^2y''+xy'+(x^2-n)y=0, y\left(\frac{\pi}{2}\right)=2, y'\left(\frac{\pi}{2}\right)=-\frac{2}{\pi}, n=\frac{1}{2}$.

解 在 Matlab 中的命令为

dsolve('x 2 * D2y + x * Dy + (x 2 - 1 /2) * y = 0','y (pi/2) = 2, Dy (pi/2) = - 2/pi','x')

运行结果为

ans = 2 (1 /2) * pi (1 /2) /x (1 /2) * sin (x),

即所求的解为 $y=\sqrt{\frac{2\pi}{x}}\sin x$.

例 3. 求方程组$\begin{cases}\frac{dx}{dt}=x+y\\ \frac{dy}{dt}=-x+y\end{cases}$在 $x|_{t=0}=1, y|_{t=0}=2$ 的特解.

解 Matlab 的命令为

[x,y] = dsolve('Dx = x + y','Dy = - x + y','x (0) = 1','y (0) = 2')

运行结果为

x = exp (t) * (cos (t) + 2 * sin (t)), y = exp (t) * (- sint (t) + 2 * cos (t)).

2.2.2 Matlab 中常微分方程的数值求解法

求微分方程的数值解一般用龙格—库塔方法. 设微分方程的形式为

$$y'=f(t,y)$$

其中 t 为自变量,y 为因变量 (y 可以是向量,比如微分方程组).

Matlab 中,求一阶微分方程的数值解一般用 2 阶 (3 阶) 龙格—库塔公式和 4 阶 (5 阶) 龙格—库塔公式. 分别为

命令形式 1:[t,y] = oder23('fun', t_s, y_0, options) 2/3 龙格—库塔公式

命令形式 2：[t,y] = oder45 (‘fun’,t_s,y0,options)　　4/5 龙格—库塔公式

其中 fun 是待解方程写成的 M 函数文件名．输入 t_s 的取法有几种，当 t_s = [t_0,t_f] 时，t_0,t_f 分别表示自变量的初始值和终止值；若 t_s = [t_0,t_1,t_2,…,t_f]，则输出在指定时刻 t_0,t_1,t_2,…,t_f 给出；对于等步长时用 t_s = t_0:k:t_f，则输出在区间 [t_0,t_f] 的等分点给出．y_0 为函数的初值．options 用于设定误差限（可以缺省），缺省时设定相对误差为 10^{-3}，绝对误差为 10^{-6}，程序为

options = odeset (‘retol’,rt,‘abstol’,at)

这里 rt,at 分别为设定的相对误差和绝对误差．

而 [t,y] 为输出矩阵，分别表示自变量 t 和因变量 y 的取值．

注意：

(1) 求微分方程数值解的函数命令中，函数 fun 必须以 dx 为输出量，以 t,y 为输入量．

(2) 用于解 n 个未知函数的方程组时，M 函数文件中的待解方程组应以 x 的向量形式写成．

例 4. 用微分方程的数值解法求解微分方程 $y'' + y = 1 - \frac{t^2}{2\pi}$. 设自变量 t 的初始值 $t_0 = 0$，终止值 $t_f = 3\pi$；初始条件 $y|_{x=0} = 0$，$y'|_{x=0} = 0$.

解　(1) 先将高阶微分方程化为一阶微分方程组，其左端分别为变量 x 的两个元素的一阶导数，即：设 $x_1 = y$，$x_2 = x'_1 = y'$，则方程可以化为

$$x'_1 = x_2,\ x'_2 = -x_1 + 1 - \frac{t^2}{2\pi}$$

设 $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$，则原方程写成矩阵形式为

$$x' = \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\left(1 - \frac{t^2}{2\pi}\right) = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\left(1 - \frac{t^2}{2\pi}\right)$$

变量 x 的初始条件是 x (0) = [0,0]，这就是所求的微分方程组的标准形式．

(2) 将上面导数表达式的右端写成一个 exf. m 函数程序，内容如下：

```
function xdot = exf (t, x)
u = 1 - (t. 2) / (2 * pi)
xdot = [0 1 ;- 1 0] * x + [0  1]' * u.
```

(3) 主程序中调用已有的数值积分函数进行积分，其内容如下：

```
clf, t0 = 0 ;tf = 3 * pi ;x0 = [0 ;0]      % 给出初始值.
[t,x] = ode23 ('exf', [t0,tf] ,x0)         % 此处显示结果.
y = x (:,1)                                % y 为 x 的第二列.
```

本题的解析结果为 y2(I) = (1 + 2/(pi 2)) * (1 - cos(t(I))) - t(I) 2/(pi 2).

例 5. 求二阶微分方程 $x^2y'' + xy' + \left(x^2 - \frac{1}{2}\right)y = 0, y\left(\frac{\pi}{2}\right) = 2, y'\left(\frac{\pi}{2}\right) = -\frac{2}{\pi}$ 的数值解.

解 (1) 建立 M 函数文件:

```
function    f = jie3(x,y)
f = [y(2);- y(2)/x + ((1/2) 2/x 2 - 1) * y(1)].
```

(2) 在主程序中调用建立的 M 函数文件:

```
[x,y] = ode23('jie3',[pi/2,pi],[2,- 2/pi])
```

运行结果为:

x = 1.5708	y = 2.0000	- 0.6366
1.6074	1.9758	- 0.6869
1.7645	1.8518	- 0.8879
1.9215	1.6982	- 1.0631
2.0786	1.5912	- 1.2108
2.2357	1.3193	- 1.3293
2.3928	1.1032	- 1.4174
2.5499	0.8756	- 1.4744
2.7069	0.6416	- 1.5002
2.8640	0.4060	- 1.4951
3.0211	0.1735	- 1.4602
3.1416	0.0002	- 1.4140

例 6. 求微分方程 $\frac{d^2x}{dt^2} - 2(1 - x^2)\frac{dx}{dt} + x = 0$ 在初始条件 $x(0) = 1, \frac{dx(0)}{dt} = 0$ 情况下的解.

解 (1) 建立 M 函数命令文件:

```
function ydot = DyDt(t,y)
ydot = [y(2);2 * (1 - y(1) 2) * y(2) - y(1)].
```

(2) 在主程序中调用已建立的 M 函数文件:

```
tspan = [0,30];y0 = [1;0]
[tt,yy] = ode45('DyDt',tspan,y0).
```

oder23 是微分方程(组)数值解的低阶方法,oder45 是较高阶方法,与 oder23 类似.

求高阶微分方程数值解的命令为:[t,y] = oder113('fun',t_s,y0).

习　题　3

1. 求解微分方程 $y' = \frac{x\sin x}{\cos y}$.

2. 求解微分方程组 $\begin{cases} x' = x + 3y \\ y' = x + 4 \end{cases}$.

3. 求下列微分方程

(1) $y'' - y = x - 2, y(0) = 2, y(1) = 1$；

(2) $(1 + x)y'' = 2y - 4, y(0) = 0, y(1) - 2y'(1) = 0$.

4. 在 Matlab 中求下列微分方程组的解

(1) 线性微分方程组 $X' = AX$，其中矩阵 A 的定义如下：

1) $\begin{bmatrix} 2 & 1 \\ -3 & 6 \end{bmatrix}$,　　2) $\begin{bmatrix} 0 & 1 & 0 \\ 4 & 3 & -4 \\ 1 & 2 & 1 \end{bmatrix}$；

(2) 非线性微分方程组 1) $x' = -y - x^2, y' = x$，2) $x' = x - 4y\sqrt{|xy|}, y' = -y + 4x\sqrt{|xy|}$.

5. 求解一阶微分方程 $\frac{dy}{dx} = 1 + y^2$.

6. 求解二阶微分方程

$$x^2y'' + xy' + (x^2 - n^2)y = 0, y\left(\frac{\pi}{2}\right) = 2,$$

$$y'\left(\frac{\pi}{2}\right) = -\frac{2}{\pi}, n = \frac{1}{2}.$$

7. 求题 6 中方程 $x^2y'' + xy' + (x^2 - n^2)y = 0$ 的数值解.

8. 盐水的混合问题. 一个圆柱形的容器，内装 350L 的均匀混合的盐水溶液，如果纯水以每秒 14L 的速度从容器顶部流入，同时，容器内的混合的盐水以每秒 10.5L 的速度从容器底部流出. 开始时，容器内盐的含量为 7kg. 求经过时间 t 后容器内盐的含量.

9. 遗传模型. 孟德尔第一定律：配子的基因是从其父辈的两个基因型中随机地选择的. 实际应用中，将比例作为概率 $P_k(A) = \text{Prob}\{AA \text{ 或 } Aa\}$；$P_k(a) = \text{Prob}\{aa\}$，并记 $X_k = P_k(a)$. 得到如下遗传模型：

(1) 致死基因遗传模型：$X_{k+1} = \frac{X_k}{1 + X_k}$. 试讨论 X_k 的变化趋势.

(2) 自然选择基因遗传模型：$X_{k+1} = \frac{(\beta - 1)X_k^2 + X_k}{1 + (\beta - 1)X_k^2}$. 其中：$\beta = \frac{r_1}{r_2}$，$r_1$ 和 r_2 分

别表示在总人口数量中，新生儿基因为（AA 或 Aa）和（aa）所占的比例．对不同的 β 取值，试讨论 X_k 的变化趋势，选取初值 $X_0 = 0.9$.

(3) 突变基因遗传模型：$X_{k+1} = (1-\mu)X_k + \mu$. 其中 μ 为 A 突变为 a 的概率（比例一般为 $10^{-6} \sim 10^{-5}$）．试对不同的 μ 讨论 X_k 的变化情况，考虑初值 $X_0 = 0.1$.

10. 求解一阶微分方程 $\frac{dy}{dx} = 1 + y^2$ 的通解及 $x = 0$ 时的特解．

11. 求解二阶微分方程 $x^2y'' + xy' + (x^2 - n)y = 0$，$y\left(\frac{\pi}{2}\right) = 2$，$y'\left(\frac{\pi}{2}\right) = \frac{2}{\pi}$，$n = \frac{1}{2}$.

12. 求方程组 $\begin{cases} \frac{dy}{dt} = x + y \\ \frac{dy}{dt} = -x + y \end{cases}$ 在 $x(0) = 1$，$y(0) = 2$ 的特解．

13. 求微分方程 $\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0$ 在初始条件 $x(0) = 1$，$\frac{dx(0)}{dt} = 0$ 的数值解．

14. 在 Matlab 中求微分方程 $xy' = y\ln(xy) - y$ 的通解．

15. 在 Matlab 中求微分方程 $v'(t) + 2t = 0$，$v(1) = 5$ 的特解．

16. 在 Matlab 中求微分方程 $y'' - (a + b)y' + aby = 0$ 的通解．

17. 求 $y'' = x^* \sin x$ 积分曲线方程，使积分曲线通过点 $\left(0, \frac{1}{2}\right)$，且在该点的切线斜率为 2.

第三章　概率论基础知识及其在 Matlab 中的实现

概率论与应用数理统计是研究随机现象统计规律性的一门科学，本章学习随机现象及其发生的概率，多维随机变量的分布规律，参数估计与假设检验，方差分析与回归分析等概率统计的基本方法以及这些知识在 Matlab 中的实现方法.

§3.1　随机时间及其概率

3.1.1　古典概率及其模型

由古典概率的定义知，古典概率基于这样两个原则：(1) 所有可能发生的结果只有有限个；(2) 每一种可能出现的结果机会是相同的. 在 Matlab 中提供了一个在 [0,1] 区间上均匀分布的随机函数 rand，其命令形式为：

命令形式 1：rand (N)

功能：返回一个 N × N 的随机矩阵.

命令形式 2：rand (N,M)

功能：返回一个 N × M 的随机矩阵.

命令形式 3：rand (P_1,P_2,…,P_n)

功能：返回一个 $P_1 \times P_2 \times \cdots \times P_n$ 的随机矩阵.

可以用计算机模拟掷硬币这一过程，为了模拟硬币出现正面或反面，规定随机数小于 0.5 时为反面，否则为正面，可以用 round () 函数将其变成 0—1 矩阵，然后将整个矩阵的各元素值加起来再除以总的原始个数即为出现正面的概率. round () 函数的命令形式为：

命令形式：round (X)

功能：对向量或矩阵 X 的每个分量四舍五入取整.

现以联系掷 10000 次硬币为例，重复做 100 次实验模拟出现正面的概率.

在 Matlab 中的程序如下：

```
for i = 1 :100
   a (i) = sum (sum (round (rand (100)))) /10000 ;
end
mx = max (a) ;
mn = min (a) ;
ma = mean (a) ;
a , mx , mn , ma
```

在该程序输出的四项中，a 为实验 100 次中每次出现正面的频率，mx 和 mn 分别为 100 次实验中出现正面频率的最大值和最小值，ma 为 100 次实验出现正面的平均频率．运行结果如下：

```
a =
   Columns 1 through 6
     0.5107    0.5006    0.5021    0.5033    0.5030    0.4974
   Columns 7 through 12
     0.4931    0.5026    0.5008    0.5087    0.5027    0.5065
   Columns 13 through 18
     0.4964    0.5006    0.4992    0.5038    0.4995    0.5008
   Columns 19 through 24
     0.5009    0.5043    0.4954    0.5097    0.4994    0.4955
   Columns 25 through 30
     0.5010    0.5039    0.4958    0.5030    0.4939    0.4986
   Columns 31 through 36
     0.5027    0.4949    0.4979    0.5018    0.5020    0.5023
   Columns 37 through 42
     0.4938    0.4895    0.5043    0.4953    0.5048    0.4955
   Columns 43 through 48
     0.5048    0.4957    0.4910    0.5035    0.4906    0.4930
   Columns 49 through 54
     0.4994    0.4956    0.4945    0.4908    0.5019    0.5007
   Columns 55 through 60
     0.4977    0.4998    0.5075    0.5040    0.5050    0.4973
   Columns 61 through 66
     0.5053    0.4933    0.4963    0.4919    0.4972    0.5121
```

```
Columns 67 through 72
    0.5032    0.4980    0.4971    0.5021    0.5026    0.5037
Columns 73 through 78
    0.4978    0.5153    0.4952    0.5062    0.4985    0.5020
Columns 79 through 84
    0.5075    0.5107    0.4936    0.4987    0.5017    0.4951
Columns 85 through 90
    0.5055    0.5074    0.5049    0.4964    0.5010    0.4993
Columns 91 through 96
    0.5060    0.5020    0.5081    0.4937    0.5072    0.4985
Columns 97 through 100
    0.4926    0.5038    0.4885    0.5069
mx =  0.5153
mn =  0.4885
ma =  0.5004
```

下面介绍 Matlab 中取整的几个命令函数：

(1) 命令形式：fix (x)

功能：对 x 朝零方向取整.

(2) 命令形式：floor (x)

功能：对 x 朝负无穷大方向取整.

(3) 命令形式：ceil (x)

功能：对 x 朝正无穷大方向取整.

3.1.2　统计概率及其模型

由于古典概率是建立在事件发生的等可能基础上的概率，而现实生活中许多现象的出现并不是等可能的. 例如某品种的玉米，当种下一粒后，其发芽与不发芽的机会并不相同. 那么，这个概率就不能建立在等可能基础上，即不能使用古典概率的定义.

而统计概率的定义是建立在频率基础上的，就是说某事件出现频率如果稳定在某数值 α 附近，则称数值 α 为该事件出现的概率. 由于统计概型中的概率 α 是一个理论上的数值，实际问题中根本无法直接得到该数值，因而通常在实验次数充分多时，利用频率值近似代替概率值. 在掷硬币的实验中，在实验次数充分多的情况下，掷硬币出现正面和反面的频率均在 0.5 左右，故出现正面和反面的概率均为 0.5. 下面来看掷硬币时，当样本容量分别为

n = 10,100,1000,10000,100000,1 000 000

时频率的变化．在 Matlab 中实现时程序代码如下：

```
for i = 1:6
   a(i) = sum(round(rand(1,10 - i)))/10 i;
end
```

运算结果为：

```
a =
   Columns 1 through 10
      0.5000    0.0400    0.0010    0.0002    0.0000    0.0000
0.4931    0.5026    0.5008    0.5087
   Columns 11 through 20
      0.5027    0.5065    0.4964    0.5006    0.4992    0.5038
0.4995    0.5008    0.5009    0.5043
   Columns 21 through 30
      0.4954    0.5097    0.4994    0.4955    0.5010    0.5039
0.4958    0.5030    0.4939    0.4986
   Columns 31 through 40
      0.5027    0.4949    0.4979    0.5018    0.5020    0.5023
0.4938    0.4895    0.5043    0.4953
   Columns 41 through 50
      0.5048    0.4955    0.5048    0.4957    0.4910    0.5035
0.4906    0.4930    0.4994    0.4956
   Columns 51 through 60
      0.4945    0.4908    0.5019    0.5007    0.4977    0.4998
0.5075    0.5040    0.5050    0.4973
   Columns 61 through 70
      0.5053    0.4933    0.4963    0.4919    0.4972    0.5121
0.5032    0.4980    0.4971    0.5021
   Columns 71 through 80
      0.5026    0.5037    0.4978    0.5153    0.4952    0.5062
0.4985    0.5020    0.5075    0.5107
   Columns 81 through 90
      0.4936    0.4987    0.5017    0.4951    0.5055    0.5074
0.5049    0.4964    0.5010    0.4993
```

```
Columns 91 through 100
    0.5060    0.5020    0.5081    0.4937    0.5072    0.4985
  0.4926    0.5038    0.4885    0.5069
```

从上述运算的结果中可以看出，当样本容量不够大时，其频率的波动范围很大，即频率不够稳定，即使有时达到 0.5，但最大时已达到 0.9，然而随着样本容量的增加，频率的波动范围越来越小，相差仅有 10^{-3} 左右.

3.1.3　条件概率、全概率公式与伯努利概率

若事件 B 的发生会影响到事件 A 的发生，则在事件 B 发生的条件下，事件 A 发生的概率称为条件概率，条件概率的计算公式为

$$P(A|B)=\frac{P(AB)}{P(B)}$$

若事件 A 的发生与事件 B 的发生与否没有关系，即事件 B 发生与否不会影响到事件 A 的发生，反之亦然，则称事件 A 与事件 B 是相互独立的，这时有

$$P(AB)=P(A)P(B)$$

例 1. 袋中有 10 个球，其中白球 7 个，黑球 3 个. 分有放回和无放回两种情况，分三次取球，每次取一个，试分别求：(1) 第三次摸到了黑球的概率，(2) 第三次才摸到黑球的概率，(3) 三次都摸到了黑球的概率.

解　当有放回地摸球时，由于三次摸球互不影响，因此三次摸球相互独立，从理论上可以求得：(1) 第三次摸到黑球的概率为 $\frac{3}{10}=0.3$；(2) 第三次才摸到黑球的概率为 $\frac{7}{10}\times\frac{7}{10}\times\frac{3}{10}=0.147$；(3) 三次都摸到黑球的概率为 $\frac{3}{10}\times\frac{3}{10}\times\frac{3}{10}=0.027$.

在 Matlab 中模拟这一过程时，可以在 [0,1] 区间上产生三次随机数来模拟三次摸球，当随机数小于 0.7 时可以认为摸到了白球，否则认为摸到了黑球. 重复 10^6 次分别求上述三种情况出现的概率. 程序如下：

```
a = round(rand(1000000,3) - 0.2);
for i = 1:6
   b = a(1:10^i,3);
   c(i) = sum(b)/(10^i);
end
c
for i = 1:6
```

```
    b = (~a(1:10 i,1)) & (~a(1:10 i,2)) &a(1:10 i,3);
    d(i) = sum(b)/(10 i);
end
d
forI = 1:6
    b = a(1:10 i,1) &a(1:10 i,2) &a(1:10 i,3);
    e(i) = sum(b)/(10 i);
end
e
```

运行结果为

c = 0.4000	0.2900	0.3040	0.2971	0.2970	0.3000
d = 0.2000	0.1000	0.1540	0.1448	0.1457	0.1474
e = 0	0	0	0	0	0.0267

执行结果中 c 为第三次摸到黑球的概率，d 为第三次才摸到黑球的概率，e 为三次都摸到黑球的概率．可以看到，随着实验次数的增加，其频率都会逐渐稳定在理论值附近．

当无放回地摸球时，由于第二次摸球会受到第一次的影响，而第三次摸球又会受到前两次的影响，因而三次摸球相互影响，并不独立．从理论上可以求得：

(1) 第三次摸到黑球的概率为

$$\frac{7}{10}\times\frac{6}{9}\times\frac{3}{8}+\frac{7}{10}\times\frac{3}{9}\times\frac{2}{8}+\frac{3}{10}\times\frac{7}{9}\times\frac{2}{8}+\frac{3}{10}\times\frac{2}{9}\times\frac{1}{8}=0.3$$

(2) 第三次才摸到黑球的概率为

$$\frac{7}{10}\times\frac{6}{9}\times\frac{3}{8}=0.175$$

(3) 三次都摸到了黑球的概率为

$$\frac{3}{10}\times\frac{2}{9}\times\frac{1}{8}=0.008.$$

用计算机模拟该过程时，在 [0,1] 区间模拟第一次摸球，当值小于 0.7 时认为摸到了白球，否则认为摸到了黑球；第二次摸球时由于少了一个球，故可以在区间长度为 0.9 的区间上模拟，若第一次摸到白球，可以将区间设为 [0.1,1]，否则区间设为 [0,0.9]；第三次摸球可以依此类推，其模拟程序如下：

```
a = rand(1000000,3);
a(:,1) = round(a(:,1) - 0.2);
a(:,2) = round(a(:,2)*0.9 - 0.2 - 0.1*(a(:,1) - 1));
```

```
a(:,3) = round(a(:,3)*0.8 - 0.2 - 0.1*(a(:,1) - 1) - 0.1*(a(:,2) - 1));
for i = 1:6
   b = a(1:10 i,3);
   c(i) = sum(b)/(10 i);
end
c
for i = 1:6
   b = (~a(1:10 i,1)) & (~a(1:10 i,2)) & a(1:10 i,3);
   d(i) = sum(b)/(10 i);
end
d
for i = 1:6
   b = a(1:10 i,1) & a(1:10 i,2) & a(1:10 i,3);
   e(i) = sum(b)/(10 i);
end
e
```

运行结果为

c = 0	0.2300	0.2870	0.3035	0.2989	0.2997
d = 0	0.1700	0.1610	0.1810	0.1749	0.1749
e = 0	0	0.0100	0.0086	0.0080	0.0084

上述在理论上计算第三次摸到黑球的概率时，用到了全概率公式.

若 $A_1, A_2, \cdots, A_n$ 构成一个完备事件组，且事件 B 的发生总是伴随着事件 A_i $(i = 1, 2, \cdots, n)$ 中的某一个发生而发生，则

$$P(B) = \sum_{i=1}^{n} P(A_i) P(B \mid A_i)$$

下面将用到伯努利 (Bernoulli) 概型，所谓伯努利概型是指：在相同条件下，进行 n 次独立重复实验，每次实验只有事件 A 发生或不发生两种结果，且

$$P(A) = p, \quad P(\bar{A}) = 1 - p = q$$

这里第三次摸到黑球的四种情况分别是：{白，白，黑}，{白，黑，黑}，{黑，白，黑}，{黑，黑，黑}. 这四种情况构成了完备事件组. 现考虑下述问题：(1) 当不放回时，已知第三次摸到了黑球，问前两次是黑球的概率为多少？(2) 若有放回地连续摸 10 次，则恰有三次摸到黑球的概率是多少？

第一问是一逆概率问题，由逆概率公式即贝叶斯 (Bayes) 公式得到其概率应

为

$$\frac{3}{10}\times\frac{2}{9}\times\frac{1}{8}\times\frac{3}{10}=\frac{1}{36}\approx 0.0278$$

第二问则属伯努利概型，这里 A 为{摸到的是黑球}，故 P（A）= 0.3，P（$\bar{A}$）= 0.7. 于是由二项概率公式有，10 次有放回地摸球中，恰有三次摸到黑球的概率为

$$C_{10}^{3}\times 0.3^{3}\times 0.7^{7}\approx 0.2668$$

在 Matlab 中实现这两个过程的程序如下：

```
a = rand(1000000,3);
a(:,1) = round(a(:,1) - 0.2);
a(:,2) = round(a(:,2)*0.9 - 0.2 - 0.1*a(:,1) - 1);
a(:,3) = round(a(:,3)*0.8 - 0.2 - 0.1*(a(:,1) - 1) - 0.1*
(a(:,2) - 1));
for i = 1:6
  b = a(1:10 i,3);
  c(i) = sum(b);
  b = a(1:10 i,1) & a(1:10 i,2) & a(1:10 i,3);
  d(i) = sum(b);
  e(i) = d(i)./c(i);
end
c, d, e
```

c = 2	42	446	4405	43708	438604
d = 0	4	108	1024	9973	100414
e = 0	0.0952	0.2422	0.2325	0.2282	0.2289

```
a = round(rand(1000000,10) - 0.2);
for i = 1:6
b = sum(a(1:10 i, :),2) - 3;
c(i) = sum(-b)/(10 i);
end
c
```

运行结果为

c = 0.8000 -0.0100 -0.0180 -0.0067 -0.0039 0.0013

§3.2　随机变量的分布及其数字特征

随机变量的统计行为完全决定于其概率分布,按随机变量的取值不同,通常可以将其分为离散型、连续型和奇异型三大类. 由于奇异型在实际应用中很少遇到,因此只讨论离散型和连续型两类随机变量的概率分布及其数字特征.

3.2.1　离散型随机变量的分布及其数字特征

如果随机变量 X 的所有可能取值为有限个或无穷可列个,则称 X 为离散型随机变量. 设 X 的所有可能值为 $X_1, X_2, \cdots$,并且 X 取这些值的概率为

$$P\{X = X_k\} = p_k, \quad k = 1, 2, \cdots$$

则称其为随机变量 X 的概率分布. 该分布满足下面的性质:

(1) $p_k \geqslant 0, k = 1, 2, \cdots$;

(2) $\sum_{k=1} p_k = 1.$

称 $F(x) = \sum_{x_k \leqslant x} p_k$ 为累积概率分布.

在研究随机变量时,主要就是研究随机变量的概率分布、累积分布和分布的数字特征. 常用的离散型随机变量的分布有:二项分布、泊松分布和超几何分布.

1. 二项分布

若随机变量 X 的所有可能取值为 0,1, …,n,其概率分布为

$$P\{X = k\} = C_n^k p^k q^{n-k}, \quad k = 0, 1, 2, \cdots, n$$

其中,$q = 1 - p$,则称 X 服从参数为 n 和 p 的二项分布,记做 $X \sim B(n, p)$. 二项分布的数学期望为 $E(X) = np$,方差为 $D(X) = npq$.

在 Matlab 中提供的二项分布的统计函数有:binopdf ()、binocdf ()、binoinv ()、binornd ( ),以及计算二项分布均值和方差的函数 binostat (),这些函数的命令形式如下:

命令形式:binopdf (X,N,P)

功能:计算二项分布的密度函数. 其中 X 为随机变量,N 为独立实验的重复数,P 为事件发生的概率.

命令形式:binocdf (X,N,P)

功能:计算二项分布的累积分布函数. 其中 X 为随机变量,N 为独立实验的重复数,P 为事件发生的概率.

命令形式：binoinv (X,N,P)

功能：计算二项分布的逆累积分布函数．其中 X **为随机变量，**N **为独立实验的重复数，**P **为事件发生的概率**

命令形式：binornd (N,P,m,n)

功能：产生服从二项分布的 m × n **阶随机矩阵．其中** N **为独立实验的重复数，**P **为事件发生的概率，**m **和** n **分别是所产生随机矩阵的行数和列数．**

若不指定 m **和** n**，则返回一个随机数；若指定** m **和** n**，则返回一个服从二项分布的** m × n **阶随机矩阵．**

命令形式：binostat (N,P)

功能：求二项分布的数学期望与方差．N **为独立实验的重复数，**P **为事件发生的概率．**

例 2. x = 0：0.1：1.

```
binoinv (x,10,0.4)
ans = 0   2   3   3   4   4   4   5   5   6   10
binoinv (x,10,0.3)
ans = 0   1   2   2   3   3   3   4   4   5   10
binoinv (x,50,0.7)
ans = 0   31   32   33   34   35   36   37   38   39   50
binoinv (x,50,0.3)
ans = 0   11   12   13   14   15   16   17   18   19   50.
```

例 3. **生成一个或多个服从二项分布的随机数．**

```
binornd (10,0.7)
ans = 6
binornd (10,0.7,2,3)
ans = [8   7   9;3   5   5].
```

例 4. **求二项分布的数学期望** (e) **和方差** (d)．

```
[e d] = binostat (10,0.3)
e = 3,   d = 2.1000
[e,d] = binostat (20,0.7)
e = 14,   d = 4.2000.
```

2. **泊松分布**

如果随机变量的概率分布为

$$P\{X=k\}=\frac{\lambda^k}{k!}\exp(-\lambda),\quad k=0,1,2,\cdots$$

其中，$\lambda > 0$ 为常数，则称 X 服从参数为 λ 的泊松 (Poisson) 分布，记做 $X \sim P(\lambda)$，泊松分布的数学期望为 $E(X) = \lambda$，方差为 $D(X) = \lambda$.

在 Matlab 中，提供如下有关泊松分布的统计函数：

命令形式：poisspdf (X,LMD)

功能：求泊松分布的密度函数．其中 X 为随机变量，LMD 为参数．

命令形式：poisscdf (X,LMD)

功能：求泊松分布的累积分布函数．其中 X 为随机变量，LMD 为参数．

命令形式：poissinv (Y,LMD)

功能：求泊松分布的逆累积分布函数．其中 Y 为显著概率值，LMD 为参数.

命令形式：poissrnd (LMD,M,N)

功能：产生服从泊松分布的随机数．其中 LMD 为参数，m 和 n 为产生随机矩阵的行数和列数．

命令形式：poisstat (LMD)

功能：求泊松分布的数学期望与方差．其中 LMD 为参数．

可以利用逆累积概率分布函数求一定显著概率条件下，泊松分布假设检验临界值

```
x = 0:0.1:1;
poissinv (x,5)
ans = 0   2   3   4   4   5   5   6   7   8   Inf
poissinv (x,10)
ans = 0   6   7   8   9   10   11   12   13   14   Inf
poissinv (x,100)
ans = 0   87   92   95   97   100   102   105   108   113   Inf
```

在 Matlab 中求服从泊松分布的随机数及数学期望与方差如下：

```
poissrnd (1)          ans = 1
poissrnd (5)          ans = 5
poissrnd (5,5,10)
ans = 7    7    7    3    8    6    5    9    4    10
      4    3    5    4    5    5    4    6    6    4
      3    4    2    8    4    3    4    6    12   8
      6    2    3    3    5    8    6    3    6    5
      6    5    9    5    8    3    5    10   8    5
[e,d] = poisstat (5)
e = 5,      d = 5
```

[e,d] = poisstat (10)

e = 10, d = 10

3. **超几何分布**

如果随机变量 X **所有可能取值为** 0,1,2,…,L (L = min {M,N}),X **的概率分布为**

$$P\{X=k\}=\frac{C_M^k C_{N-M}^{n-k}}{C_N^n},\ (k=0,1,2,\cdots,L),$$

其中整数 M,N > 0,**且** n ≤ N - M,**则称** X **服从参数为** N,M,n **的超几何分布,记做**

$$X \sim H(N,M,n)$$

Matlab **中超几何分布的统计函数为:**

命令形式:hygepdf (M,n,k,N)

功能:求超几何分布的密度函数.

命令形式:hygepcdf (M,n,k,N)

功能:求超几何分布的累积分布函数.

命令形式:hygeinv (P,n,k,N)

功能:求超几何分布的逆累积分布函数.

命令形式:hygestat (n,k,N)

功能:求超几何分布的数学期望与方差.

命令形式:hygernd (n,k,N,mr,mc)

功能:产生满足超几何分布的随机数. **其中** mr **和** mc **分别为所产生随机矩阵的行数和列数**. mr **和** mc **省略时产生一个随机数**.

用逆累积概率分布函数求一定显著概率条件下,超几何分布假设检验临界值的程序如下:

```
x = 0:0.1:1;
hygeinv (x,10,5,6)
ans = 0    2    2    3    3    3    3    3    4    4    5
hygeinv (x,15,5,9);
ans = 0    2    2    3    3    3    3    3    4    4    5
hygeiv (x,20,8,10)
ans = 0    3    3    3    4    4    4    5    5    5    8
```

求服从超几何分布的随机数及数学期望与方差的程序如下:

```
hygernd (15,7,9)      ans = 5
hygernd (15,7,9,2,3)
ans = 4    5    4
```

```
        4      5      6
[e,d] = hygestat(15,7,9)
e = 4.2000,        d = 0.9600
[e,d] = hygestat(20,8,10)
e = 4,             d = 1.2632.
```

3.2.2　连续型随机变量的分布及其数字特征

设随机变量 X 的分布函数为 F(x),若存在非负函数 f(x),使对任意实数 x,有

$$F(x) = P\{X \leqslant x\} = \int_{-}^{x} f(t)\,dt$$

则称 X 为连续型随机变量,并称 f(x) 为 X 的概率密度,f(x) 满足下列性质:

(1) $f(x) \geqslant 0$,　$-\quad < x < +\quad$;

(2) $\int_{-}^{+} f(x)\,dx = 1$;

(3) $P\{a < x \leqslant b\} = F(b) - F(a) = \int_{a}^{b} f(x)\,dx$;

(4) $P\{x = a\} = 0$.

性质(4)和离散型随机变量截然不同,性质(4)表明概率为零的事件并不一定是不可能事件. 常用的三种连续型随机变量的概率分布是均匀分布、指数分布和正态分布.

1. 均匀分布

若连续型随机变量 X 的概率密度为

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a < x < b \\ 0, & \text{其他} \end{cases}$$

则称 X 在区间(a,b)上服从参数为 a 和 b 的均匀分布,记做 $X \sim U(a,b)$.

Matlab 中提供的均匀分布的函数如下:

命令形式:unifpdf(X,A,B)

功能:求均匀分布的密度函数. 其中 X 为随机变量,A、B 为均匀分布参数.

命令形式:unifcdf(X,A,B)

功能:求均匀分布的累积分布函数. 其中 X 为随机变量,A、B 为均匀分布参数.

命令形式:unifinv(P,A,B)

功能:求均匀分布的逆累积分布函数. 其中 P 为概率值,A、B 为均匀分布参

数.

命令形式:unirnd (A,B,m,n)

功能:产生服从均匀分布的随机数. 其中 A、B 为均匀分布参数,m 和 n 为生成随机数矩阵的行数和列数.

命令形式:unifstat (A,B)

功能:求均匀分布的数学期望与方差. 其中 A、B 为均匀分布参数.

2. 指数分布

如果随机变量 X 的概率密度为

$$f(x)=\begin{cases}\lambda\exp(-\lambda x), & x\geqslant 0\\ 0, & x<0\end{cases}$$

其中 λ 为常数,则称 X 为服从参数为 λ 的指数分布,记做 $X\sim e(\lambda)$.

Matlab 中指数分布的函数如下:

命令形式:exppdf (X,L)

功能:求指数分布的密度函数. 其中 X 为随机变量,L 为参数 λ.

命令形式:expcdf (X,L)

功能:求指数分布的累积函数. 其中 X 为随机变量,L 为参数 λ.

命令形式:expinv (P,L)

功能:求指数分布的逆累积分布函数. 其中 P 为显著概率,L 为参数 λ.

命令形式:exprnd (X,L,m,n)

功能:产生服从指数分布的随机数. 其中 X 为随机变量,L 为参数 λ,m 和 n 为随机数矩阵的行数和列数.

命令形式:expstat (L)

功能:求指数分布的数学期望和方差. 其中 L 为参数 λ.

3. 正态分布

如果随机变量 X 的概率密度为

$$f(x)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),\quad -\quad <x<+$$

其中 μ,σ 均为常数,且 $\sigma>0$,则称 X 服从参数为 μ 和 σ^2 的正态分布,记做 $X\sim N(\mu,\sigma^2)$,当 $\mu=0,\sigma=1$ 时,称 X 服从标准正态分布,记做 $X\sim N(0,1)$.

Matlab 中提供的正态分布的函数如下:

命令形式:normpdf (X,M,C)

功能:求正态分布的密度函数. 其中 X 为随机变量,M 为正态分布参数 μ,C 为参数 σ.

命令形式:normcdf (X,M,C)

功能：求正态分布的累积分布函数．其中 X 为随机变量，M 为正态分布参数 μ，C 为参数 σ.

命令形式：norminv (P，M，C)

功能：求正态分布的逆累积分布函数．其中 P 为显著概率，M 为正态分布参数 μ，C 为参数 σ.

命令形式：normrnd (M，C，m，n)

功能：产生服从正态分布的随机数．其中 M 为正态分布参数 μ，C 为参数 σ，m 和 n 为随机矩阵的行数和列数．

命令形式：normstat (M，C)

功能：求正态分布的数学期望和方差．其中 M 为正态分布参数 μ，C 为参数 σ.

在 Matlab 中求标准正态分布的密度函数及累积分布函数和一般正态分布的密度函数及累积分布函数的程序如下：

```
x = -4:0.01:4;
y = normpdf(x,0,1);z = normcdf(x,0,2);
subplot(2,2,1);plot(x,y,'k');
axis([-4,4,-0.1,0.5]);
subplot(2,2,2);plot(x,z,'k');
axis([-4,4,-0.1,1.1]);
x = -4:0.01:16;
y1 = normpdf(x,6,1);z1 = normcdf(x,6,1);
y2 = normpdf(x,6,4);z2 = normcdf(x,6,1);
y3 = normpdf(x,6,0.6);z3 = normcdf(x,6,0.6);
subplot(2,2,3);
plot(x,y1,'k',x,y2,'k',x,y3,'k');
axis([-4,16,-0.1,0.8]);
subplot(2,2,4);plot(x,z1,'k',x,y2,'k',x,y3,'k');
axis([-4,16,-0.1,1.1]);
```

其计算结果如图 3.1 所示．

用正态分布的逆累积分布函数求一定显著概率条件下，假设检验临界值如下：

```
x = 0:0.1:1;
norminv(x,0,1)
ans = -Inf   -1.2816 -0.8416   -0.5244   -0.2533   0   0.2533   0.5244
      0.8416   1.2816   Inf
```

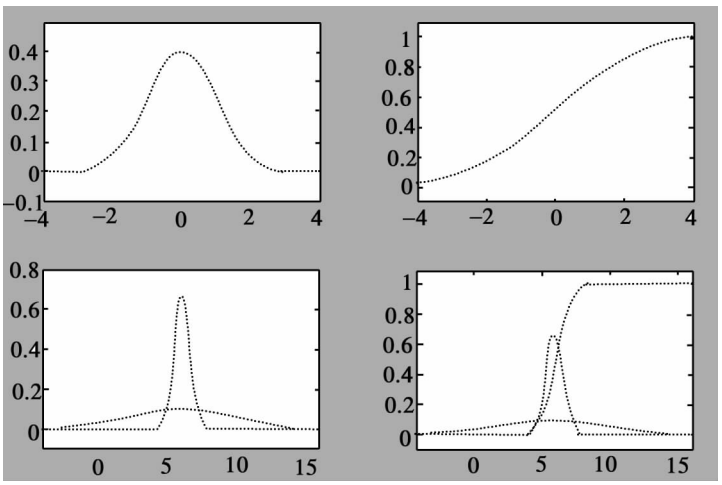


图 3.1　标准正态分布和一般正态分布的密度函数及累积分布函数图

```
norminv (x,0. 1,0. 01)
ans = [- Inf,0. 0872,0. 0916,0. 0948,0. 0975,0. 1,0. 1025,0. 1052,0. 1084,
  0. 1128,Inf]
norminv (x,2,4)
ans = [- Inf, - 3. 1262, - 1. 3665, - 0. 0976,0. 9866,2. 0000,3. 0134,4. 0976,
  5. 3665,7. 1262,Inf]
norminv (x,2,7)
ans = [- Inf, - 6. 9709, - 3. 8913, - 1. 6708,0. 2266,2. 0000,3. 7734,5. 6708,
  7. 8913,10. 9709,Inf]
```

产生服从标准正态分布或一般正态分布的随机数的程序如下：

```
normrnd (0,1)          ans = - 0. 4326
normrnd (2,4)          ans = - 4. 6623
normrnd (2,4,5,7)
ans = [4. 6744   - 2. 8098   1. 3731 ;6. 7634   1. 9208   - 4. 4163]
normrnd (0,1,5,7)
ans = [0. 2573   1. 4151   0. 5287 ;- 1. 0565   - 0. 8051   0. 2193]
```

求正态分布数学期望和方差的例子如下：

```
[e d] = normstat (0,1)     e = 0     d = 1
[e d] = normstat (2,4)     e = 2     d = 16
```

[e d] = normstat (2,0.1)　　e = 2　　d = 0.0100.

3.2.3　χ^2 分布、t 分布和 F 分布

设二维随机变量 (X,Y) 的分布函数为 F (x,y),如果存在非负函数 f (u,v),使对任意实数 x,y 有

$$F(x,y) = \int_{-}^{x}\int_{-}^{y} f(u,v)\,dudv$$

则称 (X,Y) 为连续型随机变量,称 f (u,v) 为二维连续型随机变量 (X,Y) 的概率密度,或称为 X 与 Y 的联合概率密度.

若二维随机变量最多只能取有限对或无穷可列对值 (x_i, y_j),i,j = 1,2,…,则称 (X,Y) 为二维离散型随机变量,称 (X,Y) 的所有可能取值的概率规律

$$P\{X = x_i, Y = y_i\} = p_{ij}, i,j = 1,2,\cdots$$

为二维离散型随机变量 (X,Y) 的概率分布,或称为 X 与 Y 的联合概率分布.

统计分析中常用的分布有 χ^2 分布、t 分布和 F 分布.

1. χ^2 分布

设随机变量 $X_1, X_2, \cdots, X_n$ 相互独立,且均服从 N (0,1),令

$$X = X_1^2 + X_2^2 + \cdots + X_n^2$$

则 X 的分布称为具有自由度 n 的 χ^2 分布,记做 $X \sim \chi^2(n)$.

Matlab 中计算 χ^2 分布的函数如下:

命令形式:chi2pdf (X,N)

功能:计算 χ^2 分布的密度函数. 其中 X 为随机变量,N 为 χ^2 分布自由度.

命令形式:chi2cdf (X,N)

功能:计算 χ^2 分布的累积分布函数. 其中 X 为随机变量,N 为 χ^2 分布自由度.

命令形式:chi2inv (P,N)

功能:计算 χ^2 分布的逆累积分布函数. 其中 P 为显著概率,N 为 χ^2 分布自由度.

命令形式:chi2rnd (N,m,n)

功能:产生服从 χ^2 分布的随机数. 其中 N 为 χ^2 分布自由度,m 和 n 为产生的随机数矩阵的行数和列数.

命令形式:chi2stat (N)

功能:求 χ^2 分布的数学期望与方差. N 为 χ^2 的分布自由度.

在 Matlab 中绘制 χ^2 分布的密度函数和累积分布函数图的程序如下:

```
x = 0:0.01:40;
```

```
y = chi2pdf(x,1);z = chi2cdf(x,1);
y1 = chi2pdf(x,4);z1 = chi2cdf(x,4);
y2 = chi2pdf(x,10);z2 = chi2cdf(x,10);
y3 = chi2pdf(x,20);z3 = chi2cdf(x,20)
subplot(1,2,1);
plot(x,y,'k',x,y1,'k',x,y2,'k',x,y3,'k');
axis([0,40,0,0.2]);
subplot(1,2,2);
plot(x,z,'k',x,z1,'k',x,z2,'k',x,z3,'k');
axis([0,40,0,1.01]);
```

其运算结果如图 3.2 所示.

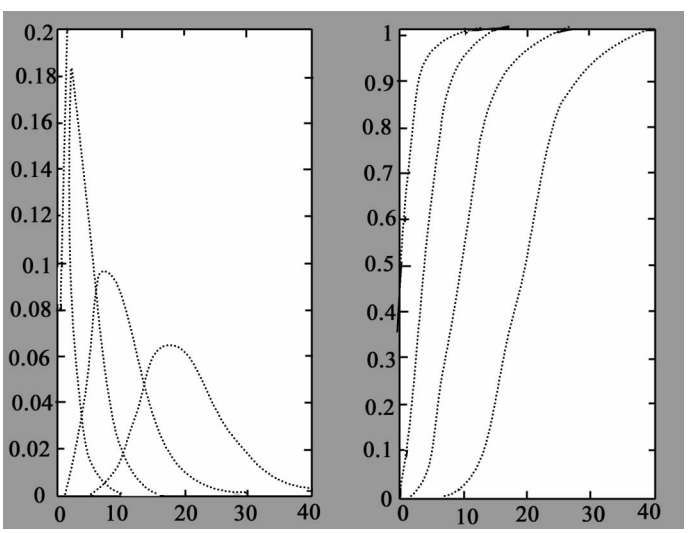


图 3.2 χ^2 分布的密度函数和累积分布函数图

2. t 分布

设随机变量 X 与 Y 独立,且 $X \sim N(0,1)$,$Y \sim \chi^2(n)$,令

$$T = \frac{X}{\sqrt{Y/N}}$$

则称 T 为具有自由度 n 的 t 分布,记做 $T \sim t(n)$.

Matlab 中计算 t 分布的命令函数如下:

命令形式:tpdf(X,N)

功能：计算 t 分布的密度函数．其中 X 为随机变量，N 为 t 分布的自由度．

命令形式：tcdf (X,N)

功能：计算 t 分布的累积分布函数．其中 X 为随机变量，N 为 t 分布的自由度．

命令形式：tinv (P ,N)

功能：计算 t 分布的逆累积分布函数．其中 P 为显著概率，N 为 t 分布的自由度．

命令形式：trnd (N,m,n)

功能：产生服从 t 分布的随机数．其中 N 为 t 分布的自由度，m 和 n 为产生的随机数矩阵的行数和列数．

命令形式：tstat (N)

功能：求 t 分布的数学期望与方差．其中 N 为 t 分布的自由度．

在 Matlab 中绘制 t 分布的密度函数和累积分布函数图的程序如下：

```
x = -5:0.01:5;
y = tpdf(x,1);z = tcdf(x,1);
y1 = tpdf(x,2);z1 = tcdf(x,2);
y2 = tpdf(x,10);z2 = tcdf(x,10);
subplot(1,2,1);
plot(x,y,'k',x,y1,'k',x,y2,'k');
axis([-5,5,0,0.4]);
subplot(1,2,2);
plot(x,z,'k',x,z1,'k',x,z2,'k');
axis([-5,5,0,1.01]);
```

其运算结果如图 3.3 所示．

3. F 分布

设随机变量 X 与 Y 独立，且 $X \sim \chi^2(m)$，$Y \sim \chi^2(n)$，令

$$F = \frac{X/m}{Y/n}$$

则称 F 服从第一自由度为 m，第二自由度为 n 的 F 分布，记做 $F \sim F(m,n)$．

Matlab 中计算 F 分布的命令函数如下：

命令形式：fpdf (X,M,N)

功能：计算 F 分布的密度函数．

命令形式：fcdf (X,M,N)

功能：计算 F 分布的累积分布函数．

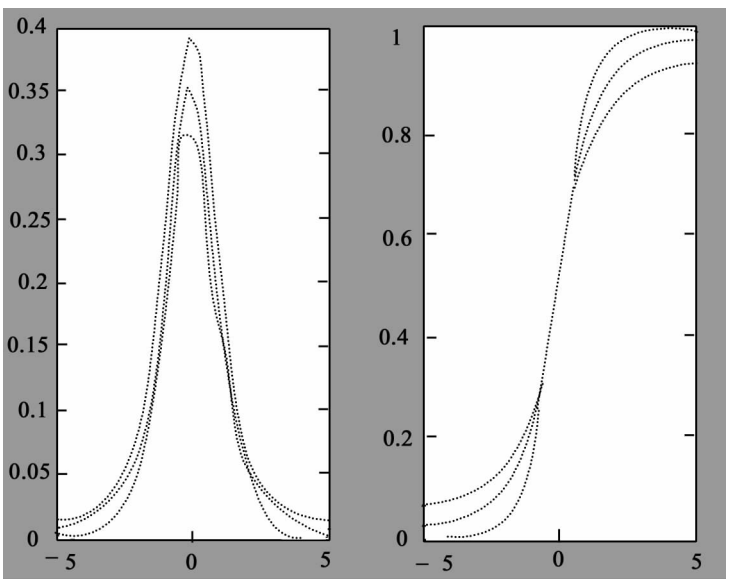


图 3.3　t 分布的密度函数和累积分布函数图

命令形式：finv (P ,M,N)

功能：计算 F 分布的逆累积分布函数 .

命令形式：frnd (M,N,m,n)

功能：产生服从 F 分布的随机数 .

命令形式：fstat (M,N)

功能：计算 F 分布的数学期望与方差 .

其中 X 为随机变量,M 和 N 分别为 F 分布的第一自由度和第二自由度,P 为显著概率,m 和 n 为所产生随机数矩阵的行数和列数 .

例 5. 试绘出 F 分布的分布函数曲线和概率密度函数曲线 .

解　在 Matlab 中的程序如下：

```
x = 0:0.01:4;
y1 = fpdf(x,10,50);
z1 = fcdf(x,10,50);
y2 = fpdf(x,10,5);
z2 = fcdf(x,10,5);
plot(x,y1,x,y2);
plot(x,z1,x,z2);
```

```
gtext ('F (10,50) ');
gtext ('F (10,5) ');
```

其计算结果如图 3.4 所示.

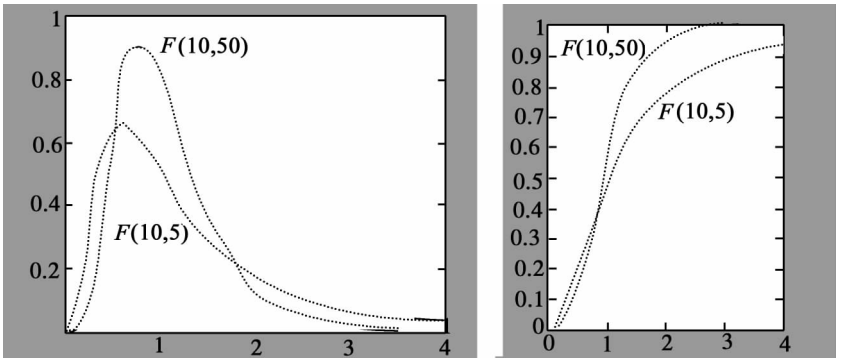


图 3.4　F 分布的分布函数曲线和概率密度函数曲线

在 Matlab 中用逆累积分布函数求一定显著概率条件下,服从 F 分布的假设检验临界值如下:

```
finv (0.99,10,4)        ans = 14.5459
finv (0.95,10,4)        ans = 5.9644
finv (0.95,10,20)       ans = 2.3479
x = 0:0.1:1
finv (x,10,4)
ans  =
   Columns 1 through 10
            0        0.3838        0.5469        0.7100        0.8926
1.1126        1.3971        1.8002        2.4596        3.9199
   Column 11
         Inf
finv (x,10,20)
ans =
   Columns 1 through 10
         0        0.4544        0.5944        0.7166        0.8375        0.9663
1.1122        1.2901        1.5313        1.9367
   Column 11
```

```
        Inf
```

产生服从 F 分布的随机数的程序如下：

```
frnd (10,4)  ans = 2.8855
frnd (10,20)  ans = 1.5200
frnd (2,10,5,7)
ans  = 1.1583    0.8586    0.1159    2.0684    19.6538    3.3214    2.0390
       0.2848    0.8727    2.5215    0.5639    0.9910     1.2247    0.6866
       0.2655    1.1704    0.7414    0.3396    0.7085     2.9459    0.4724
       3.6858    0.8175    0.2032    2.2460    1.2893     0.3896    2.1736
       0.9331    1.3250    0.0716    3.6767    1.3283     2.8547    2.1526
```

求 F 分布数学期望与方差的程序为：

```
[e    d] = fstat (10,4)      e  =  2       d  = NaN
[e    d] = fstat (2,10)      e  = 1.2500 d  = 2.6042
[e    d] = fstat (4,15)      e  = 1.1538 d  =  1.0288.
```

§3.3 参数估计与假设检验

3.3.1 样本的数字特征

1. 集中趋势（位置）度量

数据样本集中趋势度量的目的是在对于数据样本在数据分布线上分布的中心予以定位，即中心位置的度量．集中趋势度量包括几何平均值、调和平均值、算术平均值、中位数和修正的样本均值等．在 Matlab 中求这些度量值的函数为：

命令形式：geomean (X)

功能：求几何样本的平均值．其中 X 为样本数据．

命令形式：harmmean (X)

功能：求样本的调和平均值．其中 X 为样本数据．

命令形式：mean (X)

功能：求样本的算术平均值．其中 X 为样本数据．

命令形式：median (X)

功能：求样本的中位数．其中 X 为样本数据．

命令形式：trimmean (X,P)

功能：求修正的样本均值．其中 X 为样本数据，P 为计算时剔除样本数据

中最高 P% 和最低 P% 后的均值.

注意:在上述命令中,若 X 为向量,则返回该向量的各度量值. 若 X 为矩阵,则返回矩阵各列的度量值组成的行向量.

2. 离中趋势(散布)度量

离中趋势的度量可以理解为样本的数据偏离中心位置的程度,或称为离差. 离中趋势度量一般包括极差、标准差、方差、平均绝对偏差和内四分位数间距等. Matlab 中求离中趋势度量的函数如下:

命令形式:iqr(X)

功能:求样本的内四分位数间距.

命令形式:mad(X)

功能:求样本的平均绝对差.

命令形式:rang(X)

功能:求样本数据的极差.

命令形式:var(X)

功能:求样本数据的方差.

命令形式:std(X)

功能:求样本数据的标准差.

命令形式:cov(X) 或 cov(X, Y)

功能:求样本数据的协方差.

其中 X 与 Y 均为样本数据. 计算协方差时,若用 cov(X),则 X 必须为矩阵;若用 cov(X,Y),X 和 Y 均为长度相等的列向量.

3. 中心矩

中心矩是关于数学期望的矩. 对于任意的 $k > 0$,称 $\mu_k = E(X - E(X))^k$ 为随机变量 X 的 k 阶中心矩. 显然,一阶中心矩为 0,二阶中心矩为方差. 在 Matlab 中,用命令 moment() 求中心矩:

命令形式:moment(X, k)

功能:求样本数据 X 的 k 阶中心矩. 其中 X 为随机样本点组成的向量或矩阵,k 为正整数. 若 X 为向量,则返回该向量的 k 阶中心矩;若 X 为矩阵,则返回由 X 的每一列组成的向量的中心矩构成的行向量.

例 6. X = normrnd(2.5, 1.25, 10, 7)

```
X = 1.9593   2.2666   2.8680   2.0001   0.4949   1.2367   2.5001
    0.4180   3.4072   0.8298   3.3625   2.8216   3.2681   2.1027
    2.6567   1.7646   3.3929   3.5195   1.1794   3.1347   3.8688
    2.8596   5.2290   4.5295   3.3899   4.2689   4.6155   0.1575
```

```
    1.0669    2.3295    1.6353    4.1128    1.4936    3.2391    3.0352
    3.9886    2.6424    3.5725    3.3358    3.1609    1.6955    3.6195
    3.9865    3.8335    4.0675    3.9885    2.7742    2.9754    3.4137
    2.4530    2.5741    0.5078    0.9969    1.3476    1.2386    3.2223
    2.9091    2.3804    0.6988    2.4753   -0.2133    2.4756    2.5504
    2.7183    1.4596    3.2139    2.3041    2.4260    2.4397    3.4464
   Y = normrnd(5,2,1,7)
Y  = 6.1378    4.4887    4.2451    4.4082    2.0497    4.5320    5.2369
   moment(x,2)
ans = 1.1477    1.0962    1.9909    0.8627    1.6465    0.9905    1.0341
   var(x)
ans = 1.2753    1.2180    2.2121    0.9585    1.8295    1.1006    1.1490
   moment(y,2)
ans = 1.3258
   var(y)
ans = moment(x,3)
ans = 1.5468
   moment(y,3)
ans = -1.1909
```

4. **样本的峰度与偏度**

峰度也称为陡峭度,它是单峰分布曲线"峰的平坦程度"的度量. 一般峰度的定义为

$$S_4 = \frac{E(X-\mu)^4}{\sigma^4}$$

注意:在不同的书中,峰度的定义不同,有的定义为 $S_4 - 3$. **由定义可知,正态分布的峰度为** 3 **;比正态分布曲线平坦的分布,其峰度大于** 3 **;比正态分布曲线陡峭的分布,其峰度小于** 3.

Matlab **中求峰度的函数是** kurtosis**:**

命令形式:kurtosis(X)

功能:求样本数据 X **的峰度. 若** X **为矩阵,则按列求每列的峰度.**

偏度是样本数据围绕其均值对称程度的度量. 若偏度为负,则数据分布偏向于其均值的左边,反之则偏向于其右边. 偏度的定义为

$$S_3 = \frac{E(X-\mu)^3}{\sigma^3}$$

正态分布的偏度为 0. Matlab 中关于偏度的函数命令形式为：

命令形式：skewness (X)

功能：求样本数据 X 的偏度．若 X 为矩阵，则按列求每列的偏度．

例 7. X = normrnd (1. 5，2. 5，10，7)

```
X =  2.2870  -0.3551   2.9224  -1.0099  -1.8732   3.8280  -0.2103
     5.1088   4.2057  -0.5543  -0.8679   0.8472   1.5281  -1.7298
     0.6226   1.1713   0.8360   0.5639   3.8837  -0.1129   1.3177
     3.0581   2.4747  -1.4694  -1.4647   1.8216   3.5143   0.6735
     3.4976   1.7200  -4.0058  -1.1398   3.1412   2.0791  -0.6091
     3.8522  -0.0887   3.9658   5.1812  -1.4195  -0.9744   2.7444
    -0.9802   0.1011   0.2034   1.6394   0.3485   4.8490   5.2212
     2.0301   2.6091   2.3184  -1.5433   0.8439   2.2238   0.1338
     2.0947  -0.8748   2.0851   1.3969  -1.5329   5.1973  -0.6169
    -1.0194   3.4530   1.5537  -1.3209  -1.7986   4.3451   0.8842
```

X = exprnd (5，10，7)

```
Y = 5.0388  2.9691   9.9262   3.2259   4.6574   4.7980   1.0146
    6.8550  0.9647   0.9567   1.6976   0.2549   2.5522  29.9376
    7.0845  0.5726   1.1692   1.2314   2.7991   2.3167   4.4690
   14.3272 12.2124   0.9931   0.3457   7.4404   1.0225   4.3528
    2.7335  0.3793  11.1182   6.5931   5.8484   0.8443   2.5957
    6.0112  7.6960   1.3943  20.3255  15.4368   1.2452   2.3921
    0.7863  1.9015   2.0190   2.8783   2.2682  21.4168   2.1481
    3.6144  6.5606   1.9957   0.1141   2.8118   3.7561   4.0099
    1.5546  1.3145   5.9027  13.5476   0.1030  19.3513   5.9520
    4.3258  5.5233   4.3128   0.6446   5.7040  11.8794   0.1310
```

Z = frnd (4，9，10，7)

```
Z = 0.6742  0.4729   1.1256   0.8491   0.9569   1.7705   1.9318
    0.5966  1.4175   0.8467   0.5931   0.5403   0.1570   2.0527
    1.6190  0.6739   0.4495   0.5290   1.1145   0.1077   1.7414
    0.7263  0.2140   0.3481   1.3562   1.3781   2.2211   0.0275
    5.6694  0.7416   2.2082   0.6834   0.7657   0.4978   0.8285
    0.8728  0.4744   0.3925   0.3668   0.5122   0.3118   2.0733
    0.1409  4.7881   0.6781   0.2286   0.4975   1.0946   3.2432
```

0. 2723 0. 8678 8. 4007 0. 7523 0. 5071 0. 5538 0. 4036

1. 0423 0. 4954 2. 0077 0. 3113 0. 2107 1. 4939 0. 1935

1. 8269 0. 6990 1. 8069 0. 2907 2. 6811 0. 3298 0. 4321

kurtosis (X)

ans = 2. 1204 1. 7240 2. 8096 4. 0361 1. 8296 2. 0417 3. 5930

kurtosis (Y)

ans = 4. 2666 2. 7992 2. 4426 3. 6906 4. 3997 2. 4386 7. 5055

kurtosis (Z)

ans = 6. 5155 7. 2449 6. 8845 3. 4648 4. 9065 2. 0378 1. 9995

skewness (X)

ans = - 0. 3118 0. 1742 - 0. 6893 1. 4059 0. 3521 - 0. 4672 1. 0894

skewness (Y)

ans = 1. 2555 0. 9404 1. 0016 1. 4403 1. 3596 1. 0498 2. 4693

skewness (Z)

ans = 2. 1626 2. 4019 2. 2789 1. 0325 1. 6315 0. 6987 0. 3801.

5. **相关系数**

若随机变量 X **与** Y **的方差** D (X) ,D (Y) **都存在,且均大于零,则称**

$$\rho = \frac{\mathrm{cov}(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

为 X **与** Y **的相关系数** .

相关系数是随机变量 X **与** Y **之间线性相依程度的度量,它满足:**$|\rho| \leqslant 1$,(1) **当** ρ **为正时,表明** X **与** Y **之间有相互促进作用,称为正相关;**(2) **当** ρ **为负时,表明** X **与** Y **之间有相互抑制作用,称为负相关** . $|\rho|$ **越接近** 1 **,表明作用越强烈;**(3) **当** $\rho = 0$ **时,表明** X **与** Y **之间不存在相互抑制作用** .

Matlab **中求相关系数的命令函数为** corrcoef.

命令形式:corrcoef (X)

功能:求矩阵 X **的相关系数矩阵,该命令把** X **的每一列作为一个随机变量,而把行作为样本点** .

例 8. x = 1 :10

Y = 2 * X + normrnd (0, 1, 1, 10)

Y = 2. 6630 3. 1458 4. 7987 7. 8801 9. 9347 12. 4853

13. 4045 15. 8503 17. 5652 19. 9207

corrcoef (X,Y)

ans = 1. 0000 0. 9957

```
        0.9957        1.0000
exprnd (3, 10, 7)
ans = 1.8393   0.9962   4.2635   1.9680   1.2180   0.2129   5.8741
      0.2048   2.1038   0.2187   3.3544   1.7555   4.8018   4.0793
      3.8712   0.6060   3.7005   1.4911   7.2072   1.4918   0.5886
      1.0305   1.4544   0.9825   1.5424   0.5936   1.6151   2.3133
      0.0607   2.1352   6.0379   2.9008   4.1743   9.2117   4.1413
      1.8473   1.7855   1.8264   0.5083   2.6781   0.2907   1.3577
      11.2784  0.3150   1.8014   0.7732   4.9370   4.3959   0.1000
      1.7053   2.3082   7.3066   4.8717   0.1990   0.9954   9.0673
      7.9290   2.6961   9.2262   1.8379   0.0812   0.8215   0.0126
      2.2155   4.5490   8.5052   4.0233   1.5786   15.5518  2.1274
corrcoef (X)
ans = 1.
```

3.3.2　参数估计

参数估计时总体分布的数学形式已知，且可以用有限个参数表示的估计问题．参数估计一般分为点估计和区间估计．参数估计一般有矩阵估计法、最小二乘估计法和极大似然函数估计法三种，最常用的参数估计法是极大似然函数估计法．

参数的点估计是指根据样本求出参数的估计值；而区间估计则是给定一可信度度量 $1-\alpha$ 和一区间，使估计值以概率 $1-\alpha$ 的机会落入给定区间，从而求出该区间端点值的估计．

Matlab 提供了常用分布的参数估计函数，这些函数均采用极大似然函数估计法计算估计量．

命令形式：binofit (X, N, a)

功能：计算二项分布的参数估计．其中 X 为样本数据点，N 为二项分布的实验次数，a 为纳伪概率．

命令形式：expfit (X, a)

功能：计算指数分布的参数估计．其中 X 为样本数据点，a 为纳伪概率．

命令形式：normfit (X, a)

功能：计算正态分布的参数估计．其中 X 为样本数据点，a 为纳伪概率．

命令形式：poissfit (X, a)

功能：计算泊松分布的参数估计．其中 X 为样本数据点，a 为纳伪概率．

命令形式：unifit（X，a）

功能：计算均匀分布的参数估计．其中 X **为样本数据点，**a **为纳伪概率．**

注意　上述命令中，参数 a **可以省略，省略时默认** a = 0.05. **各命令的具体用法见下述例子．**

例 9. **求二项分布的参数估计．**

解　X = binornd（10，0.6，1，10）

X = 5　8　3　5　8　5　10　5　8　7

[d i] = binofit（X，10）

d = 0.5000　0.8000　0.3000　0.5000　0.8000　0.5000　1.0000　0.5000　0.8000　0.7000

i =

Columns 1 through 10

0.1871　0.4439　0.0667　0.1871　0.4439　0.1871　0.6915　0.1871　0.4439　0.3475

Columns 11 through 20

0.8129　0.9748　0.6525　0.8129　0.9748　0.8129　1.0000　0.8129　0.9748　0.9333

[d i] = binofit（X，10，0.01）

d =

0.5000　0.8000　0.3000　0.5000　0.8000　0.5000　1.0000　0.5000　0.8000　0.7000

i =

Columns 1 through 10

0.1283　0.3518　0.0370　0.1283　0.3518　0.1283　0.5887　0.1283　0.3518　0.2649

Columns 11 through 20

0.8717　0.9891　0.7351　0.8717　0.9891　0.8717　1.0000　0.8717　0.9891　0.9630

上述输出结果中 d **为参数的点估计，**i **为参数的区间估计，**i **的第一行为区间的左边界，第二行为区间的右边界．**

例 10. **求泊松分布的参数估计．**

解　X = poissrnd（2.5，10，7）

X = 1　1　1　4　5　3　3

1　3　5　1　2　4　4

```
     3     3     3     5     2     3     2
     6     1     2     3     2     3     3
     7     3     0     0     2     2     1
     3     2     2     1     5     0     2
     4     0     3     1     3     2     3
     1     0     3     4     3     1     1
     4     1     3     3     5     1     2
     3     2     1     1     2     7     4
[d i] = poissfit (X)
d = 3.3000   1.6000   2.3000   2.3000   3.1000   2.6000   2.5000
i = 2.2716   0.9145   1.4580   1.4580   2.1063   1.6984   1.6179
    4.6344   2.5983   3.4511   3.4511   4.4002   3.8096   3.6905
[d i] = poissfit (X, 0.01)
d = 3.3000   1.6000   2.3000   2.3000   3.1000   2.6000   2.5000
i = 2.0079   0.7567   1.2521   1.2521   1.8534   1.4741   1.3995
    5.0888   2.9482   3.8484   3.8484   4.8439   4.2251   4.1000
```

上述输出结果中，d 为参数的点估计，i 为参数的区间估计.

例 11. 求正态分布的参数估计.

解　X = normrnd (4, 2.5, 10, 7)

```
X = 7.8379   -1.1358   2.9267   3.0255   3.4722   5.1551   -1.3011
    2.4838   4.3314   4.1395   0.5468   6.9756   3.1975   2.3883
    0.6316   7.9824   3.0803   4.7889   1.2095   7.0914   2.2392
    5.1735   6.5460   2.8376   7.8831   5.5882   2.4218   1.4547
    1.7411   0.0490   4.9274   5.7697   2.4965   -1.8130  3.5448
    4.0897   3.8033   5.8207   8.8935   5.3780   0.9209   7.8025
    2.4312   2.2959   9.2804   5.2614   1.2504   6.6391   3.9039
    5.3385   1.4386   0.6068   8.6613   4.2150   3.7169   7.0686
    5.3822   0.9141   1.4435   3.1505   -1.0114  4.9481   2.2595
    3.4908   4.7220   6.5946   1.1506   2.7673   6.3605   4.0188
[a, b, c, d] = normfit (X)
a = 3.8600   3.0947   4.1657   4.9131   3.2341   3.8638   3.3379
b = 2.1367   2.9019   2.5806   2.9716   2.4031   2.8014   2.6461
c = 2.3315   1.0188   2.3197   2.7873   1.5150   1.8599   1.4450
    5.3885   5.1706   6.0118   7.0389   4.9532   5.8678   5.2308
```

d = 1.4697 1.9960 1.7750 2.0440 1.6530 1.9269 1.8201
3.9008 5.2978 4.7111 5.4250 4.3872 5.1142 4.8307

[a, b, c, d] = normfit (X, 0.01)

a = 3.8600 3.0947 4.1657 4.9131 3.2341 3.8638 3.3379
b = 2.1367 2.9019 2.5806 2.9716 2.4031 2.8014 2.6461
c = 1.6642 0.1124 1.5137 1.8592 0.7644 0.9849 0.6186
6.0559 6.0770 6.8177 7.9670 5.7038 6.7427 6.0573
d = 1.3198 1.7925 1.5940 1.8355 1.4844 1.7303 1.6344
4.8665 6.6094 5.8775 6.7682 5.4734 6.3804 6.0268

[a, b, c, d] = normfit (X(:), 0.01)

a = 3.7813
b = 2.5988
c = 2.9585
4.6042
d = 2.1271
3.3116.

上述输出结果中,有四个输出参数,a 是 μ 的点估计,b 是 σ 的点估计,c 是 μ 的区间估计,d 是 σ 的区间估计.

例 12. **求均匀分布的参数估计.**

解 X = unifrnd (3, 8, 2, 3)

X = [3.9651 3.0648 7.3768; 5.2268 4.5437 7.1763]

[a, b, c, d] = unifit (X, 0.05)

a = 3.9651 3.0648 7.1763
b = 5.2268 4.5437 7.3768
c = [-0.4156 -2.0702 6.4803; 3.9651 3.0648 7.1763]
d = [5.2268 4.5437 7.3768; 9.6075 9.6787 8.0728]

[a, b, c, d] = unifit (X(:), 0.05)

a = 3.0648
b = 7.3768
c = [0.2726; 3.0648]
d = [7.3768; 10.1690]

上述结果中,输出参数有四个,其中 a 和 b 为均匀分布的两个参数的点估计,c 和 d 分别是 a 和 b 的区间估计.

3.3.3　假设检验

假设检验是统计推断的基本问题之一，是确定关于样本总体特征的判断是否合理的过程，假设检验就是利用样本提供的信息，对所作的初始假设：零假设或称为无效假设，按一定的规则判定是否成立，以决定是接受还是拒绝零假设．假设检验的判断和结论是根据样本作出的，故具有“概率性”，所作的结论可能会犯两种错误：弃真和纳伪．一般情况下，伴随着无效假设 H_0 的确定，总能写出备择假设 H_A，备择假设也称对立假设，即拒绝 H_0 一定会接受 H_A，反过来也是一样．这种拒绝或接受是在一定概率意义下作出的，因而一般必须预先给定显著性水平 α，即 α 为犯错误的概率．假设检验的置信区间是指检验的假设量落入该区间就接受该假设，否则拒绝假设．

1. 正态总体参数的假设检验

对于正态总体来讲，一般可以分为方差 σ^2 已知和未知两种情况．当方差已知时，即假设数据来自同一正态分布总体的独立样本，可以用 Z 检验；当方差未知时，需要根据数据样本对方差作出估计，这时用估计出来的方差代替总体方差，可以用 t 检验．这两种检验具有相关的变异度

$$Z = \frac{\bar{X} - \mu}{\sigma}, \quad T = \frac{\bar{X} - \mu}{s}$$

其中

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

若零假设为真，则 Z 服从标准正态分布，T 服从自由度为 n - 1 的 t 分布．Matlab 中有如下检验命令：

命令形式：ttest（X，m，a，t）

功能：计算单样本均值的 t 检验．其中 X 为正态总体样本，m 为欲检验的均值，a 为显著水平，t 为备择假设选项．t 只有三个值：0，1 或 - 1，其中 0 表示“≠”，1 表示“>”，- 1 表示“<”．缺省时 t = 0.

命令形式：ttest2（X，Y，a，t）

功能：计算双样本均值差异的 t 检验．其中 X 和 Y 为正态总体的样本，a 为显著性水平；t 的含义同上．

命令形式：ztest（X，m，s，a，t）

功能：计算已知方差的单样本 Z 检验．其中 X 为正态总体样本，m 为欲检验的均值，a 为显著性水平，t 的含义同上．

下面分别举例说明这三个命令的用法．

例 13. ttest 的用法举例．

单样本均值的 t 检验主要用于判断在方差未知的情况下,给定的总体样本均值是否为确定值,即 $H_0: \mu = \mu_0$,命令中的 m 即为这里的 μ_0,可以用随机函数产生一个正态样本,求其单样本均值的 t 检验. 在 Matlab 中的程序如下:

```
x = normrnd (0, 1, 1, 100)
mean (x)        ans = 0.0479
var (x)         ans = 0.7543
[a  b  c] = ttest (x, 0, 0.05, 0)
a = 0      b = 0.5823      c = [-0.1244, 0.2203]
[a b c] = ttest (x, 0, 0.05, 1)
a = 0      b = 0.2911      c = [-0.0963, 0.1921]
[a  b  c] = ttest (x, 0, 0.05, -1)
a = 0      b = 0.7089      c = [-0.0963, 0.1921]
[a  b  c] = ttest (x, 0.7, 0.05, -1)
a = 0      b = 0.7089      c = [-0.0963, 0.1921].
```

上述结果中有三个输出项:其中 a 为 0 表示接受 H_0,拒绝 H_A;a 为 1 表示拒绝 H_0,接受 H_A;b 与 T 统计量有关,b 表示假设 X 的均值等于 μ 时,T 的观察值较大的概率;c 是样本均值的 $1-\alpha$ 置信区间.

例 14. ttest2 的用法举例.

对于两个正态总体样本均值的差异性检验,在 Matlab 中的程序如下:

```
x = normrnd (2, 2, 1, 100)
y = normrnd (4, 2, 1, 100)
mean (x)        ans = 1.9803
mean (y)        ans = 3.7048
var (x)         ans = 3.9819
var (y)         ans = 3.7316
[a  b  c] = ttest2 (x, y, 0.01, 0)
[a  b  c] = ttest2 (x, y, 0.05, 0)
[a  b  c] = ttest2 (x, y, 0.01, -1)
[a  b  c] = ttest2 (x, y, 0.05, -1)
[a  b  c] = ttest2 (x, y, 0.01, 1)
[a  b  c] = ttest2 (x, y, 0.05, 1)
```

输出参数的意义:其中 a 为 0 表示接受 H_0,拒绝 H_A;a 为 1 表示拒绝 H_0,接受 H_A;b 与 T 统计量有关,b 表示假设 X 的均值等于 μ 时,T 的观察值较大的概率;c 表示两样本均值差,即 $\bar{X}-\bar{Y}$ 的置信区间.

当方差已知时,要检验来自正态总体样本均值与某一数值有无显著差异,可以用命令 ztest, ztest 的用法如下.

例 15. ztest 的用法举例.

```
x = normrnd (4, 2, 1, 100)
mean (x)        ans = 3.8646
var (x)         ans = 3.8239
[a  b  c] = ztest (x, 4, 2, 0.05, 0)
a = 1
b = 1.1268e - 089
c = - 0.4078        0.3762
[a  b  c] = ztest (x, 4, 2, 0.01, 0)
a = 1    b = 1.1268e - 089    c = - 0.5310        0.4994
[a  b  c] = ztest (x, 4.5, 2, 0.05, 0)
a = 1
b = 6.9737e - 113
c = - 0.4078        0.3762
[a  b  c] = ztest (x, 4, 2, 0.05, - 1)
a = 1
b = 5.6340e - 090
c = - Inf        0.3132
[a  b  c] = ztest (x, 4, 2, 0.05, 1)
a = 0
b = 1
c = - 0.3448        Inf
[a  b  c] = ztest (x, 4.5, 2, 0.01, 1)
a = 0
b = 1
c = - 0.4811        Inf
```

输出参数的意义:其中 a 为 0 表示接受 H_0,拒绝 H_A;a 为 1 表示拒绝 H_0,接受 H_A;b 与 T 统计量有关,b 表示假设 X 的均值等于 μ 时,T 的观察值较大的概率;c 是样本均值的 $1-\alpha$ 置信区间.

2. 非参数假设检验

参数检验方法一般用于服从正态分布或近似服从正态分布的总体. 如果对总体分布服从何种分布不确定,则需要用到非参数检验方法. Matlab 中有三个

非参数检验函数：

命令形式 1：ranksum（X，Y，a）

功能：对变量进行威尔科克秩和检验.

命令形式 2：signran（X，Y，a）

功能：对变量进行威尔科克符号秩检验.

命令形式 3：signtest（X，Y，a）

功能：计算成对样本的符号检验.

其中 X 和 Y 为两个总体样本，a 为指定的显著性概率水平.

秩和检验法的目的是检验两个总体 X 和 Y 的分布是否相同，调用格式为

[p，h] = ranksum（X，Y，a）

返回参数中，p 为总体 X 和 Y 分布相同的概率. h = 0 表示接受分布相同的假设；h = 1 表示拒绝分布相同的假设，即 X 和 Y 的分布明显不同.

符号秩检验法的目的是检验两个配对样本 X 和 Y 的中位数是否相同的假设，这里 X 和 Y 的样本容量必须相同，符号秩检验的调用格式为

[p，h] = signrank（X，Y，a）

返回参数中，p 为中位数相等的概率，h = 0 表示接受中位数相等的假设，h = 1 表示拒绝中位数相等的假设.

成对样本的符号检验法是用以检验两个总体样本中位数相等的假设. 若 X 和 Y 均为样本，则其样本数必须相等. 调用格式为

[p，h] = signrank（X，Y，a）

返回参数 p 和 h 含义与上述相同.

§3.4 方差分析与回归分析

3.4.1 方差分析

方差分析法是基于不太多的统计数据，定量地分析一个或多个因素对某个（些）响应变量的影响和作用的显著性，这种显著性是基于一定概率条件下而言的，其前提条件是在各因素的作用下，响应变量的分布具有正态性和等方差性.

1. 单因素方差分析

单因素方差分析的基本问题是比较和估计多个等方差正态总体的均值，其基本模型为

$$y_{ij} = \alpha_j + \varepsilon_{ij}$$

其中 y_{ij}为数据观测值，α_j 为各正态总体的数学期望，ε_{ij}为各数据的随机误差．在 Matlab 中，单因素方差分析函数为 anoval，其使用格式为：

命令形式 1：p = anoval (X)

功能：返回无效假设成立的概率，其中 X 为 r × n 的数据矩阵．函数将矩阵的每一列当做一个整体，矩阵的行数即为样本重复数．若函数返回的概率值接近于零，则无效假设值得怀疑，表明各列的均值事实上是不同的．

命令形式 2：p = anoval (X，g)

功能：返回无效假设成立的概率，其中 X 为一向量，g 是与 X 同长度的向量，且 g 中的值为整数，最小值为 1，最大值为数据的组数，每一组至少有一个数，但并不要求每组中元素个数相同．

注意：

(1) 单因素方差分析的第一种格式用于等重复的单因素方差分析；第二种格式用于重复数不等的单因素方差分析．

(2) 方差分析同时还显示一幅图与一张表，表即为方差分析表，它与一般的方差分析表一样；而图则给出了各列数据的 box 图，这种图为一“盒子”形状，称为 box 图，其特征为：1）盒子的上底与下底间为内四分位间距；盒子的上、下两条线分别为样本的 25% 和 75% 分位数；2）盒子的中间线为样本的中位数，如果中位数不在盒子中间，表明样本存在一定偏度；3）虚线贯穿盒子上下，显示了样本其余部分，如果没有奇异值，则样本的最大值为虚线顶点，最小值为虚线底端．默认奇异值为距盒子底端和顶端超过 1.5 倍内四分位间距的点．在图中，奇异值为超出虚线底端的点，用“ + ”表示一个奇异值；4）切口是样本中位数的置信区间，可以用 box 图对样本进行多重比较．

例 16. 为考察 5 名工人的劳动生产率是否相同，记录了每人 4 天的产量，并计算出其平均值，数据如表 3.1 所示．试从这些数据推断出他们的生产率有无显著差别．

表 3.1　工人各天的生产量及其平均值

天＼工人	W_1	W_2	W_3	W_4	W_5
1	256	254	250	248	236
2	242	330	277	280	252
3	280	290	230	305	220

续表

天 \ 工人	W_1	W_2	W_3	W_4	W_5
4	298	295	302	289	252
平均生产量	269.0	292.25	264.75	280.50	240.00

解　考察工人生产率的程序如下：

% **首先把数据排列成** 4 × 5 **矩阵形式**

$$X = \begin{bmatrix} 256 & 254 & 250 & 248 & 236 \\ 242 & 330 & 277 & 280 & 252 \\ 280 & 290 & 230 & 305 & 220 \\ 298 & 295 & 302 & 289 & 252 \end{bmatrix}$$

% **用** anoval **进行方差分析**

p = anova1 (X)

p = 0.1109

ANOVA TABLE

Source	SS	df	MS	F	Prob > F
Columns	6125.7	4	1531.43	2.26	0.1109
Error	10156.5	15	677.1		
Total	16282.2	19			

运算结果如图 3.5 **所示**.

由于返回的概率值 p = 0.1109 > 0.05**，故接受** H_0**，即** 5 **名工人的生产率没有显著差异**.

例 17. **将三种不同菌型的伤寒病毒分别接种于** 10 **只、**9 **只和** 11 **只小白鼠身上，观察存活天数，结果以** X **为** g **表示，**X **为存活天数，**g **为组标识，表明数据** X **属于哪一组的值**. **试分析不同菌型的伤寒病毒对于小白鼠存活的影响是否显著**.

解　在 Matlab **中的程序如下**

X = [2 4 3 2 4 5 5 2 5 4 5 6 8 5 10 7 12 6 6 7 11 6 6 7 9 5 10 6 3 10]；

g = [1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3]；

p = anova1 (X,g)

p = 5.6835e - 004

ANOVA TABLE

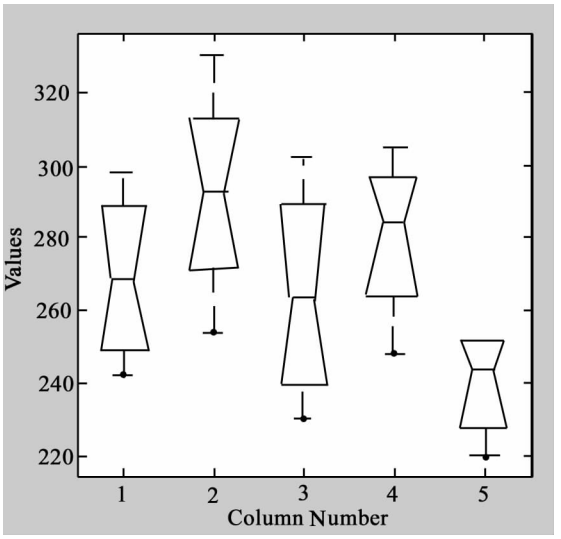


图 3.5

Source	SS	df	MS	F	Prob > F
Groups	88.829	2	44.4146	9.98	0.0006
Error	120.137	27	4.4495		
Total	208.967	29			

运算结果如图 3.6 所示.

由方差分析结果 p = 5.6835e - 004 < 0.01 知,不同菌型的伤寒病毒对小白鼠存活天数的影响差异极显著.

2. 双因素方差分析

双因素方差分析是一种两因素多水平检验实验数据的统计分析方法,目的在于确定来自不同组的数据是否具有相同的均值. 其基本模型为

$$y_{ijk} = \mu + \alpha_j + \beta_i + \gamma_{ij} + \varepsilon_{ijk}$$

其中 y_{ijk} 为数据观测值,μ 为样本总均值,α_j 为因素 A 各列的均值,β_i 为因素 B 各行的均值,γ_{ij} 是因素 A 与因素 B 的交互作用,ε_{ijk} 为观测值的随机误差.

在 Matlab 中,计算等重复实验的双因素方差分析的函数为 anova2,它的使用方式如下:

命令形式:p = anova2 (X, r)

功能:计算双因素方差分析,其中 X 为数据观测值 (因素 A 按列放,因素 B

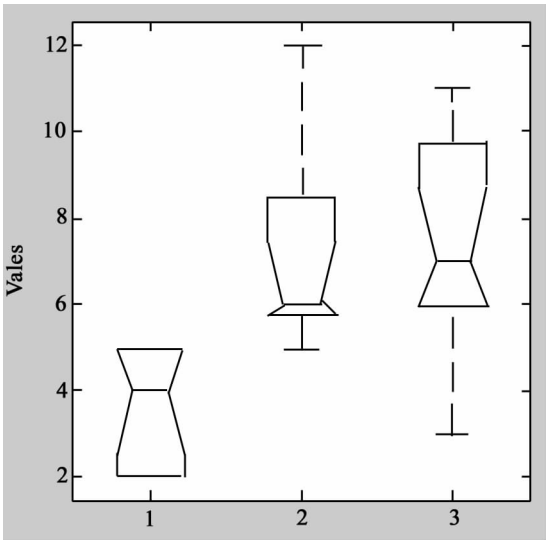


图 3.6

按行放,即矩阵 X 的列为因素 A 的水平. 当重复数为 1 时,矩阵 X 的行为因素 B 的水平;当重复数大于 1 时,前面 r 行为因素 B 的第一水平,r + 1 行到 2r 行为因素 B 的第二水平,依次类推,r 为重复数. 矩阵 X 的行数应为因素 B 的水平数乘以重复数 r).

