華南理工大學

# 实 验 报 告

课程名称：数学实验

姓名：

序号：

班级：

日 期：

**数学学院**

# 目 录

实 验 一　　试利用 Matlab 的符号运算功能解决以下问题......1
实 验 二　　Matlab 编程与作图......5
实 验 三　　代数模型实验......10
实 验 四　　代数变换模型......15
实 验 五　　代数方程模型实验......22
实 验 六　　分形实验......35
实 验 七　　插值实验......46
实 验 八　　微分方程实验......53
实 验 九　　蒙特卡洛模拟实验......58

# 实验一　试利用 Matlab 的符号运算功能解决以下问题

问题 1 设 n=10

$$X=\begin{bmatrix} 0 & a_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & a_2 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & a_{n-1} \\ a_n & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

其中$a_i \neq 0$ ，$i=1,2,\cdots,n$ ，求$X^{-1}$

实验过程：

```
n=10;
% 建立符号矩阵
syms A;
for i=1:n
  A(i)=sym (['a',num2str(i)]);
end
% 建立题目中的矩阵
for i=1:n-1
  X(i,i+1)=A(i);
end
X(n,1)=A(n);
% 对 X 矩阵求逆
result=X^-1;
```

实验结果及分析：

```
result =

[    0,    0,    0,    0,    0,    0,    0,    0,    0, 1/a10]
[ 1/a1,    0,    0,    0,    0,    0,    0,    0,    0,     0]
[    0, 1/a2,    0,    0,    0,    0,    0,    0,    0,     0]
[    0,    0, 1/a3,    0,    0,    0,    0,    0,    0,     0]
[    0,    0,    0, 1/a4,    0,    0,    0,    0,    0,     0]
[    0,    0,    0,    0, 1/a5,    0,    0,    0,    0,     0]
[    0,    0,    0,    0,    0, 1/a6,    0,    0,    0,     0]
[    0,    0,    0,    0,    0,    0, 1/a7,    0,    0,     0]
[    0,    0,    0,    0,    0,    0,    0, 1/a8,    0,     0]
[    0,    0,    0,    0,    0,    0,    0,    0, 1/a9,     0]
```

问题 2 设 m=3,n=4，A，B 分别为$n \times m$ 和$m \times n$ 矩阵。证明

$$\begin{vmatrix} E_m & B \\ A & E_n \end{vmatrix} = |E_n - AB| = |E_m - BA|$$

其中$E_m$ ，$E_n$分别为 m 和 n 阶单位矩阵。

实验过程：

```
m=3;
n=4;
% 建立题目中的矩阵
syms A B; % A=nxm
for i=1:m
  for j=1:n
  A(j,i)=sym (['a',num2str(j),num2str(i)]);
  B(i,j)=sym (['b',num2str(i),num2str(j)]);
  end
end
Em=eye(m);
En=eye(n);
%三个式子依次为
X1=det([Em B;A En]);
X2=det([En-A*B]);
X3=det([Em-B*A]);
% 作差
result=[X2-X1 X3-X2]
```

实验结果及分析：如图所示

```
result =

    0
```

等式左边右边作差，差为零，所以左边右边相等，原式得证。

问题 3 设 n=3，A,B,C,D 都是$n\times n$矩阵，且$|A|\neq 0$ ，$AC=CA$ 证明

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |AD-CB|$$

题目分析：因为$|A|\neq 0$，所以 A 为可逆矩阵，又因为$AC=CA$，所以有$C=ACA^{-1}$

将 CB 换成$ACA^{-1}B$。

实验过程：

```
clear all
n=3;
syms A B C D;
for i=1:n
  for j=1:n
    A(i,j)=sym (['a',num2str(i),num2str(j)]);
    B(i,j)=sym (['b',num2str(i),num2str(j)]);
    C(i,j)=sym (['c',num2str(i),num2str(j)]);
    D(i,j)=sym (['d',num2str(i),num2str(j)]);
  end
end
res=det([A B;C D])-det([A*D-A*C*inv(A)*B]);
```

实验结果及分析：如图所示

```
res =

0
```

等式左边右边作差，差为零，所以左边右边相等，原式得证。

问题 4 设 n=5,A 是一个$n\times n$矩阵，rank（A）=1，证明

$$1）A=\begin{pmatrix}a_1\\a_2\\\vdots\\a_n\end{pmatrix}(b_1\quad b_2\quad\cdots\quad b_n)$$

$$2）A^2=kA$$

并确定 k 的值。

实验过程：

```
n=5;
% 第一小问
syms ax ay A;
for j=1:n
  ax(1,j)=sym (['a',num2str(j)]);
  ay(j,1)=sym (['b',num2str(j)]);
end
A=ax*ay;
ra=rank(A);
% 第二小问，建立变量 k，通过 solve 求解
syms k;
res=solve(A*A-k*A==0,k);
```

实验结果及分析：

```
ra =

    1

>> res

res =

a1*b1 + a2*b2 + a3*b3 + a4*b4 + a5*b5
```

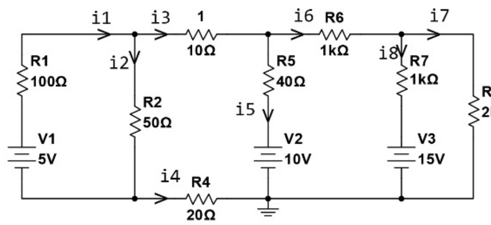
由（1）可知 k 为两个向量的内积

注：每次课的课外作业作为一次实验（可包括多个问题），做完一次实验除按时上交给课代表外，自己保存。下次作业在前一次基础上接着进行（若前次为实验一，这次则为实验二，…）

# 实验二　Matlab 编程与作图

1、自选本专业领域中的 2 个计算问题，用 Matlab 编写函数 M 文件进行求解；

Q1.1 求解图中各支路电流




解：由支路电流法，列出 KCL KVL 则有：

$$i_1 = i_2 + i_3$$

$$i_2 = i_1 + i_4$$

$$i_3 = i_6 + i_5$$

$$i_6 = i_1 + i_8$$

$$50i_2 + 100i_1 = 5$$

$$40i_5 - 20i_4 - 50i_2 + 100i_3 = -10$$

$$1000i_8 + 1000i_6 - 40i_5 = -5$$

$$2000i_7 - 1000i_8 = 15$$

用 Matlab 列出方程求解。

```
syms i;
for index = 1:8
    i(index) = sym(['i', num2str(index)]);
end
syms eq
eq(1)=i(1)==i(2)+i(3);
eq(2)=i(2)==i(1)+i(4);
eq(3)=i(3)==i(6)+i(5);
eq(4)=i(6)==i(8)+i(7);
eq(5)=50*i(2)+100*i(1)==5;
eq(6)=-50*i(2)+10*i(3)-20*i(4)+40*i(5)==-10;
eq(7)=1000*i(8)+1000*i(6)-40*i(5)==-5;
eq(8)=2000*i(7)+1000*i(8)==15;
res=solve(eq,i);
```

Q1.2 绘制互感耦合的频率响应曲线。

由《高频》相关知识可知其关系式为

$$\alpha = \frac{2\eta}{d\overline{(1+\eta^2)^2+2(1-\eta^2)\xi^2+\xi^4}}$$

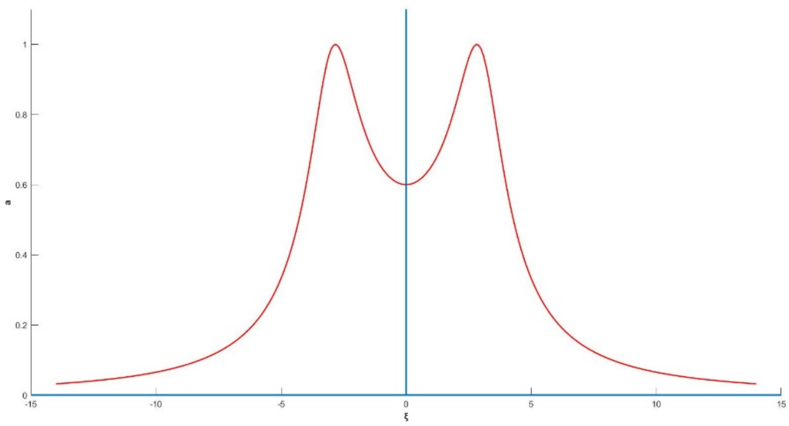
取$\eta = 3$，则

$$\alpha = \frac{6}{\sqrt{100-16\xi^2+\xi^{\varphi}}}$$

编写 matlab 程序，即可画出函数曲线

```
s=-14:0.1:14
a=6./sqrt(100-16.*s.^2+s.^4)
plot(s,a,'r','LineWidth', 1.5)
axis([-15 15 0 1.1]);
line([0 0],[0 1.1],'LineWidth', 2);
line([-15 15],[0 0],'LineWidth',2);
xlabel('ξ','FontWeight','bold');
ylabel('a','FontWeight','bold');
box off;
```

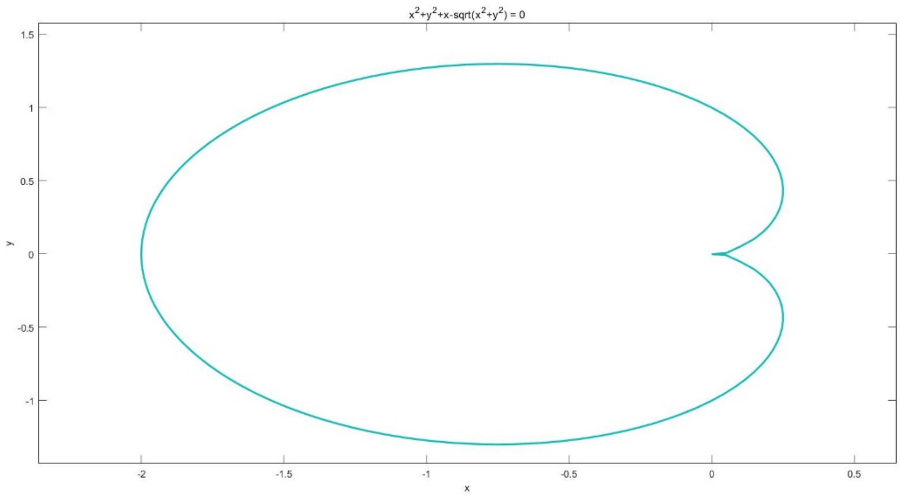
函数曲线



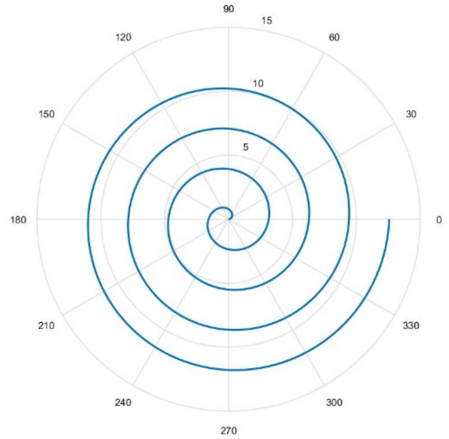
2、用 Matlab 绘制心形线、螺旋线、双曲线、抛物线和椭圆曲线（不是椭圆）等几何曲线

```
% 心形线
ezplot('x^2+y^2+x-sqrt(x^2+y^2)');

% 费马螺线
x=0:0.01*pi:8*pi;
y=x.^1/2;
polar(x,y,'-');

% 双曲线
ezplot('x^2/2-y^2/4-1');
grid on;
line([0 0],[-6 6],'LineWidth', 1.5);
line([-6 6],[0 0],'LineWidth', 1.5);

% 抛物线
ezplot('x*-3+y^2');

% 椭圆曲线
ezplot('y^2-x*(x-1)*(x-2)');
```