函数返回值为一向量 p,当 r = 1 时,p 有两个元素,第一个为因素 A 各水平均值相等的概率,第二个为因素 B 各水平均值相等的概率;当 r > 1 时,p 中有三个元素,第三个元素为因素 A 与因素 B 交互无显著作用的概率. 该函数除返回概率值外,还显示一个方差分析表.

例 18. 三个学生对四个品种的玉米含氮量 (mg) 各作了一次分析,数据为下面矩阵 X,每行为一个学生的分析数据,每列为一种玉米的数据,求其方差分析并分析结果.

解　在 Matlab 中的程序如下

$$X = \begin{bmatrix} 2.2000 & 2.3000 & 2.6000 & 2.7000 \\ 2.2000 & 2.0000 & 2.5000 & 2.7000 \\ 2.0000 & 2.3000 & 2.7000 & 2.8000 \end{bmatrix}$$

```
p = anova2(X, 1)
p = 0.0021    0.4472
```

ANOVA　Table

Source	SS	df	MS	F	Prob > F
Columns	0. 78333	3	0. 26111	18. 08	0. 0021
Rows	0. 02667	2	0. 01333	0. 92	0. 4472
Error	0. 08667	6	0. 01444		
Total	0. 89667	11			

由方差分析结果知，因素 A（列）均值相等的概率为 p = 0. 0021 < 0. 01，故差异极显著；而因素 B（行）均值相等的概率为 0. 4472 > 0. 05，因而差异不显著.

例 19. 为分析 4 种化肥和 3 个水稻品种对水稻产量的影响，把一块试验田等分为 24 小块，对种子和化肥的每一组合种植两小块田. 产量如表 3. 2 所示，试分析品种、化肥及二者的交互作用对水稻产量有无显著影响.

表 3. 2　　水稻产量试验数据表

品种＼化肥	A_1	A_2	A_3	A_4
B_1	173，172	174，176	177，199	172，173
B_2	175，173	178，177	174，175	170，171
B_3	177，175	174，174	174，173	169，169

解　%把数据写成矩阵形式，注意数据的排列方式为：将双因素试验数据表处于（A_i，B_j）格子内的数据按列的方向排序

$$X = \begin{bmatrix} 173 & 174 & 177 & 172 \\ 172 & 176 & 179 & 173 \\ 175 & 178 & 174 & 170 \\ 173 & 177 & 175 & 171 \\ 177 & 174 & 174 & 169 \\ 175 & 174 & 173 & 169 \end{bmatrix}$$

p = anova2（a，2）

运算结果为

p = 0. 0000　　0. 0367　　0. 0006

方差表为：

ANOVA　Table

Source	SS	df	MS	F	Prob > F
Columns	90.833	3	30.2778	33.03	0
Rows	8.083	2	4.0417	4.41	0.0367
Interaction	51.917	6	8.6528	9.44	0.0006
Error	11	12	0.9167		
Total	161.833	23			

p 的三个数值分别为因素 A(列)、因素 B(行)和交互因素 A×B 的概率. 显然,因素 A 和交互因素 A×B 的影响非常显著. 而 0.01 < 0.0367 < 0.05,因素 B 的影响也显著. 方差分析的结果就是要注意化肥与品种的交互作用.

3.4.2 回归分析

回归分析是研究变量之间关系的一种统计方法,即利用统计数据求变量间关系的近似表达式,并利用所得公式进行统计描述、分析和推断,一般情况下都是进行线性回归分析.

线性回归包括一元线性回归和多元线性回归,这种回归关于未知数和回归变量都是线性关系,线性回归的变量关系为

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \varepsilon$$

其中 β_i 是回归系数,ε 为随机误差;当 m = 1 时为一元线性回归,当 m > 1 时为多元线性回归.

Matlab 中求多元线性回归的函数是 regress,其用法为:

命令形式:[b, bint, r, rint, stats] = regress (Y, X, a)

功能:计算多元线性回归. 其中 Y 为观测到的随机变量,X 为自变量矩阵. 若回归关系中包括常数项,则 X 的第一列应全部为 1,X 与 Y 的行数应相等,X 的列数等于参数的个数;当 X 为两列且第一列全为 1 时为一元线性回归. a 为输出各种置信区间用的显著性水平.

共有 5 个输出结果:b 为参数的点估计;bint 为参数的区间估计;r 为残差的点估计;rint 为残差的区间估计,当点估计落在区间估计之外时,拒绝无效假设;stats 中包含了三个项,第一个是回归方程的决定系数 R^2,第二个是回归方程的 F 统计量,第三个是拒绝无效假设的概率.

例 20. 对某地区生产同一产品的 8 个不同规模的企业进行生产费用调查,调查得到的产量 X 和生产费用 Y 的数据如下,试对其进行回归分析.

$$X' = \begin{matrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.5 & 2.0 & 3.0 & 4.5 & 7.5 & 9.1 & 10.5 & 12.0 \end{matrix}$$

$$Y' = \begin{matrix} 5.6 & 6.6 & 7.2 & 7.8 & 10.1 & 10.8 & 13.5 & 16.5 \end{matrix}$$

解　X' = [1.0　1.0　1.0　1.0　1.0　1.0　1.0　1.0
　　1.5　2.0　3.0　4.5　7.5　9.1　10.5　12.0]；
　Y' = [5.6　6.6　7.2　7.8　10.1　10.8　13.5　16.5]；
[b bint r rint stat] = regress (Y, X, 0.05)

运行结果为

```
b = 9.7625
bint = 6.6296        12.8954
r = -4.1625          rint = -12.1630      3.8380
    -3.1625                 -11.5788      5.2538
    -2.5625                 -11.1669      6.0419
    -1.9625                 -10.7127      6.7877
     0.3375                 -8.6095       9.2845
     1.0375                 -7.8592       9.9342
     3.7375                 -4.4562       11.9312
     6.7375                  0.5851       12.8899
stat = 1.0e-030 *
     0.2568          NaN          NaN
```

[b2　int2　r2　rint2　stat2] = regress (Y, X, 0.01)

运行结果为

```
int2 = 5.1261        14.3989
b2 = 9.7625
r2 = -4.1625         rint2 = -16.0027        7.6777
     -3.1625                 -15.6181        9.2931
     -2.5625                 -15.2964        10.1714
     -1.9625                 -14.9122        10.9872
      0.3375                 -12.9034        13.5784
      1.0375                 -12.1290        14.2040
      3.7375                 -8.3886         15.8636
      6.7375                 -2.3676         15.8426
stat2 = 1.0e-030 *
     0.2568          NaN          NaN.
```

注意　Matlab 还提供了一个逐步回归的函数 stepwise ()，该函数是交互形式的图形工具函数，可以试着用该函数进行分析.

习 题 4

1. 某市电子工业公司有 14 个所属企业，各企业的年设备能力与年劳动生产率统计数据如表 3.3 所示．试分析企业年设备能力与年劳动生产率的关系．若该公司计划建一个设备能力为 9.2 千瓦/人的企业，试问估计劳动生产率将为多少？

表 3.3

企业	设备能力/(千瓦/人)	劳动生产率/(千元/人)	企业	设备能力/(千瓦/人)	劳动生产率/(千元/人)
1	2.8	6.7	8	4.8	9.8
2	2.8	6.9	9	4.9	10.6
3	3.0	7.2	10	5.2	10.7
4	2.9	7.3	11	5.4	11.1
5	3.4	8.4	12	5.5	11.8
6	3.9	8.8	13	6.2	12.1
7	4.0	9.1	14	7.0	12.4

2. 对经营同一类产品出口业务的 13 家公司进行抽样调查，其出口换汇成本与商品流转费用率资料如表 3.4 所示．试分析两个变量之间的关系，并估计某家公司商品流转费用率是 6.5% 的出口换汇成本．

表 3.4

公司	出口换汇成本/(人民币元/美元)	商品流转费用率/%	公司	出口换汇成本/(人民币元/美元)	商品流转费用率/%
1	1.40	4.20	8	1.60	5.50
2	1.20	5.30	9	2.00	4.10
3	1.00	7.10	10	1.00	5.00
4	1.90	3.70	11	1.60	4.00
5	1.30	6.20	12	1.80	3.40
6	2.40	3.50	13	1.40	6.90
7	1.40	4.80			

3. 表 3.5 是统计的某些地区不同行业的职工工资水平及年人均收入. 试分析：

(1) 不同行业的职工工资水平是否有显著差异？

(2) 不同地区、不同行业的职工工资水平是否有显著差异？

表 3.5

地　区	工业	建筑业	交通运输、邮电通讯业	教育、文化艺术和广播电视事业
上　海	1327	1737	1529	1244
重　庆	1161	1631	1503	1162
湖　北	1113	1428	1245	1026
江　西	1210	1493	1297	1108
贵　州	1246	1499	1332	1199

4. 表 3.6 是某种化工过程在三种浓度、四种温度下得率的数据：

表 3.6

温度 /(°) / 浓度 / %	10		24		38		52	
2	14	10	11	11	13	9	10	12
4	9	7	10	8	7	11	6	10
6	5	11	13	14	12	13	14	10

在化工过程中，希望得率指标越高越好，试分析浓度、温度及交互作用对试验指标(得率)是否有显著影响，并找出最佳水平组合状态.

5. 某人射击中射中的概率为 P，若射击直到中靶为止，令射击次数为 k. 试求：

(1) P(X = k)；

(2) 绘出 k = 0，1，…，20 的概率曲线图(p = 0.5，0.2，0.8)；

(3) 求 E(X)；

(4) 求使 P(X = k) 最大的 k.

6. 试分别画出 $\chi^2(5)$，$\chi^2(10)$，t(20)，t(2)，N(1，9) 的分布密度函数曲线和概率密度函数曲线.

7. 调查了 339 名 50 岁以上人群中吸烟习惯与患慢性气管炎的关系后，结

果如表 3.7 所示.

表 3.7

是否吸烟 / 是否患病	吸　烟	不吸烟	总　和
患慢性气管炎	43	13	56
未患慢性气管炎	162	121	283
总和	205	134	339
患病率	21.0%	9.7%	16.5%

试问吸烟习惯与慢性气管炎是否有关？

8. 表 3.8 中的数据是某建筑公司今年 20 个地区的销售量（Y,千方）,推销开支、实际账目、同类商品竞争数和地区销售潜力分别是影响建筑材料销售量的因素. 试建立回归模型,并分析哪些是主要的影响因素.

表 3.8

地区	推销开支 (X_1)	实际账目数 (X_2)	同类商品竞争数 (X_3)	地区销售潜力 (X_4)	销售量 Y
1	5.5	31	10	8	79.3
2	2.5	55	8	6	200.1
3	8.0	67	12	9	163.2
4	3.0	50	7	16	200.1
5	3.0	38	8	15	146.0
6	2.9	71	12	17	177.7
7	8.0	30	12	8	30.9
8	9.0	56	5	10	291.9
9	4.0	42	8	4	160.0
10	6.5	73	5	16	339.4
11	5.5	60	11	7	159.6
12	5.0	44	12	12	86.3
13	6.0	50	6	6	237.5
14	5.0	39	10	4	107.2
15	3.5	55	10	4	155.0
16	8.0	70	6	14	201.4
17	6.0	40	11	6	100.2
18	4.0	50	11	8	135.8
19	7.5	62	9	13	223.3
20	7.0	59	9	11	195.0

第四章　最优化方法及其在 Matlab 中的实现

优化模型的一般形式为

$$\min_{X} \quad z = f(X), X = (x_1, x_2, \cdots, x_n)^T \tag{4.1}$$

$$\text{s.t. } g_i(X) \leqslant 0, i = 1, 2, \cdots, m \tag{4.2}$$

其中 $f(x)$ 称为目标函数，$g_i(X)$ 称为约束条件．只满足式 (4.2) 的 X 称为可行解；同时满足式 (4.1)、式 (4.2) 两式的解 $X = X^*$ 称为最优解．

由式 (4.1)、式 (4.2) 组成的模型属于约束优化，若只有式 (4.1) 就是无约束优化．一般情况下，优化问题都是有约束的，但是如果最优解不是在可行域的边界上，而是在可行域的内部，那么就可以用无约束优化作比较简单的处理．

若 f, g_i 均为线性函数，优化模型式 (4.1)、式 (4.2) 称为线性规划，否则称为非线性规划．

§4.1　线性规划

4.1.1　线性规划

线性规划是运筹学的一个重要分支，应用很广．线性规划问题可以描述为求一组非负变量，这些非负变量在一定线性约束的条件下，使一个线性目标函数取得极小 (或极大) 值的问题．

1. 线性规划的标准形式

目标函数　$\min z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$

约束条件
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \qquad \vdots \qquad\qquad \vdots \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\ x_1, x_2, \cdots, x_n \geqslant 0 \end{cases}$$

这里 $x_1, x_2, \cdots, x_n$ 是变量，c_i, a_{ij}, b_i 都是已知常数，且 $b_i \geqslant 0$，约束条件常用

s. t. 表示. 线性规划用矩阵表示就是

$$\min z = cX, \quad X = (x_1, x_2, \cdots, x_n)^T$$

s. t. $AX = b, x \geqslant 0, A = (a_{ij})_{m \times n}\ (m \leqslant n), b = (b_1, b_2, \cdots, b_n)^T$.

2. 线性规划的一般形式

目标函数 $\min z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$

约束条件
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \ \bigcirc\ b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \ \bigcirc\ b_2 \\ \vdots \qquad \vdots \qquad\qquad \vdots \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \ \bigcirc\ b_n \end{cases}$$

式中的 ○ 可以是关系符号：$>, <, =, \geqslant, \leqslant$ 中的任意一个.

3. 线性规划化为标准形的方法和基本性质

把线性规划化为标准形：

(1) 目标函数一律化为求极小(如果是求极大,则利用 $\max z \Leftrightarrow \min(-z)$ 化为求极小).

(2) 对约束条件中 $Ax \leqslant b$ 的不等式,利用加入松弛变量的方法化为等式. 如果原约束条件中有"$\geqslant b$"形式的约束,可以在不等式两边同时加负号化为"$\leqslant -b$"的形式.

(3) 标准形中一般要求 $x_i \geqslant 0$. 如果某个 x_i 无此约束,可以引入两个新变量 x_i', x_i'',令 $x_i = x_i' - x_i''$, $x_i', x_i'' \geqslant 0$;如果原来的约束为 $x_i \geqslant l_i$,可以令 $x_i' = x_i - l_i$, $x_i' \geqslant 0$.

4. 线性规划的基本性质

线性规划有以下基本性质：

(1) 若存在可行域,可行域必为凸集；

(2) 基可行解对应于可行域的顶点；

(3) 若有最优解,必在可行域的顶点取得.

4.1.2 线性规划在 Matlab 中的求解方法

用 Matlab 优化工具箱求解线性规划必须化为如下形式

$$\min z = cX, \tag{4.3}$$
$$\text{s. t.} \quad AX \leqslant b,$$

求解线性规划有如下一些命令格式：

命令形式 1：X = lp(c,A,b)

命令形式 2：X = lp(c,A,b,v1)

命令形式 3：X = lp (c,A,b,v1,v2)

命令形式 4：X = lp (c,A,b,v1,v2,x0)

命令形式 5：X = lp (c,A,b,v1,v2,x0,ne)

命令形式 6：X = lp (c,A,b,v1,v2,x0,ne,dis)

命令形式 7：[X,lag] = lp (c,A,b,…)

命令形式 8：[X,lag,how] = lp (c,A,b,…)

其中输入参数 c,A,b 如式 (4.3) 所示,输出 X 为最优解;v1,v2 表示 X 的下界和上界,即有约束 v1≤X≤v2 (v1 或 v2 的维数可以小于 X 的维数 n,这时 v1 或 v2 仅表示 X 前 k 个分量的下界或上界);x0 表示初始解 (缺省时程序自动取 x0 = 0);ne 为等式约束的个数,且需将等式约束置于不等式约束的前面;dis 给出警告信息,如解无界或不可行等.

输出参数 lag 是拉格朗日 (Lagrange) 乘子,维数等于约束条件的个数,其非零分量对应于起作用的约束条件;how 给出错误信息:无可行解 (infeasible),无有界解 (unbounded) 或问题顺利求解 (ok).

例 1. 求解

$$\min z = x_1 + x_2$$
$$\text{s.t.}\quad x_1 - x_2 \leqslant 1,\quad x_1 \geqslant 0.$$

解　在 Matlab 中可以用两种方法输入

```
c = [1,1]
a = [1,-1]
b = 1
v1 = 0
X = lp (c,a,b,v1)
```

或把线性规划写成矩阵形式

$$z = x_1 + x_2 = (1,1)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\text{s.t.}\quad \begin{pmatrix} 1 & -1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leqslant \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Matlab 的命令为

```
c = [1,1]
a = [1,-1;-1,0]
b = [1,0]
X = lp (c,a,b)
```

两种方法得到同样的结果 X = (0,1). 在前一种方法中如果不输入 v1,将导致错误.

例 2. 求线性规划问题

$$\min\ m = 13x - y + 5z$$

$$\text{s. t.}\quad \begin{cases} x + y \geqslant 7 \\ y + z < 10 \\ x > 2 \\ y > 0 \\ z > 0 \end{cases}$$

解 (1) 把线性规划问题写成矩阵形式

$$m = 13x - y + 5z = (13, -1, 5)\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\text{s. t.}\quad \begin{pmatrix} -1 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} \leqslant \begin{pmatrix} -7 \\ 10 \end{pmatrix}, x > 0, y > 0, z > 0$$

(2) Matlab 命令为

```
c = [13,-1,5], A = [-1,-1,0;0,1,1], b = [-7,10]
v0 = [2,0,0]
X = (lp(c,A,b,v0))'
```

其计算结果为:

X = 2 10 0

当 X = (2,10,0) 时,取得极小值 z = 16.

例 3. 现有三种食品 A_1, A_2, A_3,各含有两种营养成分 B_1, B_2,每单位食品 A_i 含有 B_j 成分的数量及每种食物的单价如表 4.1 所示.

表 4.1

种类 / 成分	A_1	A_2	A_3	营养成分需要量
B_1	2	0	4	5
B_2	2	3	1	4
单 价	4	2	3	

试问应如何选购食物，才能既满足对营养成分 B_1，B_2 的需要，又使费用最少？

解　设购买食品 A_1，A_2，A_3 的数量分别为 x_1，x_2，x_3，花费的费用为 S，则该问题可以用下面的数学模型来描述

$$\min \quad S = 4x_1 + 2x_2 + 3x_3$$

$$\text{s. t.} \quad \begin{cases} 2x_1 + 4x_3 \geqslant 5 \\ 2x_1 + 3x_2 + x_3 \geqslant 4 \\ x_1, x_2, x_3 \geqslant 0 \end{cases}$$

(1) 首先把线性规划问题写成矩阵形式

$$S = 4x_1 + 2x_2 + 3x_3 = (4, 2, 3)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\text{s. t.} \begin{cases} \begin{pmatrix} -2 & 0 & -4 \\ -2 & -3 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leqslant \begin{pmatrix} -5 \\ -4 \end{pmatrix} \\ x_1, x_2, x_3 \geqslant 0 \end{cases}$$

(2) Matlab 的命令为

```
c = [4,2,3], A = -[2,0,4;2,3,1], b = [-5,-4]
v0 = [0,0,0]
X = (lp(c,A,b,v0))'
```

其计算结果为：

X = 0　　0.9167　　1.2500

所以，当购买 0.9617 数量的食品 A_2，1.25 数量的食品 A_3 时可以满足本问题的要求，此时花费的费用为 5.5734 元.

§4.2　非线性规划

4.2.1　无约束非线性规划及其在 Matlab 中的求解方法

1. 无约束非线性规划的基础知识

将式 (4.1) 重新写成

$$\min_{X} f(x), \quad X = (x_1, x_2, \cdots, x_n)^T \tag{4.4}$$

一般 f 为非线性函数,式(4.4)称为无约束非线性规划,实际上就是一个多元函数无条件极值问题. 应当注意的是,极值问题的解,即极值点,都是局部最优解,全局最优解只能从局部最优解的比较中得到,下面所讲的最优解均指局部最优解.

将 $f(x)$ 的一阶导数记做 $\nabla f(X) = (f_{x_1}, f_{x_2}, \cdots, f_{x_n})^T$,则 $\nabla f(X)$ 是 n 维向量;$f(x)$ 的二阶导数记做 $\nabla^2 f(X) = (f_{x_i x_j})$,$\nabla^2 f(X)$ 是 $n \times n$ 阶矩阵,称为海塞(Hessian)矩阵. 由多元函数极值问题最优解的条件知道,$X = X^*$ 是最优解的必要条件为

$$\nabla f(X^*) = 0 \tag{4.5}$$

$X = X^*$ 是最优解的(注意,这里是求极小值)的充分条件为

$$\nabla f(X^*) = 0, \text{且 } \nabla^2 f(X^*) \text{ 正定} \tag{4.6}$$

对于有实际意义的极值问题,通常只用必要条件,理论上只需求解方程组 $\nabla f(X) = 0$ 即可. 但对于非线性函数 $f(X)$,$\nabla f(X)$ 一般比较复杂,$\nabla f(X) = 0$ 的解析解很难求出,所以通常都是用数值迭代法来求其数值解.

常用的数值迭代法有:牛顿法、拟牛顿法等.

2. 非线性最小二乘拟合

假设有一组数据 (t_i, y_i),$i = 1, 2, \cdots, n$,要拟合一个已知函数

$$y = f(t, X)$$

其中 $X = (x_1, x_2, \cdots, x_m)^T$,$m \leqslant n$,X 为待定系数. 记

$$r_i(X) = y_i - f(t_i, y_i), \quad r(X) = (r_1(X), \quad r_2(X), \cdots, r_n(X))^T$$

拟合误差的定义为 $r_i(X)$ 的平方和,这样问题就表示为如下的优化模型

$$\min_{x} R(X) = r^T(X) r(X)$$

当 $f(X)$ 对 X(的某些分量)是非线性函数时,称为非线性最小二乘拟合. 实际上这是一个无约束非线性规划,可以用前面介绍的非约束线性规划的方法来求解,但由于该问题中目标函数是 $r_i(X)$ 的二次函数这一特殊性,可以构造一些相对简单的算法来求解.

记 $r(X)$ 的雅可比(Jacobi)矩阵为

$$J(X) = \left(\frac{\partial r_i}{\partial x_j}\right)_{n \times m}$$

则

$$\nabla R = 2J(X)^T r(X)$$

$$\nabla^2 R = 2J(X)^T J(X) + 2S, \quad S = \sum_{i=1}^{n} r_i(X) \nabla^2 r_i(X)$$

$$\nabla^2 r_i(X) = \left(\frac{\partial^2 r_i}{\partial x_k \partial x_l}\right)_{m\times m}, \quad k,l = 1,2,\cdots,m$$

如果用牛顿法计算，就要计算 $(\nabla^2 R)^{-1}$，其中 S 的计算量很大，主要是 $\nabla^2 r_i(X)$ 中的二阶导数 $\left(\frac{\partial^2 r_i}{\partial x_k \partial x_l}\right)$ 计算非常困难.

非线性最小二乘拟合就是探讨在计算时忽略 S 或对 S 取一近似值的问题. 非线性最小二乘拟合常用的方法有：高斯—牛顿 (Gauss-Newton) 法和 LM 方法.

3. 无约束非线性规划和最小二乘拟合在 Matlab 中的求解方法

(1) 无约束非线性规划在 Matlab 中的求解方法

对于无约束极小值 (非线性规划) 问题 $\min_x f(X)$，$X \in R^n$，在 Matlab 中的求解命令如下：

命令形式 1：X = fminu ('fun', X0)

功能：从点 X0 开始求函数 fun 的极小值点 X，其中 fun 是 M 函数文件名；X，$XO \in R^n$

命令形式 2：X = fminu ('fun', X0, opt)

功能：从点 X0 开始求函数 fun 的极小值点 X，其中 fun 是 M 函数文件名；X，$X0 \in R^n$，opt (1) = 1 有中间结果输出；opt (1) = -1 给出警告信息.

例 4. 求解 $\min\left(\frac{x^2+y^2}{2}\right)$.

解　1) 建立 M 函数文件

function y = fun (X)

Y = X (1) 2/2 + X (2) 2/2 % 用 X (1) 代替题目中的 X，用 X (2) 代替题目中的 Y

2) 建立命令函数文件

X0 = [1, 1]；

X = fminu ('fun', X0)；

其计算结果为

X = 1.0e - 009 * (- 0.3649, - 0.3649)

即所求的极小值点为：$x = -0.3469 \times 10^{-9}$，$y = -0.3469 \times 10^{-9}$.

本题的精确结果为：x = y = 0.

(2) 非线性最小二乘拟合在 Matlab 中的求解方法

命令形式 1：X = leastsq ('fun', X0)

功能：从点 X0 开始求函数 fun 的极小值点 X，其中 fun 是 M 函数文件名；X，$X0 \in R^n$.

命令形式 2 :X = leastlq ('fun' ,X0 ,opt)

功能 :从点 X0 开始求函数 fun 的极小值点 X,其中 fun 是 M 函数文件名 ;X, X0 ∈ R^n,opt 表示输入的参数信息 .

注意 fminu 与 leastsq 中的参数 opt 在下面将给出详细的用法解释 .

例 5. 用表 4. 2 中一组数据拟合 $c(t) = re^{-kt}$ 中的系数 r,k.

表 4. 2

t	0. 25	0. 5	1	1. 5	2	3	4	6	8
c	19. 21	18. 15	15. 36	14. 10	12. 89	9. 32	7. 45	5. 24	3. 01

解 1) 建立 ct. m 文件计算函数值

```
function    f = ct (X)
t = [0. 25    0. 5    1    1. 5    2    3    4    6    8] ;
c = [19. 21    18. 15    15. 36    14. 10    12. 89    9. 32    7. 45    5. 24
        3. 01] ;
f = c - X (1) * exp (X (2) * t) ;
```

2) 建立命令函数文件

```
X0 = [10 ,0. 5] ;
X = leastsq ('ct',X0)
```

其计算结果为

X = 20. 2413 - 0. 2420

即 r = 20. 2413,k = - 0. 2420.

(3) fminu 和 leastsq 中控制参数 options 的设置

在大多数优化程序中有一个 (向量) 控制参数 options,由 18 个元素组成,供使用时控制精度要求、输出形式、算法选择、迭代次数等 . options 的 18 个元素的功能及其缺省值如表 4. 3 所示 .

表 4. 3 控制参数 options 的功能表

序号	功 能	缺省值及含义	说明 (opt = options)
1	输出形式	0,无中间结果输出	opt (1) = 1,有中间结果输出 opt (1) = - 1,给出警告信息

续表

序号	功　能	缺省值及含义	说明 (opt = options)
2	解 X (k) 的精度	$\left\|\frac{x^{(k+1)}-x^{(k)}}{x^{(k)}}\right\| \leqslant 1e-4$	用 opt (2) 设置 $X^{(k)}$ 的精度
3	函数 f (k) 的精度	$\left\|\frac{f^{(k+1)}-f^{(k)}}{f^{(k)}}\right\| \leqslant 1e-4$	用 opt (3) 设置 $f^{(k)}$ 的精度
4	约束 g (k) 的精度	$\| g^{(k+1)} - g^{(k)} \| \leqslant 1e-7$	用 opt (4) 设置 $g^{(k)}$ 的精度
5	主要算法	0,视具体问题而定	用 opt (5) 选择 (视具体问题而定)
6	搜索方向方法	0,视具体问题而定	用 opt (6) 选择 (视具体问题而定)
7	步长一维搜索	0,视具体问题而定	用 opt (7) 选择 (视具体问题而定)
8	函数值输出		用 opt (8) 输出迭代结束时的函数值
9	梯度检验	0,不作检验	opt (9) = 1,在最初几步将给出的梯度与差分计算的梯度比较
10	函数计算次数		opt (10) 输出函数计算次数
11	梯度计算次数		opt (11) 输出梯度计算次数
12	约束计算次数		opt (12) 输出约束梯度计算次数
13	等式约束	0,等式约束的个数为 0	用 opt (13) 输入等式约束个数,必须将等式约束置前
14	最大迭代次数	100n,n 为变量 X 的维数	用 opt (14) 输入最大迭代次数
15	目标优化	0	用 opt (15) 输入必须达到目的的目标个数
16	差分步长最小值	1e - 8	用差分计算梯度时步长的下限
17	差分步长最大值	0. 1	用差分计算梯度时步长的上限
18	步　长		步长参数,第一次迭代置 1

(4) fminu 和 leastsq 的算法选择

fminu 为无约束优化的搜索方向提供了三种算法,由options (6) 控制:

options (6) = 0,计算时采用拟牛顿法的 BFGS 公式;

options (6) = 1,计算时采用拟牛顿法的 DFP 公式;

options (6) = 2,计算时采用最速下降法.

fminu 为无约束优化的步长一维搜索提供了两种算法,由 options (7) 控制:

options (7) = 0,混合的二次和三次多项式插值;

options (7) = 1,三次多项式插值.

Leastsq 为非线性最小二乘拟合的下降方向提供了两种算法,由 options (5) 控制:

options (5) = 0,计算时下降方向采用 LM 方法.

options (5) = 1,计算时下降方向采用高斯—牛顿 (GN) 法.

Leastsq 为非线性最小二乘拟合的步长一维搜索提供了两种算法,由 options (7) 控制,使用方法与 fminu 相同.

例 6. 求解 $\min\left(\frac{x^2}{10}+\frac{y^2}{1}\right)$,观察中间结果;将解和函数值的精度提高到 10^{-8},给出迭代次数及结果的函数值.

解 这里需要利用控制参数 options,即函数命令中的 opt,命令形式为

X = fminu ('fun',X0,opt)

(1) 建立 M 函数文件

```
function y = fun (x)
Y = X (1) 2 /10 + X (2) 2 /1, % 用 X (1) 代替原函数中的 X,用 X (2) 代替原函数中的 Y.
```

(2) 建立命令函数文件

```
% 要观察中间结果,只需令 opt (1) = 1 即可.
X0 = [1,1];% 初始值.
opt (1) = 1 ;% 输出中间结果.
X = fminu ('fun',X0,opt)
```

其运算结果为:

```
para =
      Columns 1 through 11
      1. 0000      0. 0001      0. 0001      0. 0000      0      0      0
  0      0      0      0
      Columns 12 through 18
            0            0    200. 0000    0    0. 0000    0. 1000    0
f - COUNT      FUNCTION      STEP-SIZE      GRAD/SD
```

```
   4          1.1          0.544554        -4.04
   9          0.0809191    0.504496        -1.28e-008
  15 1.69215e-016          4.95545         -2.25e-009
Optimization Terminated Successfully
Search direction less than 2*options(2)
Gradient in the search direction less than 2*options(3)
NUMBER OF FUNCTION EVALUATIONS = 15
X = 1.0e-007*
    0.0513        -0.1291
```

表明经 15 次迭代得到 $x_1 = 0.0513 \times 10^{-7}$，$x_2 = -0.1291 \times 10^{-7}$，中间结果给出了迭代次数、函数值、步长、搜索方向梯度值.

要将精度提高到 10^{-8}，只需令 opt(2) = 1e-8 和 opt(3) = 1e-8 即可. 在 Matlab 中命令如下：

```
X0 = [1,1];
opt(2) = 1e-8;opt(3) = 1e-8;
[X,opt] = fminu('fun',X0,opt);
Y = opt(8);
n = opt(10);
```

其输出结果为：

```
X = 1.0e-008*0.0533        0.8812
Y = 7.7679e-017
n = 22(迭代次数).
```

例 7. 设 X0 = (-1.9,2)，函数 $f(x_1,x_2) = 100(x_2 - x_1)^2 + (1 - x_1)^2$，试用不同算法（不同的搜索方向和步长搜索）求函数的极小值点和极小值，初始值选为 X0.

解　用三种搜索方向（BFGS，DFP 和最速下降法）及两种步长搜索（混合二、三次插值和三次插值）来计算，由 opt(6) = 0,1,2 和 opt(7) = 0,1 控制.

下面是用 DFP 和混合二、三次插值法计算时的程序和计算结果：

(1) 建立 M 函数文件

```
function f = fun(X)
f = 100*(X(2) - X(1)2 + (1 - X(1))2
```

(2) 建立命令函数文件

```
X0 = [-1.9,2];
opt(6) = 1;
```

```
[X, opt] = fminu ('fun',X0,opt);
f = opt (8);n = opt (10);
```

其计算结果为

X = 1.0000　　0.9999,　f = 2.1689e - 009,　n = 201

现将各种算法得到的结果列入表 4.4.

表 4.4

搜索方向	步长搜索	最优解	最优值	迭代次数
BFGS	混合二、三次插值	(1.00, 0.9999)	1.0967×10^{-9}	140
DFP	混合二、三次插值	(1.00, 0.9999)	2.1689×10^{-9}	201
BFGS	三次插值	(1.00, 1.0000)	8.0489×10^{-10}	202
DFP	三次插值	(0.5955,0.3459)	0.1712	220
最速下降	混合二、三次插值	(-0.6411,0.4265)	2.7064	1000

该函数的最优(极小)解为 $X^* = (1,1)$,极小值为 $f^* = 0$.

4.2.2 带约束的非线性规划及其在 Matlab 中的求解方法

1. 带约束的非线性规划的基础知识

将优化模型式(4.1)、式(4.2)更具体的表示为如下形式

$$\begin{aligned} &\min z = f(X), X \in R^n \\ &\text{s.t.} \quad h_i(x) = 0, i = 1,2,\cdots,m \\ &\quad\quad g_j(x) \leqslant 0, j = 1,2,\cdots,l \end{aligned} \tag{4.7}$$

其中 f, h_i, g_j 有非线性函数,则称式(4.7)为带约束的非线性规划.

由微积分知识知,如果只有等式约束 h_i,则可以用拉格朗日乘子法构造函数

$$L(X,\lambda_i) = f(X) + \sum_{i=1}^{m} \lambda_i h_i(X)$$

把约束优化问题化为无约束优化问题,然后利用无约束优化问题的必要条件求解. 因此一般情况下都考虑只有不等式约束 g_j 的情况,即优化模型为

$$\begin{aligned} &\min z = f(X), X \in R^n \\ &\text{s.t.} \quad g_j(X) \leqslant 0, j = 1,2,\cdots,l \end{aligned} \tag{4.8}$$

2. 可行方向与下降方向

(1) 起作用约束与不起作用约束

记带约束的非线性规划(4.8)的可行域为:$G: g_j(X) \leqslant 0, j = 1,2,\cdots,l$,设 X

为可行解，使

$$g_j(X)=0, j\in J_1;\quad g_j(X)<0, j\in J_2$$

则称 $g_j(j\in J_1)$ 为起作用的约束，$g_j(j\in J_2)$ 为不起作用的约束，且 $J_1\cap J_2=$.

(2) 可行方向

对于 $X\in G$ 和一方向 d，若存在实数 λ_0，使得 $X+\lambda d\in G(0<\lambda<\lambda_0)$，称 d 为 X 的可行方向．下面考察当 X 位于有效约束上可行方向的条件．

在有效约束上，即 $g_j(X)=0, j\in J_1$，将 $g_j(X+\lambda d)$ 在 X 处 Taylor 展开，得

$$g_j(X+\lambda d)=g_j(X)+\lambda\ \nabla g_j(X)^T d+o(\lambda^2)$$

因此，只要

$$\nabla g_j(X)^T d<0$$

当 λ 足够小时就有 $g_j(X+\lambda d)\leqslant 0$，于是 $X+\lambda d\in G$，d 为 X 的可行方向．

(3) 下降方向

对于 $X\in G$ 和一方向 d，若存在 λ_0，使

$$f(X+\lambda d)<f(X),\ (0<\lambda<\lambda_0)$$

则称 d 为 X 的下降方向．由 Taylor 展开式可知，若

$$\nabla f(X)^T d<0$$

则 d 必为下降方向．

由上述讨论可知，若在 $X\in G$ 处，d 既是可行方向又是下降方向，则 X 继续沿方向 d 移动时，目标函数 f 将减小，于是 X 不是最优解．反之，若 X 是最优解，就一定不存在可行、下降方向，因此可以知道：如果最优解在有效约束上，即在 $g_j(X)=0, j\in J_1$ 上，则一定不存在 d，使

$$\nabla g_j(X)^T d<0, j\in J_1$$

$$\nabla f(X)^T d<0$$

因此，也就可以得到最优解的必要条件．

(4) 最优解的必要条件

若 X 是带约束的非线性规划 (4.8) 的最优解，且 $\nabla g_j(X)\ (j\in J_1)$ 线性无关，则存在 $\lambda_1,\lambda_2,\cdots,\lambda_l\geqslant 0$ 使得

$$\nabla f(X)+\sum_{j=1}^{l}\lambda_j\nabla g_j(X)=0 \tag{4.9}$$

$$\lambda_j g_j(X)=0, j=1,2,\cdots,l \tag{4.10}$$

式 (4.9)、式 (4.10) 两式称为 K—T 条件．满足 K—T 条件的点称为 K—T 点．所以，最优解一定是 K—T 点．

对于式 (4.7)，若 X 是式 (4.7) 的最优解，且 $\nabla h_i(X)$，$\nabla g_i(X)\ (j\in J_1)$ 线性无关，则存在 $u_1,u_2,\cdots,u_m$ 和 $\lambda_1,\lambda_2,\cdots,\lambda_m$，使

$$\nabla f(X) + \sum_{i=1}^{m} u_i \nabla h_i(X) + \sum_{j=1}^{l} \lambda_j \nabla g_j(X) = 0$$

$$\lambda_j g_j(X) = 0, \quad j = 1,2,\cdots,l.$$

3. **凸二次规划及其在 Matlab 中的实现**

当目标函数是二次函数,约束为线性时,模型 (4.8) 可以简化为

$$\min f(X) = \frac{1}{2}X^T HX + cX \tag{4.11}$$

$$\text{s. t.} \quad AX \leqslant b$$

其中 X,A,b **与线性规划相同,如果** $H \in R^{n\times n}$ **为对称矩阵,则称式 (4.11) 为二次规划. 特别地,当** H **正定时,目标函数为凸函数,线性约束下可行域是凸集,则式 (4.11) 称为凸二次规划.**

凸二次规划具有下面两个比较好的性质:

性质 1 K—T **条件不仅是最优解的必要条件,而且是充分条件.**

性质 2 **局部最优解就是全局最优解.**

对于不等式约束的二次规划 (4.11),求解的一个重要方法是有效集方法. 有效集方法是一种迭代的方法. 二次规划在 Matlab **中的求解命令是** qp,qp **就是利用有效集方法来求解二次规划的. 命令** qp **有下面几种命令形式:**

命令形式 1:X = qp (H,c,A,b)

命令形式 2:X = qp (H,c,A,b,v1)

命令形式 3:X = qp (H,c,A,b,v1,v2)

命令形式 4:X = qp (H,c,A,b,v1,v2,X0)

命令形式 5:X = qp (H,c,A,b,v1,v2,X0,ne)

命令形式 6:X = qp (H,c,A,b,v1,v2,X0,ne,dis)

命令形式 7:[X,lag] = qp (H,c,A,b,...)

命令形式 8:[X,lag,how] = qp (H,c,A,b,...)

其中 H,c,A,b **是式 (4.11) 中的参数,其余的参数与解线性规划的命令** lp **中的含义相同.**

例 8. **求解**

$$\min f(x_1,x_2) = x_1^2 - 2x_1x_2 + 2x_2^2 - 4x_1 - 12x_2$$

$$\text{s. t.} \quad \begin{cases} x_1 + x_2 = 2 \\ x_1 - 2x_2 \geqslant -2 \\ 2x_1 + x_2 \leqslant 2 \\ x_1, x_2 \geqslant 2 \end{cases}$$

解 将上面的二次规划写成如式 (4.11) 的标准形式得

$$X=\begin{pmatrix}x_1\\x_2\end{pmatrix},\quad H=\begin{pmatrix}1 & -1\\-1 & 2\end{pmatrix},\quad c=[-4,-12]$$

$$A=\begin{pmatrix}1 & 1\\-1 & 2\\2 & 1\end{pmatrix},\quad b=\begin{pmatrix}2\\2\\3\end{pmatrix}$$

计算时,Matlab 命令如下:

```
H = [1,-1;-1,2];
c = [-4,-12];
A = [1,1;-1,2;2,1];
b = [2,2,3];
[X,lag] = qp(H,c,A,b,zeros(2,1),[],[],1);
f = X'*H*X/2 + c*X
```

其计算结果为

$X=(0.6667,1.3333)$,　$lag=(6.4444,1.7778,0,0,0)$,　$f=-17.5556$

因为 lag 中第 1、2 元素非零,所以第 1、2 两个约束是起作用约束.

4. 带约束非线性规划的解法及其在 Matlab 中的实现

非线性规划有很多种解法,如:可行方向法、罚函数法、梯度投影法等. 而 Matlab 优化工具箱中用的是逐步二次规划法,该方法被认为是解带约束的非线性规划更有效的方法.

逐步二次规划的基本原理在此不作详细叙述,很多书中都对这类知识有比较详细的介绍. Matlab 中用逐步二次规划法求解带约束的非线性规划的命令是 constr,该命令有下面几种命令形式:

命令形式 1:X = constr('fun',X0)

功能:从点 X0 开始求带约束条件的函数 fun 的极小值点 X,其中 fun 是 M 函数文件名,$X,X0\in R^n$.

命令形式 2:X = constr('fun',X0,opt)

功能:从点 X0 开始求带约束条件的函数 fun 的极小值点 X,其中 fun 是 M 函数文件名,$X,X0\in R^n$. opt(1) = 1,有中间结果输出;opt(1) = -1 给出警告信息.

命令形式 3:X = constr('fun',X0,opt,v1,v2,'grad')

命令形式 4:X = constr('fun',X0,opt,v1,v2,'grad',p1,p2,…)

命令形式 5:[X,opt] = constr('fun',X0,…)

constr 的基本用法和解无约束非线性规划的程序 X = fminu 相同,其中 v1,v2 表示 X 的下界和上界.

需要注意的是：M 函数文件 fun. Matlab 文件中要同时给出目标函数 f 和约束条件 g，形式为

[f,g] = fun (X)

grad. Matlab 文件中要（用分析的方法）同时给出目标函数 f 和约束条件 g 的梯度，形式为

[df,dg] = grad (X).

例 9. 求函数 $f(x_1,x_2)=100(x_2-x_1^2)^2+(1-x_1)^2$ 带约束条件

$$\begin{cases} x_1+x_2\leqslant 1.5 \\ x_1+x_2\geqslant 0 \end{cases}$$ 的极小值.

解　分不给出解析梯度和给出解析梯度两种情况.

(1) 建立 M 函数文件

```
function [f,g] = fun3 (X)
f = 100 * (X (2) - X (1) 2) 2 + (1 - X (1)) 2;
g (1) = X (1) + X (2) - 1. 5;
g (2) = - X (1) - X (2);
```

(2) 建立命令函数文件

```
[X,Y] = meshgrid (- 2 :0. 1 :2, - 1 :0. 1 :3);
z = 100 * (Y - X. 2). 2 + (1 - X). 2;
subplot (1,2,1); mesh (X,Y,Z);
title ('f (X,Y)'),axis ([- 2,2, - 1,3, - 1000,3000]);
subplot (1,2,2); contour (X,Y,Z,20); title ('等高线');
X0 = [- 1. 9,2]; opt (1) = - 1;
```

不给出解析梯度时：

```
X = constr ('fun3',X0,opt)
```

其运算结果为：

```
X = 0. 9077        0. 8222
```

给出解析梯度时：

```
% 创建梯度函数
function [df,dg] = grad (X)
df = [- 400 * X (1) * (X (2) - X (1) 2) - 2 * (1 - X (1)),
      200 * (X (2) - X (1) 2)];
dg = [2 * X (1), - 1; 2 * X (2), - 1];
X = constr ('fun3',X0,opt, [], [],'grad')
```

其运算结果为：

X = 0.9072　　0.8288

本题精确结果为：X1 = X2 = 1.

§4.3　Matlab 的优化工具箱

Matlab 的优化工具箱（Optimization toolbox）在 Matlab 中的名称是 toolbox h optim，其主要命令如下：

命令形式 1：X = fmin ('fun', X0)

功能：求解非线性规划的无约束极小，即求 minf (X)，$X \in R$ 的解．其中 X0 表示初始点，fun 是 M 函数文件名．

命令形式 2：X = fminu ('fun', X0) 和 X = fmins ('fun', X0)

功能：求解非线性规划的无约束极小，即求 minf (X)，$X \in R^n$ 的解．其中 X0 表示初始点，fun 是 M 函数文件名．

（注意：命令 1 中函数是一元函数，而命令 2 中函数是多元函数．）

命令形式 3：X = leastsq ('fun', X0) 和 X = curvefit ('fun', X0)

功能：求解非线性最小二乘，即求 $\min f^T(X) f(X)$．其中 fun 是 M 函数文件名．

命令形式 4：X = constr ('fun', X0)

功能：求解非线性规划的约束极小，即求 $\begin{cases} \min f(X), X \in R^n \\ \text{s.t.} \quad g(X) \leqslant 0 \end{cases}$

其中 X0 表示初始点，fun 是 M 函数文件名．

命令形式 5：X = lp (c, A, b)

功能：求解线性规划，即求 $\begin{cases} \min c^T X \\ \text{s.t. } AX \leqslant b. \end{cases}$

命令形式 6：X = qp (H, c, A, b)

功能：求解二次规划，即求 $\begin{cases} \min X^T HX/2 + c^T X \\ \text{s.t.} \quad AX \leqslant b \end{cases}$

命令形式 7：X = minimax ('fun', X0)

功能：求约束优化中函数极大值中的极小值，即求 $\begin{cases} \min(\max f(X)) \\ \text{s.t.} \quad g(X) \leqslant 0. \end{cases}$

命令形式 8：X = nnls (A, b)

功能：求非负线性最小二乘，即求 $\begin{cases} \min \| AX - b \| \\ \text{s.t.} \quad X \geqslant 0. \end{cases}$

命令形式 9：X = conls (A, b, c, d)

功能：求约束线性最小二乘，即求 $\begin{cases} \min \| AX - b \| \\ \text{s. t.} \quad cX \leqslant d. \end{cases}$

命令形式 10：X = fzero ('fun', X0)

功能：求一元函数 fun **在** X0 **附近的零点，即求** $f(X) = 0, X \in R$. **其中** fun **是被求零点的函数文件名，**X0 **是一个具体的值**.

注意：该命令求出的解为数值解，而且该命令不仅可以求零点，还可以求函数等于任何常数值的点.

命令形式 11：X = fsolve ('fun', X0)

功能：求多元非线性方程 fun **在** X0 **附近的近似解，即求** $f(X) = 0, X \in R^n$ **的数值解**. **其中** fun **是** M **函数文件名，**X0 **是一个具体的值**.

习 题 5

1. **在** Matlab **中求线性规划问题**

$$\max S = 17x_1 - 20x_2 + 18x_3$$

$$\text{s. t.} \quad \begin{cases} x_1 - x_2 + x_3 < 10 \\ x_1 + x_3 < 5 \\ x_1 < 5. \end{cases}$$

2. **求线性规划问题**

$$\min f = -x - 3y - 3z$$

$$\text{s. t.} \quad \begin{cases} 3x + y + 2z + v = 5 \\ x + z + 2v + w = 2 \\ x + 2z + u + 2v = 6 \\ x, y, z, u, v, w > 0. \end{cases}$$

3. **求解** $\min\left(\frac{x^2}{a} + \frac{y^2}{b}\right)$, $a = 1, b = 1$ **和** $a = 9, b = 1$ **并观察其中间结果**.

4. **在** Matlab **中求解**

$$\min e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

取初值为 $(-1, 1)$**，试分析比较不同算法的运算结果**.

5. **在** Matlab **中求解**

$$\min (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

6. **在** Matlab **中求解**

$$\min z = x_1 + x_2$$

$$\text{s. t.} \quad x_1 - x_2 \leqslant 1, x_1 \geqslant 0 \quad (x = (0, -1))$$

7. **求解非线性规划问题**

$$\min f = \exp(x_1 x_2 x_3 x_4 x_5)$$

$$\text{s. t.}\quad \begin{cases} x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10 \\ x_2 x_3 - 5x_4 x_5 = 0 \\ x_1 + x_2 + 1 = 0 \\ -2.3 \leqslant x_i \leqslant 2.3,\text{当 } i = 1,2 \text{ 时} \\ -2.3 \leqslant x_i \leqslant 3.2,\text{当 } i = 3,4,5 \text{ 时}. \end{cases}$$

8. 求解线性规划问题

$$\min z = -x_1 + 2x_2 + x_3$$

$$\text{s. t.}\quad \begin{cases} x_1 - x_2 - 2x_3 \geqslant -2 \\ x_1 + x_2 - x_3 \leqslant 4 \\ x_1 + 2x_2 \leqslant 4 \\ x_2 + x_3 \leqslant 4 \\ x_2 - x_3 = 1 \\ x_1, x_2, x_3 \geqslant 0. \end{cases}$$

9. 某饮料厂生产甲、乙两种产品，一件甲用 A 原料 1kg，B 原料 5kg；一件乙用 A 原料 2kg，B 原料 4kg. 现有 A 原料 20kg，B 原料 70kg. 甲、乙产品每件售价分别为 20 元和 30 元. 试问如何安排生产使收入最大？

10. 某食品公司生产的两种面包很出名，一种是叫“宋师”的白面包，另一种是叫“唐韵”的黑面包. 每个宋师面包的利润是 0.05 元，每个唐韵面包的利润是 0.08 元，两种面包的月生产成本固定是 4 000 元，不管生产多少面包. 该公司的面包厂分为两个部，分别是烤制部和调配部. 烤制部有 10 座大烤炉，每座烤炉的容量是每天出 140 台，每台可容纳 10 个宋师面包或 5 个更大的唐韵面包. 可以在一台上同时放两种面包，每个唐韵面包所占的空间是宋师面包的 2 倍. 调制部每天最多可以调制 8 000 个宋师面包和 5 000 个唐韵面包，有两个自动调配器分别用于两种面包的调配而不至于发生冲突. 试找出这两种面包的最佳产量比例，即确定两种面包的日产量，使得在公司面包厂现有条件下利润最高.

11. 一电路由三个电阻 R_1, R_2, R_3 并联，再与电阻 R_4 串联而成，记 R_k 上电流为 I_k，电压为 V_k，试在下列条件下确定 R_k，使电路总功率最小（$k = 1,2,3,4$）：

(1) $I_1 = 4, I_2 = 6, I_3 = 8, 2 \leqslant V_k \leqslant 10$；

(2) $V_1 = V_2 = V_3 = 6, V_4 = 4, 2 \leqslant I_k \leqslant 6$.

第二部分　建模实例

第五章　初等数学模型在 Matlab 中的求解方法

§5.1　卸煤台问题的优化

实验内容：

某煤炭公司有一个大型的煤炭装卸场，每列运煤的火车将装卸场上的煤往外运走．铁路调度部门每天发三列标准煤车到装卸货场．煤车到达装卸场的时间是早上 5:00 到晚上 8:00 之间的任意时刻．每列标准煤车由三节车厢组成，装满一列车煤要 3 个小时．装卸场的容量相当于 1.5 列车标准煤载量．当煤车到达装卸场时，若不能立即装煤，那么煤炭公司需向铁路部门交纳停滞费，每节车厢 5 000 美元/小时．此外铁路部门每周四发一列大载量煤车，该煤车有 5 节车厢，其载煤量是标准煤车的两倍，其到达时间是上午 11:00 到下午 1:00 之间．从矿井采的煤被运到，若场地是完全空的，那么一个工作班用 6 个小时可以卸满场地．使用一个工作班和相应的设备付 9 000 美元/小时，如果要提高向煤场卸煤的速度，可以调用第二个工作班帮助卸煤，其费用为 12 000 美元/小时．为了安全起见，当煤从矿井运到场地时，要停止向煤车装煤；而延误装车时间，煤炭公司要向铁路管理部门付停滞费．现在，煤炭公司的管理部门需要确定煤场的装卸操作的年预算，在这个预算内应包含下列几个主要因素：

(1) 应调用几次第二个工作班？

(2) 预期的月停滞费是多少？

(3) 煤炭装卸场每天能否允许第四列标准煤车运煤？

5.1.1　问题分析及模型的建立

1. 问题假设

(1) 工作班在一天 24 小时内，随时可以到场地工作，煤炭公司的管理部门

在煤车将驶入装卸场前,可以通知工作班到工作地点.

(2) 向煤车装煤不需要工作班工作,不收费.

(3) 每个工作班的工作效率相同.

(4) 为提高向煤场卸煤的速度,需要调用第二个工作班工作时,煤炭公司的管理部门可以立即调用.

(5) 煤车到达的时间是在早 5:00 点到晚 8:00 之间的任意时刻,每列标准煤车在这个时间段内到达的时刻服从随机均匀分布. 大载量煤车在上午 11:00 到下午 1:00 之间到达的时刻也服从随机均匀分布.

(6) 大载量煤车的装车时间是标准煤车的两倍,即需要 6 小时才能装满车.

(7) 各列煤车到达的时间是独立的,即煤车 A 的到达与煤车 B、煤车 C 的到达无关.

(8) 装卸场地的服务原则是先进先出,即先到的煤车装满煤开出装卸场后,下一列煤车才能开进装卸场装煤.

(9) 被调来的工作班,至少要工作半小时,以后,根据需要可以随时停止工作.

2. 目的及问题分析

建立该模型的目的是要计算最小年操作经费和最小每月的停滞费. 平均每年有 52.18 个星期,平均每个月有 4.35 个星期. 若求出每星期的最小费用,那么可以得出每月及每年的最小预算.

如果有煤车停滞,需要付停滞费,标准煤车为 $5\ 000 \times 3 = 15\ 000$ 美元/小时,大载量煤车为 $5\ 000 \times 5 = 25\ 000$ 美元/小时,而调用第二个工作班需要支付的费用为 12 000 美元/小时,所以应调用第二个工作班,以尽量避免煤车的停滞.

工作班调用方案的建立:

由上述分析知,建立模型的目的是希望求出每星期的最小费用. 煤车的到达是随机的,所以可以用计算机模拟每星期费用的平均值. 而该值是否为最小值,则受工作班调用方案策略的影响. 工作班调用的方案应包括任意时刻调用的第二个工作班,何时向煤场卸煤,何时向煤车装煤. 一般有下面 5 个方案可以使用:

(1) 无论任何情况,只使用一个工作班,而且在装煤车前,场地必须卸满煤.

(2) 无论任何情况,都使用第二个工作班,而且在装煤车前,场地必须卸满煤.

(3) 只使用一个工作班,在装煤车前,场地应有足够的煤装满一标准煤车,如果没有煤车等待装煤,场地应继续补充煤,直到场地完全装满或是下一列煤车

到达.

(4) 若使用第二个工作班,在装煤车前,场地应有足够的煤装满一标准煤车,如果没有煤车等待装煤,场地应继续补充煤,直到场地完全装满或是下一列煤车到达.

(5) 如果场地的煤不足一标准列车的载量,则调用两个工作班,否则只调用一个工作班. 在装煤前,场地应有足够的煤装满一标准煤车,如果没有煤车等待装煤,场地应继续补充煤,直到场地完全装满或是下一列煤车到达.

下面从这 5 个方案中选出一个最好的,通过计算机模拟来实现.

5.1.2 建模所需的知识点在 Matlab 中的实现方法

1. 产生随机矩阵

(1) rand (n)

产生一个 n × n 阶的随机矩阵,矩阵元素的值在 (0,1) 之间.

(2) rand (m,n)

产生一个 m × n 阶的随机矩阵,矩阵元素的值在 (0,1) 之间.

(3) randn (n) 或 randn (m,n)

产生一个符合正态分布的 n × n 阶或 m × n 阶的随机矩阵.

2. 向量排序

1) sort (X)

如果 X 是向量,则按照 X 元素值的大小从小到大进行排序 如果 X 是矩阵,则对 X 的每一列进行排序.

3. 求向量平均值

mean (X)

如果 X 为一向量,则求 X 所有元素的平均值 如果 X 是一矩阵,则求 X 每一列的平均值.

5.1.3 模型在 Matlab 中的实现

1. 以方案 (5) 为例进行计算机模拟

在程序中定义一天的工作时间早上 5:00 至 0:00,晚上 8:00 至 15:00,上午 11:00至 6:00,下午 1:00 至 8:00. 同时定义下面的一些变量:

tt [1,2,3,4] ——在一天里的煤车到达时刻的数组;

sfee ——总费用;

sfeede ——总的停滞费;

feen ——第 j 辆煤车开始装煤时,上一辆煤车走后工作班的费用;

de ——第 j 辆煤车走后,工作班将煤场装满一车煤的时刻;

tte ——第 j 辆煤车走后,工作班将煤场完全装满的时刻;

tts ——第 j 辆煤车开始装煤的时刻;

qus ——第 j 辆煤车开始装煤时,煤场中的煤量.

对于该方案,编写程序的流程如下:

```
e = 0;tte = 0;tts = - 3 ;que = 1.5;sfee = 0;sfeede = 0
for k = 1 to n (模拟 n 个星期)
    {for I = 1 to 7} (一个星期的 7 天)
    {If I = = 4}
    {生成 3 个 [0,15] 区间的随机数及一个 [6,8] 区间的随机数放入数组
      tt [1,2,3,4],并从小到大排序,且确定其中 [6,8] 区间的随机数的位
      置 pp.}
            for j = 1 to 4
    {计算上一辆煤车走后,第 j 辆煤车开始装煤时工作班的费用 feen,
      sfee + = feen}
      if j = = pp
    {按第二辆标准煤车处理大载量煤车的 tts,qus,feede,并累加 sfeede 和
      sfee}
      else
     {处理标准煤车的 tts,qus,feede,并累加 sfeede 和 sfee}
      修改 de,tte}     }
        Else
         {
    生成 3 个 [0,15] 区间上的随机数放入数组 tt [1,2,3],并从小到大排
序;
            For  j = 1 to 3
              {计算上一辆煤车走后,第 j 辆煤车开始装煤时工作班的费
用 feen,sfee + = feen;
                再计算第 j 辆煤车的 tts,qus,feede 累加 sfeede 和 sfee;
                然后修改 de ,tte}
                }
            一天后修改参数:tts,de,tte ;
            }
            算出一星期的平均费用:sfee/n,sfeede/n;
```

}

下面给出该模型在 Matlab 中模拟时的程序代码：

```
clear all , clc
de = 0;tts = -3;tte = 0;sfee = 0;feen = 0;sfeede = 0;feede = 0;
for k =1 :1000
    for i = 1 :7
  tt = rand(1,3) * 15;tt = sort(tt);
  if i = = 4
      tt1 =6 + rand(1) * 2;tt2 = [tt, tt1];tt2 = sort(tt2);
      for m = 1 :4
          if tt2(m) == tt1(1)
              pp = m;
          end
    end;
    for j = 1 :4
        if tt2(j) < = de
            feen = (de - tts - 3) * 21;
        elseif tt2(j) < tte
            feen = (de - tts - 3) * 21 + (tt2(j) - de) * 9;
        else
            feen = (de - tts - 3) * 21 + (tte - de) * 9;
        end
    sfee = sfee + feen;
        if j ~ = pp
            if de < = tte(j)
                tts = tt2(j);
            else
                tts = de;
            end;
            if tt2(j) < = de
                qus =1;
            elseif tt2(j) < = tte
                qus =1 = (tt2(j) - de) * 0.5;
            else
```

```
            qus = 1.5;
        end
         feede = (tts  tt2(j)) * 15;
        sfeede = sfeede + feede;
        de = (2 - qus) * 2 + tts + 3;  tte = de + 2;
    else
        if de <= tt2(j)
           tts = tt2(j);
        else
            tts = de;
         end;
        if tt2(j) <= de
           qus = 1;
        elseif tt2(j) <= tte
                qus = 1 + (tt2(j) - de) * 0.5;
         else
                 qus = 1.5;
         end
       feede = (tts - tt2(j)) * 25;
       sfeede = sfeede + feede;
       sfee = sfee + feede;
       de = (2 - qus) * 2 + tts + 3;  tte = de + 2;
       if tt2(j) <= de
          feen = (de - tts - 3) * 21;
       elseif  tt2(j) < tte
           feen = (det - tts - 3) * 21 + (tt2(j) - de) * 9;
       else
           feen = (de - tts - 3) * 21 + (tte - de) * 9;
       end;
    sfee = sfee + feen;
    tt2(j) = tts + 3;tts = de;qus = 1;
    feede = (tts - tt2(j)) * 25;
    sfeede = sfeede + feede;
    sfee = sfee + feede;
```

```
        de = (2 - que) * 2 + tts + 3; tte = de + 2;
        end
    end
  else
    for j = 1:3
        if tt(j) <= de
          feen = (de - tts - 3) * 21;
        elseif tt(j) <= tte
          feen = (de - tts - 3) * 21 + (tt(j) - de) * 9;
        else
          feen = (de - tts - 3) * 21 + (tte - de) * 9;
        end
        sfee = sfee + feen;
        if de <= tt(j)
           tts = tt(j);
        else
           tts = de;
        end
        if tt(j) <= de
          qus = 1;
        elseif tt(j) <= tte
           qus = 1 + (tt(j) - de) * 0.5;
         else
            qus = 1.5;
         end;
        feede = (tts - tt(j)) * 15;
        sfeede = sfeede + feede;
        de = (2 - qus) * 2 + tts + 2;
        tte = de + 2;
        end;
    end;
        tts = tts - 24; de = de - 24; tte = tte - 24;
    end;
end;
```

```
feen = (de - tts - 3) * 21 + (tte - de) * 9; sfee = sfee + feen;
sfee = sfee/1000
sfeede = sfeede/1000
```

其运算结果为：

```
sfee = 1.7075e + 003
sfeede = 791,1552
```

同理对其余几种方案进行模拟，最后得到的结果如表 5.1 所示.

表 5.1

	方案 (1)	方案 (2)	方案 (3)	方案 (4)	方案 (5)
总费用	3 795.34	1 910.5	3 298.86	1 758.91	1 707.5
停滞费	2 967.34	944.499	2 470.86	792.911	791.1552

结果分析：

从上述计算的结果中可以看出：采用方案 (5) 时，总费用为 1 707.5 美元，停滞费为 791.1552 美元，为最好的方案.

§5.2　工厂选址

实验内容：

工厂 A 到铁路的垂直距离为 3km，垂足 B 到火车站 C 为 5km，汽车运费 20 元/t·km，铁路运费 15 元/t·km，为使运费最省，在 M 点建一转运站，且 M 点在铁路线 BC 之间，试问转运站 M 应建在何处？

5.2.1　问题分析及模型的建立

该问题实际上是一个计算极小值的问题，求解该模型的目的就是要在铁路线 BC 之间找到一个中转站，使得货物从工厂运到火车站时所花费的路费最少.

设转运站 M 距 B 点的距离为 xkm，则 M 离火车站的距离就是 5 - xkm，根据所给的条件知道，此时所需的总路费为

$$p(x) = 20\sqrt{x^2+9} + 15(5-x),\ x \geqslant 0$$

因此，只需求出 p(x) 的极小值点，该问题就解决了.

模型的建立：

该模型的建立比较简单，由以上分析知，用初等数学方法即可以建立该模型，所花费用的函数为

$$p(x) = 20\sqrt{x^2+9} + 15(5-x), x \geqslant 0.$$

在 Matlab **中求解该问题时，可以用两种方法求解：第一种方法：求** $p(x)$ **的一阶导数** $p'(x)$**，然后求方程** $p'(x) = 0$ **的解** x^***；**

绘出 $p(x)$ **的函数图像，通过图像观察其在** $[0,5]$ **上极小值点的位置，然后利用求函数极小值的命令** fmin**，求出** $p(x)$ **的极小点值即可**.

采用第二种方法更有效，因为该命令采用的是数值方法，它不仅将极小值点与极小值同时给出，而且在函数不可导时仍可以求解，在实际计算时很实用.

5.2.2 建模所需的知识点在 Matlab 中的实现方法

1. **一元函数极值的求法在** Matlab **中的实现方法**

(1) fmin(fun,x0,x1)

在区间 [x0,x1] **内求函数** fun **的极小值点**.

(2) fminu(fun ,x0)

从 x0 **开始寻找函数** fun **的极小值点**.

2. **一元函数的导数在** Matlab **中的实现方法**

diff(fun,var,n)

对函数 fun **按变量** var **求** n **阶导数**.

3. **求方程的根**

(1) solve(s,v)

对方程 s **按指定的变量** v **进行求解**.

(2) fzero(fun,x0)

求函数 fun **在** x0 **附近的零解**.

5.2.3 模型在 Matlab 中的实现

计算该模型的 Matlab **代码如下：**

```
clear all;
syms x;
ff = 20*(9+x2)(1/2)+15*(5-x);
%用一阶导数等于0来求极小值点.
dff = diff(ff,x);
x* = solve(dff,'x').
```

其运算结果为

$x^* = 3.40168$

% 直接求函数的极小值.

$x_1^* =$ fmin (ff,0,5);

其运算结果为

$x^* = 3.40168.$

§5.3　商品市场占有率问题

实验内容：

有 R 和 S 两家公司经营同类产品,这两家公司相互竞争. 每年 R 公司保持有 1/4 的顾客,而 3/4 转移向 S 公司;每年 S 公司保持有 2/3 的顾客,而 1/3 转移向 R 公司. 当产品开始制造时 R 公司占有 3/5 的市场份额,而 S 公司占有 2/5的市场份额.

试问两年以后,两家公司所占有的市场份额变化怎样?5 年以后会怎样?10 年以后呢?是否有一组初始市场份额分配数据使以后每年的市场分配成为稳定不变?

5.3.1　问题分析及模型的建立

根据两家公司每年顾客转移的数据资料,形成下面的转移矩阵

$$A = \begin{pmatrix} \frac{1}{4} & \frac{1}{3} \\ \frac{3}{4} & \frac{2}{3} \end{pmatrix}$$

根据产品制造之初市场的初始分配数据可得如下向量

$$X_0 = \begin{pmatrix} \frac{3}{5} \\ \frac{2}{5} \end{pmatrix}$$

所以一年后,市场分配为

$$X_1 = AX_0 = \begin{pmatrix} \frac{1}{4} & \frac{1}{3} \\ \frac{3}{4} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} \frac{3}{5} \\ \frac{2}{5} \end{pmatrix}$$

两年以后,市场分配为

$$X_2 = AX_1 = A^2 X_0$$

设 n 年后市场分配的份额为 X_n,则有

$$X_n = AX_{n-1} = A^n A_0, \quad (n = 1, 2, \cdots)$$

设数据 a,b 为 R 公司和 S 公司的初始市场份额,则有

$$a + b = 1$$

为了使以后每年的市场份额分配不变,根据顾客转移的规律,有

$$\begin{pmatrix} \frac{1}{4} & \frac{1}{3} \\ \frac{3}{4} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

即

$$\begin{pmatrix} -\frac{3}{4} & \frac{1}{3} \\ \frac{3}{4} & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

这是一个齐次方程组问题,如果方程组有解,则应该在非零解的集合中选取正数解作为市场份额稳定的初始份额.

模型的建立:

根据上述分析可知,该问题的数学模型为求两个线性方程

$$X_n = AX_{n-1} = A^n X_0$$

和

$$\begin{pmatrix} -\frac{3}{4} & \frac{1}{3} \\ \frac{3}{4} & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

5.3.2 建模所需的知识点在 Matlab 中的实现方法

对线性方程组的求解:

(1) 可以用 [x1,x2] = solve (s1,s2,v1,v2) 来求方程组 AX = 0 的解.

(2) 可以用命令 rref (A),把 A 化为上三角阵,然后再求解.

5.3.3 模型在 Matlab 中的实现

该模型在 Matlab 中的求解代码为:

```
A = [1/4,1/3;3/4,2/3];
x0 = [3/5;2/5];
```

$x_2 = A2 * x_0$

$x_5 = A5 * x_0$；

$x_{10} = A10 * x_0$；

可得到数据结果

$x_2 = 0.3097 \quad 0.6903$

$x_5 = 0.3077 \quad 0.6923$

$x_{10} = 0.3077 \quad 0.6923$

为了求 a 和 b 作为 R 公司和 S 公司稳定的初始市场份额，需要求解齐次方程组，下面利用 rref 命令来求解：

```
format rat;
rref(A - eye(2))
```

其运算结果为：

```
ans = 1      - 4/9
      0      0
```

由此得化简后的方程为

$$a - \frac{4}{9}b = 0$$

结合约束条件

$$a + b = 1$$

可以得到

$$a = \frac{4}{13} \approx 31\%$$

$$b = \frac{9}{13} \approx 69\%$$

这就是使市场稳定的两家公司的初始份额.

习　题　6

1. 赛艇是一种靠选手划桨前进的小船，分单人艇、双人艇、四人艇、八人艇四种. 八人艇还分重量级（选手平均体重 86kg）和轻量级（选手平均体重 73kg）. 各种艇虽然大小不同，但形状相似，表 5.2 是各种赛艇在 1964～1970 年四次 2 000m 比赛的最好成绩（包括 1964 年和 1968 年的两次奥运会和两次世界锦标赛），发现它们之间有相当一致的差别. 经研究，发现比赛成绩与选手数量之间存在某种联系. 试建立一个模型来解释这种关系. 通过分析还发现，八人艇重量级组的成绩比轻量级组的成绩约好 5%，试用自己建立的模型解释这一现象.

表 5.2 各种艇的比赛成绩和规格

艇 种	2000m 成绩 t/min					艇长 L/m	艇宽 B/m	L/B	艇重/选手数 kg/n
	1	2	3	4	平均				
单 人	7.16	7.25	7.28	7.17	7.21	7.93	0.293	27.0	16.3
双 人	6.87	6.92	6.95	6.77	6.88	9.76	0.356	27.4	13.6
四 人	6.33	6.42	6.48	6.13	6.32	11.75	0.574	21.0	18.1
八 人	5.87	5.92	5.82	5.73	5.84	18.28	0.610	30.0	14.7

2. 在相距 100m 的两个塔(高度相等的点)上悬挂一根电缆,允许电缆在中间下垂 10m,试计算在这两个塔之间所用电缆的长度.

3. 某旅游景点从山脚到山顶有一缆车索道,全长约 1 471m,高差为 380m,采用循环单线式修建. 缆绳悬挂在下站到上站的行程中的 8 个铁塔上,这 8 个铁塔依山势走向而距离不等,从下站到第一铁塔的水平距离为 d_0,高差为 h_0;从第一铁塔到第二铁塔的水平距离为 d_1,高差为 h_1,……,从第 8 个铁塔到上站的水平距离为 d_8,高差为 h_8,具体数据如表 5.3 所示.

表 5.3

d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
220	200	140	120	100	120	140	200	220
h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
50	45	40	38	34	38	40	45	50

每一段缆绳下垂的最低点不低于两端铁塔最低塔顶悬绳处 1m,试估算整个索道工程所用的缆绳总长度.

4. 一栋楼房后面有一个很大的花园,在花园中紧靠着楼房有一个温室,温室伸入花园宽 2m,高 3m,温室正上方是楼房的窗台. 清洁工打扫窗台周围,他得用梯子越过温室,一头放在花园中,一头靠在楼房的墙上. 因为温室是不能承受梯子的压力的,所以梯子太短是不行的. 现清洁工只有一架 7m 长的梯子,试问梯子能达到要求吗?能满足要求的梯子的最小长度为多少?

第六章　微积分方法模型在 Matlab 中的求解方法

§6.1　水箱的水流问题

在许多供水单位由于没有测量流入或流出水箱流量的设备，而只能测量水箱中的水位．试通过测得的某时刻水箱中水位的数据，估计在任意时刻 t 流出水箱的流量 f(t)．某社区有一供水水箱，在居民用水过程中，当水箱中的水位下降到最低水位 l 时，水泵就自动向水箱输水直到最高水位 H，此期间不能测量水泵的供水量，因此，当水泵正在输水时不容易建立水箱中水位和用水量之间的关系．水泵每天输水一次或两次，每次约 2 小时．

已知该水箱是一个高为 40ft（英尺），直径为 57ft 的圆柱体，表 6.1 是该居民区一天中水箱水位的数据，当水位降至 27.00ft 时水泵开始工作，水位升到 35.50ft 时，停止输水（1ft = 0.304 8m）．

表 6.1　社区某天水箱水位

时间/s	水位/0.01ft	时间/s	水位/0.01ft
0	9.68	12.95	10.02
0.92	9.45	13.88	9.94
1.84	9.31	14.98	9.65
2.95	9.13	15.90	9.41
3.87	8.98	16.83	9.18
4.98	8.81	17.93	8.92

续表

时间/s	水位/0.01ft	时间/s	水位/0.01ft
5.90	8.69	19.04	8.66
7.00	8.52	19.96	8.43
7.93	8.39	20.84	8.22
8.97	8.22	22.02	水泵供水
9.98	水泵供水	22.96	水泵供水
10.93	水泵供水	23.88	10.59
10.95	10.82	24.99	10.35
12.03	10.50	25.91	10.18

6.1.1 问题分析及模型的建立

由于水箱是正圆柱体,横截面积是常数,所以在水泵不工作时段,流量很容易根据水位和相对时间的变化计算出来,但如何估计水泵供水时段的流量比较困难. 水泵供水时段的流量只能靠供水时段前后的流量经插值或拟合得到. 因为水泵不工作时段的流量要作为用于插值或拟合的原始数据,因此水泵不工作时段的流量越精确越好. 这些流量可以用两种方法来计算:(1) 对表 6.1 中的数据用数值微分计算出各时段的流量,用它们拟合其他时刻或连续时间的流量;(2) 先用表 6.1 中数据拟合水位—时间函数,然后求导数就可以得到连续时间的流量.

有了每个时刻的流量,就可以计算水箱一天的总流量. 水泵不工作时段的用水量可以由测量记录直接得到,由表 6.1 的数据可以直接计算出;水泵工作时水箱的流量通过拟合出来的流量函数计算出. 这样就可以计算出水箱一天的总流量.

对模型做以下假设:

1. 将流量看做是时间的连续函数,为了便于计算,不妨将流量定义为单位时间流出的水的高度,即水位对时间变化率的绝对值,水箱的截面积为

$$S = \frac{\pi}{4} \times (57 \times 0.3024)^2 = 233.2292 \quad (m^2)$$

在计算总流量时把上面得到的结果乘以 S 即可.

2. 流量只取决于水位差,与水位本身无关,即从水龙头流出的水的流速正比于水面高度的平方根. 题目给出水箱的最低和最高水位分别是 8.164 8m 和 10.735 2m (设出口的水位为 0),计算得 sqrt (10.735 2/8.164 8) ≈1.146 7,约为 1,所以可以忽略水位对流速的影响.

流量估计方法:

用一种比较简单的方法计算水箱流量与时间的关系,将表 6.1 中的数据分为 5 段,按时间 t 排列:第 1 段:0 ~ 8.97;第 2 段:9.38 ~ 10.93;第 3 段:10.95 ~ 20.84;第 4 段:22.02 ~ 22.96;第 5 段:23.88 ~ 25.91. 然后对每一段的数据做如下处理:设某段数据为 $\{(x_0,y_0),(x_1,y_1),\cdots,(x_n,y_n)\}$,相邻数据中点的平均流速用公式

流速 = (左端点的水位 - 右端点的水位) /区间长度

即

$$v\left(\frac{x_{i+1}+x_i}{2}\right)=\frac{y_i-y_{i+1}}{x_{i+1}-x_i}$$

计算:每一段数据首尾点的流速用下面公式计算

$$v(x_0)=(3y_0-4y_1+y_2)/(x_2-x_0)$$

$$v(x_n)=(-3y_n+4y_{n-1}-y_{n-2})/(x_n-x_{n-2})$$

根据上面的公式,可以计算出时间与流速之间的数据如表 6.2 所示.

表 6.2　时间与流速之间的数据表

时间/h	流速/(cm/h)	时间/h	流速/(cm/h)	时间/h	流速/(cm/h)
0	29.89	8.45	16.35	17.38	23.64
0.46	21.74	8.97	19.29	18.49	23.42
1.38	18.48	9.98	水泵供水	19.50	25.00
2.395	16.22	10.93	水泵供水	20.40	23.86
3.41	16.30	10.95	33.50	20.84	22.17
4.425	15.32	11.49	29.63	22.02	水泵供水
5.44	13.04	12.49	31.52	22.96	水泵供水
6.45	15.45	13.42	29.03	23.88	27.09
7.465	13.98	14.43	26.36	24.43	21.62
		15.44	26.09	25.45	18.48
		16.37	24.73	25.91	13.30

用两种计算方法来建立模型:

(1) 插值法

由表 6.2,对水泵不工作时段采取插值方法,可以得到任意时刻的流速,从而知道任意时刻的流量,这里分别采用拉格朗日(Lagrange)插值法、分段线性插值法和三次样条插值法进行插值. 对于水泵工作时段 2 应用前后期的流速进行插值,由于第 5 段水泵不工作时的数据太少,将它与水泵工作时段 4 合并一起进行插值. 这样就总共需要对 4 段数据进行插值(第 1,3 段未供水时段,第 2 段供水时段,第 4、5 时段的混合时段).

(2) 曲线拟合法

拟合水位—时间函数. 根据表 6.1 的测量记录知,一天有两次供水时段和三次未供水时段,分别对 1、3 未供水时段的测量数据直接作多项式拟合,可以得到水位函数(根据多项式拟合的特点,拟合多项式的次数不宜太高,一般在 3 ~ 6 次之间),然后由水位—时间函数确定流量—时间函数,这样也可以求出一天总用水量的估计.

6.1.2 求解模型所需的知识点在 Matlab 中的实现方法

1. 计算向量或矩阵的长度

length (X)

如果 X 为向量,则计算向量 X 的长度 如果 X 为矩阵,则返回 X 中行和列的较大值.

2. 计算矩阵的尺寸

[m,n] = size (X)

返回矩阵 X 的行值 m 和列值 n.

3. 一维插值

Y = interp1 (X0, Y0, X, 'method')

根据初始值 X0 和 Y0,按照 method 的方法进行插值. 其中常用的插值方法有:nearest, linear, cubic 和 spline.

4. 计算多项式的值

polyval (p, X0)

计算多项式 p 在 X0 处的值.

5. 多项式求导

polyder (p)

返回多项式 p 的一阶导数值.

6. 多项式拟合

a = polyfit (X0, Y0, n)

其中 n 表示拟合多项式的最高阶数,X0,Y0 为待拟合的数据. 输出参数为

拟合多项式

$$y = a_1x^n + \cdots + a_nx + a_{n+1}$$

的系数 $[a_1,\cdots,a_n,a_{n+1}]$.

6.1.3　模型求解在 Matlab 中的实现

1. 插值法求解在 Matlab 中的实现

以第一段未供水时数据为例分别用三种插值方法计算出流量函数和用水量. 由于 Matlab 中没有直接提供拉格朗日插值法的命令函数,这里先给出用 Matlab 语言实现的拉格朗日插值法的函数 lglrcz. m.

```
function   Y = lglrcz (X0,Y0, X)
n = length (X0)
m = length (X) ;
for i =1 :m
    z = X (i)
    s =0 ;
    for k =1 :n
    p =1. 0
    for j =1 :n
        if j ~ = k
        p = p * (z - X0 (j) ) / (X0 (k) - X0 (j) ) ;
        end
    end
     s = p * Y0 (k) + s
    end
     Y (i) = s ;
     end
% 由表 6. 4 可得到
t = [0, 0. 46, 1. 38, 2. 395, 3. 41, 4. 425, 5. 44, 6. 45, 7. 465, 8. 45,
8. 97] ;
v = [29. 89, 21. 74, 18. 48, 16. 22, 16. 30, 15. 32, 13. 04, 15. 45, 13. 98,
16. 35, 19. 27] ;
t0 =0 :0. 1 :8. 097
lr = lglrcz (t, v, t0) ;                    % 拉格朗日插值法
lrjf = 0. 1 * trapz (lr)
```

```
fdcz = interp1(t, v, t0);          % 分段线性插值法
fdczjf = 0.1 * trapz(fdcz)
scz = interp1(t, v, t0, 'spline');     % 三次样条插值法
sanczjf = 0.1 * trapz(scz)
plot(t ,v ,'*', t0, lr, 'r', t0, fdcz, 'g', to ,scz, 'b')
gtext('lglr');gtext('fdxx');gtext('scyt');
```

其运算结果为：

lr = 145.623 1　　fdczjf = 147.1430　　sanczjf = 145.6870.

一般情况下，三次样条插值方法具有比较好的性质，大多数情况下都采用该方法．其他三段的处理方法与第 1 段未供水时段的处理方法类似，只给出结果，如表 6.3 所示．

表 6.3　　各时段和一天的总用水量（用水高度）

	第 1 段（未供水段）	第 2 段（供水段）	第 3 段（供水段）	第 4 段（混合段）	全天
拉格朗日插值法	145.6231	258.8664	54.0689	92.1337	550.6921
分段线性插值法	147.1430	258.9697	49.6051	76.4688	532.1866
三次样条插值法	145.6870	258.6547	53.3334	81.7699	539.4450

2. 拟合法在 Matlab 中的实现

（1）拟合水位—时间函数

设 t，h 分别为已输入的时刻和水位测量记录（由表 6.1 得到，水泵供水的 4 个时刻不输入），第一未供水时段各时刻的水位可以由下面程序实现，如图 6.1 所示．

```
t = [0, 0.92, 1.84, 2.95, 3.87, 4.98,5.90, 7.00, 7.93, 8.97, 10.95,
12.03, 12.95, 13.88, 14.98, 15.90, 16.83, 17.93, 19.04, 19.96, 20.84,
23.88, 24.99, 25.66];
h = [9.68, 9.48, 9.31, 9.13, 8.98, 8.81, 8.69, 8.52, 8.39, 8.22,
10.82, 10.50, 10.21, 9.94, 9.65, 9.41, 9.18, 8.92, 8.66, 8.43, 8.22,
10.59,10.35, 10.18];
c1 = polyfit(t(1:10),h91:10),3);
tp1 = 0:0.1:8.9
x1 = polyval(c1,tp1);
```

```
plot (tp1 ,x1) ;
```

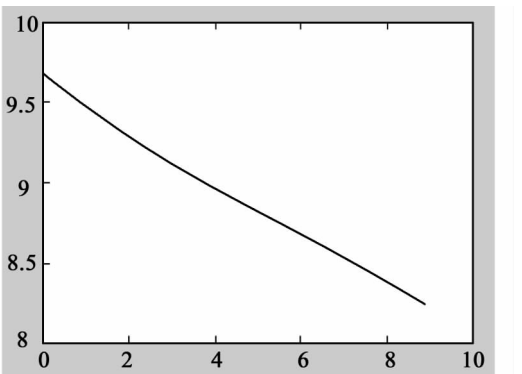


图 6.1　第一未供水时段的时间—水位拟合函数图

变量 X1 中存放了以 0.1 为步长计算出的各个时刻的水位高度. 第二未供水时段时间—水位图可以由下面程序实现,如图 6.2 所示.

```
c2 = polyfit (t (11: 21) ,h (11: 21) ,3) ;
tp2 = 10.9 : 0.1: 20.9
X2 = - polyval (c2 ,tp2) ;
plot (tp2 ,X2) .
```

(2) 确定流量 - 时间函数

```
c1 = polyfit (t (1:10) ,h (1:10) ,3) ;
c2 = polyfit (t (11:21) ,h (11:21) ,3) ;
a1 = polyder (c1) ;
a2 = polyder (c2) ;
tp1 = 0 :0.01:8.97
tp2 = 10.95:0.01:20.84
X13 =- polyval (a1 ,tp1) ;
X113 = - polyval (a1 , [0:0.01:8.97]) ;
wgsysll = 100 * trapz (tp1 ,X113) ;
X14 =- polyval (a1 , [7.93 ,8.97]) ;
X23 =- polyval (a2 ,tp2) ;
X114 =- polyval (a2 , [10.95:0.01:20.84]) ;
wgsys12 = 100 * trapz (tp2 ,X114) ;
```

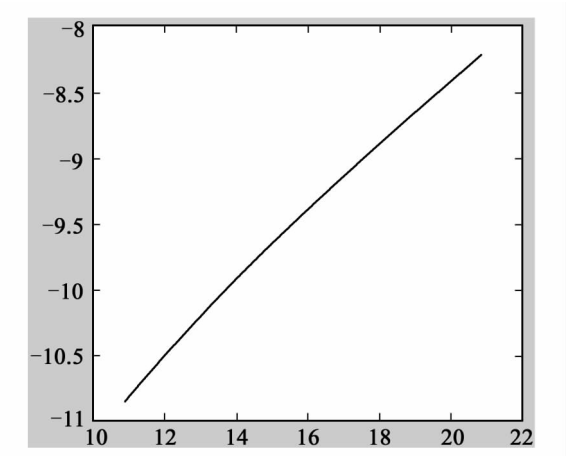


图 6.2　第二未供水时段的时间—水位拟合函数图

```
X24 =- polyval (a2, [10.95,12.03]) ;
X25 =- polyval (a2, [19.96,20.84]) ;
subplot (1,2,1) ;
plot (tp1,X13 * 100) ;
subplot (1,2,2) ;
plot (tp2,X23 * 100) .
```

其运算结果如图 6.3 所示.

第二供水时段的流量则用前后时期的流量进行拟合得到,为使流量函数在 t = 9和 t = 11 连续,只取四个点,用三次多项式拟合得到第二供水时段的时间—流量图像如图 6.4 所示. 实现的程序代码如下:

```
dygsdsj = [7.93, 8.97, 10.95, 12.03] ;
dygsdls = [X14,X24] ;
nhjg = polyfit (dygsdsj,dygsdls,3) ;
nhsj = 7.93 :0.1 :12.03
nhlsjg = polyval (nhjg,nhsj) ;
gssj1 = 8.97 :0.01 :10.95
gs1 = polyval (nhjg, [8.97 :0.01 :10.95]) ;
gsysll = 100 * trapz (gssj1,gs1) ;
plot (nhsj, 100 * nhlsjg)
```

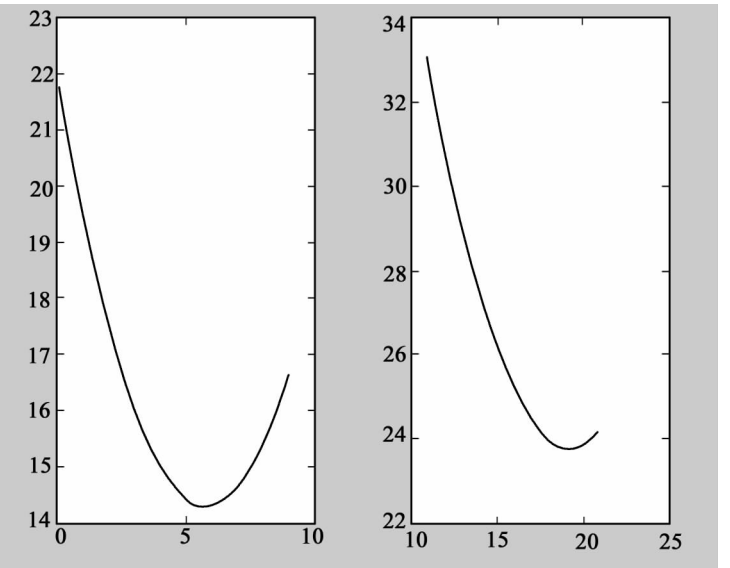


图 6.3　时间—流量函数图

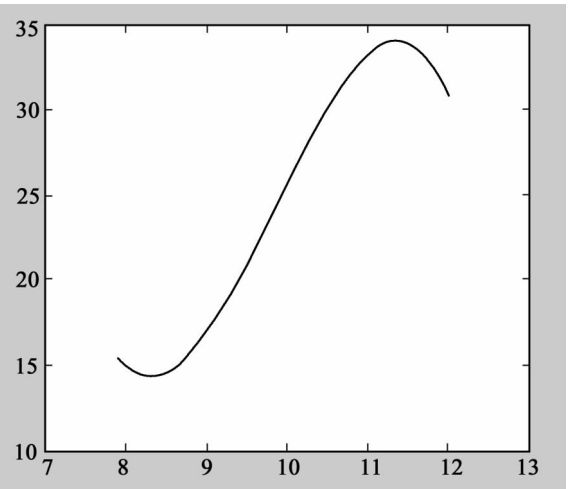


图 6.4　第二供水时段时间—流量图

在第四供水时段之前取 t = 19.96，20.84 两点的流量，用第五未供水时段的

三个记录做差分得到两个流量数据 21.62，18.48，然后用这四个数据做三次多项式拟合得到第四供水时段与第五未供水时段的时间—流量函数，如图 6.5 所示．程序如下：

```
t3 = [19.96, 20.84, t(22), t(23)];
ls3 = [X25 * 100, 21.62, 18.48];
nhhhddxsxs = polyfit(t3, ls3, 3);
tp3 = 19.96: 0.01: 25.91;
XX3 = polyval(nhhddxsxs, tp3);
gssj2 = 20.84: 0.01: 24;
gs2 = polyval(nhhddxsxs, [20.84: 0.01: 24]);
gsysl2 = trapz(gssj2, gs2);
plot(tp3, XX3);
```

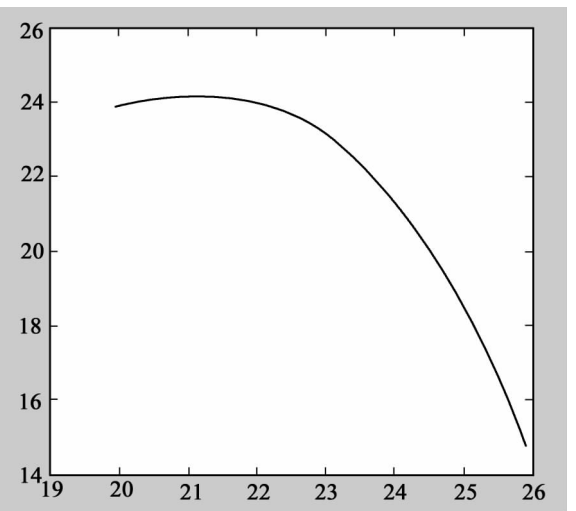


图 6.5 第四、第五时段时间—流量函数图

(3) 一天总用水量的估计

分别对供水的两个时段和不供水的两个时段积分（流量对时间）并求和得到一天的总用水量约为 526.8935 ft^2（总用水高度，单位为 cm）．各时段用水量如表 6.4 所示．

表 6.4　各时段用水量及一天总用水量　(单位:cm)

时　段	第 1 时段(未供水段)	第 2 时段(供水段)	第 3 时段(未供水段)	第 4 时段(混合段)	全天用水 ft^2
用水高度	145.67	260.66	46.60	73.9635	526.8935

§6.2　卫星轨道的长度和射击命中概率

问题 1:卫星轨道的长度

人造地球卫星的轨道可以视为平面上的椭圆,中国第一颗人造卫星近地点距离地球表面 439km,远地点距离地球表面 2 384km,地球半径为 6 371km,试求该卫星的轨道长度.

问题 2:射击命中概率

炮弹射击的目标为一正椭圆形区域,当瞄准目标的中心发射时,在众多因素的影响下,弹着点与目标中心有随机偏差. 可以合理地假设弹着点围绕中心呈二维正态分布,且偏差在 x 方向和 y 方向相互独立. 若椭圆形区域在 x 方向半轴长 120m,y 方向半轴长 80m,设弹着点偏差的均方差在 x 方向和 y 方向均为 100m,试求炮弹落在椭圆形区域内的概率.

问题 3:人口增长率

20 世纪美国人口统计数据如表 6.5 所示,计算表中这些年份的人口增长率.

表 6.5　20 世纪美国人口增长率

年　份	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990
人口 $(\times 10^6)$	76.0	92.0	106.5	123.2	131.7	150.7	179.3	204.0	226.5	251.4

又已知某地区 20 世纪 70 年代的人口增长率如表 6.6 所示,且 1970 年人口为 210 万,试估计该地区 1980 年的人口.

表 6.6 某地区 20 世纪 70 年代人口增长率数据

年　份	1970	1972	1974	1976	1978	1980
年增长率/(%)	0.87	0.85	0.89	0.91	0.95	1.10

6.2.1 问题分析及模型的建立

1. 卫星轨道的长度

卫星轨道的参数方程为 $x = a\cos t, y = b\sin t, 0 \leqslant t \leqslant 2\pi$，a，b 分别为轨道椭圆的长、短半轴．根据计算参数方程弧长的公式，椭圆长度可以用下面公式来计算

$$L = 4\int_0^{\frac{\pi}{2}} (a^2\sin^2 t + b^2\sin^2 t)^{\frac{1}{2}} dt \tag{6.1}$$

式 (6.1) 称为椭圆积分，式 (6.1) 不能用解析方法计算．

2. 射击命中概率

根据射击情况可知，弹着点与目标中心的偏差服从二维正态分布，且在 x 方向和 y 方向相互独立．设目标中心为 $x = 0, y = 0$，则弹着点 (x，y) 的概率密度函数是

$$p(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} \tag{6.2}$$

其中 $\sigma_x = \sigma_y = 100$ (m)．式 (6.2) 也无法用解析方法求解，对式 (6.2) 作如下变换．

令 $x = au, y = bv$，且以 100m 为 1 单位，即 $\sigma_x = \sigma_y = 1, a = 1.2, b = 0.8$，于是

$$P = ab\iint_{\Omega} \tilde{p}(u,v)\, dudv \tag{6.3}$$

$$\tilde{p}(u,v) = \frac{1}{2\pi} e^{-\frac{1}{2}(a^2u^2+b^2v^2)}, \overline{\Omega}: u^2 + v^2 \leqslant 1$$

可以用式 (6.3) 计算炮弹命中的概率．

3. 人口增长率

(1) 若记时刻 t 的人口为 x (t)，则人口 (相对) 增长率 $r(t) = \dfrac{\frac{dx}{dt}}{x(t)}$，表示每年人口增长的比例．对于题目给出的人口数据，记 1900 年为 k = 0；1910，1920，…，1990 年依次为 k = 1，2，…，9．相应地，人口记为 x_k，年增长率为 r_k，用数值微分

计算公式可得

$$r_k = \frac{x_k - x_{k-1}}{20x_k}, \quad k = 1,2\cdots,8$$

$$r_0 = \frac{-3x_0 + 4x_1 - x_2}{20x_0}, \quad r_9 = \frac{x_7 - 4x_8 + 3x_9}{20x_9}.$$

(2) 人口增长满足微分方程

$$\frac{dx}{dt} = r(t)x(t) \tag{6.4}$$

式 (6.4) 在初始条件 $x(0) = x_0$ 下的解为

$$x(t) = x_0 e^{\int_0^t r(u)du}$$

因为人口增长率 $r(u)$ 以离散数据给出,所以计算该积分时要用数值积分进行计算.

模型的建立:

由上述分析可知,计算卫星轨道长度的数学模型为

$$L = 4\int_0^{\frac{\pi}{2}} (a^2\sin^2 t + b^2\sin^2 t)^{\frac{1}{2}} dt \tag{6.5}$$

炮弹射击命中概率的数学模型为

$$P = ab\iint_{\bar{\Omega}} \bar{p}(u,v)\,dudv \tag{6.6}$$

$$\bar{p}(u,v) = \frac{1}{2\pi}e^{-\frac{1}{2}(a^2u^2 + b^2v^2)}, \bar{\Omega}: \ u^2 + v^2 \leqslant 1 \tag{6.7}$$

20 世纪美国人口的增长率的计算数学模型为

$$r_k = \frac{x_k - x_{k-1}}{20x_k}, \quad k = 1,2,\cdots,8 \tag{6.8}$$

$$r_o = \frac{-3x_0 + 4x_1 - x_2}{20x_0}, \quad r_9 = \frac{x_7 - 4x_8 + 3x_9}{20x_9}$$

对某地区 1980 年的人口进行估计的数学模型为

$$x(t) = x_0 e^{\int_0^t r(u)du} \tag{6.9}$$

6.2.2　建模所需的知识点在 Matlab 中的实现方法

1. 向量求和

sum (X)

输入数组 X,输出为 X 的和. 可以用于按矩形求积公式计算定积分的值.

2. 梯形法计算定积分

trapz (X, Y)

输入数组 X,Y 为同长度的数组,输出 Y 对 X 的积分(按梯形公式计算).

3. 2 阶辛普生法计算定积分

quad('fun',a,b)

用辛普生公式(2 阶)计算以 fun.M 文件命令的函数在区间[a,b]上的积分,自动选择步长,相对误差为 10^{-3}.

4. 8 阶辛普生法计算定积分

quad8('fun',a,b,tol)

用辛普生(8 阶)公式计算以 fun.M 文件命令的函数在[a,b]上的积分,相对误差为 tol.

5. 产生均匀分布的随机数

rand(1,n)

产生 n 个(0,1)随机数.

6.2.3 模型求解在 Matlab 中的实现

1. 计算卫星轨道的长度

在 Matlab 中用梯形公式和辛普森公式计算的程序如下:

```
function Y = X5(t)
a = 8755;b = 6810;
Y = sqrt(a 2 * sin(t).2 + b 2 * cos(t).2);
t = 0:pi/10:pi/2;
Y1 = X5(t);
L1 = 4 * trapz(t,Y1)
L2 = 4 * quad('X5',0,pi/2,le - 6)
```

其运算结果为

$$L_1 = 4.9089965267852e + 004$$

$$L_2 = 4.9089965318304 + 004$$

即轨道的长度近似为 4.909×10^4km.

2. 射击命中概率

在 Matlab 中实现的程序代码如下:

```
a =1.2;b = 0.8;m = 0;z = 0;n = 100000;
    for i =1: n
        X = rand(1,2);
        Y = 0;
        if X(1)2 + X(2)2 <= 1
```

```
        Y = exp( - 0.5 (a 2 * X (1) 2 + b 2 * X (2) 2) ) ;
        Z = Z + Y ;
        m = m + 1 ;
end
end
p = 4 * a * b * a /2 /pi /n , m
```

这里 n = 100 000 次,运算结果命中概率为 P = 0.3572, m = 78 552.

§6.3　森林救火模型

森林失火了,消防站接到报警后派多少消防员前去救火呢?派的队员越多,森林的损失越小,但是救援的开支会越大,所以需要综合考虑森林损失费和救援费与消防队员人数之间的关系,以总费用最小来决定派出队员的数目.

6.3.1　问题分析及模型的建立

问题分析:

损失费通常正比于森林烧毁的面积,而烧毁面积与失火、灭火(指火被扑灭)的时间有关,灭火时间又取决于消防队员数目,消防队员越多灭火越快. 救援费除与消防队员人数有关外,也与灭火时间长短有关. 记失火时刻的时间 $t = 0$,开始救火的时刻为 $t = t_1$,灭火时刻为 $t = t_2$. 设时刻 t 时森林的烧毁面积为 B(t),则造成损失的森林烧毁面积为 $B(t_2)$. 建模时要对函数 B(t) 作出合理的简单假设.

研究$\frac{dB}{dt}$比研究 B(t) 更为直接和方便. $\frac{dB}{dt}$是单位时间烧毁的面积,表示火势蔓延的程度,在消防队员到达之前,即 $0 \leqslant t \leqslant t_1$,火势越来越大,即$\frac{dB}{dt}$随着 t 的增加而增加;开始救火后,即 $t_1 \leqslant t \leqslant t_2$,如果消防队员救火能力足够强,火势会越来越小,即$\frac{dB}{dt}$应减小,并且当 $t = t_2$ 时,$\frac{dB}{dt} = 0$.

救援可以分为两部分;一部分是灭火器材的消耗与消防队员的薪金等,与消防队员人数及灭火所用的时间均有关,另一部分是运送消防队员和器材等一次性支出,只与消防队员人数有关.

模型的建立:

需要对烧毁森林的损失费、救援费及火势蔓延程度$\frac{dB}{dt}$的形式作出假设.

(1) 损失费与森林烧毁面积 $B(t_2)$ 成正比,比例系数为 c_1,c_1 即烧毁单位面积的损失费.

(2) 从失火到开始救火这段时间 $(0 \leqslant t \leqslant t_1)$ 内,火势蔓延程度 $\frac{dB}{dt}$ 与时间 t 成正比,比例系数 β 称为火势蔓延速度.

(3) 派出消防队员 x 名,开始救火以后 $(t \geqslant t_1)$ 火势蔓延速度降为 $\beta - \lambda x$,其中 λ 可以视为每个队员的平均灭火速度,显然应有 $\beta < \lambda x$.

(4) 每个消防队员单位时间的费用为 c_2,于是每个消防队员的救火费用是 $c_2(t_2 - t_1)$;每个消防队员的一次性支出是 c_3.

对第 (2) 条假设可以作如下解释:火势以失火点为中心,以均匀速度向四周呈圆形蔓延,所以蔓延的半径 r 与时间 t 成正比,又因为烧毁面积 B 与 r^2 成正比,故 B 与 t^2 成正比,从而 $\frac{dB}{dt}$ 与 t 成正比.

根据假设条件 (3),火势蔓延程度 $\frac{dB}{dt}$ 在 $0 \leqslant t \leqslant t_1$ 线性地增加,在 $t_1 \leqslant t \leqslant t_2$ 线性地减小.

记 $t = t_1$ 时,$\frac{dB}{dt} = b$. 烧毁面积 $B(t_2) = \int_0^2 \frac{dB}{dt} dt$ 恰是图中三角形的面积,显然有 $B(t_2) = \frac{1}{2} b t_2$,而 t_2 满足

$$t_2 - t_1 = \frac{b}{\lambda x - \beta} \tag{6.10}$$

于是

$$B(t_2) = \frac{1}{2} b t_1 + \frac{b^2}{2(\lambda x - \beta)} \tag{6.11}$$

根据假设条件 (1) 和 (4),森林损失费为 $c_1 B(t_2)$,救援费为 $c_2 x(t_2 - t_1) + c_3 x$. 将式 (6.10)、式 (6.11) 代入,得到救火总费用为

$$C(x) = \frac{1}{2} c_1 b t_1 + \frac{c_1 b^2}{2(\lambda x - \beta)} + \frac{c_2 b x}{\lambda x - \beta} + c_3 x \tag{6.12}$$

问题归结为求 x 使 C(x) 达到最小.

令 $\frac{dC}{dx} = 0$,可以得到应派出的队员人数为

$$x = \sqrt{\frac{c_1 c_3 \lambda b^2 + 2 c_2 \beta b}{2 c_3 \lambda^2}} + \frac{\beta}{\lambda} \tag{6.13}$$

结果解释:首先,应派出队员数目由两部分组成,其中一部分 $\frac{\beta}{\lambda}$ 是为了把火

扑灭所必需的最低限度．因为 β 是火势蔓延速度，而 λ 是每个队员的平均灭火速度，所以这个结果是明显的．通过计算可以看出，只有当 $x > \frac{\beta}{\lambda}$ 时，斜率为 $\lambda x - \beta$ 的直线才会与 t 轴有交点 t_2.

其次，派出队员的另一部分，即在最低限度之上的人数，与问题的各个参数有关．当队员灭火速度 λ 和救援费用系数 c_3 增大时，队员人数减少；当火势蔓延速度 β、开始救火时的火势 b 及损失费用系数 c_1 增加时，队员人数增加．这些结果与常识是一致的．式 (6.13) 还表明，当救援费用 c_2 变大时，队员人数也增加．

在应用这个模型时，c_1，c_2，c_3 是已知常数，β、λ 是由森林类型、消防队员素质等因素决定的．建立这个模型的关键是对 $\frac{dB}{dt}$ 的假设．

6.3.2　建模所需的知识点在 Matlab 中的实现方法

1. 定义符号变量 sym 或 syms

命令形式 1：x = sym ('x')

功能：sym 用来定义符号变量或符号方程，一次只能定义一个符号变量．

命令形式 2：syms　x　y　z

功能：用来定义多个符号变量，注意所定义的变量 (x、y 等) 要用空格隔开．

2. 对符号函数求导 diff ( )

命令形式 1：diff (f)

功能：求函数 f 的一阶导数，其中 f 为符号函数．

命令形式 2：diff (f,n)

功能：求函数 f 的 n 阶导数，其中 f 为符号函数．

3. 解方程 (或方程组) solve ( )

命令形式 1：solve (s,v)

功能：对方程 s 按指定的变量 v 求解．

命令形式 2)：[x1,x2,x3,…,xn] = solve (s1,s2,…,sn,v1,v2,…,vn)

功能：对 n 个方程的指定变量 v1,v2,…,vn 求解，并将求解的结果赋给 x1,x2,x3,…,xn.

4. 合并同类项 collect ( )

命令形式：r = collect (S,v)

功能：合并同类项，S 是符号表达式，v 是变量或表达式，r 是合并同类项后的结果．

6.3.3 模型求解在 Matlab 中的实现

求救火费用最小值在 Matlab 中的实现程序如下：

```
syms b x c1 c2 c3 t1 t2 ;
syms r % 用 r 表示模型公式中的 λ.
syms v % 用 v 表示建模公式中的 β.
t2 = t1 + b/(r*x - v);
Bt2 = 1/2*b*t2

ShenglingCost = c1*Bt2;% 森林损失的费用.
JiuyuanCost = c2*x*(t2 - t1) + c3* x ;% 救援的费用.

% 总的费用
Cost = ShenglingCost + JiuyuanCost;

% 通过 C'(x) = 0,求出森林灭火的最小费用.
dd = diff(Cost, x);
minCost = solve(dd,x)
```

其运算结果为：

```
minCost =
[1/4/c3*(4*c3*v+2*(2*c3*c1*b 2*r+4*c3*c2*b*v)
(1/2))/r]
[1/4/c3*(4*c3*v - 2*(2*c3*c1*b 2*r+4*c3*c2*b*v)
(1/2))/r]
ans =
[1/r*v+1/2/c3/r*(2*c3*c1*b 2*r+4*c3*c2*b*v) (1/2)]
[1/r*v - 1/2/c3/r*(2*c3*c1*b 2*r+4*c3*c2*b*v) (1/2)]
```

因为结果必定为正数,所以第二个根舍弃. 可以看到,求解的结果与前面理论分析时得到的结果完全一致.

习 题 7

1. 测得活塞中气体压力 P 和体积 V 的一组数据如表 6.7 所示.

表 6.7

P	60	80	100	120	140	160	180
V	80.0	69.2	60.0	52.0	45.0	38.6	32.5

试求 (1) V = 60，50 处，V 改变 1 时 P 的变化量；(2) V 从 70 减至 40 时气体所做的功.

2. 核物理手册上给出氢核 (质子) 和重氢核的初始能量与它们在空气中射程的关系如表 6.8 所示.

表 6.8

E/MeV	1.0	2.0	3.0	4.0	5.0
P/cm	2.30	7.20	14.10	23.10	33.90
D/cm	1.72	4.61	8.78	14.40	20.80

其中 E 为初始能量 (10^6 电子伏特)，P 和 D 分别为氢核和重氢核的射程.

(1) 试估计 E = 1.4，2.5，5.7 (MeV) 时，P 和 D 的值是多少？

(2) 估算 P = 8.21 时，E 等于多少 (此时 E 的精确值为 2.1698)？D = 10.42 时，E 的值是多少 (此时 E 的精确值为 3.3140)？P = 12.45 时，D 的值是多少 (此时 D 的精确值为 7.7740)？

3. 在化工生产中常常需要知道丙烷在各种温度 T 和压力 P 下的导热系数 K. 如表 6.9 所示是通过实验得到的一组数据.

表 6.9

T/	P (10^3 kN/m^2)	K	T/	P (10^3 kN/m^2)	K
68	9.7981	0.0848	106	9.7918	0.0696
68	13.324	0.0897	106	14.277	0.0753
87	9.0078	0.0762	140	9.6563	0.0611
87	13.355	0.0807	140	12.463	0.0651

试求 T = 99　和 P = 10.3 (10^3kN/m^2) 时的导热系数 K.

4. 增加生产、发展经济所依靠的主要因素有增加投资、增加劳动力以及技

术革新等,在研究国民经济产值与这些因素的数量关系时,由于技术水平不像资金、劳动力那样容易定量化,作为初步的模型,可以认为技术水平不变,只讨论产值和资金、劳动力之间的关系. 在科学技术发展不快时,如资本主义发展的前期,这种模型是有意义的.

用 Q, K, L 分别表示产值、资金、劳动力,要寻求的数量关系 Q(K, L),经过简化假设与分析,在经济学中,推导出一个著名的 Cobb-Douglas 生产函数

$$Q(K,L) = aK^{\beta}L^{\gamma}, 0 < \beta, \gamma < 1$$

式中的 a, β, γ 主要由经济统计数据确定. 现有美国某个州 1900 ~ 1926 年对上述三个经济指标的统计数据,如表 6.10 所示,试用数据拟合的方法,求出 $Q(K,L) = aK^{\beta}L^{\gamma}$ 中的参数 a, β, γ.

表 6.10

t	Q	K	L	t	Q	K	L
1900	1.05	1.04	1.05	1914	2.01	3.24	1.65
1901	1.18	1.06	1.08	1915	2.00	3.24	1.62
1902	1.29	1.16	1.18	1916	2.09	3.61	1.86
1903	1.30	1.22	1.22	1917	1.96	4.10	1.93
1904	1.30	1.27	1.17	1918	2.20	4.36	1.96
1905	1.42	1.37	1.30	1919	2.12	4.77	1.95
1906	1.50	1.44	1.39	1920	2.16	4.75	1.90
1907	1.52	1.53	1.47	1921	2.08	4.54	1.58
1908	1.46	1.57	1.31	1922	2.24	4.54	1.67
1909	1.60	2.05	1.43	1923	2.56	4.58	1.82
1910	1.69	2.51	1.58	1924	2.34	4.58	1.60
1911	1.81	2.63	1.59	1925	2.45	4.58	1.61
1912	1.93	2.74	1.66	1926	2.58	4.54	1.64
1913	1.95	2.82	1.68				

5. 在某海域测得一些点 (x, y) 处的水深为 z (单位:ft),数据如表 6.11 所示,水深数据是在低潮时测得的. 船的吃水深度为 5ft,假设海底是平滑的,试用插值法估计在矩形区域 (75,200) × (-50,150) 里的哪些地方船应避免进入.

表 6.11　　水道水深的测量数据　　(单位:ft)

X	129.0	140.0	103.5	88.0	185.5	195.0	105.5
Y	7.5	141.5	23.0	147.0	22.5	137.5	85.5
Z	4	8	6	8	6	8	8
X	157.5	107.5	77.0	81.0	162.0	162.0	117.5
Y	-6.5	-81.0	3.0	56.5	-66.5	84.0	-33.5
Z	9	9	8	8	9	4	9

6. 在以盛产石油著称的波斯湾地区,浩瀚的沙漠覆盖着大地,水资源十分缺乏,不得不采用淡化海水的办法为国民提供用水. 成本大约是每立方米 0.1 英镑. 有些专家提出从相距 9600km 外的南极用拖船运送冰山到波斯湾,以取代淡化海水的办法.

在运送冰山的过程中,拖船的租金、运量、燃料消耗及冰山运送过程中融化速率等方面的数据如下:

(1) 三种拖船的日租金和最大运量如表 6.12 所示.

表 6.12

船 型	小	中	大
日租金/(英镑)	4.0	6.2	8.0
最大运量/m^3	5×10^6	5×10^6	5×10^6

(2) 燃料消耗(英镑/km),主要依赖于船速和所运冰山的体积,船型的影响可以忽略,如表 6.13 所示.

表 6.13

冰山体积/m^3 / 船　速 /km/h	10^5	10^6	10^7
1	8.4	10.5	12.6
3	10.8	13.5	16.2
5	13.2	16.5	19.8

(3) 冰山运输过程中的融化速率(m/d),指在冰山与海水接触处每天融化

的深度．融化速率除与船速有关外，还与运输过程中冰山到达处与南极的距离有关，这是由于冰山要从南极运往赤道附近的缘故．如表 6.14 所示．

表 6.14

与南极距离/ km 船 速 /km/h	0	1 000	>4 000
1	0	0.1	0.3
3	0	0.15	0.45
5	0	0.2	0.6

试选择拖船的船型与船速，使冰山到达目的地后，可以得到的每立方米水所花的费用最低，并与海水淡化的费用相比较．拖船在拖运冰山的过程中，有以下假设：

(1) 拖船航行过程中船速不变，航行不考虑天气等任何因素的影响，总航行距离 9 600km；

(2) 冰山形状为球形，球面各点的融化速率相同；

(3) 冰山到达目的地后，$1m^3$ 的冰可以融化成 $0.85m^3$ 的水．

第七章　微分方程模型在 Matlab 中的实现方法

§7.1　动物种群的相互竞争与相互依存的模型

在生物的种群关系中，一种生物以另一种生物为食的现象，称为捕食．一般说来，由于捕食关系，当捕食动物数量增长时，被捕食动物数量即逐渐下降，捕食动物由于食物来源短缺，数量也随之下降，而被捕食动物数量却随之上升．这样周而复始，捕食动物与被捕食动物的数量随时间变化形成周期性的震荡．

田鼠及其天敌的田间种群消长动态规律也是如此，如图 7.1 所示，为实验调查数据的连线图．图 7.1 表明无论是田鼠还是其天敌的数量都呈周期性的变化，田鼠—天敌作用系统随时间序列推移，田鼠密度逐渐增加，其天敌密度也随之增加，但时间上落后一步；由于天敌密度增加，必然降低田鼠密度，而田鼠密度的降低，则其天敌密度亦将减少，如此往复循环，形成一定的周期．试用数学模型来概括这一现象，并总结出其数量变化的近似公式．

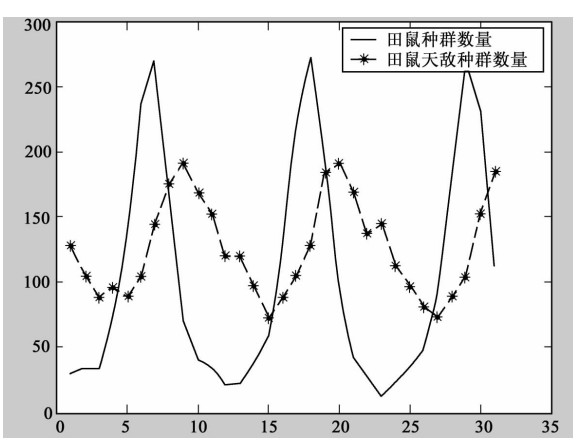


图 7.1　田鼠及其天敌数量变化图

7.1.1 问题分析及模型的建立

1. 问题分析

设 $x(t)$ 和 $y(t)$ 分别表示 t 时刻田鼠与其天敌的数量，如果单独生活，田鼠的增长速度正比于当时的数量，即

$$\frac{dx}{dt} = \lambda x \tag{7.1}$$

而田鼠的天敌由于没有被捕食对象，其数量减少的数率正比于当时的数量，即

$$\frac{dy}{dt} = -\mu x \tag{7.2}$$

现在田鼠与其天敌生活在一起，田鼠一部分遭到其天敌的消灭，于是以一定的速率 α 减少，减少的数量正比于天敌的数量，因此有

$$\frac{dx}{dt} = (\lambda - \alpha)x \tag{7.3}$$

类似地，田鼠的天敌有了食物，数量减少的速率 μ 将减少 β，减少的量也正比于田鼠的数量，因此有

$$\frac{dy}{dt} = -(\mu - \beta)x \tag{7.4}$$

上述公式中，最后两个方程联合起来 称为 Volterra-Lotoka 方程，其中 α，β，λ，μ 均为正数，初始条件为

$$x(0) = x_0, y(0) = y_0$$

2. 模型的建立

现在通过实验调查所得到的数据，建立合理的数学模型．下面是每隔两个月田间调查一次，得到的田鼠及其天敌种群数量的记录，数量的单位经过处理，如表 7.1 和表 7.2 所示．

表 7.1 田鼠种群数量记录

29.7	33.1	32.5	69.1	134.2	236.0	269.6	162.3	69.6	39.8	34.0
20.7	21.7	37.6	57.6	124.6	215.8	272.7	195.7	95.0	41.9	25.7
10.9	22.6	33.6	48.1	92.5	183.3	268.5	230.6	111.1		

表 7.2　　田鼠天敌种群数量记录

1.6	1.3	1.1	1.2	1.1	1.3	1.8	2.2	2.4	2.1	1.9	1.5	1.5	1.2	0.9	1.1	1.3
1.6	2.3	2.4	2.1	1.7	1.8	1.4	1.2	1.0	0.9	1.1	1.3	1.9	2.3			

Volterra-Lotoka 方程的解析解和 x,y 的显式解很难求出,因此式 (7.3) 与式 (7.4) 的参数方程不宜直接用 Matlab 函数来拟和解. 可以用下面的方法来求其参数的近似解.

变换 Volterra-Lotoka 方程可以写成

$$\begin{cases} d\ln x = (\lambda - ay)dt \\ d\ln y = (-\mu + \beta x)dt \end{cases}$$

在区间 $[t_i, t_{i-1}]$ 上积分,得

$$\ln x_i - \ln x_{i-1} = \lambda(t_i - t_{i-1}) - \alpha S_{1i}$$

$$\ln y_i - \ln y_{i-1} = -\mu(t_i - t_{i-1}) + \beta S_{2i}$$

其中

$$S_{1i} = \int_{t_{i-1}}^{t_i} y dt, \quad S_{2i} = \int_{t_{i-1}}^{t_i} x dt, \qquad i = 1, 2, \cdots, m$$

于是得到方程组

$$\begin{cases} A_1 P_1 = B_1 \\ A_2 P_2 = B_2 \end{cases}$$

其中

$$A_1 = \begin{pmatrix} t_1 - t_0 & -S_{11} \\ t_2 - t_1 & -S_{12} \\ \vdots & \vdots \\ t_m - t_{m-1} & -S_{1m} \end{pmatrix}, \quad A_2 = \begin{pmatrix} t_1 - t_0 & -S_{21} \\ t_2 - t_1 & -S_{22} \\ \vdots & \vdots \\ t_m - t_{m-1} & -S_{2m} \end{pmatrix}$$

$$P_1 = \begin{pmatrix} \lambda \\ \alpha \end{pmatrix}, \quad P_2 = \begin{pmatrix} -\mu \\ \beta \end{pmatrix}$$

$$B_1 = \left(\ln\frac{x_1}{x_0}, \ln\frac{x_2}{x_1}, \cdots, \ln\frac{x_m}{x_{m-1}}\right)^T, \quad B_2 = \left(\ln\frac{y_1}{y_0}, \ln\frac{y_2}{y_1}, \cdots, \ln\frac{y_m}{y_{m-1}}\right)^T$$

因此方程组参数的最小二乘解为

$$P_1 = (A_1^T A_1)^{-1} A_1^T B_1, \quad P_2 = (A_2^T A_2)^{-1} A_2^T B_2$$

由于 x(t) 和 y(t) 均为未知,因此 S_{1i},S_{2i} 不能直接求出,利用数值积分法中的梯形公式得

$$S_{1i} = \int_{t_{i-1}}^{t_i} y dt \approx \frac{t_i - t_{i-1}}{2}(y_i + y_{i-1}),$$

$$S_{2i} = \int_{t_{i-1}}^{t_i} x dt \approx \frac{t_i - t_{i-1}}{2}(x_i + x_{i-1})$$

这样可以求得参数的近似值.

7.1.2 求解模型所需的知识点及其在 Matlab 中的实现方法

1. 矩阵求逆

命令:inv(A)

功能:求矩阵 A 的逆矩阵.

注意 矩阵求逆还有一个命令 pinv(A),用来求矩阵 A 的广义逆矩阵.

2. 求微分方程(组)的数值解

命令:[t,Y] = ode23('fun',tspan,Y0) 或 [t,y] = ode45('fun',tspan,y0)

功能:用龙格—库塔(Runge-Kutta)法求函数 fun 的数值解. fun 表示定义函数的文件名,tspan = [t0,tfina] 表示积分的起始值和终止值,Y0 是初始状态列向量.

7.1.3 模型求解在 Matlab 中的实现

1. 该模型在 Matlab 中的实现程序如下:

```
clear all, clc
X = [29.7  33.1  32.5  69.1  134.2  236.0  269.6  162.3  69.6
     39.8  34.0  20.7  21.7  37.6  57.6  124.6  215.8  272.7
     195.7  95.0  41.9  25.7  10.9  22.6  33.6  48.1  92.5
     183.3  268.5  230.6  111.1];
Y = [1.6  1.3  1.1  1.2  1.1  1.3  1.8  2.2  2.4  2.1
     1.9  1.5  1.5  1.2  0.9  1.1  1.3  1.6  2.3  2.4
     2.1  1.7  1.8  1.4  1.2  1.0  0.9  1.1  1.3  1.9
     2.3];
N = [X, Y];
T = [0:2:60];
for i = 2:31
A(i-1,1) = T(i) - T(i-1);
A(i-1,[2,3]) = (T(i) - T(i-1))/2 * [N(1,i) + N(1,i-1) - (N
```

```
(2,i) + N(2,i - 1))];
B(i - 1,:) = log([N(1,i)/N(1,i - 1) - N(2,i)/N(2,i - 1)]);
end;
A1 = A(:,[1 3]);
P1 = inv(A1' * A1)) * A1' * B(:,1)
A2 = A(:,[1 2]);
P2 = inv((A2' * A2)) * A2' * B(:,2)
```

将上述结果代入 Volterra-Lotoka 方程,然后用 Matlab 函数 ode45 求方程在时间 [0,60] 的数值解,作图观察田鼠及其天敌数量的周期震荡. 如图 7.2 所示,其中实线和虚线分别为田鼠和天敌的实际观测值,田鼠的数量等于 y 轴坐标值乘以 100.

2. 求方程 Volterra-Lotoka 的数值解的程序如下:

```
clear all, clc
X = [29.7  33.1  32.5  69.1  134.2  236.0  269.6  162.3  69.6
     39.8  34.0  20.7  21.7  37.6  57.6  124.6  215.8  272.7
     195.7  95.0  41.9  25.7  10.9  22.6  33.6  48.1  92.5
     183.3  268.5  230.6  111.1];
Y = [1.6  1.3  1.1  1.2  1.1  1.3  1.8  2.2  2.4  2.1  1.9
     1.5  1.5  1.2  0.9  1.1  1.3  1.6  2.3  2.4  2.1
     1.7  1.8  1.4  1.2  1.0  0.9  1.1  1.3  1.9  2.3];
T = [0:2:60];
[t,Y] = ode45(@vlok,[0:0.5:60],[29.7 1.6]);
plot(t,Y(:,1)/100,'k');
hold on;
plot(t,Y(:,2),'-.k');
title('田鼠及其天敌的 Volterra-Lotoka 模型拟合曲线');
xlabel('时间');ylabel('数量(只/每百亩)');
text(3,2,2,'田鼠 h rightarrow');gtext('hleftarrow 天敌');
legend('田鼠','天敌');
```

程序中所定义的函数 vlok 为:

```
[vlok.m]
function dydt = vlok(T,Y)
dydt = [(0.8958 - 0.5632 * Y(2)) * Y(1);(-0.1037 + 0.0010 * Y(1)) * Y(2)];
```

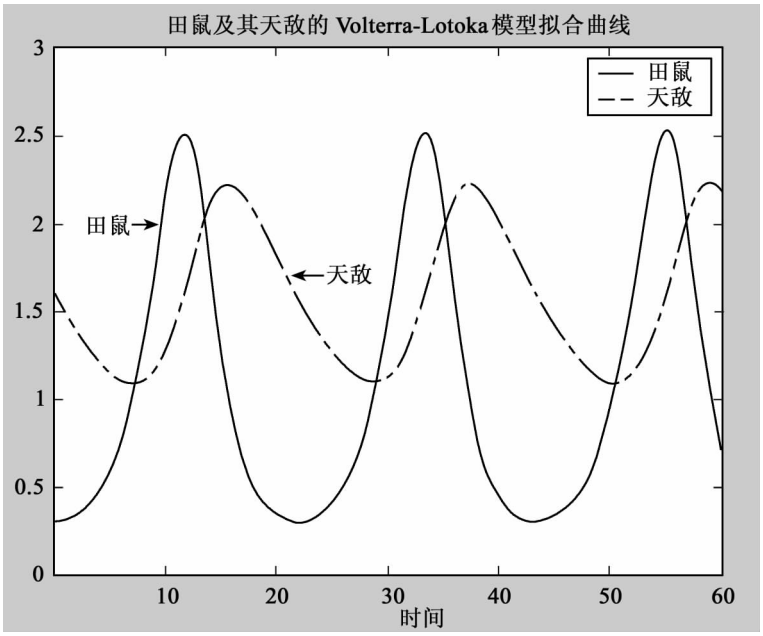


图 7.2　田鼠及其天敌的模拟曲线图

§7.2　核废料的妥善处理问题

有一段时间,美国原子能委员会是这样处理浓缩核放射性废物的:他们把这些废物装入密封性能很好的圆桶里,然后扔到水深 300ft (1 ft = 0.3048m,合 91.44m) 的海里,这种做法是否会造成核废料放射性污染,自然引起了生态学家和社会各界的关注. 当时美国原子能委员会一再保证,用来装核废料的圆桶非常坚固,投放到 300ft 的海里决不会破裂,因此也就不会造成核废料的泄漏,这种做法是绝对安全的. 但是一些工程技术人员却对此表示怀疑,他们认为圆桶在和海底相撞时有可能发生破裂. 而美国原子能委员会有些专家仍然坚持自己的意见. 于是,在当时引起了一场争论. 已知当时美国装核废料使用的是 55gal (加仑),(1gal = 3.785411L) 的圆桶,装满核放射性废料后的重量为 427.436pdl (磅) (1pdl = 0.4536kg),经过多次试验测得圆桶在下沉时所受的浮力为 f = 470.327pdl (合 2090.735N),阻力系数为 c = 0.08,圆桶发生破裂的直线极限速度是 40ft/s (合 12.192m/s). 现在约定圆桶在以直线方向往下沉的情况下分析

圆桶是否可能破裂.

7.2.1　问题分析及模型的建立

判断圆桶在沉入海底的过程中是否会发生破裂,关键在于圆桶在不破裂的情况下能承受的最大冲撞速度以及在圆桶到达海底时的末速度. 已知圆桶发生破裂的直线极限速度是 12.192m/s,所以,只要计算出圆桶沉入海底时的末速度,就可以知道其结果.

设圆桶在下沉的过程中所受到的阻力为 d,则有

$$d = cv$$

其中 $c = 0.08$ 为阻力常数,v 为下沉速度. 在垂直方向下的有向直线为 y 轴,在海平面上取一条有向直线为 x 轴,根据牛顿第二定理,圆桶下沉应满足下面的微分方程

$$m\frac{d^2y}{dt^2} = mg - f - cv \tag{7.5}$$

这里 m 为圆桶装满核废料时的质量,g 为重力加速度. 根据 $v = \frac{dy}{dt}$可得

$$\begin{cases} \frac{d^2y}{dt^2} + \frac{c}{m}\frac{dy}{dt} = \frac{mg - f}{m} \\ y(0) = 0, y'(0) = 0 \end{cases} \tag{7.6}$$

这是圆桶下沉位移 y 与时间 t 之间的二阶线性微分方程. 因为 $v = \frac{dy}{dt}$,因此可以将方程 (7.6) 化为速度 v 与时间 t 之间的一阶线性微分方程

$$\begin{cases} \frac{dv}{dt} + \frac{c}{m}v = \frac{mg - f}{m} \\ v(0) = 0 \end{cases} \tag{7.7}$$

还可以通过另外一种方法求得圆桶下沉时位移 y 与速度 v 之间的函数关系. 根据复合函数求导的链式法则得

$$\frac{dv}{dt} = \frac{dv}{dy}\frac{dy}{dt} \tag{7.8}$$

将式 (7.8) 代入式 (7.7) 得

$$v\frac{dv}{dy} + \frac{c}{m}v = \frac{mg - f}{m}$$

化简得

$$\begin{cases} \frac{dy}{dv} = \frac{mv}{mg - f - cv} \\ y(0) = 0 \end{cases} \tag{7.9}$$

式 (7.9) 就是圆桶在下沉过程中位移 y 与速度 v 之间的一阶微分方程.

7.2.2 求解模型所需的知识点及其在 Matlab 中的实现方法

1. 求微分方程 (组) 的解析解

命令形式 1:dsolve ('eq1,eq2,…')

功能:求解微分方程 (组) eq1 (eq2,eq3,…) 的通解.

命令形式 2:dsolve ('eq1,eq2,…','cond1,cond2,…')

功能:求解微分方程 (组) eq1 (eq2,eq3,…) 在条件 cond1 (cond2,cond3,…) 下的特解.

2. 求函数极限

命令形式:limit (f,x,a)

功能:求函数 f (x) 的极限 $\lim\limits_{x\to a} f(x)$.

3. 求微分方程的数值解

命令形式:ode23 ('fun',tspan,y0) 或 ode45 ('fun',tspan,y0)

功能:用 2 阶 (3 阶) 或 4 阶 (5 阶) 龙格—库塔方法求微分方程数值解.

7.2.3 模型在 Matlab 中的实现

1. 在 Matlab 中求圆桶下沉速度的解析解

首先求方程 (7.6) 的通解,用 dsolve 命令即可求得:

dsolve ('D2y + c/m * Dy = (m * g - f) /m','Dy (0) = 0','y (0) = 0')

其运算结果为

ans = - 1/c * (- m * g + f) * t + m * (- m * g + f) /c 2 - m * (- m * g + f) /c 2 * exp (- c/m * t)

写成一般格式即为

$$y = -m\frac{f-mg}{c^2}e^{-\frac{c}{m}t} + \frac{mg-f}{c}t + m\frac{f-mg}{c^2}$$

求解方程

$$-m\frac{f-mg}{c^2}e^{-\frac{c}{m}t} + \frac{mg-f}{c}t + m\frac{f-mg}{c^2} = 91.44$$

可得圆桶沉入 91.44m 海底所用的时间. 方程 (7.6) 是一个可降阶的线性微分方程,令 $\frac{d^2y}{dt^2} = \frac{dP}{dt}$,此时即为方程 (7.7),可以用一阶微分方程的求解公式 dsolve 来求其通解,其命令形式为:

dsolve ('Dv + c/m * v = (m * g - f) /m','v (0) = 0')

其运算结果为：

```
ans  = 1/c*m*g - 1/c*f + exp(- c/m*t)/c*(- m*g + f)
```

写成一般格式即为

$$v(t) = \frac{mg - f}{c}\left(1 - e^{-\frac{c}{m}t}\right)$$

当 $t\to$ 时，可以求得圆桶下沉时的极限速度为

$$\lim_{t\to} v(t) = \lim_{t\to}\frac{mg - f}{c}\left(1 - e^{-\frac{c}{m}t}\right) = \frac{mg - f}{c} \approx 3173.3\text{m/s}$$

再求方程 (7.9) 的解析解，其命令形式：

```
dsolve('Dy = m*v/(m*g - f - c*v)','y(0) = 0','v')
```

其运算结果为

```
ans = - m/c*v - 1/c^2*log(- m*g + f + c*v)*m^2*g +
      (m/c^2*log(- m*g + f + c*v)*f - m*log(- m*g + f)*(- m*g + f)/c^2
```

写成一般格式即为

$$y = \frac{m}{c^2}\left[-vc + (f - mg)\ln\frac{mg - f - cv}{mg - f}\right] \tag{7.10}$$

要求得圆桶的末速度，只需求出式 (7.10) 这个函数 $y = 91.44$ 时相应的速度即可．但是，从式 (7.10) 这个方程很难直接求出速度 $v(y)$，即方程

$$\frac{m}{c^2}\left[-vc + (f - mg)\ln\frac{mg - f - cv}{mg - f}\right] = 91.44$$

很难求出．

可以验证，式 (7.10) 所表示的函数在定义域内是单调上升的，所以可以将圆桶能够承受的极限速度代入函数计算得相应的 y 值，看是否超过了 91.44，将 $v = 12.192$ 代入式 (7.10) 可以计算出相应的 y 值为 70.22187281370590．显然，圆桶在未下沉到海底，而是下沉到 70.23m 后就可能开始泄漏了．

2. 在 Matlab 中求模型的数值解

从上述分析可知：该模型中描述问题的微分方程比较复杂，尽管所论方程是一阶的，但也很难求解．在这种情况下一般求微分方程的数值解．

下面用 Matlab 求方程 (7.6)、方程 (7.7)、方程 (7.9) 的数值解，应用龙格—库塔命令 ode23 或 ode45 来求解．

模型实现的 Matlab 程序如下：

```
% 定义微分方程函数
function y = hefeiliao(v,y)
m = 239.2450 ;g = 9.8 ;
f = 2090.735 ;c = 0.08 ;
```

```
y = (m*v)/(m*g - f - c*v);
fplot('hefeiliao',[0,14]);
% 这是圆桶下沉速度与深度之间的关系曲线,横轴为 v 轴
[v,y] = ode23('hefeiliao',0,12.192)
```

其运算结果为：

```
v = 0    1.2912    2.4384    3.6576    4.8768    6.0960
    7.3152    8.5344    9.7536    10.9728    12.1920
y = 0    0.7006    2.8031    6.3086    11.2182    17.5329
    25.2539    34.3822    44.9189    56.8561    70.2219
```

从上述数据可以看到,当圆桶下沉速度达到安全状态的极限速度时,圆桶才下沉了 70.2219m,距离 94.44m 还有很远,因此可知这个深度很难保证圆桶不泄漏核废料.

如果用 4,5 阶龙格—库塔命令 ode45 来计算,结果也是一样的. 计算的命令形式为：[v, y] = ode45('hefeiliao', 0,13.91).

§7.3 状态转移方程组模型

随着计算机通信网络系统特别是 Internet 网络的应用日益广泛,计算机网络可靠性分析显得越来越重要,提高系统的可靠性意义重大. 研究和分析具有实用性的高可靠性计算机通信网络系统,是国际上非常活跃的一个研究方向. 计算机随时可能发生三种状态——无故障、间歇故障和永久故障. 因此,计算机一般处于三种工作状态：无故障工作、带故障工作和不工作. 这三种状态之间的转移过程如图 7.3 所示. 试建立该系统的状态转移模型,并进行可靠性分析.

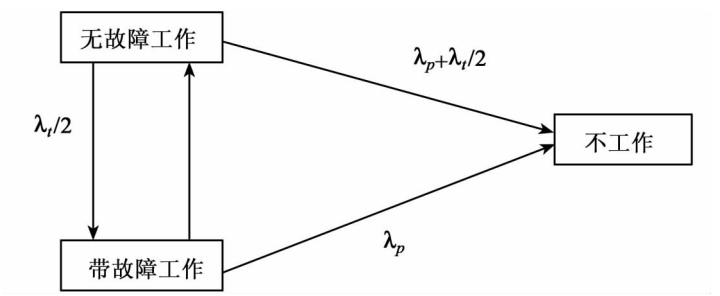


图 7.3

7.3.1　问题分析及模型的建立

该问题属于状态转移问题，利用马尔科夫状态转移原理，用 $P_1(t)$、$P_2(t)$ 和 $P_3(t)$ 分别表示系统处于无故障工作、带故障工作和不工作三种状态的概率，则有以下状态转移方程组

$$\begin{cases} \dfrac{dP_1(t)}{dt} = -\left[\left(\lambda_p + \dfrac{\lambda_t}{2}\right) + \dfrac{\lambda_t}{2}\right]P_1(t) + \gamma P_2(t) \\ \qquad\qquad = -(\lambda_p + \lambda_t)P_1(t) + \gamma P_2(t) \\ \dfrac{dP_2(t)}{dt} = \left(\dfrac{\lambda_t}{2}\right)P_1(t) - [\gamma + (\lambda_p + \lambda_t)]P_2(t) \\ \dfrac{dP_3(t)}{dt} = \left(\lambda_p + \dfrac{\lambda_t}{2}\right)P_1(t) + (\lambda_p + \lambda_t)P_2(t) \end{cases} \tag{7.11}$$

初始条件为 $P_1(0)=1$，$P_2(0)=0$，$P_3(0)=0$，参数取值：

$$\lambda_p: 10^{-5} \sim 10^{-4}, \lambda_t: 10^{-4} \sim 10^{-3}, \gamma: 0.01 \sim 0.1$$

这是一个带参数的微分方程组模型．根据上面分析可知，微分方程组(7.11)即为该问题的数学模型．

7.3.2　求解模型所需的知识点及其在 Matlab 中的实现方法

1. 求矩阵特征根和特征向量

命令形式：[V,D] = eig(A)

功能：返回方阵 A 的特征向量矩阵 V 和特征值矩阵 D.

2. 求矩阵 A 的逆矩阵

命令形式：inv(A)

功能：求矩阵 A 的逆矩阵．

3. 求微分方程的数值解

命令形式：ode23() 或 ode45()

功能：用龙格—库塔方法求微分方程数值解．

7.3.3　模型求解在 Matlab 中的实现

求解这个带参数的微分方程组模型有两种方法：一是用特征根法求解析解，另一种是用数值解法求数值解．

假定模型中的参数取下限值，即 $\lambda_p = 10^{-5}$，$\lambda_t = 10^{-4}$，$\gamma = 0.01$，分别求解如

下：

1. 求解析解

利用常系数微分方程组的特征根法求解析解，程序如下：

```
lp = 10 (-5);lt = 10 (-4);
gm = 0.01;
A = [-(lp + lt), gm, 0;lt/2, -(gm + lp + lt), 0;lp + lt/2, lp + lt, 0];
    [l,v] = eig(A)
e = inv(l) * [1;0;0]
```

其运算结果为：

```
l =0    0.7053    -0.7053
   0    -0.7089    -0.0035
   1.0000    0.0035    0.7089
V =0      0          0
   0      -0.0102    0
   0      0          -0.00015
```

即矩阵 A 对应的特征值为（0，-0.0102，-0.00015），而最终结果为

```
e =1.0000
   0.0070
   -1.4108.
```

2. 数值解法

求微分方程组数值解的 Matlab 程序如下：

```
%首先写出微分方程组的 M 函数文件
Function    xdot = eqs0(t,p,flag,lp,lt,gm)
    A = [-(lp + lt), gm,0;lt/2, -(gm + lp + lt),0;lp + lt/2,lp + lt,0];
    p = [p(1);p(2);p(3)];
    xdot = a * p;
```

命令函数文件：

```
    ts = [0 1000];
    p0 = [1;0;0];
    lp = 10 (-5);
    lt = 10 (-4);gm =0.01;
    [t,p] = ode23('eqs0',ts,p0,[],lp,lt,gm)
    plot(t,1 - p(:,3));
    xlabel('时间 t(小时)');
```

```
ylabel('可靠度 R(t)');
title('参数取值 lp = 0.00001;lt = 0.0001;gm = 0.01');
grid on;
```

输出结果如下(如图 7.4 所示):

```
p = 1.0000        0        0
    0.9999   0.0001   0.0001
    0.9991   0.0004   0.0005
    0.9955   0.0021   0.0025
    0.9845   0.0071   0.0085
    0.9735   0.0121   0.0145
    0.9625   0.0171   0.0205
    0.9515   0.0221   0.0265
    0.9405   0.0271   0.0325
    0.9295   0.0321   0.0385
    0.9185   0.0371   0.0445
    0.9075   0.0421   0.0505
    0.8965   0.0471   0.0565
    0.8900   0.0500   0.0600
t = 1.0e+003 *
         0
    0.0013
    0.0080
    0.0413
    0.1413
    0.2413
    0.3413
    0.4413
    0.5413
    0.6413
    0.7413
    0.8413
    0.9413
    1.0000
```

计算结果表明:在时间 t = 1000h 的情况下,

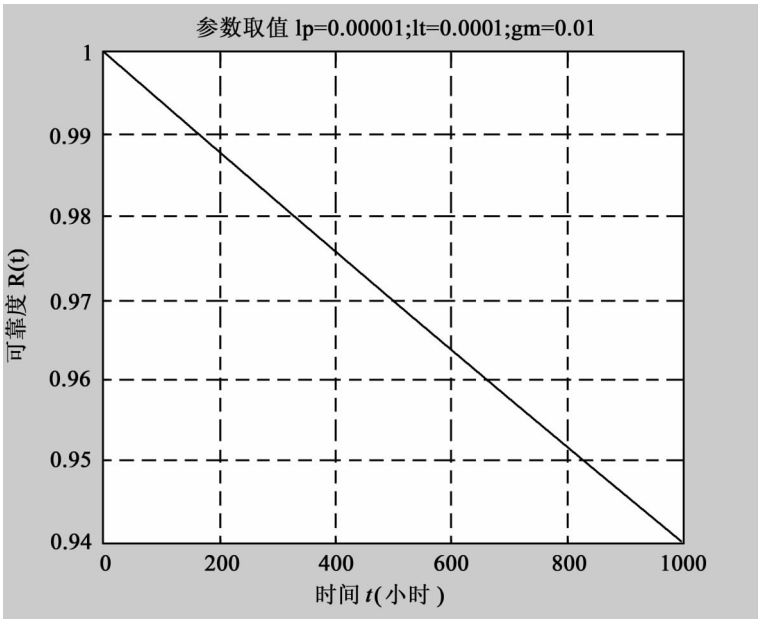


图 7.4

[p1 (t) , p2 (t) , p3 (t)] = [0.9369, 0.0047, 0.0584] .

显示系统工作的概率为 94.18%，从图 7.4 中还可以看到系统可靠性变化更加细致的情况，何时可靠度达到 99%，何时达到 98% 等 .

§7.4 真题解析：彩票中的数学

问题的提出：

近年来“彩票飓风”席卷中华大地，巨额诱惑使越来越多的人加入到“彩民”的行列，目前流行的彩票主要有“传统型”和“乐透型”两种类型 .

“传统型”采用“10 选 6 + 1”方案：先从 6 组 0 ~9 号球中摇出 6 个基本号码，每组摇出一个，然后从 0 ~4 号球中摇出一个特别号码，构成中奖号码. 投注者从 0 ~9 十个号码中任选 6 个基本号码（可以重复），从 0 ~4 中选一个特别号码，构成一注，根据单注号码与中奖号码相符的个数多少及顺序确定中奖等级 . 以中奖号码“abcdef + g”为例说明中奖等级，如表 7.3 所示（X 表示未选中的号码）.

表 7.3

中奖等级	10 选 6 + 1 (6 + 1/10)		
	基本号码	特别号码	说　明
一等奖	abcdef g		选 7 中 (6 + 1)
二等奖	abcdef		选 7 中 (6)
三等奖	abcdeX　Xbcdef		选 7 中 (5)
四等奖	abcdXX　XbcdeX　XXcdef		选 7 中 (4)
五等奖	abcXXX　XbcdXX　XXcdeX　XXXdef		选 7 中 (3)
六等奖	abXXXX　XbcXXX　XXcdXX　XXXdeX　XXXXef		选 7 中 (2)

“乐透型”有多种不同的形式，比如“33 选 7”的方案：先从 01 ~ 33 个号码球中一个一个地摇出 7 个基本号，再从剩余的 26 个号码球中摇出一个特别号码．投注者从 01 ~ 33 个号码中任选 7 个组成一注（不可重复），根据单注号码与中奖号码相符的个数多少确定相应的中奖等级，不考虑号码顺序．又如“36 选 6 + 1”的方案，先从 01 ~ 36 个号码球中一个一个地摇出 6 个基本号，再从剩下的 30 个号码球中摇出一个特别号码．从 01 ~ 36 个号码中任选 7 个组成一注（不可重复），根据单注号码与中奖号码相符的个数多少确定相应的中奖等级，不考虑号码顺序．这两种方案的中奖等级如表 7.4 所示．

表 7.4

中奖等级	33 选 7 (7/33)			6 选 6 + 1 (6 + 1/36)		
	基本号码	特别号码	说 明	基本号码	特别号码	说 明
一等奖	●●●●●●●		选 7 中 (7)	●●●●●●	★	选 7 中 (6 + 1)
二等奖	●●●●●●○	★	选 7 中 (6 + 1)	●●●●●●		选 7 中 (6)
三等奖	●●●●●●○		选 7 中 (6)	●●●●●○	★	选 7 中 (5 + 1)
四等奖	●●●●●○○	★	选 7 中 (5 + 1)	●●●●●○		选 7 中 (5)
五等奖	●●●●●○○		选 7 中 (5)	●●●●○○	★	选 7 中 (4 + 1)
六等奖	●●●●○○○	★	选 7 中 (4 + 1)	●●●●○○		选 7 中 (4)
七等奖	●●●●○○○		选 7 中 (4)	●●●○○○	★	选 7 中 (3 + 1)

注：●为选中的基本号码；★ 为选中的特别号码；○ 为未选中的号码.

以上两种类型的总奖金比例一般为销售总额的 50%，投注者单注金额为 2 元，单注若已得到高级别的奖就不再兼得低级别的奖. 现在常见的销售规则及相应的奖金设置方案如表 7.5 所示，其中一、二、三等奖为高项奖，后面的为低项奖. 低项奖数额固定，高项奖按比例分配，但一等奖单注保底金额 60 万元，封顶金额 500 万元，各高项奖额的计算方法为

[(当期销售总额 ×总奖金比例) - 低项奖总额] ×单项奖比例

(1) 根据这些方案的具体情况，综合分析各种奖项出现的可能性、奖项和奖金额的设置以及对彩民的吸引力等因素评价各方案的合理性.

(2) 设计一种“更好”的方案及相应的算法，并据此给彩票管理部门提出建议.

(3) 给报纸写一篇短文，供彩民参考.

表 7.5

序号	奖项 / 方案	一等奖比例	二等奖比例	三等奖比例	四等奖金额	五等奖金额	六等奖金额	七等奖金额	备注
1	6 + 1 /10	50%	20%	30%	50				按序
2	6 + 1 /10	60%	20%	20%	300	20	5		按序
3	6 + 1 /10	65%	15%	20%	300	20	5		按序
4	6 + 1 /10	70%	15%	15%	300	20	5		按序
5	7 /29	60%	20%	20%	300	30	5		
6	6 + 1 /29	60%	25%	15%	200	20	5		
7	7 /30	65%	15%	20%	500	50	15	5	
8	7 /30	70%	10%	20%	200	50	10	5	
9	7 /30	75%	10%	15%	200	30	10	5	
10	7 /31	60%	15%	25%	500	50	20	10	
11	7 /31	75%	10%	15%	320	30	5		
12	7 /32	65%	15%	20%	500	50	10		
13	7 /32	70%	10%	20%	500	50	10		
14	7 /32	75%	10%	15%	500	50	10		
15	7 /33	70%	10%	20%	600	60	6		

续表

序号	奖项 / 方案	一等奖比例	二等奖比例	三等奖比例	四等奖金额	五等奖金额	六等奖金额	七等奖金额	备注
16	7/33	75%	10%	15%	500	50	10	5	
17	7/34	65%	15%	20%	500	0	6		
18	7/34	68%	12%	20%	500	50	10	2	
19	7/35	70%	15%	15%	300	50	5		
20	7/35	70%	10%	20%	500	100	30	5	
21	7/35	75%	10%	15%	1000	100	50	5	
22	7/35	80%	10%	10%	200	50	20	5	
23	7/35	100%	20%	20%	4	2			无特别号
24	6+1/36	75%	10%	15%	500	100	10	5	
25	6+1/36	80%	10%	10%	500	100	10		
26	7/36	70%	10%	20%	500	50	10	5	
27	7/37	70%	15%	15%	1500	100	50		
28	6/40	82%	10%	8%	200	10	1		
29	5/60	60%	20%	20%	300	30			

评价一个方案的优劣或合理性如何，主要取决于彩票公司和广大彩民两方面的利益．事实上，公司和彩民各得销售总额的 50% 是确定的，双方的利益主要取决于销售总额的大小，即双方的利益都与销售额成正比．因此，人们关心的问题是怎样才能有利于销售额的增加？即公司采用什么样的方案才能吸引广大的彩民积极踊跃地购买彩票？具体地讲，问题涉及到一个方案的设置使彩民获奖的可能性有多大？奖金额有多少？对彩民的吸引力有多大？广大彩民如何看待各奖项的设置？即彩民的心理曲线怎样？另外，一个方案对彩民的影响程度可能与区域有关．即与彩民所在地区的经济状况以及收入和消费水平有关．为此，我们要考查一个方案的合理性问题，需要考虑以上这些因素的影响，这是我们建立数学模型的关键所在．

7.4.1　模型假设与符号说明

彩票摇奖是公平、公正的，各号码的出现是随机的；彩民购买彩票是随机

的独立事件；对同一方案中高级别奖项的奖金比例或奖金额不应低于相对低级别的奖金比例或奖金额；根据我国的现行制度，假设我国居民的平均工作年限为 $T=35$ 年，给出以下参数定义：

r_j——第 j 等（高项）奖占高项奖总额的比例，$j=1,2,3$；

x_i——第 i 等奖奖金额均值，$1\leqslant i\leqslant 7$；

p_i——彩民中第 i 等奖 x_i 的概率，$1\leqslant i\leqslant 7$；

$\mu(x_i)$——彩民对某个方案第 i 等奖的满意度，即第 i 等奖对彩民的吸引力，$1\leqslant i\leqslant 7$；

λ——某地区的平均收入和消费水平的相关因子，称为“实力因子”，一般为常数；

F——彩票方案的合理性指标，即方案设置对彩民吸引力的综合指标.

7.4.2 模型的准备

1. 彩民获各项奖的概率

从已给的 29 种方案可知，可以将其分为四类，K_1：10 选 6+1（6+1/10）型；K_2：n 选 m（m/n）型；K_3：n 选 m+1（m+1/n）型；K_4：n 选 m（m/n）无特别号型. 分别给出各种类型方案的彩民获各奖项的概率公式：

(1) K_1：10 选 6+1（6+1/10）型

$$p_1=\frac{1}{5\times 10^6}=2\times 10^{-7},\quad p_2=\frac{4}{5\times 10^6}=8\times 10^{-7},\quad p_3=\frac{2\times C_9^1}{10^6}=1.8\times 10^{-5}$$

$$p_4=\frac{2C_9^1C_{10}^1+C_9^1C_9^1}{10^6}=2.61\times 10^{-4}$$

$$p_5=\frac{2C_9^1C_{10}^1C_{10}^1+2C_9^1C_9^1C_{10}^1}{10^6}=3.42\times 10^{-3}$$

$$p_6=\frac{2\times C_9^1C_{10}^1C_{10}^1C_{10}^1+3\times 2C_9^1C_9^1C_{10}^1C_{10}^1-(3\times C_9^1\times C_9^1+2\times C_9^1)}{10^6}=4.1995\times 10^{-2}$$

(2) K_2：n 选 m（m/n）型

$$p_1=\frac{1}{C_n^m},\quad p_2=\frac{C_m^{m-1}}{C_n^m},\quad p_3=\frac{C_m^{m-1}C_{n-(m+1)}^1}{C_n^m},\quad p_4=\frac{C_m^{m-2}C_{n-(m+1)}^1}{C_n^m},$$

$$p_5=\frac{C_m^{m-2}C_{n-(m+1)}^2}{C_n^m},\quad p_6=\frac{C_m^{m-3}C_{n-(m+1)}^2}{C_n^m},\quad p_7=\frac{C_m^{m-3}C_{n-(m+1)}^3}{C_n^m}.$$

(3) K_3：n 选 m+1（m+1/n）型

$$p_1=\frac{1}{C_n^{m+1}},\quad p_2=\frac{C_n^1-(m_1+1)}{C_n^{m+1}}$$

$$p_3 = \frac{C_n^{m-1} C_{n-(m+1)}^1}{C_n^{m+1}}$$

$$p_4 = \frac{C_n^{m-1} C_{n-(m+1)}^2}{C_n^{m+1}}$$

$$p_5 = \frac{C_m^{m-2} C_{n-(m+1)}^2}{C_n^{m+1}}$$

$$p_6 = \frac{C_m^{m-2} C_{n-(m+1)}^3}{C_n^{m+1}}$$

$$p_7 = \frac{C_m^{m-3} C_{n-(m+1)}^3}{C_n^{m+1}}.$$

(4) K_4：n 选 m (m/n) 无特别号型

$$p_1 = \frac{1}{C_n^m}, \quad p_2 = \frac{C_m^{m-1} C_{n-m}^1}{C_n^m}, \quad p_3 = \frac{C_m^{m-2} C_{n-m}^2}{C_n^m},$$

$$p_4 = \frac{C_m^{m-3} C_{n-m}^3}{C_n^m}, \quad p_5 = \frac{C_m^{m-4} C_{n-m}^4}{C_n^m}$$

各种方案的各个奖项获奖概率 p_i 及获奖总概率 $P = \sum p_i$ 计算如表 7.6 所示.

表 7.6

方　案	p_1 $\times 10^{-7}$	p_2 $\times 10^{-7}$	p_3 $\times 10^{-5}$	p_4 $\times 10^{-4}$	p_5 $\times 10^{-3}$	p_6 $\times 10^{-4}$	p_7	$P = \sum_i p_i$
6 + 1 /10	2	8	1. 8	2. 61	3. 42	4. 1995		0. 045695
7 /29	6. 40705	4. 48494	9. 4184	2. 8255	2.8255^{3}	4. 7092	0. 029825	0. 037742
6 + 1 /29	6. 40705	1. 4096	8. 4573	8. 8880	2. 2200	1. 4800	0. 019734	0. 037742
7 /30	4. 91207	3. 43845	7. 5646	2. 2694	2. 3828	3. 9714	0. 026476	0. 033137
7 /31	3. 80290	2. 66203	6. 1227	1.8368^{-4}	2. 0205	3.3675^{3}	0. 023572	0. 029208
7 /32	2. 97101	2. 07971	4. 0991	1. 4974	1. 722	2. 8700	0. 021047	0. 025832
7 /33	2. 34080	1. 63856	4. 0964	1.2289^{4}	1. 4747	2.4578^{3}	0. 018843	0. 022941
7 /34	1. 85887	1. 30121	3. 3831	1.0149^{4}	1. 2687	2. 1145	0. 016916	0. 020436
7 /35	1. 48709	1. 04097	2. 8106	8. 4318	1.0961^{-3}	1.8269^{-3}	0. 015224	0. 08261

续表

方案	p_1 $\times 10^{-7}$	p_2 $\times 10^{-7}$	p_3 $\times 10^{-5}$	p_4 $\times 10^{-4}$	p_5 $\times 10^{-3}$	p_6 $\times 10^{-4}$	p_7	$P = \sum_i p_i$
7/36	1.19794	8.38556	2.3480	7.0439	9.5092	1.5849	0.013736	0.016367
6+1/36	1.19794	3.47402	2.0844^5	2.9182	7.2954	6.5659	0.008755	0.016367
7/37	9.71301	6.79911	1.9717	5.9152	8.2813	1.3802	0.012422	0.014710
6/40	2.6053	1.5632	5.1584	1.2896	2.0634^3	2.7512	0.028428	0.033425
5/60	1.831	9.155	4.9437	9.8874	2.6202	2.6202	0.045416	0.050806

2. 确定彩民的心理曲线

一般说来，人们的心理变化是一个模糊的概念. 在此，彩民对一个方案的各个奖项及奖金额的看法（即对彩民的吸引力）的变化就是一个典型的模糊概念. 由模糊数学隶属度的概念和心理学的相关知识，根据人们通常对一件事物的心理变化一般遵循的规律，不妨定义彩民的心理曲线为

$$\mu(x) = 1 - e^{-\left(\frac{x}{\lambda}\right)^2} \qquad (\lambda > 0)$$

其中 λ 表示彩民平均收入的相关因子，称为实力因子，一般为常数.

3. 计算实力因子 λ

实力因子是反映一个地区的彩民的平均收入和消费水平的指标，确定一个地区的彩票方案应该考虑所在地区的实力因子，在我国不同地区的收入和消费水平是不同的，因此，不同地区的实力因子应有一定的差异，目前各地区现行的方案不尽相同，要统一来评估这些方案的合理性，就应该对同一个实力因子进行研究. 为此，我们以中等地区的收入水平（或全国平均水平）为例进行研究. 根据相关网站的统计数据，不妨取人均年收入为 1.5 万元，按我国的现行制度，平均工作年限 $T = 35$ 年，则人均总收入为 52.5 万元，于是，当 $x_0 = 52.5$ 万元时，取

$$\mu(x_0) = 1 - e^{-\left(\frac{x_0}{\lambda}\right)^2} = 0.5 \qquad \text{（即吸引力的中位数）}$$

则有

$$\lambda = \frac{5.25 \times 10^5}{\sqrt{-\ln 0.5}} \approx 6.30589 \times 10^5.$$

同理，可以计算出年收入 1 万元、2 万元、2.5 万元、3 万元、4 万元、5 万元、10 万元的实力因子如表 7.7 所示.

表 7.7

年收入 / 指标	1 万元	1.5 万元	2 万元	2.5 万元	3 万元	4 万元	5 万元	10 万元
λ	420 393	630 589	840 786	1 050 982	1 261 179	1 681 571	2 101 964	4 203 928

7.4.3　模型的建立与求解

问题（一）

要综合评价这些方案的合理性，应该建立一个能够充分反映各种因素的合理性指标函数．因为彩民购买彩票是一种风险投资行为，为此，我们根据决策分析的理论，考虑到彩民的心理因素的影响，可取

$$\mu(x) = 1 - e^{-\left(\frac{x}{\lambda}\right)^2} \qquad (\lambda > 0)$$

为风险决策的益损函数，于是作出如下的指标函数

$$F = \sum_{i=1}^{7} p_i \mu(x_i) \tag{7.12}$$

即表示在考虑彩民的心理因素的条件下，一个方案的奖项和奖金设置对彩民的吸引力．

另一方面，由题意知，单注所有可能的低项奖金总额为 $L = \sum_{i=4}^{7} p_i x_i$，根据高项奖的计算公式得单注可能的第 j 项（高项奖）奖金额为

$$p_j x_j = (1 - L) r_j = \left(1 - \sum_{i=4}^{7} p_i x_i\right) r_j, \quad j = 1,2,3$$

故平均值为

$$x_j = \frac{\left(1 - \sum_{i=4}^{7} p_i x_i\right) r_j}{p_j}, \quad j = 1,2,3 \tag{7.13}$$

于是由式（7.12）、式（7.13）得

$$\begin{cases} F = \sum_{i=1}^{7} p_i \mu(x_i) \\ x_j = \dfrac{\left(1 - \sum_{i=4}^{7} p_i x_i\right) r_j}{p_j}, \quad j = 1,2,3 \\ \mu(x_i) = 1 - e^{-\left(\frac{x_i}{\lambda}\right)^2}, i = 1,2,\cdots,7 \\ \lambda = 6.30\ 589 \times 10^5 \end{cases} \tag{7.14}$$

利用 Matlab 可以计算出 29 种方案的合理性指标值 F 及高项奖的期望值，排在前三位的如表 7.8 所示.

表 7.8

方案 \ 指标		F	x_1	x_2	x_3	排序
9	7/30	4.009×10^{-7}	1.086×10^{6}	20 679	1 410	1
11	7/31	3.784×10^{-7}	1.704×10^{6}	32 448	2 116	2
5	7/29	3.637×10^{-7}	7.557×10^{5}	35 984	1 714	3

问题（二）

根据问题（一）的讨论，现在的问题是取什么样的方案 m/n（n 和 m 取何值）、设置哪些奖项、高项奖的比例 r_j（j = 1，2，3）为多少和低项奖的奖金额 x_i（i = 4，5，6，7）为多少时，使目标函数 $F=\sum_{i=1}^{7}p_i\mu(x_i)$ 有最大值.

设以 m，n，r_j（j = 1，2，3），x_i（i = 4，5，6，7）为决策变量，以它们之间所满足的关系为约束条件，则可以得到非线性规划模型

$$\max F=\sum_{i=1}^{7}p_i\mu(x_i)$$

$$\text{s.t.}\begin{cases} x_j=\dfrac{\left(1-\sum_{i=4}^{7}p_i x_i\right)r_j}{p_j},\quad j=1,2,3 & (1)\\ \mu(x_i)=1-e^{-\left(\frac{x_i}{\lambda}\right)^2},(i=1,2,\cdots,7),\lambda=6.30589\times10^5 & (2)\\ r_1+r_2+r_3=1 & (3)\\ 0.5\leqslant r_1\leqslant 0.8 & (4)\\ 6\times10^5\leqslant x_1\leqslant 5\times10^6 & (5)\\ a_i\leqslant\dfrac{x_i}{x_{i+1}}\leqslant b_i,i=1,2,\cdots,6 & (6)\\ p_i<p_{i+1},i=1,2,\cdots,6 & (7)\\ 5\leqslant m\leqslant 7 & (8)\\ 29\leqslant n\leqslant 60 & (9)\\ r_j>0,x_i\geqslant 0;m,n\text{ 为正整数} & \end{cases}$$

关于约束条件的说明：

1. 条件 (1)、(2) 同问题 (一).

2. 条件 (3)、(4) 是对高项奖的比例约束，r_1 的值不能太大或太小，条件 (4) 是根据已知的方案确定的.

3. 条件 (5) 是根据题意中一等奖的保底额和封顶额确定的.

4. 条件 (6) 中的 a_i，b_i ($i=1,2,\cdots,6$) 分别为 i 等奖的奖金额 x_i 比 $i+1$ 等奖的奖金额 x_{i+1} 高的倍数，可以由问题 (一) 的计算结果和已知各方案的奖金数额统计得

$a_1=10$，$b_1=233$；$a_2=4$，$b_2=54$；$a_3=3$，$b_3=17$；$a_4=4$，$b_4=20$；$a_5=2$，$b_5=10$；$a_6=2$，$b_6=10$

5. 条件 (7) 是根据实际问题确定的，实际中高等奖的概率 p_i 应小于低等奖的概率 p_{i+1}，其值主要由 m，n 确定.

6. 条件 (8)、(9) 是对方案中 m，n 取值范围的约束，是由已知的方案确定的.

这是一个较复杂的非线性 (整数) 规划，其中概率 p_i 的取值分为四种不同的情况，K_1，K_2，K_3，K_4 由整数变量 m，n 确定，一般的求解是困难的. 为此，利用 Matlab 可以求得最优解为 $\{K_2, 6, 32, 0.8, 0.09, 0.11, 200, 10, 1, 0\}$，最优值为 $F=6.8399\times10^{-7}$. 故对应的最优方案为：32 选 6 (6/32)，一、二、三等奖的比例分别为 8%、9%、11%，四、五、六、七等奖的金额分别为 200 元、10 元、1 元、0 元.

上述是针对中等收入水平的彩民情况考虑的，对于经济发达地区和欠发达地区应有所不同. 这里分别对年收入 1 万元、2 万元、2.5 万元、3 万元、4 万元、5 万元、10 万元，工作年限均 35 年的情况进行了讨论，给出适用于相应各种情况的最优方案，如表 7.9 所示.

表 7.9

年收入 / 指标	1 万元	2 万元	2.5 万元	3 万元	4 万元	5 万元	10 万元
λ	420393	840786	1050982	1261179	1681571	2101964	4203928
最优方案	5+1/33	6/32	7/30	6/37	6+1/32	7/33	7/35
F	8.255×10^{-7}	4.623×10^{-7}	4.103×10^{-7}	3.223×10^{-7}	2.475×10^{-7}	2.075×10^{-7}	1.828×10^{-7}
r_1	0.80	0.80	0.73	0.70	0.73	0.73	0.80
r_2	0.10	0.9	0.17	0.15	0.19	0.18	0.13

续表

年收入 / 指标	1 万元	2 万元	2.5 万元	3 万元	4 万元	5 万元	10 万元
r_3	0.10	0.11	0.10	0.15	0.07	0.09	0.07
x_1	6.5×10^5	6.18×10^5	1.38×10^6	1.46×10^6	2.23×10^6	2.99×10^6	3.91×10^6
x_2	3037	120004	47506	52172	22721	1.07×10^5	94252
x_3	607	600	1235	1739	1507	1974	1746
x_4	138	200	100	200	100	200	103
x_5	7	10	10	20	20	10	20
x_6	1	1	5	2	2	2	5
x_7	0	0	0	0	0	0	3

问题（三）（略）.

说明：

(1) 研究该问题必须要考虑心理曲线，但心理曲线可能会有不同的形式，主要是看对问题解释是否合理，实力因子 λ 在不同地区可以取不同的值，对方案的评判结果也会有差别.

(2) 问题的合理性指标函数与心理曲线有关，但应该在风险决策的意义下确定出损益函数，损益函数的确定不是惟一的.

(3) 问题中概率公式的形式应是惟一的.

习 题 8

1. 自然界中不同种群之间存在着这样一种非常有趣的相互依存、相互制约的生存方式，种群 A 依靠丰富的自然资源生长，而种群 B 靠捕食 A 为生，鱼和鲨鱼、兔和山猫、落叶松和蚜虫都是这样生存方式的典型. 生态学上称种群 A 为食饵，种群 B 为捕食者，二者共处组成食饵—捕食系统. 数十年来许多数学家和生态学家对这一系统进行了深入的研究，建立了一系列数学模型，下面是由意大利数学家 Voltera 提出的最简单的一个模型.

食饵 A 和捕食者 B 在时刻 t 的数量分别记做 $x(t)$，$y(t)$，当 A 独立生存时它的（相对）生长率为 r，即 $\dot{x}(t)=rx$，而 B 的存在使 A 的增长率减小，设减小的程度与 B 的数量成正比，于是 $x(t)$ 满足方程

$$\dot{x}(t)=x(r-ay)=rx-axy \tag{1}$$

比例系数 a 反映捕食者掠取食饵的能力.

捕食者离开食饵无法生存，设 B 独自存在时死亡率为 d，即 $\dot{y}(t) = -dy$，A 为 B 提供食物相当于使 B 的死亡率降低，并促使其增长，设这个作用与 A 的数量成正比，于是 y(t) 满足方程

$$\dot{y}(t) = y(-d + bx) = -dy + bxy \tag{2}$$

比例系数 b 反映食饵对捕食者的供应能力.

设食饵和捕食者的初始数量分别为

$$x(0) = x_0, \quad y(0) = y_0 \tag{3}$$

微分方程 (1)、微分方程 (2) 及初始条件 (3) 确定了食饵和捕食者数量 x(t),y(t) 的演变过程,但是该方程无解析解. 试用数值解讨论以下问题：

1. 设 $r = 1$，$d = 0.5$，$a = 0.1$，$b = 0.02$，$x_0 = 25$，$y_0 = 2$，求微分方程 (1)、微分方程 (2) 在条件 (3) 下的数值解. 并绘出函数 x(t),y(t) 的图形.

2. 描述振荡器的经典范德波尔微分方程为

$$\frac{d^2x}{dt^2} - (1 - x^2)\frac{dx}{dt} + x = 0$$

试在 Matlab 中用数值解法求该微分方程的解.

3. 研究种群竞争模型. 设有甲、乙两个种群，当它们独自生存时数量演变服从 Logistic 规律，即

$$\dot{x}(t) = r_1 x\left(1 - \frac{x}{n_1}\right), \quad \dot{y}(t) = r_2 y\left(1 - \frac{y}{n_2}\right), \tag{4}$$

其中 x(t),y(t) 分别为甲、乙种群的数量，r_1，r_2 为它们的固有增长率，n_1，n_2 为它们的最大容量.

当两个种群在同一环境中生存时，它们之间的一种关系是为了争夺同一资源而进行竞争. 考查由于乙消耗有限的资源对甲的增长产生的影响，可以合理地将种群甲的方程修改为

$$\dot{x}(t) = r_1 x\left(1 - \frac{x}{n_1} - s_1\frac{y}{n_2}\right) \tag{5}$$

这里 s_1 的含义是，对于供养甲的资源而言,单位数量乙 (相对 n_2) 的消耗为单位数量甲 (相对 n_1) 消耗的 s_1 倍. 类似地,甲的存在也影响了乙的增长,乙的方程应改写为

$$\dot{y}(t) = r_2 y\left(1 - s_2\frac{x}{n_1} - \frac{y}{n_2}\right) \tag{6}$$

对 s_2 可作相应的解释. 当给定种群的初始值

$$x(0) = x_0, y(0) = y_0 \tag{7}$$

及参数 r_1，r_2，s_1，s_2，n_1，n_2 后，式 (4) ~式 (7) 确定了两个种群数量的变化规律. 方程 (6)、(7) 无解析解，一般用数值解法研究该问题，试用数

值解法研究下面的问题：

设 $r_1 = r_2 = 1$，$n_1 = n_2 = 100$，$s_1 = 0.5$，$s_2 = 2$，$x_0 = y_0 = 10$，计算 $x(t)$，$y(t)$ 并绘出它们的图形，求出时间 t 充分大以后 $x(t)$，$y(t)$ 的变化趋势.

4. 一条长为 20m 的链条悬挂在一个钉子上，初始时，一边 12m，另一边 8m，由静止启动，分别在以下情况下计算链条下滑的时间.

(1) 不计摩擦力和空气阻力.

(2) 阻力为 1m 的链条重.

(3) 阻力与速度成正比.

5. 一敌舰在某海域沿正北方向航行时，我方战舰恰位于敌舰的正西方向 1 海里处. 我舰向敌舰发射制导鱼雷，敌舰速度为 0.42 海里/分，鱼雷速度为敌舰速度的两倍. 试问敌舰航行多远时将被击中？

第八章　概率统计模型在 Matlab 中的求解方法

许多实际问题往往需要对大量的数据进行分析，其中统计分析更为重要．如在各种实验过程中获取的观察数据以及对数据的分析整理、统计预报中的预测、经验公式中的参数确定等，常常要用到各种统计方法．

§8.1　保险储备策略问题

某企业每年耗用某种材料 3 650 件，每日平均耗用 10 件，材料单价 10 元，一次订购费每件 25 元，每件年储存费 2 元，每件缺货一次费用 4 元，平均交货期 10 天，交货期内不同耗用量 X 的概率分布如表 8.1 所示．

表 8.1

X_i	80	85	90	95	100	105	110	115	120	125	130
P_i	0.01	0.02	0.05	0.15	0.25	0.20	0.15	0.10	0.04	0.02	0.01

试求使平均费用达到最小的订货量、订购次数及含有保险储量的最佳订货点．

8.1.1　问题分析及模型的建立

在企业的经济活动中，有一个保险储备的概念，保险储备是指：企业在经济活动中，按照某一经济订货批量，在订货点发出订货单后，如果需求增大或送货延迟，就会发生缺货或供货中断．为防止由此造成的损失，需要多储备一些存货以备应急之需，称为保险储备．这些存货在正常情况下不动用，只有当存货过量使用或送货延迟时才动用．

对上述问题作以下假设：

C_1——订购费（元/次）；　　C_2——储存费（元/件×天）；

C_3——缺货费（元/次×件）；　　U——单价（元/件）；

D——年需求量；　　R——日平均需求量；

T——订货周期；　　Q——订购量；

N——订购次数；　　S——订货点；

L——平均送货期；　　B——保险储备量.