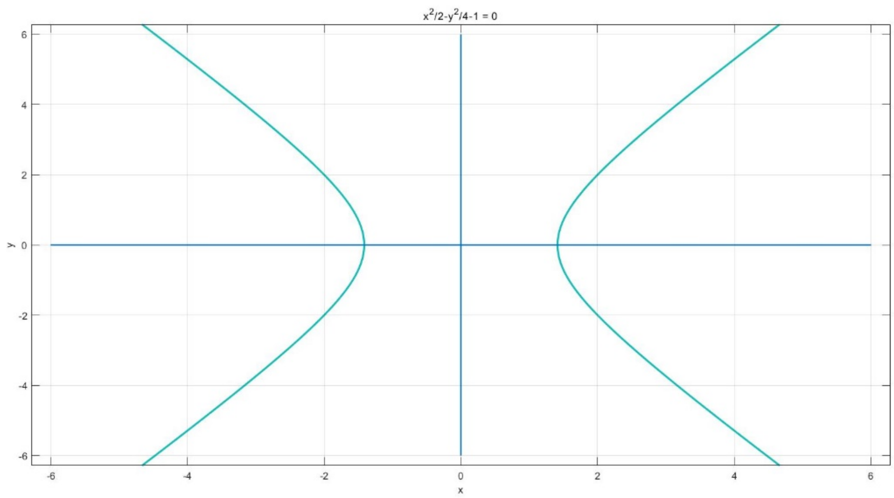
心形线



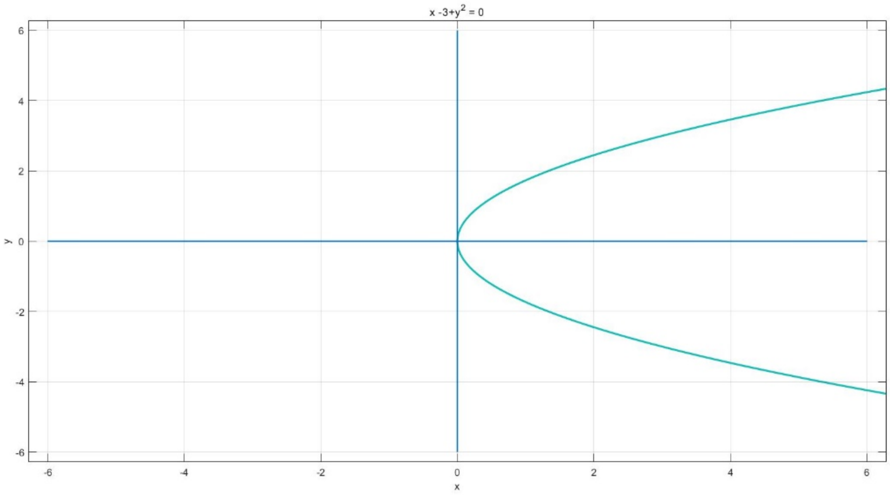
螺旋线


90
15
120
60
10
150
30
5
180
0
210
330
240
300
270

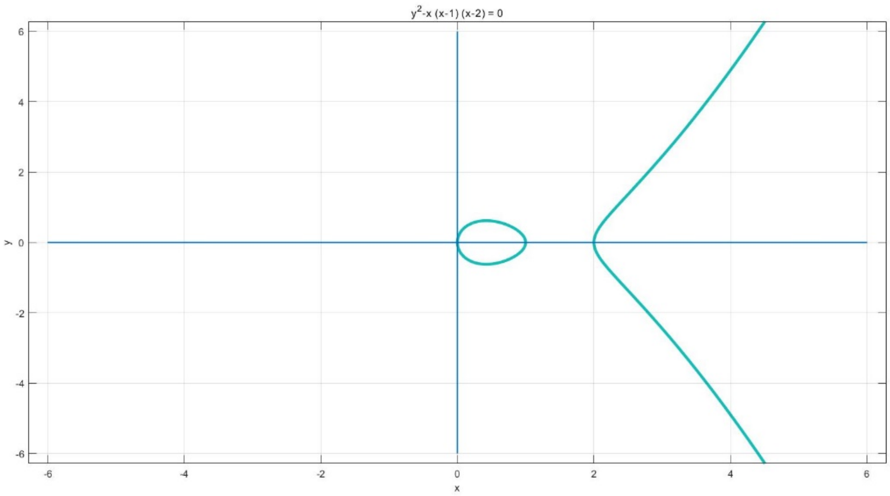
双曲线


x^2/2-y^2/4-1 = 0
6
4
2
y 0
-2
-4
-6
-6
-4
-2
0
2
4
6
x

抛物线




椭圆曲线



# 实 验 三　代数模型实验

1.　（经济计划上的应用）某城市发展经济，主要依靠它的农业、工业及服务业，现经过调查统计，知道该市的“投入”系数如下表 5-5 所示：

表 5-5　某市的投入系数表

| | 农业 | 工业 | 服务业 |
|---|---|---|---|
| 农业 | 0.2 | 0.3 | 0.2 |
| 工业 | 0.4 | 0.1 | 0.2 |
| 服务业 | 0.1 | 0.3 | 0.2 |

现该市预算明年在农业上可有 10 亿元盈余，工业有 5 亿元盈余，服务业有 6 亿元盈余。要达到这项预算，问该市明年农业、工业和服务业的生产总值分别是多少？

题目分析：“投入系数”即可看成 *W. Leontief* 的投入产出数学模型中的直接消耗系数，盈余应该是“新创价值”中的“纯收入”一项，可以归入$z_i$中。若要求 z，相当于将直接消耗系数矩阵 A 转置求 y，那么即可列出程序求解。

实验过程：

```
clear all;
% “投入”系数矩阵
A=[0.2 0.3 0.2;
0.4 0.1 0.2;
0.1 0.3 0.2];
% 盈余，即相当于 z1...zn,把 A 转置之后相当于 y
profit=[10;5;6];
n = size(profit,1);
W = eye(n) - A';
x = W\profit %总投入/总产出
```

实验结果与分析：

```
x =

   24.8958
   20.1042
   18.7500
```

即农业、工业和服务业的生产总值分别是 24.8958、20.1042、18.7500 亿元才能达到在农业上可有 10 亿元盈余，工业有 5 亿元盈余，服务业有 6 亿元盈余的预算这一目标。

2. （进出口贸易上的应用）假设某地区只有 A、B、C 三个经济部门，它的投入产出表如表 5-6 所示。在此表的基础上，若计划有了改变，即 A 部门要进口 30 吨产品；C 部门产品要出口 36 吨，问总产品与各部门之间的投入将发生什么变化？

表 5-6 投入产出表

| 产出 / 投入 | A | B | C | 最终产品 | 总产品 |
|---|---|---|---|---|---|
| A | 30 | 20 | 50 | 150 | 250 |
| B | 20 | 30 | 35 | 115 | 200 |
| C | 16 | 18 | 2 | 64 | 100 |

题目分析：首先，由题目的信息，可以求出其直接消耗系数矩阵 A。又因为 A 部门要进口 30 吨产品，相当于最终产品减少 30 吨，而 C 部门产品要出口 36 吨，相当于最终产品增加 36 吨。记原来的最终产品数量为矩阵 Y，变化数量为$\Delta y$，则现在的最终产品数量$current_Y$为$Y+\Delta y$。那么由$x=[I-A]^{-1}y$即可求出$current_X_total$。在技术水平没有显著提高的条件下，可以假设直接消耗系数不变。由$a_{ij}=\frac{x_{ij}}{x_j}$得$x_{ij}=a_{ij}\cdot x_j$，那么由$current_X_total$和$A$就可求出$current_X$。

实验过程：

```
clear all;
%原始投入产出值
X=[30 20 50;
20 30 35;
16 18 2];
Y=[150;115;64];                 % 原始最终产品
x_total=[250 200 100];          % 原始总产品
x_rep = repmat( x_total,3,1);   % 复制3份，扩展为3x3矩阵
A=X./x_rep;                     % 求直接消耗系数
delta_y=[-30;0;36];             % y 的变化量
current_Y=Y+delta_y;            % 现在的 Y
% 由 x=(I-A)^-1*y 得
current_x_total=(eye(3)-A)\current_Y;
current_x_rep=repmat(current_x_total',3,1);
% 在技术水平没有显著提高的条件下，假设直接消耗系数不变
% 即可求得各产品与各部门之间的变化
current_X=current_x_rep.*A;
```

实验结果及分析：

| 产出<br>投入 | A | B | C | 最终产品 | 总产品 |
|---|---|---|---|---|---|
| A | 28.648 | 21.430 | 68.656 | 120 | 238.735 |
| B | 19.099 | 32.146 | 48.059 | 115 | 214.304 |
| C | 15.279 | 19.287 | 2.746 | 100 | 137.313 |

上面表格即为变化之后总产品与各部门之间的变化。

3. 设国民经济由农业、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、外部需求、初始投入等如表 5-7 所示。

表 5-7 国民经济三个部门之间的投入产出表

| 产出<br>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 | 0 | 70 | 150 |
| 初始投入 | 35 | 110 | 75 | | |
| 总投入 | 100 | 200 | 150 | | |

根据表 5-7 回答下列问题：

（1） 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，问这三个部门的总产出分别应为多少？

（2） 如果三个部门的外部需求分别增加 1 个单位，问它们的总产出应分别增加多少？

题目分析：首先，由题目的信息，可以求出其直接消耗系数矩阵 A。又因为农业、制造业和服务业的外部需求分别为 50，150，100 亿元，即为$current_Y$。那么由$x=[I-A]^{-1}y$即可求出$current_X_total$。即为总产出。因为$\Delta x=(I-A)^{-1}\Delta y$，当农业的外部需求增加 1 个单位,而其余部门的外部需求不变时，令$\Delta y=[1,0,0]^T$，则 3 个部门的总产出增加$\Delta x$等于$(I-A)^{-1}$的第一列。类似，当制造业的外部需求增加 1 个单位，而其余部门的外部需求不变时，3 个部门的总产出增加等于的第二,以此类推。

实验过程：

```
clear all;
%原始投入产出值
X=[15 30 20;
20 10 60;
30 45 0;];
Y=[35;110;75];                        % 原始最终产品
x_total=[100 200 150];                % 原始总产品
x_rep = repmat( x_total,3,1);         % 复制3份，扩展为3x3矩阵
A=X./x_rep;                           % 求直接消耗系数
current_Y=[50;150;100];               % 现在的 Y
% 由 x=(I-A)^-1*y 得总产出
current_x_total=(eye(3)-A)\current_Y;
% 外部需求分别增加1个单位,总产出增量
delta_x=inv(eye(3)-A);
```

实验结果及分析：

| *current_x_total* |
|---|
| 138.758 |
| 272.561 |
| 202.954 |

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，问这三个部门的总产出分别应为 138.758、272.561、202.954 亿元。

| *Delta_x* | | |
|---|---|---|
| 1.3459 | 0.2817 | 0.2921 |
| 0.5008 | 1.2676 | 0.5738 |
| 0.5164 | 0.3697 | 1.2167 |

从上面表格可知，当农业的外部需求增加 1 个单位,而其余部门的外部需求不变时，三个部门总产出增加$\Delta x$分别为 1.3459、0.5008、0.5164 亿元。当制造业的外部需求增加 1 个单位,而其余部门的外部需求不变时，三个部门总产出增加$\Delta x$分别为 0.2817、1.2676、0.3697 亿元。当服务业的外部需求增加 1 个单位,而其余部门的外部需求不变时，三个部门总产出增加$\Delta x$分别为 0.2921、0.5738、1.2167 亿元。

# 实 验 四　代数变换模型

1. 对明文“Mr Hill made this code.” ，运用不同的密钥矩阵，实现 Hill 加密和解密过程。

实验过程： 实验过程中采用一下对应方法，不区分大小写。

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *a* | *b* | *c* | *d* | *e* | *f* | *g* | *h* | *i* | *j* | *k* | *l* | *m* | *n* |
| *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* |
| *o* | *p* | *q* | *r* | *s* | *t* | *u* | *v* | *w* | *x* | *y* | *z* | | . |
| *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* | *23* | *24* | *25* | *26* | *27* | *28* |

那么明文对应的关系为：

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | *r* | | *h* | *i* | *l* | *l* | | *m* | *a* | *d* | *e* |
| *13* | *18* | *27* | *8* | *9* | *12* | *12* | *27* | *13* | *1* | *4* | *5* |
| | *t* | *h* | *i* | *s* | | *c* | *o* | *d* | *e* | . | |
| *27* | *20* | *8* | *9* | *19* | *27* | *3* | *15* | *4* | *5* | *28* | *27* |

用空格补齐，取 3×3 的密钥，那么明文可以构造的矩阵如下

$$\begin{bmatrix} 13 & 18 & 27 & 8 & 9 & 12 & 12 & 27 \\ 13 & 1 & 4 & 5 & 27 & 20 & 8 & 9 \\ 19 & 27 & 3 & 15 & 4 & 5 & 28 & 27 \end{bmatrix}$$

密钥 K 取下列矩阵，

$$K = \begin{matrix} 12 & x & y \\ 8 & 3 & 5 \\ 7 & 9 & 3 \end{matrix}$$

$det(K) = 11x + 51y - 432$ ,取 $det(K)=-1$，经过枚举法，可以得到特解：

$$\begin{cases} x = 16 \\ y = 5 \end{cases}$$

那么，$K = \begin{matrix} 12 & 16 & 5 \\ 8 & 3 & 5 \\ 7 & 9 & 3 \end{matrix}$

在 MATLAB 中输入如下代码，即可得到实验结果。

```
% 明文
M=[13,18,27,8,9,12,12,27;
13,1,4,5,27,20,8,9;
19,27,3,15,4,5,28,27;
]
% 密钥
K=[12 16 5; 8 3 5; 7 9 3];

det(K) %验证密码矩阵是否可逆

C = mod(K * M,28) %密文
INV_K = inv(K);
INV_K = round(INV_K); %以整数形式参与运算
K2 = mod(INV_K,28) %解密的钥匙
Y = mod(K2 * C,28) %恢复明文
```

实验结果与分析：

恢复明文 $Y = \begin{matrix} 13 & 18 & 27 & 8 & 9 & 12 & 12 & 27 \\ 13 & 1 & 4 & 5 & 27 & 20 & 8 & 9 \\ 19 & 27 & 3 & 15 & 4 & 5 & 0 & 27 \end{matrix}$，

对比输入 M，能够匹配，说明完成加密任务功能。

再看看密文 $C = \begin{matrix} 11 & 3 & 11 & 27 & 0 & 13 & 20 & 15 \\ 14 & 2 & 19 & 14 & 5 & 13 & 8 & 14 \\ 13 & 20 & 10 & 6 & 10 & 27 & 16 & 15 \end{matrix}$，

翻译成字母即：**kck .mtonbsnemhnmtjfj po**

换另一个加密矩阵，取 $K = \begin{matrix} 5 & 2 & 3 & 4 \\ x & 5 & y & 4 \\ 1 & 1 & 3 & 3 \\ 5 & 4 & 3 & 1 \end{matrix}$

$\det(K) = 15 * x + 31 * y - 276$ ,令$\det(K) = -1$,

通过穷举法，可以求得特解：$\begin{cases} x = 8 \\ y = 5 \end{cases}$

即$K = \begin{matrix} 5 & 2 & 3 & 4 \\ 8 & 5 & 5 & 4 \\ 1 & 1 & 3 & 3 \\ 5 & 4 & 3 & 1 \end{matrix}$