下面分两个步骤进行讨论：

1. 求最佳订货量及订货次数

由于日需求量 R 为已知常数，则可以假定为确定性不允许缺货模型，货物订货量

$$Q = RT \tag{8.1}$$

均匀下降，当降到零时订货即可到达，目标函数为每天的平均费用.

记任意时刻 t 的库存量为 q(t).

因为在 $0 \leqslant t < T$ 间无订货，所以对于足够小的 t 有

$$q(t + \Delta t) = q(t) - R\Delta t, \quad 0 \leqslant t < T$$

即 $q(t) = -R$，又 $q(0) = Q$，故 $q(t) = Q = at$. 即

$$q(t) = RT - at, \quad 0 \leqslant t < T$$

由式(8.1)可得一周期内的储存量为

$$\int_0^t q(t)\,dt = \frac{1}{2} - RT^2$$

于是每天的储存量为

$$C_2 \frac{\frac{1}{2}RT^2}{T} = \frac{1}{2}C_2RT$$

每天的订货费为

$$\frac{C_1 + URT}{T} = \frac{C_1}{T} + RT$$

每天的平均费用为

$$C(T) = \frac{C_1}{T} + UR + \frac{1}{2}C_2RT$$

就是最佳订货量 Q^* 及订货次数，归结为求订货周期 T 使 C(T) 最小.

2. 求最佳订货点和保险储备量

考虑送货期需求量的随机性，订货点 S 除满足送货期 L 的平均需求外，还需维持保险储备量 B，则

$$S = LR + B$$

当库存量降到 S 时应订货. 订货期内发生缺货则采取缺货不供应处理

方式. 令

$$Y_i = \begin{cases} X_i - S, & X_i > S \\ 0, & X_i \leqslant S \end{cases}$$

则送货期内需求量增加而引起的平均缺货量为

$$E(Y) = \sum_{i=1}^{11} Y_i P_i$$

因此得年度缺货费为 $N^* C_3 E(Y)$，保险储备费为 $C_2 B$. 与保险储备有关的总费用为

$$T = N^* C_3 E(Y) + C_2 (S - LR)$$

欲求 T 最小，即由 $\min \{N^* C_3 E(Y) + C_2 (S - LR)\}$ 确定 S^*，B^*.

模型的建立：

(1) 定义：$C_1 = 25$ 元/次，$C_2 = 2$ 元/年 $= 2/365$ 天，$C_3 = 4$ 元/件 × 次 $U = 10$ 元/件，$D = 3\ 650$ 件/年，$R = 10$ 件/天，$L = 10$ 天.

(2) 定义：$Q = RT$

$$C(T) = \frac{C_1}{T} + UR + \frac{1}{2} C_2 RT$$

利用微分法求极值，令 $\frac{dC}{dT} = 0$ 求得

$$T^* = \sqrt{\frac{2C_1}{RC_2}}$$

代入 $Q = RT$ 得

$$Q^* = \sqrt{\frac{2C_1 R}{C_2}}, \quad N^* = \frac{D}{Q^*}$$

(3) 定义：

$$S = LR + B$$

$$Y_i = \begin{cases} X_i - S, & X_i > S \\ 0, & X_i \leqslant S \end{cases}$$

$$E(Y) = \sum_{i=1}^{11} Y_i P_i$$

用枚举法求 S^*，使

$$\min \{N^* C_3 EL(Y) + C_2 (S - LR)\}$$

再由 $B^* = S^* - LR$ 确定 B^*，最后求得的 Q^*，N^*，S^*，B^* 就是所要求的

的结果.

8.1.2 求解模型所需的知识点及其在 Matlab 中的实现方法

(1) 求向量 (或矩阵) 的最小值 min:

命令形式:min (X)

功能:如果 X 是向量,则返回 X 中值最小的元素;如果 X 是矩阵,则返回 X 每一列的最小值.

8.1.3 模型求解在 Matlab 中的实现

模型在 Matlab 中的实现:

在 Matlab 中编辑 M 函数文件:BX.m,程序代码如下:

```
h = 10 ;  n = 12 ;  g = 4;  l = 10;  d = 10 ;
B = 0 : 5 : 30
E = 1 : 7 ;
C = 1 : 7 ;
H = 1 : 7 ;
T = 1 : 7 ;
X = 80: 5 : 130
Q = 1 : 11 ;
P = [0.01,0.02,0.05,0.15,0.25,0.2,0.15,0.10,0.04,0.02,0.01] ;
For i = 1: 7
    S = l * d + B (i);
For j = 1: 11
    If X (j) > s
        Q (j) = X (j) - s ;
    Else
        Q (j) = 0 ;
    End
End
   Q ;
E (i) = Q * p';
C (i) = n * g * E (i);
H (i) = h * B (i);
T (i) = C (i) + H (i) ;
```

```
End
E ，C，H，T
Mint = min （T'）
```

其运算结果为：

```
E = （5. 6000    3. 0000    1. 40000    0. 55000    0. 2000    0. 050    0）
C = 268. 800    144. 000    67. 200    26. 400    9. 600    2. 4000    0
H = 0    50    100    150    200    250    300
T = 26. 800    194. 000    167. 200    176. 400    209. 600    252. 400
    300. 00
minT = 167. 2000
B* = 10
S* = 10.
```

结果分析：

由上述计算结果可以看出：(1) 不采用储存策略，缺货费用较多；(2) 保存较多的库存量，储备费用较多；(3) 建立合理的保险储备量，则企业的年度平均费用最少.

§8. 2　回归分析——火柴消费与各因素之间的关系分析

火柴公司的火柴销售量与各方面因素都有很大的联系，根据往年的销售情况，收集到了以下的一些数据，如表 8. 2 所示.

表 8. 2　火柴销售与各影响因素表

年份	火柴销售量 /（万件）	煤气、液化气用户 /（万户）	卷烟销售量 /（万箱）	蚊香销售量 /（十万盒）	打火石销售量 /（万粒）
1971	17. 84	27. 43	21. 43	11. 09	25. 78
1972	18. 27	29. 95	24. 96	14. 48	28. 16
1973	20. 29	33. 53	28. 37	16. 97	24. 26
1974	22. 61	37. 31	42. 57	20. 16	30. 18
1975	26. 71	41. 16	45. 16	26. 39	17. 08
1976	31. 19	45. 73	52. 46	27. 04	7. 39
1977	30. 5	50. 59	45. 3	23. 08	3. 88

续表

年份	火柴销售量 /（万件）	煤气、液化气用户 /（万户）	卷烟销售量 /（万箱）	蚊香销售量 /（十万盒）	打火石销售量 /（万粒）
1978	29.63	58.82	46.8	24.46	10.53
1979	29.69	65.28	51.11	33.82	20.09
1980	29.25	71.25	53.29	33.57	21.22
1981	31.05	73.37	55.36	39.59	12.63
1982	32.28	76.68	54.0	48.49	11.17

试分析火柴的销售量和各个因素之间的关系.

8.2.1 问题分析及模型的建立

首先对该问题作以下假设：

1. 火柴销售量为研究指标 Y，煤气、液化气户数、卷烟销售量、蚊香销售量、打火石销售量分别为自变量 x_1、x_2、x_3、x_4.

2. Y 与自变量 x_1、x_2、x_3、x_4 成线性函数关系.

3. Y 是随机变量，服从均值为零的正态分布，所以可以建立多元线性回归模型

$$\begin{cases} y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + \varepsilon \\ \varepsilon \sim N(0,\sigma^2) \end{cases}$$

8.2.2 建模所需的知识点及其在 Matlab 中的实现方法

1. 求解多元线性回归 regress（）

命令形式 1： b = regress（Y，X）

命令形式 2： [b，bint，r，rint，stats] = regress（Y，X，alpha）

其中因变量数据向量 Y 和自变量数据矩阵 X 按以下排列方式输入

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

而 alpha 为显著性水平（缺省时设定为 0.05）.

功能：b 为回归系数的估计值 β_0，β_1，…，β_m，bint 为回归系数估计值 b 的置信区间；r，rint 为残差向量及其置信区间；stats 是用于检验回归模型的统计量，第一个是 R^2，其中 R 是相关系数，第二个是 F 统计量，第三个是与统计量 F 相对应的概率 p，当 $p < \alpha$ 时拒绝 H_0，说明回归模型假设（即 Y 与 X 的关系）成立.

2. **求矩阵尺寸** size（）

命令形式：[m，n] = size（X）

功能：返回矩阵 X 的行数 m 和列数 n.

8.2.3　模型求解在 Matlab 中的实现

在 Matlab **中求解该模型的程序代码如下：**

```
% 按表 8.2 输入原始数据
x1 = [17.84   27.43   21.43   11.09   25.78
      18.27   29.95   24.96   14.48   28.16
      20.29   33.53   28.37   16.97   24.26
      22.61   37.31   42.57   20.16   30.18
      26.71   41.16   45.16   26.39   17.08
      31.19   45.73   52.46   27.04   7.39
      30.5    50.59   45.3    23.08   3.88
      29.63   58.82   46.8    24.46   10.53
      29.69   65.28   51.11   33.82   20.09
      29.25   71.25   53.29   33.57   21.22
      31.05   73.37   55.36   39.59   12.63
      32.28 ,  76.68 , 54.0 ,  48.49 , 11.17];
X = [ones (size (x1 (:, 1))), x1 (:, 2:5)] ;
Y = x1 (:, 1);
[b, bint, r, rint, stats] = regress (Y, X, 0.05)
b, bint, stas
```

其运算结果为：

```
b = 17.0557   0.0507   0.2606   - 0.0057   - 0.2367
bint = 14.4594   19.6521
       - 0.0089   0.1104
        0.1905    0.3307
       - 0.1037   0.0924
```

$-0.2922 \quad -0.1812$

stats = 0.9940 291.9381 0.0000

从上述计算结果可知；

回归方程

$$Y = 17.0557 + 0.0507 * x_1 + 0.2606\ x_2 - 0.0057x_3 - 0.2367 * x_4$$

方差估计 $\sigma^2 = \dfrac{Q}{n-2} = 0.18828$

其中 $Q = r' * r = 1.8828$**，相关系数的平方** $R^2 = 0.9940$**，回归方程的显著性检验 F 统计量**

$$F = \frac{U/m}{Q/(n-m+2)} = 291.938$$

注意　若出现某一项因素 x_i **的系数过小，则可以删除这个因子，重新计算回归方程.**

§8.3　回归分析——商品销量与价格的关系

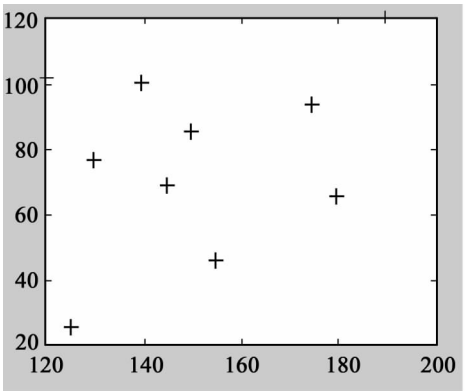
某厂生产的一种电器的销量 y 与竞争对手的价格 x_1 **和本厂的价格** x_2 **有关，表 8.3 是该商品在 10 个城市的销售记录，试根据这些数据建立 y 与** x_1 **和** x_2 **的关系式，对得到的模型和系数进行检验. 若某市本厂产品销价 160 元，竞争对手销价 170 元，预测商品在该市的销量.**

表 8.3　商品销量 y 与价格 x_1 **和** x_2

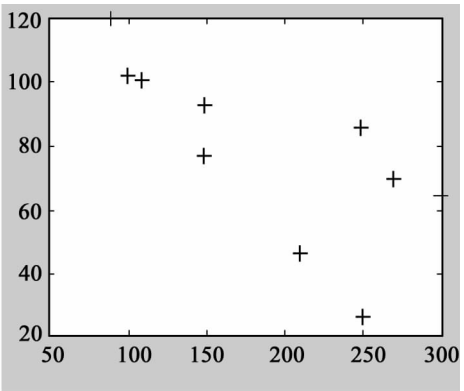
x_1/**元**	120	140	190	130	155	175	125	145	180	150
x_2/**元**	100	110	90	150	210	150	250	270	300	250
y/**个**	102	100	120	77	46	93	26	69	65	85

8.3.1　问题分析及模型的建立

将 (x_1, y) **和** (x_2, y) **各 10 个点绘在图 8.1(a) 和图 8.1(b) 中，可以看出 y 与** x_2 **有比较明显的线性关系，而 y 与** x_1 **之间的关系则难以确定，用回归分析进行比较分析.**



(a)　x_1与y的关系图



(b)　x_2与y的关系图

图 8.1

回归分析中最简单的形式是

$$y = \beta_0 + \beta_1 x$$

x，y 均为标量，β_0，β_1 为回归系数，称为一元回归．该回归的一个推广就是 x 为多元变量，形如

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m \tag{8.2}$$

其中 $m \geqslant 2$，更一般的形式是

$$y = \beta_0 + \beta_1 f_1(x) + \cdots + \beta_m f_m(x) \tag{8.3}$$

其中 $x = (x_1, x_2, \cdots, x_m)$，$f_j$ $(j = 1, 2, \cdots, m)$ 是已知函数．这里 y 对回归系数

$$\beta = (\beta_0, \beta_1, \cdots, \beta_m)$$

是线性的，称为多元线性回归．显然，可以将式 (8.3) 化为式 (8.2)，所以一般以式 (8.2) 为多元线性回归的标准形．

在回归分析中自变量 $x = (x_1, x_2, \cdots, x_m)$ 是影响变量 y 的主要因素，是人能控制和观察的，且 y 还受到随机因素的干扰，可以合理地假设这种干扰服从于零均值的正态分布，所以模型可以记做

$$\begin{cases} y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m \\ \varepsilon \sim N(0, \sigma^2) \end{cases} \tag{8.4}$$

其中 σ 未知，现得到 n 个独立观察数据 $(y_i, x_{i1}, x_{i2}, \cdots, x_{im})$，$i = 1, 2, \cdots, n$，$n > m$.

由式 (8.4) 得

$$\begin{cases} y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im} + \varepsilon_i \\ \varepsilon_i \sim N(0, \sigma^2), i = 1, 2, \cdots, n \end{cases} \tag{8.5}$$

记

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \vdots & \\ 1 & x_{n1} & \cdots & x_{nm} \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}, \beta = (\beta_0, \beta_1, \cdots, \beta_m)^T$$

则式（8.5）用矩阵的形式表示即为

$$\begin{cases} Y = X\beta + \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} \sim N(0, \sigma^2) \end{cases}$$

由上述分析可知，对于商品销售量与价格问题，其回归模型为

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2.$$

8.3.2 求解模型所需的知识点及其在 Matlab 中的实现方法

求解多元线性回归 regress（）：

命令形式 1： b = regress（Y，X）

命令形式 2： [b，bint，r，rint，stats] = regress（Y，X，alpha）

8.3.3 模型求解在 Matlab 中的实现

求解该模型在 Matlab 中的命令为：

```
x1 = [120 140 190 130 155 175 125 145 180 150];
x2 = [100 110 90 150 210 150 250 270 300 250];
y = [102 100 120 77 46 93 26 69 65 85];
x = [ones (10, 1) x1' x2'];
[b, bint, r, rint, stats] = regress (y, x)
```

其运行结果为

```
b = 66.5176
    0.4139
   -0.2698
bint = -32.5060    165.5411
       -0.2018      1.0296
       -0.4611     -0.0785
r = 12.7907                    rint = -21.7003    47.2816
    5.2102                           -36.4308    46.8511
```

- 0. 8812	- 31. 3095	29. 5470
- 2. 8594	- 45. 3733	39. 6544
- 28. 0205	- 64. 1141	8. 0731
- 5. 4856	- 46. 7678	35. 7966
- 24. 8118	- 56. 4726	6. 8490
15. 3055	- 23. 8482	54. 4592
4. 9119	- 29. 3625	39. 1863
23. 8403	- 13. 1180	60. 7986

stats = 0. 6527　6. 5786　0. 0247.

§8. 4　单因素方差分析——广告宣传对产品销量的影响分析

某公司为了研究三种内容的广告宣传对某种洗衣机销售量的影响，进行了统计调查，经过广泛宣传后，按寄回的广告上的订购数计算，一年四个季度的销售量统计如表 8. 4 所示 .

表 8. 4　广告与销售量

季　度	广告类型		
	A_1	A_2	A_3
1	163	184	206
2	176	198	191
3	170	179	218
4	185	190	224

其中 A_1 是强调运输方便性的广告；A_2 是强调节省能源的经济性广告；A_3 是强调噪音低的优良性广告 .

试问哪一种类型的广告促进洗衣机销量增加所起的宣传效果最好？

8. 4. 1　问题分析及模型的建立

单因素方差分析：

若只考虑一个因素对实验指标的影响，而且用方差进行分析，这种方法被称为单因素方差分析 . 方差分析的主要目的是通过实验数据分析推断因素 A 对实验指标影响是否显著，即当因素 A 取不同水平的实验指标有无显著差异 .

假设检验的因子有 m 种水平，X_1，X_2，…，X_m 是 m 个相互独立的正态总体，分别服从于 $N(\mu,\sigma^2)$，$i=1,2,\cdots,m$. 另外，x_{ij}（$i=1$，2，…，m；$j=1$，2，…，n）是抽得的分别服从于正态分布的简单随机样本．则单因素方差分析模型为

$$x_{ij}=\mu_i+\varepsilon_{ij}$$
$$\varepsilon_{ij}\sim N(0,\sigma^2)$$

对于上述模型中所提出的多个正态总体均值是否相等的问题，也就是假设检验

$$H_0:\mu_1=\mu_2=\cdots=\mu_m$$

定义

$$n=\sum_{i=1}^{m}n_i,\bar{x}_i=\frac{1}{n_i}\sum_{j=1}^{n_i}x_{ij},\ \bar{x}=\frac{1}{n}\sum_{i=1}^{m}n_i\bar{x}_i$$

则有平方和分解公式

$$\begin{aligned}Q&=\sum_{i=1}^{m}\sum_{j=1}^{n_i}(x_{ij}-\bar{x})^2\\&=\sum_{i=1}^{m}\sum_{j=1}^{n_i}(x_{ij}-\bar{x}_i)^2+\sum_{i=1}^{m}(x_i-\bar{x})^2\\&=Q_1+Q_2\end{aligned}$$

其中，Q_1 被称为内离差平方和，Q_1 是反映数据 x_{ij} 在抽样过程中产生总的程度的一个评价指标．Q_2 是各组平均值与总平均值的离差平方和，反映了各总体的样本平均值之间的差异程度，被称为组间平方和．通过 Q_2 取值的大小可以反映原假设 H_0 是否成立．

还可以构造 F 统计量

$$F=\frac{Q_2/(m-1)}{Q_1/(n-m)}\sim F(m-1,n-m)$$

给定显著性水平 α，当 $F>F_\alpha(m-1,n-m)$ 时，则拒绝 H_0. 对于单因素方差分析，其方差分析表如表 8.5 所示．

表 8.5　　方差分析表

方差来源	平方和	自由度	方　差	F 值
因　子	Q_2	m - 1	$s_2^2=Q_2/(m-1)$	$\frac{s_2^2}{s_1^2}$

续表

方差来源	平方和	自由度	方　差	F 值
误　差	Q_1	n - m	$s_1^2 = Q_1 / (n - m)$	
总　和	Q	n - 1	$s_2^2 = Q / (n - 1)$	

方差分析一般用的显著性水平是：取 $\alpha = 0.01$，拒绝 H_0，称因素 A 的影响非常显著；取 $\alpha = 0.01$，不拒绝 H_0，但取 $\alpha = 0.05$，拒绝 H_0，称因素 A 的影响显著；取 $\alpha = 0.05$，不拒绝 H_0，称因素 A 无显著影响.

洗衣机销售是一个试验指标，新闻广告是影响试验指标的一个因素 A，三种不同广告内容可以看做三种不同状态，称为水平，现记做 A_1，A_2，A_3，该试验是一个单因子三水平的试验. 对于该试验，要从以下几个方面来分析：

1. 虽然用同一种广告，但在同一年里不同季度的销售量不同，分析原因可以认为是由于其他随机原因造成.

2. 不同的新闻广告引起洗衣机销售量的不同，可能是广告内容的不同所致，也可能是其他随机因素所致.

3. 由于存在其他随机因素，为了便于简化与可操作性，假设这些随机因素对洗衣机销售量的影响是次要的，并且假设三种广告类型为三种不同的总体，由于经常遇到的是正态总体，因此，假定它们分别是方差相同的正态总体.

所以该问题就是一个单因素方差分析模型.

8.4.2　求解模型所需的知识点及其在 Matlab 中的实现方法

单因素方差分析 anoval（）：

命令形式：p = anoval（X）或 anoval（X）

其中 X 为 m × n 阶的数据矩阵.

功能：返回值 $p = P\{F > F_\alpha(m - 1, n - 1)\}$，当 $p > \alpha$ 时，表示接受 H_0. 在命令形式 anoval（X）中，除给出 p 外，还输出一个方差表和一个 Box 图.

8.4.3　模型求解在 Matlab 中的实现

在 Matlab 中求解模型的命令如下：

```
fm = [163 184 206;176 198 191;170 179 218;185 190 224];
anoval (fm)
```

其运算结果为：

ans = 0.0039 (p 值)

其方差分析表如表 8.6 所示.

表 8.6

Source	SS	df	MS	F	Prob > F
Columns	2668.17	2	1334.08	10.93	0.0039
Error	1098.5	9	122.06		
Total	3766.67	11			

方差分析图如图 8.2 所示.

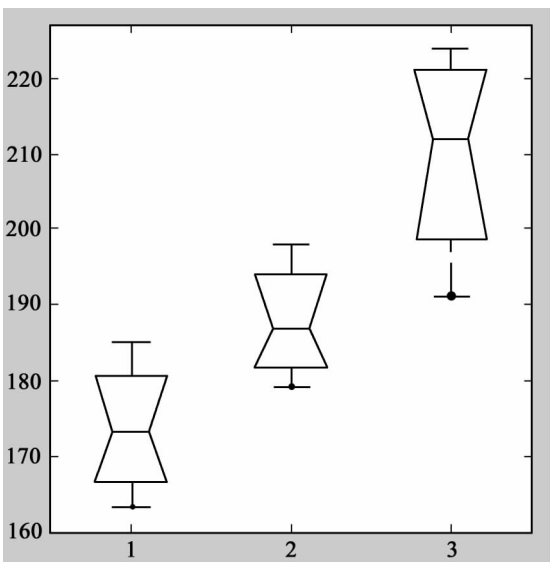


图 8.2

由计算结果知 $p = 0.0039 < \alpha = 0.05$，故拒绝 H_0. 查表可以求得 $F_{0.05}(2,9) = 4.26$，由方差分析表可知 $F = 10.93 > F_{0.05}(2.9) = 4.2$，所以拒绝 H_0，即认为不同类型的广告内容对洗衣机销售有显著影响. 进一步进行统计分析，确定哪一种广告形式最好. 定义

$$\mu = \frac{1}{m}\sum_{i=1}^{m} n_i\mu_i, \alpha_i = \mu_i - \mu, \quad i = 1,2,\cdots,m$$

α_i 称为第 i 个水平对试验结果的效应值，α_i 反映第 i 个水平因子对试验指标作

用的大小．一般情况下，点估计值 $\alpha_i = \bar{x}_i - \bar{x}$ 是 α_i 的无偏估计，可以用点估计值 $\alpha_i = \bar{x}_i - \bar{x}$ 来估计参数 a_i．根据上述计算可知，广告内容对洗衣机销量有显著影响，因此需要计算各水平效应值

$$\alpha_1 = \bar{x}_1 - \bar{x} = 173.5 - 190.33 = -16.83$$

$$\alpha_2 = \bar{x}_2 - \bar{x} = 187.75 - 190.33 = -2.58$$

$$\alpha_3 = \bar{x}_3 - \bar{x} = 209.75 - 190.33 = 19.42$$

所以效应值 α_3 最大，这说明广告 A_3 引起的洗衣机销量最多，即广告 A_3 对洗衣机销量所起的宣传效果最好．

§8.5　双因素方差分析——影响火箭射程的因素分析

影响火箭射程有两个重要因素：燃料 A 和推进器 B．在火箭的研制和发射中，研究人员很想知道这两种因素的不同水平组合对火箭射程是否有显著影响？它们的最佳水平组合是多少？在对火箭射程进行模拟的试验中，取出了四种燃料和三种推进器，每种燃料和每种推进器的组合各发射火箭一次，得到数据如表 8.7 所示．

表 8.7　火箭射程试验数据　（单位：km）

推进器（B） 燃料（A）	B_1	B_2	B_3
A_1	63.2	56.2	60.3
A_2	59.1	54.1	41.6
A_3	68.1	70.9	39.2
A_4	75.8	58.2	40.7

试分析不同燃料、不同推进器对射程的影响．

8.5.1　问题分析及模型的建立

双因素方差分析：

很多实际问题中，对试验指标的影响不只有一个因素，可能需要同时考虑几个因素对试验指标的影响．这种同时分析多个因素对试验指标的影响并且使

用方差进行问题分析的方法称为多因素方差分析，特别地，若只考虑两个因素被称为双因素方差分析.

设 A 取 r 个水平 A_1，A_2，…，A_r，B 取 s 个水平 B_1，B_2，…，B_s；又设在水平组合（A_i，B_j）下作了 n 次试验，所得结果记为 x_{ijk}，i = 1，2，…，r；j = 1，2，…，s；k = 1，2，…，n. 则双因素方差分析的数学模型为

$$x_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk}$$

$$\sum_{i=1}^{r} \alpha_i = 0, \quad \sum_{j=1}^{s} \beta_j = 0, \quad \sum_{i=1}^{r} \gamma_{ij} = \sum_{j=1}^{s} \gamma_{ij} = 0$$

$x_{ijk} \sim N(\mu_{ij}, \sigma^2)$，i = 1，2，…，r，j = 1，2，…，s，k = 1，2，…，n，并且 x_{ijk} 相互独立. 其中 μ 为总的平均值；α_i 是因子 B（列）的效应值，并且行和为零；β_j 是因子 A（行）的效应值，并且列和为零；γ_{ij} 是交互效应值（所在行与所在列的和为零）；它们的表达式为

$$\mu = \frac{1}{rs}\sum_{i=1}^{r}\sum_{j=1}^{s}\mu_{ij}, \mu_i = \frac{1}{s}\sum_{j=1}^{s}\mu_{ij}, \alpha_i = \mu_i - \mu$$

$$\mu_j = \frac{1}{r}\sum_{i=1}^{r}\mu_{ij}, \beta_j = \mu_j - \mu, \gamma_{ij} = \mu_{ij} - \mu - \alpha_i - \beta_j$$

原假设为

$$H_{01}: \alpha_1 = \alpha_2 = \cdots = \alpha_r = 0,$$

$$H_{02}: \beta_1 = \beta_2 = \cdots = \beta_r = 0$$

$$H_{03}: \gamma_{11} = \gamma_{12} = \cdots = \gamma_{rs} = 0$$

双因素方差分析的方差分析表如表 8.8 所示.

表 8.8　　双因素方差分析表

方差来源	平方和	自由度	均　方	F 值
因子 A	Q_1	r - 1	$S_A = Q_1/(r-1)$	$F_A = S_A/S_E$
因子 B	Q_2	s - 1	$S_B = Q_2/(s-1)$	$F_B = S_B/S_E$
交互作用	Q_3	(r - 1)(s - 1)	$S_{A\times B} = Q_3/(r-1)(s-1)$	$F_{A\times B} = S_{A\times B}/S_E$
误差	Q_4	rs(n - 1)	$S_E = Q_4/rs(n-1)$	
总和	Q	rsn - 1		

由上述分析可知，这是一个双因素方差问题，可以用双因素方差分析对这个问题作出推断.

8.5.2　建模所需的知识点及其在 Matlab 中的实现方法

双因素方差分析 anova2 （）：

命令形式：p = anova2 (X)

其中 X 为 m × n 矩阵

功能：返回值 p = [p_1，p_2]，其中 p_1，p_2 分别为因素表 X 中因素 A（即表 X 中的列）和因素 B（即表 X 中的行）的概率；若 $p_1 > a = 0.05$，表示因素 A 无显著影响；若 $p_1 < a = 0.05$，则表示因素 A 有显著影响．同样，若 $p_2 > a = 0.05$，表示因素 B 无显著影响；若 $p_2 < a = 0.05$，则表示因素 B 有显著影响．

8.5.3　模型求解在 Matlab 中的实现

求解该模型的 Matlab 的程序代码如下：

```
% 数据输入：将实验数据以 s × r 的形式输入
fm = [63.2   56.2   60.3;
      59.1   54.1   41.6;
      68.1   70.9   39.2;
      75.8   58.2   40.7];
p = anova2 (fm)
```

其运算结果为：

```
p = 0.0410    0.6687
```

方差分析表如表 8.9 所示：

表 8.9

Source	SS	df	MS	F	Prob > F
Columns	929.95	2	464.973	5.7	0.041
Rows	133.59	3	44.53	0.55	0.6687
Error	489.28	6	81.547		
Total	1552.82	11			

p 的两个值分别为因素 B（行）、因素 A（列）的概率 P_r，因为 $0.6687 > \alpha = 0.05$，所以因素 A（燃料）无显著影响，而 $0.0410 < \alpha = 0.05$，所以因素 B（推进器）有显著影响．

§8.6 真题解析：车灯线光源的优化设计

8.6.1 问题的提出：车灯线光源的优化设计

安装在汽车头部的车灯的形状为一旋转抛物面，车灯的对称轴水平地指向正前方，其开口半径为36mm，深度为21.6mm. 经过车灯的焦点，在与对称轴相垂直的水平方向，对称地放置一定长度的均匀分布的线光源. 要求在某一设计规范标准下确定线光源的长度.

该设计规范经简化后可以描述为：在焦点F正前方25m处的A点放置一测试屏，屏与FA垂直，用以测试车灯的反射光. 在屏上过A点引出一条与地面相平行的直线，在该直线A点的同侧取B点和C点，使AC = 2AB = 2.6m. 要求C点的光强度不小于某一额定值（可以取为1个单位），B点的光强度不小于该额定值的两倍（只须考虑一次反射）.

试解决下列问题：

(1) 在满足该设计规范的条件下，计算线光源长度，使线光源的功率最小.

(2) 对得到的线光源长度，在有标尺的坐标系中绘出测试屏上反射光的亮区.

(3) 讨论该设计规范的合理性.

8.6.2 模型的建立

建立坐标系如图8.3所示，记线光源长度为l，功率为W，B、C点的光强度分别为$h_B(l)$ W和$h_C(l)$ W，先求$h_B(l)$和$h_C(l)$的表达式，再建立整个问题的数学模型.

以下均以毫米为单位，由所给信息不难求出车灯反射面方程为$z=\frac{x^2+y^2}{60}$，焦点坐标为（0，0，15）.

1. 位于点P(0，w，15)的单位能量的点光源反射到点C(0，2600，25015)的能量

设反射点的坐标为$Q\left(x, y, \frac{x^2+y^2}{60}\right)$. 记入射向量为a，该点反射面外法线方向为b，不难得到反射向量c满足

$$c = a - \frac{2a \cdot b}{|b|^2}b.$$

记 $r^2 = x^2 + y^2$，由

$$a = \left(x, y - w, \frac{r^2}{60} - 15\right),$$

$$b = \left(\frac{x}{30}, \frac{y}{30}, -1\right).$$

从而得 $c = (c_x, c_y, c_z)$ 的表达式

$$c_x = \frac{2xyw}{r^2 + 900}$$

$$c_y = \frac{w(2y^2 - r^2 - 900)}{r^2 + 900}$$

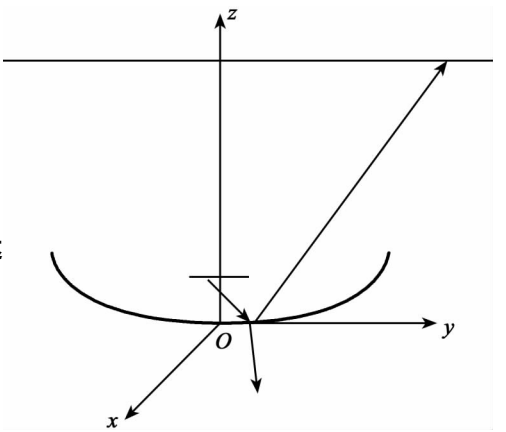


图 8.3

$$c_z = \frac{r^4 + 1\,800r^2 - 3\,600wy + 810\,000}{60(r^2 + 900)}$$

注意到反射光通过 C 点，应有

$$kc_x = -x,$$

$$kc_y = 2\,600 - y,$$

$$kc_z = 25\,015 - \frac{r^2}{60}$$

其中 k 为常数．从上述第一式可以解得 $x = 0$ 或 $k = -\frac{r^2 + 900}{2wy}$. 由此得反射点坐标满足以下两组方程

$$\begin{cases} x = 0 \\ y^5 - (w + 2600)y^4 + 1800y^3 + (1498200w - 4680000)y^2 + (9360000w + \\ \quad 810000)y - 1350810000w - 2106000000 = 0. \end{cases}$$

$$\begin{cases} y = \dfrac{3750w}{13(w - 2600)} \\ x = \pm\sqrt{5200y - 900 - y^2} \end{cases}$$

通过计算可知，存在 $w_0^C \approx -1.56$，当 $w > w_0^C$ 时，第一组方程不存在满足 $r^2 \leqslant 36^2$ 的实根，即无反射点．而当 $w < w_0^C$ 时，有两个反射点 $Q_i\left(0, y_i, \frac{y_i^2}{60}\right)$，$i = 1, 2$.

而第二组方程仅当 $-3.8119 < w < -1.5609$ 时存在满足 $r^2 \leqslant 36^2$ 的一对实

根，即有两个反射点$\left(\pm x, y, \frac{x^2+y^2}{60}\right)$，记为 Q_3，Q_4.

若反射点的坐标为 Q（x，y，z），则位于点 P（0，w，15）的单位能量点光源经 Q 点反射到 C 点的能量密度（单位面积的能量，正比于光强度）为

$$L = \frac{\cos\beta}{4\pi\,\overline{PQ}^2}$$

其中

$$\overline{PQ} = \sqrt{x^2 + (y-w)^2 + \left(\frac{r^2}{60} - 15\right)^2}$$

而 β 为反射向量与 z 轴的夹角

$$\cos\beta = \frac{25015 - \frac{r^2}{60}}{\overline{QC}}.$$

2. $h_B(l)$，$h_C(l)$ 的表达式

长 l 的具有单位能量的线光源位于点 P（0，w，15）的长 dw 的微小线光源段反射到 C 点的能量密度为

$$E(w) = \sum_{i=1}^{4} f_i(w)/l$$

其中

$$f_i(w) = \begin{cases} \dfrac{\cos\beta_i}{4\pi\,\overline{PQ_i}^2}, w \in \left[-\dfrac{l_0}{2}, w_0^C\right], i = 1,2 \\ 0, w \notin [-30, w_0^C] \end{cases}$$

$$f_i(w) = \begin{cases} \dfrac{\cos\beta_i}{4\pi\,\overline{PQ_i}^2}, w \in [-3.8119, -1.5609] \\ 0, w \notin [-3.8119, -1.5609] \end{cases}, i = 3,4$$

长 l 的具有单位能量的线光源反射到 C 点的能量密度为

$$h_C(l) = \int_{-\frac{1}{2}}^{\frac{1}{2}} E(w)\,dw.$$

类似地可得 $h_B(l)$ 的表达式．相应的反射点方程为

$$\begin{cases} x = 0 \\ y^5 - (w+1300)y^4 + 1800y^3 + (1498200w - 2340000)y^2 + (4680000w + \\ \quad 810000)y - 1350810000w - 1053000000 = 0. \end{cases}$$

$$\begin{cases} y = \dfrac{7500w}{13(w-1300)} \\ x = \pm\sqrt{2600y - 900 - y^2} \end{cases}$$

相应的 $w_0^B \approx -0.78$，而第二组方程有两个反射点的范围为 $w \in [-1.906, -0.7800005]$.

3. 优化设计的数学模型

设线光源的功率为 W，则 W 反射到 B 点和 C 点的能量密度分别为 $h_B(l) \cdot W$ 和 $h_C(l) \cdot W$. 问题的数学模型为

$$\begin{cases} \min\limits_{0 \leqslant l \leqslant l_0} W \\ \text{s.t.} \quad h_B(l)\ W \geqslant 2 \\ \qquad\ \ h_C(l)\ W \geqslant 1. \end{cases}$$

8.6.3　模型的求解

$h_B(l)$，$h_C(l)$ 可以用数值积分求得. $h_B(l)$ 应具备下列性质

$$h_B(l) = \begin{cases} 0, & 0 < l < l_B = 2|w_0^B| \\ \uparrow, & l_B \leqslant l \leqslant l_B' \\ \downarrow, & l_B' < l \leqslant l_0 \end{cases}$$

其中 l_B 为起亮值，l_B' 为最大值点，l_0 为考察的最大范围，例如取为 20mm. $h_C(l)$ 也有类似的性质，且起亮值和最大值点均相应地右移. 数值求解 $h_B(l) = 2h_C(l)$，记其解为 $l = l_*$，再求出 l_B'，l_C'，不难看出 $l_B' < l_C'$ 且 l_* 落在 (l_B', l_C') 之中.

令 $w_* = 1/h_C(l_*) = 2/h_B(l_*)$，现证 w_* 为问题之最优值. 事实上，对可行域中任一 (l, w)：

当 $l \geqslant l_*$ 时，有 $w \geqslant 2H/h_B(l) > 2H/h_B(l_*) = w_*$，（用到 $l \geqslant l_*$ 时 $h_B \downarrow$）.

当 $l < l_*$ 时，有 $w \geqslant H/h_C(l) > H/h_C(l_*) = w_*$，（用到 $l < l_*$ 时 $h_C \uparrow$）.

这就证明了 w_* 的确是最小值.

事实上数值结果为 $l_B' \approx 3.16$，$l_C' \approx 6.22$，$l_* \approx 3.62$.

8.6.4　反射光亮区的计算

分别将线光源和车灯反射面离散化为点光源和面元的集合，计算每一点光源关于每一车灯反射面元的反射光线，判断其是否与车灯反射面相交，若相交，一次反射光不能到达测试屏，否则求出该反射光线与反射屏平面的交点，即为反射亮点. 所有这些亮点的集合即为反射光亮区. 亮区的上半部分如图

8.4 所示（横坐标为 x 轴，纵坐标为 y 轴，单位为 mm），下半部分与上半部分是关于 x 轴对称的.

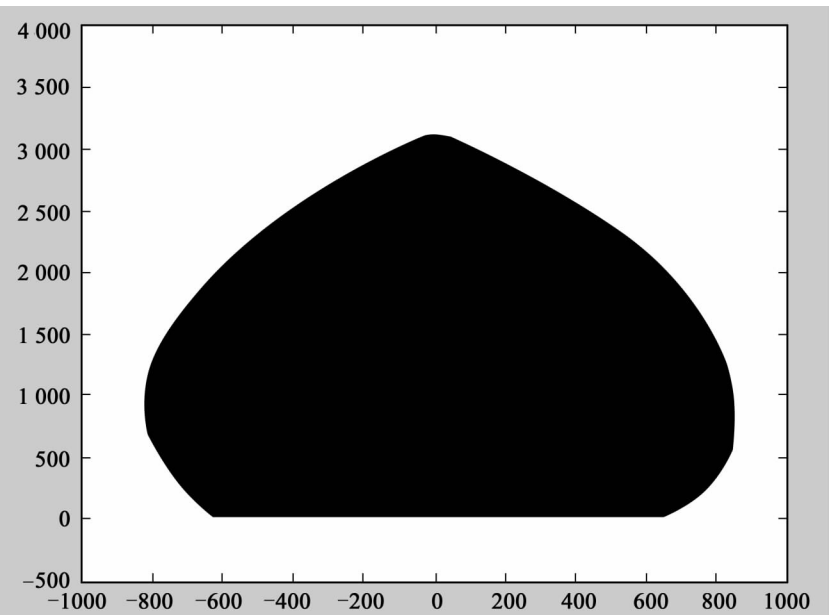


图 8.4

8.6.5 注记

1. 计算 $h_B(l)$，$h_C(l)$ 的另一种方法是建立问题的数值模型用数值模拟的方法加以解决. 具体的做法是：在得到反射光线和反射到测试屏上能量的数学模型后，分别将线光源和车灯反射面离散化为点光源和面元的集合，在测试屏 B（或 C）点附近取一微小面元. 计算每一点光源关于每一车灯反射面元的反射光线，将所有能到达该面元的反射光线的反射能量迭加起来，除以面元的面积即为 B（或 C）点的反射能量密度.

但用这样的方法必须十分注意结果的检验，注意计算精度.（必须考察线光源和反射面的部分密度和测试屏 B（或 C）点附近小面元的取法等）.

2. 以上没有考虑线光源本身对反射光线的遮挡问题，即假设线光源是透明的. 如果假设线光源是不透明的，似乎更符合现实. 此时需要考虑线光源本身对反射光线的遮挡，计算会更复杂些，计算结果也会有所不同.

习　题　9

1. 甲于 12 点 50 分从广汉车站打电话告知成都的乙，他所坐的火车大约在 13 点开出，火车从广汉到成都的运行时间均值为 30min，标准差为 2min 的随机变量，乙接到电话在 10min 后开车到火车站接甲．试分别根据相对频率求出及时接到甲的概率．

2. 为比较 5 种合成木板的耐久性，对每个品牌取 4 个样品作摩擦试验，测量磨损量，得到数据如表 8.10 所示．

表 8.10

品牌 A：	2.2	2.1	2.4	2.5
品牌 B：	2.2	2.3	2.4	2.6
品牌 C：	2.2	2.0	1.9	2.1
品牌 D：	2.4	2.7	2.6	2.7
品牌 E：	2.3	2.5	2.3	2.4

试分析：

(1) 它们的耐久性有无显著差异；

(2) 计算每种品牌的均值．

3. 表 8.11 记录了 三位工人分别在四台机器上三天的日产量，试分析工人之间的差异是否显著，机器之间的差异是否显著，二者之间的交互作用是否显著．

表 8.11

	工人甲	工人乙	工人丙
机器 A	15，15，17	19，19，16	16，18，21
机器 B	17，17，17	15，15，15	19，22，22
机器 C	15，17，16	18，17，16	18，18，18
机器 D	18，20，22	15，16，17	17，17，17

4. A、B 两地之间欲架设若干条电话线，已知 A 到 B 平均每 10s 有一次呼

叫，B 到 A 平均每 12s 有一次呼叫，通话长度平均都为 4min，设它们都服从指数分布．当线全占满时呼叫被拒绝，需重打．试通过模拟确定应架设多少条电话线，使呼叫被拒绝的比例不超过 5%．

5. 一汽车检修站的工作情况如图 8.5 所示：汽车到达的间隔为指数分布，平均 2h 一辆，检查时间服从 U(0.25，1.05)（小时），经验表明，30% 的汽车需修理，有两位修理工，修理时间均为 U(2.1，4.5)（小时），模拟 5 次，每次 160h，试求检查和修理处的平均队长、平均等待时间．

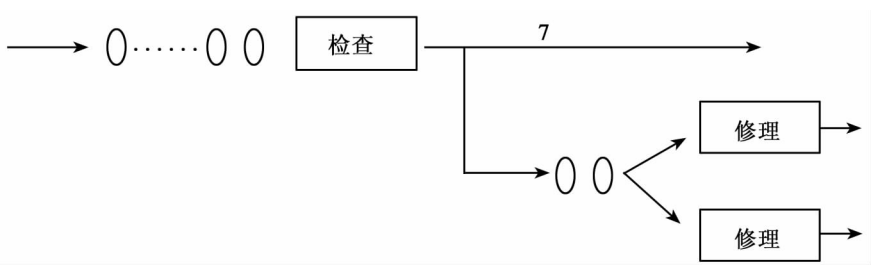


图 8.5

6. 将土质基本相同的一块地分成均等的 5 块，每块又分成均等的 4 个小区．在每一块地内，把 4 个品种的小麦分种在 4 个小区，每一小区小麦的播种量相同，测得收获量如表 8.12 所示．

表 8.12 （单位：kg）

品种＼地块	A_1	A_2	A_3	A_4	A_5
B_1	32.3	34.0	34.7	36.0	35.5
B_2	33.2	33.6	36.8	34.3	36.1
B_3	30.8	34.4	32.3	35.8	32.8
B_4	29.5	26.2	28.1	28.5	29.4

试从这组数据考查地块和品种对小麦收获量有无显著影响．

7. 电影院调查电视广告费用和报纸广告费用对每周收入的影响，得到如表 8.13 所示的数据，试建立回归模型进行检验，并判断是否有异常点．

表 8.13

每周收入	96	90	95	92	95	95	94	94
电视广告费用	1.5	2.0	1.5	2.5	3.3	2.3	4.2	2.5
报纸广告费用	5.0	2.0	4.0	2.5	3.0	3.5	2.5	3.0

8. 某炼铜厂测得铜的硬度与抗张强度的数据如表 8.14 所示，试建立回归模型，对模型和回归系数进行检验，给出硬度为 65 时的抗张强度和置信区间.

表 8.14

硬度	68	53	70	84	60	72	51	83	70	64
抗张强度	268	298	349	343	290	354	283	324	340	286

9. 为了研究合成纤维收缩率和拉伸倍数对纤维弹性的影响，进行了一些试验. 收缩率取 4 个水平：0，4，8，12；拉伸倍数也取 4 个水平：460，520，580，640，对二者的每个组合重复做两次试验，测得弹性数据如表 8.15 所示.

表 8.15

	$A_1=460$	$A_2=520$	$A_3=580$	$A_4=640$
$B_1=0$	71，73	72，73	75，73	77，75
$B_2=4$	73，75	76，74	78，77	74，74
$B_3=8$	76，73	79，77	74，75	74，73
$B_4=12$	75，73	73，72	70，71	69，69

试分析：

(1) 拉伸倍数、收缩率及其交互作用对弹性有无显著影响.

(2) 使纤维弹性达到最大的生产条件是什么.

第九章　代数模型在 Matlab 中的求解方法

§9.1　植物基因的分布

农场的植物园中，某种植物的基因型为 AA，Aa，aa，农场计划采用 AA 型植物与每种基因型植物相结合的方案培育植物后代，已知双亲体基因型与其后代基因型的概率如表 9.1 所示.

表 9.1　　父体-母体基因表

		父体—母体基因型		
		AA—AA	AA—Aa	AA—aa
后代基因类型	AA	1	1/2	0
	Aa	0	1/2	1
	aa	00	0	00

试问若干年后三种基因型分布如何？

9.1.1　问题分析及模型的建立

用 a_n，b_n，c_n 分别表示第 n 代植物中，基因型 AA，Aa，aa 的植物占植物总数的百分率（$n=1,2,\cdots$），令 $x^{(n)}$ 为第 n 代植物基因型分布：$x^{(n)}=(a_n,b_n,c_n)^T$，$n=0$ 时，$x^{(0)}=(a_0,b_0,c_0)^T$，显然初始分布有

$$a_0+b_0+c_0=1$$

由表 9.1 可得关系式

$$a_n=1,a_{n-1}+\frac{1}{2}b_{n-1}+0,\quad c_{n-1}$$

$$b_n=0,a_{n-1}+\frac{1}{2}b_{n-1}+1,\quad c_{n-1}$$

$$c_n=1,a_{n-1}+0,b_{n-1}+0,\quad c_{n-1}$$

即

$$x^{(n)}=Mx^{(n-1)}$$

其中
$$M=\begin{pmatrix}1 & \frac{1}{2} & 0\\ 0 & \frac{1}{2} & 1\\ 0 & 0 & 0\end{pmatrix}$$

从而
$$x^{(n)}=Mx^{(n-1)}=M^2x^{(n-2)}=\cdots=M^{(n)}x^{(0)}.$$

问题转化为 $M^{(n)}$，为求 $M^{(n)}$，将 M 对角化，即求可逆矩阵 P，使
$$P^{-1}MP=D$$

亦即
$$M=P^{-1}DP$$

D 为对角阵.

由于
$$\det(\lambda E-A)=\begin{vmatrix}\lambda-1 & -\frac{1}{2} & 0\\ 0 & \lambda-\frac{1}{2} & -1\\ 0 & 0 & \lambda\end{vmatrix}=\lambda(\lambda-1)\left(\lambda-\frac{1}{2}\right)$$

所以 M 的特征值为 $\lambda_1=1,\lambda_2=\frac{1}{2},\lambda_3=0.$

对于 $\lambda_1,\lambda_2,\lambda_3$ 特征向量分别可取
$$e_1=\begin{pmatrix}1\\0\\0\end{pmatrix},\quad e_2=\begin{pmatrix}1\\-1\\0\end{pmatrix},\quad e_3=\begin{pmatrix}1\\-2\\1\end{pmatrix}$$

令
$$P=(e_1,e_2,e_3)=\begin{pmatrix}1 & 1 & 1\\ 0 & -1 & -2\\ 0 & 0 & 1\end{pmatrix}$$

所以可以计算 P^{-1}. 从而
$$P^{-1}MP=D=\begin{pmatrix}1 & 0 & 0\\ 0 & \frac{1}{2} & 0\\ 0 & 0 & 0\end{pmatrix}$$

由于 $M=P^{-1}DP$，于是
$$M^n=P^{-1}D^nP$$
$$x^{(n)}=\cdots=P^{-1}D^nPx^{(0)}$$

即

$$\begin{cases} a_n = a_0 + b_0 + c_0 - \left(\frac{1}{2}\right)^n b_0 - \left(\frac{1}{2}\right)^{n-1} c_0 \\ b_n = \left(\frac{1}{2}\right)^n b_0 + \left(\frac{1}{2}\right)^{n-1} c_0 \\ c_n = 0 \end{cases}$$

容易看出,当 $n\to$ 时,$a_n\to 1$,$b_n\to 0$,$c_n = 0$. 所以在极限的情况下,培育的植物都是 AA 型.

9.1.2 求解模型所需的知识点及其在 Matlab 中的实现方法

1. 求矩阵的特征根和特征向量 eig()

命令形式 1:d = eig (A)

功能:求方阵 A 的特征根.

命令形式 2:[V, D] = eig (A)

功能:求方阵 A 的特征根矩阵 D 与特征向量矩阵 V,满足 AV = VD.

2. 对符号函数求极限 limit()

命令形式:limit (f, x, a)

功能:对符号函数 f 按变量 x 求在点 a 处的极限.

9.1.3 模型求解在 Matlab 中的实现

植物基因分布模型在 Matlab 中的求解程序:

```
M = [1, 1/2, 0;0, 1/2, 1 ;0, 0, 0];
a0 = 1/2 ;b0 = 1/3;c0 = 1/6;
x0 = [a0, b0, c0] ;
syms n ;
% 取 n = 100 直接求 xn = M^n x0 的值.
n = 100;
Xn = M n * x0'
```

其运算结果为

```
Xn = 1    0.000    0
% 利用分解后的式子 M^n = PD^nP^-1,x^(n) = … = PD^nP^-1 x^(0) 求解:
[P, D] = eig (M) ;
P, D
Dn = D n ;
Xn = P * Dn * inv (P) * x0 ;
```

X_n

其运算结果为

```
P =  1.0000      - 0.7071      0.4082
          0       0.7071     - 0.8165
          0            0      0.4082
D =  1.0000            0           0
          0       0.5000           0
          0            0           0
Xn =  1     0.000        0.
```

§9.2　城市交通流量问题

图 9.1 给出了某城市部分单行街道的交通流量（每小时过车数）.

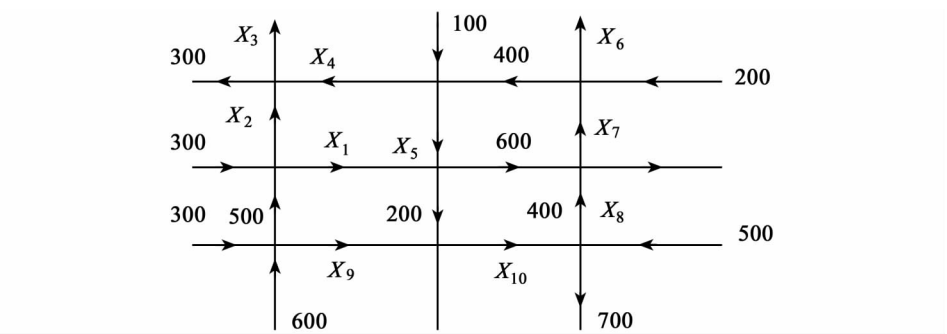


图 9.1　交通流量图

假设：(1) 全部流入网络的流量等于全部流出网络的流量.

(2) 全部流入一个节点的流量等于全部流出该节点的流量.

试建立数学模型确定该交通网络未知部分的具体流量.

9.2.1　问题分析及模型的建立

由网络流量假设，所给问题满足如下线性方程组

$$\begin{cases} x_2 - x_3 + x_4 = 300 \\ x_4 + x_5 = 500 \\ x_7 - x_6 = 200 \\ x_1 + x_2 = 800 \\ x_1 + x_5 = 800 \\ x_7 + x_8 = 1000 \\ x_9 = 400 \\ x_{10} - x_9 = 200 \\ x_{10} = 600 \\ x_8 + x_3 + x_6 = 1000 \end{cases}$$

9.2.2 求解模型所需的知识点及其在 Matlab 中的实现

1. 求矩阵的秩 rank ()

命令形式：rank (a)

功能：返回矩阵 a 的秩.

2. 通过行变换把矩阵化为上三角阵 rref ()

命令形式：rref (a)

功能：通过行变换，把矩阵 a 变为上三角阵.

9.2.3 模型求解在 Matlab 中的实现

在 Matlab 中的实现程序如下：

由 Matlab 软件的编辑器构造 M 函数文件：jt. m 如下：

```
A5 = [ 0   1   -1  1   0   0   0   0   0   0
       0   0   0   1   1   0   0   0   0   0
       0   0   0   0   0   -1  1   0   0   0
       1   1   0   0   0   0   0   0   0   0
       1   0   0   0   1   0   0   0   0   0
       0   0   0   0   0   0   1   1   0   0
       0   0   0   0   0   0   0   0   1   0
       0   0   0   0   0   0   0   0   -1  1
       0   0   0   0   0   0   0   0   0   1
       0   0   1   0   0   1   0   1   0   0];
```

b = [300，500，200，800，800，1000，400，200，600，1000]；

B_5 = [A_5'；b]'；

Rank (A_5)；

Rank (B_5)；

B_6 = rref (B_5)。

在 Matlab 中输入 M 函数文件名 jt 得

增广矩阵 B_5 =[

0	1	-1	1	0	0	0	0	0	0	300
0	0	0	1	1	0	0	0	0	0	500
0	0	0	0	0	-1	1	0	0	0	200
1	1	0	0	0	0	0	0	0	0	800
1	0	0	0	1	0	0	0	0	0	800
0	0	0	0	0	0	1	1	0	0	1000
0	0	0	0	0	0	0	0	1	0	400
0	0	0	0	0	0	0	0	-1	1	200
0	0	0	0	0	0	0	0	0	1	600
0	0	1	0	0	1	0	1	0	0	1000]；

系数矩阵的秩　　rank (A_5) = 8

增广矩阵的秩　　rank (B_5) = 8

增广矩阵阶梯形最简形式为：

B_6 = [

1	0	0	0	1	0	0	0	0	0	800
0	1	0	0	-1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	200
0	0	0	1	1	0	0	0	0	0	500
0	0	0	0	0	1	0	1	0	0	800
0	0	0	0	0	0	1	1	0	0	1000
0	0	0	0	0	0	0	0	1	0	400
0	0	0	0	0	0	0	0	0	1	600
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0]；

其对应的齐次同解方程组为

$$\begin{cases} x_1 + x_5 = 0 \\ x_2 - x_5 = 0 \\ x_3 = 0 \\ x_4 + x_5 = 0 \\ x_6 + x_8 = 0 \\ x_7 + x_8 = 0 \\ x_9 = 0 \\ x_{10} = 0 \end{cases}$$

其中 (x_5, x_8) 为自由取值未知量，分别赋两组值为 $(1,0)$，$(0,1)$，得齐次方程组基础解系中两个向量

$$\eta_1 = \begin{pmatrix} -1 \\ 1 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \eta_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

其对应的非齐次方程组为

$$\begin{cases} x_1 + x_5 = 800 \\ x_2 - x_5 = 0 \\ x_3 = 200 \\ x_4 + x_5 = 500 \\ x_6 + x_8 = 800 \\ x_7 + x_8 = 1000 \\ x_9 = 400 \\ x_{10} = 600 \end{cases}$$

赋值给自由未知量 (x_5, x_8) 为 $(0,0)$ 得非齐次方程组的特解

$$x^* = \begin{pmatrix} 800 \\ 0 \\ 200 \\ 500 \\ 0 \\ 800 \\ 1000 \\ 0 \\ 400 \\ 600 \end{pmatrix}$$

于是方程组的通解为 $X = k_1\boldsymbol{\eta}_1 + k_2\boldsymbol{\eta}_2 + x^*$，其中 k_1, k_2 为任意常数，X 的每一个分量即为交通网络未知部分的具体流量. X 有无穷多解.

§9.3　常染色体的隐性疾病

有些疾病是先天性疾病，这是基因遗传的结果. 在常染色体的遗传中，后代是从每个父母的基因对中各继承一个基因，形成自己的基因型，基因型确定了后代所表现的特征. 如果基因 A 和 a 控制某种遗传疾病，其中 A 为显性基因，a 为隐性基因，则根据这种遗传病对应的基因型可以将人口分为三类：AA 基因型的正常人，Aa 基因型的隐性患者，aa 基因型的显性患者.

由于后代是各从父体或母体的基因对中等可能地得到一个基因形成自己的基因对，故父母代的基因对和子代各基因对之间的转移概率如表 9.2 所示.

表 9.2　　父母代—子代基因转移概率

概　率		父体 - 母体基因型				
		AA—AA	AA—Aa	AA—aa	Aa—Aa	aa—aa
子代基因型	AA	1	1/2	0	1/4	0
	Aa	0	1/2	1	1/2	1/2
	aa	0	0	0	1/4	1/2

设这些患者在第 n 代人口中所占的比例分别为 $x_1^{(n)}, x_2^{(n)}, x_3^{(n)}$，在控制结合的情况下，当前社会中没有显性患者，只有正常人和隐性患者，且他们分别占总人口的 85% 和 15%. 考虑下面两种结合方式对后代该遗传病基因型分布的影响.

(1) 同类基因型结合.

(2) 显性患者不允许生育,隐性患者必须与正常人结合.

9.3.1 问题分析及模型的建立

设当前该遗传病的人口比例状况为初始分布 $x_1^{(1)}, x_2^{(1)}, x_3^{(1)}$;以后第 n 代的分布为 $x_1^{(n)}, x_2^{(n)}, x_3^{(n)}$,令

$$A = \begin{pmatrix} 1 & 1/4 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1/4 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1/2 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad X^{(n)} = \begin{pmatrix} x_1^n \\ x_2^n \\ x_3^n \end{pmatrix}, \quad X^{(1)} = \begin{pmatrix} 85\% \\ 15\% \\ 0 \end{pmatrix}$$

则

$$X^{(n)} = A^{n-1} X^{(1)}, \quad X^{(n)} = B^{n-1} X^{(1)}$$

9.3.2 求解模型所需的知识点及其在 Matlab 中的实现

求矩阵的特征值和特征向量 eig ():

命令形式 1:d = eig (A)

功能:返回方阵 A 的全部特征值组成的特征向量 d.

命令形式 2:[V, D] = eig (A)

功能:返回矩阵 A 的特征值矩阵 D 与特征向量矩阵 V,满足 AV = VD.

9.3.3 模型求解在 Matlab 中的实现

1. 下面程序是在第一种方式下模拟 20 代以后该遗传病基因型的分布在 Matlab 中的程序实现:

```
clear ; A = [1  1/4  0 ;0  1/2  0 ;0  1/4  1] ;
X = [0.85 ;0.15 ;0] ;
for  j = 1:20
X = A * X ;
end
X20 = X
X = [0.85 ;0.15 ;0] ;
c = [1 1 1]' ;
 n = 0;
 while     X ~ = c
           c = X ;
```

```
        n = n +1 ;
        X = A * X ;
  end
  format long ;
  X, n.
```

其程序运算结果为

```
X20 = 0.924999928        X = 0.92500000        n =31
      0.000000143              0.00000000
      0.074999928              0.07500000
```

由上述结果可以看出,按第一种方式结合,第 20 代后,基因分布趋于稳定.程序中的 while 语句是计算在该疾病基因型分布稳定所需要的代数及稳定时的基因型分布,结果表明第 51 代后该疾病基因型分布稳定,将出现 7.5% 的稳定显性患者,而隐性患者消失.

2. 按第二种结合方式,以下程序计算 20 代后的基因型分布程序在 Matlab 中的程序实现:

```
clear ;
B = [1  1/2  0; 0  1/2  0 ;0  0  0] ;
X = [0.85 ;0.15 ;0] ;
For  j =1 :20
X = B * X;
End
Format shot ;
X
```

其运算结果为

```
X = 1.000
    0.000
    0.000.
```

结果表明,若干代以后,不但不会出现显性患者,连隐性患者也趋于消失,所以为了避免某些遗传疾病的发生,最好采用一些有效的控制结合手段.

下面用特征值和特征向量以及相似对角形矩阵的理论作进一步分析,矩阵 A 的特征值和特征向量由以下命令可以求出:

```
A = [1, 2/4, 0 ;0 , 1/2, 0 ;0, 1/4, 1] ;
  [P, T] = eig (A)
P = 1.0000      0        - 0.4082
```

```
      0          0          0.8165
      0          1.000      - 0.4082
T  =  1.0000     0            0
      0          1.0000       0
      0          0            0.5000
```

求得三个特征值 1,1,0.5,对应的特征向量为

(1,0,0)',(0,0,1)',(-0.4082,0.8165,-0.4082)'

由于三个特征向量线性无关,从而 A 可以相似对角化,那么有

$$A = PTP^{-1},$$

$$A^n = (PTP^{-1})^n = P\begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix}P^{-1}$$

因此,$\lim_{n\to} A^n$ 可以由下面命令求出:

```
D  =  P * diag([1,1,0]) * inv(P)
D = 1.0000     0.5000     0
    0          0          0
    0          0.5000     1.0000
```

设 x_1, x_2, x_3 为任意初始分布,则 $\lim_{n\to} X$ 可以由以下命令求出:

```
syms   x1   x2   x3
D *  [x1;x2;x3]
Ans =  [x1 + 1/2 * x2]
       [          0]
       [1/2 * x2 + x3].
```

§9.4 真题解析:治理环境的投入和收益问题

随着工业化社会的发展,环境污染已成为人们面临的三大困境(环境污染、能源短缺、人口膨胀)之一.然而,环境污染又与人类生产活动密切相关,环境中的污染物大都来自人类生产活动,人们在利用各种资源创造大量物质财富的同时,也排出了大量的污染物,从而造成了环境污染.因而将环境污染问题的研究与人类生产活动联系在一起考虑,投入产出分析是对人类生产活动和各行业之间相互联系进行分析的一种方法.列昂季耶夫对环境保护问题提出了以下的投

入产出分析模型.

该环境保护投入产出模型的基本结构如表 9.3 所示,表中除了通常几个生产部门外,增加了几个污染部门.

表 9.3　　环境保护的投入产出表

		中间产品		最终产品及最终产品领域所产生的污染	总产品及污染物产生总量
		生产部门 1,2,…,n	消除污染部门 1,2,…,m		
生产部门	1	$x_{11}\cdots x_{1n}$	$E_{11}\cdots E_{1m}$	y_1	x_1
	2	$x_{21}\cdots x_{2n}$	$E_{21}\cdots E_{2m}$	y_2	x_2
	⋮	⋮　⋮	⋮　⋮	⋮	⋮
	n	$x_{n1}\cdots x_{nn}$	$E_{n1}\cdots E_{nm}$	y_n	x_n
污染部门	1	$P_{11}\cdots P_{1n}$	$F_{11}\cdots F_{1m}$	R_1	Q_1
	2	$P_{21}\cdots P_{2n}$	$F_{21}\cdots F_{2m}$	R_2	Q_2
	⋮	⋮　⋮	⋮　⋮	⋮	⋮
	n	$P_{m1}\cdots P_{mn}$	$F_{m1}\cdots F_{mm}$	R_n	Q_n
固定资产折旧		$d_1,d_2,\cdots,d_n$	$\bar{d}_1,\bar{d}_2,\cdots,\bar{d}_n$		
新创造价值	劳动报酬	$v_1,v_2,\cdots,v_n$	$\bar{v}_1,\bar{v}_2,\cdots,\bar{v}_n$		
	社会纯收入	$m_1,m_2,\cdots,m_n$	$\bar{m}_1,\bar{m}_2,\cdots,\bar{m}_n$		
总产品及污染物消除总量		$x_1,x_2,\cdots,x_n$	$\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_n$		

表 9.3 中各项符号的意义如下：

x_i——第 i 个部门产品的总产出；

y_i——第 i 个部门产品的最终产出；

x_{ij}——第 j 个部门生产过程中所消耗的第 i 个部门产品的数量；

E_{ij}——第 j 个消除污染部门在消除污染过程中所消耗的第 i 个部门产品的数量；

P_{ij}——第 j 个部门生产过程中所产生的第 i 种污染物的数量；

F_{ij}——第 j 个消除污染部门本身所产生的第 i 种污染物的数量；

R_i——最终需求领域所产生的第 i 种污染物的数量；

Q_i——第 i 种污染物的总量；

S_j——第 j 个消除污染部门消除污染物的总消除量；

d_j——第 j 个生产部门的固定资产折旧；

$\bar{d}_j$——第 j 个消除污染部门的固定资产折旧；

v_j——第 j 个生产部门的劳动报酬；

$\bar{v}_j$——第 j 个消除污染部门的劳动报酬；

m_j——第 j 个生产部门所创造的社会纯收入；

$\bar{m}_j$——第 j 个消除污染部门所创造的社会纯收入.

从表 9.3 的水平方向看，有两组平衡方程：一组是产品的生产与消耗的平衡方程；另一组是污染物的形成方程，即

$$\begin{cases} \sum_{j=1}^{n} x_{ij} + \sum_{j=1}^{m} E_{ij} + y_i = x_i & (i = 1,2,\cdots,n) \\ \sum_{j=1}^{n} P_{ij} + \sum_{j=1}^{m} F_{ij} + R_i = Q_i & (i = 1,2,\cdots,m) \end{cases}$$

上式中第一式说明，总产品 x_i 除去最终产品 y_i 以外，其余则用做产品生产的消耗和消除污染部门的消耗；第二式说明，污染物 Q_i 来自三个方面，即生产领域、最终需求领域和消除污染部门.

用 a_{ij} 表示消除污染部门消除一个单位的第 j 种污染物所消耗的第 i 个部门产品的数量，a_{ij} 称为消除污染部门的直接消耗系数；p_{ij} 表示第 j 个部门单位产品生产过程中所产生的第 i 种污染物的数量，p_{ij} 称为生产部门污染物的产生系数；f_{ij} 表示第 j 个消除污染部门在消除一个单位污染物中所新产生的第 i 种污染物数量，f_{ij} 称为消除污染部门污染物的产生系数. 于是得到以下系数矩阵：

生产部门直接消耗系数矩阵为

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

消除污染部门直接消耗系数矩阵为

$$E = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1m} \\ e_{21} & e_{22} & \cdots & e_{2m} \\ \vdots & \vdots & & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{nm} \end{pmatrix}$$

生产部门污染物产生系数矩阵为

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{pmatrix}$$

消除污染部门污染物产生系数矩阵为

$$F = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1m} \\ f_{21} & f_{22} & \cdots & f_{2m} \\ \vdots & \vdots & & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mm} \end{pmatrix}$$

并令 $X = (x_1, x_2, \cdots, x_n)^T$, $Y = (y_1, y_2, \cdots, y_n)^T$, $S = (S_1, S_2, \cdots, S_m)^T$, $R = (R_1, R_2, \cdots, R_m)^T$, $Q = (Q_1, Q_2, \cdots, Q_m)^T$. 则两组平衡方程的矩阵表示为

$$\begin{cases} AX + ES + Y = X \\ PX + FS + R = Q \end{cases}$$

设 a_i $(0 \leqslant a_i \leqslant 1)$ 表示第 i 种污染物的消除比例，则

$$S_i = a_i Q_i$$

引入对角矩阵

$$a = \begin{pmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_m \end{pmatrix}$$

则向量 S 与 Q 就有关系：$S = aQ$，将其代入平衡方程，并整理可得

$$\begin{pmatrix} I - A & -Ea \\ -P & I - Fa \end{pmatrix} \begin{pmatrix} X \\ Q \end{pmatrix} = \begin{pmatrix} Y \\ R \end{pmatrix}$$

解得

$$\begin{pmatrix} X \\ Q \end{pmatrix} = \begin{pmatrix} I - A & -Ea \\ -P & I - Fa \end{pmatrix}^{-1} \begin{pmatrix} Y \\ R \end{pmatrix}$$

上式表明，如果已知最终产品 Y 和最终产品领域所产生的污染量 R，就可以求出应生产的总产品量 X 和产生的污染物总量 Q. 污染物的残存量为

$$Q_{残} = Q - aQ = (I - a)Q$$

以上是从表 9.3 的水平方向研究得到的结论，再从垂直方向上研究. 并以价值单位作为生产部门的计量单位，来研究消除污染的费用及其对产品价格的影响.

9.4.1 生产部门的费用构成

对于生产部门,在考虑消除污染费用之前的平衡关系是

$$\sum_{i=1}^{n} x_{ij} + d_j + v_j + m_j = x_j \quad (j = 1,2,\cdots,n)$$

如果进行消除污染活动,势必要提高产品价格. 设 π_i $(i=1,2,\cdots,n)$ 表示第 i 个部门产品价格提高率;φ_i $(i=1,2,\cdots,m)$ 表示消除一个单位第 i 种污染物的费用. 这时,新的平衡关系式为

$$\sum_{i=1}^{n} (1+\pi_i) x_{ij} + \sum_{i=1}^{m} \varphi_i a_i P_{ij} + d_j + v_j + m_j = (1+\pi_j) x_j \quad (j = 1,2,\cdots,n)$$

由以上两组平衡关系式可得

$$\sum_{i=1}^{n} \pi_i a_{ij} + \sum_{i=1}^{m} \varphi_i a_i P_{ij} = \pi_j \quad (j = 1,2,\cdots,n)$$

令 $H = (\pi_1,\pi_2,\cdots,\pi_n)^T$,$\boldsymbol{\Phi} = (\varphi_1,\varphi_2,\cdots,\varphi_n)^T$,则

$$A^T H + P^T a \boldsymbol{\Phi} = H.$$

9.4.2 消除污染部门的费用

第 j 个消除污染部门的费用总额为 $\varphi_j S_j$,因而第 j 个消除污染部门的费用之平衡关系为

$$\sum_{i=1}^{n} (1+\pi_i) E_{ij} + \sum_{i=1}^{m} \varphi_i a_i F_{ij} + \bar{d}_j + \bar{v}_j + \bar{m}_j = \varphi_j S_j, \quad (j = 1,2,\cdots,m)$$

两边同除以 S_j,并令

$$h_j = \sum_{i=1}^{n} e_{ij} + \frac{\bar{d}_j + \bar{v}_j + \bar{m}_j}{S_j} \quad (j = 1,2,\cdots,m),$$

则

$$\sum_{i=1}^{n} \pi_i e_{ij} + \sum_{i=1}^{m} \varphi_i a_i f_{ij} + h_j = \varphi_j \quad (j = 1,2,\cdots,m)$$

用矩阵表示,则有

$$E^T H + F^T a \boldsymbol{\Phi} + H = \boldsymbol{\Phi},$$

式中

$$H = (h_1,h_2,\cdots,h_m)^T.$$

上述两个结论可以合并得到

$$\begin{pmatrix} H \\ \boldsymbol{\Phi} \end{pmatrix} = \begin{pmatrix} A^T & P^T a \\ E^T & F^T a \end{pmatrix} \begin{pmatrix} H \\ \boldsymbol{\Phi} \end{pmatrix} + \begin{pmatrix} 0 \\ H \end{pmatrix}$$

解得

$$\begin{pmatrix} H \\ \boldsymbol{\Phi} \end{pmatrix} = \begin{pmatrix} I - A^T & -P^T a \\ -E^T & I - F^T a \end{pmatrix} \begin{pmatrix} 0 \\ H \end{pmatrix}$$

由此可以消除一个单位污染物的费用向量 $\boldsymbol{\Phi}$,以及由于消除污染各生产部门价格的上涨率为 H.

欧洲曾于 20 世纪 70 年代用类似的方法计算出消除污染各部门产品价格的影响,如表 9.4 所示.

表 9.4　　消除污染对各部门产品价格的影响

部门	农业	纺织业	煤矿	石化品	煤油	金属制品及机械制造	建筑业
中期 (H)	0.22	1.00	0.10	0.47	0.11	0.11	0.18
长期 (H)	1.67	6.25	0.06	2.99	1.65	0.97	1.30

由表 9.4 可以看出,中期消除污染对各部门产品价格的影响的百分率比长期小,这是由于中期各种污染物的消除比例较长期低. 相关统计如表 9.5 所示.

表 9.5

	中间产品生产部门	消除污染部门	最终产品及最终产品领域所产生的污染	总产品及污染物产生总量
生产部门	$x_{11} \cdots x_{1n}$ $x_{21} \cdots x_{2n}$ $\vdots \quad \vdots$ $x_{n1} \cdots x_{nn}$	$E_{11} \cdots E_{1m}$ $E_{21} \cdots E_{2m}$ $\vdots \quad \vdots$ $E_{n1} \cdots E_{nm}$	y_1 y_2 $\vdots$ y_n	x_1 x_2 $\vdots$ x_n
污染部门	$P_{11} \cdots P_{1n}$ $P_{21} \cdots P_{2n}$ $\vdots \quad \vdots$ $P_{m1} \cdots P_{mn}$	$F_{11} \cdots F_{1m}$ $F_{21} \cdots F_{2m}$ $\vdots \quad \vdots$ $F_{m1} \cdots F_{mm}$	R_1 R_2 $\vdots$ R_n	Q_1 Q_2 $\vdots$ Q_n

续表

	中间产品生产部门	消除污染部门	最终产品及最终产品领域所产生的污染	总产品及污染物产生总量
固定资产折旧	d_1 d_2 … d_n	$\bar{d}_1$ $\bar{d}_2$ … $\bar{d}_n$		
新创造价值 劳动报酬	v_1 v_2 … v_n	$\bar{v}_1$ $\bar{v}_2$ … $\bar{v}_n$		
新创造价值 社会纯收入	m_1 m_2 … m_n	$\bar{m}_1$ $\bar{m}_2$ … $\bar{m}_n$		
总产品及污染物消除总量	x_1 x_2 … x_n	$\bar{x}_1$ $\bar{x}_2$ … $\bar{x}_n$		

习 题 10

1. 某农场饲养的某种动物所能达到的最大年龄为 15 岁,将其分为三个年龄组,第一组为 0 ~ 5 岁,第二组为 6 ~ 10 岁,第三组为 11 ~ 15 岁. 动物从第二年龄组开始繁殖后代,经过长期统计,第二年龄组的动物在其年龄段平均繁殖了 4 个后代,第三年龄组的动物在其年龄段平均繁殖了 3 个后代,第一年龄组和第二年龄组的动物能顺利进入下一年龄组的存活率分别为 1/2 和 1/4. 假设农场现有三个年龄组的动物各 1000 头,试问 15 年后三个年龄段的动物各多少头?

2. 一个木工,一个电工,一个油漆工,三人相互同意彼此装修他们自己的房子. 在装修前,他们达成了如下协议:(1) 每人总共工作 10 天 (包括给自己家干活在内);(2) 每人的日工资根据一般的市价在 60 ~ 80 元之间;(3) 每人的日工资数应使得每人的总收入与总支出相等. 表 9.6 是他们协商后制定出的工作天数的分配方案.

表 9.6

	木 工	电 工	油漆工
在木工家工作的天数	2	1	6
在电工家工作的天数	4	5	1
在油漆工家工作的天数	4	4	3

试计算木工、电工和油漆工他们的日工资各为多少？

3. 种群的数量因繁殖而增加，因自然死亡而减少，对于人工饲养的种群（比如家畜）而言，为了保证稳定的收获，各个年龄的种群数量应维持不变. 种群因雌性个体的繁殖而改变，为方便起见以下种群数量均指其中的雌性.

种群年龄记做 x_k，繁殖率记做 b_k（每个雌性个体一年繁殖的数量），自然存活率记做 s_k（$s_k = 1 - d_k$，d_k 为一年的死亡率），收获量记做 h_k，则来年年龄 k 的种群数量 $\tilde{x}_k$ 应为

$$\tilde{x}_k = \sum_{k=1}^{n} b_k x_k - h_1, \tilde{x}_{k+1} = s_k x_k - h_{k+1}, k = 1, 2, \cdots, n$$

要求各个年龄的种群数量每年维持不变就是要使 $\tilde{x}_k = x_k$.

(1) 若 b_k，s_k 已知，给定收获量 h_k，试建立求各年龄的稳定种群数量 x_k 的模型（用矩阵、向量表示）.

(2) 设 $n = 5$，$b_1 = b_2 = b_5 = 0$，$b_3 = 5$，$b_4 = 3$，$s_1 = s_4 = 0.4$，$s_2 = s_3 = 0.6$，如果要求 $h_1, h_2, \cdots, h_5$ 为 500，400，200，100，100，试求 $x_1, x_2, \cdots, x_5$.

(3) 欲使 $h_1, h_2, \cdots, h_5$ 达到 500，试问如何达到？

4. 国民经济各个部门之间存在着相互依存的关系，每个部门在运转中将其他部门的产品或半成品经过加工（称为投入）变成自己的产品（称为产出）. 设国民经济仅由农业、制造业和服务业三个部门构成，已知某年它们之间的投入产出关系、需求关系、初始投入如表 9.7 所示（数字表示产值量，单位为亿元）.

表 9.7　（单位：亿元）

产出 / 投入	农　业	制造业	服务业	外部需求	总产出
农　业	15	20	30	35	100
制造业	30	10	45	115	200
服务业	20	60	/	70	110
初始投入	35	110	75		
总投入	100	200	150		

表 9.7 中第一行数字表示，农业总产出为 100 亿元，其中 15 亿元农产品用

于农业生产本身,20 亿元用于制造业,30 亿元用于服务业,剩下 35 亿元农产品用来满足外部需求. 第二、三行数字意义相同. 第一列数字中,15 亿元如上所述,30 亿元是制造业对农业的投入,20 亿元是服务业对农业的投入,35 亿元的初始投入包括工资、税收、进口等,总投入 100 亿元与总产出相等.

假定每个部门的产出与投入是成正比的,由表 9.7 能确定这三个部门的投入产出表,如表 9.8 所示.

表 9.8

产出 / 投入	农　业	制造业	服务业
农　业	0.15	0.10	0.20
制造业	0.30	0.05	0.30
服务业	0.20	0.30	0

表 9.8 中第一行,第二列的数字 0.10 表示生产一个单位产值的制造业产品需投入 0.10 个单位产品的农产品,这是由表 9.7 中 20 亿元农产品投入制造业可以产出 200 亿元制造业总产值而来的 (20/200 = 0.1). 同样,第三行、第一列的数字 0.20 表示,生产 1 个单位产值的农产品需要 0.20 个单位的服务业产值,因为表 9.7 中 20 亿元的服务业产值投入农业,得到 100 亿元的农业总产值. 表 9.8 中的数字称为投入系数或消耗系数,如果技术水平没有变化,可以假设投入系数是常数.

(1) 设有 n 个部门,已知投入系数,给定外部需求,试建立求解各部门总产出的模型.

(2) 设投入系数如表 9.8 所示,如果今年对农业、制造业和服务业的外部需求分别为 50,150,100 亿元,试问这三个部门的总产出分别应为多少?

(3) 如果三个部门的外部需求分别增加 1 个单位,试问它们的总产出应分别增加多少?

(4) 如果对于任何给定的、非负的外部需求,都能得到非负的总产品,模型就称为可行的. 试问为使模型可行,投入系数应满足什么条件?

第十章　图论方法模型在 Matlab 中的求解

§10.1　图、最短路径和最小生成树

10.1.1　图的基本概念及其矩阵表示法

1. 图中顶点与边的基本概念

(1) 若边 e 的顶点为 u,v,则称 e 与顶点 u,v 相关联.

(2) 若顶点 u,v 之间有边相连,则称 u 与 v 相邻.

(3) 若边 e_1,e_2 与同一顶点相关联,则称边 e_1,e_2 相邻.

(4) 端点相同的两条边称为重边. 两端点为同一点的一条边称为环.

例如在图 10.1 中,e_1 与 v_1 相关联;v_1 与 v_2 两顶点相邻,e_1 与 e_2 两边相邻,e_4 与 e_5 是重边,e_6 是环.

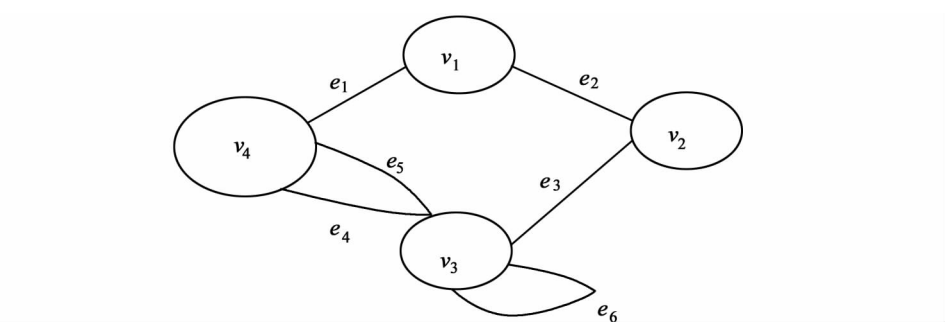


图 10.1

下面介绍两种特殊图:

(1) 简单图:无重边,无环的图称为简单图.

(2) 完全图:任意两顶点之间皆有边相连的简单图,称为完全图. 完全图记为 K_n,n 为图的顶点数.

若图的每条边都有方向,则称为有向图. 若给每条边赋予一个或多个实数,则称该图为网络,记为 d (v) ;一般用 v (G) 和 ε (G) 分别表示图 G 的顶点数和边数.

在有向图中,从顶点 v 引出的边的数目称为 v 的出度,记为 d^{*} (v) ;指向顶点 v 的边的数目称为 v 的入度,称为 d^{-} (v) .

2. 用矩阵表示图的方法

各对象元素之间的二元关系可以用图来描述,这种描述方式很直观、形象,可以使一些抽象的关系变得简单明了. 但通过计算机来解决图的问题时,必须将图转化为矩阵来表示. 下面是表示图的几种矩阵形式.

(1) 邻接矩阵

无向图的邻接矩阵 :$A = (a_{ij})_{n\times n}$,其中

$$a_{ij} = \begin{cases} 1, & \text{当 } v_i \text{ 与 } v_j \text{ 相邻} \\ 0, & \text{当 } v_i \text{ 与 } v_j \text{ 不相邻} \end{cases}$$

图 10.2 对应的邻接矩阵为

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

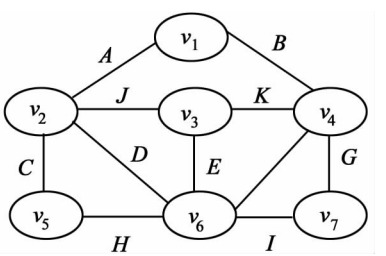


图 10.2

无向图的邻接矩阵是对称的,每行的元素之和为对应顶点的次数,每列的元素之和也为对应顶点的次数.

对有向图,其邻接矩阵 $A = (a_{ij})_{n\times n}$ 中的元素 a_{ij} 取为 v_i 指向 v_j 的有向边的数目,图 10.3 中有向图的邻接矩阵为 :

加权有向图的带权邻接矩阵 :$A = (a_{ij})_{n\times n}$,其中元素 a_{ij} 取为 v_i 指向 v_j 的有向边上的权. 当无边时取为 ,对角线上的元素为 0.

加权无向图的邻接矩阵可以类似定义,其邻接矩阵是对称矩阵.

$$\begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

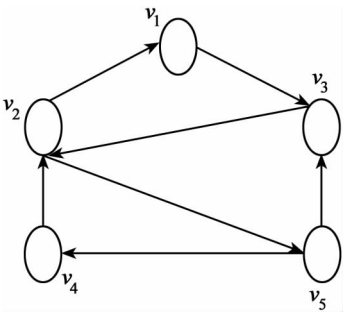


图 10.3

(2) 关联矩阵

无向图的关联矩阵：$M = (m_{ij})_{n \times m}$，其中

$$m_{ij} = \begin{cases} 1, & 若\ v_i\ 和\ e_j\ 相关联 \\ 0, & 若\ v_i\ 和\ e_j\ 不相关联 \end{cases}$$

图 10.1 中无向图的关联矩阵为

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

注意：关联矩阵每行的元素之和为对应顶点的次数，每列的元素之和为 2.

有向图的关联矩阵：$M = (m_{ij})_{n \times m}$，其中 m_{ij} 的取值为 1，-1，0，分别对应于 v_i 是 e_j 的起点，v_i 是 e_j 的终点和 v_i 不是 e_j 的端点三种情形.

(3) 边权矩阵

定义一个 $2 \times m$ 的矩阵 E，第一、二行分别存放边的起点和终点. 若第 i 条边 e_i 的起点和终点分别为 v_j，v_k，则 $E(1,i) = j$，$E(2,i) = k$.

图 10.1 对应的边权矩阵为

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

对加权图，只需增加一行来存放各条边上的权，这样的矩阵称为边权矩阵.

10.1.2 最小生成树算法及其在 Matlab 中的实现

树是图论中很重要的一类图,下面先看几个关于树的基本概念:

(1) 连通图:任意两点之间都有路径的图称为连通图,如图 10.4 就是一个连通图.

(2) 圈:当一条路径的起点和终点是同一顶点时,称这条路径为圈. 如图 10.4 中的路径 1 - 2 - 5 - 4 - 1 就是一个圈.

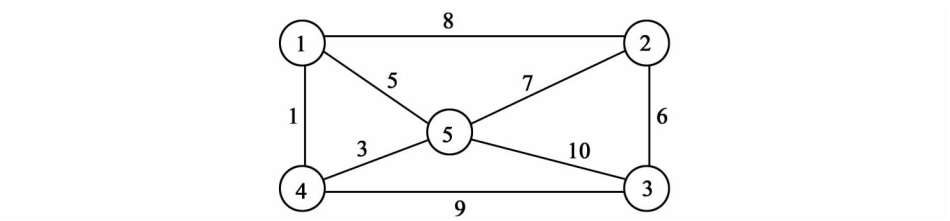


图 10.4

树的定义如下:

树:没有圈的连通图称为树. 如图 10.5 就是一棵树.

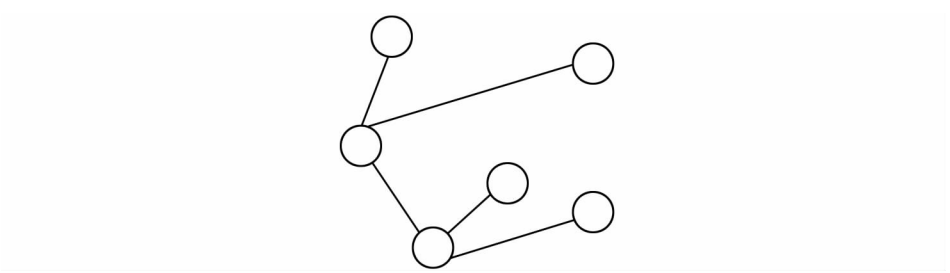


图 10.5

树具有很多比较好的性质:

(1) 树中任意两点之间有惟一路径.

(2) 树的边数恰好等于顶点数减 1.

(3) 在树中任意去掉一条边,将会不连通.

(4) 树中任意两个不相邻顶点间添一条边后,就恰好含一个圈.

在实际问题中,很多时候都要求树的最小生成树或最大生成树等. 它们的定义如下:

子图：

图 G 的子图：由 G 的一些边和一些顶点组成，这些边和顶点是 G 的一个部分图，且必须满足：当一条边在子图中时，这条边的两个端点也要在子图中.

生成树：图 G 的生成树是图 G 的子图，其顶点集等于 G 的顶点集.

生成树具有下面的性质：

(1) 每个生成树对应的线路费用都低于原图对应的线路费用.

(2) 生成树中没有多余边，随意去掉其中的一条边都会破坏其连通性.

(3) 定义生成树的权为所有边权之和.

最小生成树和最大生成树的定义：

最小生成树：在一个加权连通图 G 中，权最小的那棵生成树称为 G 的最小生成树.

最大生成树：在一个加权连通图 G 中，权最大的那棵生成树称为 G 的最大生成树.

注意：一个简单连通图只要不是树，其生成树就不惟一，而且非常多. 一般地，n 个顶点的完全图，其生成树有 n^{n-2} 个. 要求最小生成树或最大生成树，一般不能用穷举法，因为 30 个点完全图，其生成树就有 30^{28} 个，用计算机来穷举其最小生成树，时间上是不可接受的.

10.1.3　最小生成树算法及其在 Matlab 中的应用

求图的最小生成树最常用的有两种算法：Prim 算法和 Kruskal 算法，这两种方法都是由贪婪法的思想发展而来. 贪婪法的应用非常广泛，该方法具有以下特点：

(1) 贪婪法把解看成是由若干个部件构成，每一步求出解的一个部件，但这不是从整体的角度去考虑求最优，而只是求局部或当前的最好解. 求出的一个个部件组合而作为最终的解.

(2) 贪婪法只是一种试探法，计算上简便、有效，可以提供一个近似的正确解. 但一般情况下，并不能保证输出的解是正确的.

1. Prim 算法

Prim 算法是一种贪婪算法，该方法的基本思路是：任选一个顶点 v_1，将其涂红，其余顶点为白点；在一个端点为红色，另一个端点为白色的边中，找一条权最小的边涂红，把该边的白端点也涂成红色；这样下去，每次将一条边和一个顶点涂成红色，直到所有的顶点都成红色为止. 最终的红色边和顶点便是最小生成树. 这就是 Prim 算法寻找最小生成树的逐步生长过程.

从 Prim 算法的思想中可以看出，Prim 算法的关键就是如何找出连接红点与

白点的具有最小权的边. 若把 G 看成完全图,当前有 k 个红点,则有 k(n - k) 条连接红、白点的白边. 从如此多的白边中选取最短边很费时间. 可以构造一个较小的候选边集,只要保证最短边在里面即可.

事实上,对每个白点,从该点到各红点的边中,必存在一条最短的白边,只要将所有 n - k 个白点所关联的最短边作为候选集,就必定能保证所有 k(n - k) 条连接红、白点的白边属于该候选集.

在 Matlab 中实现 Prim 算法的基本思路如下:

输入加权图的带权邻接矩阵 $[a(i,j)]_{n\times n}$.

(1) 建立初始候选边集 B,T = ϕ;

(2) 从候选边中选取最短边 (u,v),T = T∪(u,v);

(3) 调整候选边集 B;

(4) 重复 (2),(3) 直到 T 含有 n - 1 条边.

例 1. 在 Matlab 中用 Prim 算法求图 10.4 所示的加权图的最小生成树.

解 Matlab 的实现程序如下:

```
% Prim.m
a = [0     8     inf    1     5
     8     0     6      inf   7
     inf   6     0      9     10
     1     inf   9      0     3
     5     7     10     3     0];
T = [];
e = 0; v = 1; n = 5;  sb = 2:n;  %1 代表第一个红点,sb 代表白点集.
for j = 2:n            %构造初始候选边的集合.
 b(1,j-1) = 1;
 b(2,j-1) = j;
 b(3,j-1) = a(1,j);
end
while size(T,2) < n-1
      [min,i] = min(b(3,:));%在候选边中找最短边.
      T(:;size(T,2) + 1) = b(:,i);
      e = e + b(3,i);
      v = b(2,i);      %v 表示新涂的红点.
      temp = find(sb == b(2,i));
```

```
        sb(temp) = [];   b(:, j) = [];
        For  j = 1 :length(sb)        % 调整候选边.
        d = a(v, b(2, j));
        If  d < b(3, j)
            B(1,j) = v;  b(3, j) = d;
        End
    End
End
T,c % 显示所求的最小生成树的边集合以及最小生成树各边权值的和.
```

程序运行的结果为

```
T = 1   4   5   2
    4   5   2   3
    11  3   7   6
c = 17.
```

因此,所求的图的最小生成树的边集合为{(1,4),(4,5),(5,2),(2,3)};其边权值的和为 17.

2. Kruskal 算法

Kruskal 算法也是一种贪婪法.

假设给定了一个加权连通图 G,G 的边集合为 E,顶点个数为 n,则 Kruskal 算法寻找 G 的最小生成树 T 的基本思想为:

假设最小生成树 T 中的边和顶点均涂为红色,其余边为白色. 初始时 G 中的边均为白色.

(1) 将所有的顶点涂成红色.

(2) 在白色边中,挑选一条权最小的边,使其与红色边不形成圈,将该白色边涂红.

重复(2)直到有 n - 1 条红色边,这 n - 1 条红色边就构成了最小生成树 T 的边集合.

注意:在用 Kruskal 算法求最小生成树时,在第(2)步中,判断是否形成圈在用程序实现时比较麻烦.

注意到在该算法的执行过程中,红色顶点和红色边会形成一个或一个以上的连通分支,它们都是 G 的子树,一条边与红色边形成圈当且仅当这条边的两个端点属于同一个子树. 因此判断一条边是否与红色边形成圈,只需判断这条边的两端点是否属于同一个子树.

可以用下面的方法来实现该判断:给每个子树一个不同的编号,对每一个顶

点引入一个标记 t,表示这个顶点所在的子树的编号. 当加入一条红色边时,就会使该边两端点所在的两个子树连接起来,成为一个子树,从而两个子树中的顶点标记要改变成一样.

寻找最小树的 Kruskal 算法的编程基本思路如下:

首先定义程序中要用到的变量:

c——生成树的费用;

T——生成树的边集合;

j——迭代的次数;

k——记录已经被选入生成树的边数.

输入加权连通图 G 的边权矩阵 $[b(i,j)]_{m\times 3}$,顶点数为 n.

(1) 整理边权矩阵:将 $[b(i,j)]_{m\times 3}$ 按第三行由小到大的次序重新排列,得到新的边权矩阵 $[B(i,j)]_{m\times 3}$;

(2) 数据初始化: $j=0, T=\phi, c=0, k=0$;对于所有的 $i, t(i)=i$;

(3) 更新 $T, c, t, (i)$; $j=j+1$,若 $t(B(1,j))=t(B(2,j))$,则转步骤(4);否则,若

$$t(B(1,j))\neq t(B(2,j))$$

则 $T=T\cup(B(1,j),B(2,j)), c=c+B(3,j), k=k+1$,对所有的 i,

$$t(i)=\min[t(B(1,j)),t(B(2,j))];$$

(4) 若 $k=n-1$ 或 $j=n$,则终止,输出 T,c,否则返回步骤(3).

从上述分析的 Kruskal 算法中可以看出,Kruskal 算法具有以下特点:

(1) Kruskal 算法最终输出的必定是最小生成树,是最优解.

(2) Kruskal 算法的时间复杂度为 $O(m\log_2 m)$.

例 2. 在 Matlab 中用 Kruskal 算法求图 10.4 所示的加权图的最小生成树.

加权图的存储结构采用边权矩阵 $[b(i,j)]_{m\times 3}$,该算法在 Matlab 中实现的程序代码如下:

```
% Kruskal.m
b = [1    1    1    2    2    3    3    4
     2    4    5    3    5    4    5    5
     8    1    5    6    7    9    10   3];
[B, I] = sortnows(b', 3); B = B';
m = size(b, 2); n = 5;
t = 1:n; k = 0; T = []; c = 0;
for I = 1:m
  if t(B(1, i)) ~= t(B(2, i)),% 判断第 I 条边是否与树中的边形成圈.
```

```
        k = k +1 ; T(k, 1:2) = B(1:2, i), c = c + B(3, i);
            tmin = min (t (B(1, i), t(B(2, i))) ;
            tmax = max(t (B(1, i), t(B(2, i))) ;
            for j = 1 :n
                if t(j) == tmax
                    t (j) = tmin;
            end
            end
    end
     if k == n - 1
                break;
    end
end
T, c        % 输出结果
```

程序运行的结果为

```
T = 1     4
    4     5
    2     3
    2     5
c = 17.
```

因此,图 10.4 的最小生成树的边集合为 {(1, 4), (4, 5), (2, 3), (2, 5)},费用为 17.

10.1.4　最短路算法及其在 Matlab 中的实现

在生产管理,交通运输和通讯领域,经常会碰到这样的问题,沿着哪条路线可以用最短的时间或最少的费用把货物运到目的地?沿哪条路径传送信息最可靠或最快捷?这些都可以看成是在给定的加权图中求最短路径的问题.

求最短路径的常用算法有 Dijkstra 算法和 Floyd 算法. 下面先了解一下图中关于路径的几个定义:

(1) 普通路径长度:普通路径长度定义为该路径所包含的全体边的长度之和.

(2) 最短路径:在图中,从顶点 u 到顶点 v 的路径中普通路径长度最短的路径,称为从 u 到 v 的最短路径.

这里为了方便讨论,设图的顶点集 V = {1, 2, …, n},并假定所有边上

的权值均是表示长度的非负数.

1. 单源最短路径

单源最短路径问题是:对于给定的有向图网络 G = (V,E) 及单个源点 v,求从 v 到 G 的其余各顶点的最短路径.

如图 10.6 所示的有向网络 G,假定以顶点 1 为源点,则源点到其余各顶点的最短路径如表 10.1 所示.

表 10.1

源　　点	中间点	终　　点	路径长度
1		2	10
1		4	30
1	4	3	50
1	4,3	5	60

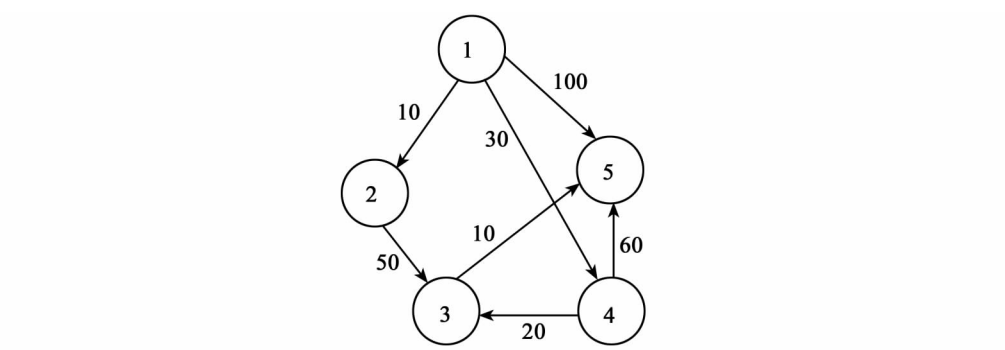


图 10.6

从有向网络 G 可以看出,顶点 1 到顶点 5 的路径有四条:1,5;1,4,5;1,2,3,5 和 1,4,3,5. 这四条路径的长度分别是:100,90,70 和 60. 因而从源点 1 到顶点 5 的最短路径是 1,4,3,5.

一般把源点 v 到终点 w 的最短路径简称为顶点 w 的最短路径.

如何寻找任一给定有向图的单源最短路径呢?表 10.1 是 G 的诸最短路径按长度递增次序排列的一张表. 仔细观察表 10.1 会发现:若按长度递增的次序生成从源点 v 到其他顶点的最短路径,则当前正在生成的最短路径上除终点以外,其余顶点的最短路径均已生成. 当然必须将源点的最短路径看做是已生成的源点到其自身的长度为 0 的路径. 例如,图 G 中,若当前生成的是顶点 3 的最短

路径,则该路径 1,4,3 上顶点 1 和 4 的最短路径在此之前已生成,这是因为它们的最短路径长度比顶点 3 的最短路径长度小.

Dijkstra 算法是按路径长度递增次序产生诸点的最短路径算法. 算法的基本思想是:设置并逐步扩充一个集合 S,存放已求出其最短路径的顶点,则尚未确定最短路径的顶点集合是 V - S. 为了直观,设想 S 中顶点均被涂成红色,V - S 中顶点均被涂成蓝色. 算法初始化时,红点集中仅有一个源点,以后的每一步都是按最短路径长度递增的顺序,逐个把蓝点集中的顶点涂成红色后,加入到红点集中. 换言之,就是在当前蓝点集中选择一个最短路径长度最短的蓝点来扩充红点集. 若从源点到某个蓝点的路径不存在,则可以假设该蓝点的最短路径是一条长度为无穷大的虚拟路径. 这样的蓝点在算法中必定是最后才被涂成红色. 在这样的假定之下,算法直到图中的所有顶点都被涂成红色时终止. 该算法描述为:

while (S 中的红点数 < n)

在当前蓝点集中选择一个路径长度最短的蓝点 k 扩充到红点集中.

现在的问题是如何在当前蓝点集中选择一个路径长度最短的蓝点 k 来扩充红点集合. 由表 10.1 得到的启发是:这种蓝点随对应的最短路径上,除终点外其余顶点都是红点. 因此,对于图中的每一个顶点 i,必须记住从源点 v 到 i 且中间只经过红点(更确切地说是中间不经过蓝点)的最短路径的长度,并将该长度称为顶点 i 的距离. 为此定义一个数组 D[n]来存放各顶点的距离值. 显然每个红点的距离值就是该点的最短路径长度,而蓝点的距离值则不一定是该点的最短路径长度,因为从源点到该蓝点可能存在包含其他蓝点为中间点的更短路径. 但是可以证明,若当前蓝点集中具有最小距离的蓝点是 k,则其距离值 D[k - 1]是 k 的最短路径长度,并且 k 是蓝点集中路径长度最短的顶点.

具体的扩充红点集的方法即为:每一步只要在当前蓝点集中选择一个具有最小距离值的蓝点 k 扩充到红点集中,k 被涂成红色后,剩余的蓝点的距离值可能由于增加了新红点 k 而发生变化(即减少). 因此,必须调整当前蓝点集合中各蓝点的距离值. 该算法描述为

S = {v};

置初始蓝点集中各蓝点的距离值;

while (S 中红点数 < n)

在当前蓝点集中选择距离值最小的顶点 k.

S = S + {k}; /* 将 k 涂成红色加入红点集 */.

调整剩余蓝点的距离值.

现在要解决的问题是调整蓝点的距离值,调整蓝点距离值的方法是:对蓝点集扫描检查,若某蓝点 j 的原距离值 D[j - 1]大于新路径的长度 D[k - 1] + 边

〈k, j〉上的权,则将 D [j - 1] 修改为该长度.

2. 每对顶点间的最短路径算法

显然对问题可以由重复 Dijkstra 算法来解决,每次取决一个顶点作起点,但这需要大量重复计算,效率不高. Floyd 另辟蹊径,提出了比这更好的算法,可以一次性地求出任意两点间的最短路径和距离,其思想方法很有创意,与 Dijkstra 算法截然不同.

Floyd 算法的基本思路是:从图的带权邻接矩阵 $A = [a(i,j)]_{n\times n}$ 开始,递归地进行 n 次更新,即由矩阵 $D^{(0)} = A$,按一个公式,构造出矩阵 $D^{(1)}$;又由 $D^{(1)}$ 用同样的公式构造出矩阵 $D^{(2)}$;最后又由 $D^{(n-1)}$ 用同样的公式构造出矩阵 $D^{(n)}$. 矩阵 $D^{(n)}$ 的 i 行 j 列元素便是 i 号顶点到 j 号顶点的最短路径长度,称 $D^{(n)}$ 为图的距离矩阵,同时还可以引入一个后继点矩阵 path 来记录两点间的最短路径. 其递推公式为

$$D^{(0)} = A$$

$$D^{(1)} = [d_{ij}^{(1)}]_{n\times n}, \text{其中 } d_{ij}^{(1)} = \min\{d_{ij}^{(0)}, d_{i1}^{(0)} + d_{1j}^{(0)}\}$$

$$D^{(2)} = [d_{ij}^{(2)}]_{n\times n}, \text{其中 } d_{ij}^{(2)} = \min\{d_{ij}^{(1)}, d_{i1}^{(1)} + d_{1j}^{(1)}\}$$

$$\vdots$$

$$D^{(n)} = [d_{ij}^{(n)}]_{n\times n}, \text{其中 } d_{ij}^{(n)} = \min\{d_{ij}^{(n-1)}, d_{i1}^{(n-1)} + d_{1j}^{(n-1)}\}$$

注意 $d_{ij}^{(1)}$ 表示中间只允许经过 1 号顶点,从 i 到 j 的路径中最短路径的长度;$d_{ij}^{(2)}$ 表示中间只允许经过 1、2 号顶点,从 i 到 j 的路径中最短路径的长度;……;$d_{ij}^{(k)}$ 表示中间只允许经过 1,2,…,k 号顶点,从 i 到 j 的路径中最短路径的长度;……;$d_{ij}^{(n)}$ 表示中间允许经过 1,2,…,n 号顶点(即任何顶点),从 i 到 j 的路径中最短路径的长度,此即为 i 到 j 的最短路径长度.

上述矩阵序列 $\{D^{(k)}\}$ 可以递归地产生,利用循环迭代便可以简便求出,算法的详细步骤如下:

Floyd 算法步骤:

d (i,j):$d_{ij}^{(k)}$;

path (i,j):对应于 $d_{ij}^{(k)}$ 的路径上 i 的后继点,最终的取值为 i 到 j 的最短路径上 i 的后继点.

输入带权邻接矩阵 $A = [a(i,j)]_{n\times n}$.

(1) 赋初值

对所有 i,j,d (i,j) = a (i,j);当 a (i,j) = 时,path (i,j) = 0,否则 path (i,j) = j;k = 1.

(2) 更新 d (i,j),path (k,j)

对所有 i,j,若 d(i,k) + d(k,j) > = d(i,j),则转(3);否则d(i,j) = d(i,k) + d(k,j),path(i,,j) = path(i,k),k = k + 1,继续执行(3).

(3) 重复(2)直到 k = n + 1.

例 3. 在 Matlab 中,用 Floyd 算法求图 10.6 所示的加权有向图中任意两点间的最短路径及长度.

解　加权有向图的存储结构采用带权邻接矩阵[a(i,j)].

Matlab 程序:

```
% Floyd's Algorithm
function [D, path] = floyd1(n)
n = size(a, 1);
% 设置 D 和 path 的初值.
D = a; path = zeros(n, n);
for i = 1:n
    for j = 1:n
    if D(i,j) ~ = inf;
        path(i,j) = j
      end
   end
end
% 做 n 次迭代,每次迭代均更新 D(i,j) 和 path(i,j)
for k = 1:n
    for i = 1:n
        for j = 1:n
            if D(j, k) + D(k, j) < D(i, j)
                D(i ,j) = D(i,k) + D(k,j)
                path(i, j) = path(i, k)
              end
        end
      end
  end
```

在 Matlab 命令窗口输入:

```
a = [0,50,inf,inf,inf;inf,0,inf,inf 80;inf,30 ,0 ,20 ,inf ;inf, inf,in],
0,70;65,inf ,100 ,inf ,0];
[D,path] = floyd1(a)
```

其运算结果为

$$
D = \begin{matrix} 0 & 50 & 230 & 250 & 130 \\ 145 & 0 & 180 & 200 & 80 \\ 155 & 30 & 0 & 20 & 90 \\ 135 & 185 & 170 & 0 & 70 \\ 65 & 115 & 100 & 120 & 0 \end{matrix}
$$

$$
path = \begin{matrix} 1 & 2 & 2 & 2 & 2 \\ 5 & 2 & 5 & 5 & 5 \\ 4 & 2 & 3 & 4 & 4 \\ 5 & 5 & 5 & 4 & 5 \\ 1 & 1 & 3 & 3 & 5. \end{matrix}
$$

因此，由最短距离矩阵 D 和最短路径矩阵 path，可以很容易地得出任意两点之间的最短路径及其长度. 如，顶点 1 到顶点 3 的最短路径长度 D (1，3) = 230，最短路径为：1 - 2 - 5 - 3. 这是因为 path (1，3) = 2 表示顶点 1 的后继顶点为 2，又 path (2，3) = 5 表示顶点 2 的后继顶点是 5. 依此类推，所以可知顶点 1 到顶点 3 的最短路径为：1 - 2 - 5 - 3.

§10.2 截断切割问题

问题背景：

某些工业部门（如贵重石材加工等）经常需要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），通常采用截断切割的加工方式，这里“截断切割”是指将物体沿某个切割平面分成两部分. 因此在一般情况下，需经过 6 次截断切割，分别截去原长方体的前、后、左、右、上、下 6 个方向多余的部分.

设水平切割单位面积的费用是垂直切割单位面积的 r 倍，且当先后两次垂直切割的平面不平行时，因调整刀具需额外费用 e.

显然，若截去各方向多余小块的先后顺序不同，则加工费用不同. 试设计一种确定最优加工次序的方法，此处的最优是指加工费用最少（由工艺要求，与水平工作台接触的长方体底面是事先指定的）.

用下列实例验证其切割方法：待加工长方体与成品长方体的长、宽、高分别为 10，14.5，19 和 3，2，4，二者左面、前面、底面之间的距离分别为：6，7，9（单位：cm），垂直切割费用为 1 元/cm^2，r 和 e 的数据有以下 4 组：

(1) r = 1，e = 0；　　(2) r = 1.5，e = 0；

(3) $r=8, e=0$；　　(4) $r=1.5, 2\leqslant e\leqslant 15$.

10.2.1　问题分析及模型的建立

1. 问题分析

这是一个最优化问题,求切割顺序,使加工费用最低.

决策变量为切割顺序,用 $X=(x_1,x_2,\cdots,x_6)$ 来表示切割顺序,x_i 表示第 i 次切割,可以取 1,2,…,6,分别表示左、右、前、后、上、下方向的切割,$x_1,x_2,\cdots,x_6$ 互不相同,可以取 1,2,…,6 的任意全排列.

目标函数:加工费用,由切割费用和刀具调整费用构成.

该问题已经比较明确,问题中的已知条件有:

(1) 待加工长方体与成品长方体对应表面平行;

(2) 切割费用与切割面的面积成正比,具体地说就是垂直切割费用为 1 元/cm^2,水平切割费用为 r 元/cm^2,且仅当先后两次垂直切割的切割面不平行时,才需调整刀具,调整刀具的费用为 e;

(3) 水平工作台接触的长方体底面是事先指定的;

(4) 不考虑第一次切割前的刀具调整费用.

2. 建立模型

这里用图论建立模型并求其解. 下面分两种情况分别讨论:

情形一:$e=0$ 时,构造一个赋权有向图 $G=(V,E)$.

把整个加工过程中长方体工件的所有可能状态作为顶点,若长方体工件可以由顶点 u 所代表的状态,经一次切割变为顶点 v 所代表的状态,则作一条从 u 到 v 的有向边,同时把此次切割的费用作为该边的权. 可以用 0,1 组成的 6 维数组来代表各种状态或顶点,左、右、前、后、上、下侧多余的块已被切除,分别对应这个数组的第 1,2,…,6 分量取 1,还未被切去时,相应分量取 0. 例如,(0,0,0,0,0,0) 代表原长方体,(1,1,1,1,1,1) 代表最终的成品长方体,(0,1,1,0,0,1) 就代表原长方体右侧、前面和下面的多余块被截去后余下的新尺寸长方体,顶点集

$$V=\{(a_1,a_2,a_3,a_4,a_5,a_i)\mid a_i=0\text{ 或 }1, i=1,2,\cdots,6\}$$

共有 $2^6=64$ 个顶点.

在 G 中的任何一条从顶点 (0,0,0,0,0,0) 到顶点 (1,1,1,1,1,1) 的有向路径代表一连串切割施于原长方体,长方体的状态变化过程,对应的一连串切割就是一种加工方式,该有向路径的权便是加工费用. 反之,任何一种加工方式都对应于 G 中的一条有向路径. 求最小费用的加工方式就转化为求赋权有向图 $G=(V,E)$ 中从顶点 (0,0,0,0,0,0) 到顶点 (1,1,1,1,1,1) 的最短路径.

因此,只考虑平行切割边距大者先切的加工方式,从而在加工过程中工件的状态数减少. 对两平行切割只考虑边距大者先切所导致的长方体工件的状态,可以用 0,1,2 组成的 3 维数组来代表各种状态,(x,y,z) 表示左右、前后、上下分别被切去 x,y,z 刀. 例如,(0,0,0) 代表原长方体,(2,2,2) 代表成品长方体,(0,2,1) 就代表原长方体的前面、后面已被切去,上下侧厚的那一块已被切去余下的新尺寸长方体. 顶点集

$$V' = \{(a_1, a_2, a_3) \mid a_i = 0,1 \text{ 或 } 2, i = 1,2,\cdots,6\}$$

共有 27 个顶点,这些顶点代表待加工长方体在加工过程中的 27 种状态. 若待加工长方体可以由顶点 u 所代表的状态,经一次切割变为顶点 v 所代表的状态,则作一条从 u 到 v 的有向边,同时把该次切割的费用作为该边的权. 如此得到一个规模更小的赋权有向图 G' = (V',E').

问题转化为求赋权有向图 G' = (V',E') 中从顶点 (0,0,0) 到顶点 (2,2,2) 的最短路径,可以用 Dijkstra 算法求解.

若要求出所有的最优加工方式,则需求出从顶点 (0,0,0) 到顶点 (2,2,2) 的全部最短路径,可以在 Dijkstra 算法中加入一些记录和语句.

10.2.2 模型求解所需的知识点及其在 Matlab 中的实现方法

1. 求向量或矩阵的长度 length ()

命令形式:length (a)

功能:如果 a 是向量,则返回向量的长度;如果 a 是矩阵,则返回矩阵行数 m 和列数 n 中的较大值,即返回值为 max (m, n).

2. 求矩阵或向量的范数 norm

命令形式:norm (a, 2)

功能:求矩阵 a 的 2 范数.

3. 文件写入命令 fprintf

命令形式:fprintf (fp,format, var)

功能:把数据 var 按 format 的形式写入文件 fp 中. 从文件中读取某种数据的命令为:fscanf. 还有相类似的命令:fread,fwrite,fget 等.

10.2.3 模型求解在 Matlab 中的实现

求解 e = 0 时的 Matlab 程序如下:

```
%   jieduan_path_e = 0
%   weight (u, v, vertices, a0, a1, d1, r) : weight of arc (u,v)
function   cutpathe0 (a0, a1, d1, r)
```

```
        vertices = [];
        for i =0:2
            for j = 0:2
                for k = 0:2
                    vertices (size (vertices ,1) +1 , :) = [i j k];
                end, end , end
% 用修正的 Dijkstra 算法求顶点 1 到其余各顶点的最短路径及最短距离.
n = size (vertices, 1) ; s = [];
s (length (s) +1) = 1 ;
sb = [] ;  lasb = [] ;
label 91) = 0 ; u = 1 ; father = zeros (n ,n) ;
for k =2 :n
    label (k) = inf ;
    sb (length (sb) +1) = k ;
    labsb ( length (sb) +1) = inf ;
end
while length (sb) ~ = 0
        for i =1 :length (sb) ;
            v = sb (i) ;
            w = weight (u, v, vertices, a0, a1, d1, r) ;
            if label (v) > label (u) +w
                label (v) = label (u) + w ;
                father (v, 1) = u ; father (v, 2, n) = 0 ;
             elseif  label (v) == label (u) +w
                    father (v, nnz (father ( v, :)) +1 ) = u ;
            end
                labsb ( i ) =label (v) ;
            end
            [y, i] = min (labsb) ; u = sb (i) ;
            s (length (s) + 1 ) = sb ( i ) ;
            sb ( i ) = [] ;
            labsb ( i ) = [] ;
        end
```

```
% 输出顶点 1 (即 (0,0,0)) 到顶点 27 (即 (2,2,2)) 的全部最短路径.
    mark = ones (1, n) ;    again = 1 ;
    fprintf (1, 'cost:%fhn', label (27)) ;
    while  again ==1
            again = 0 ;
            v = 27 ;
            fprintf (1, 'path  : (2 2 2) - - ') ;
            flag = 0 ; num = 0 ; again = 0;
            while father (v, mark (v)) ~ = 0
                u = v ;
                v = father (u, mark (u)) ;
                if  father (u, mark (u) + 1) ~ = 0 & num == 0
                    again = 1 ; num = 1;
                    mark (u) = mark (u) +1 ;
                end
                fprintf (1,' (%d  %d  %d) - - ',vertices (v , :)) ;
            end
        fprintf (1, ' hn') ;
        end
function  w = weight (u , v, vertices, a0, a1, d1, r)
    w = 0;
    v1 = vertices (u, :) ;
    v2 = vertices (v, :) ;
    d2 = a0 - a1 - d1 ;   a = a0 ;
    temp1 = norm (v2 - v1, 2 ) ;temp2 = ones (1,3). * (v2 - v1) ;
    if  temp1 ~ = 1 | temp2 ~ = 1
        w = inf ;
    else
        for k =1 :3
            if  v1 (k) >0
                switch v1 (k)
                case 1
                    a (k) = a0 (k) - max (d1 (k), d2 (k)) ;
                case 2
```

```
                    a (k) = a0 (k) - d1 (k) - d2 (k) ;
        end , end, end
        k = find (v2, - v1) ;
        if k ==3
            w = a (1) * a (2) * r ;
        else
            v= a (1) * a (2) * a (3) /a (i) ;
        end , end
```

在 Matlab 窗口中键入：

```
a0 = [10, 14.5 , 19] ;
a1 = [3, 2, 4] ;
d1 = [6, 7, 9] ; r = 1.5 ;
cutpather0 (a0, a1, d1, r) ;
```

其运行结果为

```
cost :437. 500
path : (2, 2, 2) - (1,2,2) - (1,2,1) - (1,1,1) - (0,1,1) -
       (0,1,0) - (0,0,0)
path :2,2,2) - (1,2,2) - (1,2,1) - (1,1,1) - (1,1,0) - (0,0,0)
```

即得到从 (0,0,0) 到 (2,2,2) 的最短路径有两条，其费用为 437.5 元，说明最优加工方式有两种，由于平行切割中厚块分别为左面、前面和下面. 因此，两种最短路径对应的最优加工方式为：前下左后上右，前左下后上右，其加工费用为 437.5 元.

情形二：$e \neq 0$ 时.

此时，两次切割之间可能需要调整刀具，其加工费用与当前最后一次垂直切割的方向有关，上述情形一的结论依然成立，可以如下构造一个赋权有向图 $G'' = (V'', E'')$，把数次切割以后待加工物体所处的状态和这些切割中最后一次垂直切割的方向合在一起作为一个顶点，若待加工长方体可以由顶点 u 所代表的状态，经过一次切割变为顶点 v 所代表的状态，则作一条从 u 到 v 的有向边，同时把该次加工的费用（若需调整刀具，则包括调整刀具的费用）作为该边的权. 一个顶点用一个四维数组表示，前三个分量取值 0，1 或 2 表示状态，第四个分量为 0，1 或 2，分别代表从开始切割到该状态为止，没有垂直切割，最后一次垂直切割的方向为左右或前后，例如 (1,2,0,1) 代表原长方体前后两多余块和左右之厚块，且最后一次垂直切割为左右方向，令

$$V_1 = \{(0,0,z,0) \mid z=0,1,2\}$$

$$V_2 = \{(x,y,z,1) \mid x=1,2;y,z=0,1,2\}$$

$$V_3 = \{(x,y,z,2) \mid x,z=0,1,2;y=1,2\}.$$

顶点集 $V''=V_1\cup V_2\cup V_3$，共 39 个顶点. 问题就转化为求赋权有向图 $G''=(V'',E'')$ 中从顶点 (0,0,0,0) 到顶点 (2,2,2,1) 或 (2,2,2,2) 的所有最短路径.

$e\neq 0$ 时试验结果：

当 $a_0=[10\quad 14.5\quad 19]$，$a_1=[3\quad 2\quad 4]$，$d_1=[6\quad 7\quad 9]$ 时，用前面所介绍的方法求出下述四种情况的最优加工方式：

(1) $r=1,e=0$；　　(2) $r=1.5,e=0$；

(3) $r=8,e=0$；　　(4) $r=1.5,2\leqslant e\leqslant 15$.

所得的结果如表 10.2 所示.

表 10.2

r	e	最优切割方式	最少费用
1	0	531642，536142	374
1.5	0	351462，315462	437.5
8	0	314526	540.5
1.5	[2，2.5]	315362，351462	437.5 + 3e
1.5	2.5	315462，351462，354162	445
1.5	(2.5，15)	354162	442.5

习　题　11

1. 某石油公司在墨西哥湾拥有几个钻井平台，每个平台开采出的石油需要经过路易斯安娜州的炼油场. 要在平台与路易斯安娜州之间建造一个管道网. 试问管道如何设计，才能使建造费用最低. 图 10.7 中，顶点代表钻井平台及炼油场，边表示其两点之间可以铺设管道，边上的权代表该管道的建设费用.

2. 某公司在 6 个城市 C_1,C_2,C_3,C_4,C_5,C_6 都有分公司，公司成员经常往来于这 6 个城市之间，已知从 C_i 到 C_j 的直达航班票价由矩阵 A 的第 i 行、第 j 列给出（　表示无直达航班），试设计一张任意两城市之间的最廉价路线表.

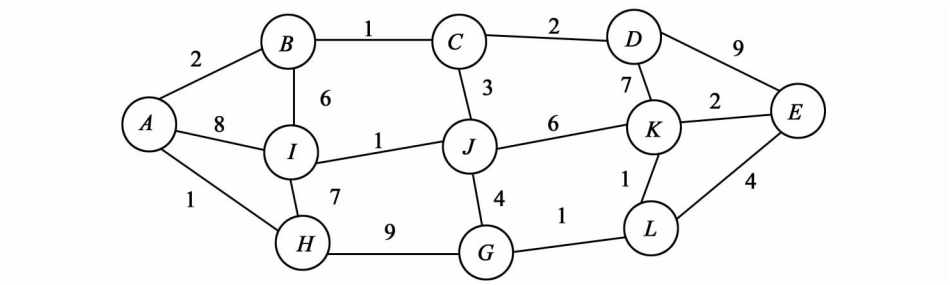


图 10.7

$$A=\begin{bmatrix} 0 & 50 & & 40 & 25 & 10 \\ 50 & 0 & 15 & 20 & & 25 \\ & 15 & 0 & 10 & 20 & \\ 40 & 20 & 10 & 0 & 10 & 25 \\ 25 & & 20 & 10 & 0 & 55 \\ 10 & 25 & & 25 & 55 & 0 \end{bmatrix}$$

提示：该问题实际上就是求以矩阵 A 为带权邻接矩阵的加权图中，任意两点之间的最短路径及其长度的问题.

3. 城市电信局有很多业务，如收费、营业、110 等，希望在全市实现计算机联网服务，共享各种资源. 人们主要关心的问题是：用数据通讯线路把一组站点连接起来，而不允许通讯线路在非站点处连接，试问如何连接可以使通讯线路的花费最少？任意两个站点可以通过若干中介站点取得联系. 图 10.8 是在站点比较少的情况下的一个模拟图，顶点为站点，边为连接两站点之间的通讯线路，边的权为费用. 试问如何连接这几个站点，而使得通讯线路的花费最少？

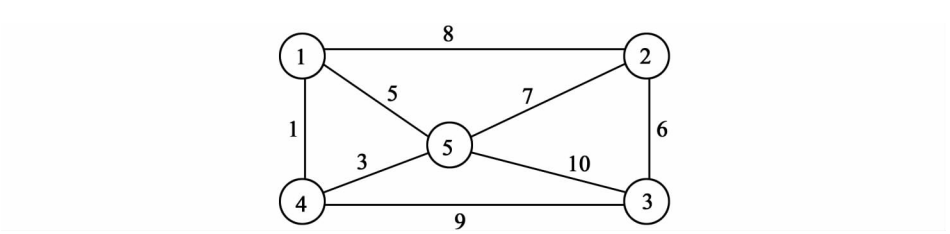


图 10.8

第十一章　最优化方法模型在 Matlab 中的求解

§11.1　线性规划和非线性规划及其在 Matlab 中的求解方法

11.1.1　线性规划及在 Matlab 中的解法

线性规划是运筹学的一个重要分支，在科学实践中有着广泛的应用，不仅许多实际问题属于线性规划问题，而且运筹学其他分支中的一些问题也可以转化为线性规划问题来计算，因此，线性规划在最优化学科中占有重要的地位.

首先来看几个例子：

例 1. 试验问题

问题 1（任务分配问题）某车间有甲、乙、丙三台车床可以用于加工三种零件，这三台车床可以用于工作的最多时间分别为 700h、800h 和 900h，需要加工的三种零件的数量分别为 300、400 和 500，不同车床加工不同的零件所用的时间数和费用如表 11.1 所示，试问：在完成任务的前提下，如何分配加工任务才能使得加工费用最低？

表 11.1　加工不同零件所用的时间数和费用表

车床名称	加工单位零件所需时数			加工单位零件所需的费用			可用于工作的时数
	零件 1	零件 2	零件 3	零件 1	零件 2	零件 3	
甲	0.6	0.5	0.5	7	8	8	700
乙	0.4	0.7	0.5	8	7	8	800
丙	0.8	0.4	0.6	7	9	8	900

问题 2（人员安排问题）某城市的巡逻大队要求每天的各个时间段都有一定数量的警员值班，以便随时处理突发事件，每人连续工作 6h，中间不休息. 表 11.2 是一天 8 个班次所需值班警员的人数情况统计，现在在不考虑时间段中警员上班和下班的情况下，巡逻大队至少需要多少警员才能满足值班需要？

表 11.2　　巡逻时间和巡逻人数表

班次	时间段	人数	班次	时间段	人数
1	6 00 ~ 9 00	70	5	18 00 ~ 21 00	80
2	9 00 ~ 12 00	80	6	21 00 ~ 24 00	100
3	12 00 ~ 15 00	65	7	24 00 ~ 3 00	120
4	15 00 ~ 18 00	90	8	3 00 ~ 6 00	90

这两个问题虽然内容不同，但是它们涉及的数学知识是相似的，都是在一定条件下求某些问题的最大值或最小值.

下面对这两个问题进行分析，并建立其求解的数学模型.

问题 1 的数学模型：可以设分配给车床甲加工三种零件的数量分别为 x_1，x_2，x_3，分配给车床乙加工三种零件的数量分别为 x_4，x_5，x_6，分配给车床丙加工三种零件的数量分别为 x_7，x_8，x_9，则根据问题的实际要求可得以下数学模型

$$\min Z = 7x_1 + 8x_2 + 8x_3 + 8x_4 + 7x_5 + 8x_6 + 7x_7 + 9x_8 + 8x_9$$

$$\text{s. t.}\begin{cases} x_1 + x_4 + x_7 = 300 \\ x_2 + x_5 + x_8 = 400 \\ x_3 + x_6 + x_9 = 500 \\ 0.6x_1 + 0.5x_2 + 0.5x_3 \leqslant 700 \\ 0.4x_4 + 0.7x_5 + 0.5x_6 \leqslant 800 \\ 0.8x_7 + 0.4x_8 + 0.6x_9 \leqslant 900 \\ x_i \geqslant 0, i = 1,2,\cdots,9 \end{cases}$$

问题 2 的数学模型：可设第 i 个班次开始上班的警员数为 x_i，那么根据题意可得以下数学模型

$$\min Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$$

$$\text{s. t.}\begin{cases} x_1 + x_8 \geqslant 70 \\ x_1 + x_2 \geqslant 80 \\ x_2 + x_3 \geqslant 65 \\ x_3 + x_4 \geqslant 90 \\ x_4 + x_5 \geqslant 80 \\ x_5 + x_6 \geqslant 100 \\ x_6 + x_7 \geqslant 120 \\ x_7 + x_8 \geqslant 90 \\ x_i \geqslant 0, i = 1,2,\cdots,8 \end{cases}$$

上面两个问题的具体含义虽然不同,但它们有共同的特点:求一组非负变量 $x_1, x_2, \cdots, x_n$,使得关于这组变量的线性函数 $f(x_1, x_2, \cdots, x_n)$ 在一定的约束条件(关于变量的线性等式或不等式)下取得最大值或最小值. 这类问题可以用下面一般的数学模型表示

$$\min Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

$$\text{s. t.}\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geqslant (\leqslant) b_1 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geqslant (\leqslant) b_m \\ x_i \geqslant 0, j = 1,2,\cdots,n \end{cases} \tag{1}$$

这类问题就是线性规划问题,也就是 LP 问题. 一般可以写成下面的矩阵形式

$$\min Z = C^Tx$$

$$\text{s. t.}\begin{cases} Ax \geqslant (\leqslant) b \\ x \geqslant 0 \end{cases} \tag{2}$$

这里矩阵 A 称为约束矩阵,$x_1, x_2, \cdots, x_n$ 称为决策变量,而函数 $f(x_1, x_2, \cdots, x_n)$ 称为目标函数.

通常在解线性规划问题时都是先将其一般形式化为下面的标准形式

$$\max Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

$$\text{s. t.}\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\ x_i \geqslant 0, j = 1,2,\cdots,n \end{cases} \tag{3}$$

写成矩阵形式就是

$$\max Z = C^T x$$
$$\text{s. t.} \begin{cases} Ax = b \\ x \geqslant 0 \end{cases} \tag{4}$$

注意：也可以将标准形式的线性规划模型定义为求目标函数的最小值，二者只是形式上的不同，本质上是一样的.

将一般的线性规划问题化为标准形式的线性规划问题，分下面三步完成：

(1) 目标函数的转换：标准形式的线性规划问题是求目标函数的最大值，如果实际问题要求目标函数的最小值，则可以通过给目标函数乘以 - 1 来转换.

(2) 约束条件的转换：标准形式的线性规划问题的约束条件是一些线性等式，如果实际问题给出的约束条件是线性不等式

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant (\leqslant) b_i$$

则可以通过引入松弛变量 $x_{n+i} \geqslant 0$，将不等式约束化为等式约束

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j - x_{n+i} = b_i \\ x_{n+i} \geqslant 0 \end{cases}$$

(3) 变量转换：标准形式的线性规划问题中的变量都是非负的，如果某个变量的约束条件为 $x_j \geqslant l_j$ 或 $x_j \leqslant l_j$，则可以令 $y_j = x_j - l_j$ 或令 $y_j = l_j - x_j$，这样 y_j 变为非负变量；如果某个变量 x_j 无非负限制，则可以令

$$\begin{cases} x_j = x'_j - x''_j \\ x'_j \geqslant 0, x''_j \geqslant 0 \end{cases}$$

代入原问题，将自由变量替换掉.

例 2. 将下面线性规划问题化为标准形式

$$\min Z = -2x_1 + x_2 + 3x_3$$
$$\text{s. t.} \begin{cases} 5x_1 + x_2 + x_3 \leqslant 7 \\ x_1 - x_2 - 4x_3 \geqslant 2 \\ -3x_1 + x_2 + 2x_3 = -5 \\ x_1, x_2 \geqslant 0, x_3 \text{ 为自由变量}. \end{cases}$$

解　引入松弛变量 x_4, x_5，令 $x_3 = x'_3 - x''_3$，代入方程后得

$$\max Z' = 2x_1 - x_2 - 3x'_3 + 3x''_3$$

$$\text{s. t.}\begin{cases}5x_1+x_2+x_3'-x_3''+x_4=7\\x_1-x_2-4x_3'+4x_3''-x_5=2\\-3x_1+x_2+2x_3'-2x_3''=-5\\x_1,x_2,x_3',x_3'',x_4,x_5\geqslant 0\end{cases}$$

上式即为所求问题的标准形式.

对于线性规划问题,可以从理论上进行求解,在 Matlab 中,也有现成的函数可以用来求解线性规划问题,下面分别介绍它们的求解方法. 首先从理论上探讨线性规划问题的解法.

1. 理论解法

首先给出线性规划问题的几个定义和性质.

性质 1. 线性规划 (LP) 问题的可行域 $D=\{x\mid Ax=b,x>0\}$ 是凸多面体 (凸集). 特别地,当目标函数是二元函数时,可行域是凸多边形.

定义 1. 设 A 是秩为 m 的 $m\times n$ 阶矩阵,A 的 m 个线性无关的列构成的子矩阵 A_B 称为 LP 问题的一个基,A_B 的列向量称为基列,相应于基列的变量称为基变量.

定义 2. 设 A 是秩为 m 的 $m\times n$ 阶矩阵,由

$$Ax=(A_B,A_N)\begin{pmatrix}x_B\\x_N\end{pmatrix}=A_Bx_B+A_Nx_N=b$$

可得

$$x_b=A_B^{-1}(b-A_Nx_N)$$

那么

$$x=\begin{pmatrix}x_B\\x_N\end{pmatrix}=\begin{pmatrix}A_B^{-1}(b-A_Nx_N)\\x_N\end{pmatrix}$$

为 $Ax=b$ 的一个解,若令 $x_N=0$,则

$$x'=\begin{pmatrix}A_B^{-1}b\\0\end{pmatrix}$$

称为 LP 问题的一个基本解,若 $A_B^{-1}b\geqslant 0$,则称 x' 为 LP 问题的一个基本可行解,简称基可行解,这时的 A_B 称为可行基.

性质 2. x 是 LP 问题可行域 D 的顶点等价于 x 是 LP 问题的基本可行解.

性质 3. 如果 LP 问题存在最优解 x^* 则最优解 x^* 一定能在可行域的顶点中取到.

由上述性质可以看出,若 LP 问题存在最优解,则其解一定是 LP 问题的一个基本可行解. 由于线性规划问题的基本可行解至多有 C_n^m 个,所以当 C_n^m 较小

时,可以通过求所有的基本可行解,然后再从这些可行解中取使目标函数值最优的一个解,即为线性问题的最优解. 下面由一个例子来看该问题的解法.

例 3. 解下列线性规划问题

$$\max Z = x_1 + 3x_2$$

$$\text{s. t.} \begin{cases} x_1 + 2x_2 \leqslant 8 \\ x_2 \leqslant 2 \\ x_1, x_2 \geqslant 0. \end{cases}$$

解　首先引入松弛变量 x_3, x_4,将上述线性规划问题的一般形式化为标准形式

$$\max Z = x_1 + 3x_2$$

$$\text{s. t.} \begin{cases} x_1 + 2x_2 + x_3 = 8 \\ x_2 + x_4 = 2 \\ x_j \geqslant 0, j = 1,2,3,4 \end{cases}$$

现在求其基本可行解

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [P_1, P_2, P_3, P_4]$$

由于每确定一个基本矩阵 B,就能解得一个基本解,下面分别选择不同的基 B,求出所有基本解,再从中找出基本可行解.

令 $B = [P_1, P_2] = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$,解得基本解为 $x^{(1)} = (x_1, x_2, x_3, x_4)^T = (4,2,0,0)^T$;

令 $B = [P_1, P_4] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$,解得基本解为 $x^{(2)} = (x_1, x_2, x_3, x_4)^T = (8,0,0,2)^T$;

令 $B = [P_2, P_3] = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$,解得基本解为 $x^{(3)} = (x_1, x_2, x_3, x_4)^T = (0,2,4,0)^T$;

令 $B = [P_2, P_4] = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$,解得基本解为 $x^{(4)} = (x_1, x_2, x_3, x_4)^T = (0,4,0,-2)^T$;

令 $B = [P_3, P_4] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$,解得基本解为 $x^{(5)} = (x_1, x_2, x_3, x_4)^T = (0,0,8,2)^T$.

以上对于所有的基都进行了计算,得到 5 个基本解,其中 $x^{(1)}, x^{(2)}, x^{(3)}, x^{(5)}$ 是基本可行解,$x^{(4)}$ 则不是,因为 $x^{(4)}$ 的第 4 个分量是负数,然后将上述 4 个基本可行解代入目标函数进行比较,易知 $x^{(1)}$ 为问题的最优点,此时目标函数的最优值为 10.

通过例 3,可以得到线性规划问题理论解法的一般步骤为:

(1) 将线性规划问题的模型标准化,找出线性规划问题的所有基矩阵;

(2) 对找出的每一个基矩阵求得一个基本解;

(3) 在所有的基本解中找出基本可行解;

(4) 在所有的基本可行解中找出最优解;

(5) 算出最优解处目标函数的最优值.

当线性规划问题是二元规划时,只要有最优解;当线性规划问题是二元以上且规模较小时,只要有最优解,用理论方法也可以得到其最优解. 但是当 C_n^m 较大时,采用理论解法计算量会很大,显然是不可取的,此时就要用计算机来编程求解. 下面介绍 Matlab 中求解线性规划问题的方法.

2. 软件解法

Matlab 优化工具箱中提供了两个专门解线性规划问题的命令 lp 和 linprog,下面介绍这两个命令的用法.

函数 lp 的几种常用格式:

(1) x = lp (c, a, b)

该命令用来解规划问题

$$\min Z = cx$$
$$\text{s.t.} \quad ax \leqslant b$$

(2) x = lp (c, a, b, vlb, vub) 该命令用来解规划问题

$$\min Z = cx$$
$$\text{s.t.}\ ax \leqslant b$$
$$vlb \leqslant x \leqslant vub$$

这里 vlb 和 vub 表示变量分量的上、下限.

(3) x = lp (c, a, b, vlb, vub, x0)

该命令用来解规划问题 (1) ~ (4),其中 x0 是给出的初始点向量.

(4) x = lp (c, a, b, vlb, vub, x0, n)

该命令用来解规划问题 (1) ~ (4),其中 x0 是给出的初始点向量,n 表示约束条件 ax≤b 中的前 n 个约束是等式约束.

下面是例 1 中问题 2 在 Matlab 中的求解程序:

```
% JingYuanXL. m
c = [1, 1, 1, 1, 1, 1, 1, 1];
a = [-1, 0, 0, 0, 0, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 0 ;
0, -1, -1, 0, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0 ;
0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0 ;
0, 0, 0, 0, 0, -1, -1, 0, 0, 0,0, 0, 0, 0, -1, -1] ;
```

```
b = [-70 ;-80 ;-65;-90 ;-80;-100;-120;-90];
vlb = [0;0] ;
vub = [] ;
x = lp (c, a, b, vlb, vub) ;
x
    minZ = c*x
```

其程序运行的结果为

```
    x = 40.0000
        40.0000
        45.0000
        45.0000
        37.5000
        62.5000
        57.5000
        32.5000
minZ = 360
```

由此可知,至少需要的警员的人数是

$$40+40+45+45+38+63+58+33=362$$

函数 linprog 的常用格式：

(1) x = linprog (c, a, b)

(2) x = linprog (c, a, b, aeq, beq)

(3) x = linprog (c, a, b, aeq, beq, lb, ub)

(4) x = linprog (c, a, b, aeq, beq, lb, ub, x0)

(5) x = linprog (c, a, b, aeq, beq, lb, ub, x0, options)

linprog 的用法与 lp 的用法相同,命令中各参数的含义如下：

函数 linprog 是用来解下面形式的线性规划问题的

$$\min Z = cx$$

$$\text{s.t.}\begin{cases} ax \leqslant b \\ aeq \cdot x = beq \\ lb \leqslant x \leqslant up \end{cases}$$

在命令 linprog 中,输入参数 c 是赋权向量;x 是决策向量;a 是不等式约束条件的系数矩阵,b 是不等式约束条件的右端常数向量;lb 与 ub 分别为变量取值范围的下界和上界;x0 是设定的初始值;当线性规划问题的规模很大时,一般不用初始值选项;格式 (5) 中 options 选项是用来指定优化参数. 输出参数 x 是

LP 问题的最优解.

例 1 中问题 1 在 Matlab 中的求解程序：

```
% RenWuFP. m
c = [7, 8, 8,8,7, 8, 7, 9, 8];
aeq = [1, 0, 0, 1, 0, 0, 1, 0, 0 ;
       0, 1, 0, 0, 1, 0, 0, 1, 0 ;
       0, 0, 1, 0, 0, 1, 0, 0, 1 ] ;
beq = [300 ;400 ;500 ],
a = [0. 6, 0. 5, 0. 5, 0, 0, 0, 0, 0, 0 ;
0, 0, 0, 0. 4 , 0. 7, 0. 5, 0, 0, 0 ;
0, 0, 0, 0, 0, 0, 0. 8, 0. 4, 0. 6 ];
b = [ 700 ;800 ;  900] ;
lb = zeros (9, 1) ;
x = linprog (c, a, b, aeq, beq, lb)
```

其程序运行的结果为

```
x = 98. 2599
    0. 0000
    145. 7326
    0. 0000
    400. 0000
    145. 9377
    201. 7401
    0. 0000
    208. 3298.
```

11. 1. 2 非线性规划及在 Matlab 中的求解方法

首先还是先看一个非线性规划的例子：

例 4. 建筑材料场位置的选择和供货量. 某建筑公司有 6 个建筑工地要开工，每个工地的位置用 a, b 表示（单位：km），每个工地每天所需水泥的数量用 d 表示，工地位置 a，b 和水泥日用量 d 如表 11. 3 所示. 现有两个供货材料场位于 A(5, 1) 和 B (2, 7)，日存储量各有 20t. 假设从材料场到工地之间均有直线道路相连，试确定每天的供应计划，即从 A，B 两材料场分别向各工地运送多少吨水泥，使总的吨千米数最小.

为了进一步减少吨千米数，准备不用这两个临时材料场，改建两个新的材料

场,新材料场的日存储量仍各为 20t,那么这两个新的材料场应建在何处,节省的吨千米数有多大?

表 11.3　工地的位置及水泥日用量

	1	2	3	4	5	6
a	1.25	8.75	0.5	5.75	3	7.25
b	1.25	0.75	4.75	5	6.5	7.75
c	3	5	4	7	6	11

1. 问题分析及模型的建立

记第 i 个工地的位置为 (a_i, b_i),该工地每天的水泥日用量为 d_i,其中 $i=1,2,\cdots,6$;设第 j 个材料场的位置为 (x_j, y_j),日存储量为 m_j,其中 $j=1,2$;设从材料场 j 向工地 i 的运送量为 p_{ij}. 则从材料场向各个工地运送材料时的总吨千米数 f 为

$$f = \sum_{j=1}^{2} \sum_{i=1}^{6} p_{ij}\sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}$$

要使总的吨千米数最小,也就是求 min f. 所以该问题的目标函数是

$$\min f = \min\left(\sum_{j=1}^{2} \sum_{i=1}^{6} p_{ij}\sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}\right) \tag{11.1}$$

因为每天必须使各工地的日用量得到满足,所以

$$\sum_{i=1}^{2} c_{ij} = d_i, \quad i = 1,\cdots,6 \tag{11.2}$$

而各材料场的运送量不能超过它的日存储量,所以

$$\sum_{i=1}^{6} p_{ij} \leqslant m_j, \quad j = 1,2 \tag{11.3}$$

由上述分析可知,当使用临时材料场,求总的吨千米数的最小值时,决策变量为 p_{ij};在求改建的新材料场的最佳位置时,决策变量为 p_{ij}和 x_j, y_j.

所以最终问题归结为在约束条件(11.2)、(11.3)下,求目标函数(11.1)的最小值. 由于目标函数 f 对 x_j, y_j 是非线性的,所以在新建材料场时是一个非线性规划模型.

非线性规划有很多种解法,如可行方向法、罚函数法、梯度投影法等,Matlab 工具箱中使用的是逐步二次规划法,也称为 SQP 方法.

2. 逐步二次规划法的基本原理

用 SQP 方法解带约束的非线性规划问题的基本原理是:构造拉格朗日函数

$$L(x,\mu,\lambda) = f(x) + \sum_{i=1}^{m}\mu_i h_i(x) + \sum_{i=1}^{m}\lambda_j g_j(x)$$

用二次函数近似 $L(x,\mu,\lambda)$ 后化为二次规划问题（即 QP 问题），然后求解一系列如下形式的二次规划问题

$$\min \frac{1}{2}d^T G_k d + \nabla f(x_k)^T d$$
$$\text{s.t.}\begin{cases}\nabla h_i(x_k)^T d + h_i(x_k) = 0, & i = 1,\cdots,m \\ \nabla g_j(x_k)^T d + g_j(x_k) \leqslant 0, & j = 1,\cdots,l\end{cases} \tag{11.4}$$

其中 x_k 表示第 k 次迭代的初始点，G_k 是 $L(x_k,\mu,\lambda)$ 的海赛矩阵 $\nabla^2 L$ 的近似，由条件（11.4）得到的最优解 d_k 取做第 k 次迭代的搜索方向，新的迭代点为 $x_{k+1} = x_k + \alpha_k d_k$，其中 α_k 是按一定搜索准则得到的步长. 一般地，SQP 求解时包括三个主要部分：

（1）求解二次规划子问题（11.4）；

（2）用线性搜索计算步长 α_k；

（3）确定矩阵 G_k 的迭代公式.

上面已经讨论了第一个问题的解决方法；解决第二个问题是构造一个既包括目标函数，又包括约束条件信息的指标函数

$$P(x,r,s) = f(x) + \sum_{i=1}^{m} r_i h_i(x) + \sum_{j=1}^{l} s_j \max(0, g_j(x))$$

选择步长 α_k 使指标函数充分降低，其中 r_i, s_j 是解 QP 问题得到的拉格朗日乘子 μ_i, λ_j 有关的罚因子，可以自动调整；解决第三个问题的方法是设法利用无约束优化中拟牛顿法公式

$$G_{k+1} = G_k + \frac{\nabla f^k (\nabla f^k)^T}{(\nabla f^k)^T \nabla x^k} - \frac{G^k \nabla x^k (\nabla x^k)^T G^k}{(\nabla x^k)^T G^k \nabla x^k}$$

并且利用上述指标函数中包含的约束条件对上述公式中的 ∇f^k 进行修正.

3. 用 Matlab 优化工具箱解带约束的非线性规划问题

Matlab 中求解 SQP 问题的函数为 constr，该命令的几种常用形式如下：

```
x = constr('fun', x0)
x = constr('fun', x0, ops)
x = constr('fun', x0, ops, v1, v2, 'grad')
x = constr('fun', x0, ops, v1, v2, 'grad', p1, p2, …)
[x, ops] = constr('fun', x0)
```

该函数的用法在前面已经详细介绍过，在使用该函数时要注意：fun. m 文件中要同时给出目标函数 f 和约束函数 g，形式为 [f, g] = fun(x)；如果命令中使

用参数 grad,则在 grad. m 文件中要用分析的方法同时给出目标函数 f 和约束函数 g 的梯度,形式为 [df, dg] = grad (x) ;

下面介绍例 4 在 Matlab 中的求解方法:

(1) 使用两个临时材料场 A (5,1) , B (2,7) . 求从材料场 j 向工地 i 的运送量 p_{ij},在各工地用量必须满足且各材料场运送量不超过日存储量的条件下,使总吨千米数最小. 这是一个线性规划问题,前述已经建立了该问题线性规划的模型. 直接套用求解线性规划的函数 lp () 求解即可,其计算结果如表 11.4 所示.

表 11.4

i	1	2	3	4	5	6
c_{i1} (材料场 A)	3	5	0	7	0	1
c_{i2} (材料场 B)	0	00	4	0	6	10

运送总吨千米数为 136. 2t · km.

(2) 改建两个新材料场,要同时确定材料场的位置 (x_j, y_j) 和运送量 p_{ij},其中 i = 1,2,…,6,j = 1,2,在同样约束条件下要使总吨千米数最小,这是一个非线性规划问题,该模型在 Matlab 中的求解程序如下:

例 4 中新材料场选址问题的实现程序:

```
% bestnewliaochang. m
x0 = [zeros (1,12) , 5 , 1 , 2 , 7] ;% 取原供料场位置为新供料场位置的初始值.
v1 = zeros (1 , 12) ;
op (13) = 6 ;op (14) = 2000 ;
[x, op] = constr ('liaochang' , x0 , op , v1)
% 目标函数和约束函数:
% liaochang. m
function [f, g] = liaochang (x)
a = [1. 25 , 8. 75 , 0. 5 , 5. 75 , 3 , 7. 25] ;
b = [1. 25 , 0. 75 , 4. 75 , 5 , 6. 5 , 7. 75] ;
d = [3 , 5 , 3 , 7 , 6 , 11] ;
e = [20 , 20] ;
f1 = 0 ;
for i = 1:6
   s ( i ) = sqrt ( (x (13) - a ( i ) ) 2 + (x (14) - b ( i ) ) 2) ;
```

```
    f1 = s(i)*x(i) + f1;
end
f2 = 0;
for i = 7:12
    s(i) = sqrt((x(15) - a(i - 6))2 + (x(16) - b(i - 6))2);
    f2 = s(i)*x(i) + f2;
end
f = f1 + f2;
for j = 1:6
g(j) = x(j) + x(j+6) - d(j);%供料必须满足各工地的用量.
end
g(7) = sum(x(1:6)) - e(1);
g(8) = sum(x(7:12)) - e(2);
```

其运算结果如表 11.5 所示.

```
x =
  Columns 1 through 8
  3.0000    5.0000    3.0000    7.0000    2.0000    -0.0000
  -0.0000   -0.0000
  Columns 9 through 16
   -0.0000   -0.0000    4.0000   11.0000    5.7500    5.0000
    7.2500    7.7500.
```

表 11.5

i	1	2	3	4	5	6	新材料场位置 (x_j, y_j)
p_{i1}	3	5	3	7	2	0	(5.7500 5.0000)
p_{i2}	0	0	0	0	4	11	(7.2500 7.7500)

运送总吨千米数为 89.9t·km,比使用原材料场减少了 46.3t·km.

§11.2　捕鱼业的持续收获(求函数极值)

渔业资源是一种再生资源,再生资源要注意适度开发,不能为了一时的高产去"竭泽而渔".应该在持续稳产的前提下追求产量或最优经济效益.考察一个渔场,其中的鱼量在天然环境下按一定规律增长,如果捕捞量恰好等于增长量,那么渔场鱼量将保持不变,这个捕捞量就可以持续.

现对某个渔场的最优捕鱼量进行设计,先只考虑对某种鱼的最优捕捞策略:假设这种鱼分为 4 个年龄组,称 1 龄鱼,……,4 龄鱼.各年龄组每条鱼的平均重量分别为 5.02 两,10.98 两,18.23 两,23.04 两,各年龄组鱼的自然死亡率为 0.8/年,这种鱼为季节性集中产卵繁殖,平均每条 4 龄鱼的产卵量为 1.132×10^{11}个,3 龄鱼的产卵量为 4 龄鱼的一半,2 龄鱼和 1 龄鱼捕产卵,产卵和孵化期为每年的最后 4 个月,卵孵化成活为 1 龄鱼,成活率(1 龄鱼的条数与产卵量 m 之比)为

$$1.22\times10^{11}/(1.22\times10^{11}+n)$$

渔业管理部门规定,每年只允许在产卵孵化期前的 8 个月内进行捕捞作业.如果每年投入的捕捞能力(如渔船数,下网次数等)不变,这时单位时间内的捕捞量将与各年龄组鱼群条数成正比.比例系数一般称为捕捞强度系数.通常使用 11mm 网眼的拉网,这种网只能捕捞 3 龄鱼和 4 龄鱼,其两个捕捞强度系数之比为 0.42:1,称这种方式为固定量捕捞.现要解决下面两个问题:

(1) 如何实现可持续捕捞(即每年开始捕捞时渔场中各年龄组鱼群条数不变),并且得到最高的年捕捞总重量.

(2) 某渔业公司承包这种鱼的捕捞业 5 年,合同要求 5 年后鱼群的生产能力不能受到太大破坏,已知承包时各年龄组鱼群的数量分别为:124,28.6,10.3,3.31 ($\times10^9$条),如果仍用固定量的捕捞方式,这家公司该采取怎样的捕捞策略才能使总收获量最高.

11.2.1 问题分析及模型的建立

1. 问题假设

对该问题作以下假设:

(1) 鱼群总量的增加虽然是离散的,但对于大规模的鱼群而言,可以设鱼群总量的变化随时间是连续的.

(2) 4 龄鱼在年末留存的数量占全部数量的比例很小,可以假设全部死亡.

(3) 设鱼群每年在 8 月底产卵完毕,卵在 12 月底全部孵化完毕.

(4) i 龄鱼到第二年分别长 1 岁成为 i + 1 龄鱼,i = 1,2,3.

(5) 持续捕获使各年龄组的鱼群数量呈周期变化,周期为 1 年,可以只考虑鱼群数量在 1 年内的变化情况.

2. 问题分析

(1) 现对鱼场中各数量作以下规定:

设在 t 时刻 i 龄鱼的条数为 $x_i(t)$,其中 i = 1,2,3,4;设 4 龄鱼捕捞强度系数为 k,n 为每年产卵量,a_i 为每年初 i 龄鱼的数量,其中 i = 1,2,3,4.

(2) 鱼群死亡率的含义. 已知鱼群的自然死亡率为 0.8/年,该死亡率也就是平均死亡率,是单位时间内鱼群死亡的数量与现有鱼群数量的比例系数. 由假设可知,该系数是一个与环境等其他因素无关的常数,鱼群的数量是连续变化的,且 1 龄鱼、2 龄鱼在全年的数量只与死亡率有关,3 龄鱼、4 龄鱼在后 4 个月的数量也只与死亡率有关. 所以各龄鱼的数量满足

$$\frac{dx_i(t)}{dt} = -0.8x_i(t), \quad i = 1,2,3,4$$

(3) 捕捞强度系数的理解. 单位时间内 4 龄鱼捕捞量与 4 龄鱼群总数成正比,比例系数即为捕捞强度 k,k 是一定的,且只在捕捞期内(即每年前 8 个月)捕捞 3 龄鱼和 4 龄鱼. 所以捕捞强度系数 k 决定了 3 龄鱼、4 龄鱼在捕捞期内的数量,其变化规律为

$$\frac{dx_3(t)}{dt} = -(0.8 + 0.42k)x_3(t), \quad \frac{dx_4(t)}{dt} = -(0.8 + k)x_4(t)$$

另外,由捕捞强度系数可知,t 时刻捕捞 3 龄鱼、4 龄鱼的数量分别为 $0.42kx_3(t)$ 和 $kx_4(t)$.

(4) 鱼苗成活率的含义. 由于只有 3 龄鱼、4 龄鱼在每年的 8 月底一次产卵,因此可以将每年的产卵量 n 表示为

$$n = 1.132 \times 10^5 \times \left[0.5x_3\left(\frac{2}{3}\right) + x_4\left(\frac{2}{3}\right)\right]$$

已知成活率为 $\frac{1.22 \times 10^{11}}{1.22 \times 10^{11} + n}$,所以每年初的 1 龄鱼的数量为

$$x_1(0) = n \times \frac{1.22 \times 10^{11}}{1.22 \times 10^{11} + n}.$$

3. 第一问模型的建立

由上述分析可知,对于第一个问题可以建立如下模型

$$\max(\text{total}(k)) = 17.86\int_0^{\frac{2}{3}} 0.42kx_3(t)\,dt + 22.99\int_0^{\frac{2}{3}} kx_4(t)\,dt \qquad (11.5)$$

$$
\text{s.t.}\begin{cases}
\dfrac{dx_1(t)}{dt} = -0.8x_1(t), t \in [0,1] \\
\dfrac{dx_2(t)}{dt} = -0.8x_2(t), t \in [0,1] \\
\dfrac{dx_3(t)}{dt} = -(0.8+0.42k)x_3(t), t \in \left[0,\dfrac{2}{3}\right] \\
\dfrac{dx_3(t)}{dt} = -0.8x_3(t), t \in \left[\dfrac{2}{3},1\right] \\
\dfrac{dx_4(t)}{dt} = -(0.8+k)x_4(t), t \in \left[0,\dfrac{2}{3}\right] \\
\dfrac{dx_4(t)}{dt} = -0.8x_4(t), t \in \left[\dfrac{2}{3},1\right]
\end{cases} \tag{11.6}
$$

11.2.2　求解模型所需的知识点及其在 Matlab 中的实现

(1) 求一元函数的极小值 fminbnd()

命令形式：fminbnd(fun, x1, x2)

功能：在区间 [x1, x2] 内求函数 fun 的极小值.

注意：求多元函数极小值的命令是：fminsearch (fun, x0) 和 fminunc (fun, x0).

(2) 求常微分方程(组)的符号解 dsolve()

命令形式：dsolve('equation1, equation2, …', 'cond1, cond2, …', 'var1, var2, …')

功能：求常微分方程(组) equation1 (equation2, …) 满足初始条件 cond1 (cond2, …) 的特解. 其中 var 表示自变量.

(3) 绘制符号函数的二维曲线图 ezplot()

命令形式：ezplot(F, [xmin, xmax])

功能：绘制符号函数 F 在区间 [xmin, xmax] 内的图像.

11.2.3　模型求解在 Matlab 中的实现

第一个问题的模型在 Matlab 中的求解程序如下：

求解式(11.5)的程序如下：

```
% 首先建立约束关系的 M 函数文件：
[catchfish. M]
function    y = catchfish (x)
```

```
% f10，f20，f30，f40 表示 4 个鱼龄的鱼群初始量，k 表示捕捞强度.
syms k f10 f20 f30 f40 ;

% total 表示捕捞总量
global f10，f20，f30，f40，total，k ;
% 下面开始求解微分方程组 (11.6) 中的 6 个微分方程
x1 = dsolve ('Dx1 = - 0.8 * x1','x1 (0) = f 10') ;
t = 1 ;f 20 = subs (x1) ;

x2 = dsolve ('Dx2 = - 0.8 * x2','x2 (0) = f 20') ;
t = 1 ;f 30 = subs (x2) ;

x31 = dsolve ('Dx31 = - (0.8 + 0.42 * k) * x31','x31 (0) = f 30') ;
t = 2 /3 ;f 31 = subs (x31) ;

x32 = dsolve ('Dx32 = - 0.8 * x32','x32 (2 /3) = f 31') ;
t = 1 ;f 40 = subs (x32) ;

x41 = dsolve ('Dx41 = - (0.8 + k) * x41 (0) = f40') ;
t = 2 /3 ;f 41 = subs (x41) ;

x42 = dsolve ('Dx42 = - 0.8 * x42','x42 (2 /3) = f41') ;

nn = 1.109 * 10 5 * (0.5 * f 31 + f41) ;
eq1 = f 10 - nn * 1.22 * 10 11 / (1.22 * 10 11 + nn) ;% 求出 1 龄鱼群的存活数量.
s = solve (eq1 , f 10) ;f 10 = s (2) ;

% 下面开始求解式 (11.5) 总的积分方程的值
sym t ;
t3 = subs (subs (int (0.42 * k * x31，t，0，2 /3))) ;
t4 = subs (subs (int (k * x41，t，0，2 /3))) ;
total = 17.86 * t3 + 22.89 * t4 ;
k = x ;
```

```
y = subs (- total);

[bestcatchfish. m]
global a10, a20, a30, a40, total;
[k. mtotal] = fminbnd ('catchfish', 16, 18) ;% 求最优的捕捞强度和捕捞总量.
ezplot ( total, 0, 25) ;% 绘出捕捞强度和捕捞总量的关系图.
xlabel ('捕捞强度系数 k') ;
ylabel ('总收获量') ;
title (' 捕捞强度——总收获量曲线图') ;
format long;
k
total = - total;
a10 = eval (a10) ;
a20 = eval (a20) ;
a30 = eval (a30) ;
a40 = eval (a40) ;
format short;
clear.
```

其程序运行结果为

```
k = 17. 36293
total = 3. 88708
f 10 = 1. 19599e + 011
f 20 = 5. 37395e + 010
f 30 = 2. 41487e + 010
f 40 = 8. 39551e + 007
```

当 k = 17. 36293 时,最高年收获量为 total = 3. 88708e + 011 (两) ,此时每年年初四种鱼的数量分别为

1. 19599e + 011	5. 37395e + 010
2. 41487e + 010	8. 39551e + 007

第二个问题的分析与在 Matlab 中的求解:

由于渔业公司承包期为 5 年,并且已知年初时各年龄组的鱼群的数量,在一定的捕捞强度系数 k 下,5 年的总收获量是 k 的一元函数 total (k) ,但在 5 年中,每年初各年龄组的鱼群数量是不一样的.

% 公司承包 5 年最优捕鱼量的计算程序：

```
[fish. M]
function    y = fish (x)
global total, k, temp ;
syms k fish10 fish20 fish30 fish40 ;
syms qp pq temp ;
x1 = despve ('Dx1 = - 0. 8 * x1', 'x1 (0) = fish10') ;
x2 = dsolve ( 'Dx2 = 0. 8 * x2',' x2 (0) = fish20') ;
x31 = dsolve ('Dx31 = - (0. 8 + 0. 42 * k) * x31', 'x31 (0) = fish30') ;
t = 2 /3 ;fish31 = subs (x31) ;
x32 = dsolve ('Dx32 = - 0. 8 * x32' , 'x32 (2 /3) = fish31') ;
x41 = dsovle ( 'Dx41 = - (0. 8 + k) * x41', 'x41 (0) = fish40') ;
t = 2 /3 ;fish41 = subs (x41) ;
s = pq * zeros (6, 4) ;
s (1, :) = [122. 0 29. 7   10. 1   3. 29] * 10 9 ;
temp = [ pq pq pq pq] ;
qp = [pq pq pq pq] ;
for j = 1: 5
    fish10 = s (j, 1) ;fish20 = s (j, 2) ;fish30 = s (j, 3) ;fish40 = s (j,4) ;
    t = 2 /3 ;
    fish31 = subs (fish31) ;fish41 = subs (fish41) ;
    nn = 1. 109 * 10 5 * (0. 5 * fish31 + fish41) ;
    syms t ;
    t3 = subs (subs (int (0. 42 * k * x31, t, 0, 2 /3))) ;
    t4 = subs (subs (int (k * x41, t, 0, 2 /3))) ;
    qp (i) = 17. 86 * t3 + 22. 99 * t4 ;
    temp (1) = nn * 1. 22 * 10 11 / (1. 22 * 10 11 + nn) ;
    t = 1 ;
    temp (2) = subs (x1) ;
    temp (3) = subs (x2) ;
    temp (4) = subs (x32) ;
    s (j + 1, :) = temp ;
end
total = sum (qp) ;
```

```
k = x ;
y = - subs (total) .
% 建立求最优捕捞强度和最大总收获量的 M 程序：
[zuiyoubulao. M]
format long ;
global  totalweight  k  mytemp ;
% 用最优化方法求 5 年中的最优捕捞强度和总的捕捞量.
[fishk , fishtotal] = fminbnd ( ' fish ' , 17 , 18 , optimset (' tolx ' , 1e - 16) ) ;
fishk ; % 捕捞强度系数.
fishtotal = - fishtotal ; % 总的捕捞量.
ezplot (totalweight , 0 , 25) ;
xlabel (' 捕捞强度系数 ') ;
ylabel (' 总捕鱼量 ') ;
title (' 5 年中捕捞强度——总收获量曲线图 ')
```

其程序运行结果为：

```
fishk = 17. 43804874105403
fishtotal = 1. 598793940035355e + 012
```

当 k = 17. 4380480487 时，5 年的最高收获量为

$$\text{fishtotal} = 1.598793940035355 \times 10^{9}\text{kg}.$$

§11.3　化工公司产品生产计划（线性规划）

科明科化工公司生产两种主要产品 A 和 B，两种产品都需要相同的两道工序. 生产每公斤 A 产品第一道工序需要 2h，第二道工序需要 3h. B 产品第一道工序需要 3h，第二道工序需要 4h. 用于两道工序的设施可以同时使用，但同一时间某道工序的设施只能用于生产一种产品. 在每一天中，启用第一道工序设施的时间不能超过 16h，第二道工序的设施可以连续不停的使用.

在生产产品 B 时将同时生产副产品 C，生产每公斤 B 产品可得 2kg C 产品，但对副产品 C 而言，一日内售出的部分可获利，剩余的由于必须销毁，因此反而产生费用.

每公斤 A 产品的售价是 400 元，B 产品是 1 000 元. 每公斤副产品 C 售价是 300 元，但一日内未售出的部分则必须销毁，每公斤的销毁费用是 200 元. 根据市场调查所作出的预测表明，副产品 C 的日销售量在 5kg 以下. 现需要制定该公司 A 和 B 两种产品的日产量计划，使利润达到最大.

11.3.1 问题分析及模型的建立

由所给条件可知,该问题中的决策变量为:

x_1:产品 A 的日产量 (kg).

x_2:产品 B 的日产量 (kg).

x_3:副产品 C 的日售出量 (kg).

x_4:副产品 C 的日销毁量 (kg).

这个范例的难点在于,如何将某些非线性规划问题用线性模型来描述,由于副产品 C 的单位利润有正、负两个值,使得目标函数呈非线性,因为目标函数中副产品 C 的利润系数取决于副产品 C 的日产量,处理的奥妙体现在引入了两个附加的决策变量 x_3 和 x_4,从而把副产品 C 的日产量分为两部分,一部分是被出售的,另一部分是被销毁的,这样目标函数就可以表示成这四个决策变量的线性函数了.

下面是求解该问题时的约束条件:

每公斤 B 产品对应 2kg 副产品 C:$x_3 + x_4 = 2x_2$;

第一道工序设施启用时间不超过 16h:$2x_1 + 3x_2 \leqslant 16$;

第二道工序设施启用时间不超过 24h:$3x_1 + 4x_2 \leqslant 24$;

副产品 C 的日售量不会大于 5kg:$x_3 \leqslant 5$;

所有决策变量显然非负:$x_1 \geqslant 0, x_2 \geqslant 0, x_3 \geqslant 0, x_4 \geqslant 0$.

所以,该问题的目标函数 (利润最大) 为

$$\max z = 400x_1 + 1\,000x_2 + 300x_3 - 200x_4.$$

下面给出该问题的线性规划模型

$$\max z = 400x_1 + 1\,000x_2 + 300x_3 - 200x_4$$

$$\text{s.t.}\begin{cases} -2x_2 + x_3 + x_4 = 0 \\ 2x_1 + 3x_2 \leqslant 16 \\ 3x_1 + 4x_2 \leqslant 24 \\ 0 \leqslant x_3 \leqslant 5 \\ x_1 \geqslant 0, x_2 \geqslant 0, x_4 \geqslant 0 \end{cases}$$

只有当副产品 C 的日产量大于 5kg 时,其剩余部分才应销毁. 换句话说,如果 C 的日售出量 (x_3) 小于 5kg,日销毁量 (x_4) 应当为 0. 这种约束很难直接处理,但注意到目标函数中 x_4 的系数为负,求极大值时将自动地使之尽量靠近其下界 (0). 为了可靠起见的一个简单措施是对所求出的解进行验证.

11.3.2　求解模型所需的知识点及其在 Matlab 中的实现

求线性规划的函数 lp. 在 Matlab 中,同一个函数有多种形式,求解线性规划的函数 lp 也是如此,lp 最实用的形式是

```
x = lp (c, A, b, xLB, xUB, x0, nEq)
```

lp 用于求解下列线性规划模型

$$\min f = c'x$$

$$\text{s. t.} \begin{cases} Ax \leqslant b \\ xLB \leqslant x \leqslant xUB \end{cases}$$

函数中的 x0 参数是算法迭代的初点,可以任取一个点.

11.3.3　模型求解在 Matlab 中的实现

模型求解问题在 Matlab 中实现的程序如下:

```
echo off ;
close all hidden ;
fclose ('all') ;
clear ;
clc ;
format short ;
c = [ - 400 ; - 1000 ; - 300 ;200] ;
A = [0 - 2 1 1 ;2 3 0 0 ; 3 4 0 0] ;
B = [0 ;16 ;24]
xLB = zeros (4, 1) ;
xUB = inf * ones (4, 1) ;
xUB (3) = 5 ;
x0 = 0 * ones ( 4, 1) ;
nEq = 1 ;
x = lp (c, A, b, xLB, xUB, x0, nEq) ;

ProductA = x (1) ;
ProductB = x (2) ;
ByProductC = x (3) + x (4) ;
Csold = x (3) ;
Cdestroyed = x (4) ;
```

```
Profit = - c' * x ;
```

其程序运行的结果为

```
ProductA = 3. 4483          Csold = 5
ProductB = 3. 0345          Cdestroyed = 1. 0690
ByProductC = 6. 0690        Profit = 5700
```

由计算结果可知：产品 A 的日产量为 3. 4483kg，产品 B 的日产量为 3. 0345kg；副产品 C 的日产量为 6. 069kg，其中 5. 000kg 售出，1. 0690kg 销毁，可以获得最大利润 5 700 元.

§11. 4 围墙所围土地的面积（非线性规划）

非线性规划模型的特征是目标函数和约束函数中含有非线性函数，这增加了问题求解的难度. 非线性规划一般是指非线性约束优化，但只要目标函数是非线性的，也可以讨论无约束优化问题.

问题：在长江中有一风景秀丽的小岛，某集团承包下该小岛的开发权后想在岛上建一度假村. 开发的过程中，按照惯例，要先用砖围一个矩形围墙，以便存放建筑材料. 岛上的建筑拆迁时，恰好留下了一批旧砖，可以利用这批旧砖来建围墙，旧砖的长度是 0. 2m，厚度是 0. 05m，砖的总数量是 12 000 块. 在建围墙时，要求围墙的高度不能低于 2m，围墙围住的面积越大越好. 试问该如何来建围墙？

11. 4. 1 问题分析及模型的建立

设建围墙所用的旧砖的型号如下：

砖的长度为 lbm；

砖的厚度为 tbm；

旧砖的总数量为 Nb 块.