那么，明文对应的矩阵 $M = \begin{matrix} 13 & 18 & 27 & 8 & 9 & 12 \\ 12 & 27 & 13 & 1 & 4 & 5 \\ 27 & 20 & 8 & 9 & 19 & 27 \\ 3 & 15 & 4 & 5 & 28 & 0 \end{matrix}$

对应的 MATLAB 代码为：

```
% ============= K2 =============
M=[13,18,27,8,9,12;
12,27,13,1,4,5;
27,20,8,9,19,27;
3,15,4,5,28,0;]
% 密钥
K=[5,2,3,4;
8,5,5,4;
1,1,3,3;
5,4,3,1;];

det(K) %验证密码矩阵是否可逆
C = mod(K * M,28) %密文
INV_K = inv(K);
INV_K = round(INV_K); %以整数形式参与运算
K2 = mod(INV_K,28) %解密的钥匙
Y = mod(K2 * C,28) %恢复明文
```

实验结果与分析：

$$恢复明文\ Y = \begin{matrix} 13 & 18 & 27 & 8 & 9 & 12 \\ 12 & 27 & 13 & 1 & 4 & 5 \\ 27 & 20 & 8 & 9 & 19 & 27 \\ 3 & 15 & 4 & 5 & 0 & 0 \end{matrix}$$

对比输入 M，能够匹配，说明完成加密任务功能。

$$再看看密文\ C = \begin{matrix} 14 & 12 & 5 & 5 & 26 & 11 \\ 3 & 19 & 1 & 22 & 19 & 4 \\ 3 & 10 & 20 & 23 & 14 & 14 \\ 1 & 21 & 19 & 20 & 6 & 21 \end{matrix},$$

翻译成字母即：**nlee. kcsavsdcjtwnnaustfu**

2. 甲方截获了一段密文：O J W P I S W A Z U X A U U I S E A B A U C R S I P L B H A A M M L P J J O T E N H（中间没有空格）。经分析这段密文是用 *Hill2*（密钥矩阵为 2 阶矩阵）密码编译的，且这段密码的字母 U C R S 一次代表字母 T A C O，问能否破译这段密文的内容？

实验过程：实验过程中采用一下对应方法，不区分大小写。

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *a* | *b* | *c* | *d* | *e* | *f* | *g* | *h* | *i* | *j* | *k* | *l* | *m* | *n* |
| *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* |
| *o* | *p* | *q* | *r* | *s* | *t* | *u* | *v* | *w* | *x* | *y* | *z* | | |
| *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* | *23* | *24* | *25* | *26* | | |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***O*** | ***J*** | ***W*** | ***P*** | ***I*** | ***S*** | ***W*** | ***A*** | ***Z*** | ***U*** | ***X*** | ***A*** | ***U*** | ***U*** |
| *15* | *10* | *23* | *16* | *9* | *19* | *23* | *1* | *26* | *21* | *24* | *1* | *21* | *21* |
| ***I*** | ***S*** | ***E*** | ***A*** | ***B*** | ***A*** | ***U*** | ***C*** | ***R*** | ***S*** | ***I*** | ***P*** | ***L*** | ***B*** |
| *9* | *19* | *5* | *1* | *2* | *1* | *21* | *3* | *18* | *19* | *9* | *16* | *12* | *2* |
| ***H*** | ***A*** | ***A*** | ***M*** | ***M*** | ***L*** | ***P*** | ***J*** | ***J*** | ***O*** | ***T*** | ***E*** | ***N*** | ***H*** |
| *8* | *1* | *1* | *13* | *13* | *12* | *16* | *10* | *10* | *15* | *20* | *5* | *14* | *8* |

因为 U C R S 一次代表字母 T A C O，那么

$$\begin{pmatrix} 21 & 18 \\ 3 & 19 \end{pmatrix} = K \cdot \begin{pmatrix} 20 & 3 \\ 1 & 15 \end{pmatrix}$$

可以求得 $K$ 的模 26 逆矩阵$K_n = K^{-1}\,mod\,26$，再通过$M = K_n \cdot C$即可求得对应的明文。

考虑到是 *Hill2* 加密，那么密文矩阵为

$$\binom{15}{10}\binom{23}{16}\binom{9}{19}\binom{23}{1}\binom{26}{21}\binom{24}{1}\binom{21}{21}\binom{9}{19}\binom{5}{1}\binom{2}{1}\binom{21}{3}\binom{18}{19}\binom{9}{16}\binom{12}{2}\binom{8}{1}\binom{1}{13}\binom{13}{12}\binom{16}{10}\binom{10}{15}\binom{20}{5}\binom{14}{8}$$

MATLAB 程序如下，事先定义函数$GetAn()$，用于求矩阵的模 26 逆矩阵。

```
function An = GetAn(A)
%GetAn 模26逆矩阵
%
% Syntax: output = GetAn(input)
%
% 输入矩阵A，如果其模26逆矩阵存在的话则返回其模26逆矩阵

DA=round(mod(det(A),26));
for x = 1:100
    if mod(x*DA,26)==1
        res=x;
        An=round(mod(inv(A)*det(A)*res,26));
        break;
    end
end
end
```

主程序如下

```
clear all
C=[21 18; 3 19];
M=[20 3;1 15];
full_C=[15,10,23,16,9,19,23,1,26,21,24,1,21,21,9,19,5,1,2,1,21,3,18,19,
9,16,12,2,8,1,1,13,13,12,16,10,10,15,20,5,14,8];


% 由C=K*M 求 M

Kn=mod(M*GetAn(C),26);
syms Full_M
for i = 1:2:length(full_C)
    dC=[full_C(i);full_C(i+1)];
    dM=mod(Kn*dC,26);
    Full_M(i)=dM(1);
    Full_M(i+1)=dM(2);
end
```

实验结果与分析：

```
>> Q2_v
>> Full_M

Full_M =

[ 3, 12, 9, 14, 20, 15, 14, 9, 19, 7, 15, 9, 14, 7, 20, 15, 22, 9, 19, 9, 20, 1, 3, 15, 21, 14, 20, 18, 25, 9, 14, 13, 9, 4, 4, 12, 5, 5, 1, 19, 20, 20]
```

经过翻译、添加空格断句之后，则

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **3** | **12** | **9** | **14** | **20** | **15** | **14** | | **9** | **19** | | **7** | **15** |
| C | L | I | N | T | O | N | | I | S | | G | O |
| **9** | **14** | **7** | | **20** | **15** | | **22** | **9** | **19** | **9** | **20** | |
| I | N | G | | T | O | | V | I | S | I | T | |
| **1** | | **3** | **15** | **21** | **14** | **20** | **18** | **25** | | **9** | **14** | |
| A | | C | O | U | N | T | R | Y | | I | N | |
| **13** | **9** | **4** | **4** | **12** | **5** | | **5** | **1** | **19** | **20** | **20** | |
| M | I | D | D | L | E | | E | A | S | T | T | |

那么对应的明文为：***Clinton is going to visit a country in middle east.***

如果是 O J W P 对应 T A C O 的话，那么有$C = \begin{pmatrix} 15 & 23 \\ 10 & 16 \end{pmatrix}$，

所以有$det\,C(mod\,26) = 10$，即不存在 C 的模 26 逆矩阵。所以无法解密。

# 实 验 五　代数方程模型实验

1. 某中药厂用 9 种中草药 A-I，根据不同的比例配制成了 7 种特效药，各用量成分见表 6-3（单位：克）。

表 6-3 7 种特效药的成分

| 中药 | 1号成药 | 2号成药 | 3号成药 | 4号成药 | 5号成药 | 6号成药 | 7号成药 |
|---|---|---|---|---|---|---|---|
| A | 10 | 2 | 14 | 12 | 20 | 38 | 100 |
| B | 12 | 0 | 12 | 25 | 35 | 60 | 55 |
| C | 5 | 3 | 11 | 0 | 5 | 14 | 0 |
| D | 7 | 9 | 25 | 5 | 15 | 47 | 35 |
| E | 0 | 1 | 2 | 25 | 5 | 33 | 6 |
| F | 25 | 5 | 35 | 5 | 35 | 55 | 50 |
| G | 9 | 4 | 17 | 25 | 2 | 39 | 25 |
| H | 6 | 5 | 16 | 10 | 10 | 35 | 10 |
| I | 8 | 2 | 12 | 0 | 2 | 6 | 20 |

试解答：（1）某医院要购买这 7 种特效药，但药厂的第 3 号药和第 6 号药已经卖完，请问能否用其他特效药配制出这两种脱销的药品。

（2）现在该医院想用这 7 种草药配制三种新的特效药，表 6-4 给出了三种新的特效药的成分，请问能否配制？如何配制？

表 6-4 3 种新特效药的成分

| 中药 | 1号新药 | 2号新药 | 3号新药 |
|---|---|---|---|
| A | 40 | 162 | 88 |
| B | 62 | 141 | 67 |
| C | 14 | 27 | 8 |
| D | 44 | 102 | 51 |
| E | 53 | 60 | 7 |
| F | 50 | 155 | 80 |
| G | 71 | 118 | 38 |
| H | 41 | 68 | 21 |
| I | 14 | 52 | 30 |

题目分析：

由题目：

$$A=\begin{pmatrix}10&2&14&12&20&38&100\\12&0&12&25&35&60&55\\5&3&11&0&5&14&0\\7&9&25&5&15&47&35\\0&1&2&25&5&33&6\\25&5&35&5&35&55&50\\9&4&17&25&2&39&25\\6&5&16&10&10&35&10\\8&2&12&0&2&6&20\end{pmatrix}$$

可以求出 A 的最大无关组，进而求得其他结果。

实验过程：

① 求 A 的最大无关组和秩的 Matlab 代码：

```
clear
% 原始数据
m1=[10; 12; 5; 7; 0; 25; 9; 6; 8];
m2=[2; 0; 3; 9; 1; 5; 4; 5; 2];
m3=[14; 12; 11; 25; 2; 35; 17; 16; 12];
m4=[12; 25; 0; 5; 25; 5; 25; 10; 0];
m5=[20; 35; 5; 15; 5; 35; 2; 10; 2];
m6=[38; 60; 14; 47; 33; 55; 39; 35; 6];
m7=[100; 55; 0; 35; 6; 50; 25; 10; 20];
M=[m1,m2,m3,m4,m5,m6,m7];
[M0,jb]=rref(M)
r=length(jb)
```

② 求成药 3 的配方的 Matlab 代码：

```
B=[m1,m2,m4,m5,m6,m7];
x3=B\m3;
```

③ 求 3 种新药的配方的 Matlab 代码：

```
% 三种新药
nm1=[40,62,14,44,53,50,71,41,14]';
nm2=[162,141,27,102,60,155,118,68,52]';
nm3=[88,67,8,51,7,80,38,21,30]';

B_nm1=[m1,m2,m4,m5,m6,m7,nm1];
B_nm1_0=rref(B_nm1)

B_nm2=[m1,m2,m4,m5,m6,m7,nm2];
B_nm2_0=rref(B_nm2)

B_nm3=[m1,m2,m4,m5,m6,m7,nm3];
B_nm3_0=rref(B_nm3)
```

实验结果与分析：

```
M0 =
     1     0     1     0     0     0     0
     0     1     2     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
jb =
     1     2     4     5     6     7
r =
     6
```

对于第一题，可以看出他的秩是 6，也就是说他的最大无关组内有六个，所以不能不能由另外五个完全配出 3 和 6 号药，但是又有，

```
x3 =
    1.0000
    2.0000
    0.0000
    0.0000
   -0.0000
    0.0000
```

也就是说由 1 份的 1 号药加 2 份的 2 号药就可以配出 3 号药。

综上所述，能用其他特效药脱销的 3 号药，但配不出 6 号药。

下面是关于三种新药的结果

```
B_nm1_0 =
     1     0     0     0     0     0     1
     0     1     0     0     0     0     3
     0     0     1     0     0     0     2
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
```

由上述结果，可以配得出新药 1，配方为 1 份 1 号成药，3 份 2 号成药，2 份 4 号成药，配出 1 份新药 1。

```
B_nm2_0 =
     1     0     0     0     0     0     3
     0     1     0     0     0     0     4
     0     0     1     0     0     0     2
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     1
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
```

由上述结果，可以配得出新药 2，配方为 3 份 1 号成药，4 份 2 号成药，2 份 4 号成药，6 份 7 号成药，配出 1 份新药 2。

```
B_nm3_0 =
     1     0     0     0     0     0     0
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
```

由上述结果，配不出新药 3。

2. （营养食谱问题）一个饮食专家计划一份膳食，提供一定量的维生素 C、钙和镁。其中用到 3 种食物，它们的质量用适当的单位计量。这些食品提供的营养以及食谱需要的营养如下表给出。

表 7-4 营养食谱问题

| 营养 | 单位食谱所含的营养（毫克） | | | 需要的营养总量（毫克） |
|---|---|---|---|---|
| | 食物1 | 食物2 | 食物3 | |
| 维生素C | 10 | 20 | 20 | 100 |
| 钙 | 50 | 40 | 10 | 300 |
| 镁 | 30 | 10 | 40 | 200 |

针对这个问题写出一个向量方程。说明方程中的变量表示什么，然后求解这个方程。

实验过程：

可以列出向量方程：

$$\begin{pmatrix} 10 & 20 & 20 \\ 50 & 40 & 10 \\ 30 & 10 & 40 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 100 \\ 300 \\ 200 \end{pmatrix}$$

在 Matlab 中列出方程求解：

① 求系数矩阵的秩的 Matlab 代码：

```
f1=[10; 50; 30];
f2=[20; 40; 10];
f3=[20; 10; 40];
F=[f1,f2,f3];
rank(F);
```

② 求解未知数的 Matlab 代码：

```
Need=[100; 300; 200];
x=F\Need
```

实验结果与分析：

可以求出系数矩阵的秩为 3，等于未知数个数，因而方程有唯一解。解得

```
x =
    4.5455
    1.5152
    1.2121
```

由上述结果可知，每天要摄入 4.55 个单位的食物 1、1.52 个单位的食物 2、1.21 个单位的食物 3 才能满足需要的营养总量。

3．在美国黄衫森林中，斑点猫头鹰主要以鼷鼠为食。假设这两个种群的“捕食者-被捕食者”矩阵为$A=\begin{matrix}0.4 & 0.3\\ -p & 1.2\end{matrix}$。

（1）证明：如果捕食参数 *p=0.325*，则两个种群都会增长。估计长期增长率及猫头鹰与鼷鼠的最终比值。

（2）证明：如果捕食率为 *p=0.5*，则猫头鹰和鼷鼠最终都将灭绝。

（3）试求一个 *p* 值，使得猫头鹰和鼷鼠的数量都趋于稳定。此时，对应的种群数量是多少？

实验过程：

① 证明 *p=0.325* 两个种群都会增长的 Matlab 代码：

```
A1=[0.4,0.3;
    -0.325,1.2];
[pc,lambda] = eig(A1); %求A 的特征值和对应的特征向量
[Y,I] = sort(diag(abs(lambda)),'descend');%对特征值的绝对值降序排列
temp = diag(lambda);
lambda = temp(I);      %输出按特征值的绝对值降序排列的特征值
pc = pc(:,I);          %与特征值对应的特征向量
```

② 证明 *p=0.5* 猫头鹰和鼷鼠最终都将灭绝的 Matlab 代码：

```
A2=[0.4,0.3;
    -0.5,1.2];
[pc,lambda] = eig(A2);
[Y,I] = sort(diag(abs(lambda)),'descend');
temp = diag(lambda);
lambda = temp(I)
pc = pc(:,I)
```

③ 假设捕食参数为 p，求出矩阵的特征值，令较大的特征值为 1，即可求出此时的 p，也就是令得猫头鹰和鼷鼠的数量都趋于稳定的捕食参数 p。

实现的 Matlab 代码：

```
syms p;
Ap=[0.4,0.3;
    -p,1.2];
[pc,lambda] = eig(Ap);
[Y,I] = sort(diag(abs(lambda)),'descend');
temp = diag(lambda);
lambda = temp(I)
p=solve(lambda(1)==1) %令较大的特征值为1，使得种群趋于稳定
Ap=[0.4,0.3;
    -p,1.2];
[pc,lambda] = eig(Ap);
[Y,I] = sort(diag(abs(lambda)),'descend');
temp = diag(lambda);
lambda = temp(I)
pc = pc(:,I)
```

实验结果与分析：

① 捕食参数 *p=0.325* 的时候有：

```
lambda =
    1.0500
    0.5500
pc =
   -0.4191   -0.8944
   -0.9080   -0.4472
```

显然，这两个特征向量（即 pc 的第一列和第二列）是线性无关的，它们构成的一组基。为消除小数，我们选取

$$v_1 = \binom{1}{2}, \quad v_2 = \binom{2}{1}$$

那么存在$c_1, c_2$，使得

$$x_k = c_1\lambda^k v_1 + c_2 u^k v_2 = c_1 \cdot 1.05^k \cdot v_1 + c_2 \cdot 0.55^k \cdot v_2$$

当$k \to \infty$时，$(0.55^k)$迅速趋于 0。那么有

$$x_k \approx c_1 \cdot 1.05^k \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

即有 5%的增长率，最终比值为 1：2

② 捕食参数 *p=0.5* 的时候有：

```
lambda =
    0.9000
    0.7000
pc =
   -0.5145   -0.7071
   -0.8575   -0.7071
```

显然，这两个特征向量（即 pc 的第一列和第二列）是线性无关的，它们构成的一组基。为消除小数，我们选取

$$v_1 = \begin{pmatrix} 51 \\ 86 \end{pmatrix}, \qquad v_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

那么存在$c_1, c_2$，使得

$$x_k = c_1 \lambda^k v_1 + c_2 u^k v_2 = c_1 \cdot 0.9^k \cdot v_1 + c_2 \cdot 0.7^k \cdot v_2$$

当$k \to \infty$时，$(0.9^k)$、$(0.7^k)$都迅速趋于 0。也就是说猫头鹰和鼹鼠最终都将灭绝。

③ 趋于平衡的时候，

```
lambda =
 (2^(1/2)*(8 - 15*p)^(1/2))/10 + 4/5
 4/5 - (2^(1/2)*(8 - 15*p)^(1/2))/10

p = 2/5

lambda =
   1
 3/5

pc =
[ 1/2, 3/2]
[   1,   1]
```

可以看出，捕食参数 p 为 0.4，此时对应的$v_1 = \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix}$

$$x_k = c_1 1^k \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} + c_2 \cdot 0.6^k \begin{pmatrix} \frac{3}{2} \\ 1 \end{pmatrix}$$

当$k \to \infty$时，$(0.6^k)$迅速趋于 0。也就是说$x_k = c_1 \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix}$。

4. 杂交育种的目的是培养优良品种，以提高农作物的产量和质量。如果农作物的三种基因型分别为 AA，Aa，aa，其中 AA 为优良品种。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（见下表）。问：经过若干年后三种基因型分布如何？ 要求：

（1）建立代数模型，从理论上说明最终的基因型分布。

（2）用 MATLAB 求解初始分布为 0.8，0.2，0 时，20 年后的基因分布，是否已经趋于稳定？

表 8-1 基因的转移

<table>
<tr><td colspan="2" rowspan="2">概率</td><td colspan="3">父体-母体基因型</td></tr>
<tr><td>AA-AA</td><td>AA-Aa</td><td>AA-aa</td></tr>
<tr><td rowspan="3">后代的<br>基因型</td><td>AA</td><td>1</td><td>1/2</td><td>0</td></tr>
<tr><td>Aa</td><td>0</td><td>1/2</td><td>1</td></tr>
<tr><td>aa</td><td>0</td><td>0</td><td>0</td></tr>
</table>

实验过程：

① 求矩阵 A 的特征值和特征向量的 Matlab 代码：

```
clear all
clc
A=[1,1/2,0; 0,1/2,1; 0,0,0;];
[pc,lambda] = eig(A);
[Y,I] = sort(diag(abs(lambda)),'descend');
temp = diag(lambda);
lambda = temp(I)
lambda_norm=[norm(lambda(1));norm(lambda(2));norm(lambda(3))]
pc = pc(:,I)
```

② 根据初始状态确定基因型分布模型中的常系数$c_i$的 Matlab 代码：

```
n=[1.0000,-0.7071;
    0,0.7071];
rs=[0.8; 0.2];
c=n\rs
```

③ 求 20 年后基因型分布的 Matlab 代码：

```
n=20;
x_n=[0.8; 0; 0]+0.4*0.5^(n)*[-0.7071; 0.7071; 0]
```

实验结果与分析：

```
lambda =
    1.0000
    0.5000
         0
lambda_norm =
    1.0000
    0.5000
         0
pc =
    1.0000   -0.7071    0.4082
         0    0.7071   -0.8165
         0         0    0.4082
```

由上述结果，可以得知他的三个特征值

$$\lambda = 1, \qquad u = 0.5, \qquad w = 0$$

以及三个对应的特征向量如下

$$v_1 = (1 \quad 0 \quad 0)^T$$

$$v_2 = (-0.7071 \quad 0.7071 \quad 0)^T$$

$$v_3 = (0.4082 \quad -0.8165 \quad 0.4082)^T$$

那么，$x_{k+1} = Ax_k$的通解形如

$$x_k = c_1\lambda^k v_1 + c_2 u^k v_2 = c_1 \cdot (1 \quad 0 \quad 0)^T + c_2 \cdot 0.5^k \cdot (-0.7071 \quad 0.7071 \quad 0)^T$$

也就是说，最终的基因型分布$x_k$

$$x_k = c_1 \cdot (1 \quad 0 \quad 0)^T + c_2 \cdot 0.5^k \cdot (-0.7071 \quad 0.7071 \quad 0)^T$$

之后由初始条件$x_0 = \begin{pmatrix} 0.8 \\ 0.2 \\ 0 \end{pmatrix}$，那么

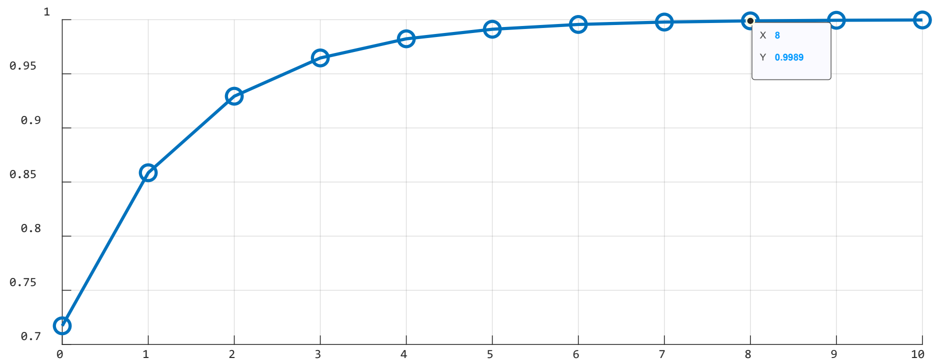
```
c =
    1.0000
    0.2828
```

所以$x_k = (1 \quad 0 \quad 0)^T + 0.2828 \cdot 0.5^k \cdot (-0.7071 \quad 0.7071 \quad 0)^T$

20 年之后的结果为：

```
x_n =
    1.0000
    0.0000
         0
```

显然，20 年后已经达到稳定。事实上，我们可以绘出 AA 基因型的增长曲线，由下图可知，可以认为在第八年已经达到稳定状态，种群中 AA 占比达到 99.89%了。



# 实 验 六　　分形实验

1. 对一个等边三角形，每条边按照 Koch 曲线的方式进行迭代，产生的分形图称为 Koch 雪花。编制程序绘制出它的图形，并计算 Koch 雪花的面积，以及它的分形维数。

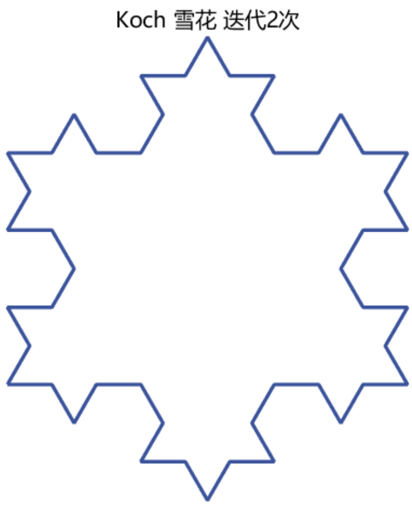
实验过程：

定义一个名为 KochSnow 的函数，用于绘制 Koch 雪花

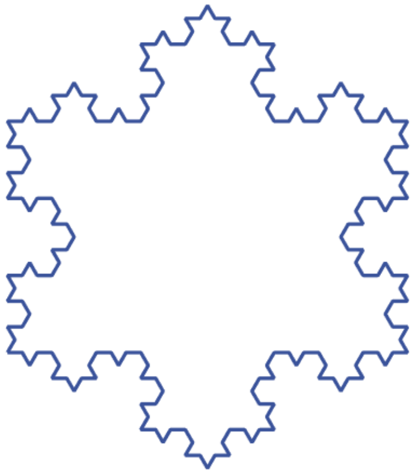
```
function KochSnow(length, k)
    %KochSnow - KochSnow 分形图
    %
    % Syntax: KochSnow(length,k)
    %
    % 输入参数 length 为三角形边长，k 为迭代次数

    % 依次画三角形的三条边
    for myedge = 1:3
        switch myedge
            case 1% 三角形的第一条边的起点终点
                p = [0, 0; 0.5 * length, sqrt(3) / 2 * length];
            case 2% 三角形的第二条边的起点终点
                p = [0.5 * length, sqrt(3) / 2 * length; length, 0];
            case 3% 三角形的第三条边的起点终点
                p = [length, 0; 0, 0];
        end
        n = 1; %存放线段的数量，初始值为1
        A = [cos(pi / 3), -sin(pi / 3); sin(pi / 3), cos(pi / 3)];
        for s = 1:k%实现迭代过程，计算所有的结点的坐标
            j = 0; %
            %以下根据线段两个结点的坐标，计算迭代后它们之间增加的三个
            %结点的坐标，并且将这些点的坐标按次序存暂时放到 r 中
            for i = 1:n%每条边计算一次
                q1 = p(i, :); %目前线段的起点坐标
                q2 = p(i + 1, :); %目前线段的终点坐标
                d = (q2 - q1) / 3; %
                j = j + 1; r(j, :) = q1; %原起点存入 r
                j = j + 1; r(j, :) = q1 + d; %新1点存入 r
                j = j + 1; r(j, :) = q1 + d + d * A'; %新2点存入 r
                j = j + 1; r(j, :) = q1 + 2 * d; %新3点存入 r
            end %原终点作为下条线段的起点，在迭代下条线段时存入 r
```

```
            n = 4 * n; %全部线段迭代一次后，线段数量乘4
            clear p%清空p ，注意：最后一个终点q2不在r 中
            p = [r; q2]; %重新装载本次迭代后的全部结点
        end
        clear r
        % 储存各边的结点数据
        switch myedge
            case 1
                myedge_1 = p;
            case 2
                myedge_2 = p;
            case 3
                myedge_3 = p;
        end
    end
    wholeEdge = [myedge_1; myedge_2; myedge_3];
    plot(wholeEdge(:, 1), wholeEdge(:, 2))%显示各结点的连线图
    axis equal%各坐标轴同比例
end
```