设所围起的地的长度为 xm，宽为 ym；所建起的围墙的高为 zm；且在建围墙时，砖是平放于地上的. 现在来确定修建该围墙时所要满足的条件：

(1) 围墙的长、宽、高都不能是负数，即：$x > 0, y > 0, z > 0$；

(2) 围墙的高度不能低于 2m，即：$z \geqslant 2$；

(3) 围墙建好后，考虑围墙内侧的表面，该表面积为：$S = 2(x + y)z$. 砖由长和厚组成的那一侧的面积为 $lb \times tb$，因为旧砖的数量 Nb 是一定的，所以所有砖的该侧面的总面积为 $lb \times tb \times Nb$. 因为建围墙时，砖平放于地上，所以围墙的内侧的表面积 S 应小于或等于 $lb \times tb \times Nb$，即

$$2(x+y)z \leqslant lb \times tb \times Nb$$

围墙所围的地的面积为 xy,现在要在现有的条件下使得 xy 达到最大. 所以对于该问题可以建立下面的模型

$$\max f(x,y) = xy$$

$$\text{s.t.}\begin{cases}(2(x+y)z \leqslant lb \cdot tb \cdot Nb \\ x>0, y>0, z>0 \\ z \geqslant 2\end{cases}$$

由线性规划的知识可知,该模型就是一个非线性规划问题. 把已知的数据代入该模型,可以得到最后的模型为

$$\max f(x,y) = xy$$

$$\text{s.t.}\begin{cases}2(x+y)z \leqslant 0.2 \times 0.05 \times 12\,000 = 120 \\ x>0, y>0, z>0 \\ z \geqslant 2\end{cases}$$

11.4.2　建模所需的知识点及其在 Matlab 中的实现方法

(1) 求带约束的非线性规划 constr()

命令形式:X = constr('fun', x0)

功能:从点 x0 开始求带约束条件的函数 fun 的极小值点 X. 其中 fun 是 M 函数文件名,$X, x0 \in R^n$.

(2) ss = num2str(ff)

把数字 ff 转化为字符串 ss.

11.4.3　模型求解在 Matlab 中的实现

现在在 Matlab 中求该非线性规划问题的解. % 首先建立目标函数和约束条件的 M 函数文件.

```
function [S, G] = YueshuFun(X)
% S 表示目标函数的相反数,G 表示围墙内表面积的约束函数.
S = - X(1) * X(2);
G = 2 * (X(1) + X(2)) * X(3) - 120;
% 下面建立求解围墙所围的最大面积的 M 函数文件.
clear all;
clc ;
fclos('all') ;
format short;
```

```
X = [10 ;10 ;2] ;
options (13) = 0 ;% 对非线性规划求解函数 constr 的参数设置.
% 设置参数的范围:
XL = [0 ;0 ;2] ;
XU = [inf ;inf ;inf] .

% 直接调用 Matlab 中的函数 constr 求非线性规划的解.
[X, options] = constr ('YueshuFun' , X,options, XL, XU) ;

disp ('运算结果如下:\n') ;
disp ('所围土地的最大面积为:'num2str ( - options (8)) ) ;
disp ('\n') ;
disp ('面积最大时,围墙的长、宽、高分别为:')
disp (num2str (X)) ;
```

Matlab 的输出结果为:

所围土地的最大面积为:900m^2

面积最大时,围墙的长、宽、高分别为:30m、30m、2m

§11.5 真题解析:截断切割问题

问题:某些工业部门(如贵重石材加工等)经常需要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),通常采用截断切割的加工方式,这里"截断切割"是指将物体沿某个切割平面分成两部分.因此在一般情况下,需经过 6 次截断切割,分别截去原长方体的前、后、左、右、上、下 6 个方向多余的部分.

设水平切割单位面积的费用是垂直切割单位面积的 r 倍,且当先后两次垂直切割的平面不平行时,因调整刀具需额外费用 e.

显然,若截去各方向多余小块的先后顺序不同,则加工费用不同.试设计一种确定最优加工次序方法,此处的最优是指加工费用最少(由工艺要求,与水平工作台接触的长方体底面是事先指定的).

用下列实例验证所设计的方法:待加工长方体与成品长方体的长、宽、高分别为 10,14.5,19 和 3,2,4,二者左面、前面、底面之间的距离分别为:6,7,9(单位:cm),垂直切割费用为 1 元/cm^2,r 和 e 的数据有以下 4 组:

(1) $r=1, e=0$；　(2) $r=1.5, e=0$；

(3) $r=8, e=0$；　(4) $r=1.5, 2\leqslant e\leqslant 15$.

11.5.1　问题分析

这是一个优化问题，求切割顺序，使加工费用最低. 决策变量为切割顺序，用 $X=(x_1, x_2, \cdots, x_6)$ 来表示切割顺序，x_i 表示第 i 次切割，可以取 1，2，…，6 分别表示左、右、前、后、上、下方向的切割，$x_1, x_2, \cdots, x_6$ 互不相同，可以取 1，2，…，6 的任意全排列.

目标函数：加工费用由切割费用和刀具调整费用构成.

问题中的已知条件有：

(1) 待加工长方体与成品长方体对应表面平行.

(2) 切割费用与切割面的面积成正比，具体地说就是垂直切割费用为 1 元/cm^2，水平切割费用为 r 元/cm^2，且仅当先后两次垂直切割的切割面不平行时，才需调整刀具，调整刀具的费用为 e.

(3) 水平工作台接触的长方体底面是事先指定的.

(4) 不考虑第一次切割前的刀具调整费用.

11.5.2　建立数学模型

设待加工长方体的长——a、宽——b、高——c 为常数，待加工长方体与成品长方体两者的左、右、前、后、上、下面之间的距离 $a_1, a_2, b_1, b_2, c_1, c_2$ 也为常数. 可变参数有：水平切割费用 r 元/平方厘米，调整刀具的费用 e. 在切割方式 X 下，对应的加工费用可以表示为 $f(X, e, r)$. 可以建立如下的组合优化模型：

求一个切割方式 $X=X_{min}$，使加工费用 $f(X, e, r)$ 达到最小，即

$$\min_{X\in S} f(X, e, r)$$

其中　$S=\{X=(x_1, x_2, \cdots, x_6) \mid x_i=1, 2, \cdots, 6, x_i\neq x_j \text{ 当 } i\neq j\}$

由于集合 S 为有限集，只有 6！=720 种切割方式，当 e，r 取定，切割顺序给定，很容易计算出加工费用. 可以依次求出各切割方式下的切割费用，比较最小者，便可得到最小费用的加工顺序.

11.5.3　模型在 Matlab 中的求解

情形一：e=0 时.

先用穷举法求出 720 种切割方式下的费用，存放在数组 c 中，再用函数 min (c) 和 find (c==minc) 求最小费用及其对应的切割方式.

e=0 时，模型在 Matlab 中的求解程序：

```
% jieduan      e = 0
% a0 ：三维向量，各分量为待加工长方体的长，宽，高.
% a1 ：三维向量，各分量为成品长方体的长，宽，高.
% d1 ：三维向量，各分量为待加工与成品长方体两者的左面，前面，底面之间的距离.
% r ：水平切割单位面积的切割费用.
% minc：最小费用.
% minX：列数为 6 的矩阵，各行为最小费用对应的切割顺序.
a0 = [10, 14.5 , 19] ;
a1 = [3, 2, 4] ;
d1 = [6, 7, 9] ;
r = 1;
d2 = a0 - a1 - d1 ;d = [d1 , d2] ;
d = d([1,4,2, 5, 3, 6]) ;
p = 0.
%建立可行的加工顺序表.
for I = 1:6
    for j = 1:6, if (j - I) ~ = 0,
        for k = 1:6, if (k - I) * (k - j) ~ = 0,
            for l = 1:6, if (l - I) * (l - k) ~ = 0,
                for m = 1 6, if (m - I) * (m - j) * (m - k) * (m - l) ~ = 0;
                        for n = 1:6
                            if (n - I) * (n - j) * (n - k) * (n - l) * (n - m) ~ = 0,
                            p = p + 1 ;
                             X (p, :) = [I , j, k, l, m, n] ;
End,end, end, end , end, end, end, end, end.
%建立加工顺序表 X 对应的切割费用表
f = [1, 1, 2, 2, 3, 3] ;
for p = 1:720
    o = X (p, :) ;cost = 0 ;a = a0 ;
    for I = 1:6
        j = o (i) ;aa = a ;aa (f (j) ) = [] ;
    if f (j) == 3
        cost = cost + r * aa (1) * aa (2) ;
```

```
        else
            cost = cost + aa(1) * aa(2);
        end
            a(f(j)) = a(f(j)) - d(j);
        end
            c(p) = cost;
    end.
    % 求最小费用及其对应的加工顺序：
        minc = min(c), find(c == minc);
        minx = x(ans, :);
```

其程序运行结果为：

```
minc = 374
minx =   5   3   1   6   4   2
         5   3   6   1   4   2.
```

因此，当 $r=1$，$e=0$ 时，最优加工顺序为“下—前—左—上—后—右”，或“下—前—上—左—后—右”，切割费用为 374 元.

情形二：$e\neq 0$ 时.

加工费用是由切割费用和调整刀具的费用两者组成，即

$$f(X,e,r)=f(X,0,r)+z\times e$$

其中 z 为加工顺序是 X 时的调整刀具次数，显然 $z=1,2,3$. 把全体切割顺序按调整刀具次数划分为三类，同类的刀具调整费用是相等的. 可以先分别求出在 $e=0$ 时，每一类的最小费用及相应的加工顺序，它们便是各类的最优加工顺序. 再将每一类的最小切割费用加上相应的刀具调整费用，得到加工总费用. 将各类的最优加工顺序进行比较，便可得整体的最优加工顺序.

$e>0$ 时，模型在 Matlab 中的求解程序：

```
% jieduan_e > 0
function [min, minx1, minx2, minx3] = cutordel(a0, a1, d, r)
minc = [inf, inf, inf]; minx1 = []; minx2 = []; minx3 = [];
k1 = 0; k2 = 0; k3 = 0;
v1 = [1, 2, 3, 4, 5, 6];
% 建立三类可行的加工顺序表 x1, x2, x3 及相应的切割费用表.
% c1, c2, c3
for i1 = 1:6, ol = v1(i1); v2 = v1; v2(i1) = [];
for i2 = 1:5, o2 = v2(i2); v3 = v2; v3(i2) = [];
```

```
    for i3 = 1:4, o3 = v3(i3);v4 = v3;v4(i3) = [];
        for i4 = 1:3, o4 = v4(i4);v5 = v4;v5(i4) = [];
            for i5 = 1:2, o5 = v5(i5);o6 = (3 - i5);
                x = [o1, o2, o3, o4, o5, o6];
                c = cost(x, a0, a1, d1, r);
                z = adjustnum(x);
                switch z
                case 1
                    k1 = k1 + 1;x1(k1, :) = x;c1(k1) = c;
                case 2
                    k2 = k2 + 1;x2(k2, :) = x;c2(k2) = c;
                case 3
                    k3 = k3 + 1;x3(k3, :) = x;c3(k3) = c;
end, end, end, end, end, end.
minc = [min(c1), min(c2), min(c3)];
find(c1 == minc(1));
minx1 = x1(ans, :);find(c2 == minc(2));
minx2 = x2(ans, :);find(c3 = minc(3));
minx3 = x3(ans, :).
% 求切割顺序是 x 时,切割费用的子函数 cost 为.
function c = cost(x, a0, a1, d1, r)
c = 0;
d2 = a0 - a1 - d1;a = a0;
for p = 1:6
switch x(p)
case 1
    c = c + a(2) * a(3);
    a(1) = a(1) - d1(1);
case 2
    c = c + a(2) * a(3);
    a(1) = a(1) - d2(1);
case 3
    c = c + a(1) * a(3);a(2) = a(2) - d1(2);
case 4
```

```
        c = c + a (1) * a (3) ;
        a (2) = a (2) - d2 (2) ;
case 5
        c = c + r * a (1) * a (2) ;
        a (3) = a (3) - d1 (3) ;
case 6
        c = c + r * a (1) * a (2) ;
        a (3) = a (3) - d2 (3) ;
end
  end.
% 求加工顺序 x 的调整刀具次数的子函数 adjustnum (x) 为：
function   z = adjustnum (x)
z = - 1 ; v0 = 0 ;
for      p = 1 : 6
        if x (p) < 5
            if x (p) < 3
                v = 1 ;
            else
                v = 2 ;
            end
            if (v0 - v) ~ = 0
              z = z + 1 ;
              v0 = v ;
            end
          end
     end
```

在 Matlab 中输入命令：

```
  a0 = [10, 14.5, 19] ; a1 = [3, 2, 4] ; d1 = [6, 7, 9] ; r = 1.5 ;
   [minc, minx1, minx2, minx3] = cutrode (a0, a1, d1, r)
```

其程序运行结果为

```
  minc =  442.500    456.500    437.500
  minx1 = 3   5   4   1   6   2
  minx2 = 1   3   5   4   6   2
  minx3 = 3    1     5    4    6    2
```

3 5 1 4 6 2

因此,每一类的最小费用分别为:

(1) $C_1(e)=442.5+e$,此时调整一次刀具.

(2) $C_2(e)=456.5+2e$,此时调整两次刀具.

(3) $C_3(e)=437.5+3e$,此时调整三次刀具.

对应的加工顺序为(3,1,5,4,6,2)和(3,5,1,4,6,2).

习 题 12

1. 某公司下设三个工厂,生产同一种产品,现要把三个工厂生产的产品运送给四个订户. 工厂的供应量、订户的需求量以及从三个工厂到四个订户的单位运费如表 11.6 所示(表内数字为单位运费).

表 11.6

工 厂	订 户				供应量
	1	2	3	4	
1	5	2	6	7	30
2	3	5	4	10	20
3	4	3	2	13	40
需求量	20	10	45	25	

现在要作出一个调运计划,依次满足下列各项要求:

(1) 订户 4 的订货量首先要保证全部予以满足;

(2) 其余用户的订货量满足程度应不低于 80%;

(3) 工厂 3 调运给订户 1 的产品量应不少于 15 个单位;

(4) 因线路限制,工厂 2 应尽可能不分配给订户 4;

(5) 订户 1 和订户 3 的需求满足程度应尽可能平衡;

(6) 力求使总运费最小.

试建立上述问题的目标规划模型(不必求解).

2. 一般从事猪的商业性饲养和销售总是希望获得利润,因此饲养某种猪是否获利,怎样获得最大利润,是饲养者必须考虑的问题. 如果把饲养技术水平、猪的类型等因素视为不变的,且不考虑市场的需求变化,那么影响获利大小的一个主要因素是如何选择猪的出售时机,即何时把猪卖出获利最大. 也许有人认为,猪养得越大,售出后获利越大. 其实不一定,因为随着猪的生长,单位时间消耗的饲养费也就越多,同时其体重的增长速度却不断下降,所以饲养时间过长是不合

算的.

设某养殖厂的猪开始进行商业饲养的时刻 $t=0$，x_0 为 $t=0$ 时猪的体重，即 $x(0)=x_0$，$x(t)$ 为一头猪在时刻 t 的体重，X 为该品种猪的最大体重；$y(t)$ 为一头猪时刻 t 共消耗的饲养费用（包括饲料费，饲养人员工资等），$y(0)=0$；x_s 为猪可以出售的最小体重，即体重不超过 x_s 的猪，收购站不予收购. t_s 为猪从重 x_0 长至重 x_s 所需的时间；$C(x)$ 为猪的单位重量售价，C_0 为刚出生小猪的单位价格. 假设：

(1) 只对某一品种猪进行讨论，涉及猪的性质的相关参数可以视为固定的常数.

(2) 由于开始进行商业性饲养时已具有一定体重，所以可以假设猪体重增长的速度将不断减慢. 设反映猪体重增长速度的参数为 a.

(3) 由于猪体重越大，单位时间消耗的饲养费用就越多，达到最大体重后，单位时间消耗的饲养费接近某一常数 γ.

(4) 通过调查了解，$C(x)$ 随 x 的变化幅度不大，故可以将 $C(x)$ 视为常数，设其为 C.

假定某品种的猪，$X=200$（kg），$x_s=75$（kg），$a=0.5$（kg/天），$C=6$（元/kg），$\gamma=1.5$（元/天），$\beta=1$（元/天），$x_0=5$（kg）. 试建立猪的最佳销售时机的数学模型.

3. 某工厂生产一种产品有甲、乙两个牌号，讨论在产销平衡的情况下如何确定各自的产量，使总的利润最大. 所谓产销平衡是指工厂的产量等于市场上的销量.

利润即取决于销量和单位价格，也依赖于产量和单位成本. 按照市场经济规律，甲的价格 p_1 会随其销量 x_1 的增长而降低，同时乙的销量 x_2 的增长也会使甲的价格有稍微的下降，可以简单地假设价格与销量呈线性关系，即

$$p_1=b_1-a_{11}x_1-a_{12}x_2,\quad b_1,a_{11},a_{12}>0,a_{11}>a_{12}$$

乙的价格 p_2 也遵循同样的规律，有

$$p_2=b_2-a_{21}x_1-a_{22}x_2,\quad b_2,a_{21},a_{22}>0,a_{22}>a_{21}$$

甲的成本随其产量的增长而降低，且有一渐近值，合理地假设为负指数系，即

$$q_1=r_1e^{-\lambda_1x_1}+c_1,\quad r_1,\lambda_1,c_1>0$$

乙的成本 q_2 遵循同样的规律，有

$$q_2=r_2e^{-\lambda_2x_2}+c_2,\quad r_2,\lambda_2,c_2>0$$

于是总利润为

$$z(x_1,x_2)=(p_1-q_1)x_1+(p_2-q_2)x_2$$

设　$b_1=100,a_{11}=1,a_{12}=0.1,b_2=280,a_{21}=0.2,r_1=30,\lambda_1=0.015,c_1=$

20，$r_2=100$，$\lambda_2=0.02$，$c_2=30$，求甲、乙两个牌号的产量 x_1，x_2，使总利润最大.

4. 某饮料厂生产甲、乙两种口味的饮料，每百箱甲饮料需用原料 6kg，工人 10 人，可获利 10 万元；每百箱乙饮料需用原料 5kg，工人 20 人，可获利 9 万元. 今工厂共有原料 60kg，工人 150 名，又由于其他条件所限制，甲饮料产量不能超过 800 箱. 试问如何安排生产计划，即两种饮料各生产多少获利最大.

5. 某炼油厂将 A、B、C 三种原油加工成甲、乙、丙三种汽油. 一桶原油加工成一桶汽油的费用为 4 元，每天至多能加工汽油 14 000 桶. 原油的买入价、买入量、辛烷值、硫含量及汽油的卖出价、需求量、辛烷值、硫含量如表 11.7 所示. 试问如何安排生产计划，在满足需求的条件下使利润最大？

一般来说，做广告可以增加销量，估计一天向一种汽油投入一元广告费，可以使这种汽油日销量增加 10 桶. 试问如何安排生产和广告计划使利润最大？

表 11.7

原油类别	买入价/(元/桶)	买入量/(桶/天)	辛烷值/(%)	硫含量/(%)
A	45	≤5000	12	0.5
B	35	≤5000	6	2.0
C	25	≤5000	8	3.0
汽油类别	**卖出价/(元/桶)**	**需求量/(桶/天)**	**辛烷值/(%)**	**硫含量/(%)**
甲	70	3000	≥10	≤1.0
乙	60	2000	≥8	≤2.0
丙	50	1000	≥6	≤1.0

6. 某部门现有资金 10 万元，5 年内有以下投资项目供选择：

项目 A，从第一年到第四年每年初投资，次年末收回本金且获利 15%；

项目 B，第三年初投资第五年末收回本金且获利 25%，最大投资额为 4 万元；

项目 C，第二年初投资，第五年末收回本金且获利 40%，最大投资额为 3 万元；

项目 D，每年初投资，年末收回本金且获利 6%.

试问如何确定投资策略，使 5 年末本息总额最大.

第三部分　Matlab 基础和高级编程

A　Matlab 软件使用简介

A1. Matlab 的变量与表达式

A1.1　Matlab 的变量

计算机是通过变量名字找到所论变量在内存中位置的. Matlab 的变量名除定义的保留字外,可以用一个字母开头,后面最多跟 19 个字母或数字来定义,如 x,z,aa3 等都是合法的变量名. 变量名不能以数字开头的字符串来表示. 应该注意不要与 Matlab 中的内部函数或命令相混淆. Matlab 中的变量名是区分大、小写字母的,如 ab 与 Ab 就表示两个不同的变量.

在 Matlab 中使用变量前不必先定义变量类型,可以即取即用,这可以给我们使用 Matlab 带来很大的方便. 但是,如果使用与原来定义的变量一样的名字来赋值,原变量就会被自动覆盖,系统不会给出错误信息.

A1.2　Matlab 的运算符

(1) 数学运算符

+ (加号), - (减号), * (乘号), h (左除), / (右除), (乘幂).

(2) 关系运算符

< (小于), > (大于), <= (小于等于), >= (大于等于), == (等于), ~= (不等于).

(3) 逻辑运算符

& (逻辑与运算), | (逻辑或运算), ~ (逻辑非运算).

A1.3 Matlab 的表达式

Matlab 采用的是表达式语言,用户输入的语句由 Matlab 系统解释运行. Matlab 语句由变量与表达式组成. Matlab 语句有两种最常见的形式.

形式 1:表达式.

形式 2:变量 = 表达式.

表达式由运算符、函数、变量名和数字组成. 在形式 1 中,表达式运算后产生的结果如果为数值类型,系统自动赋值给 ans. 但对于重要结果一定要用形式 2,在形式 2 中,对等式右边表达式产生的结果,系统自动将其存储在左边的变量中,并同时显示在屏幕上.

如果不想显示形式 1 或形式 2 的运算结果,可以在命令中表达式后加上“;”即可.

例 1. 用两种形式计算 $5^6 + \sin\pi + e^3$ 的算术运算结果.

解 Matlab 的命令为:

形式 1

```
5 6 + sin (pi) + exp (3)
ans = 1.5645e + 004.
```

形式 2

```
a = 5 6 + sin (pi) + exp ( 3 )
a = 1.5645e + 004.
```

如果在表达式后面加“;”,即

```
a = 5 6 + sin (pi) + exp ( 3 ) ;
```

则运算后不显示运算结果.

A1.4 Matlab 的数据显示格式

虽然 Matlab 系统中数据的存储和计算都是双精度进行的,但 Matlab 中可以利用菜单或 Format 的命令来调整数据的显示格式. Format 命令的格式和作用如下:

format	short	5 位定点表示.
format	long	15 位定点表示.
format	short e	5 位浮点表示.
format	long e	15 位浮点表示.
format	short g	系统选择 5 位定点和 5 位浮点中更好的表示.
format	long g	系统选择 15 位定点和 15 位浮点中更好的表示.

format rat	近似的有理数的表示.
format hex	十六进制的表示.
format + (plus)	表示大矩阵时分别用 +、- 和空格表示矩阵的正数、负数和零.
format bank	用元、角、分 (美制) 定点表示.
format compact	变量之间没有空行.
format loose	变量之间有空行.

用菜单来调整数据的显示格式的方法为:在 Matlab 命令窗口单击 File | Preferences 调出显示格式的设置界面,如图 A. 1 所示,然后在图左边的选项 (Numeric Format) 中选择需要的格式即可.

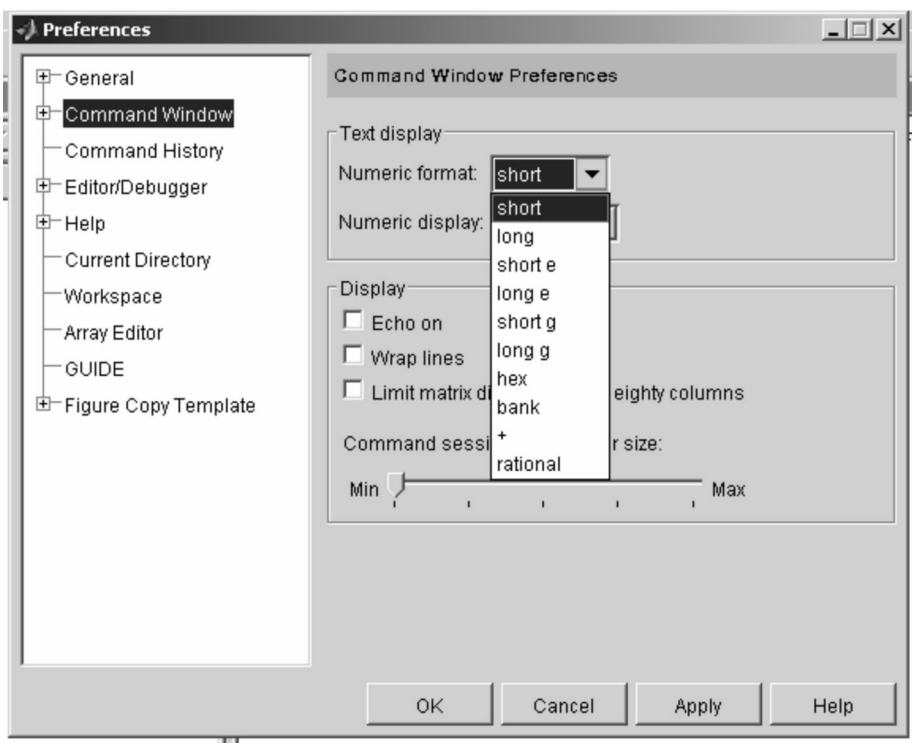


图 A. 1 显示格式的设置

A2. Matlab 的常用函数

Matlab 有很丰富的内部函数,这些函数是 Matlab 系统自带的函数. 函数名一般使用数学中的英文单词,只要输入相应的函数名,就可以方便地使用这些函数. 内部函数既有数学中常用的函数,也有工程中用的特殊函数. 用户也可以自己定义所需要的函数.

Matlab 的常用内部函数如表 A. 1 和表 A. 2 所示.

表 A. 1 常用的内部函数

函数名称	函数功能	函数名称	函数功能
sin (x)	正弦函数 sin x	asin (x)	反正弦函数 arcsin x
cos (x)	余弦函数 cos x	acos (x)	反余弦函数 arccos x
tan (x)	正切函数 tan x	atan (x)	反正切函数 arctan x
con (x)	余切函数 cot x	acot (x)	反余切函数 arccot x
sec (x)	正割函数 sec x	asec (x)	反正割函数 arcsec x
sinh (x)	双曲线函数 sinh x	asinh (x)	反双曲线函数 arcsin x

表 A. 2 常用的计算函数

函数名称	函数功能
abs (x)	求变量 x 的绝对值
angle (x)	求变量 x 的辐角
sqrt (x)	求变量 x 的平方根
real (x)	求复数 x 的实部
image (x)	求复数 x 的虚部
conj (x)	求复数 x 的共轭复数
round (x)	四舍五入到最近整数
fix (x)	无论正、负,舍去小数至最近整数
ceil (x)	加入正小数至最近整数
floor (x)	舍去正小数至最近整数
rat (x)	将实数化为分数表示
rats (x)	将实数化为多项式分数表示
sign (x)	符号函数
rem (x)	求 x 除以 y 的余数
gcd (x,y)	整数 x 和 y 的最大公因数
1cm (x,y)	整数 x 和 y 的最小公倍数
exp (x)	自然指数 e^x
pow2 (x)	2 的指数 2^x
log (x)	自然对数 ln x
log2 (x)	以 2 为底的对数
log10 (x)	以 10 为底的对数

A3.　Matlab 的基本对象

Matlab 中最基本的处理对象是矩阵、数组与字符串.

A3.1　矩阵

Matlab 的最基本单位是矩阵.

1. 矩阵的输入

矩阵的输入主要有三种方式. 第一种方式是直接输入,该方式适用的对象是维数较少的矩阵;第二种方式是利用矩阵编辑器来输入矩阵,该方式适用于维数较大的矩阵;第三种方式是利用矩阵函数来创建特殊矩阵.

(1) 直接输入创建矩阵

输入方法是先键入左方括号“[”,然后输入矩阵的所有元素,最后键入右方括号“]”.

注意:整个矩阵以“[”和“]”作为首、尾,同行的元素用“,”或空格隔开;不同行的元素用“;”或按 Enter 键来分隔;如果进行的是数值计算,矩阵的元素可以为数字也可以为表达式.

例 2. 直接输入创建矩阵 $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 15 & 60\\ 7 & 8 & 9\end{pmatrix}$.

解　Matlab 命令为

```
A = [1,2 ,3 ;4, 15, 60 ;7, 8, 9]
A = 1    2    3
    4   15   60
    7    8    9.
```

(2) 用矩阵编辑器来创建、修改矩阵

当输入的矩阵较大,不适合用手工直接输入时,可以用矩阵编辑器来进行输入和修改. 但要注意,在调用编辑器前需定义一个变量,无论是一个数值还是一个矩阵均可. 利用矩阵编辑器来输入矩阵的步骤如下:

1) 双击命令窗口快速工具栏中的工作区浏览器,选中变量 A 就可以对变量 A 做删除与修改操作.

2) 双击左键或单击 Open 按钮打开矩阵编辑器,如图 A.2 所示.

3) 在矩阵编辑器的右上方的两个文本框(由 by 连在一起)中可以修改矩阵的维数.

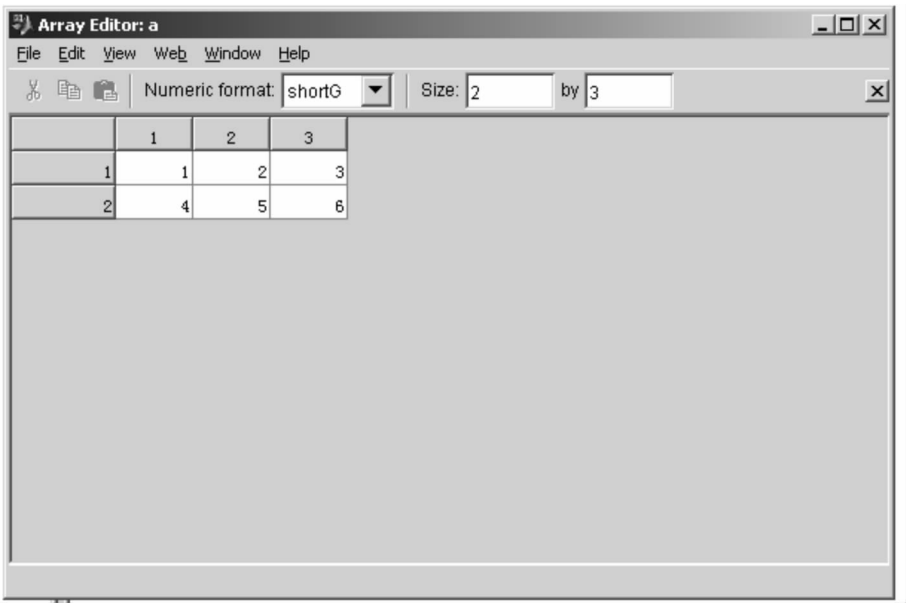


图 A.2　矩阵编辑器

(3) 用矩阵函数来生成矩阵

常用的矩阵函数如表 A.3 所示.

表 A.3　　常用的矩阵函数

函数名称	函数功能	函数名称	函数功能
zreos (m,n)	生成 m 行 n 列的零距阵	eig (A)	求矩阵 A 的特征值
eye (n)	生成 n 阶单位矩阵	poly (A)	求矩阵 A 的特征多项式
ones (m,n)	生成 m 行 n 列的元素为 1 的矩阵	trance (A)	求矩阵 A 的迹
rand (m,n)	生成 m 行 n 列的随机矩阵	cond (A)	求矩阵 A 的条件数
randn (m,n)	生成 m 行 n 列的正态随机矩阵	rref (A)	求矩阵 A 的行最简形式
magic (n)	生成 n 阶魔方矩阵	inv (A)	求矩阵 A 的逆矩阵
hess (A)	生成 Hess 矩阵	det (A)	求矩阵 A 的行列式
sqrtrn (A)	求矩阵 A 的平方根	expm (A)	求矩阵 A 的指数值
funrn (A)	按矩阵 A 计算的函数值	logm (A)	求矩阵 A 的对数值
rank (A)	求矩阵 A 的秩	norm (A)	求矩阵 A 的范数

例 3. 输入矩阵$\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}$和矩阵$\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$.

解　Matlab 的命令为

```
ones (3)
ans = 1   1   1
      1   1   1
          1   1   1
zeros  (2,4)
ans = 0   0   0   0
      0   0   0   0
```

2. 操作符“:”的说明

j :k　表示步长为 1 的等差数列构成的数组：[j,j + 1,j + 2,…,k].

j :i :k　表示步长为 i 的等差数列构成的数组：[j,j + i,j + 2 ∗ i,…,k].

A(i :j)　表示 A(i),A (i + 1), …,A(j).

3. 矩阵的简单运算

加法：+ ；减法：- ；乘法：∗ ；左除：h ；右除：/ ；

乘幂： ；A 的转置：A' ；数乘以 A :k ∗ A ；

A 的行列式：det (A) ；A 的秩：ran (A) ；求 A 的逆：inv (A)

A3.2　数组

在 Matlab 中数组就是一行或一列的矩阵，Matlab 提供了一些创建数组的命令：

命令形式 1：linespace (a, b, n)

功能：把区间 [a, b] 等分成 n 个数据. 即把区间 [a, b] 做 n - 1 等分，公差为$\frac{b-a}{n-1}$.

命令形式 2：logspace (a, b, n)

功能：在区间 $[10^a,10^b]$ 上创建包含 n 个数据的等比数列，公比为 $10^{\frac{b-a}{n-1}}$.

在 Matlab 中，数组的运算和矩阵的运算不同，Matlab 中专门为数组提供了一些特殊的运算：乘法为 ∗ ，左除为 h，右除为 / ，乘幂为 .

设数组 $a = [a_1,a_2,\cdots,a_n]$，$b = [b_1,b_2,\cdots,b_n]$，则对应的运算具体为：

$$a \pm b = [a_1 \pm b_1, a_2 \pm b_2, \cdots, a_n \pm b_n]$$

$$a.*b = [a_1 b_1, a_2 b_2, \cdots, a_n b_n]$$

$$a.k = [a_1^k, a_2^k, \cdots, a_n^k]$$

$$a./b = \left[\frac{a_1}{b_1}, \frac{a_2}{b_2}, \cdots, \frac{a_n}{b_n}\right]$$

$$a.\backslash b = \left[\frac{b_1}{a_1}, \frac{b_2}{a_2}, \cdots, \frac{b_n}{a_n}\right].$$

A3.3 字符串

虽然 Matlab 注重的是矩阵的计算与处理,但是处理字符串的功能还是非常强大的. 在 Matlab 中,字符串用单引号‘ 括起的一串字符表示,如‘ asd ’,‘ 2 + 2 ’,‘ tan (x) ’等都是字符串,即任何内容经过单引号 ’ 括起来,就称为字符串.

注意 字符串不能用双引号“ ”代替单引号.

1. 字符串的输入

字符串通常赋值给变量,这样可以使字符串处理变得简单.

字符串变量 s 的定义命令为:

s = ‘字符串’

字符串矩阵 SA 的定义命令为

SA = [‘字符串 1’,‘字符串 2’,‘…’];

注意 字符串矩阵的每一行字符串元素的个数可以不定,但是字符的总数必须相同,否则系统将出错.

2. 将字符串表达式作为命令执行

命令形式:a = eval (‘字符串表达式’)

功能:该函数返回由字符串表达式执行的结果,也就是求字符串表达式的值. 这个函数在 M 文件中进行交互式执行命令时很有用.

例 4. a = ‘1 + 2’, % 定义 a 为字符串变量.

```
a = 1 + 2
b = eval (a)
b = 3
```

注意 还有一个和 eval 功能相似的函数 feval ().

A4. M 文件与 M 函数

Matlab 有两种常用的工作方式:一种是直接交互的指令操作方式;另一种是

M 文件的编程工作方式. 在前一种工作方式下,Matlab 当做一种高级的“数学演算与图形器”;在后一种工作方式下,M 文件类似于其他的高级语言,是一种程序化的编程语言.

Matlab 中的 M 文件有两种形式:一是命令文件,或称为脚本文件;另一种是 M 函数文件. 它们都是由若干 Matlab 语句或命令组成的文件. 两种文件的扩展名都为 M .

A4.1 命令文件

当用户要运行的指令比较多时,直接从键盘上逐行输入指令比较麻烦,而命令函数文件可以较好地解决这一问题. 用户可以将一组相关指令编辑在同一个命令文件中,运行时只需要输入文件名字,Matlab 就会按顺序执行文件中的命令.

M 文件有两种运行方式:一是在命令窗口直接写入文件名,按 Enter 键;二是在编辑窗口打开菜单 Tools,再单击 Run. M 文件保存的路径一定要在搜索路径上,否则 M 文件不能运行.

例 5. 用 M 命令文件绘出衰减振荡曲线 $y = e^{-\frac{t}{3}}\sin 3t$ 及它的包络线 $y_0 = e^{-\frac{1}{3}}.t$ 的取值范围是 $[0,4\pi]$.

解 (1) 新建一个 M 文件.

(2) 在 M 文件编辑窗口写下下列语句:

```
t = 0 :pi/50 :4 * pi;
y0 = exp(- t/3);
y = exp (- t/3).* sin (3 * t)
plot (t, y, ' - r',t ,y0, ' :b', t, - y0,' :b')
```

(3) 保存文件,并且保存在搜索路径上,文件名为 file1. m. 然后运行该文件即可. 其运行结果如图 A. 3 所示.

M 命令文件中的语句可以访问 Matlab 工作空间中的所有变量与数据,同时 M 命令文件中的所有变量都是全局变量,可以被其他的命令文件与函数文件访问,并且这些全局变量一直保存在内存中,可以用 clear 来清除这些全局变量.

A4.2 函数文件

如果 M 文件的第一行包含关键字 Function,该文件就是 M 函数文件. 每一个 M 函数文件都定义一个函数. M 函数文件实际上是 Matlab 的一个子函数,其作用与其他语言定义的子函数基本相同,都是为了方便实现功能而定义的.

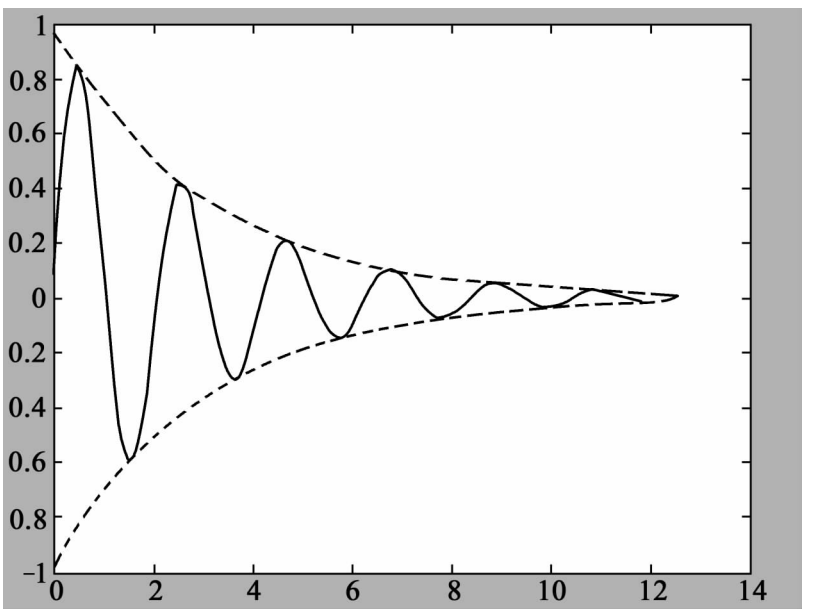


图 A.3　衰减振荡曲线

M 函数文件的一般形式为：

Function〈因变量〉 = 〈函数名〉（〈自变量〉）

函数文件与命令文件的主要区别在于：

(1) 函数文件一般都带有参数，都要有返回结果，而命令文件没有参数也没有返回结果；

(2) 函数文件的变量是局部变量，运行期间有效，运行完毕就自动被清除，而命令文件的变量是全局变量，执行完后仍被保存在内存中；

(3) 函数文件要定义函数名，且保存函数文件的函数名必须是函数名. m.

例 6. 设可逆方阵为 A，试编写同时求 $|A|$，A^2，A^{-1}，A' 的 M 函数文件.

解　(1) 新建一个 M 函数文件.

(2) 在编辑窗口写下下面的代码：

```
function [da, a2, inva , traa] = comp4 (x)
    da = det (x) ;
    a2 = x 2 ;
    inva = inv (x) ;
    traa = x' .
```

(3) 保存该 M 函数文件,文件名为 comp4. M.

(4) 在命令窗口中输入下列语句:

```
A = [1 ,2 ;5, 8] ;
Comp4 (A)
```

其运行结果为

```
 a = - 2
a2 = 11   18
     45   74
inva = - 4.0000    1.0000
       2.5000      - 0.5000
traa = 1   5
       2   8
```

A5. 程 序 结 构

在 Matlab 中,循环由 while 和 for 语句来实现,分支结构由 if 语句来实现.

A5.1 顺序结构

在具有顺序结构的可执行文件中,语句在程序文件中的物理位置就反映了程序的执行顺序.

例 7. 建立一个文件名为 exam2. m 的顺序文件.

解 在 Matlab 中的命令如下:

```
disp ('运行结果') ;
disp ('the first line') ;
disp ('the second line') ;
disp ('the third line') ;
```

其程序运行结果为:

```
>> 运行结果
   the first line;
   the second line;
   the third line.
```

A5.2 循环结构

循环是计算机解决问题的主要手段,许多实际问题都包含规律的重复计算

和对某些语句的重复执行. 在循环结构中,被循环执行的那一组语句就是循环体,每个循环语句都要有循环条件,以判断是否要继续进行下去. Matlab 中的循环语句主要是 for-end 语句和 while-end 语句.

1. for-end 循环结构

for 循环的语法是:

```
for i = 表达式
  可执行语句 1, …,可执行语句 n,
end
```

注意:

(1) 表达式是一个向量,可以是 m: n, m: s: n,也可以是字符串、字符串矩阵等.

(2) for 循环的循环体中,可以多次嵌套 for 和其他的结构体.

例 8. 利用 for 循环求 1 ~ 100 的整数之和.

解 建立命令文件 exam3. m.

```
sum = 0 ;
for i = 1: 1: 100;
sum = sum + i;
  end
  sum.
```

执行命令文件 exam3. m,其程序运行结果为

sum = 5050.

2. while-end 循环结构

while 循环将循环体中的语句执行不定次数.

其基本语法是:

```
while 表达式;
循环体语句;
end.
```

说明:表达式一般是由逻辑运算和关系运算以及一般的运算组成的表达式,以判断循环要继续进行还是停止循环. 只要表达式的值非零,即逻辑为"真",程序就继续循环,只要表达式的值为零就停止循环.

例 9. 试用 while 循环计算 1 !+ 2 !+ … + 50 !.

解 (1) 建立命令函数文件 exam4. m:

```
sum = 0;
i = 1;
```

```
        while i < 51 ;
            prd = 1 ;
            j = 1 ;
            While j <= i
                prd = prd * j ;
                j = j  + 1 ;
    end.
sum = sum + prd ;
i = i + 1 ;
end
Disp ('1 !+ 2 !+ … + 50 ! 的和为 :')
sum.
```

(2) 执行命令文件 exam4. m

```
exam4. m
1 !+ 2 !+ … + 50 ! 的和为
                    sum = 3. 1035e + 064.
```

A5. 3 分支结构

在计算中通常遇到要根据不同的条件来执行不同的语句的情况,当某些条件语句满足时只执行其中的某一条或某几条命令,在这种情况下就要用到分支结构. Matlab 提供了两种分支结构,一种是 if—else—end 语句;另一种是 switch—case—end 语句.

1. if—else—end 语句

该分支结构一般有三种形式:

(1) if 表达式

```
    执行语句
end.
```

功能:如果表达式的值为真,就执行语句,否则执行 end 后面的语句.

(2) if 表达式

```
    执行语句 1
else ;
    执行语句 2
end.
```

功能:如果表达式的值为真,就执行语句 1,否则执行语句 2.

(3) if 表达式 1
　　执行语句 1
　elseif 表达式 2；
　　执行语句 2
　else；
　　⋮
　　执行语句 n
　end.

功能：如果表达式 1 的值为真，就执行语句 1，然后跳出 if 执行语句；否则判断表达式 2，如果表达式 2 的值为真，就执行语句 2，然后跳出 if 执行语句；否则判断表达式 3，依此类推，一直进行下去. 如果所有表达式的值都为假，就执行 end 后面的语句.

注意　假如有一种选择，就使用第一种形式；如果有两种选择，就使用第二种形式；当有三种或更多种的选择时就使用第三种形式.

例 10. 计算函数值

$$f(x)=\begin{cases} x+1, & \text{当 } x<2 \\ 3x, & \text{当 } 2\leqslant x\leqslant 8 \\ 4x-5, & \text{当 } 8<x\leqslant 20 \\ \cos x+\sin x, & \text{当 } x>20 \end{cases}$$

解 (1) 建立 M 函数文件 ff. m：

```
function y = ff (x)
if x<2
   y = x+1;
elseif x>=2 & x<=8
   y = 3*x;
elseif x>8 & x <=20
   y = 4*x - 5;
else
   y = cos (x) + sin (x);
end.
```

(2) 调用 M 函数文件计算 f (0.1)，f (1)，f (9)，f (22)，f (2π)

```
yy = [ff (0.1), f (1), f (9), f (22), f (2*pi)]
yy = 1.1000   2.0000   31.0000   - 1.0088   18.8496.
```

2. switch—case—end 分支结构

switch 语句是多分支语句,虽然在某些场合 switch 的功能可以由 if 语句的多层嵌套完成,但这样会使程序变得复杂,而利用 switch 语句构造多个分支选择则比较简单明了.

Switch 语句的形式为:

```
switch  表达式
        case  常量表达式 1
              语句块 1;
        case  常量表达式 2;
              语句块 2;
              ⋮
        case        常量表达式 n;
                    语句块 n
        otherwise
                    语句块 n + 1. .
   end.
```

功能:switch 语句后面的表达式可以为任何类型;每个 case 后面的常量表达式可以是多个,也可以是不同类型;与 if 语句不同的是,各个 case 和 otherwise 语句出现的先后顺序不会影响程序运行的结果.

例 11. 编写一个转换成绩等级的函数文件,其中成绩等级转换标准为考试成绩分数,在 [90,100] 分数段内显示优秀,在 [80,90) 分数段内显示良好;在 [60,80) 分数段内显示及格;在 [0,60) 分数段内显示不及格.

解 (1) 建立 M 函数文件 ff2. m:

```
function result = ff2 (x)
n = fix (x/10);
switch  n
case {9, 10}
   disp ('优秀');
case 8
   disp ('良好');
case {6, 7}
   disp ('及格');
otherwise
   disp ('不及格')
```

end.

(2) 调用 M 函数文件判断 98 分属于哪个范围：

ff2 (98)

其程序运行结果为：

优秀.

A6. 符号计算

A6.1 符号变量的创建

数学计算有数值计算与符号计算之分. 这两者的根本区别就是:数值计算的表达式、矩阵变量中不允许有未定义的自由变量,而符号计算可以含有未定义的符号变量. 在数值计算过程中,所运作的变量都是被赋了值的数值变量. 而在符号计算的整个过程中,所运作的是符号变量.

注意:在符号计算中,所出现的数字也都是被当做符号来处理的.

A6.2 符号表达式的创建

符号变量是利用指令 sym 和 syms 来创建的,这类指令的使用形式为：

命令形式:S = sym ('A')

功能:定义单个符号变量 S.

命令形式:syms a b c,…

功能:定义多个符号变量 a,b,c,…

注意:用 syms 命令定义符号变量时,变量之间只能用空格隔开,不能用‘,’隔开.

命令 Syms 的使用要比命令 Sym 简便,命令 Syms 一次可以定义多个符号变量.

A6.3 符号方程的创建

创建符号表达式的目的就是把表达式赋值给一符号变量,以方便表达式的使用. 创建符号表达式有两种方法.

(1) 方法 1:直接创建法

该方法直接用 S = sym (‘表达式’) 创建符号表达式.

例 12. 定义表达式 $3ax^3 + 4bx^2 + 2x + 6$ 的符号表达式.

解 Matlab 的命令为：

ff = sym ('3 * a * x 3 + 4 * b * x 2 + 2 * x + 6').

(2) 方法 2 间接创建法

间接创建法是在创建符号表达式之前,先把符号表达式中的所有变量定义为符号变量,然后直接输入表达式.

例 13. 定义表达式 $ax^3 + bx^2 + c$ 为符号表达式,并计算 x = 3 时对应的函数值.

解 Matlab 的命令为:

```
syms a b c x;
f = a * x 3 + b * x 2 + c;
tt = subs (f,'x',3);
vpa (tt);
ans = 27. * a + 9. * b + c.
```

A6.4 符号方程的设计

符号方程和符号表达式不同,表达式只是一个由数字和变量组成的代数式,而方程则是由表达式和等号组成的等式. 符号方程的创建方法只有一种:

命令形式:equ = sym ('equation')

功能:把方程 equation 定义为符号方程.

例 14. 定义方程 $3x = 6a + b$, $x^2 + 2y^2 + 3z^2 = 2$ 为符号方程.

解 Matlab 的命令为:

```
eq1 = sym ('3 * x = 6 * a + b');
eq2 = sym ('x 2 + 2 * y 2 + 3 * z 2 = 2').
```

A6.5 符号矩阵的创建

1. 直接创建法

利用命令 sym,矩阵元素可以是任何符号变量、符号表达式及方程,且元素的长度可以不同.

例 15. 创建符号矩阵 $\begin{pmatrix} \sin 2x & \frac{1}{x} \\ abc & y^2 \end{pmatrix}$.

解 Matlab 的命令为:

```
A = sym ('[sin (x), 1/x ;abc, y 2]')
A = .
```

2. 间接创建法

间接创建符号矩阵的方法为:在创建符号矩阵之前,先把符号矩阵的所有变量定义为符号变量,然后按创建普通矩阵的格式输入矩阵.

例 16. 用间接法创建例 10 的矩阵.

解 Matlab 的命令为:

```
syms x y a b c;
A = [sin(x), 1/x; a*b*c, y^2];
A = [sin(x), 1/x]
    [abc, y^2].
```

A7. Matlab 的绘图

Matlab 有着很强大的数据可视化和图像处理的功能,该软件基本上可以满足一般实际工程、科学计算中的所有图形、图像处理的需要.

A7.1 Matlab 的二维曲线绘图

1. 基本绘图指令 plot

Matlab 软件中指令 plot 是最简单且使用最广泛的一个线性绘图指令,利用该指令可以绘出折线、曲线和参数方程曲线的图形. plot 绘图命令有如下一些常用形式:

命令形式 1:plot(y)

功能:绘一条或多条折线. 其中 y 是数值向量或数值矩阵. 当 y 是数值向量时,plot(y) 在坐标系中顺序地用直线段连接定点(i, y(i))绘出一条折线图;当 y 是数值矩阵时,Matlab 为矩阵的每一列绘出一条折线,绘图时,以矩阵 y 每列元素的相应行下标值为横坐标,以 y 的元素为纵坐标绘制连线图.

例 17. 绘出向量[1, 2, 4, 0, 5, 7]的图形.

解 Matlab 的命令为

```
y = [1, 2, 4, 0, 5, 7];
plot(y).
```

其程序运行结果如图 A.4 所示.

命令形式 2:plot(x, y)

功能:绘出一条或多条折线. 其中 x 可以是长度为 n 的数值向量或 n × m 的数值矩阵,y 也可以是长度为 n 的数值向量或 n × m 的数值矩阵.

(1) 当 x,y 都是长度为 n 的数值向量时,plot(x, y) 在坐标系中顺序地用

直线段连接点 (x (i) , y (i)) ,绘出一条折线图.

(2) 当 x 是长度为 n 的数值向量且 y 是 n × m 的数值矩阵时,plot (x, y) 用向量 x 分别与矩阵 y 的每一列匹配,在同一坐标系中绘出 m 条不同颜色的折线图.

(3) 当 x 和 y 都是 n × m 的数值矩阵时,plot (x, y) 分别用矩阵 x 的第 i 列与矩阵 y 的第 i 列匹配,在同一坐标系中绘出 m 条不同颜色的折线图.

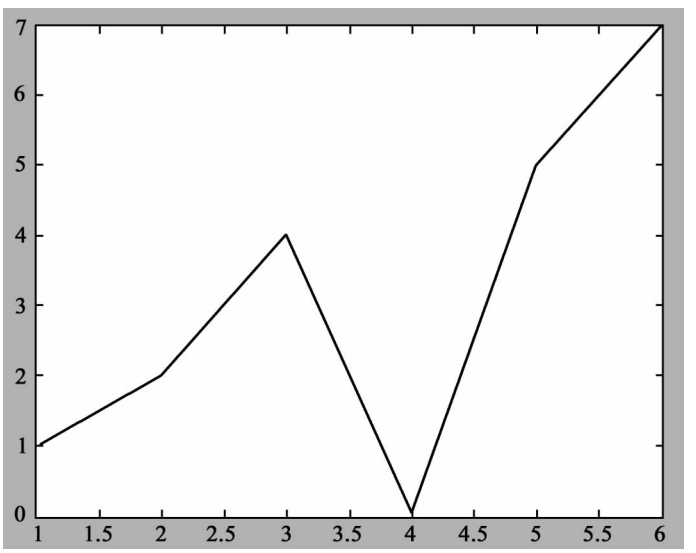


图 A. 4

注意:命令 plot (x, y) 可以用来绘通常的函数 f (x) 的图形,此时向量 x 常用命令

$$x = a: h: b$$

的形式获得 f (x) 在绘图区间 [a ,b] 上的自变量点向量数据. 对应的函数向量的取值为

$$y = f(x)$$

这里的步长 h 可以任意选择.

例 18. 绘出函数 $y = \cos x^2$ 在区间 $-4 \leqslant x \leqslant 4$ 的图形.

解 Matlab 的命令形式为:

```
x = - 5:1:5;
y = cos (x . ^2);
```

plot (x, y) ;grid on ;

其程序运行结果如图 A. 5 所示.

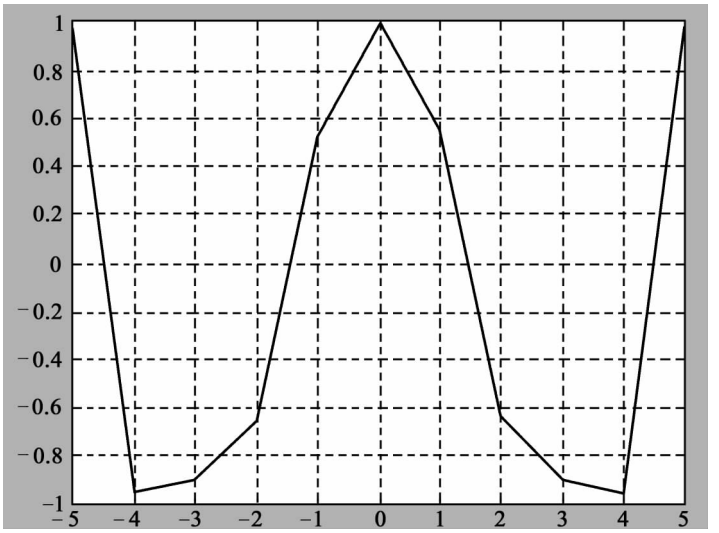


图 A. 5

命令形式 3：plot (x1 ,y1 ,x2 , y2 , …)

功能：在同一窗口绘出多条不同颜色的曲线,这组曲线的关系为

$y_1 = f(x_1)$, $y_2 = f(x_2)$,….

例 19. 在同一图像窗口绘出三个函数 $y = \cos 2x$, $y = x^2$, $y = x$ 的图形, $-2 \leqslant x \leqslant 2$.

解 Matlab 的命令形式为：

```
x = - 2:0.1:2 ;
plot (x, cos (2 * x) , '. ', x, x .^2,'k - ' , x, x , 'k') ;
legend ('cos (2x) ', 'x ∧2 ', 'x') ;
```

其程序运行结果如图 A. 6 所示.

2. 基本绘图控制参数

图形窗口 figure：

figure 是所有 Matlab 的图形输出的专用窗口. 可以用 figure 命令自动创建窗口,使用方法如下：

figure 打开一个新窗口；

figure (n) 打开第 n 个图形窗口.

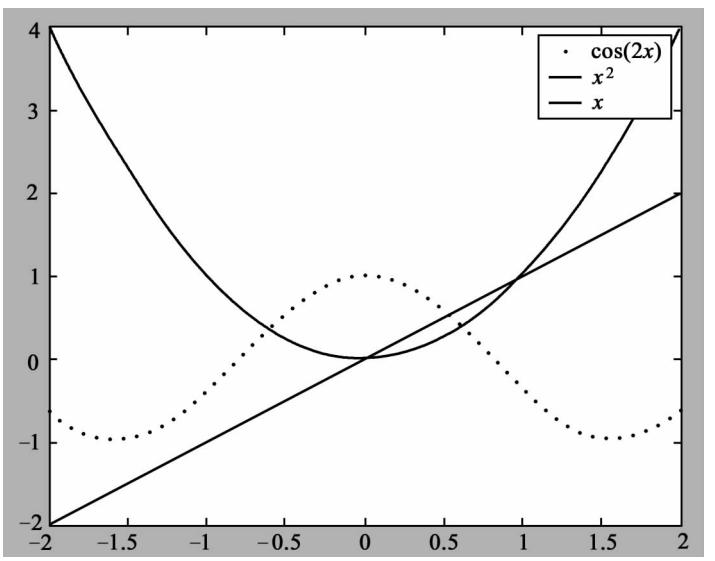


图 A. 6

还有一些其他的常用命令如下：

(1) 清除图形窗口命令 clf.

(2) 控制分隔线命令 grid：

grid 在 grid on 与 grid off 之间进行切换；

grid on 在图中使用分隔线；

grid off 在图中消隐分隔线.

(3) 图形的重叠绘制 hold：

hold 在 hold on 与 hold off 之间进行切换；

hold on 保留当前图形和该图的轴，使此后的图形叠放在当前图形上；

hold off 返回 Matlab 的缺省状态，此后图形指令运行将抹掉当前窗口中的旧图形，然后绘出新图形.

(4) 取点指令 ginput.

该命令的作用是在二维图形中记录下鼠标所选点的坐标值. 使用形式有下面两种：

ginput 可以无限制地选点，当选择完毕时，按 Enter 键结束命令.

ginput (n) 必须选择 n 个点才可以结束命令.

(5) 图形缩放指令 zoom：

zoom 在 zoom on 和 zoom off 之间进行切换；

zoom on 使系统处于可放大状态；

zoom off 使系统回到非放大状态，但前面放大的结果不会改变；

zoom out 使系统回到非放大状态，并将图形恢复原状；

zoom xon 对 Ox 轴有放大作用；

zoom yon 对 Oy 轴有放大作用.

3. 线型、定点标记、颜色

函数 plot 可以接受字符串输入变量，这些字符串输入变量指定不同的线型、数据点的标志符以及每条曲线的颜色.

命令形式 1：plot（x，‘String’）

命令形式 2：plot（x，y，‘String’），其中 y = f (x)

命令形式 3：plot（x1，y1，‘String1’，x2，y2，‘String2’，…）

其中，String 是由 1 ~ 3 个字母组成的字符串，用来指定所绘制曲线的线型、颜色和数据点标志. String 为空时，表示按系统缺省的定义进行处理.

例如：plot（x，y，‘- ro’）. 表示：将绘制一条红色实线，并且在每个数据点上，都用符号“0”进行标记.

Matlab 中控制线型、颜色和标记点的设置如表 A. 4 ~ 表 A. 6 所示.

表 A. 4 线型控制字符表

线型符号	线　型
-	实　线
:	点　线
- .	点 划 线
- -	虚　线

表 A. 5 颜色控制字符表

颜色字符	颜　色	颜色字符	颜　色
y/yellow	黄　色	g/green	绿　色
m/magenta	洋　红	b/blue	蓝　色
c/cyan	青　色	w/white	白　色
r/red	红　色	k/black	黑　色

表 A.6 数据点控制字符表

绘图字符	数据点	绘图字符	数据点
●	黑　　点	d	菱形标记
O	小圆圈		朝上的三角形
X	叉型标记	<	朝左的三角形
+	加号标记	>	朝右的三角形
*	星号标记	p	五角星符号
S	正方形标记	h	六角星符号

4. 图形的标注

Matlab 可以在绘出的图形上加各种标注及文字说明. 它们的实现命令如下：

(1) 图名标注 title

命令形式：title ('string')

功能：在当前图形的顶端加注文字 String 为图名.

(2) 坐标轴标注 xlabel，ylabel，zlabel

命令形式：xlabel ('String') 或 ylabel ('String') 或 zlabel ('String')

功能：在当前图形的 Ox 或 Oy 或 Oz 轴旁边加注文字.

(3) 图形标注 text，gtext

命令形式：text (x，y，'String') 或 text (x，y，z，'String')

功能：在二维图形的点 (x，y) 处加注文字 String (或三维图形的点 (x，y，z) 处加注文字 String).

(4) 图例标注 legend

当在一幅图中出现多种曲线时，结合在绘制时的不同线型与颜色等特点，可以用 legend 命令进行说明.

命令形式：legend ('String1'，'String2'，…)

功能：对当前图形进行图例标注.

例 20. 在同一坐标系中绘出两个函数 y = cos2x，y = x 的图形，自变量范围为 $-2 \leqslant x \leqslant 2$；函数 y = cos2x 为红色实线，函数 y = x 为洋红色虚线，并加注标题、坐标轴和图例标注.

解　(1) 建立命令文件 exam5.m：

```
clf;
x = -2:1:2;
y1 = cos(2*x);y2 = x;
plot(x,y1,'-r',x,y2,'-.m');
grid on;
```

```
title ('曲线 y = cos (2x) 与 y = x');
xlabel ('x 轴');ylabel ('y 轴');
x00 = - 2:2;
y00 = [1.5, 1.0, 0.56, - 1.5];
hold on, plot (x00,y00,'bp');
legend ('y = cos (2x)','y = x','5 点图').
```

(2) 执行命令文件 exam5. m,其程序运行结果如图 A. 7 所示.

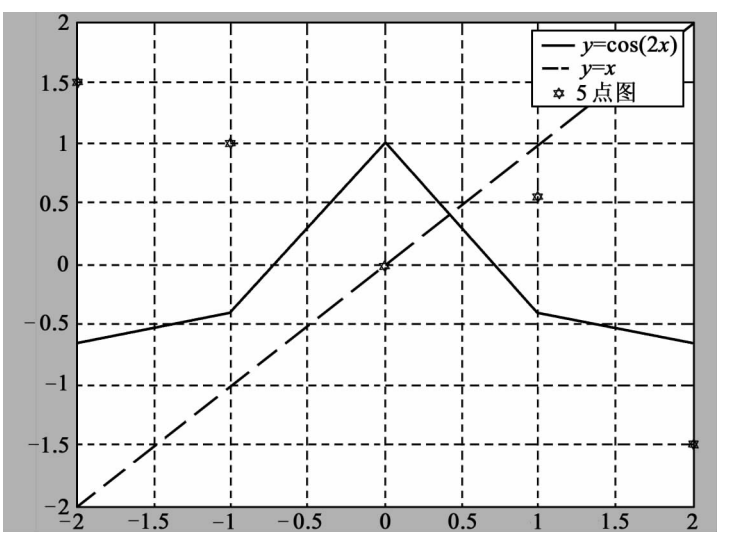


图 A. 7 曲线 y = cos (2x) 与 y = x 及点图

5. 一个图形中多个子图的绘制

可以在 Matlab 图形窗口中显示多个图形,要实现该功能就要利用函数 subplot.

命令形式:subplot (m, n, p)

功能:把当前图形窗口分为 m × n 个子图,并把第 p 个子图当做当前图形窗口.

例 21. 把一个图形窗口分为两个子图,每个子图绘制不同的图形.

解 Matlab 的命令如下:

```
clf;
x = - 2 :0.2 :2
y1 = x + sin (x);
```

```
y2 = sin (x) . / x ;
subplot (1 ,2 ,1) ;plot (x, y1, ' - m . ') ;
grid on ;title (' y = x + sinx ') ;
subplot (1, 2, 1) ;plot (x, y2, ' - rp ') ;
grid on ;title (' y = sinx/x ') ;
```

其程序运行结果如图 A. 8 所示.

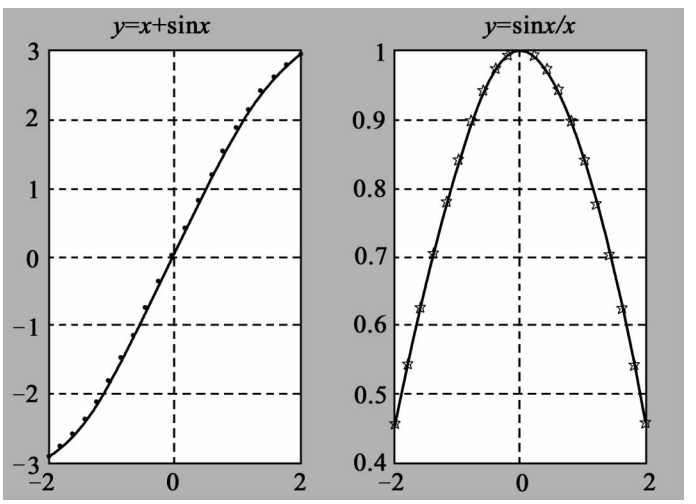


图 A. 8

6. 绘制数值函数二维曲线的命令 fplot

绘图指令 plot (x,y) 在绘图时,必须先定义自变量的一组取值点,再求出这组数据点对应的函数值,然后根据这组数据点绘制出所需的曲线. 而指令 fplot 的特点是:其绘图数据点是自适应产生的. 在函数曲线平坦处,函数所取的数据点比较稀疏;在函数曲线变化剧烈处,函数将自动取较密的数据点. 因而对于导数变化比较大的函数,用 fplot 指令比用 plot 指令效果要好. fplot 的使用形式为:

命令形式:fplot (fun, [xmin, xmax])

功能:绘函数 fun 在区间 [xmin, xmax] 的图像. 其中 fun 是函数名,可以是 Matlab 已有的函数,也可以是自定义的 M 函数,还可以是字符串函数;[xmin, xmax] 定义 x 的取值区间.

7. 绘制符号函数二维曲线的命令 ezplot

绘符号函数二维曲线的命令是 ezplot,该命令的使用形式是:

命令形式:ezplot (fun, [xmin, xmax])

功能:绘出符号函数 fun 在区间 [xmin, xmax] 内的图像.

注意:fun 必须是符号函数并且只含有一个变量. 如果区间 [xmin, xmax] 缺省,则默认区间为 [- 2πi, 2πi].

例 22. 绘出 $y = \frac{2}{3}e^{-\frac{t}{2}}\cos\frac{3}{2}t$ 在 [0,4π] 间的图形.

解 Matlab 命令为:

```
syms t;
ezplot ('2/3 * exp (- t/2) * cos (3/2 * t)', [0,4 * pi])
```

其程序运行结果如图 A. 9 所示.

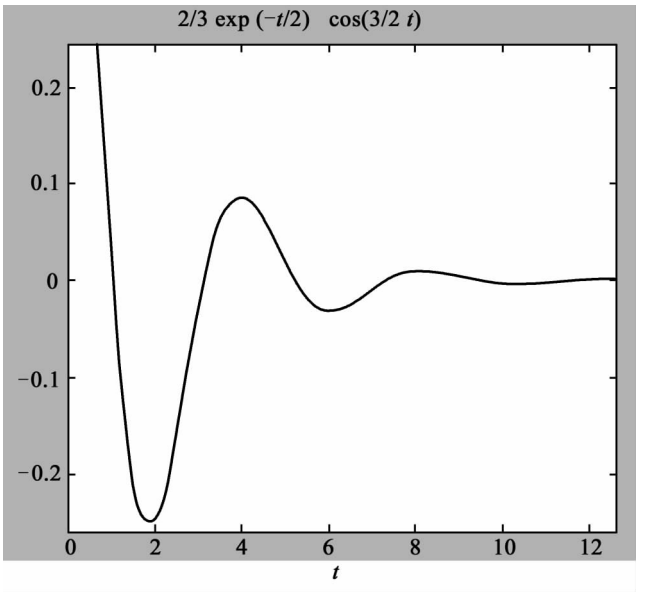


图 A. 9

A7. 2 Matlab 中绘制特殊图形的命令

除了指令 plot 外,Matlab 还提供了许多其他的绘图指令,绘制特殊图形的指令如表 A. 7 所示.

表 A.7 绘制特殊图形的指令表

函数名称	功 能	函数名称	功 能
bar	直方图	semilogx	Ox 轴对数坐标曲线
barh	垂直的直方图	semilogy	Oy 轴对数坐标曲线
bar3	三维直方图	polar	极坐标曲线
bar3h	垂直的三维直方图	stairs	阶梯图
hist	统计直方图	stem	火柴棍图
pie	饼图	pcolor	伪彩图
pie3	三维饼图	area	面积图
gplot	绘制拓扑图	errorbar	误差图
fill	平面多边形填色	quiver	矢量场图
loglog	双对数曲线	ribbon	带状图

如下面的程序就是绘制特殊图形的例子：

```
x = 1:5
subplot (2, 3, 1) , bar (x) , title ('直方图');
subplot (2, 3, 2) , stairs (x) , title ('阶梯图')
subplot (2, 3, 3) , stem (x,'rp') , title ('火柴棍图');
subplot (2, 3, 4) , pie (x) , title ('饼图');
subplot (2, 3, 5) , pie3 (x) , title ('三维饼图');
subplot (2, 3, 6) , area (x) , title ('面积图').
```

其程序运行结果如图 A.10 所示.

A7.3 Matlab 的空间曲线绘图

空间曲线的方程为 x = x (t) ,y = y (t) ,z = z (t) ,参数 t 连接变量 x,y,z 的函数关系. Matlab 中绘制空间曲线的函数是 plot3.

1. 三维空间曲线命令 plot3

函数 plot3 与函数 plot 完全相同,其命令形式如下：

命令形式 1：plot3 (x, y, z)

命令形式 2：plot3 (x, y, z, 'String')

命令形式 3：plot3 (x1 , y1 ,z1 ,'String1 ',x2 , y2 , z2 , 'String2 ',…)

功能：当 x、y、z 为长度相同的向量时,plot3 命令将绘得一条分别以向量 x、y、z 为坐标值的空间曲线. String 用来控制曲线的颜色、线型和数据点形式. 命令形式 3 是在同一图形窗口绘多条空间曲线.

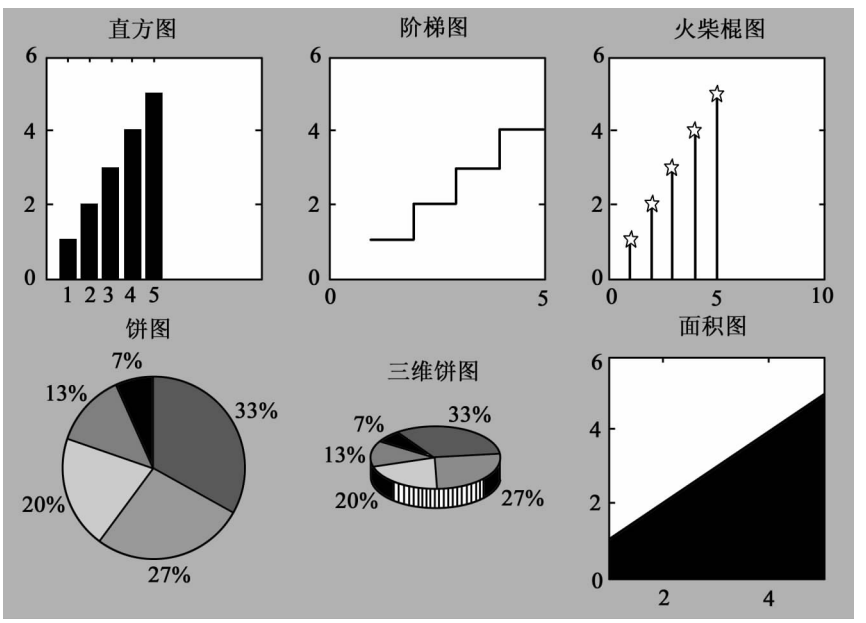


图 A.10　特殊绘图函数

例 23. 绘制螺旋线 $\begin{cases} x = \sin t \\ y = \cos t \\ z = t \end{cases}$，$(0 \leqslant t \leqslant 10\pi)$ 的图像.

解　在 Matlab 中的命令如下：

```
t1 = 0: pi/25: 10 * pi
x1 = sin (t1);
y1 = cos (t1);
z1 = t1;
plot3 (x1, y1, z1, 'r');
title ('螺旋线'), xlabel ('x 轴'), ylabel ('y 轴'), zlabel ('z 轴'),
```

其程序运行结果如图 A.11 所示.

2. 坐标轴的控制

在 Matlab 中用 axis 来完成对坐标轴的控制，其使用形式为：

命令形式 1：axis ([xmin, xmax, ymin, ymax])

功能：设定 Ox 轴的范围为 [xmin, xmax]，Oy 轴的范围为 [ymin, ymax].

命令形式 2：axis ([xmin, xmax, ymin, ymax, zmin, zmax])

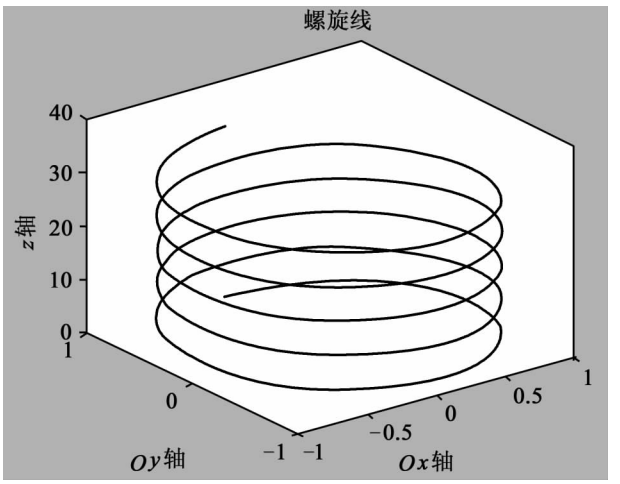


图 A.11　螺旋线

命令形式 3：axis auto

功能：将坐标轴的取值范围设为默认值.

命令形式 4：axis ij

功能：坐标原点设置在图形窗口的左上角，坐标轴 i 垂直向下，j 水平向右.

命令形式 5：axis xy

功能：设定为笛卡儿坐标系.

命令形式 6：axis equal

功能：使坐标轴在三个方向上长度相同.

命令形式 7：axis on

功能：恢复消隐的坐标轴.

命令形式 8：axis off

功能：使坐标轴消隐.

A7.4　Matlab 的空间曲面绘图

二元函数 z = f (x,y) 的图形是三维空间曲面，空间曲面图形在帮助人们了解二元函数特性上具有较大作用. 现介绍绘制空间曲面图形的命令.

1. meshgrid 命令

二元函数 z = f (x,y) 的图形是三维空间曲面，在 Matlab 中总是假设函数 z =

f (x,y) 是定义在矩形区域 $D = [x_0, x_m] \times [y_0, y_n]$ 上的. 为了绘制三维曲面, Matlab把 $[x_0, x_m]$ 分成 m 份,把 $[y_0, y_n]$ 分成 n 份,这时区域被分成小矩形块. 每个小矩形块有 4 个顶点,连接四个顶点得到一个空间中的四边形片. 所有这些四边形就构成函数的空间网格曲面. 而函数 meshgrid 就用来生成 xOy 平面上的小矩形顶点坐标值的矩阵,也称为点格矩阵. 函数 meshgrid 也适用于三元函数 u = f (x,y,z) .

meshgrid 的调用形式是:

(1) [X, Y] = meshgrid (x, y) ,绘制二维图形时生成小矩形的格点.

(2) [X, Y] = meshgrid (x) ,等价于 [X, Y] = meshgrid (x, x)

(3) [X, Y, Z] = meshgrid (x, y, z) ,绘制三维图形时生成空间曲面的格点.

(4) [X, Y, Z] = meshgrid (x) ,等价于 [X, Y, Z] = meshgrid (x, x, x) .

说明:x 是区间 $[x_0, x_m]$ 上划分点组成的 m 维向量,而 y 是区间 $[y_0, y_m]$ 上划分点组成的 n 维向量. 输出变量 X 与 Y 都是 n × m 矩阵,而矩阵 X 的行向量都是向量 x,矩阵 Y 的列向量都是向量 y.

2. 三维网格图命令 mesh

由函数 meshgrid 生成格点矩阵后,就可以求出各格点对应的函数值,然后利用三维网格图命令 mesh 与三维表面图命令 surf 绘出空间曲面. 函数 mesh 用来生成函数网格曲面,即各网格线段组成的曲面,而函数 surf 用来生成函数的表面曲面,即对网格曲面的网格块区域进行了着色.

函数 mesh 的命令形式如下:

(1) mesh (X, Y, Z) ,X, Y, Z 是同维数的矩阵.

(2) mesh (x,y,Z) , x, y 是向量,而 Z 是矩阵,等价于

$$\begin{cases} [X,Y] = \text{meshgrid}(x,y) \\ \text{mesh}(X,Y,Z) \end{cases}$$

(3) mesh (Z) , 若提供参数 x, y, 等价于 mesh (x, y, Z) , 否则默认 x = 1 :n, y = 1 :m.

另外还有两个与 mesh 相似的函数 meshc 和 meshz:meshc 和 meshz 的调用方式与 mesh 相同. meshc 除了生成网格曲面外,还在 xOy 平面上生成曲面的等高线图形,而函数 meshz 的作用除了生成与 mesh 相同的网格曲面之外,还在曲面下面加上一个长方形的台柱.

例 24. 分别用指令 mesh, meshc, meshz 绘出函数

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

在 $-8 \leqslant x, y \leqslant 8$ 上的图形.

解 在 Matlab 中的程序如下：

```
t = - 8:0.3:8
[x, y] = meshgrid (t) ;
r = sqrt (x. 2 + y. 2) + eps ;
z = sin (r) . /r ;
subplot (1, 3, 1) , meshc (x, y, z) ;
title ('meshc') ,axis ([- 8 8 - 8 8 - 0.5 0.8]) ;
subplot (1, 3, 2) , meshz (x, y, z) ;
title ('meshz') ,axis ([- 8 8 - 8 8 - 0.5 0.8]) ;
subplot (1, 3, 3) ,k mesh (x, y, z) ;
title ('mesh') ,axis ([- 8 8 - 8 8 - 0.5 0.8]) .
```

其程序运行的结果如图 A.12 所示。

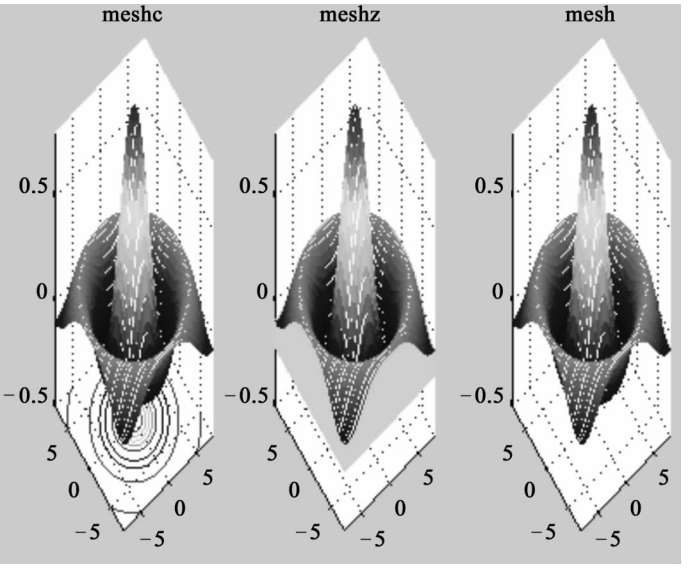


图 A.12 三维网格图

3. 三维表面图命令 surf

surf 的调用方式与 mesh 相同，与 mesh 不同的是 surf 绘制的是曲面而不是网格.

例 25. 绘出函数 $z = 3 - x^2 - y^2$，$-1 \leqslant x, y \leqslant 1$ 的三维网格图和三维表面图.

解 Matlab 的命令为：

```
t = -1:0.1:1
[x, y] = meshgrid (t);
z = 3 - x.^2 - y.^2.
subplot (1, 2, 1), mesh (x, y, z), title ('网格图');
subplot (1, 2, 2), surf (x, y, z), title ('表面图').
```

其程序运行结果如图 A.13 所示.

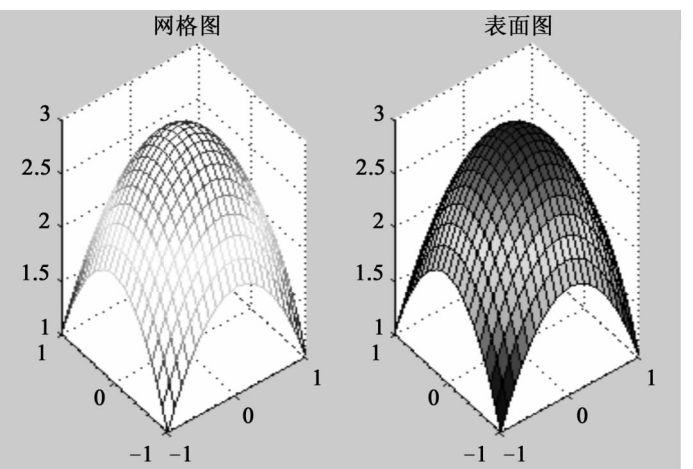


图 A.13 三维表面图

4. 球面的绘制

在 Matlab 中用命令 sphere 来绘制球面，其使用形式为：

命令形式 1：sphere (n)

功能：绘制一个单位球面，且球面上分格线条数为 n.

命令形式 2：[x, y, z] = sphere (n)

功能：x, y, z 是返回的 (n + 1) × (n + 1) 阶矩阵，且 surf (x, y, z) 正好为单位球面.

例 26. 试绘出函数 $x^2 + y^2 + z^2 = 1$ 与 $x^2 + y^2 + z^2 = 4$ 的图形.

解 Matlab 中的命令如下：

```
% 绘半径为 1 的球面
v = [-2 2 -2 2 -2 2]; subplot (1, 2, 1);
sphere (30), title ('半径为 1 的球面'),
```

```
axis (v) ;
% 绘半径为 2 的球面
[x, y, z] = sphere (30) ;
subplot (1,2,2), surf (2 * x, 2 * y, 2 * z) ;
title ('半径为 2 的球面') ;axis (v) ;
```

其程序运行结果如图 A. 14 所示.

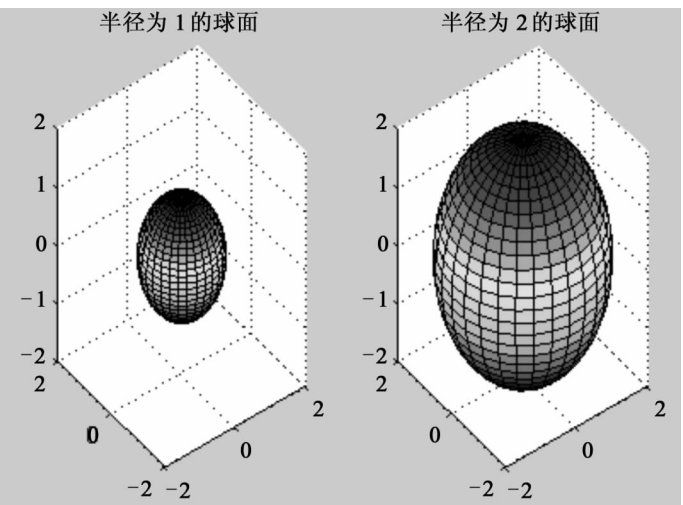


图 A. 14 球体图

5. 色彩控制

(1) colormap

在 Matlab 中,主要由函数 colormap 来完成对图形色彩与表现的控制.

命令形式 1:colormap ([R, G, B])

功能:用单色绘图, [R ,G, B] 代表一个配色方案,R 代表红色,G 代表绿色,B 代表蓝色,且 R. G,B 必须在 [0,1] 区间内.

命令形式 2:colormap (T)

功能:T 是一个 m × 3 的色图矩阵 (也就是调色板),用多种颜色绘图.

常用的调色板名称及其产生的函数如表 A. 8 所示.

表 A.8 调色板名称及产生函数

调色板名称	产生函数
蓝色调灰色调色板	bone
青红浓淡色调色板	cool
线性纯铜色调色板	copper
红白蓝黑交错调色板	flag
线性灰度调色板	gray
黑红黄白色调色板	hot
饱和色调色板	hsv
色图的变体调色板	jet
粉红色调色板	pink
光谱色调色板	prism

(2) 图形的透视命令函数 hidden 与光照控制命令函数 shading

函数 hidden 和 shading 的使用方式如下：

命令形式：hidden on

功能：消隐重叠线.

命令形式：hidden off

功能：透视重叠线.

命令形式：shading flat

功能：网格线的分块着色.

命令形式：shading faceted

功能：默认的着色方式,网格线是黑色.

命令形式：shading interp

功能：着色光顺性最好.

例 27. 着色性能的例子.

解 在 Matlab 中的命令如下：

```
[x0 ,y0, z0 ;] = sphere (30 ) ;
X = 2 * X0 , Y = 2 * Y0, Z = 2 * Z0 ;
% 绘一半径为 1 的单位球.
surf (X0, Y0 ,Z0) ;
% 使单位球的着色光顺性最好.
shading interp ;
% 在同一图形窗口中再绘一个半径为 2 的球.
hold on ;mesh (X, Y, Z) ;
```

```
colormap (hot) , hold off;
% 透视重叠线.
hidden off;
% 消隐坐标轴.
axis equal, axis off
```

其程序运行结果如图 A. 15 所示.

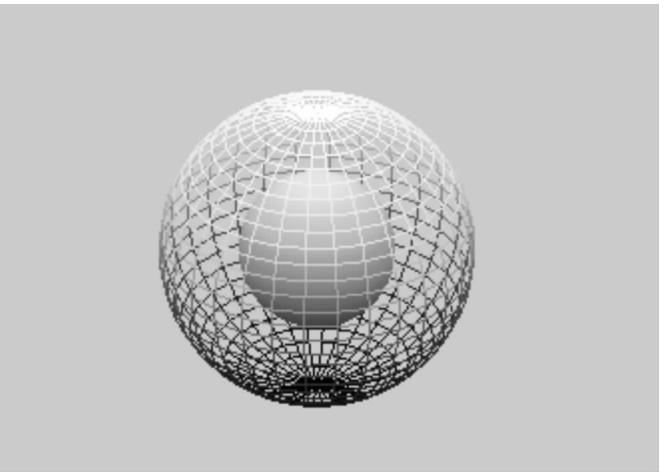


图 A. 15 着色性能的例子

B 高级 Matlab 图形编程——句柄图形

句柄图形已经深入到 Matlab 可视化功能的内核，了解句柄图形，第一可以更深入的理解高层绘图指令，绘制出更精细、更生动、更具个性化的图形；第二可以利用低层图形指令和图形对象属性开发专用绘图函数.

句柄图形 (Handle Graphics) 是一种面向对象的绘图系统，该系统提供创建计算机图形所必须的各种软件. 这类软件所支持的指令，可以直接创建线、文字、网线、面以及图形用户界面. 句柄图形也被称为低层图形. 低层图形使用起来，不像高层指令那样数学概念清晰、调用格式简明易懂. 但低层指令直接操作绘图要素，可以更细致、更具个性地表现图形，更自然、贴切地展现应用场合的物理意义. 下面通过例子来详细地讲述 Matlab 中句柄图形的使用方法.

B1. 连续变焦和飞驰图形

在高层指令中有 zoom 指令，可以使图形放大或缩小. 其实，这种变焦效果很容易通过照相机属性 CameraPosition 或 CameraViewAngle 的设置实现. 在下面例子中将讲述如何通过连续变焦而产生飞驰效果.

例 1. 通过 CameraPosition 设置的不断变化，使地球对着 Camera 迎面飞来，从 Camera 中贯穿而过，然后从 Camera 的另一面飞离而去. 但在整个飞行过程中，照相机镜头始终对着地球.

解 (1) 编写三维地球生成程序 earth_zzy. m，程序代码如下：

[earth_zzy. m]

```
function earth_zzy (ap)

load topo;
figure ('colormap',topomap1,'Color',[.8 .8 .8]);
[x,y,z] = sphere (50);
azzy. DataAspectRatio = [1 1 1];
azzy. PlotBoxAspectRatioMode = 'auto';
```

```
fa = axes ('Visible','off',azzy);
szzy1.AmbientStrength = 0.1;
szzy1.DiffuseStrength = 1;
szzy1.SpecularColorReflectance = .5;
szzy1.SpecularExponent = 20;
szzy1.SpecularStrength = 1;
surface (x, y, z, szzy1, 'FaceLighting', 'phong', 'FaceColor', 'texture',
'EdgeColor','none','Cdata',topo,'Parent',fa);
if ap ==1, set (fa,'CameraViewAngle',0.1*get (fa,'CameraViewAngle'));end
light ('position', [-1 0 1],'color', [0.5 1 0.5]);
light ('position', [-1.5 0.5 -0.5],'color', [0.6 0.2 0.2]);
light ('position', [1.5,1.5,-1]);
light ('position', [0 -1.5 0],'color', [0.6 0.6 1]);
view ([-17 26]).
```

(2) 运行 earth_zzy 可以绘出如下三维地球图,如图 B.1 所示.

```
earth_zzy (0)
```

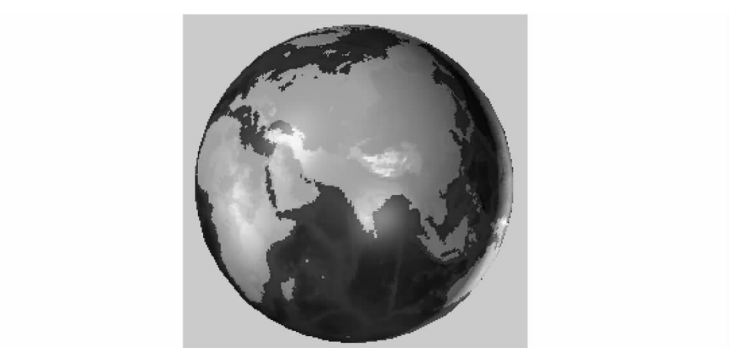


图 B.1 三维地球图形

(3) 编写飞驰程序 fly_zzy.m,程序代码如下:

[fly_zzy.m]

```
% fly_zzy.m 连续改变相机的位置,产生贯穿地球的效果.
earth_zzy (0);
set (gca,'CameraViewAngleMode','manual');
pos = get (gca,'CameraPosition');          % 获取照相机初始位置.
```

```
tar = get (gca,' CameraTarget ') ;          % 获取照相机目标位置.
kk = (0 :2 :40) /15 ;nk = length (kk) ;
for i = 1 :nk - 1
    newpos = pos - kk (i) * (pos-tar) ;
    set (gca,' CameraPosition ',newpos) ;  % 设置照相机新位置.
    drawnow
end.
```

(4) 在 Matlab 的指令窗口运行以下指令,就会产生贯穿地球的效果.

```
fly_zzy.
```

例 2. 利用照相机属性 CameraViewAngle 产生飞驰效应. 要求也用地球图像演示,在照相机视角连续变化下,地球飞离,直到消失.

解 在 Matlab 中实现该程序的代码如下:

[fly_zzy2. m]

```
% fly_zzy2. m    飞离而去的地球消失在远方.
earth_zzy (1) ;
set (gca,' CameraViewAngleMode ',' manual ') ;
ang = get (gca,' CameraViewAngle ') ;
kk = (1 :50) /50 ;nk = length (kk) ;
for i = 1 :nk
    newang = ang + kk (i) * (180 - ang) ;
    set (gca, ' CameraViewAngle ',newang) ;
    drawnow
end.
```

B2. 实 时 动 画

在所有图形对象中,"线"、"面"、"块"、"像"、"字"5 种对象都有一个控制重画方式的属性"EraseMode". 在 Matlab 中,常见的动画有两种:

(1) 影片动画:预先制作图形,存放在图形缓冲区,然后逐帧播放. 这种动画适用于难以实时快速绘制的复杂画面. 这种方法计算量大、占用内存多.

(2) 实时动画:保持图形窗口绝大部分的像素颜色不变,而只更新部分像素的颜色构成运动图像. 这种动画适用于每次变化较少、图形精度要求不很高的场合.

B 2.1 擦除属性‘EraseMode’

如何显示新对象,擦除旧对象,而又不破坏背景图案,这对一般计算机编程语言来说不是一件容易的事情. 但在 Matlab 图形系统中的擦除属性‘EraseMode’,能比较容易地实现上述过程.‘EraseMode’的属性值如表 B. 1 所示.

表 B. 1

可选值	含　义
Normal	计算整个画面的数据,重画整个图形. 这种模式产生的图形最准确,但最慢.
Background	将旧对象的颜色变为背景颜色,实现擦除. 这种模式将损坏被擦对象下面的对象. 但新对象总能正确着色.
None	不作任何擦除. 在该模式下,显示图形不能被正确打印.
Xor	异或方式;对象的绘制和擦除由该对象颜色与屏幕颜色的异或而定. 只绘与屏幕色不一致的新对象点;只擦除与屏幕色不一致的原对象点. 该方式不损坏被擦对象下面的其他图像.

B 2.2 屏幕刷新指令 drawnow

当新对象属性设置后,应刷新屏幕,使新对象显示出来. 这些操作依靠指令 drawnow 完成. Drawnow 指令迫使 Matlab 暂停目前的任务序列而去刷新屏幕. 若没有 drawnow 指令,Matlab 要待任务序列执行完后才会去刷新屏幕.

B 2.3 动画制作示例

制作实时动画的基本思路:

(1) 画出初始图形.

(2) 计算活动对象的新位置,并在新位置上将活动对象显示出来.

(3) 擦除原位置上原有对象,刷新屏幕.

(4) 重复前面两个步骤.

例 3. 制作红色小球沿一条带封闭路径的下旋螺旋线运动的实时动画.

解　程序代码如下:

(1) 编写函数文件 anim_zzy1. m.

```
function f = anim_zzy1 (K,ki)
```

```
t1 = (0 :1000) /1000 * 10 * pi ;
x1 = cos (t1) ;
y1 = sin (t1) ;
z1 = - t1 ;
t2 = (0 :10) /10 ;
x2 = x1 (end) * (1 - t2) ;
y2 = y1 (end) * (1 - t2) ;
z2 = z1 (end) * ones (size (x2)) ;

t3 = t2 ;
z3 = (1 - t3) * z1 (end) ;
x3 = zeros (size (z3)) ;
y3 = x3 ;
t4 = t2 ;x4 = t4 ;y4 = zeros (size (x4)) ;z4 = y4 ;

x = [x1 x2 x3 x4] ;
y = [y1 y2 y3 y4] ;
z = [z1 z2 z3 z4] ;
plot3 (x,y,z,' b ') ;
axis off.
h = line (' Color ', [1 0 0] ,' Marker ',' . ',' MarkerSize ',40,' EraseMode ',' xor ') ;
n = length (x) ;i = 1 ;j = 1.

while 1
    set (h,' xdata ',x (i) ,' ydata ',y (i) ,' zdata ',z (i) ) ;
    drawnow
    pause (0. 0005) ;
    i = i + 1 ;
    if nargin = = 2 & nargout = = 1
        if (i = = ki & j = = 1) ;
        f = getframe (gcf) ;
      end
    end
    if i > n
```

```
        i = 1 ;j = j + 1 ;
        if j > K;
                break ;
        end
    end
end.
```

(2) 在指令窗口中运行以下指令,就可以看到实时动画图形:

```
f = anim_zzy1 (2 ,450) .
```

(3) 显示拍摄的照片,如图 B. 2 所示.

```
image (f,cdata) ,axis off.
```

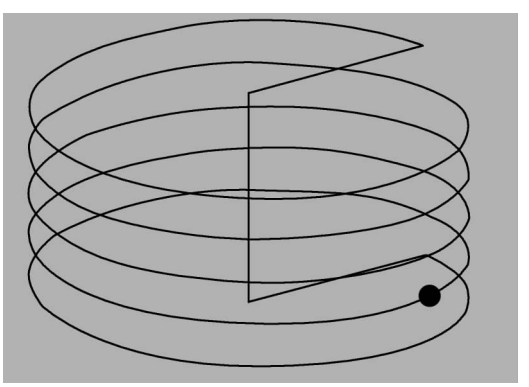


图 B. 2 红球沿下旋螺线运动的瞬间

例 4. 鼠标拖动字符对象,要求实现两个功能:(1) 把当前图形窗口中已存在的任何字对象拖放到所需的位置;(2) 向当前图形窗口输入字对象,并拖放到任何所需位置.

解 程序在 Matlab 中的程序代码如下:

```
[textzzy. m]
function textzzy (arg)

if ~ nargin ;arg = 0 ;end
if ischar (arg) | iscell (arg)
    PT. Units = ' normalized ' ;
    PT. Position = [0. 01 0. 01] ;
```

```
        PT. String = arg ;
        PT. HorizontalAlignment = ' left ' ;
        PT. VerticalAlignment = ' baseline ' ;
        ht = text (PT) ;
        textzzy (0) ;
    elseif arg = = 0
        hf = get (0 , ' CurrentFigure ') ;
        if isempty (hf)
            error (' 图形窗不存在 ') ;
        end.
        PF1. BackingStor = ' off ' ;
        PF1. WindowButtonDownFcn = ' textzzy (1) ' ;
        set (hf,PF1) ;
        figure (hf) ;
    elseif arg = = 1 & strcmp (get (gco, ' Type ') , ' text ') ;
        PO1. Units = ' data ' ;
        PO1. EraseMode = ' xor ' ;
        PO1. HorizontalAlignment = ' left ' ;
        PO1. VerticalAlignment = ' baseline ' ;
        set (gco,PO1) ;
        PF2. Pointer = ' fleur ' ;
        PF2. WindowButtonMotionFcn = ' textzzy (2) ' ;
        PF2. WindowButtonUpFcn = ' textzzy (999) ' ;
        set (gcf,PF2) ;
    elseif arg = = 2
        curpoi = get (gca , ' CurrentPoint ') ;
        set (gco, ' Position ' ,curpoi (1 ,1 :3))
    elseif arg = = 999
        set (gco, ' EraseMode ' , ' normal ') ;
        PF3. WindowButtonDownFcn = '' ;
        PF3. WindowButtonMotionFcn = '' ;
        PF3. WindowButtonUpFcn = '' ;
        PF3. Pointer = ' arrow ' ;
        PF3. Units = ' pixels ' ;
```

```
    PF3. BackingStore = 'on';
    set (gcf,PF3)
else
    PF4. WindowButtonDownFcn = '';
    PF4. WindowButtonMotionFcn = '';
    PF4. WindowButtonUpFcn = '';
    PF4. Pointer = 'arrow';
    PF4. Units = 'pixels';
    PF4. BackingStroe = 'on';
    set (gcf,PF4);
end.
```

在指令窗口中,运行 textzzy ('ABC'),字对象 ABC 就会出现在轴位框左下角,然后用鼠标把它拖到所需的位置.

例 5. 在 Matlab 中绘出二维和三维的卫星返回地球的运动轨迹示意图.

解 (1) 二维情况,程序代码如下:

```
shg;n = 10;t = n * pi * (0:0.0005:1);
x = sin (t);y = cos (t);
plot (x,y,'g');
axis square;hold on;
comet (x,y,0.01);
hold off.
```

(2) 三维情况,程序代码如下:

```
shg;
R0 = 1;
a = 12 * R0;b = 9 * R0;T0 = 2 * pi;
T = 5 * T0;dt = pi/100;t = [0:dt:T]';
f = sqrt (a 2 - b 2);
th = 12.5 * pi/180;
E = exp (- t/20);
x = E. * (a * cos (t) - f);
y = E. * (b * cos (th) * sin (t));
z = E. * (b * sin (th) * sin (t));
plot3 (x,y,z,'g');
[X,Y,Z] = sphere (30);
```

```
X = R0 * X;
Y = R0 * Y;
Z = R0 * Z;
grid on;
hold on;surf(X,Y,Z);
shading interp;
x1 = - 18 * R0;
x2 = 6 * R0;
y1 = - 12 * R0;
y2 = 12 * R0;
z1 = - 6 * R0;
z2 = 6 * R0;
axis([x1 x2 y1 y2 z1 z2]);
view([117 37]);
comet3(x,y,z,0.02);
hold off.
```

其程序运行结果中的某一时刻如图 B. 3 **所示**.

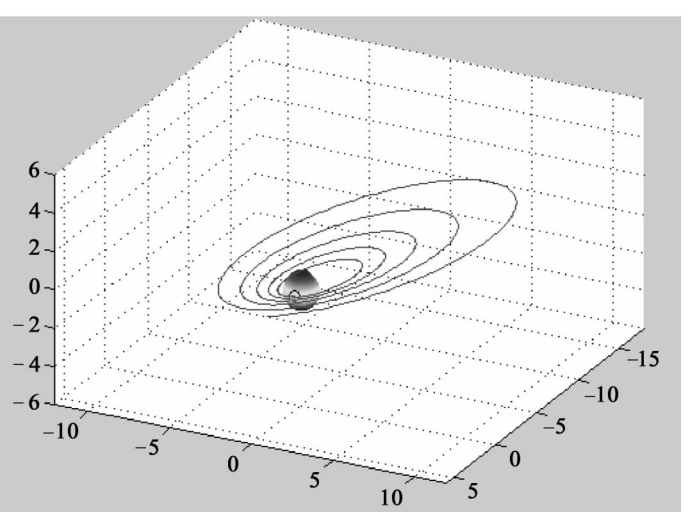


图 B. 3 **卫星返回地球轨线示意图**

B3. 其他高级绘图程序的例子

例 6. 在 Matlab 中利用图形用户界面和图形处理函数,写一个界面,在该界面上实现三维绘图的各种功能.

解 程序代码如下:

```
function graf2d2 (action) ;
if nargin < 1,
    action = ' initialize ' ;
end ;

if strcmp (action,' initialize ') ,
    figNumber = figure ( . . .
        ' Name ',' Examples of XYZ Plots in Matlab ', . . .
        ' NumberTitle ''off ', . . .
        ' Visible ',' off ') ;
    % Give the figure a nice black background
    colordef (figNumber,' black ')
    axes (. . .
        ' Units ',' normalized ', . . .
        ' Position ', [0. 10 0. 45 0. 60 0. 45]) ;

    top = 0. 35 ;
    left = 0. 05 ;
    right = 0. 75 ;
    bottom = 0. 05 ;
    labelHt = 0. 05 ;
    spacing = 0. 005 ;
    promptStr = str2mat (' ',' % Press the buttons at the right to see ', . . .
        ' % examples of some XYZ plots in Matlab ') ;
    frmBorder = 0. 02 ;
    frmPos = [left - frmBorder bottom - frmBorder. . .
        (right - left) + 2 * frmBorder (top - bottom) + 2 * frmBorder] ;
    uicontrol (. . .
```

```
        'Style','frame',...
        'Units','normalized',...
        'Position',frmPos,...
        'BackgroundColor',[0.50 0.50 0.50]);

labelPos = [left top-labelHt (right-left) labelHt];
uicontrol(...
        'Style','text',...
        'Units','normalized',...
        'Position',labelPos,...
        'BackgroundColor',[0.50 0.50 0.50],...
        'ForegroundColor',[1 1 1],...
        'String','MiniCommand Window');
mcwPos = [left bottom (right-left) top-bottom-labelHt-spacing];
mcwHndl = uicontrol(...
        'Style','edit',...
        'HorizontalAlignment','left',...
        'Units','normalized',...
        'Max',10,...
        'BackgroundColor',[1 1 1],...
        'Position',mcwPos,...
        'Callback','graf2d2('eval')',...
        'String',promptStr);

set(gcf,'UserData',mcwHndl);

labelColor = [0.8 0.8 0.8];
top = 0.95;
left = 0.80;
btnWid = 0.15;
btnHt = 0.08;
        spacing = 0.03;
% ==============================================
frmBorder = 0.02;
```

```
yPos = 0.05 - frmBorder;
frmPos = [left-frmBorder yPos btnWid + 2 * frmBorder 0.9 + 2 * frmBorder];
uicontrol(...
    'Style','frame',...
    'Units','normalized',...
    'Position',frmPos,...
    'BackgroundColor',[0.50 0.50 0.50]);
% ========================================
btnNumber = 1;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
labelStr = 'Mesh';
callbackStr = 'graf2d2('grafbutton')';
cmdStr = str2mat(...
    '% Mesh Plot of Peaks',...
    'z = peaks(25);',...
    'mesh(z);'...
);

btnPos = [left yPos-btnHt btnWid btnHt];
uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'Position',btnPos,...
    'String',labelStr,...
    'Callback',callbackStr,...
    'UserData',cmdStr);
% ========================================
btnNumber = 2;
yPos = top-(btnNumber-1) * (btnHt + spacing);
labelStr = 'Surf';
callbackStr = 'graf2d2('grafbutton')';
cmdStr = str2mat(...
    '% Surface Plot of Peaks',...
    'z = peaks(25);',...
```

```
        'surf(z);',...
        'colormap(jet);'...
);

btnPos = [left yPos-btnHt btnWid btnHt];
uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'Position',btnPos,...
        'String',labelStr,...
        'Callback',callbackStr,...
        'UserData',cmdStr);
% ========================================

btnNumber = 3;
yPos = top-(btnNumber-1)*(btnHt + spacing);
labelStr = 'Surfl';
callbackStr = 'graf2d2('grafbutton')';
cmdStr = str2mat(...
        '% Surface Plot (with Shading) of Peaks',...
        'z = peaks(25);',...
        'surfl(z);',...
        'shading interp;',...
        'colormap(pink);'...
);

btnPos = [left yPos-btnHt btnWid btnHt];
uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'Position',btnPos,...
        'String',labelStr,...
        'Callback',callbackStr,...
        'UserData',cmdStr);
```

```
% ************************************************
    btnNumber = 4;
    yPos = top-(btnNumber-1) * (btnHt + spacing);
    labelStr = 'Contour';
    callbackStr = 'graf2d2('grafbutton')';
    cmdStr = str2mat(...
        '% Contour Plot of Peaks',...
        'z = peaks(25);',...
        'contour(z,16);'...
    );
    btnPos = [left yPos-btnHt btnWid btnHt];
    uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'Position',btnPos,...
        'String',labelStr,...
        'Callback',callbackStr,...
        'UserData',cmdStr)
% ************************************************
    btnNumber = 5;
    yPos = top - (btnNumber-1) * (btnHt + spacing);
    labelStr = 'Quiver';
    callbackStr = 'graf2d2('grafbutton')';
    cmdStr = str2mat(...
        'x = -2:2:2;y = -1:2:1;',...
        '[xx,yy] = meshgrid(x,y);',...
        'zz = xx.*exp(-xx.^2 - yy.^2);',...
        '[px,py] = gradient(zz,.2,.2);',...
        'quiver(x,y,px,py,2);'...
    );
    btnPos = [left yPos-btnHt btnWid btnHt];
    uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
```

```
        'Position',btnPos,...
        'String',labelStr,...
        'Callback',callbackStr,...
        'UserData',cmdStr);
% ************************************************
    btnNumber = 6;
    yPos = top - (btnNumber - 1) * (btnHt + spacing);
    labelStr = 'Slice';
    callbackStr = 'graf2d2('grafbutton')';
    cmdStr = str2mat(...
        '[x,y,z] = meshgrid(-2:2:2,-2:2:2,-2:2:2);',...
        'v = x.*exp(-x.^2 - y.^2 - z.^2);',...
        'slice(v,[5 15 21],21,[1 10]),',...
        'axis([0 21 0 21 0 21]);',...
        'colormap(jet)'...
    );

    btnPos = [left yPos - btnHt btnWid btnHt];
    uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'Position',btnPos,...
        'String',labelStr,...
        'Callback',callbackStr,...
        'UserData',cmdStr);
% ************************************************
    uicontrol(...
        'Style','push',...
        'Units','normalized',...
        'Position',[left bottom + btnHt + spacing btnWid btnHt],...
        'String','Info',...
        'Callback','graf2d2('info')');

% ************************************************
```

```
    uicontrol (...
        'Style','push',...
        'Units','normalized',...
        'Position',[left bottom btnWid btnHt],...
        'String','Close',...
        'Callback','close(gcf)');
    set(figNumber,'Visible','on');
elseif strcmp(action,'grafbutton'),
    cmdStr = get(gco,'UserData');
    mcwHndl = get(gcf,'UserData');
    set(mcwHndl,'String',cmdStr);
    evalmcw(mcwHndl);

elseif strcmp(action,'eval'),
     cla reset;
    mcwHndl = get(gcf,'UserData');
    evalmcw(mcwHndl);

elseif strcmp(action,'info'),
    helpwin(mfilename);
end.
```

其程序运行的结果如图 B.4 所示.

例 7. 在 Matlab 中绘出卫星云图的示意图.

解　在 Matlab 中实现的代码如下：

```
function volvec(action);
if nargin < 1,
    action = 'initialize';
end;

if strcmp(action,'initialize'),
    figNumber = figure(...
        'Name','Examples of Volume and Vector Visualization in Matlab',...
        'NumberTitle','off',...
        'Visible','off');
```

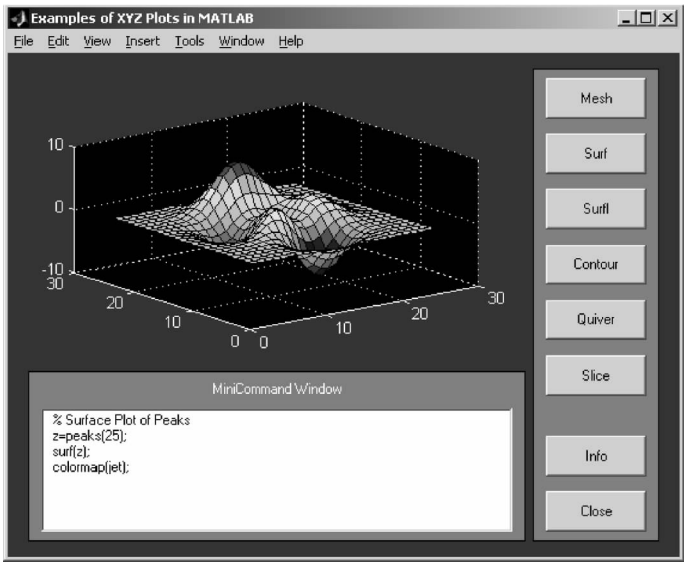


图 B. 4

```
    colordef (figNumber,'black')
axes (...
    'Units','normalized',...
    'Position',[0.10 0.45 0.60 0.45]);

top = 0.35;
left = 0.05;
right = 0.75;
bottom = 0.05;
labelHt = 0.05;
spacing = 0.005;
promptStr = str2mat('','% Press the buttons at the right to see',...
frmBorder = 0.02;
frmPos = [left - frmBorder bottom - frmBorder...
    (right - left) + 2 * frmBorder (top - bottom) + 2 * frmBorder];
uicontrol (...
    'Style','frame',...
```

```
        'Units','normalized',...
        'Position',frmPos,...
        'BackgroundColor',[0.50 0.50 0.50]);
labelPos = [left top-labelHt (right-left) labelHt];
uicontrol(...
        'Style','text',...
        'Units','normalized',...
        'Position',labelPos,...
        'BackgroundColor',[0.50 0.50 0.50],...
        'ForegroundColor',[1 1 1],...
        'String','MiniCommand Window');
         mcwPos = [left bottom (right-left) top-bottom-labelHt-spacing];
mcwHndl = uicontrol(...
        'Style','edit',...
        'HorizontalAlignment','left',...
        'Units','normalized',...
        'Max',10,...
        'BackgroundColor',[1 1 1],...
        'Position',mcwPos,...
        'Callback','volvec(''eval'')',...
        'String',promptStr);
        set(gcf,'UserData',mcwHndl);
labelColor = [0.8 0.8 0.8];
top = 0.95;
left = 0.80;
btnWid = 0.15;
btnHt = 0.08;
        spacing = 0.03;
frmBorder = 0.02;
yPos = 0.05 - frmBorder;
frmPos = [left-frmBorder yPos btnWid + 2 * frmBorder 0.9 + 2 * frmBorder];
uicontrol(...
        'Style','frame',...
        'Units','normalized',...
```

```
            'Position',frmPos,...
            'BackgroundColor',[0.50 0.50 0.50]);

        btnNumber = 1;
        yPos = top - (btnNumber - 1) * (btnHt + spacing);
        labelStr = 'Isosurface';
        callbackStr = 'volvec ('grafbutton')';
        cmdStr = str2mat(...
            '% Isosurface of MRI data',...
            'load mri;D = squeeze(D);',...
            '[x y z D] = subvolume(D, [nan nan nan nan nan 4]);',...
            'p = patch(isosurface(x,y,z,D, 5), 'FaceColor','red',
'EdgeColor', 'none');',...
            'p2 = patch(isocaps(x,y,z,D, 5), 'FaceColor', 'interp',
'EdgeColor', 'none');',...
            'isonormals(x,y,z,D,p);',...
            'view(3);axis tight;daspect([1 1 .4])',...
            'colormap(gray(100))',...
            'camva(6);box on',...
            'camlight(40, 40);camlight(-20, -10);lighting gouraud'...
        );

        btnPos = [left yPos - btnHt btnWid btnHt];
        uicontrol(...
            'Style','pushbutton',...
            'Units','normalized',...
            'Position',btnPos,...
            'String',labelStr,...
            'Callback',callbackStr,...
            'UserData',cmdStr);

    % ************************************************
        % The Coneplot button
        btnNumber = 2;
```

```
yPos = top - (btnNumber - 1) * (btnHt + spacing);
labelStr = 'Coneplot';
callbackStr = 'volvec (''grafbutton'') ';
cmdStr = str2mat (...
    '% Coneplot of wind data',...
    'load wind',...
    '[cx cy cz] = meshgrid (linspace (71,134,10),linspace (18,59,
10),3:4:15);',...
    'daspect ([1 1 1])',...
    'h = coneplot (x,y,z,u,v,w,cx,cy,cz,y,3);',...
    'set (h,'EdgeColor','none');', ...
    'colormap (hsv);',...
    'axis tight;box on;camproj perspective;',...
    'camva (24);campos ([175 10 85]);camtarget ([105 40 0])',...
    'camlight left;lighting gouraud');

btnPos = [left yPos - btnHt btnWid btnHt];
uicontrol (...
    'Style','pushbutton',...
    'Units','normalized',...
    'Position',btnPos,...
    'String',labelStr,...
    'Callback',callbackStr, ...
    'UserData',cmdStr);
btnNumber = 3;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
labelStr = 'Streamline';
callbackStr = 'volvec ('grafbutton')';
cmdStr = str2mat (....
    'load wind', ...
    '[sx sy sz] = meshgrid (80, 20:10:50, 0:5:15);', ...
    'h = streamline (x,y,z,u,v,w,sx,sy,sz);', ...
    'set (h,'Color','cyan');', ...
    'daspect ([1 1 1])', ...
```

```
    'axis tight;box on;camproj perspective;',...
    'camva(24);campos([175 10 85]);camtarget([105 40 0])',...
    'camlight left;lighting gouraud');

btnPos = [left yPos - btnHt btnWid btnHt];
uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'Position',btnPos,...
    'String',labelStr,...
    'Callback',callbackStr,...
    'UserData',cmdStr);
btnNumber = 4;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
labelStr = 'Streamtube';
callbackStr = 'volvec('grafbutton')';
cmdStr = str2mat(...
    '% Streamtubes of wind data',...
    '% Useful for visualizing divergence of a vector field',...
    'load wind',...
    '[sx sy sz] = meshgrid(80, [20 30 40], [5 10]);',...
    'daspect([1,1,1]);',...
    'h = streamtube(x,y,z,u,v,w,sx,sy,sz);',...
    'set(h,'facecolor','cyan','edgecolor','none');',...
    'axis tight;box on;camproj perspective;',...
    'axis([70 138 17 60 2.5 16]);',...
    'camva(24);campos([175 10 85]);camtarget([105 40 0])',...
    'camlight left;lighting gouraud');
    btnPos = [left yPos-btnHt btnWid btnHt];
uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'Position',btnPos,...
    'String',labelStr,...
```

```
        'Callback',callbackStr, ...
        'UserData',cmdStr);

    btnNumber = 5;
    yPos = top - (btnNumber - 1) * (btnHt + spacing);
    labelStr = 'Streamribbon';
    callbackStr = 'volvec('grafbutton')';
     cmdStr = str2mat(...
        '% Streamribbons of wind data',...
        '% Useful for visualizing curl of a vector field',...
        'load wind',...
        '[sx sy sz] = meshgrid(80, [20 30 40], [5 10]);', ...
        'daspect([1,1,1]);',...
        'h = streamribbon(x,y,z,u,v,w,sx,sy,sz);', ...
        'set(h,'facecolor','cyan','edgecolor','none')',...
        'axis tight;box on;camproj perspective;', ...
        'axis([70 138 17 60 2.5 16]);',...
        'camva(24);campos([175 10 85]);camtarget([105 40 0])', ...
        'camlight left;lighting gouraud');
    btnPos = [left yPos-btnHt btnWid btnHt];
    uicontrol(...
        'Style','pushbutton', ...
        'Units','normalized', ...
        'Position',btnPos, ...
        'String',labelStr, ...
        'Callback',callbackStr, ...
        'UserData',cmdStr);

    btnNumber = 6;
    yPos = top - (btnNumber - 1) * (btnHt + spacing);
    labelStr = 'Multiple';
    callbackStr = 'volvec('grafbutton')';
    cmdStr = str2mat(...
        '% Isosurface, isocaps, coneplot, and streamlines of wind data',...
```

```
            'load wind', ...
            'spd = sqrt (u. * u + v. * v + w. * w) ;', ...
            'p = patch (isosurface (x,y,z,spd, 40)) ;', ...
            'isonormals (x,y,z,spd, p) ', ...
            'set (p,'FaceColor','red', 'EdgeColor', 'none') ;', ...
            'p2 = patch (isocaps (x,y,z,spd, 40)) ;', ...
            'set (p2, 'FaceColor','interp','EdgeColor','none') ',...
            'daspect ([1 1 1]) ;', ...
            '[f verts] = reducepatch (isosurface (x,y,z,spd, 30) , .2) ; ', ...
            'h = coneplot (x,y,z,u,v,w,verts (:,1) ,verts (:,2) ,verts (:,3) ,
2) ;', ...
            'set (h,'FaceColor', 'cyan', 'EdgeColor', 'none') ;', ...
            '[sx sy sz] = meshgrid (80, 20 :10 :50, 0 :5 :15) ;', ...
            'h2 = streamline (x,y,z,u,v,w,sx,sy,sz) ;', ...
            'set (h2,'Color', [.4 1 .4]) ;', ...
            'colormap (jet) ', ...
            'axis tight ;box on', ...
            'camproj perspective ;camva (24) ; ',...
            'campos ([165 - 20 65]) ;camtarget ([100 40 - 5]) ', ...
            'camlight left ;lighting gouraud') ;

        btnPos = [left yPos-btnHt btnWid btnHt] ;
        uicontrol (...
            'Style','pushbutton', ...
            'Units','normalized', ...
            'Position',btnPos, ...
            'String',labelStr, ...
            'Callback',callbackStr, ...
            'UserData',cmdStr) ;
        uicontrol (...
            'Style','push', ...
            'Units','normalized', ...
            'Position', [left bottom + btnHt + spacing btnWid btnHt] , ...
            'String','Info', ...
```

```
            'Callback','volvec('info')');
        uicontrol(...
            'Style','push',...
            'Units','normalized',...
            'Position',[left bottom btnWid btnHt],...
            'String','Close',...
            'Callback','close(gcf)');

        set(figNumber,'Visible','on');

    elseif strcmp(action,'grafbutton'),
        cla reset;
        cmdStr = get(gco,'UserData');
        mcwHndl = get(gcf,'UserData');
        set(mcwHndl,'String',cmdStr);
        evalmcw(mcwHndl);

    elseif strcmp(action,'eval'),
        cla reset;
        mcwHndl = get(gcf,'UserData');
        evalmcw(mcwHndl);
    elseif strcmp(action,'info'),
        helpwin(mfilename);
    end.
```

其程序运行的结果如图 B. 5 **所示**.

例 8. **在** Matlab **中模拟湖水的波动**.

解 程序代码如下：

```
function vibes
lambda = [9.6397238445, 15.19725192, 2*pi 2, ...
    29.5214811, 31.9126360, 41.4745099, 44.948488, ...
    5*pi 2, 5*pi 2, 56.709610, 65.376535, 71.057755];
for k = 1:12
    L{k} = membrane(k);
end
```

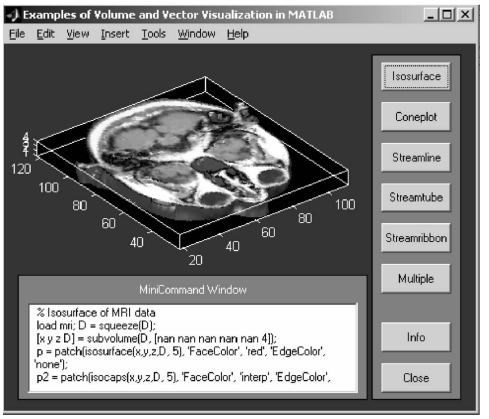


图 B. 5

```
for k = 1 :12
    c (k) = L {k} (25,23) /3 ;
end
 .
fig = figure ;
set (fig,' color ',' k ')
x = (- 15 :15) /15 ;
h = surf (x,x,L {1} ) ;
[a,e] = view ; view (a + 270,e) ;
axis ( [- 1 1 - 1 1 - 1 1]) ;
caxis (26. 9 * [- 1. 5 1]) ;
colormap (hot) ;
axis off

uicontrol (' pos ', [20 20 60 20] ,' string ',' done ',' fontsize ',12, . . .
    ' callback ',' close (gcbf) ') ;
uicontrol (' pos ', [20 40 60 20] ,' string ',' slower ',' fontsize ',12, . . .
    ' callback ',' set (gcbf,''userdata'',sqrt (0. 5) * get (gcbf, '' userdata '')) ') ;
uicontrol (' pos ', [20 60 60 20] ,' string ',' faster ',' fontsize ',12, . . .
    ' callback ',' set (gcbf,' userdata ',sqrt (2. 0) * get (gcbf,' userdata ')) ') ;
t = 0 ;
```

```
dt = 0.025;
set(fig,'userdata',dt)
while ishandle(fig)
    dt = get(fig,'userdata');
    t = t + dt;
    s = c.*sin(sqrt(lambda)*t);

    A = zeros(size(L{1}));
    for k = 1:12
    A = A + s(k)*L{k};
  end

  % Velocity
  s = lambda.*s;
  V = zeros(size(L{1}));
  for k = 1:12
    V = V + s(k)*L{k};
  end
  V(16:31,1:15) = NaN;
    set(h,'zdata',A,'cdata',V);
  drawnow
end.
```

其程序运行的结果如图 B. 6 **所示**.

例 9. **在** Matlab **中模拟实现地球的网格解析图、平面图和立体图**.

解 程序代码如下：

```
function lorenz(action)
play = 1;
stop = -1;
if nargin < 1,
    action = 'initialize';
end;
if strcmp(action,'initialize'),
    oldFigNumber = watchon;
    figNumber = figure(...
```

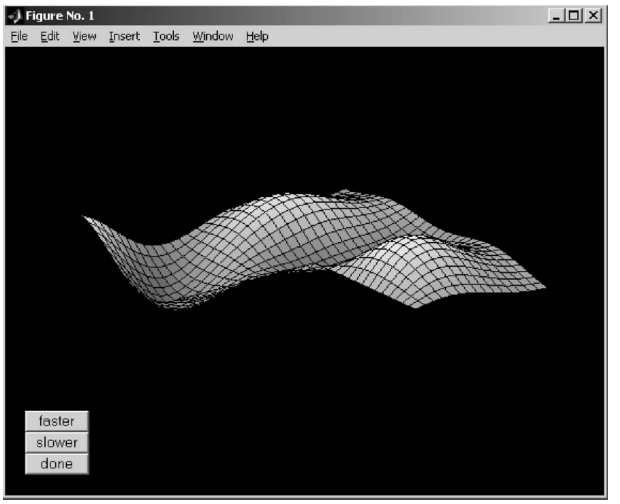


图 B. 6

```
        'Name','Lorenz Attractor',...
        'NumberTitle','off',...
        'Visible','off');
colordef(figNumber,'black')
axes(...
        'Units','normalized',...
        'Position',[0.05 0.10 0.75 0.95],...
        'Visible','off');

text(0,0,'Press the 'Start' button to see the Lorenz demo',...
        'HorizontalAlignment','center');
axis([-1 1 -1 1]);
labelColor = [0.8 0.8 0.8];
yInitPos = 0.90;
xPos = 0.85;
btnLen = 0.10;
btnWid = 0.10;
        spacing = 0.05;
```

```
frmBorder = 0. 02 ;
yPos = 0. 05 - frmBorder ;
frmPos = [xPos - frmBorder yPos btnLen + 2 * frmBorder 0. 9 + 2 * frmBor-
der] ;
h = uicontrol ( . . .
    ' Style ',' frame ', . . .
    ' Units ',' normalized ', . . .
    ' Position ',frmPos, . . .
    ' BackgroundColor ', [0. 50 0. 50 0. 50]) ;
btnNumber = 1 ;
yPos = 0. 90 - (btnNumber - 1) * (btnWid + spacing) ;
labelStr = ' Start ' ;
cmdStr = ' Start ' ;
callbackStr = ' lorenz (' start ') ;' ;
btnPos = [xPos yPos - spacing btnLen btnWid] ;
startHndl = uicontrol ( . . .
    ' Style ',' pushbutton ', . . .
    ' Units ',' normalized ', . . .
    ' Position ',btnPos, . . .
    ' String ',labelStr, . . .
    ' Interruptible ',' on ', . . .
    ' Callback ',callbackStr) ;
btnNumber = 2 ;
yPos = 0. 90 - (btnNumber - 1) * (btnWid + spacing) ;
labelStr = ' Stop ' ;
    callbackStr = ' set (gca,' Userdata ', - 1) ' ;

btnPos = [xPos yPos - spacing btnLen btnWid] ;
stopHndl = uicontrol ( . . .
    ' Style ',' pushbutton ', . . .
    ' Units ',' normalized ', . . .
    ' Position ',btnPos, . . .
    ' Enable ',' off ', . . .
    ' String ',labelStr, . . .
```

```
        'Callback',callbackStr);
    labelStr = 'Info';
    callbackStr = 'lorenz('info')';
    infoHndl = uicontrol(...
        'Style','push',...
        'Units','normalized',...
        'position',[xPos 0.20 btnLen 0.10],...
        'string',labelStr,...
        'call',callbackStr);
    labelStr = 'Close';
    callbackStr = 'close(gcf)';
    closeHndl = uicontrol(...
        'Style','push',...
        'Units','normalized',...
        'position',[xPos 0.05 btnLen 0.10],...
        'string',labelStr,...
        'call',callbackStr);

    hndlList = [startHndl stopHndl infoHndl closeHndl];
    set(figNumber,'Visible','on',...
        'UserData',hndlList);
    watchoff(oldFigNumber);
    figure(figNumber);
elseif strcmp(action,'start'),
    axHndl = gca;
    figNumber = gcf;
    hndlList = get(figNumber,'UserData');
    startHndl = hndlList(1);
    stopHndl = hndlList(2);
    infoHndl = hndlList(3);
    closeHndl = hndlList(4);
    set([startHndl closeHndl infoHndl],'Enable','off');
    set(stopHndl,'Enable','on');
    set(figNumber,'Backingstore','off');
```

```
set (axHndl, ...
    'XLim', [0 40], 'YLim', [-35 10], 'ZLim', [-10 40], ...
    'Userdata',play, ...
    'XTick', [], 'YTick', [], 'ZTick', [], ...
    'Drawmode','fast', ...
    'Visible','on', ...
    'NextPlot','add', ...
    'Userdata',play, ...
    'View', [-37.5,30]);
xlabel ('X');
ylabel ('Y');
zlabel ('Z');
global SIGMA RHO BETA
SIGMA = 10.;
RHO = 28.;
BETA = 8./3.;

FunFcn = 'lorenzeq';
y0 (1) = rand * 30 + 5;
y0 (2) = rand * 35 - 30;
y0 (3) = rand * 40 - 5;
t0 = 0;
tfinal = 100;
pow = 1/3;
tol = 0.001;

t = t0;
hmax = (tfinal - t) /5;
hmin = (tfinal - t) /200000;
h = (tfinal - t) /100;
y = y0 (:);
tau = tol * max (norm (y,'inf'),1);
L = 50;
Y = y * ones (1,L);
```

```
cla;
head = line(...
    'color','r',...
    'Marker','.',...
    'markersize',25,...
    'erase','xor',...
    'xdata',y(1),'ydata',y(2),'zdata',y(3));
body = line(...
    'color','y',...
    'LineStyle','-',...
    'erase','none',...
    'xdata',[],'ydata',[],'zdata',[]);
tail = line(...
    'color','b',...
    'LineStyle','-',...
    'erase','none',...
    'xdata',[],'ydata',[],'zdata',[]);

while (get(axHndl,'Userdata')==play) & (h >= hmin)
    if t+h > tfinal, h = tfinal - t;end
    s1 = feval(FunFcn, t, y);
    s2 = feval(FunFcn, t+h, y+h*s1);
    s3 = feval(FunFcn, t+h/2, y+h*(s1+s2)/4);
    delta = norm(h*(s1-2*s3+s2)/3,'inf');
    tau = tol*max(norm(y,'inf'),1.0);

    ts = t;
    ys = y;
    if delta <= tau
        t = t+h;
        y = y+h*(s1+4*s3+s2)/6;
        Y = [y Y(:,1:L-1)];
        set(head,'xdata',Y(1,1),'ydata',Y(2,1),'zdata',Y(3,1))
        set(body,'xdata',Y(1,1:2),'ydata',Y(2,1:2),'zdata',Y(3,1:2))
```

```
                set (tail,'xdata',Y(1,L-1:L),'ydata',Y(2,L-1:L),
'zdata',Y(3,L-1:L))
                drawnow;
            end
            if delta ~=0.0
                h = min(hmax, 0.9*h*(tau/delta) pow);
            end
        end;
          set([startHndl closeHndl infoHndl],'Enable','on');
            set(stopHndl,'Enable','off');
        elseif strcmp(action,'info');
            helpwin(mfilename);
        end;
        function ydot = lorenzeq(t,y)
        global sigma rho beta
        A = [-beta      0       y(2)
              0     -sigma     sigma
        -y(2)       rho       -1];
        ydot = A*y.
```

其程序运行结果如图 B.7 所示.

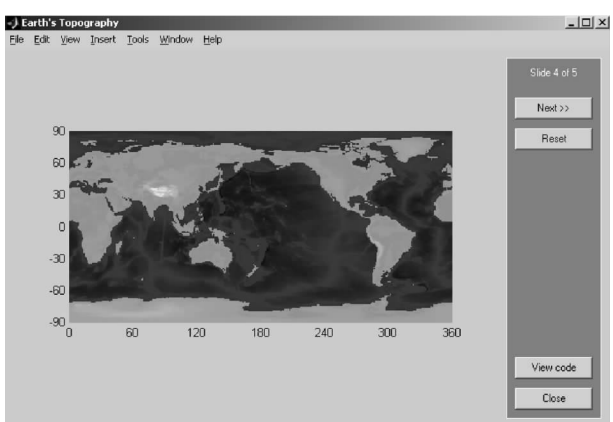


图 B.7 地球的平面模拟图

例 10. **模拟混沌星云的运行**.

解　程序代码如下：

```
    load ('topo.mat','topo','topomap1');
    whos topo topomap1

    contour(0:359,-89:90,topo,[0 0],'b')
axis equal
box on
set(gca,'XLim',[0 360],'YLim',[-90 90],…
    'XTick',[0 60 120 180 240 300 360],…
    'YTick',[-90 -60 -30 0 30 60 90]);
hold on
image([0 360],[-90 90],topo,'CDataMapping','scaled');
colormap(topomap1);

[x,y,z] = sphere(50);

cla reset
axis square off
props.AmbientStrength = 0.1;
props.DiffuseStrength = 1;
props.SpecularColorReflectance = .5;
props.SpecularExponent = 20;
props.SpecularStrength = 1;
props.FaceColor = 'texture';
props.EdgeColor = 'none';
props.FaceLighting = 'phong';
props.Cdata = topo;
surface(x,y,z,props);
light('position',[-1 0 1]);
light('position',[-1.5 0.5 -0.5],'color',[.6 .2 .2]);
view(3)
```

其程序运行结果如图 B.8 **所示**.

例 11. **茶壶的模拟图**.

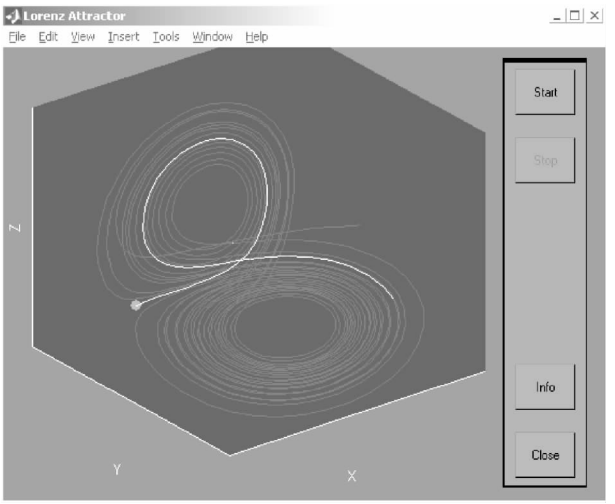


图 B. 8

解　程序代码如下：

```
function varargout = teapotdemo (varargin)
if nargin == 0

    fig = figure (openfig (mfilename,' reuse ')) ;
    handles = guihandles (fig) ;
    guidata (fig, handles) ;
    if nargout > 0
        varargout {1} = fig ;
    end

    daspect ( [1 1 1]) ;
    xlim ( [- 4 4]) ;
    ylim ( [- 3 3]) ;
    axis vis3d off ;
    view (3) ;
    light (' Position ', [0. 25 - 0. 433 - 0. 866] ,' Style ',' infinite ') ;
    light (' Position ', [- 0. 433 0. 25 0. 866] ,' Style ',' infinite ') ;
```

```
    colormap autumn;

    s = struct('resolution',12,'colorby','z','lidoffset',0,'bottom',1);
    p = teapot(12,'z',0,1);
    set(p,'UserData',s);

    set(p,'EdgeColor',[0 0 0],'LineStyle','none','FaceColor','interp');

    lighting gouraud;
    set(gcf,'UserData',p);

    elseif ischar(varargin{1})

        try
            if (nargout)
                [varargout{1:nargout}] = feval(varargin{:});
switchyard
            else
                feval(varargin{:});
          end
        catch
            disp(lasterr);
        end
      end

function runcmd(cmd)
    eval(cmd);
    cw = findobj('Tag','minicmdwin');
    set(cw,'String',cmd);

function rebuildteapot()
    cr = sprintf('\n');
    p = get(gcf,'UserData');
    s = get(p,'UserData');
```

```
    cmdStr = [' p = get (gcf,' UserData ') ;',cr, . . .
                    ' teapot (', . . .
                    num2str (s. resolution) , . . .
                    ' , ', . . .
                    s. colorby,. . .
                    ' , ', . . .
                    num2str (s. lidoffset) , . . .
                    ' , ', . . .
                    num2str (s. bottom) , . . .
                    ' ,p) ;'] ;
runcmd (cmdStr) ;
% ----------------------------------------------------
function varargout = lightingmenu_Callback (h, eventdata, handles, varargin)
    n = get (h,' value ') ;
    cmddata = {' flat ' ' gouraud ' ' phong '} ;
    runcmd ( [' lighting ' cmddata {n} ' ;']) ;

% ----------------------------------------------------
function varargout = colormapmenu_Callback (h, eventdata, handles, varargin)
    n = get (h,' value ') ;
    cmddata = {' autumn ' ' copper ' ' hsv ' ' winter '} ;
    runcmd ( [' colormap ' cmddata {n} ' ;']) ;
% ----------------------------------------------------
function varargout = materialmenu_Callback (h, eventdata, handles, varargin)
    n = get (h,' value ') ;
    cmddata = {' default ' ' shiny ' ' dull ' ' metal '} ;
    runcmd ( [' material ' cmddata {n} ' ;']) ;

% ----------------------------------------------------
function varargout = renderstylemenu_Callback (h, eventdata, handles, varargin)
    n = get (h,' value ') ;
    fmtData = {
        { ' . ', ' none ', ' none ' } ;
        { ' none ', ' - ', ' none ' } ;
        { ' none ', ' none ', ' interp ' }
```

```
    } ;

        runcmd2 = sprintf ('p = get (gcf,'UserData') ;hnset (p,'Marker',
'%s') hnset (p,'LineStyle','%s') hnset (p,'FaceColor','%s') hn',...
            fmtData {n} {}) ;
    runcmd (runcmd2) ;

% ----------------------------------------------
function varargout = colorbymenu_Callback (h, eventdata, handles, varargin)
    n = get (h,'value') ;
    p = get (gcf,'UserData') ;
    s = get (p,'UserData') ;
    cmddata = {'none' 'x' 'y' 'z' 'u' 'v' 'index'} ;
    s. colorby = cmddata {n} ;
    set (p,'UserData',s) ;
    if n == 1
        set (p,'FaceColor','none') ;
    else
        set (p,'FaceColor','interp') ;
    end
    rebuildteapot ;
% ----------------------------------------------
function varargout = transparentbutton_Callback (h, eventdata, handles, varargin)
    if get (h,'value') == 1
        runcmd ('alpha (0. 5) ;') ;
    else
        runcmd ('alpha (1) ;') ;
    end
% ----------------------------------------------
function varargout = infobutton_Callback (h, eventdata, handles, varargin)
    helpwin (mfilename) ;

% ----------------------------------------------
function varargout = closebutton_Callback (h, eventdata, handles, varargin)
    close (gcf)
```

```
% ----------------------------------------------
function varargout = edgebutton_Callback (h, eventdata, handles, varargin)

    if get (h,'value') == 0
        runcmd ('p = get (gcf,'UserData') ;set (p,'LineStyle',
'none') ;') ;
    else
        runcmd ('p = get (gcf,'UserData') ;set (p,'LineStyle',
'-') ;') ;
    end

% ----------------------------------------------------
function varargout = bottombutton_Callback (h, eventdata, handles, varargin)
    p = get (gcf,'UserData') ;
    s = get (p,'UserData') ;
    if get (h,'value') == 1
        s. bottom = 1 ;
    else
        s. bottom = 0 ;
    end
    set (p,'UserData',s) ;
    rebuildteapot ;

% ----------------------------------------------------
function varargout = resolutionslider_Callback (h, eventdata, handles, varargin)
    res = get (h,'value') ;
    p = get (gcf,'UserData') ;
    s = get (p,'UserData') ;
    s. resolution = round (res) ;
    set (p,'UserData',s) ;
    rebuildteapot ;

% ----------------------------------------------------
```

```
function varargout = lidoffsetslider_Callback (h, eventdata, handles, varargin)
    res = get (h,' value ') ;
    p = get (gcf,' UserData ') ;
    s = get (p,' UserData ') ;
    s. lidoffset = round (res) ;
    set (p,' UserData ',s) ;
rebuildteapot ;

% ----------------------------------------------------
function pout = teapot (n,colorBy,lidoffset,bottom,pin)

    if nargin < 1
        n = 12 ;
    end
    if nargin < 2
        colorBy = ' none ' ;
    end
    if nargin < 3
        lidoffset = 0 ;
    end
    if nargin < 4
        bottom = 1 ;
    end

    verts = teapotVertices ;
    quads = teapotControlPoints ;

    if lidoffset > 0
        verts (204 :269,3) = verts (204 :269,3) + lidoffset ;
    end

    if bottom = = 0
        quads = quads ( :, :,1 :28) ;
    end
```

```
pv = [] ;
pf = [] ;
pc = [] ;

for i = 1 :size (quads,3)
    % extract the control points for this bezier patch
    points = verts (quads ( :, :,i) , :) ;
    % rip the vertices into X, Y, and Z components
    x = points ( :,1) ;
    y = points ( :,2) ;
    z = points ( :,3) ;

    [f v c] = evalCubicBezierPatch (n,x,y,z,i,colorBy) ;

    numv = size (pv,1) ;
    pv = [pv ;v] ;
    pf = [pf ;f + numv] ;
    pc = [pc ;c] ;
end

if nargin < 5
    pout = patch ('faces', pf, 'vertices', pv) ;
    p = pout ;
else
    set (pin,'faces',pf,'vertices',pv) ;
    p = pin ;
end

if strcmp (colorBy,'none')
    set (p, 'facecolor', [.5 .5 .5]) ;
else
    set (p,'FaceVertexCData',pc) ;
end
```

```
function verts = teapotVertices()
verts = [1.4        0.        2.4;      1.4        -0.784     2.4;
         0.784     -1.4       2.4;      0.         -1.4       2.4;
         1.3375     0.        2.53125;  1.3375     -0.749     2.53125;
         0.749     -1.3375    2.53125;  0.         -1.3375    2.53125;
         1.4375     0.        2.53125;  1.4375     -0.805     2.53125;
         0.805     -1.4375    2.53125;  0.         -1.4375    2.53125;
         1.5        0.        2.4;      1.5        -0.84      2.4;
         0.84      -1.5       2.4;      0.         -1.5       2.4;
        -0.784     -1.4       2.4;     -1.4        -0.784     2.4;
        -1.4        0.        2.4;     -0.749      -1.3375    2.53125;
        -1.3375    -0.749     2.53125; -1.3375      0.0       2.53125;
        -0.805     -1.4375    2.53125; -1.4375     -0.805     2.53125;
        -1.4375     0.0       2.53125; -0.84       -1.5       2.4;
        -1.5       -0.84      2.4;     -1.5         0.        2.4;
        -1.4        0.784     2.4;     -0.784       1.4       2.4;
         0.         1.4       2.4;     -1.3375      0.749     2.53125;
        -0.749      1.3375    2.53125;  0.         1.3375     2.53125;
        -1.4375     0.805     2.53125; -0.805      1.4375     2.53125;
         0.         1.4375    2.53125; -1.5        0.84       2.4;
        -0.84       1.5       2.4;      0.         1.5        2.4;
         0.784      1.4       2.4;      1.4        0.784      2.4;
         0.749      1.3375    2.53125;  1.3375     0.749      2.53125;
         0.805      1.4375    2.53125;  1.4375     0.805      2.53125;
         0.84       1.5       2.4;      1.5        0.84       2.4;
         1.75       0.        1.875;    1.75      -0.98       1.875;
         0.98      -1.75      1.875;    0.         -1.75      1.875;
         2.         0.        1.35;     2.         -1.12      1.35;
         1.12      -2.        1.35;     0.         -2.        1.35;
         2.         0.        0.9;      2.         -1.12      0.9;
         1.12      -2.        0.9;      0.         -2.        0.9;
        -0.98      -1.75      1.875;   -1.75       -0.98      1.875;
        -1.75       0.        1.875;   -1.12       -2.        1.35;
```

-2.	-1.12	1.35；	-2.	0.	1.35；
-1.12	-2.	0.9；	-2.	-1.12	0.9；
-2.	0.	0.9；	-1.75	0.98	1.875；
-0.98	1.75	1.875；	0.	1.75	1.875；
-2.	1.12	1.35；	-1.12	2.	1.35；
0.	2.	1.35；	-2.	1.12	0.9；
-1.12	2.	0.9；	0.0	2.	0.9 ；
0.98	1.75	1.875；	1.75	0.98	1.875；
1.12	2.	1.35；	2.	1.12	1.35；
1.12	2.	0.9；	2.	1.12	0.9；
2.	0.	0.45；	2.	-1.12	0.45；
1.12	-2.	0.45；	0.	-2.	0.45；
1.5	0.	0.225；	1.5	-0.84	0.225；
0.84	-1.5	0.225；	0.	-1.5	0.225；
1.5	0.	0.15；	1.5	-0.84	0.15；
0.84	-1.5	0.15；	0.0	-1.5	0.15；
-1.12	-2.	0.45；	-2.	-1.12	0.45；
-2.	0.	0.45；	-0.84	-1.5	0.225；
-1.5	-0.84	0.225；	-1.5	0.	0.225；
-0.84	-1.5	0.15；	-1.5	-0.84	0.15；
-1.5	0.	0.15 ；	-2.	1.12	0.45；
-1.12	2.	0.45；	0.	2.	0.45；
-1.5	0.84	0.225；	-0.84	1.5	0.225；
0.	1.5	0.225；	-1.5	0.84	0.15；
-0.84	1.5	0.15；	0.	1.5	0.15；
1.12	2.	0.45；	2.	1.12	0.45；
0.84	1.5	0.225；	1.5	0.84	0.225；
0.84	1.5	0.15；	1.5	0.84	0.15；
-1.6	0.	2.025；	-1.6	-0.3	2.025；
-1.5	-0.3	2.25；	-1.5	0	2.25；
-2.3	0.	2.025；	-2.3	-0.3	2.025；
-2.5	-0.3	2.25；	-2.5	0.	2.25；
-2.7	0.	2.025；	-2.7	-0.3	2.025；
-3.	-0.3	2.25；	-3.	0.	2.25；

-2.7	0.	1.8；	-2.7	-0.3	1.8；
-3.	-0.3	1.8；	-3.	0.	1.8；
-1.5	0.3	2.25；	-1.6	0.3	2.025；
-2.5	0.3	2.25；	-2.3	0.3	2.025；
-3.	0.3	2.25；	-2.7	0.3	2.025；
-3.	0.3	1.8；	-2.7	0.3	1.8；
-2.7	0.	1.575；	-2.7	-0.3	1.575；
-3.	-0.3	1.35；	-3.	0.	1.35；
-2.5	0.	1.125；	-2.5	-0.3	1.125；
-2.65	-0.3	0.9375；	-2.65	0.	0.9375；
-2.	-0.3	0.9；	-1.9	-0.3	0.6；
-1.9	0.	0.6；	-3.	0.3	1.35；
-2.7	0.3	1.575；	-2.65	0.3	0.9375；
-2.5	0.3	1.125；	-1.9	0.3	0.6；
-2.	0.3	0.9；	1.7	0.	1.425；
1.7	-0.66	1.425；	1.7	-0.66	0.6；
1.7	0.	0.6；	2.6	0.	1.425；
2.6	-0.66	1.425；	3.1	-0.66	0.825；
3.1	0.	0.825；	2.3	0.	2.1；
2.3	-0.25	2.1；	2.4	-0.25	2.025；
2.4	0.	2.025；	2.7	0.	2.4；
2.7	-0.25	2.4；	3.3	-0.25	2.4；
3.3	0.	2.4；	1.7	0.66	0.6；
1.7	0.66	1.425；	3.1	0.66	0.825；
2.6	0.66	1.425；	2.4	0.25	2.025；
2.3	0.25	2.1；	3.3	0.25	2.4；
2.7	0.25	2.4；	2.8	0.	2.475；
2.8	-0.25	2.475；	3.525	-0.25	2.49375；
3.525	0.	2.49375；	2.9	0.	2.475；
2.9	-0.15	2.475；	3.45	-0.15	2.5125；
3.45	0.	2.5125；	2.8	0.	2.4；
2.8	-0.15	2.4；	3.2	-0.15	2.4；
3.2	0.	2.4；	3.525	0.25	2.49375；
2.8	0.25	2.475；	3.45	0.15	2.5125；

```
2.9      0.15     2.475 ;  3.2      0.15     2.4 ;
2.8      0.15     2.4 ;    0.       0.       3.15 ;
0.       -0.002   3.15 ;   0.002    0.       3.15 ;
0.8      0.       3.15 ;   0.8      -0.45    3.15 ;
0.45     -0.8     3.15 ;   0.       -0.8     3.15 ;
0.       0.       2.85 ;   0.2      0.       2.7 ;
0.2      -0.112   2.7 ;    0.112    -0.2     2.7 ;
0.       -0.2     2.7 ;    -0.002   0.       3.15 ;
-0.45    -0.8     3.15 ;   -0.8     -0.45    3.15 ;
-0.8     0.       3.15 ;   -0.112   -0.2     2.7 ;
-0.2     -0.112   2.7 ;    -0.2     0.       2.7 ;
0        0.002    3.15 ;   -0.8     0.45     3.15 ;
-0.45    0.8      3.15 ;   0.       0.8      3.15 ;
-0.2     0.112    2.7 ;    -0.112   0.2      2.7 ;
0.       0.2      2.7 ;    0.45     0.8      3.15 ;
0.8      0.45     3.15 ;   0.112    0.2      2.7 ;
0.2      0.112    2.7 ;    0.4      0.       2.55 ;
0.4      -0.224   2.55 ;   0.224    -0.4     2.55 ;
0.       -0.4     2.55 ;   1.3      0.       2.55 ;
1.3      -0.728   2.55 ;   0.728    -1.3     2.55 ;
0.       -1.3     2.55 ;   1.3      0.       2.4 ;
1.3      -0.728   2.4 ;    0.728    -1.3     2.4 ;
0.       -1.3     2.4 ;    -0.224   -0.4     2.55 ;
-0.4     -0.224   2.55 ;   -0.4     0.       2.55 ;
-0.728   -1.3     2.55 ;   -1.3     -0.728   2.55 ;
-1.3     0.       2.55 ;   -0.728   -1.3     2.4 ;
-1.3     -0.728   2.4 ;    -1.3     0.       2.4 ;
-0.4     0.224    2.55 ;   -0.224   0.4      2.55 ;
0.       0.4      2.55 ;   -1.3     0.728    2.55 ;
-0.728   1.3      2.55 ;   0.       1.3      2.55 ;
-1.3     0.728    2.4 ;    -0.728   1.3      2.4 ;
0.       1.3      2.4 ;    0.224    0.4      2.55 ;
0.4      0.224    2.55 ;   0.728    1.3      2.55 ;
1.3      0.728    2.55 ;   0.728    1.3      2.4 ;
```

```
  1.3        0.728      2.4 ;      0.         0.         0. ;
  1.5        0.         0.15 ;     1.5        0.84       0.15 ;
  0.84       1.5        0.15 ;     0.         1.5        0.15 ;
  1.5        0.         0.075 ;    1.5        0.84       0.075 ;
  0.83       1.5        0.075 ;    0.         1.5        0.075 ;
  1.425      0.         0. ;       1.425      0.798      0. ;
  0.798      1.425      0. ;       0.         1.425      0. ;
 -0.84       1.5        0.15 ;    -1.5        0.84       0.15 ;
 -1.5        0.         0.15 ;    -0.84       1.5        0.075 ;
 -1.5        0.84       0.075 ;   -1.5        0.         0.075 ;
 -0.798      1.425      0. ;      -1.425      0.798      0. ;
 -1.425      0.         0. ;      -1.5       -0.84       0.15 ;
 -0.84      -1.5        0.15 ;     0.        -1.5        0.15 ;
 -1.5       -0.84       0.075 ;   -0.84       -1.5       0.075 ;
  0.        -1.5        0.075 ;   -1.425      -0.798      0. ;
 -0.798     -1.425      0. ;       0.        -1.425      0. ;
  0.84      -1.5        0.15 ;     1.5       -0.84       0.15 ;
  0.84      -1.5        0.075 ;    1.5       -0.84       0.075 ;
  0.798     -1.425      0. ;       1.425     -0.798      0. ] ;
% These select which control points are used for each of the 32 bezier patches .
function quads = teapotControlPoints ()
quads = cat (3, [1 2 3 4 ;5 6 7 8 ;9 10 11 12 ;13 14 15 16 ] ,
 [4 17 18 19 ;8 20 21 22 ;12 23 24 25 ;16 26 27 28 ],
 [19 29 30 31 ;22 32 33 34 ;25 35 36 37 ;28 38 39 40 ] ,
 [31 41 42 1 ;34 43 44 5 ;37 45 46 9 ;40 47 48 13 ] ,
 [13 14 15 16 ;49 50 51 52 ;53 54 55 56 ;57 58 59 60 ] ,
 [16 26 27 28 ;52 61 62 63 ;56 64 65 66 ;60 67 68 69 ] ,
 [28 38 39 40 ;63 70 71 72 ;66 73 74 75 ;69 76 77 78 ],
 [40 47 48 13 ;72 79 80 49 ;75 81 82 53 ;78 83 84 57 ] ,
 [57 58 59 60 ;85 86 87 88 ;89 90 91 92 ;93 94 95 96 ] ,
 [60 67 68 69 ;88 97 98 99 ;92 100 101 102 ;96 103 104 105 ] ,
 [69 76 77 78 ;99 106 107 108 ;102 109 110 111 ;105 112 113 114 ] ,
 [78 83 84 57 ;108 115 116 85 ;111 117 118 89 ;114 119 120 93 ] ,
 [121 122 123 124 ;125 126 127 128 ;129 130 131 132 ;133 134 135 136 ] ,
```

```
[124 137 138 121 ;128 139 140 125 ;132 141 142 129 ;136 143 144 133 ] ,
[133 134 135 136 ;145 146 147 148 ;149 150 151 152 ;69 153 154 155 ],
[136 143 144 133 ;148 156 157 145 ;152 158 159 149 ;155 160 161 69 ] ,
[162 163 164 165 ;166 167 168 169 ;170 171 172 173 ;174 175 176 177 ] ,
[165 178 179 162 ;169 180 181 166 ;173 182 183 170 ;177 184 185 174 ] ,
[174 175 176 177 ;186 187 188 189 ;190 191 192 193 ;194 195 196 197 ] ,
[177 184 185 174 ;189 198 199 186 ;193 200 201 190 ;197 202 203 194 ] ,
[204 204 204 204 ;207 208 209 210 ;211 211 211 211 ;212 213 214 215 ] ,
[204 204 204 204 ;210 217 218 219 ;211 211 211 211 ;215 220 221 222 ] ,
[204 204 204 204 ;219 224 225 226 ;211 211 211 211 ;222 227 228 229 ] ,
[204 204 204 204 ;226 230 231 207 ;211 211 211 211 ;229 232 233 212 ] ,
[212 213 214 215 ;234 235 236 237 ;238 239 240 241 ;242 243 244 245 ] ,
[215 220 221 222 ;237 246 247 248 ;241 249 250 251 ;245 252 253 254 ] ,
[222 227 228 229 ;248 255 256 257 ;251 258 259 260 ;254 261 262 263 ] ,
[229 232 233 212 ;257 264 265 234 ;260 266 267 238 ;263 268 269 242 ] ,
[270 270 270 270 ;279 280 281 282 ;275 276 277 278 ;271 272 273 274 ] ,
[270 270 270 270 ;282 289 290 291 ;278 286 287 288 ;274 283 284 285 ] ,
[270 270 270 270 ;291 298 299 300 ;288 295 296 297 ;285 292 293 294 ],
[270 270 270 270 ;300 305 306 279 ;297 303 304 275 ;294 301 302 271 ] ) ;

function varargout = evalCubicBezierPatch (n,xc,yc,zc,index,colorBy)

    u = (0 :n - 1) / (n - 1) ;
        A = [(u. 3) (3 * u. 2 . * (1 - u)) (3 * u . * (1 - u). 2) ((1 -
u). 3)] ;

    v = (0 :n - 1) / (n - 1) ;
        B = [(v. 3) (3 * v. 2 . * (1 - v)) (3 * v . * (1 - v). 2) ((1 -
v). 3)] ;
        mat = kron (A,B) ;
    xd = mat * xc ;
    yd = mat * yc ;
    zd = mat * zc ;
```

```
x = reshape (xd,n,n) ;
y = reshape (yd,n,n) ;
z = reshape (zd,n,n) ;

if strcmp (colorBy,' x ')
    colors = x ;
elseif strcmp (colorBy,' y ')
    colors = y ;
elseif strcmp (colorBy,' z ')
    colors = z ;
elseif strcmp (colorBy,' u ')
    colors = repmat ( [0 :1 / (n - 1) :1] ,n,1) ;
elseif strcmp (colorBy,' v ')
    colors = repmat ( [0 :1 / (n - 1) :1] ',1,n) ;
elseif strcmp (colorBy,' index ')
    colors = repmat (index,n,n) ;
elseif strcmp (colorBy,' none ')
    colors = [] ;
end

% if no output args, use surface to draw the geometry
if nargout == 0
    % create surface
    s = surface (reshape (xd,n,n) ,reshape (yd,n,n) ,reshape (zd,n,n) ,
colors) ;

    % make it look nice
    set (s,' EdgeColor ',' none ',' FaceColor ',' interp ')

% otherwise, return the geometry in the correct form for patch
elseif nargout == 3
    [f v c] = surf2patch (reshape (xd,n,n) ,reshape (yd,n,n) ,reshape
(zd,n,n) ,colors) ;
    varargout {1} = f ;
```

```
        varargout{2} = v;
        varargout{3} = c;
end.
```

其程序运行结果如图 B. 9 所示 .

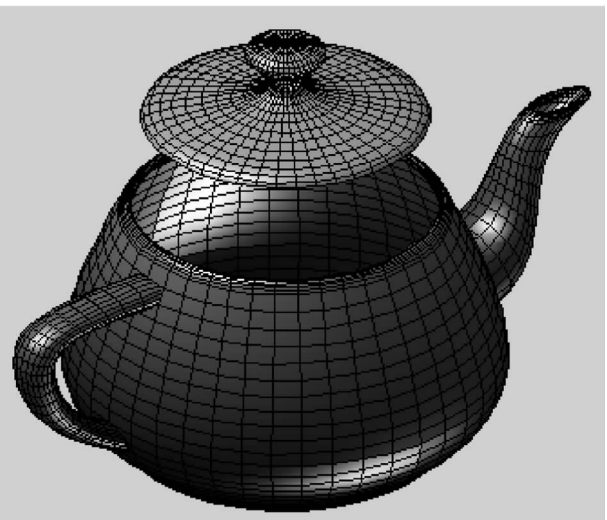


图 B. 9

第四部分 附 录

全国大学生数学建模竞赛试题选编

1994 年全国大学生数学建模竞赛题

A 题 逢山开路

要在一山区修建公路，首先测得一些地点的高程，数据如表 1 所示（平面区域 $0 \leqslant x \leqslant 5600$，$0 \leqslant y \leqslant 4800$，表 1 中数据为坐标点的高程，单位：m）. 数据显示：在 $y = 3200$ 处有一东西走向的山峰；从坐标（2400，2400）到坐标（4800，0）有一西北——东南走向的山谷；在坐标（2000，2800）附近有一山口湖，其最高水位略高于 1 350m，雨季在山谷中形成一溪流. 经调查知，雨量最大时溪流水面宽度 w 与（溪流最深处）x 坐标的关系可以近似表示为

$$w(x) = ((x - 2400 \times 3/4)/2) + 5 \quad (2400 \leqslant x \leqslant 4000).$$

公路从山脚（0，800）处开始，经居民点（4000，2000）至矿区（2000，4000）. 已知路段工程成本及对路段坡度 α（上升高程与水平距离之比）的限制如表 2 所示.

(1) 试给出一种线路设计方案，包括原理、方法及比较精确的线路位置（含桥梁、隧道），并估算该方案的总成本.

(2) 如果居民点改为 $3600 \leqslant x \leqslant 4000$，$2000 \leqslant y \leqslant 2400$ 的居民区，公路只须经过居民区即可，那么你的方案有什么改变？

表 1

4800	1350	1370	1390	1400	1410	960	940	880	800	690	570	430	290	210	15
4400	1370	1390	1410	1430	1440	1140	1110	1050	950	820	690	540	380	300	21

续表

4000	1380	1410	1430	1450	1470	1320	1280	1200	1080	940	780	620	460	370	35
3600	1420	1430	1450	1480	1500	1550	1510	1430	1300	1200	980	850	750	550	50
3200	1430	1450	1460	1500	1550	1600	1550	1600	1600	1600	1550	1500	1500	1550	155
2800	950	1190	1370	1500	1200	1100	1550	1600	1550	1380	1070	900	1050	1150	120
2400	910	1090	1270	1500	1200	1100	1350	1450	1200	1150	1010	880	1000	1050	110
2000	880	1060	1230	1390	1500	1500	1400	900	1100	1060	950	870	900	930	95
1600	830	980	1180	1320	1450	1420	1400	1300	700	900	850	840	380	780	75
1200	740	880	1080	1130	1250	1280	1230	1040	900	500	700	780	750	650	55
800	650	760	880	970	1020	1050	1020	830	800	700	300	500	550	480	35
400	510	620	730	800	850	870	850	780	720	650	500	200	300	350	32
0	730	470	550	600	670	690	670	620	580	450	400	300	100	150	25
y/x	0	400	800	1200	1600	2000	2400	2800	3200	3600	4000	4400	4800	5200	560

表 2

工程种类	一般路段	桥　梁	遂　道
工程成本（元/米）	300	2 000	1 500（长度≤300 米）；3 000（长度 >300 米）
对坡度 α 的限制	$\alpha<0.125$	$\alpha=0$	$\alpha<0.100$

1994 年全国大学生数学建模竞赛题

B 题　锁具装箱

某锁厂生产一种弹子锁具，每个锁具的钥匙有 5 个槽，每个槽的高度从 {1,2,3,4,5,6} 6 个数（单位略）中任取一数. 由于工艺及其他原因，制造锁具时对 5 个槽的高度还有两个限制：至少有 3 个不同的数；相邻两槽高度之差

不能为 5. 满足以上条件制造出来的所有互不相同的锁具称为一批. 从顾客的利益出发，自然希望在每批锁具中“一把钥匙开一把锁”. 但是在当前工艺条件下，对于同一批中两个锁具是否能够互开，有以下试验结果：若二者相对应的 5 个槽的高度中有 4 个相同，另一个的高度差为 1，则可能互开；在其他情形下，不可能互开. 销售部门在一批锁具中随意地抽取每 60 个装一箱出售. 团体顾客往往购买几箱到几十箱，他们抱怨购得的锁具会出现互相开的情形. 现聘请你为顾问，回答并解决以下问题：

(1) 每一批锁具有多少个？装多少箱？

(2) 为销售部门提供一种方案，包括如何装箱（仍是 60 个锁具一箱），如何给箱子以标志？出售时如何利用这些标志？使团体顾客不再抱怨或减少抱怨.

(3) 采取你提出的方案，团体顾客的购买量不超过多少箱，就可以保证一定不会出现互开？

(4) 按照原来的装箱办法，如何定量地衡量团体顾客抱怨互开的程度（试对购买一、二箱者给出具体结果）？

1995 年全国大学生数学建模竞赛题

A 题 一个飞行管理模型

在约 10 000m 高空的某边长 160km 的正方形区域内，经常有若干架飞机作水平飞行. 区域内每架飞机的位置和速度均由计算机记录其数据，以便进行飞行管理. 当一架欲进入该区域的飞机到达区域边缘，记录其数据后，要立即计算并判断是否会与区域内的飞机发生碰撞. 如果会碰撞，则应计算如何调整各架（包括新进入的）飞机的飞行方向角度，以避免碰撞. 现假定条件如下：

(1) 不碰撞的标准为任意两架飞机的距离大于 8km；

(2) 飞机飞行方向角调整的幅度不应超过 30°；

(3) 所有飞机飞行速度均为每小时 800km；

(4) 进入该区域的飞机在到达区域边缘时，与区域内飞机的距离应在 60km 以上；

(5) 最多需考虑 6 架飞机；

(6) 不必考虑飞机离开该区域后的状况.

试对这个避免碰撞的飞行管理问题建立数学模型，列出计算步骤，对以下数据进行计算（方向角误差不超过 0.01°），要求飞机飞行方向角调整的幅度尽量小. 设该区域 4 个顶点的坐标为 (0,0)，(160,0)，(160,160)，(0,160). 记录数据为：

飞机编号	横坐标 x	纵坐标 y	方向角 (°)
1	150	140	243
2	85	85	236
3	150	155	220.5
4	145	50	159
5	130	150	230

新进入 0 0 52

注：方向角指飞行方向与 x 轴正向的夹角. 试根据实际应用背景对你的模型进行评价与推广.

B 题　天车与冶炼炉的作业调度

某钢铁厂冶炼车间的厂房布局是，地面沿一直线依次安置着 7 个工作点辅料供应处 p；a 组 3 座转炉（冶炼成品钢）a_1，a_2，a_3；b 组 2 座冶炼炉（冶炼半成品钢，简称半钢）b_1，b_2；原料供应处 q. 这些设备的上方贯通着一条运送物料的天车轨道，上面布置着若干天车 t_1，t_2，…，t_n. 布局示意图如图 1 所示.

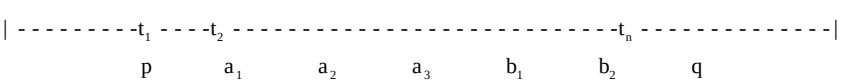


图 1

天车、冶炼炉的作业过程与工序为：天车从 q 处吊起原料一罐（吊罐时间 t_y）运至 b_1 或 b_2 处放下（放罐时间 t_i），并将上一炉的原料空罐吊起（吊空时间 t_0）返回 q 处放下（放空罐时间 t_k）. b 组炉的原料罐放下后即可在辅助作业下开始冶炼（冶炼时间 t_b），由天车吊起半钢罐（吊罐时间 t_d）运至 a_1 或 a_2、a_3 处将半钢倒入转炉（倒入时间 t_e），并将空罐返回 b_1 或 b_2 处放下（放空罐时间 t_c）. 再由天车从 p 处吊起辅料一槽（吊起时间 t_g）运至 a_1 或 a_2、a_3 处加入转炉（加入时间 t_f），并将空槽返 b_1 或 b_2 处放下（放空罐时间 t_c）. 再由天车从 p 处吊起辅料一槽（吊起时间 t_g）运至 a_1 或 a_2、a_3 处加入转炉（加入时间 t_f），并将空槽返回 p 处放下（放空槽时间 t_h）. a 组炉在半钢和辅料加入后即可开始冶炼（冶炼时间 t_a），冶炼后成品钢输出不用天车（输出时间记入 t_a）. 天车通过相邻两个工作点的运行时间都相同，记为 t_x. 由于各台天车在同一轨道上运行，因此其顺序位置 t_1，t_2，…，t_n 不可交换. 在同一时间同一座炉子上只能允许一台天车作业；但 p、q 两处可以允许多台天车同时作业. 在 p，a_1，…，q 每两个相邻工作点之间最多能容

纳 2 台天车同时停放. 天车与冶炼炉作业调度的要求为：

(1) 成品钢产量尽量高；

(2) 各台天车的作业率 (天车作业时间所占比例) 尽量均衡 (考虑到设备人员安全等因素，一般天车作业率不超过 70%)；

(3) 绝不允许天车相撞等事故；

(4) 调度规则尽量简明，以利于现场人员使用.

现设定：$t_a = 48$，$t_b = 27$，$t_i = 3$，$t_0 = 2$，$t_c = 2$，$t_d = 3$，$t_e = 5$，$t_f = 2$，$t_g = 2$，$t_h = 1$，$t_y = 3$，$t_k = 2$ (单位：min)，$t_x = 15s$；a 组炉平均每炉产量 $w_a = 120t$. 试在不超过 5 台天车的条件下，设计一种满足上述要求的天车与冶炼炉的作业调度方案：

(1) 各台天车负责哪些作业 (列出工序清单)；

(2) 在所给方案的一个周期内，每一时刻天车和冶炼炉处于什么状态 (绘出天车 - 炉子作业运行图)；

(3) 一份供现场人员使用的调度规则说明书；

(4) 在所给方案下计算各台天车的作业率.

并按每天冶炼炉数估计该车间成品钢的年产量 (扣除设备维修日，每台转炉作业日每年按 300 天计算). 实际生产中，t_a，t_b，…，t_k 都是随机的 (上面设定的数值可视为平均值)，讨论你的调度方案如何适用于实际生产过程. 试提出该车间提高钢产量到年产 300 万吨以上的建议.

1996 年全国大学生数学建模竞赛题

A 题　最优捕鱼策略

为了保护人类赖以生存的自然环境，可再生资源 (如渔业、林业资源) 的开发必须适度. 一种合理、简化的策略是，在实现可持续收获的前提下，追求最大产量或最佳效益.

考虑对某种鱼 (鲳鱼) 的最优捕捞策略：

假设这种鱼分 4 个年龄组：称 1 龄鱼，…，4 龄鱼. 各年龄组每条鱼的平均重量分别为 5.07，11.55，17.86，22.99 (g)；各年龄组鱼的自然死亡率均为 0.8 (1/年)；这种鱼为季节性集中产卵繁殖，平均每条 4 龄鱼的产卵量为 1.109×10^5 (个)；3 龄鱼的产卵量为这个数的一半，2 龄鱼和 1 龄鱼不产卵，产卵和孵化期为每年的最后 4 个月；卵孵化并成活为 1 龄鱼，成活率为 $1.22 \times 10^{11} / (1.22 \times 10^{11} + n)$ (n 为产卵总量).

渔业管理部门规定，每年只允许在产卵孵化期前的 8 个月内进行捕捞作业.

如果每年投入的捕捞能力（如渔船数、下网次数等）固定不变，这时单位时间捕捞量将与各年龄组鱼群条数成正比. 比例系数不妨称捕捞强度系数. 通常使用13mm 网眼的拉网，这种网只能捕捞 3 龄鱼和 4 龄鱼，其两个捕捞强度系数之比为 0.42:1. 渔业上称这种方式为固定努力量捕捞.

(1) 建立数学模型分析如何可持续捕获（即每年开始捕捞时渔场中各年龄组鱼群不变），并且在此前提下得到最高的年收获量（捕捞总重量）.

(2) 某渔业公司承包这种鱼的捕捞业务 5 年，合同要求鱼群的生产能力不能受到太大的破坏，已知承包时各年龄组鱼群的数量分别为：122，29.7，10.1，3.29（$\times 10^9$ 条），如果仍用固定努力量的捕捞方式，该公司采取怎样的策略才能使总收获量最高.

B 题　节水洗衣机

我国淡水资源有限，节约用水人人有责. 洗衣机在家庭用水中占有相当大的份额，目前洗衣机已非常普及，节约洗衣机用水十分重要. 假设在放入衣物和洗涤剂后洗衣机的运行过程为：加水-漂水-脱水-加水-漂水-脱水-…-加水-漂水-脱水（称“加水-漂水-脱水”为运行一轮）. 试为洗衣机设计一种程序（包括运行多少轮、每轮加多少水等），使得在满足一定洗涤效果的条件下，总用水量最少. 选用合理的数据进行计算. 对照目前常用的洗衣机的运行情况，对你的模型和结果作出评价.