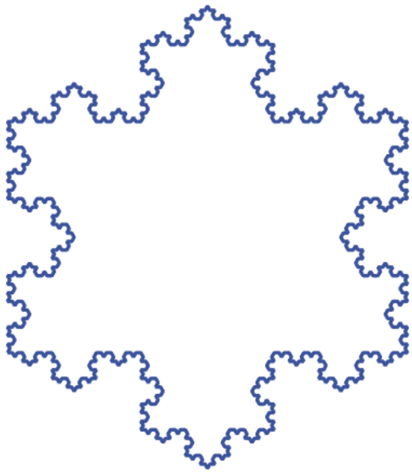
实验结果与分析：



Koch 雪花 迭代3次



Koch 雪花 迭代4次



由迭代的规律可以知道，相似形个数$m=6$，边长的放大倍数$c=3$，那么：

$$分形维数d=\frac{\ln m}{\ln c}=\frac{\ln 6}{\ln 3}\approx 1.631$$

图形的面积可以由下面的规律得到。记三角形初始边长为$a$，迭代次数为$n$ ,那么:

当$n=0$ 的时候，$S=\frac{\sqrt{3}}{4}a^2$ ；

当$n=1$ 的时候，$S=\frac{\sqrt{3}}{4}a^2+\frac{\sqrt{3}}{4}\times\left(\frac{1}{3}R\right)^2\times 3$ ；

当$n=2$ 的时候，$S=\frac{\sqrt{3}}{4}a^2+\frac{\sqrt{3}}{4}\times\left(\frac{1}{3}R\right)^2\times 3+\frac{\sqrt{3}}{4}\times\left[\left(\frac{1}{3}\right)^2R\right]^2\times 3^2$ ；

……

当$n=k$ 的时候，$S=\frac{\sqrt{3}}{4}a^2+\frac{\sqrt{3}}{4}\times\left(\frac{1}{3}R\right)^2\times 3+\cdots+\frac{\sqrt{3}}{4}\times\left[\left(\frac{1}{3}\right)^kR\right]^2\times 3^n$ ；

那么，当$n\to\infty$的时候，$S\to\infty$。

2. 对一条竖向线段，在其三分之一分点处，向左上方向画一条线段，在其三分之二分点处，向右上方向画一条线段，线段长度都是原来的三分之一，夹角都为 30°，迭代一次后变成上图 3-22。继续迭代得到分形图，可模拟树木花草。编制程序绘制出它的图形。

实验过程：

主函数：

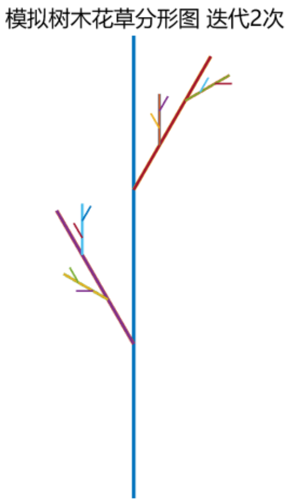
为了使得中间逻辑更加清晰，把每次绘图的部分放在另一个函数 eachTree 中

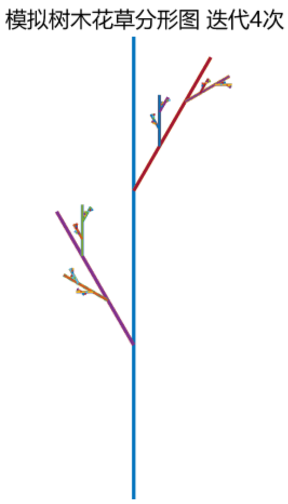
```
function tree(height, times)
    %tree - 画树
    %
    % Syntax: tree(初始高度,迭代次数)
    %
    % 对一条竖向线段，在其三分之一分点处，向左上方向画一条线段，在其三分之二分点处，向右上方向画一条线段，线段长度都是原来的三分之一，夹角都为30°

    [ori, z] = eachTree(0, height * i, times, 0);
    new_ori = [];
    new_z = [];

    for k = 1:times
        for j = 1:length(ori)
            [p, q] = eachTree(ori(j), z(j), times, k);
            new_ori = cat(2, new_ori, p);
            new_z = cat(2, new_z, q);
        end
        ori = new_ori;
        z = new_z;
    end
    ax = gca;
    ax.Title.String = ['模拟树木花草分形图 迭代',num2str(times),'次'];
    ax.Title.FontWeight =  'normal';
    ax.Title.FontSize = 12;
    ax.Title.FontName =  '微软雅黑';
    ax.YAxis.Visible =  'off'; % 设置y 轴不可见
    ax.XAxis.Visible =  'off'; % 设置x 轴不可见
end
```

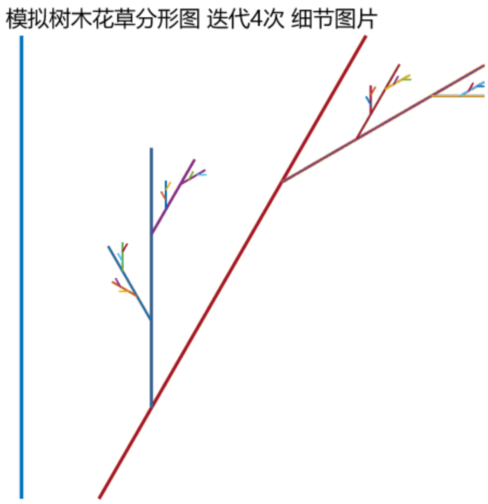
定义一个函数 eachTree，用于每次画图，并返回新分支的起点和终点

```
function [new_origin, new_ending] = eachTree(origin, ending,times,current)
    %eachTree-eachTree
    %
    % Syntax: eachTree(origin,ending)
    %
    %
    % 对一条竖向线段，在其三分之一分点处，向左上方向画一条线段，在其三分之二分点处，向右上方向画一条线段，线段长度都是原来的三分之一，夹角都为30°


    trans_vec = ending - origin;
    n1 = ending / 3 + 2/3 * origin;
    n2 = 2/3 * ending + origin / 3;
    p1 = n1 + 1/3 * abs(trans_vec) * exp(i * (angle(trans_vec) + pi / 6));
    p2 = n2 + 1/3 * abs(trans_vec) * exp(i * (angle(trans_vec) - pi / 6));
    plot([origin, ending],'LineWidth',2-1/times*current); hold on;
    plot([n1, p1],'LineWidth',2-1/times*current); hold on;
    plot([n2, p2],'LineWidth',2-1/times*current); hold on;
    new_origin = [n1, n2];
    new_ending = [p1, p2];
    axis equal
end
```

实验结果与分析：





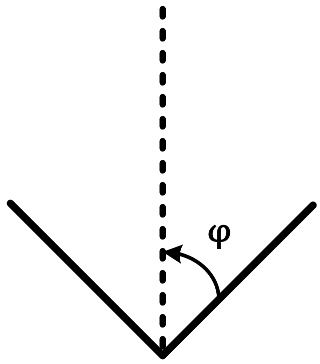




由上图，可以看出程序满足题目要求，生成正确的图形。

3. 自己构造生成元（要有创意），按照图形迭代的方式产生分形图，用计算机编制程序绘制出它的图形，并计算其分形维数。

实验过程：

对一条长度为 $l$ 的竖向初始虚拟线段，在其一个端点处，向左右各作一条长度为 $\frac{\sqrt{5}-1}{2}l$ （黄金比例）的线段，每条线段与原始线段夹角记为偏转角 $\varphi$。如右图所示。继续迭代，则产生分形图形。




Matlab 代码如下：

先定义一个用于每个结点绘制分支的函数 eachNode

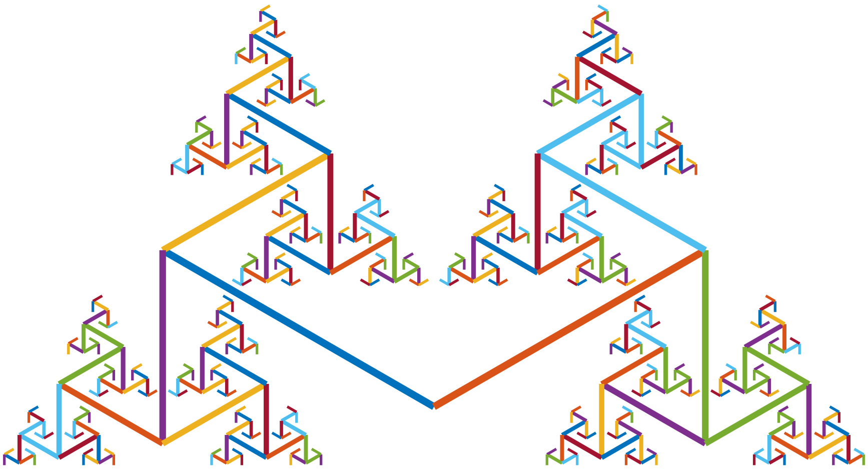
```
function [new_ori, new_ending] = eachNode(deflection_an-
gle, origin, ending, nowInex, mostWidth, iterations)
    % eachNode - 绘制每个结点的函数
    %
    % Syntax: [ori,ending] = eachNode(偏转角，起点，中点，当前层次，最粗的
线宽，迭代次数)
    %
    % 绘制每个结点的函数，返回新结点的起点和终点

    trans_vec = ending - origin;
    p1 = (sqrt(5) - 1) / 2 * abs(trans_vec) * exp(i * (deflection_an-
gle + angle(trans_vec))) + origin; %新节点1的坐标
    p2 = (sqrt(5) - 1) / 2 * abs(trans_vec) * exp(i * (-deflection_an-
gle + angle(trans_vec))) + origin; %新节点2的坐标
    plot([origin, p1],  'linewidth', mostWidth - nowInex * (most-
Width - 1) / iterations); hold on;
    plot([origin, p2],  'linewidth', mostWidth - nowInex * (most-
Width - 1) / iterations); hold on;

    new_ori = [p1, p2];
    new_ending = [origin, origin];
end
```

用于迭代调用 eachNode 的主函数 myGraphics

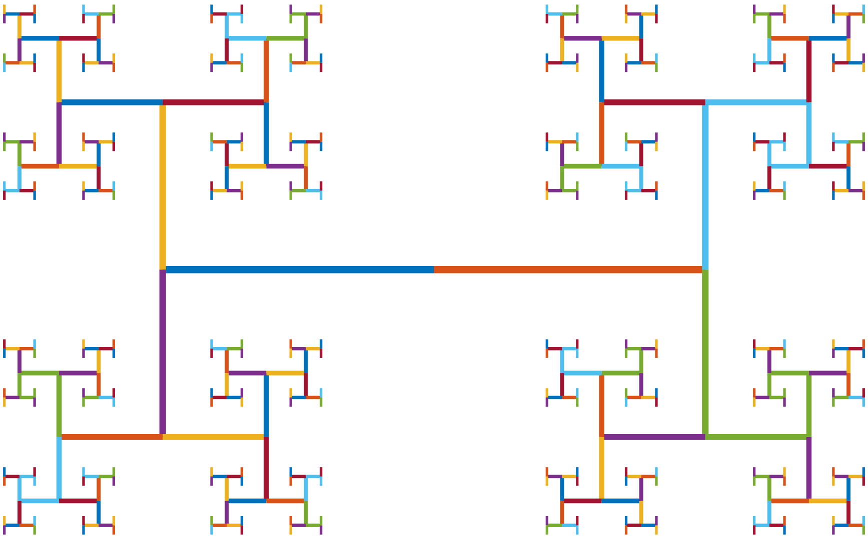
```
function myGraphics(mylength, deflection_angle, times, mostWidth)
    %myGraphics - Description
    %
    % Syntax: myGraphics(mylength,deflection_angle,times,mostWidth)
    %
    % mylength:初始长度
    % deflection_angle: 偏转角
    % times: 迭代次数
    % mostWidth: 最粗的线宽

    input_origin = i;
    input_ending = mylength * i;
    newOrigin = [];
    newEnding = [];
    origin = [input_origin];
    ending = [input_ending];

    % ==== 开始套娃 ====
    for k = 1:times
        newOrigin = [];
        newEnding = [];
        for j = 1:length(origin)
            [p, q] = eachNode(deflection_angle, origin(j), end-
ing(j), k, mostWidth, times);
            newOrigin = [newOrigin, p];
            newEnding = [newEnding, q];
        end
        origin = newOrigin;
        ending = newEnding;
    end
    axis equal%各坐标轴同比例
    ax = gca;
    ax.Title.String = ['【我的图形】 偏转角: ',num2str(rad2deg(deflec-
tion_angle)), '°迭代', num2str(times), '次'];
    ax.Title.FontWeight =  'normal';
    ax.Title.FontSize = 12;
    ax.Title.FontName = '微软雅黑';
    ax.YAxis.Visible =  'off'; % 设置y 轴不可见
    ax.XAxis.Visible =  'off'; % 设置x 轴不可见
end
```