1997 年全国大学生数学建模竞赛题

B 题　截断切割

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. 这里“截断切割”是指将物体沿某个切割平面分成两部分. 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 6 次截断切割. 设水平切割单位面积的费用是垂直切割单位面积费用的 r 倍，且当先后两次垂直切割的平面（不管它们之间是否穿插水平切割）不平行时，因调整刀具需额外费用 e. 试为这些部门设计一种安排各面加工次序（称“切割方式”）的方法，使加工费用最少（由工艺要求，与水平工作台接触的长方体底面是事先指定的）. 详细要求如下：

(1) 需考虑不同切割方式的总数.

(2) 给出上述问题的数学模型和求解方法.

(3) 试对某部门用的如下准则作出评价:每次选择一个加工费用最少的待切割面进行切割.

(4) 对于 $e = 0$ 的情形有无简明的优化准则.

(5) 用以下实例验证你的方法: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 时,二者左侧面、正面、底面之间的距离分别为 6、7、9 (单位均为 cm).

垂直切割费用为每平方厘米 1 元,r 和 e 的数据有以下 4 组:a. $r=1$, $e=0$;b. $r=1.5$, $e=0$;c. $r=8$, $e=0$;d. $r=1.5$, $2\leqslant e\leqslant 15$. 对最后一组数据应给出所有最优解,并进行讨论.

1999 年创维杯全国大学生数学建模竞赛题

A 题 自动化车床管理

一道工序用自动化车床连续加工某种零件,由于刀具损坏等原因该工序会出现故障,其中刀具损坏故障占 95%,其他故障仅占 5%. 工序出现故障是完全随机的,假定在生产任一零件时出现故障的机会均相同. 工作人员通过检查零件来确定工序是否出现故障. 现积累有 100 次刀具故障记录,故障出现时该刀具完成的零件数如表 1 所示. 现计划在刀具加工一定件数后定期更换新刀具.

已知生产工序的费用参数如下:

故障时产出的零件损失费用 $f=200$ 元/件;

进行检查的费用 $t=10$ 元/次;

发现故障进行调节使恢复正常的平均费用 $d=3\ 000$ 元/次(包括刀具费);

未发现故障时更换一把新刀具的费用 $k=1\ 000$ 元/次.

(1) 假定工序故障时产出的零件均为不合格品,正常时产出的零件均为合格品,试对该工序设计效益最好的检查间隔(生产多少零件检查一次)和刀具更换策略.

(2) 如果该工序正常时产出的零件有 2% 为不合格品;而工序故障时产出的零件有 40% 为合格品,60% 为不合格品. 工序正常而误认为有故障停机产生的损失费用为 1 500 元/次. 试对该工序设计效益最好的检查间隔和刀具更换策略.

(3) 在 (2) 的情况,能否改进检查方式获得更高的效益?

表 1　　100 次刀具故障记录完成的零件数

459	362	624	542	509	584	433	748	815	505
612	452	434	982	640	742	565	706	593	680
926	653	164	487	734	608	428	1153	593	844
527	552	513	781	474	388	824	538	862	659
775	859	755	649	697	515	628	954	771	609
402	960	885	610	292	837	473	677	358	638
699	634	555	570	84	416	606	1062	484	120
447	654	564	339	280	246	687	539	790	581
621	724	531	512	577	496	468	499	544	645
764	558	378	765	666	763	217	715	310	851

B 题　钻井布局

勘探部门在某地区找矿. 初步勘探时期已零散地在若干位置上钻井,取得了地质资料. 进入系统勘探时期后,要在一个区域内按纵横等距的网格点来布置井位,进行"撒网式"全面钻探. 由于钻一口井的费用很高,如果新设计的井位与原有井位重合(或相当接近),便可利用旧井的地质资料,不必打这口新井,因此,应尽量利用旧井,少打新井,以节约钻探费用. 比如钻一口新井的费用为 500 万元,利用旧井资料的费用为 10 万元,则利用一口旧井就节约费用 490 万元.

设平面上有 n 个点 P_i,其坐标为 (a_i, b_i),$i = 1, 2, \cdots, n$,表示已有的 n 个井位. 新布置的井位是一个正方形网格 N 的所有结点(所谓"正方形网格"是指每个格子都是正方形的网格,结点是指纵线和横线的交叉点). 假定每个格子的边长(井位的纵横间距)都是 1 单位(比如 100m). 整个网格是可以在平面上任意移动的. 若一个已知点 P_i 与某个网格结点 X_i 的距离不超过给定误差 ε ($= 0.05$ 单位),则认为 P_i 处的旧井资料可以利用,不必在结点 X_i 处打新井.

为进行辅助决策,勘探部门要求我们研究如下问题:

(1) 假定网格的横向和纵向是固定的(比如东西向和南北向),并规定两点间的距离为其横向距离(横坐标之差绝对值)及纵向距离(纵坐标之差绝对值)的最大值. 在平面上平行移动网格 N,使可以利用的旧井数尽可能大. 试提供数值计算方法,并对下面的数值例子用计算机进行计算.

(2) 在欧氏距离的误差意义下,考虑网格的横向和纵向不固定(可以旋转)

的情形,给出算法及计算结果.

(3) 如果有 n 口旧井,给出判定这些井均可以利用的条件和算法(可以任意选定一种距离).

数值例子 $n=12$ 个点的坐标如下所示:

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

a_i:0.50 1.41 3.00 3.37 3.40 4.72 4.72 5.43 7.57 8.38 8.98 9.50

b_i:2.00 3.50 1.50 3.51 5.50 2.00 6.24 4.10 2.01 4.50 3.41 0.80.

1999 年创维杯全国大学生数学建模竞赛题(大专组)

C 题 煤矸石堆积

煤矿采煤时,会产出无用废料煤矸石.在平原地区,煤矿管理部门不得不征用土地堆放矸石.通常矸石的堆积方法是:

架设一段与地面角度约为 $\beta=25°$ 的直线形上升轨道(角度过大,运矸车无法装满),用在轨道上行驶的运矸车将矸石运到轨道顶端后向两侧倾倒,待矸石堆高后,再借助矸石堆延长轨道,这样逐渐堆起一座矸石山来.

现给出下列数据:

矸石自然堆放安息角(矸石自然堆积稳定后,其坡面与地面形成的夹角) $\alpha\leqslant 55°$;

矸石容重(碎矸石单位体积的重量)约 $2t/m^3$;

运矸车所需电费为 0.50 元/度(不变);

运矸车机械效率(只考虑堆积坡道上的运输)初始值(在地平面上)约 30%,坡道每延长 10 m,效率在原有基础上约下降 2%;

土地征用费现值为 8 万元/亩,预计地价年涨幅约 10%;

银行存、贷款利率均为 5%;

煤矿设计原煤产量为 300 万吨/年;

煤矿设计寿命为 20 年;

采矿出矸率(矸石占全部采出的百分比)一般为 7% ~10%.

另外,为保护耕地,煤矿堆矸土地应比实际占地多征用 10%.

现在煤矿设计中用于处理矸石的经费(只计征地费及堆积时运矸车用的电费)为 100 万元/年,这笔钱是否够用?试制定合理的年度征地计划,并对不同

的出矸率预测处理矸石的最低费用.

1998 年全国大学生数学建模竞赛题

A 题　投资的收益和风险

市场上有 n 种资产（如股票、债券等）S_i（$i=1,\cdots,n$）供投资者选择，某公司有数额为 M 的一笔相当大的资金可以用做一个时期的投资. 公司财务分析人员对这 n 种资产进行了评估，估算出在这一时期内购买 S_i 的平均收益率为 r_i，并预测出购买 S_i 的风险损失率为 q_i. 考虑到投资越分散，总的风险越小，公司确定，当用这笔资金购买若干种资产时，总体风险可以用所投资的 S_i 中最大的一个风险来度量. 购买 S_i 要付交易费，费率为 p_i，并且当购买额不超过给定值 u_i 时，交易费按购买 u_i 计算（不买当然无须付费）. 另外，假定同期银行存款利率是 r_0，且既无交易费又无风险（$r_0=5\%$）.

(1) 已知 $n=4$ 时的相关数据如表 1 所示.

表 1

S_i	r_i /（%）	q_i /（%）	p_i /（%）	u_i /（元）
S_1	28	2.5	1	103
S_2	21	1.5	2	198
S_3	23	5.5	4.5	52
S_4	25	2.6	6.5	40

(2) 试为该公司设计一种投资组合方案，即用给定的资金 M，有选择地购买若干种资产或存银行生息，使净收益尽可能大，而总体风险尽可能小.

(3) 试就一般情况对以上问题进行讨论，并利用表 2 数据进行计算.

表 2

S_i	r_i /（%）	q_i /（%）	p_i /（%）	u_i /（元）
S_1	9.6	42	2.1	181

续表

S_i	r_i / (%)	q_i / (%)	p_i / (%)	u_i / (元)
S_2	18.5	54	3.2	407
S_3	49.4	60	6.0	428
S_4	23.9	42	1.5	549
S_5	8.1	1.2	7.6	270
S_6	14	39	3.4	397
S_7	40.7	68	5.6	178
S_8	31.2	33.4	3.1	220
S_9	33.6	53.3	2.7	475
S_{10}	36.8	40	2.9	248
S_{11}	11.8	31	5.1	195
S_{12}	9	5.5	5.7	320
S_{13}	35	46	2.7	267
S_{14}	9.4	5.3	4.5	328
S_{15}	15	23	7.6	131

2000 年网易杯全国大学生数学建模竞赛题

A 题　DNA 序列分类

2000 年 6 月,人类基因组计划中 DNA 全序列草图完成,预计 2001 年可以完成精确的全序列图,此后人类将拥有一本记录着自身生老病死及遗传进化的全部信息的“天书”. 这本大自然写成的“天书”是由 4 个字符 A,T,C,G 按一定顺序排成的长约 30 亿字符的序列,其中没有“断句”也没有标点符号,除了这 4 个字符表示 4 种碱基以外,人们对它包含的“内容”知之甚少,难以读懂. 破译这部世界上最巨量信息的“天书”是 21 世纪最重要的任务之一. 在这个目标中,研究 DNA 全序列具有什么结构,由这 4 个字符排成的看似随机的序列中隐藏着什么规律,又是解读这部天书的基础,是生物信息学最重要的课题之一. 虽然人类对这部“天书”知之甚少,但也发现了 DNA 序列中的一些规律性和结构. 例如,

在全序列中有一些是用于编码蛋白质的序列片段,即由这 4 个字符组成的 64 种不同的 3 字符串,其中大多数用于编码构成蛋白质的 20 种氨基酸. 又例如,在不用于编码蛋白质的序列片段中,A 和 T 的含量特别多些,于是以某些碱基特别丰富作为特征去研究 DNA 序列的结构也取得了一些结果. 此外,利用统计的方法还发现序列的某些片段之间具有相关性,等等. 这些发现让人们相信,DNA 序列中存在着局部的和全局性的结构,充分发掘序列的结构对理解 DNA 全序列是十分有意义的. 目前在这项研究中最普通的思想是省略序列的某些细节,突出特征,然后将其表示成适当的数学对象. 这种被称为粗粒化和模型化的方法往往有助于研究规律性和结构.

作为研究 DNA 序列的结构的尝试,提出以下对序列集合进行分类的问题:

(1) 下面有 20 个已知类别的人工制造的序列,其中序列标号 1 ~ 10 为 A 类,11 ~ 20 为 B 类. 试从中提取特征,构造分类方法,并用这些已知类别的序列,衡量你的方法是否足够好. 然后用你认为满意的方法,对另外 20 个未标明类别的人工序列 (标号 21 ~ 40) 进行分类,把结果用序号 (按从小到大的顺序) 标明它们的类别 (无法分类的不写入):A 类;B 类. 试详细描述你的方法,给出计算程序. 如果部分地使用了现成的分类方法,也要将方法名称准确注明.

(2) 在同样网址的数据文件 Nat-model-data 中给出了 182 个自然 DNA 序列,它们都较长. 用你的分类方法对它们进行分类,像 (1) 一样地给出分类结果.

提示:衡量分类方法优劣的标准是分类的正确率,构造分类方法有许多途径,例如提取序列的某些特征,给出它们的数学表示:几何空间或向量空间的元素等,然后再选择或构造适合这种数学表示的分类方法;又例如构造概率统计模型,然后用统计方法分类等.

1. aggcacggaaaaacgggaataacggaggaggacttggcacggcattacacggaggacgaggtaaaggaggcttgtctac ggccggaagtgaagggggatatgaccgcttgg
2. cggaggacaaacgggatggcggtattggaggtggcggactgttcggggaattattcggtttaaacgggacaaggaaggcgg ctggaacaaccggacggtggcagcaaagga
3. gggacggatacggattctggccacggacggaaaggaggacacggcggacatacacggcggcaacggacggaacggag gaaggagggcggcaatcggtacggaggcggcgga
4. atggataacggaaacaaaccagacaaacttcggtagaaatacagaagcttagatgcatatgtttttaaataaaatttgtatta ttatggtatcataaaaaaaggttgcga
5. cggctggcggacaacggactggcggattccaaaaacggaggaggcggacggaggctacaccaccgtttcggcggaaag gcggagggctggcaggaggctcattacggggag
6. atggaaaattttcggaaaggcggcaggcaggaggcaaaggcggaaaggaaggaaacggcggatatttcggaagtggatat taggagggcggaataaaggaacggcggcaca

7. atgggattattgaatggcggaggaagatccggaataaaatatggcggaaagaacttgttttcggaaatggaaaaaggactag gaatcggcggcaggaaggatatggaggcg
8. atggccgatcggcttaggctggaaggaacaaataggcggaattaaggaaggcgttctcgcttttcgacaaggaggcggacc ataggaggcggattaggaacggttatgagg
9. atggcggaaaaaggaaatgtttggcatcggcgggctccggcaactggaggttcggccatggaggcgaaaatcgtgggcgg cggcagcgctggccggagtttgaggagcgcg
10. tggccgcggaggggcccgtcgggcgcggatttctacaagggcttcctgttaaggaggtggcatccaggcgtcgcacgctc ggcgcggcaggaggcacgcgggaaaaaacg
11. gttagatttaacgtttttatggaatttatggaattataaatttaaaaatttatatttttttaggtaagtaatccaacgtttttattactttt ttaaaattaaatatttatt
12. gtttaattactttatcatttaatttaggttttaattttaaatttaatttaggtaagatgaatttggtttttttttaaggtagttatttaattat cgttaaggaaagttaaa
13. gtattacaggcagaccttatttaggttattattattatttggatttttttttttttttttttttaagttaaccgaattattttctttaaagacg ttacttaatgtcaatgc
14. gttagtctttttttagattaaattattagattatgcagtttttttacataagaaaattttttttttcggagttcatattctaatctgtctttat taaatcttagagatatta
15. gtattatattttttttattttttattattttagaatataatttgaggtatgtgtttaaaaaaaaattttttttttttttttttttttttttttttaaaat ttataaatttaa
16. gttatttttaaatttaattttaattttaaaatacaaaatttttactttctaaaattggtctctggatcgataatgtaaacttattgaatc tatagaattacattattgat
17. gtatgtctatttcacggaagaatgcaccactatatgatttgaaattatctatggctaaaaaccctcagtaaaatcaatcccta aacccttaaaaaacggcggcctatccc
18. gttaattatttattccttacgggcaattaattatttattacggttttatttacaatttttttttttgtcctatagagaaattacttacaaa acgttattttacatactt
19. gttacattatttattattatccgttatcgataatttttttacctctttttcgctgagtttttattcttactttttttcttctttatataggatc tcatttaatatcttaa
20. gtatttaactctctttactttttttttcactctctacattttcatcttctaaaactgtttgatttaaacttttgtttctttaaggatttttttt tacttatcctctgttat
21. tttagctcagtccagctagctagtttacaatttcgacaccagtttcgcaccatcttaaatttcgatccgtaccgtaatttagct tagatttggatttaaaggatttagattga
22. tttagtacagtagctcagtccaagaacgatgtttaccgtaacgtqacgtaccgtacgctaccgttaccggattccggaaag ccgattaaggaccgatcgaaaggg
23. cgggcggatttaggccgacggggacccgggattcgggacccgaggaaattcccggattaaggtttagcttcccgggatt tagggcccggatggctgggaccc

24. tttagctagctactttagctatttttagtagctagccagcctttaaggctagctttagctagcattgttctttattgggacccaag
ttcgacttttacgatttagttttgaccgt
25. gaccaaaggtgggctttagggacccgatgctttagtcgcagctggaccagttccccagggtattaggcaaaagctgacg
ggcaattgcaatttaggcttaggcca
26. gatttactttagcatttttagctgacgttagcaagcattagctttagccaatttcgcatttgccagtttcgcagctcagttttaac
gcgggatctttagcttcaagctttttac
27. ggattcggatttacccggggattggcggaacgggacctttaggtcgggacccattaggagtaaatgccaaaggacgctg
gtttagccagtccgttaaggcttag
28. tccttagatttcagttactatatttgacttacagtctttgagatttcccttacgattttgacttaaaatttagacgttagggcttatc
agttatggattaatttagcttattttcga
29. ggccaattccggtaggaaggtgatggcccgggggttcccgggaggatttaggctgacgggccggccatttcggtttaggg
agggccgggacgcgttagggc
30. cgctaagcagctcaagctcagtcagtcacgtttgccaagtcagtaatttgccaaagttaaccgttagctgacgctgaacgct
aaacagtattagctgatgactcgta
31. ttaaggacttaggctttagcagttactttagtttagttccaagctacgtttacgggaccagatgctagctagcaatttattatcc
gtattaggcttaccgtaggtttagcgt
32. gctaccgggcagtctttaacgtagctaccgtttagtttgggcccagccttgcggtgtttcggattaaattcgttgtcagtcgc
tctrtgggtttagtcattcccaaaagg
33. cagttagctgaatcgtttagccatttgacgtaaacatgattttacgtacgtaaattttagccctgacgtttagctaggaatttat
gctgacgtagcgatcgactttagcac
34. cggttagggcaaaggttggatttcgacccagggggaaagcccgggacccgaacccagggctttagcgtaggctgacgc
taggcttaggttggaacccggaaa
35. gcggaagggcgtaggtttgggatgcttagccgtaggctagctttcgacacgatcgattcgcaccacaggataaaagttaa
gggaccggtaagtcgcggtagcc
36. ctagctacgaacgctttaggcgcccccgggagtagtcgttaccgttagtatagcagtcgcagtcgcaattcgcaaaagtc
cccagctttagccccagagtcgacg
37. gggatgctgacgctggttagctttaggcttagcgtagctttagggccccagtctgcaggaaatgcccaaaggaggccca
ccgggtagatgccasagtgcaccgt
38. aacttttagggcatttccagttttacgggttattttcccagttaaactttgcaccattttacgtgttacgatttacgtataatttgac
cttattttggacactttagtttgggttac
39. ttagggccaagtcccgaggcaaggaattctgatccaagtccaatcacgtacagtccaagtcaccgtttgcagctaccgtt
taccgtacgttgcaagtcaaatccat
40. ccattagggtttatttacctgtttattttttcccgagaccttaggtttaccgtactttttaacggtttacctttgaaatttttggactag
cttaccctggatttaacggccagttt

2001 年全国大学生数学建模竞赛题

A 题　血管的三维重建

样本断面可用于了解生物组织、器官等的形态. 例如,将样本染色后切成厚约 1μm 的切片,在显微镜下观察该横断面的组织形态结构. 如果用切片机连续不断地将样本切成数十、成百的平行切片,可以依次逐片观察. 根据拍照及采样得到的平行切片数字图像,运用计算机可以重建组织、器官等准确的三维形态.

假设某些血管可视为一类特殊的管道,该管道的表面是由球心沿着某一曲线(称为中轴线)的球滚动包络而成. 例如圆柱就是这样一种管道,其中轴线为直线,由半径固定的球滚动包络形成.

现有某管道的相继 100 张平行切片图像,记录了管道与切片的交. 图像文件名依次为 0. bmp,1. bmp,…, 99. bmp,格式均为 bmp,宽、高均为 512 个像素. 为简化起见,假设管道中轴线与每张切片有且只有一个交点;球半径固定;切片间距以及图像像素的尺寸均为 1.

取坐标系的 Z 轴垂直于切片,第 1 张切片为平面 Z = 0,第 100 张切片为平面 Z = 99 如图 1 所示. Z = z 切片图像中像素的坐标依它们在文件中出现的前后次序为

$$
\begin{array}{lll}
(-256,-256,z),(-256,-255,z),\cdots & (-256,255,z), \\
(-255,-256,z),(-255,-255,z),\cdots & (-255,255,z), \\
\vdots \qquad\qquad\qquad \vdots & \qquad\qquad \vdots \\
(255,-256,z),\ (255,-255,z),\cdots & (255,255,z).
\end{array}
$$

试计算管道的中轴线与半径,给出具体的算法,并绘制中轴线在 XOY、YOZ、ZOX 平面的投影图.

B 题　公交车调度

公共交通是城市交通的重要组成部分,做好公交车的调度对于完善城市交通环境、改进市民出行状况、提高公交公司的经济效益和社会效益,都具有重要意义. 下面考虑一条公交线路上公交车的调度问题,其数据来自我国一座特大城市某条公交线路的客流调查和运营资料.

该条公交线路上行方向共 14 站,下行方向共 13 站,表 1 与表 2 给出的是典型的一个工作日两个运行方向各站上、下车的乘客数量统计. 公交公司配给该线

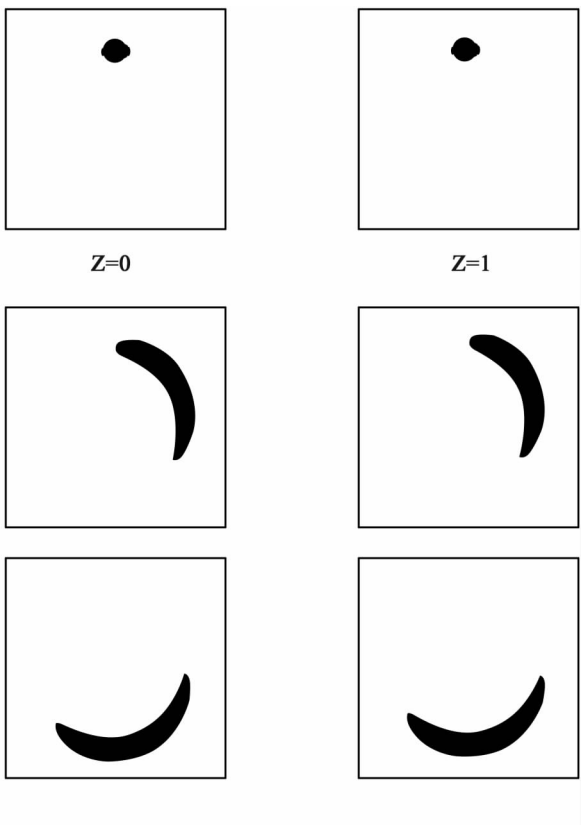


图 1

路同一型号的大客车，每辆标准载客 100 人，据统计客车在该线路上运行的平均速度为 20km/h. 运营调度要求，乘客候车时间一般不要超过 10min，早晨高峰时一般不要超过 5min，车辆满载率不应超过 120%，一般也不要低于 50%.

试根据这些资料和要求，为该线路设计一个便于操作的全天（工作日）的公交车调度方案，包括两个起点站的发车时刻表；一共需要多少辆车；这个方案以怎样的程度照顾到了乘客和公交公司双方的利益等.

如何将这个调度问题抽象成一个明确、完整的数学模型？指出求解模型的方法；根据实际问题的要求，如果要设计更好的调度方案，应如何采集运营数据？

表 1　　某路公交汽车各时组每站上、下车人数统计表　　(上行方向：A_{13} 开往 A_0)

站名		A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
站间距/(km)			1.6	0.5	1	0.73	2.04	1.26	2.29	1	1.2	0.4	1	1.03	0.53
5:00～6:00	上	371	60	52	43	76	90	48	83	85	26	45	45	11	0
	下	0	8	9	13	20	48	45	81	32	18	24	25	85	57
6:00～7:00	上	1990	376	333	256	589	594	315	622	510	176	308	307	68	0
	下	0	99	105	164	239	588	542	800	407	208	300	288	921	615
7:00～8:00	上	3626	634	528	447	948	868	523	958	904	259	465	454	99	0
	下	0	205	227	272	461	1058	1097	1793	801	469	560	636	1871	1459
8:00～9:00	上	2064	322	305	235	477	549	271	486	439	157	275	234	60	0
	下	0	106	123	169	300	634	621	971	440	245	339	408	1132	759
9:00～10:00	上	1186	205	166	147	281	304	172	324	267	78	143	162	36	0
	下	0	81	75	120	181	407	411	551	250	136	187	233	774	483
10:00～11:00	上	923	151	120	108	215	214	119	212	201	75	123	112	26	0
	下	0	52	55	81	136	299	280	442	178	105	153	167	532	385
11:00～12:00	上	957	181	157	133	254	264	135	253	260	74	138	117	30	0
	下	0	54	58	84	131	321	291	420	196	119	159	153	534	340
12:00～13:00	上	873	141	140	108	215	204	129	232	221	65	103	112	26	0
	下	0	46	49	71	111	263	256	389	164	111	134	148	488	333
13:00～14:00	上	779	141	103	84	186	185	103	211	173	66	108	97	23	0
	下	0	39	41	70	103	221	197	297	137	85	113	116	384	263
14:00～15:00	上	625	104	108	82	162	180	90	185	170	49	75	85	20	0
	下	0	36	39	47	78	189	176	339	139	80	97	120	383	239
15:00～16:00	上	635	124	98	82	152	180	80	185	150	49	85	85	20	0
	下	0	36	39	57	88	209	196	339	129	80	107	110	353	229
16:00～17:00	上	1 493	299	240	199	396	404	210	428	390	120	208	197	49	0
	下	0	80	85	135	194	450	441	731	335	157	255	251	800	557
17:00～18:00	上	2 011	379	311	230	497	479	296	586	508	140	250	259	61	0

续表

站名		A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	下	0	110	118	171	257	694	573	957	390	253	293	378	1228	793
18:00～19:00	上	691	124	107	89	167	165	108	201	194	53	93	82	22	0
	下	0	45	48	80	108	237	231	390	150	89	131	125	428	336
19:00～20:00	上	350	64	55	46	91	85	50	88	89	27	48	47	11	0
	下	0	22	23	34	63	116	108	196	83	48	64	66	204	139
20:00～21:00	上	304	50	43	36	72	75	40	77	60	22	38	37	9	0
	下	0	16	17	24	38	80	84	143	59	34	46	47	160	117
21:00～22:22	上	209	37	32	26	53	55	29	47	52	16	28	27	6	0
	下	0	14	14	21	33	78	63	125	62	30	40	41	128	92
22:00～23:00	上	19	3	3	2	5	5	3	5	5	1	3	2	1	0
	下	0	3	3	5	8	18	17	27	12	7	9	9	32	21

表 2　　某路公交汽车各时组每站上、下车人数统计表　(下行方向:A_0 开往 A_{13})

站名		A_0	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}
站间距/(km)			1.56	1	0.44	1.2	0.97	2.29	1.3	2	0.73	1	0.5	1.62
5:00～6:00	上	22	3	4	2	4	4	3	3	3	1	1	0	0
	下	0	2	1	1	6	7	7	5	3	4	2	3	9
6:00～7:00	上	795	143	167	84	151	188	109	137	130	45	53	16	0
	下	0	70	40	40	184	205	195	147	93	109	75	108	271
7:00～8:00	上	2328	380	427	224	420	455	272	343	331	126	138	45	0
	下	0	294	156	157	710	780	849	545	374	444	265	373	958
8:00～9:00	上	2706	374	492	224	404	532	333	345	354	120	153	46	0
	下	0	266	158	149	756	827	856	529	367	428	237	376	1167
9:00～10:00	上	1556	204	274	125	235	308	162	203	198	76	99	27	0
	下	0	157	100	80	410	511	498	336	199	276	136	219	556
10:00～11:00	上	902	147	183	82	155	206	120	150	143	50	59	18	0

续表

站名		A_0	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}
	下	0	103	59	59	246	346	320	191	147	185	96	154	438
11:00～12:00	上	847	130	132	67	127	150	108	104	107	41	48	15	0
	下	0	94	48	48	199	238	256	175	122	143	68	128	346
12:00～13:00	上	706	90	118	66	105	144	92	95	88	34	40	12	0
	下	0	70	40	40	174	215	205	127	103	119	65	98	261
13:00～14:00	上	770	97	126	59	102	133	97	102	104	36	43	13	0
	下	0	75	43	43	166	210	209	136	90	127	60	115	309
14:00～15:00	上	839	133	156	69	130	165	101	118	120	42	49	15	0
	下	0	84	48	48	219	238	246	155	112	153	78	118	346
15:00～16:00	上	1110	170	189	79	169	194	141	152	166	54	64	19	0
	下	0	110	73	63	253	307	341	215	136	167	102	144	425
16:00～17:00	上	1837	260	330	146	305	404	229	277	253	95	122	34	0
	下	0	175	96	106	459	617	549	401	266	304	162	269	784
17:00～18:00	上	3020	474	587	248	468	649	388	432	452	157	205	56	0
	下	0	330	193	194	737	934	1016	606	416	494	278	448	1249
18:00～19:00	上	1966	350	399	204	328	471	289	335	342	122	132	40	0
	下	0	223	129	150	635	787	690	505	304	423	246	320	1010
19:00～20:00	上	939	130	165	88	138	187	124	143	147	48	56	17	0
	下	0	113	59	59	266	306	290	201	147	155	86	154	398
20:00～21:00	上	640	107	126	69	112	153	87	102	94	36	43	13	0
	下	0	75	43	43	186	230	219	146	90	127	70	95	319
21:00～22:22	上	636	110	128	56	105	144	82	95	98	34	40	12	0
	下	0	73	41	42	190	243	192	132	107	123	67	101	290
22:00～23:00	上	294	43	51	24	46	58	35	41	42	15	17	5	0
	下	0	35	20	20	87	108	92	69	47	60	33	49	136

2002 年高教社杯全国大学生数学建模竞赛题

A 题　车灯线光源的优化设计

安装在汽车头部的车灯的形状为一旋转抛物面，车灯的对称轴水平地指向正前方，其开口半径为 36mm，深度为 21.6mm. 经过车灯的焦点，在与对称轴相垂直的水平方向，对称地放置一定长度的均匀分布的线光源. 要求在某一设计规范标准下确定线光源的长度.

该设计规范在简化后可以描述为：在焦点 F 正前方 25m 处的 A 点放置一测试屏，屏与 FA 垂直，用以测试车灯的反射光. 在屏上过 A 点引出一条与地面相平行的直线，在该直线 A 点的同侧取 B 点和 C 点，使 AC = 2AB = 2.6m. 要求 C 点的光强度不小于某一额定值（可取为 1 个单位），B 点的光强度不小于该额定值的两倍（只须考虑一次反射）.

试解决下列问题：

(1) 在满足该设计规范的条件下，计算线光源长度，使线光源的功率最小.

(2) 对得到的线光源长度，在有标尺的坐标系中绘出测试屏上反射光的亮区.

(3) 讨论该设计规范的合理性.

2003 年高教社杯全国大学生数学建模竞赛题

A 题　SARS 的传播

SARS（Severe Acute Respiratory Syndrome，严重急性呼吸道综合症，俗称：非典型肺炎）是 21 世纪第一个在世界范围内传播的传染病. SARS 的爆发和蔓延给我国的经济发展和人民生活带来了很大影响，我们从中得到了许多重要的经验和教训，认识到定量地研究传染病的传播规律，为预测和控制传染病蔓延创造条件的重要性. 试对 SARS 的传播建立数学模型，具体要求如下：

(1) 对附件 1 所提供的一个早期的模型，评价其合理性和实用性.

(2) 建立自己的模型，说明为什么优于附件 1 中的模型；特别要说明怎样才能建立一个真正能够预测以及能为预防和控制提供可靠、足够的信息的模型，这样做的困难在哪里？对于卫生部门所采取的措施做出评论，如：提前或延后 5 天采取严格的隔离措施，对疫情传播所造成的影响做出估计. 附件 2 提供的数据供参考.

(3) 收集 SARS 对经济某个方面影响的数据，建立相应的数学模型并进行预

测. 附件 3 提供的数据供参考.

(4) 给当地报刊写一篇通俗短文,说明建立传染病数学模型的重要性.

附件 1

SARS 疫情分析及对北京疫情走势的预测

2003 年 5 月 8 日

在病例数比较多的地区,用数理模型作分析有一定意义. 前几天,×××老师用解析公式分析了北京 SARS 疫情前期的走势. 在此基础上,我们加入了每个病人可以传染他人的期限 (由于被严格隔离、治愈、死亡等),并考虑在不同阶段社会条件下传染概率的变化,然后先分析香港和广东的情况以获得比较合理的参数,最后初步预测北京的疫情走势. 希望这种分析能对认识疫情,安排后续的工作生活有所帮助.

1. 模型与参数

假定初始时刻的病例数为 N_0,平均每个病人每天可以传染 K 个人 (K 一般为小数),平均每个病人可以直接感染他人的时间为 L 天. 则在 L 天之内,病例数目的增长随时间 t (单位:天) 的关系是

$$N(t) = N_0 (1 + K)^t$$

如果不考虑对传染期的限制,则病例数将按照指数规律增长. 考虑传染期限 L 的作用后,变化将显著偏离指数律,增长速度会放慢. 我们采用半模拟循环计算的办法,把到达 L 天的病例从可以引发直接传染的基数中去掉.

参数 K 和 L 具有比较明显的实际意义. L 可以理解为平均每个病人在被发现前后可以造成直接传染的期限,在此期限后该病人失去传染作用,可能的原因是被严格隔离、病愈不再传染或死去等. 从原理上讲,这个参数主要与医疗机构隔离病人的时机和隔离的严格程度有关,只有医疗机构能有效缩短这个参数. 我们分析广东、香港、北京现有的数据后发现,不论对于疫情的爆发阶段,还是疫情的控制阶段,这个参数都不能用得太小,否则无法描述各阶段的数据. 该参数放在 15 ~ 25 之间比较好,为了简单我们把它固定在 20 (天) 上,这个值有一定统计上的意义,至于有没有医学上的解释,需要其他专家分析.

参数 K 显然代表某种社会环境下一个病人传染他人的平均概率,与全社会的警觉程度、政府和公众采取的各种措施有关. 在疾病初发期,社会来不及防备,此时 K 值比较大. 为了简单起见,我们从开始至高峰期间均采用同样的 K 值 (从拟合这一阶段的数据定出),即假定这个阶段社会的防范程度都比较低,感染率比较高. 到达高峰期后,我们在 10 天的范围内逐步调整 K 值到比较小,然后保持

不变,拟合其后在控制阶段的全部数据,即认为社会在经过短期的剧烈调整之后,进入一个对疫情控制较好的常态. 显然,如果疫情出现失控或反复的状态,则 K 值需要做更多的调整.

2. 计算结果

2.1 对香港疫情的计算和分析. 香港的数据相对比较完整准确. 但在初期,由于诊断标准等不确切,在 3 月 17 日之前,没有找到严格公布的数据. 我们以报道的 2 月 15 日作为发现第一例病人的起点,2 月 27 日从报道推断为 7 例. 3 月 17 日后则都是正式公布的数据. 累积病例数在图 1 中用三角形表示. 我们然后用上述方法计算. 4 月 1 日前后 (从起点起 45 天左右) 是疫情高峰时期,在此之前我们取 K = 0.16204. 此后的 10 天,根据数据的变化将 K 逐步调整到 0.0273,然后保持 0.0273 算出后面控制期的结果. 短期内 K 调整的幅度很大,反映社会的变化比较大. 图 1 中实心方黑点是计算的累积病例数. 从计算累积病例数,很容易算出每天新增病例数 (当然只反映走向,实际状况有很大涨落). 可以看出,香港疫情从起始到高峰大约 45 天,从高峰回落到 1/10 以下 (每天几个病例) 大约 40 天 (5 月上中旬),到基本没有病例还要再经过近一个月 (到 6 月上中旬).

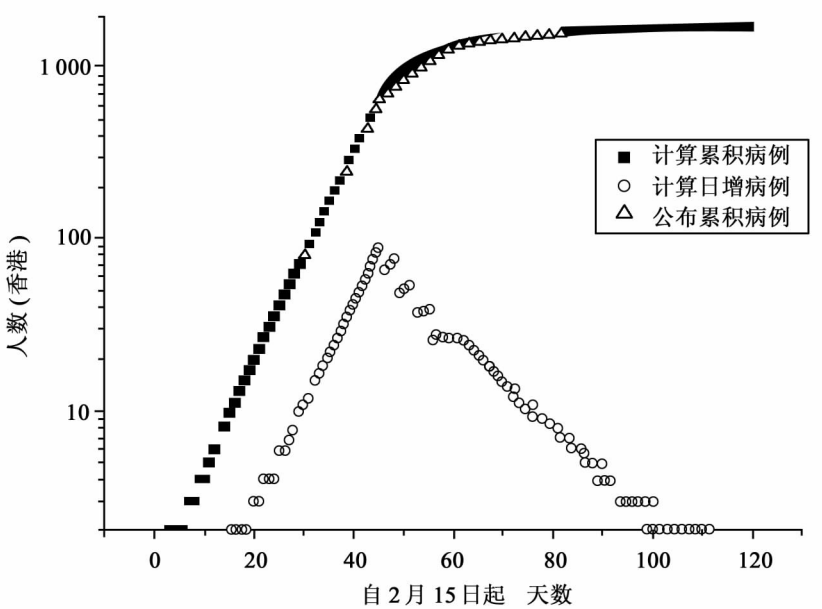


图 1 对香港疫情的拟合

2.2 对广东疫情的计算和分析. 广东的起点是 2002 年 11 月 16 日,到 2003 年 2 月下旬达到高峰,经过了近 100 天. 在 2005 年 2 月 10 日以前的数据查不到,分析比较困难. 总体上看,广东持续的时间比香港长得多,但累积的总病例数却少一些,这反映出广东的爆发和高峰都不强烈. 但广东的回落也比较慢. 从 2 月下旬高峰期到现在经过了约 70 天,还维持着每天十来个新增病例,而同样过程香港只用了约 40 天. 这种缓慢上升和下降的过程也反映到 K 值上. 比较好的拟合结果是,在高峰期之前 ($t < 101$ 天),$K = 0.0892$ 在随后的 10 天逐步调整到 0.031. 用这组参数算出的后期日增病例数比实际公布的偏小,说明实际上降低得更慢. 这种情况与疫情的社会控制状况有没有什么关系,需要更仔细的分析.

2.3 对北京疫情的分析与预测. 北京的病例起点定在 3 月 1 日,经过大约 59 天在 4 月 29 日左右达到高峰. 我们通过拟合起点和 4 月 20 日以后的数据定出高峰期以前的 $K = 0.13913$,这个值比香港的 0.16204 来得低,说明北京初期的爆发程度不如香港,但遗憾的是上升时间持续了近 60 天,而香港是 45 天,这就造成了累积病例数大大超过香港. 从图 2 中还看出 4 月 20 日以前公布的数据

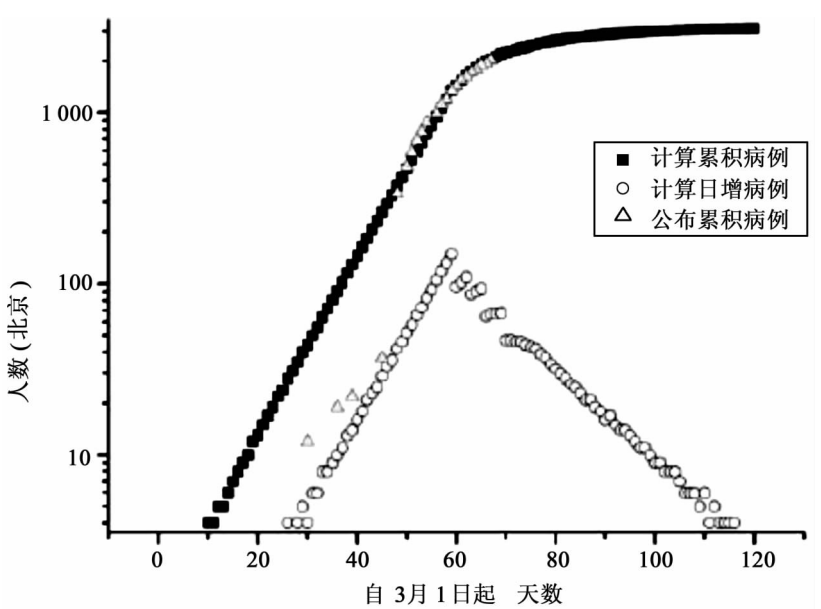


图 2 对北京疫情的分析

大大低于计算值. 而我们从对香港、广东情况的计算中,知道疫情前期我们的计算还是比较可行的. 从而可以大致判断出北京前期实际的病例数. 图 2 中的公布

数据截止到5月 7 日 (从起点起 67 天) ,其后的计算采用的是香港情况下获得的参数. 按这种估算,北京最终累积病例数将达到 3 100 例.

图 3 是计算的日增病例数. 后期下降得较快的实心方黑点是采用香港参数获得的. 这就是说,如果北京的疫情控制与香港相当或更好,就可以在高峰期后的 40 天 (从起点起 100 天) 左右,即 6 月上中旬下降到日增几例. 然后再经过约一个月,即 7 月上中旬达到日增 0 病例. 但如果北京的新病例下降速度与广东类似,则要再多花至少一个月,才能达到上述的效果,且累积总病例数会达到 3 800 左右. 至于什么原因造成香港下降速度快而广东下降速度慢,需要有关方面作具体分析.

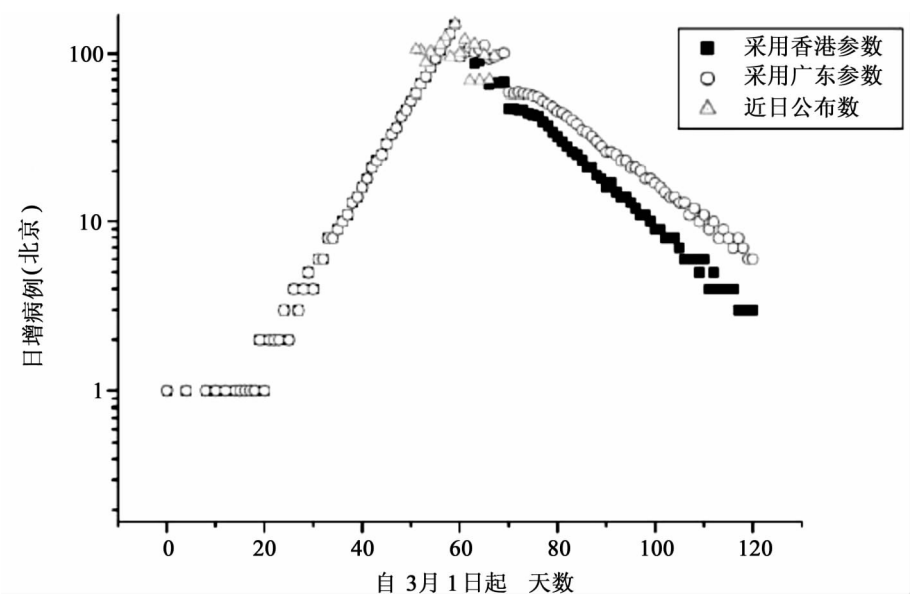


图 3 北京日增病例走势分析

3. 结论

每个病人可以造成直接感染他人的期限平均在 20 天左右,这个值在不同地区和不同疫情阶段似乎变化不大. 病人的平均每天感染率与社会状况有关,在疫情爆发期较大,疫情控制期要小很多. 香港的初期爆发情况比广东和北京都剧烈,但控制效果明显比较好. 北京后期如果控制在香港后期的感染率水平上,则有望在 6 月上中旬下降到日增几例. 然后再经过约一个月,即 7 月上中旬达到日增 0 病例. 而累积总病例数将达到 3 100 例. 但如果北京的新病例下降速度与广

东类似，则要再多花至少一个月，才能达到上述的效果，且累积总病例数会达到3 800左右.

附件2 北京市疫情的数据

（据：http://www.beijing.gov.cn/Resource/Detail.asp?ResourceID=66070）

日 期	已确诊病例累计	现有疑似病例	死亡累计	治愈出院累计
4月20日	339	402	18	33
4月21日	482	610	25	43
4月22日	588	666	28	46
4月23日	693	782	35	55
4月24日	774	863	39	64
4月25日	877	954	42	73
4月26日	988	1093	48	76
4月27日	1 114	1 255	56	78
4月28日	1 199	1 275	59	78
4月29日	1 347	1 358	66	83
4月30日	1 440	1 408	75	90
5月01日	1 553	1 415	82	100
5月02日	1 636	1 468	91	109
5月03日	1 741	1 493	96	115
5月04日	1 803	1 537	100	118
5月05日	1 897	1 510	103	121
5月06日	1 960	1 523	107	134
5月07日	2 049	1 514	110	141
5月08日	2 136	1 486	112	152
5月09日	2 177	1 425	114	168
5月10日	2 227	1 397	116	175
5月11日	2 265	1 411	120	186
5月12日	2 304	1 378	129	208
5月13日	2 347	1 338	134	244

续表

日 期	已确诊病例累计	现有疑似病例	死亡累计	治愈出院累计
5 月 14 日	2 370	1 308	139	252
5 月 15 日	2 388	1 317	140	257
5 月 16 日	2 405	1 265	141	273
5 月 17 日	2 420	1 250	145	307
5 月 18 日	2 434	1 250	147	332
5 月 19 日	2 437	1 249	150	349
5 月 20 日	2 444	1 225	154	395
5 月 21 日	2 444	1 221	156	447
5 月 22 日	2 456	1 205	158	528
5 月 23 日	2 465	1 179	160	582
5 月 24 日	2 490	1 134	163	667
5 月 25 日	2 499	1 105	167	704
5 月 26 日	2 504	1 069	168	747
5 月 27 日	2 512	1 005	172	828
5 月 28 日	2 514	941	175	866
5 月 29 日	2 517	803	176	928
5 月 30 日	2 520	760	177	1 006
5 月 31 日	2 521	747	181	1 087
6 月 01 日	2 522	739	181	1 124
6 月 02 日	2 522	734	181	1 157
6 月 03 日	2 522	724	181	1 189
6 月 04 日	2 522	718	181	1 263
6 月 05 日	2 522	716	181	1 321
6 月 06 日	2 522	713	183	1 403
6 月 07 日	2 523	668	183	1 446
6 月 08 日	2 522	550	184	1 543
6 月 09 日	2 522	451	184	1 653

续表

日 期	已确诊病例累计	现有疑似病例	死亡累计	治愈出院累计
6 月 10 日	2 522	351	186	1 747
6 月 11 日	2 523	257	186	1 821
6 月 12 日	2 523	155	187	1 876
6 月 13 日	2 522	71	187	1 944
6 月 14 日	2 522	4	189	1 994
6 月 15 日	2 522	3	189	2 015
6 月 16 日	2 521	3	190	2 053
6 月 17 日	2 521	5	190	2 120
6 月 18 日	2 521	4	191	2 154
6 月 19 日	2 521	3	191	2 171
6 月 20 日	2 521	3	191	2 189
6 月 21 日	2 521	2	191	2 231
6 月 22 日	2 521	2	191	2 257
6 月 23 日	2 521	2	191	2 277

附件 3 北京市接待海外旅游人数

（单位：万人）

年	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
1997	9.4	11.3	16.8	19.8	20.3	18.8	20.9	24.9	24.7	24.3	19.4	18.6
1998	9.6	11.7	15.8	19.9	19.5	17.8	17.8	23.3	21.4	24.5	20.1	15.9
1999	10.1	12.9	17.7	21.0	21.0	20.4	21.9	25.8	29.3	29.8	23.6	16.5
2000	11.4	26.0	19.6	25.9	27.6	24.3	23.0	27.8	27.3	28.5	32.8	18.5
2001	11.5	26.4	20.4	26.1	28.9	28.0	25.2	30.8	28.7	28.1	22.2	20.7
2002	13.7	29.7	23.1	28.9	29.0	27.4	26.0	32.2	31.4	32.6	29.2	22.9
2003	15.4	17.1	23.5	11.6	1.78	2.61	8.8	16.2				

2003 年高教社杯全国大学生数学建模竞赛题

B 题　露天矿生产的车辆安排

钢铁工业是国家工业的基础之一，铁矿是钢铁工业的主要原料基地. 许多现

代化铁矿是露天开采的,露天开采的生产主要是由电动铲车(以下简称电铲)装车、电动轮自卸卡车(以下简称卡车)运输来完成.提高这些大型设备的利用率是增加露天矿经济效益的首要任务.

露天矿里有若干个爆破生成的石料堆,每堆称为一个铲位,每个铲位已预先根据铁含量将石料分成矿石和岩石.一般来说,平均铁含量不低于25%的为矿石,否则为岩石.每个铲位的矿石、岩石数量以及矿石的平均铁含量(称为品位)都是已知的.每个铲位至多能安置一台电铲,电铲的平均装车时间为5min.

卸货地点(以下简称卸点)有卸矿石的矿石漏、2个铁路倒装场(以下简称倒装场)和卸岩石的岩石漏、岩场等,每个卸点都有各自的产量要求.从保护国家资源的角度及矿山的经济效益考虑,应尽量把矿石按矿石卸点需要的铁含量(假设要求都为29.5%±1%,称为品位限制)搭配起来送到卸点,搭配的量在一个班次(8小时)内满足品位限制即可.从长远看,卸点可以移动,但一个班次内不变.卡车的平均卸车时间为3min.

所用卡车载重量为154t,平均时速28km/h.卡车的耗油量很大,每个班次每台车消耗近1t柴油.发动机点火时需要消耗相当多的电瓶能量,故一个班次中只在开始工作时点火一次.卡车在等待时所耗费的能量也是相当可观的,原则上在安排时不应发生卡车等待的情况.电铲和卸点都不能同时为两辆及两辆以上卡车服务.卡车每次都是满载运输.

每个铲位到每个卸点的道路都是专用的宽60m的双向车道,不会出现堵车现象,每段道路的里程都是已知的.

一个班次的生产计划应该包含以下内容:出动几台电铲,分别在哪些铲位上;出动几辆卡车,分别在哪些路线上,各运输多少次(因为随机因素影响,装卸时间与运输时间都不精确,所以排时计划无效,只求出各条路线上的卡车数及安排即可).一个合格的计划要在卡车不等待条件下满足产量和质量(品位)要求,而一个好的计划还应该考虑下面两条原则之一:

1.总运量(t·km)最小,同时出动最少的卡车,从而运输成本最小;

2.利用现有车辆运输,获得最大的产量(岩石产量优先;在产量相同的情况下,取总运量最小的解).

试就两条原则分别建立数学模型,并给出一个班次生产计划的快速算法.针对下面的实例,给出具体的生产计划、相应的总运量及岩石和矿石产量.

某露天矿有铲位10个,卸点5个,现有铲车7台,卡车20辆.各卸点一个班次的产量要求:矿石漏1.2万t、倒装场Ⅰ1.3万t、倒装场Ⅱ1.3万t、岩石漏1.9万t、岩场1.3万t.

铲位和卸点位置的二维示意图如图1、图2所示,各铲位和各卸点之间的距

离 (km) 如表 1 所示.

表 1

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石漏	5.26	5.19	4.21	4.00	2.95	2.74	2.46	1.90	0.64	1.27
倒装场Ⅰ	1.90	0.99	1.90	1.13	1.27	2.25	1.48	2.04	3.09	3.51
岩　场	5.89	5.61	5.61	4.56	3.51	3.65	2.46	2.46	1.06	0.57
岩石漏	0.64	1.76	1.27	1.83	2.74	2.60	4.21	3.72	5.05	6.10
倒装场Ⅱ	4.42	3.86	3.72	3.16	2.25	2.81	0.78	1.62	1.27	0.50

各铲位矿石、岩石数量 (万 t) 和矿石的平均铁含量如表 2 所示.

表 2

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石量	0.95	1.05	1.00	1.05	1.10	1.25	1.05	1.30	1.35	1.25
岩石量	1.25	1.10	1.35	1.05	1.15	1.35	1.05	1.15	1.35	1.25
铁含量	30%	28%	29%	32%	31%	33%	32%	31%	33%	31%

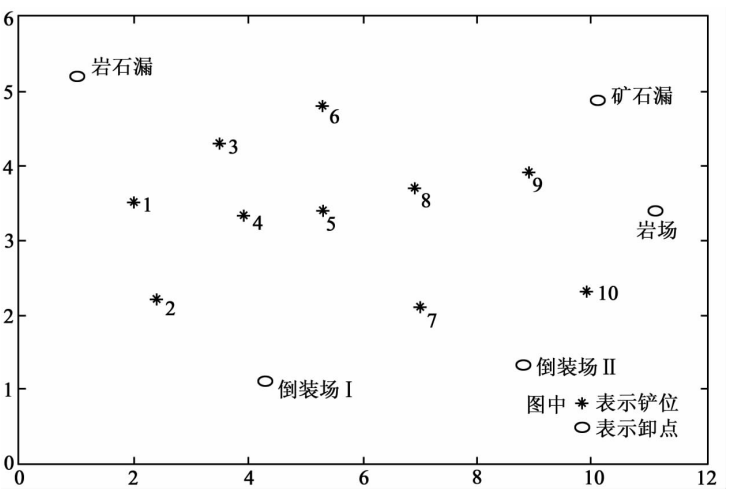


图 1　各个铲位和卸点位置的示意图

(a)电动铲车

(b)电动轮自卸卡车

(c)某露天矿左俯瞰图

(d)某露天矿右俯瞰图

图 2

习 题 答 案

第一章　习题答案

习题 1

1. (1) 10 * x 9 + log (10) * 10 x - log (10) * (1 / log (x) 2) * (1 / x) ;
   (2) - 2 / (2 - x 2) - 4 * x 2 / (2 - x 2) 2 ;
   (3) 2 * exp (2 * x) * (x + y 2 + y) + exp (2 * x) exp (2 * x) * (2 * y + l) ;
   (4) 8 * sin (2 * x + y) 2 - 8 * cos (2 * x + y) 2 2 * sin (2 * x + y) 2 - 2 * cos (2 * x + y) 2 ;
   (5) 2 * x / (x 2 + y 2 + v 2) dx + 2 * y / (x 2 + y 2 + v 2) dy + 2 * v / (x 2 + y 2 + v 2) dz ;
   (6) - 1 / 4 / (x * sin ((3 (exp (x) - log (x))) (1 / 2))) (3 / 2) * (sin ((3 (exp (x) - log (x))) (1 / 2)) + 1 / 2 * x * cos ((3 (exp (x) - log (x))) (1 / 2)) * (3 (exp (x) - log (x))) (1 / 2) * (exp (x) - 1 / x) * log (3)) 2 + 1 / 2 / (x * sin ((3 (exp (x) - log (x))) (1 / 2))) (1 / 2) * (cos ((3 (exp (x) - log (x))) (1 / 2)) * (3 (exp (x) - log (x))) (1 / 2) * (exp (x) - 1 / x) * log (3) - 1 / 4 * x * sin ((3 (exp (x) - log (x))) (1 / 2)) * 3 (exp (x) - log (x)) * (exp (x) - 1 / x) 2 * log (3) 2 + 1 / 4 * x * cos ((3 (exp (x) - log (x))) (1 / 2)) * (3 (exp (x) - log (x))) (1 / 2) * (exp (x) - 1 / x) 2 * log (3) 2 + 1 / 2 * x * cos ((3 (exp (x) - log (x))) (1 / 2)) * (3 (exp (x) - log (x))) (1 / 2) * (exp (x) + 1 / x 2) * log (3)) .
2. (1) 2 / 3 ; (2) 2 ; (3) exp (- 4) ; (4) 1 / 2 ; (5) exp (1) .
3. (1) 1 / 2 * erf (1) * pi (1 / 2) ; (2) 2 + 2 (1 / 2) * atan (2 (1 / 2)) ; (3) 4 / 15 ;
   (4) 1 / 3 * atan (2 * (2 - 3 (1 / 2)) / (4 * 2 (1 / 2) + 6)) + 1 / 3 * atan (2 * (2 - 3 (1 / 2)) / (6 - 4 * 2 (1 / 2))) .
4. (1) - 1 / 2 * x * (2 - x 2) (1 / 2) + asin (1 / 2 * x * 2 (1 / 2)) ;

(2) 1/2 * sin (x) - 1/10 * sin (5 * x) ;

(3) int (1/x/ (log (x) + a + (log (x) + b) (1/2)) (1/2) ,x) ;

(4) [- cos (x) , 1/4 * x 4]

[x * exp (x) - exp (x) , - log (cos (x))] .

5. (1) 4.5746 ; (2) 1.3333 ; (3) 0.7845 ; (4) 3.5000 ; (5) 1.255.

6. (1) 4/3 * log (2) + 8/3 ; (2) - exp (- 1) * pi + pi.

7. (1) C1 + C2 * log (1 + x) ; (2) 3 * x * (x - 2) * t + C1 + C2 * exp (t) ;

(3) - x + 2 ;

(4) 2 - 2 * i * bessely (0 ,4 * i) / (- besseli (0 ,4) * bessely (1 ,2 * i * 2 (1/2)) + i * bessely (0 ,4 * i) * besseli (1 ,2 * 2 (1/2))) * (1 + x) (1/2) * besseli (1 ,2 * 2 (1/2) * (1 + x) (1/2)) + 2 * besseli (0 ,4) / (- besseli (0 ,4) * bessely (1 ,2 * i * 2 (1/2)) + i * bessely (0 ,4 * i) * besseli (1 ,2 * 2 (1/2))) * (1 + x) (1/2) * bessely (1 ,2 * i * 2 (1/2) * (1 + x) (1/2)) .

8. (1) x - 1/6 * x 3 + 1/120 * x 5 ; (2) 1 - 1/6 * x 2 + 1/120 * x 4 ;

(3) 1 + 2 * x + 2 * x 2 + 4/3 * x 3 + 2/3 * x 4 + 4/15 * x 5 ;

(4) x + x 2 + 1/2 * x 3 + 1/6 * x 4 + 1/24 * x 5.

9. [- 1/2 * b + 1/2 * (b 2 - 4 * c) (1/2)]

[- 1/2 * b - 1/2 * (b 2 - 4 * c) (1/2)] .

10. - 1.1218 + 0.3185i

- 1.1218 - 0.3185i

- 0.5000 + 0.8660i

- 0.5000 - 0.8660i

0.2870 + 0.8929i

0.2870 - 0.8929i

0.8348 + 0.3729i

0.8348 - 0.3729i

11. [- 1.40824845843322579201684754629 25]

[- .46586916707607665574895492689981 - 1.19413222394156806940239313002 00 * i]

[- .46586916707607665574895492689981 + 1.19413222394156806940239313002 00 * i]

[.60804677070454177779568381451797 1 - .88541124247455434577495883624571 * i]

[.60804677070454177779568381451797 1 + .88541124247455434577495883624571 * i]

[1.12389325117629554760108110973 27] .

12. 0.5124.

13. (1) x = [- 3/10 * 10 (1/2)]

[3/10 * 10 (1/2)]

y = [1/10 * 10 (1/2)]

[- 1/10 * 10 (1/2)]；

(2) (0.6417,0.8011)；

(3) (0.5991,2.3959,2.0050).

14. 1.4938 3.0325

0.0686 1.1875 3.3411.

15. 略.

16. Method = nearest ans = 2

method = linear ans = 1.8833

method = spline ans = 1.8461

method = cubic ans = 1.8844.

17. 略.

习题 2

1. (1) 3； (2) 3.

2. (1) [4 - 1 - 1

0 1 - 2

1 0 - 1]；

(2) [1.0000 0 - 1.0000 0

- 3.0000 1.0000 3.0000 - 1.0000

6.0000 - 2.0000 - 5.0000 2.0000

- 24.0000 7.0000 20.0000 - 6.0000].

3. (1) 4 个向量相关,最大无关组为 3；

(2) 5 个向量相关,最大无关组为 2；

(3) 3 个向量线性无关.

4. (1) 无解；

(2) 有惟一零解；

(3) 无解；

(4) x1 = x2 = k, x3 = x4 = 0,其中 k 表示任意实数.

5. (1) P = 0 0.3015 0.3015

0 0.3015 0.3015

1.0000 - 0.9045 - 0.9045

$$D = \begin{matrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

该矩阵不能化为对角矩阵.

(2) $P = \begin{matrix} -0.7071 & -0.2425 & 0.3015 \\ 0 & 0 & 0.9045 \\ -0.7071 & -0.9701 & 0.3015 \end{matrix}$

$$D = \begin{matrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{matrix}$$

该矩阵可以对角化.

(3) $P = \begin{matrix} -0.6667 & -0.0276 & -0.7448 \\ 0.6667 & -0.4690 & -0.5793 \\ -0.3333 & -0.8828 & 0.3310 \end{matrix}$

$$D = \begin{matrix} -0.0000 & 0 & 0 \\ 0 & 9.0000 & 0 \\ 0 & 0 & 9.0000. \end{matrix}$$

6. (1) $f = -y_1^2 + 2y_2^2 + 5y_3^2$;

(2) $f = -1.4142y_1^2 + 1.4142y_3^2$;

(3) $f = 4y_4^2$

7. $\frac{15}{19}, \frac{165}{19}, \frac{30}{19}$.

8. 一年以后农村居住人口比例为:76.2%,城镇居住人口比例为:23.8%,十年以后农村居住人口比例为:50.78%,城镇居住人口比例为:49.22%,n 年以后农村居住人口比例为 x_n, y_n,其中 $\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} 0.95 & 0.01 \\ 0.05 & 0.99 \end{pmatrix}^n \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$.

第二章 习题答案

习题 3

1. y = asin (sin (x) - x * cos (x) + C1) .

2. x = - 48 / (- 1 + 13 (1 /2)) / (1 + 13 (1 /2)) - 6 /13 * (- 13 * C1 * exp (1 /2 * (1 + 13 * (1 /2)) * t) +

C1 * 13 (1/2) * exp(-1/2 * (-1 + 13 (1/2)) * t) - C1 * 13 (1/2) * exp (1/2 * (1 + 13 (1/2)) * t) - 13 * C1 * exp(-1/2 * (-1 + 13 (1/2)) * t) + 6 * C2 * 13 (1/2) * exp(-1/2 * (-1 + 13 (1/2)) * t) - 6 * C2 * 13 (1/2) * exp (1/2 * (1 + 13 (1/2)) * t)) / (-1 + 13 (1/2)) / (1 + 13 (1/2));

y = 16 / (-1 + 13 (1/2)) / (1 + 13 (1/2)) - 6/13 * (2 * C1 * 13 (1/2) * exp (-1/2 * (-1 + 13 (1/2)) * t) - 2 * C1 * 13 (1/2) * exp (1/2 * (1 + 13 (1/2)) * t) - 13 * C2 * exp (1/2 * (1 + 13 (1/2)) * t) - C2 * 13 (1/2) * exp(-1/2 * (-1 + 13 (1/2)) * t) + C2 * 13 (1/2) * exp (1/2 * (1 + 13 (1/2)) * t) - 13 * C2 * exp(-1/2 * (-1 + 13 (1/2)) * t)) / (-1 + 13 (1/2)) / (1 + 13 (1/2)).

3. (1) -x + 2;

(2) 2 - 2 * bessely (0,2 * i * 2 (1/2)) / (i * besseli (0,2 * 2 (1/2)) * bessely (1,2 * i * 2 (1/2)) + bessely (0,2 * i * 2 (1/2)) * besseli (1,2 * 2 (1/2))) * (1 + x) (1/2) * besseli (1,2 * 2 (1/2) * (1 + x) (1/2)) - 2 * i * besseli (0,2 * 2 (1/2)) / (i * besseli (0,2 * 2 (1/2)) * bessely (1,2 * i * 2 (1/2)) + bessely (0,2 * i * 2 (1/2)) * besseli (1,2 * 2 (1/2))) * (1 + x) (1/2) * bessely (1,2 * i * 2 (1/2) * (1 + x) (1/2))

4. (1) x_1 = 3/2 * C1 * exp (3 * t) - 1/2 * C1 * exp (5 * t) + 1/2 * C2 * exp (5 * t) - 1/2 * C2 * exp (3 * t)

x_2 = -3/2 * C1 * exp (5 * t) + 3/2 * C1 * exp (3 * t) - 1/2 * C2 * exp (3 * t) + 3/2 * C2 * exp (5 * t);

x_1 = 1/3 * C1 * exp (t) + 2/3 * C1 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C2 * 3 (1/2) * exp(-1/2 * t) * sin (1/2 * t * 3 (1/2)) - 1/3 * C2 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C2 * exp (t) - 1/3 * C3 * 3 (1/2) * exp(-1/2 * t) * sin (1/2 * t * 3 (1/2)) - 1/3 * C3 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C3 * exp (t)

x_2 = -1/3 * C1 * 3 (1/2) * exp(-1/2 * t) * sin (1/2 * t * 3 (1/2)) - 1/3 * C1 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C1 * exp (t) + 1/3 * C2 * exp (t) + 2/3 * C2 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C3 * 3 (1/2) * exp(-1/2 * t) * sin (1/2 * t * 3 (1/2)) - 1/3 * C3 * exp (-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C3 * exp (t)

x_3 = 1/3 * C1 * 3 (1/2) * exp(-1/2 * t) * sin (1/2 * t * 3 (1/2)) - 1/3 * C1 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C1 * exp (t) - 1/3 * C2 * 3 (1/2) * exp(-1/2 * t) * sin (1/2 * t * 3 (1/2)) - 1/3 * C2 * exp(-1/2 * t) * cos (1/2 * t * 3 (1/2)) + 1/3 * C2 * exp (t) + 1/3 * C3 * exp (t) + 2/3 * C3

$*\exp(-1/2*t)*\cos(1/2*t*3^{(1/2)})$.

5. $y=\tan(x+C1)$.

6. $y=2^{(1/2)}*pi^{(1/2)}/x^{(1/2)}*\sin(x)$.

7.

x = 1.5708	y = 2.0000	-0.6366
1.6074	1.9758	-0.6869
1.7645	1.8518	-0.8879
1.9215	1.6982	-1.0631
2.0786	1.5192	-1.2108
2.2357	1.3193	-1.3293
2.3928	1.1032	-1.4174
2.5499	0.8756	-1.4744
2.7069	0.6416	-1.5002
2.8640	0.4060	-1.4951
3.0211	0.1735	-1.4602
3.1416	0.0002	-1.4140.

8. $2000000/(100+t)^4$.

9. **略**.

10. **通解为** $y=\tan(x+C1)$.

特解为 $y=\tan(x)$.

11. **略**.

12. $x=\exp(t)*(-1/2*\cos(t)+3/2*\sin(t))$

$y=\exp(t)*(1/2*\sin(t)+3/2*\cos(t))$.

13.

t = 0.1000	x = -0.3843
0.2000	-0.2345
0.3000	-0.0464
0.4000	0.1844
0.5000	0.4622
0.6000	0.7913
0.7000	1.1758
0.8000	1.6195
0.9000	2.1256
1.0000	2.6967.

14. $y=\exp(x*\exp(-C1))/x$.

15. $v=-t^2+6$.

16. $y=C1*\exp(a*t)+C2*\exp(b*t)$.

17. y = - 2 * cos (x) - x * sin (x) + 2 * x + 5 /2.

第三章　习题答案

习题 4

1. 16. 4233 (提示:用一元回归分析求解).
2. 1. 1723.
3. 不同行业的职工工资水平有显著差异;不同地区、不同行业的职工工资水平没有显著差异(提示:利用方差分析求解).
4. 方差分析结果 p = 0. 5657　0. 0442　0. 5684.
   所以,浓度对试验指标没有显著影响;温度对试验指标有显著影响;交互作用没有显著影响. 最佳水平组合状态为:温度 51　,浓度 4. 7%.
5. 略.
6. 略.
7. 吸烟习惯与慢性气管炎有关.
8. (提示:利用多元线性回归模型来分析)回归模型为

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_4 x_4$$

   地区的销售量、同类商品竞争数和地区销售潜力分别是影响建筑材料销售量的主要因素.

第四章　习题答案

习题 5

1. x = 1. 0e + 015 *
   - 7. 0711
   - 0. 0000
   7. 0711
2. x = 1
   5
   1
   1
   1
   1.

3. a = 1，b = 1 时

f-COUNT	FUNCTION	STEP-SIZE	GRAD/SD
4	2	0.5	- 8
9	1.49645e - 017	0.5	- 1.81e - 008

NUMBER OF FUNCTION EVALUATIONS = 9

x = 1.0e - 008 *

- 0.2735 - 0.2735.

4. $x_1 = 0.5000$，$x_2 = -1.0000$.

5. $x_1 = -0.0044$，$x_2 = 0.0004$，$x_3 = -0.0017$，$x_4 = -0.0017$.

6. $x_1 = 0$，$x_2 = -1$.

7. $x_1 = 1.2456$，$x_2 = 1.2456$，$x_3 = 2.3202$，$x_4 = 2.3202$，$x_5 = 2.3202$.

8. $x_1 = 4$，$x_2 = 0$，$x_3 = 0$.

9. 生产甲产品 10 件，乙产品 5 件可以使收益达到最大.

10. 生产宋师面包 8 000 个，唐韵面包 3 000 个，可以使利润达到最大.

11. (1) $R_1 = 1/2$，$R_2 = 1/3$，$R_3 = 1/4$，$R_4 = 1/9$；

(2) $R_1 = 3$，$R_2 = 3$，$R_3 = 3$，$R_4 = 2/3$.

第五章 习题答案

习题 6

1. 略.

2. 把电缆看做是过点 (- 50，10)，(0，0) 和 (0，0)，(50，10) 的两条线段组成的折线，则电缆长度 $L_1 = 101.98$；若把电缆近似看做是过点 (- 50，10)，(0，0)，(50，10) 的抛物线，则电缆长度 $L_2 = 102.606$.

3. 1745.388m.

4. 长 7m 的梯子可以达到要求，能满足要求的梯子的最短长度为 3.61m.

第六章 习题答案

习题 7

1. 当 v = 60，v 改变 1 时，p 的变化量为 2.3721；

当 v = 50，v 改变 1 时，p 的变化量为 2.7667；

当 v 从 70 减至 40 时气体所做的功为 181610.

2. (1) 用插值法计算,当 method = ‘spline’时:

当 E = 1.4, 2.5, 5.7 (MeV) 时, P = 4.0363　10.3899　42.4628

D = 2.7467　6.5113　25.2729;

(2) 当 P = 8.21 时, E 的估算值为 2.1701,

当 D = 10.42 时, E 的估算值为 3.3177,

当 P = 12.45 时, D 的估算值为 7.7684.

3. K = 0.0766.

4. $\alpha = 9.5$, $\beta = 0.73$, $\gamma = 0.42$.

5. 略.

6. 当 u = 5 (km/h), $v_0 = 10^8$ (m^3), 即租用大型船时费用 S 最低,约为 0.055 英镑.

第七章　习题答案

习题 8

1. t = 0	x = 25.0000	2.0000
0.1000	27.0818	2.0041
0.2000	29.3344	2.0170
0.3000	31.7689	2.0394
0.4000	34.3961	2.0726
0.5000	37.2258	2.1178
0.6000	40.2673	2.1767
0.7000	⋮	⋮
⋮	20.0105	28.2919
14.5000	16.6759	27.9181
14.6000	13.9650	27.3821
14.7000	11.7742	26.7239
14.8000	9.9982	25.9788
14.9000	8.5540	25.1736
15.0000	7.3762	24.3300
	6.4158	23.4645

函数图形如图 1,图 2 所示.

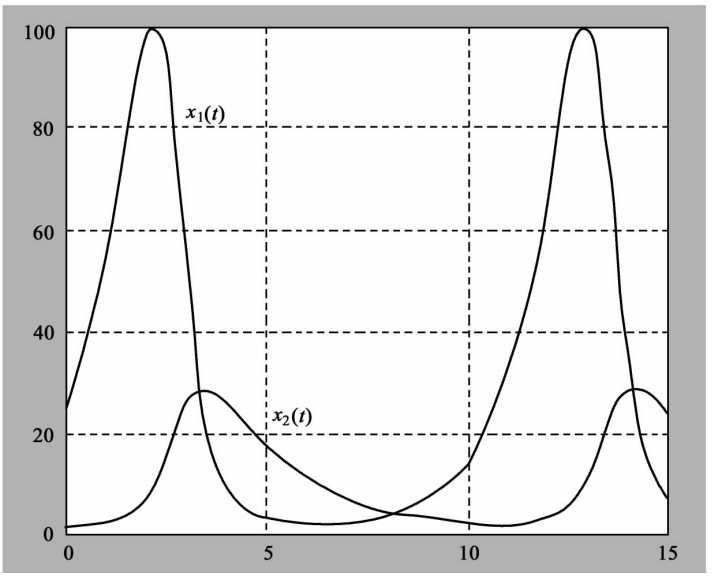


图 1　食饵—捕食者模型的数值解 x_1 (t) ,x_2 (t)

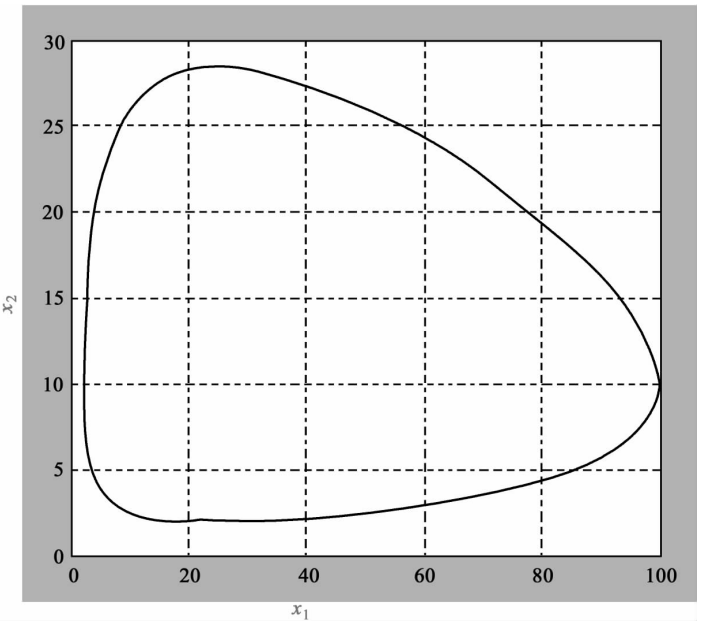


图 2　食饵—捕食者模型的相图 (x_1 ,x_2)

2. t = 0

	x = 1.0000	0
0.0001	1.0000	- 0.0001
0.0005	1.0000	- 0.0005
0.0025	1.0000	- 0.0025
0.0125	0.9999	- 0.0125
0.0625	0.9980	- 0.0624
0.1880	0.9824	- 0.1872
0.4481	0.9003	- 0.4428
0.6980	0.7584	- 0.6955
0.9204	0.5767	- 0.9452
1.1531	0.3221	- 1.2531
1.3867	- 0.0119	- 1.6141
1.5258	- 0.2518	- 1.8307
1.6649	- 0.5195	- 2.0035
1.7918	- 0.7791	- 2.0697
1.9190	- 1.0395	- 1.9970
2.0559	- 1.2972	- 1.7402
2.1578	- 1.4603	- 1.4520
2.2597	- 1.5915	- 1.1216
2.3563	- 1.6845	- 0.8067
2.4549	- 1.7492	- 0.5111
2.5658	- 1.7896	- 0.2284
2.6689	- 1.8019	- 0.0160
2.7497	- 1.7975	0.1198
2.8304	- 1.7832	0.2328
2.9214	- 1.7571	0.3382
3.0000	- 1.7274	0.4148

3. tt = 0	x (t) = 0.5000	y (t) = 0.5000
0.0800	0.5395	0.4873
0.2800	0.6319	0.4305
0.4800	0.7094	0.3385
0.6800	0.7653	0.2160
0.8736	0.7938	0.0759
1.0412	0.7954	- 0.0571
1.1754	0.7803	- 0.1687
1.3754	0.7294	- 0.3422
1.5754	0.6428	- 0.5258
1.7754	0.5180	- 0.7251
1.9754	0.3512	- 0.9477
2.0000	0.3275	- 0.9770

4. (1) 3.73s；(2) 2.54s.

5. 鱼雷追击敌舰的曲线 $y = y(x)$ 满足微分方程模型 $\begin{cases} \dfrac{dp}{dx} = \dfrac{\sqrt{1+p^2}}{2(1-x)} \\ p(0) = 0 \end{cases}$，其中 $y' = p$，$y'' = \dfrac{dp}{dx}$，$0 < x < 1$，解得 $y = \dfrac{1}{3}(1-x)^{3/2} - (1-x)^{1/2} + \dfrac{2}{3}$，当鱼雷击中敌舰时，$x = 1$，此时 $y = 2/3$. 即敌舰行至 2/3 海里处，被击中（提示：可在 Matlab 中直接求解微分方程，解为 $y = 0.667$）.

第八章　习题答案

习题 9

1. 及时接到的概率为 0.92.
2. $p = 0.0019$，差异非常显著. 两两比较知 D 与 A，C，E 有显著差异，与 B 无显著差异，而 E 与 B 无显著差异.
3. 工人之间有显著差异，机器之间无显著差异，二者的交互作用显著.
4. 25 条.
5. 平均队长是 12，9，平均修理时间 2.3，3.7.
6. 品种有显著影响.
7. 每周收入 y 与电视广告费 x_1 和报纸广告费 x_2 的线性模型为

$$y = 83.2116 + 1.2985x_1 + 2.3372x_2$$

有一个异常点,去掉后

$$y = 83.4881 + 1.2877x_1 + 2.9766x_2$$

剩余标准差更小.

8. 回归模型为 $y = 192.5663 + 1.7916x$,硬度为 65 时的抗张强度为 309.021,β_0 的置信区间为 [66.5661 318.5666],β_1 的置信区间为 [-0.0530 3.6362].

9. 拉伸倍数无显著影响,收缩率及二者的交互作用有显著影响;使纤维弹性达到最大的生产条件为 (A_2, B_2).

第九章 习题答案

习题 10

1. 15 年后,0~5 岁的有 14 375 头;6~10 岁的有 1 375 头;11~15 岁的有 875 头.

2. 根据齐次线性方程组基础解系的理论,齐次方程组的通解可以表示为

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = k\begin{pmatrix} \frac{31}{36} \\ \frac{8}{9} \\ 1 \end{pmatrix}$$

其中 k 为任意实数. 由于每个工人的日工资在 60~80 元之间,故选择 $k = 72$,以确定木工、电工及油漆工每人每天的日工资:$x_1 = 62$ 元,$x_2 = 64$ 元,$x_3 = 72$ 元.

即 木工每日工资为:62 元;

电工每日工资为:64 元;

油漆工每日工资为:72 元.

3. 提示:列出方程组,在 Matlab 中直接求解即可.

4. (1) 若共有 n 个部门,记一定时期内第 i 个部门的总产出为 x_i,其中对第 j 个部门的投入为 x_{ij},满足的外部需求为 d_i,则 $x_i = \sum_{j=1}^{n} x_{ij} + d_i$. 记第 j 个部门的单位产出需要第 i 个部门的投入为 a_{ij},则 $a_{ij} = \frac{x_{ij}}{x_j}$,所有 $x_i = \sum_{j=1}^{n} a_{ij}x_j + d_i$. 记投入系数矩阵 $A = (a_{ij})$,产出向量 $x = (x_1, \cdots, x_n)^T$,需求向量 $d = (d_1, \cdots, d_n)^T$,则有 $x = Ax + d$,所以各部门总产出的模型为 $x = (I - A)^{-1}d$.

(2) 三个部门的总产量分别应为 139.2801,267.6056,208.1377.

(3) 当对农业的需求增加 1 个单位时,农业、制造业和服务业的总产出应分别增加 1.3459,0.5634,0.4382 个单位.

(4) 只要 $\|A\|_1 < 1$,即 $\sum_{i=1}^{n} a_{ij} < 1$,模型就是可行的.

第十章　习题答案

习题 11

1. 管道设计的最小生成树矩阵为

$$T = \begin{matrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 2 & 1 & 2 & 9 \end{matrix}.$$

2. 任意两城市之间的最短距离矩阵 D 和最短路径矩阵 path 分别为

$$D = \begin{matrix} 0 & 35 & 105 & 95 & 85 & 10 \\ 70 & 0 & 15 & 20 & 30 & 60 \\ 85 & 15 & 0 & 10 & 20 & 75 \\ 60 & 20 & 10 & 0 & 10 & 50 \\ 45 & 35 & 20 & 10 & 0 & 35 \\ 10 & 25 & 80 & 70 & 60 & 0 \end{matrix}$$

$$path = \begin{matrix} 1 & 6 & 2 & 2 & 2 & 6 \\ 1 & 2 & 3 & 4 & 4 & 1 \\ 2 & 2 & 3 & 4 & 5 & 2 \\ 1 & 2 & 3 & 4 & 5 & 1 \\ 1 & 3 & 3 & 4 & 5 & 1 \\ 1 & 2 & 2 & 2 & 2 & 6 \end{matrix}$$

3. 连接方式为:1—4—5—2—3.

第十一章　习题答案

习题 12

1. 设工厂 i 运送到用户 j 的产品数量为 x_{ij},令

$$A = \begin{bmatrix} 5 & 2 & 6 & 7 \\ 3 & 5 & 4 & 10 \\ 4 & 3 & 2 & 13 \\ 20 & 10 & 45 & 25 \end{bmatrix}$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix}.$$

则目标规划模型为

$$\min \quad AX$$

s. t.

$$x_{14} + x_{24} + x_{34} \geqslant 25$$
$$x_{11} + x_{21} + x_{31} \geqslant 16$$
$$x_{12} + x_{22} + x_{32} \geqslant 8$$
$$x_{13} + x_{23} + x_{33} \geqslant 36$$
$$x_{31} \geqslant 15$$
$$x_{24} < x_{14}, x_{24} < x_{34}.$$

2. **当** $t = 128.4$ **时,出售最佳**.
3. **甲的产量为** 23.9025**,乙的产量为** 62.4977 **时,利润最大,最大利润为** 6413.5.
4. **设甲为** x **百箱,乙为** y **百箱,则优化模型为**

$$\max \quad 10x + 9y$$

s. t.

$$6x + 5y \leqslant 60$$
$$10x + 29y \leqslant 150$$
$$x \leqslant 8$$
$$x, y \geqslant 0.$$

当甲、乙两种口味的饮料各生产 6.4286 **和** 4.2857 **(百箱)时,获利最大为** 102.857 1 **万元**.

5. **加工原油** A,B,C **的数量分别为** 4253.15,5798.75,3948.25 **时,利润最大**.
6. **第一年项目** A,D **分别投资** 3.826 8 **万元和** 6.1732 **万元;**
   第二年项目 A,C **分别投资** 3.543 6 **万元和** 3 **万元;**
   第三年项目 A,B **分别投资** 0.400 8 **万元和** 4 **万元;**
   第四年项目 A **投资** 4.075 2 **万元;**
   第五年项目 D **投资** 0.460 9 **万元**.
   5 **年后总资金** 14.375 **万元,盈利** 43.75%.

参 考 文 献

[1] 电子科技大学应用数学系. 微积分(上、下册). 成都:电子科技大学出版社,2000

[2] 华东师范大学数学系. 微积分(上、下册). 北京:高等教育出版社,1999

[3] 华东师范大学数学系. 线性代数. 北京:高等教育出版社,1999

[4] 电子科技大学应用数学系. 线性代数. 成都:电子科技大学出版社,2000

[5] 刘光祖. 概率论与数理统计. 北京:高等教育出版社,2000

[6] 应玖茜等. 非线性规划及其理论. 北京:中国人民大学出版社,1998

[7] 钱颂迪. 运筹学. 北京:清华大学出版社,1990

[8] 马振华等编. 现代应用数学手册—运筹学与最优化理论卷. 北京:清华大学出版社,2002

[9] 魏权龄等. 数学规划议论. 北京:北京航空航天大学出版社,2001

[10] 席少霖等. 最优化计算方法. 上海:上海科学技术出版社,2003

[11] 管梅谷等. 线性规划. 山东:山东师范大学出版社,1996

[12] 奚梅成. 数值分析方法. 合肥:中国科学技术大学出版社,1994

[13] 金福临等. 应用常微分方程. 上海:复旦大学出版社,2001

[14] 卢开澄等. 图论及其应用. 北京:清华大学出版社,1996

[15] 刘次华等. 概率论与数理统计. 北京:高等教育出版社,1999

[16] 张志涌等. 掌握与精通 MATLAB. 北京:北京航空航天大学出版社,2004

[17] 张培强. MATLAB 语言. 合肥:中国科学技术大学出版社,2003

[18] 许波. MATLAB 工程数学应用. 北京:清华大学出版社,2003

[19] 李人厚等. 精通 MATLAB 综合辅导与指南. 西安:西安交通大学出版社,2002

[20] 王兵团等. MATLAB 与数学实验. 北京:中国铁道出版社,2002

[21] 导向科技遍著. MATLAB 程序设计与实例应用. 北京:中国铁道出版社,2003

[22] 张国权主编. 数学实验. 北京:科学出版社,2004

[23] 周义仓等. 数学建模实验. 西安:西安交通大学出版社,2001

[24] 宋世德等. 数学实验. 北京:高等教育出版社,2003

[25] 箫树铁主编. 数学实验. 北京:高等教育出版社,2000

[26] 电子科技大学应用数学系. 数学实验简明教程. 成都:电子科技大学出版社,2002

[27] 赵静等. 数学建模与数学实验. 北京:高等教育出版社,2002