实验结果与分析：

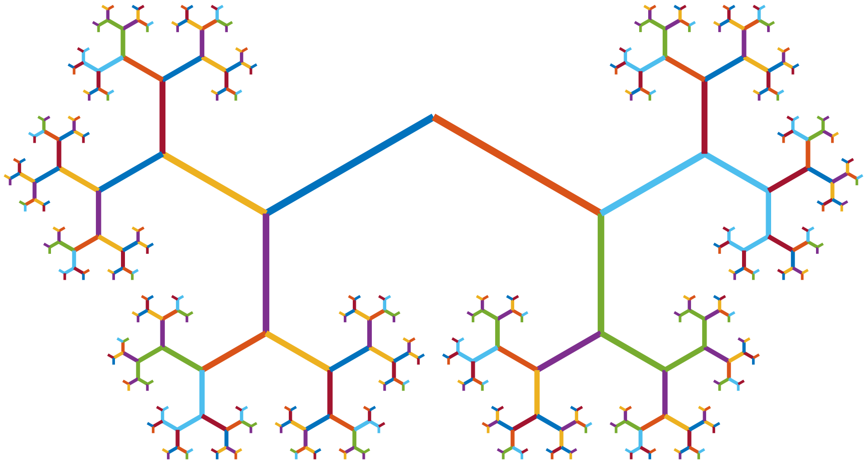
【我的图形】 偏转角：60° 迭代8次



【我的图形】 偏转角：90° 迭代8次



【我的图形】偏转角：120° 迭代8次



【我的图形】偏转角：135° 迭代5次

由迭代的规律可以知道，相似形个数$m=2$，边长的放大倍数$c=\frac{2}{\sqrt{5}-1}$，那么：

$$分形维数d=\frac{\ln m}{\ln c}=\frac{\ln 2}{\ln\left(\frac{2}{\sqrt{5}-1}\right)}=\frac{\ln 2}{\ln 2-\ln(\sqrt{5}-1)}\approx 1.4404$$

# 实验七　插值实验

1、已知欧洲一个国家的地图，为了算出它的国土面积，对地图作了如下测量：以由西向东方向为 x 轴，由南向北方向为 y 轴，选择方便的原点，并将从最西边界点到最东边界点在 x 轴上的区间适当的分为若干段，在每个分点的 y 方向测出南边界点和北边界点的 y 坐标 y1 和 y2，这样就得到下表的数据（单位：mm）。根据地图的比例，18 mm 相当于 40 km。试采用适当的方法，绘制其国界线图形，并估算其国土面积。

表 4-5 某国上下国界线测量坐标

| x | 7.0 | 10.5 | 13.0 | 17.5 | 34.0 | 40.5 | 44.5 | 48.0 | 56.0 |
|---|---|---|---|---|---|---|---|---|---|
| y1 | 44 | 45 | 47 | 50 | 50 | 38 | 30 | 30 | 34 |
| y2 | 44 | 59 | 70 | 72 | 93 | 100 | 110 | 110 | 110 |
| x | 61.0 | 68.5 | 76.5 | 80.5 | 91.0 | 96.0 | 101.0 | 104.0 | 106.5 |
| y1 | 36 | 34 | 41 | 45 | 46 | 43 | 37 | 33 | 28 |
| y2 | 117 | 118 | 116 | 118 | 118 | 121 | 124 | 121 | 121 |
| x | 111.5 | 118.0 | 123.5 | 136.5 | 142.0 | 146.0 | 150.0 | 157.0 | 158.0 |
| y1 | 32 | 65 | 55 | 54 | 52 | 50 | 66 | 66 | 68 |
| y2 | 121 | 122 | 116 | 83 | 81 | 82 | 86 | 85 | 68 |

实验过程：

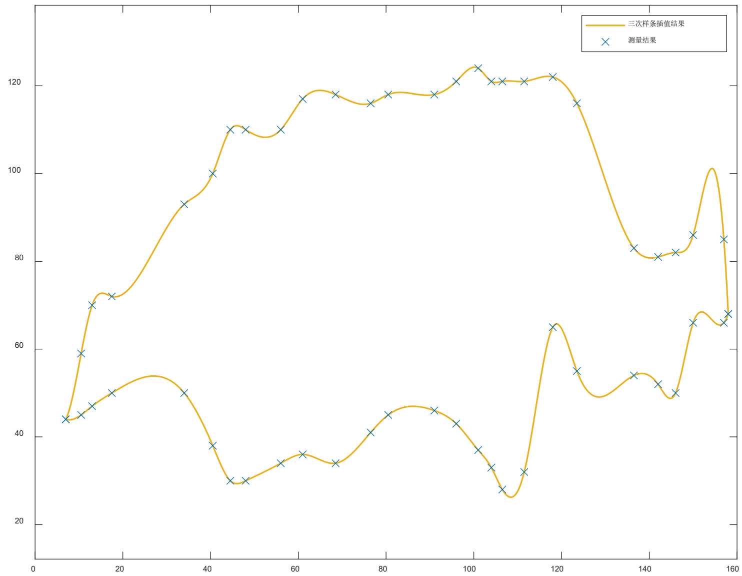
通过三次样条插值或者一维线性插值，再通过 trapz()分别对上边和下边作积分，然后求差来求面积。

Matlab 代码：

```
clear; clc;
% 原始数据
x = [7, 10.5, 13, 17.5, 34, 40.5, 44.5, 48, 56, 61, 68.5, 76.5, 80.5
, 91, 96, 101, 104, 106.5, 111.5, 118, 123.5, 136.5, 142, 146, 150,
157, 158];
y1 = [44, 45, 47, 50, 50, 38, 30, 30, 34, 36, 34, 41, 45, 46, 43, 37
, 33, 28, 32, 65, 55, 54, 52, 50, 66, 66, 68];
y2 = [44, 59, 70, 72, 93, 100, 110, 110, 110, 117, 118, 116, 118, 11
8, 121, 124, 121, 121, 121, 122, 116, 83, 81, 82, 86, 85, 68];

% 通过三次样条插值计算结果
xi = linspace(7, 158, 500);
y1_interp = interp1(x, y1, xi, 'spline'); %将此处的spline 改成lin-
ear 即可得到一维线性插值的结果
y2_interp = interp1(x, y2, xi, 'spline');
% 绘制插值结果
plot([xi, fliplr(xi)], [y1_interp, fliplr(y2_interp)], 'col-
or', '#EDB120', 'linewidth', 1.2); hold on;
% 绘制测量结果
plot([x, x], [y1, y2], 'x', 'color', '#0072BD', 'mark-
ersize', 8); hold on;
legend('三次样条插值结果','测量结果')
axis equal  %各坐标轴同比例
ratio = (40/18)^2;
S = (trapz(xi, y2_interp) - trapz(xi, y1_interp)) * ratio;
disp(['该国家的国土面积约为：', num2str(S), ' 平方千米'])
```

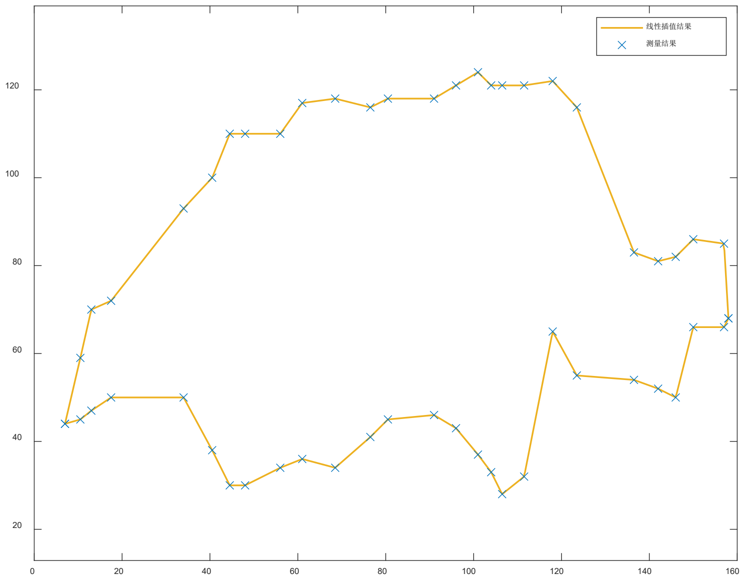
实验结果与分析：

1、通过三次样条插值的结果如下




用这种插值方法可以求得，该国家的国土面积约为：42467.3767 平方千米

2、一维线性插值的结果如下




用这种插值方法可以求得，该国家的国土面积约为：42412.6044 平方千米

可以看出两种插值方法求出的结果差距不是很大，对于边境线这种类型，一维线性插值可能更加准确。

2、已知飞机机翼断面轮廓线如下图，下轮廓线上部分数据如下表。根据加工需要，必须得到 x 坐标每改变 0.1 时 y 坐标的值，试选择几种适当方法完成所需数据，并画出相应的曲线，然后分析所用方法的优劣。

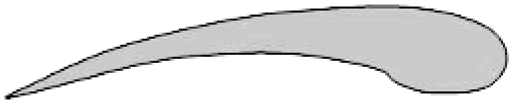


图 4-16 飞机机翼断面轮廓线

表 4-6 飞机机翼断面下轮廓线上部分数据

| x | 0 | 3 | 5 | 7 | 9 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 0 | 1.2 | 1.7 | 2.0 | 2.1 | 2.0 | 1.8 | 1.2 | 1.0 | 1.6 |

实验过程：

分别通过一维线性插值和三次样条插值求所需的数据。

Matlab 代码如下：

```
% 原始测量数据
x = [0, 3, 5, 7, 9, 11, 12, 13, 14, 15];
y = [0, 1.2, 1.7, 2, 2.1, 2, 1.8, 1.2, 1, 1.6];

xi = [0:0.1:15];
% 线性插值
y_linear = interp1(x, y, xi,  'linear');
figure(1)
plot(x, y,  'x', xi, y_linear)
legend('测量数据',  '线性插值')
axis equal  %各坐标轴同比例
% 导出 y_linear 数据到“线性插值.txt”
y_linear_Data = reshape([y_linear, zeros(1, 160 - length(y_lin-
ear))], [10, 16])';
save('线性插值.txt',  'y_linear_Data',  '-ascii')

% 三次样条插值
y_spline = interp1(x, y, xi, 'spline');
figure(2)
plot(x, y, 'x', xi, y_spline)
```

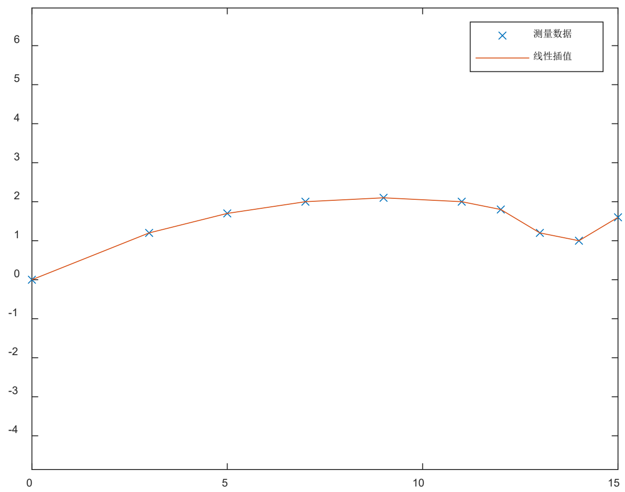
```
legend('测量数据','三次样条插值')
axis equal  %各坐标轴同比例
% 导出 y_spline 数据到“三次样条插值.txt”
y_spline_Data = reshape([y_linear, zeros(1, 160 - length(y_lin-
ear))], [10, 16])';
save('三次样条插值.txt',  'y_spline_Data',  '-ascii')
```

实验结果与结论：

1、一维线性插值结果：

需要补充的数据如下：

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.00 | 0.04 | 0.08 | 0.12 | 0.16 | 0.20 | 0.24 | 0.28 | 0.32 | 0.36 |
| 0.40 | 0.44 | 0.48 | 0.52 | 0.56 | 0.60 | 0.64 | 0.68 | 0.72 | 0.76 |
| 0.80 | 0.84 | 0.88 | 0.92 | 0.96 | 1.00 | 1.04 | 1.08 | 1.12 | 1.16 |
| 1.20 | 1.23 | 1.25 | 1.28 | 1.30 | 1.33 | 1.35 | 1.38 | 1.40 | 1.43 |
| 1.45 | 1.48 | 1.50 | 1.53 | 1.55 | 1.58 | 1.60 | 1.63 | 1.65 | 1.68 |
| 1.70 | 1.72 | 1.73 | 1.75 | 1.76 | 1.78 | 1.79 | 1.81 | 1.82 | 1.84 |
| 1.85 | 1.87 | 1.88 | 1.90 | 1.91 | 1.93 | 1.94 | 1.96 | 1.97 | 1.99 |
| 2.00 | 2.01 | 2.01 | 2.02 | 2.02 | 2.03 | 2.03 | 2.04 | 2.04 | 2.05 |
| 2.05 | 2.06 | 2.06 | 2.07 | 2.07 | 2.08 | 2.08 | 2.09 | 2.09 | 2.10 |
| 2.10 | 2.10 | 2.09 | 2.09 | 2.08 | 2.08 | 2.07 | 2.07 | 2.06 | 2.06 |
| 2.05 | 2.05 | 2.04 | 2.04 | 2.03 | 2.03 | 2.02 | 2.02 | 2.01 | 2.01 |
| 2.00 | 1.98 | 1.96 | 1.94 | 1.92 | 1.90 | 1.88 | 1.86 | 1.84 | 1.82 |
| 1.80 | 1.74 | 1.68 | 1.62 | 1.56 | 1.50 | 1.44 | 1.38 | 1.32 | 1.26 |
| 1.20 | 1.18 | 1.16 | 1.14 | 1.12 | 1.10 | 1.08 | 1.06 | 1.04 | 1.02 |
| 1.00 | 1.06 | 1.12 | 1.18 | 1.24 | 1.30 | 1.36 | 1.42 | 1.48 | 1.54 |
| 1.60 | | | | | | | | | |

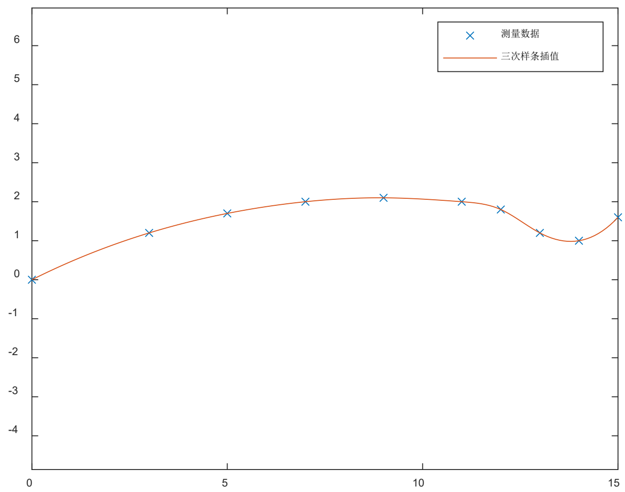



通过一维线性插值补充数据之后的曲线图形

2、三次样条插值结果：

需要补充的数据如下：

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.00 | 0.05 | 0.10 | 0.15 | 0.20 | 0.24 | 0.29 | 0.33 | 0.38 | 0.42 |
| 0.47 | 0.51 | 0.55 | 0.59 | 0.63 | 0.67 | 0.71 | 0.75 | 0.79 | 0.83 |
| 0.86 | 0.90 | 0.94 | 0.97 | 1.01 | 1.04 | 1.07 | 1.11 | 1.14 | 1.17 |
| 1.20 | 1.23 | 1.26 | 1.29 | 1.32 | 1.35 | 1.37 | 1.40 | 1.43 | 1.45 |
| 1.48 | 1.50 | 1.53 | 1.55 | 1.57 | 1.59 | 1.62 | 1.64 | 1.66 | 1.68 |
| 1.70 | 1.72 | 1.74 | 1.76 | 1.78 | 1.79 | 1.81 | 1.83 | 1.84 | 1.86 |
| 1.87 | 1.89 | 1.90 | 1.92 | 1.93 | 1.94 | 1.96 | 1.97 | 1.98 | 1.99 |
| 2.00 | 2.01 | 2.02 | 2.03 | 2.04 | 2.05 | 2.05 | 2.06 | 2.07 | 2.07 |
| 2.08 | 2.08 | 2.09 | 2.09 | 2.09 | 2.10 | 2.10 | 2.10 | 2.10 | 2.10 |
| 2.10 | 2.10 | 2.10 | 2.10 | 2.09 | 2.09 | 2.09 | 2.08 | 2.08 | 2.07 |
| 2.07 | 2.06 | 2.06 | 2.05 | 2.04 | 2.04 | 2.03 | 2.02 | 2.01 | 2.01 |
| 2.00 | 1.99 | 1.98 | 1.97 | 1.96 | 1.95 | 1.93 | 1.90 | 1.88 | 1.84 |
| 1.80 | 1.75 | 1.70 | 1.64 | 1.57 | 1.51 | 1.44 | 1.38 | 1.32 | 1.26 |
| 1.20 | 1.15 | 1.11 | 1.07 | 1.04 | 1.01 | 1.00 | 0.99 | 0.98 | 0.99 |
| 1.00 | 1.02 | 1.05 | 1.09 | 1.13 | 1.19 | 1.25 | 1.32 | 1.41 | 1.50 |
| 1.60 | | | | | | | | | |




通过三次样条插值补充数据之后的曲线图形

通过上面两个曲线图形对比，可以看出三次样条插值的效果更加好，曲面整体更加平滑。尽管一维线性插值的结果对于实际加工来说可能更加简单，但是平滑度欠佳会对其效果产生一定影响。

# 实 验 八　　微分方程实验

1.用 dsolve 函数求解下列微分方程

（1）$y'' - 2y' + 5y = e^x \sin 2x$，（2）$\begin{cases} y''(x) = y'(x) + 2y(x) \\ y(0) = 1,\ y'(0) = 0 \end{cases}$

实验过程：

Matlab 代码如下

```
%% 第 1 小问
y = dsolve('D2y-2*Dy+5*y=exp(x)*sin(2*x)', 'x')
%% 第 2 小问
y=dsolve('D2y=Dy+2*y','y(0)=1','Dy(0)=0','x')
```

实验结果及分析：

（1）运行结果为：

$$y = C1 * \cos(2*x) * \exp(x) - \frac{\sin(6*x) * \exp(x)}{32} - \cos(2*x) * \exp(x) * \left(\frac{\frac{x}{4} - \sin(4*x)}{16}\right) - \frac{\sin(2*x) * \exp(x)}{32} - C2 * \sin(2*x) * \exp(x)$$

也就是下图中的表达式

$$C_1 \cos(2x)\, e^x - \frac{\sin(6x)\, e^x}{32} - \cos(2x)\, e^x \left(\frac{x}{4} - \frac{\sin(4x)}{16}\right) - \frac{\sin(2x)\, e^x}{32} - C_2 \sin(2x)\, e^x$$

第 1 小问-运行结果

（2）运行结果为：

$$y = \frac{\exp(-x) * (\exp(3*x) + 2)}{3}$$

也就是下图中的表达式

$$\frac{e^{-x}\left(e^{3x} + 2\right)}{3}$$

第 2 小问-运行结果

2.用 ode 函数求解微分方程$\begin{cases} x' = x(1-x-y-6z) \\ y' = y(1.5x-y-z) \\ z' = z(-1+3x+0.5) \end{cases}$，并讨论解的变化情况。初值及求解区间如下：

（1）$x(0) = 0.12, y(0) = 0.003, z(0) = 0.01$，$t \in [0,30]$

（2）$x(0) = 0.01, y(0) = 0.00001, z(0) = 0.001$，$t \in [0,133]$

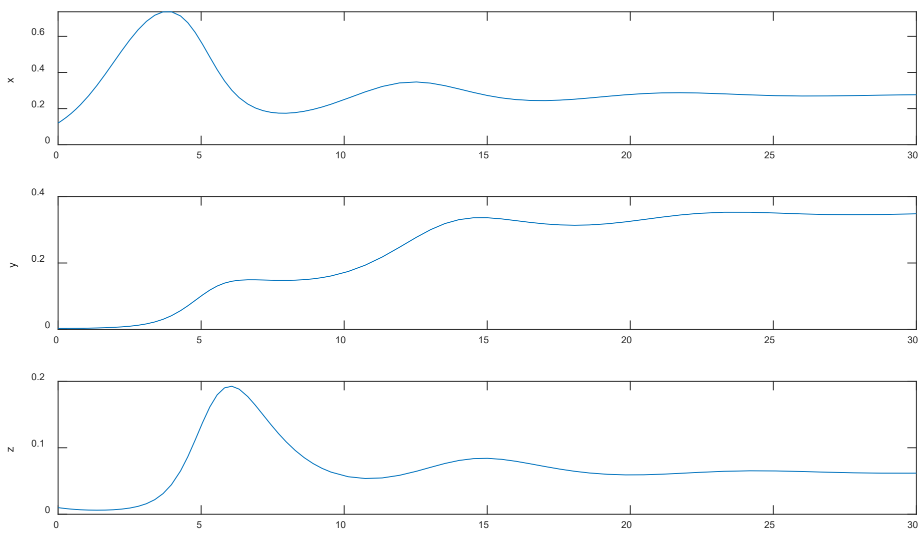
实验过程：

Matlab 代码如下：

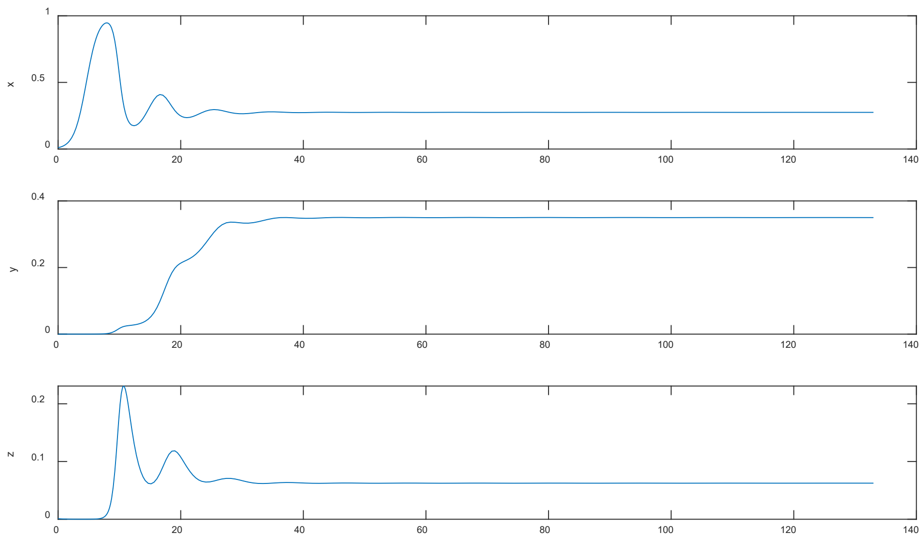
```
function dy = odefun_q2(t, y)
    dy = zeros(3, 1);
    dy(1) = y(1) * (1 - y(1) - y(2) - 6 * y(3));
    dy(2) = y(2) * (1.5 * y(1) - y(2) - y(3));
    dy(3) = y(3) * (-1 + 3 * y(1) + 0.5 * y(2));
end
```

```
%% 第 1 小问
[t,y]=ode15s('odefun_q2',[0,30],[0.12,0.003,0.01]);
subplot(3,1,1)
plot(t,y(:,1))
ylabel('x')

subplot(3,1,2)
plot(t,y(:,2))
ylabel('y')

subplot(3,1,3)
plot(t,y(:,3))
ylabel('z')

%% 第 2 小问
[t,y]=ode15s('odefun_q2',[0,133],[0.01,0.00001,0.001]);
subplot(3,1,1)
plot(t,y(:,1))
ylabel('x')

subplot(3,1,2)
plot(t,y(:,2))
ylabel('y')

subplot(3,1,3)
plot(t,y(:,3))
ylabel('z')
```

实验结果及分析：



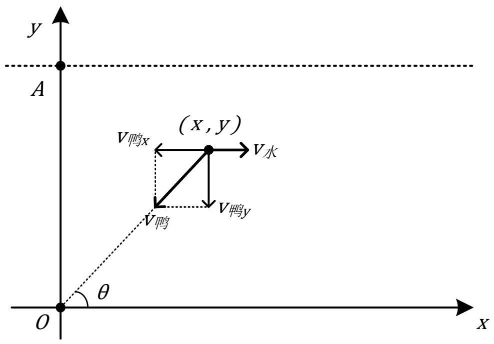

第一小问结果




第二小问结果

3. 设河边点$O$的正对岸点为$A$：河宽$OA = h$，两岸为平行直线，水流速度为0.5米每分。有一只鸭子从$A$点游向$O$点，设鸭子在静水中的游动速度为1米每分，且鸭子游动方向始终朝着点$O$，求鸭子游过的迹线方程。用 Matlab 求解，并作出轨迹图。

实验过程：




如上图所示，对于任意时刻，鸭子的位置$(x,y)$，鸭子的速度分别为$v_{水}$和$v_{鸭}$，将$v_{鸭}$作正交分解，分别为$v_{鸭x}$，$v_{鸭y}$。其中：

$$v_{鸭x} = v_{鸭} \cdot cos\,\theta = v_{鸭} \cdot \frac{x}{\sqrt{x^2+y^2}}$$

$$v_{鸭y} = v_{鸭} \cdot sin\,\theta = v_{鸭} \cdot \frac{y}{\sqrt{x^2+y^2}}$$

那么，鸭子的轨迹方程为：

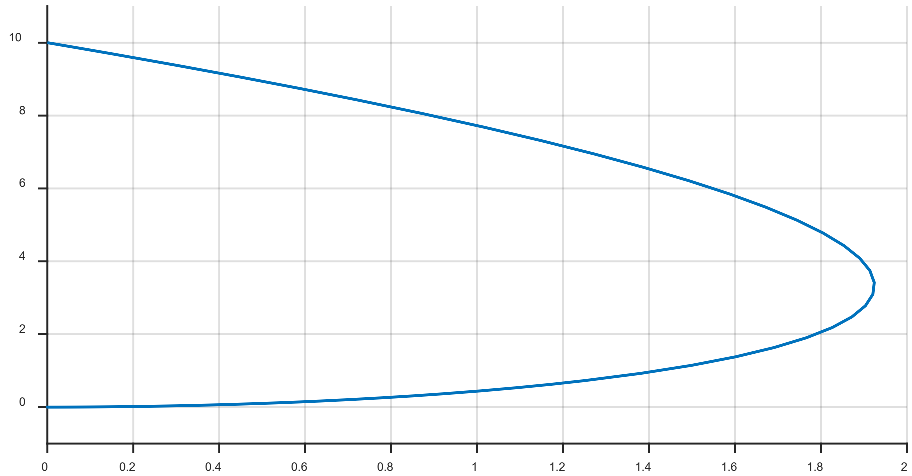
$$\begin{cases} x' = 0.5 - \dfrac{x}{\sqrt{x^2+y^2}} \\ y' = -\dfrac{y}{\sqrt{x^2+y^2}} \end{cases}$$

求解 Matlab 代码如下：（不妨设$h = 10$）

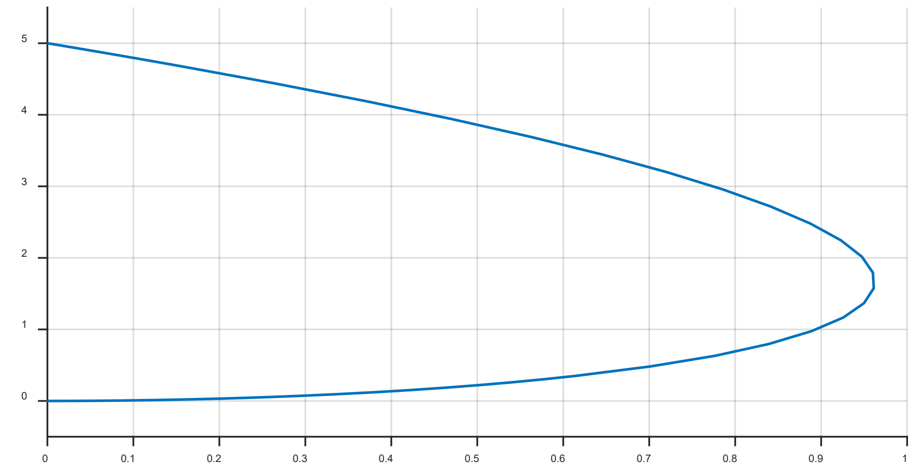
```
function dy = odefun_q3(t, y)

    dy = zeros(2, 1);
    dy(1) = 0.5 - y(1) / sqrt(y(1)^2 + y(2)^2);
    dy(2) = - y(2) / sqrt(y(1)^2 + y(2)^2);

end
```

```
% 求解路径
[t, y] = ode45('odefun_q3', [0, 15], [0, 10]);
% 绘制路径
plot(y(:, 1), y(:, 2));hold on;
```

实验结果及分析：




取$h=10$时的结果




取$h=5$时的结果

可以看出，轨迹形状与$h$的取值无关。无论$h$取什么值，最终都能到达原点，也就是题目中的$O$点。

# 实 验 九　　蒙特卡洛模拟实验

1、一袋中装有 4个白球、8个黑球，A、B、C三人蒙住眼睛依次轮流摸球。先得白球者获胜，求三人获胜的机会比。

实验过程：

首先定义一个函数，其返回值为获胜的频率，可以模拟有放回与无放回的情况。返回ABC的获胜的频率

```
function [pa, pb, pc] = q1_fun(n, type)
    % type=='back' -> 有放回
    % type=='noback' -> 无放回
    maxRange = [12, 12, 12]; %默认为有放回
    if strcmp(type, 'noback')
        maxRange = [12, 11, 10];    % 修改抽球范围
    end
    pa = 0; pb = 0; pc = 0;
    fa = 0; fb = 0; fc = 0;
    total_win = 0;
    for i = 1:n
        Aball = randi([1, maxRange(1)]); % 1-4 为白球，其余为黑球
        if Aball <= 4% A 抽中白球
            fa = fa + 1;
            total_win = total_win + 1;
        else
            Bball = randi([1, maxRange(2)]); % 1-4 为白球，其余为黑球
            if Bball <= 4% B 抽中白球
                fb = fb + 1;
                total_win = total_win + 1;
            else
                Cball = randi([1, maxRange(3)]); % 1-4 为白球，其余为黑球
                if Cball <= 4% C 抽中白球
                    fc = fc + 1;
                    total_win = total_win + 1;
                end
            end
        end
    end
    pa = fa / total_win; % A 抽中白球的概率
    pb = fb / total_win; % B 抽中白球的概率
    pc = fc / total_win; % C 抽中白球的概率
end
```

之后调用上述函数进行不同次数的蒙特卡洛实验，并将结果保存到txt文件。

```
%进行 10^i 次的蒙特卡洛实验
for i = 1:7
    % 有放回的情况
    [a_back(i), b_back(i), c_back(i)] = q1_fun(10^i, 'back');
    % 无放回的情况
    [a_noback(i), b_noback(i), c_noback(i)] = q1_fun(10^i, 'noback');
end

% 将实验结果输出到 txt
outputData = [a_back; b_back; c_back; a_noback; b_noback; c_noback]
save('蒙特卡洛实验第一题结果.txt', 'outputData', '-ascii')
```

实验结果与分析：

| | Monte Carlo 试验次数 | 10^1 | 10^2 | 10^3 | 10^4 | 10^5 | 10^6 | 10^7 |
|---|---|---|---|---|---|---|---|---|
| 有放回 | A 的胜率 | 50.00% | 45.95% | 47.69% | 47.70% | 47.21% | 47.29% | 47.36% |
| | B 的胜率 | 25.00% | 25.68% | 29.73% | 31.58% | 31.82% | 31.60% | 31.58% |
| | C 的胜率 | 25.00% | 28.38% | 22.58% | 20.72% | 20.97% | 21.12% | 21.06% |
| 无放回 | Monte Carlo 试验次数 | 10^1 | 10^2 | 10^3 | 10^4 | 10^5 | 10^6 | 10^7 |
| | A 的胜率 | 0.00% | 38.89% | 41.41% | 44.54% | 44.92% | 44.78% | 44.75% |
| | B 的胜率 | 57.14% | 31.94% | 34.24% | 32.44% | 32.11% | 32.48% | 32.50% |
| | C 的胜率 | 42.86% | 29.17% | 24.36% | 23.02% | 22.96% | 22.74% | 22.75% |

可以看出，无论是有放回还是无放回，胜率均为A最高，B次之，C最低。从直观上也能得到这样的结果，因为C要想得到抽球的机会，必须是A，B都抽不到白球。

如果是有放回的情况，他们获胜的机会比$A:B:C \approx 47.3\%:31.6\%:21.1\%$；

如果是无放回的情况，他们获胜的机会比$A:B:C \approx 44.75\%:32.5\%:22.75\%$；

2、一袋中装有 4个白球、8个黑球，甲同乙打赌他能在摸出的7个球中含有3个白球。求两人获胜的机会比。

实验过程：

首先定义一个函数q2_fun用于模拟抽球过程。其中会在1-12中随机产生7个数字，规定1-4为白球，5-12为黑球，根据产生的数字即可模拟抽取的结果。

```
function [pa, pb, Awin, Bwin] = q2_fun(n)
    Awin = 0;
    Bwin = 0;
    for k = 1:n
        counter = 0;
        balls = randperm(12, 7); % 随机抽取7个球
        for i = 1:7
            if balls(i) <= 4
                counter = counter + 1; %计算白球个数
            end
        end
        %判断 A 赢还是 B 赢
        if counter >= 3
            Awin = Awin + 1;
        else
            Bwin = Bwin + 1;
        end
    end
    %用频率近似概率
    pa = Awin / n;
    pb = Bwin / n;
end
```

调用上述函数，模拟抽球过程

```
clear all; clc;
%进行 10^i 次的蒙特卡洛实验
for i = 1:7
    [pa(i), pb(i), Awin(i), Bwin(i)] = q2_fun(10^i);
end

% 将实验结果输出到 txt
outputData = [pa; pb; Awin; Bwin]
save('蒙特卡洛实验第二题结果.txt', 'outputData', '-ascii')
```

实验结果与分析：

| Monte Carlo 试验次数 | 10^1 | 10^2 | 10^3 | 10^4 | 10^5 | 10^6 | 10^7 |
|---|---|---|---|---|---|---|---|
| A 胜利次数 | 6 | 42 | 420 | 4210 | 42484 | 424254 | 4243321 |
| A 胜利频率 | 60.00% | 42.00% | 42.00% | 42.10% | 42.48% | 42.43% | 42.43% |
| B 胜利次数 | 4 | 58 | 580 | 5790 | 57516 | 575746 | 5756679 |
| B 胜利频率 | 40.00% | 58.00% | 58.00% | 57.90% | 57.52% | 57.57% | 57.57% |

从上述结果可以看出随之试验次数增大，甲的胜利频率趋近于42.42%，而乙的趋近于57.57%，也就是说乙的胜利几率大一点，甲做这个赌注是不值得的，他更容易输。

3、用Monte Carlo法求解全局最优化及约束优化问题并通过图形作出评论。求下列函数的最大值：

(1) $f(x) = (1 - x^2)\, sin(3x), -2\pi < x < 2\pi$

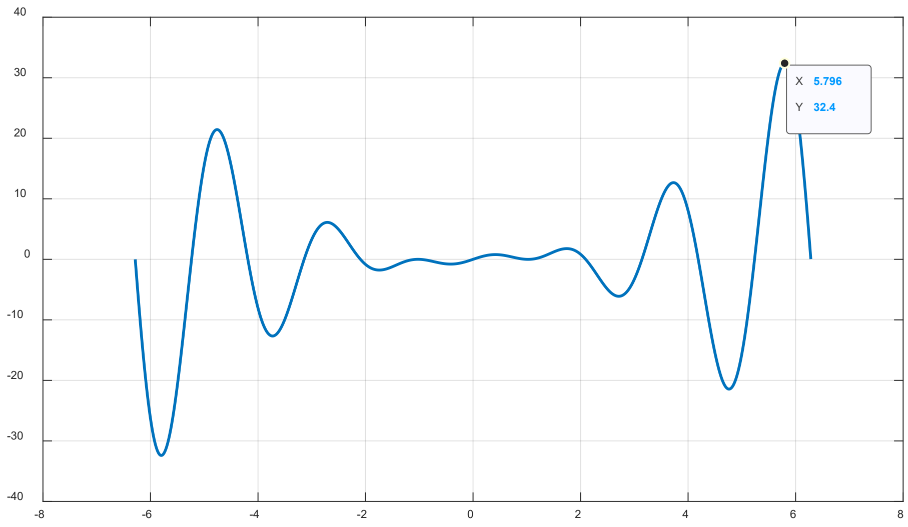
实验过程：

```
function maxVal = q3_1(n)
    % 通过 Monte Carlo 法求最大值
    x = unifrnd(-2 * pi, 2 * pi, 1, n);
    f_x = (1 - x.^2) .* sin(3 * x);
    maxVal = max(f_x);
    % 绘制图形
    draw_x = -2 * pi:0.001:2 * pi;
    draw_y = (1 - draw_x.^2) .* sin(3 * draw_x);
    plot(draw_x, draw_y);
end
```

实验结果与分析：

通过$10^7$次的Monte Carlo实验，得到最大值为$\max = 32.4006$，

通过下面图形可以看出，最大值为32.4，两者是比较接近的。




$$f(x) = (1-x^2)\,sin(3x), -2\pi < x < 2\pi \text{ 的图形}$$

(2) $\max f(x) = x_1x_2x_3$

$$s.t,\begin{cases} -x_1 + 2x_2 + 2x_3 \geq 0 \\ x_1 + x_2 + 2x_3 \leq 72 \\ 10 \leq x_2 \leq 20 \\ x_1 - x_2 = 10 \end{cases}$$

实验过程：

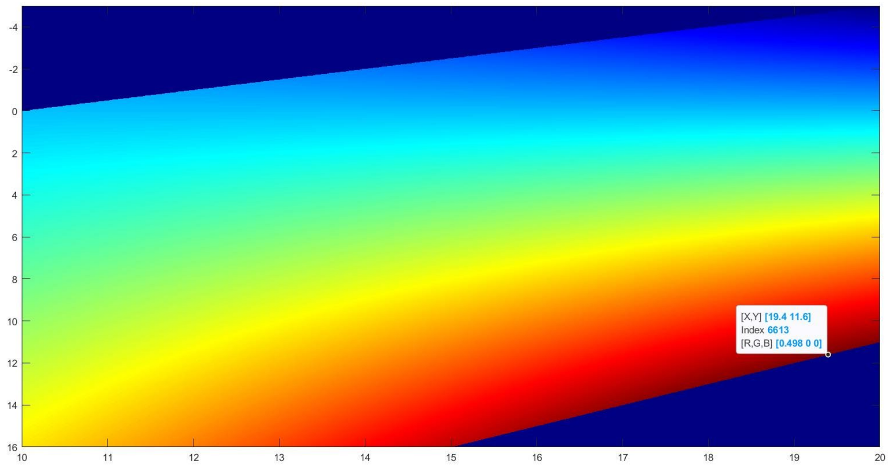
```
function maxVal = q3_2(n)
    % 通过 Monte Carlo 法求最大值
    fx = [];
    x2 = unifrnd(10, 20, 1, n);
    x1 = x2 + 10;
    x3 = unifrnd(-20, 20, 1, n);
    for i = 1:n
        if -x1(i) + 2 * x2(i) + 2 * x3(i) >= 0 &&
           x1(i) + x2(i) + 2 * x3(i) <= 72
            fx = [fx, x1(i) * x2(i) * x3(i)];
        end
    end
    maxVal = max(fx);
    % 通过 imagesc 绘制图形
    x2 = 10:0.001:20;
    x3 = -5:0.001:16;
```

```
        [X, Y] = meshgrid(x2, x3);
        err1 = X + 2 * Y < 10;
        err2 = 2 * X + 2 * Y > 62;
        X(err1) = nan;
        Y(err2) = nan;
        Z = X .* Y .* (X + 10);
        colormap('jet')
        imagesc(x2, x3, Z); hold on
    end
```

实验结果与分析：

通过$10^7$次的Monte Carlo实验，得到最大值为$\max = 6616.2$，

通过下面图形可以看出，最大值为6613，两者是比较接近的。




$f(x) = x_1x_2x_3$满足约束条件的取值的色图，其中横轴为$x_2$,纵轴为$x_3$

(3) $f(x,y) = (x^2 + 2y^2 + xy)e^{-x^2-y^2}$ , $|x| < 1.5$ , $|y| < 1.5$

实验过程：

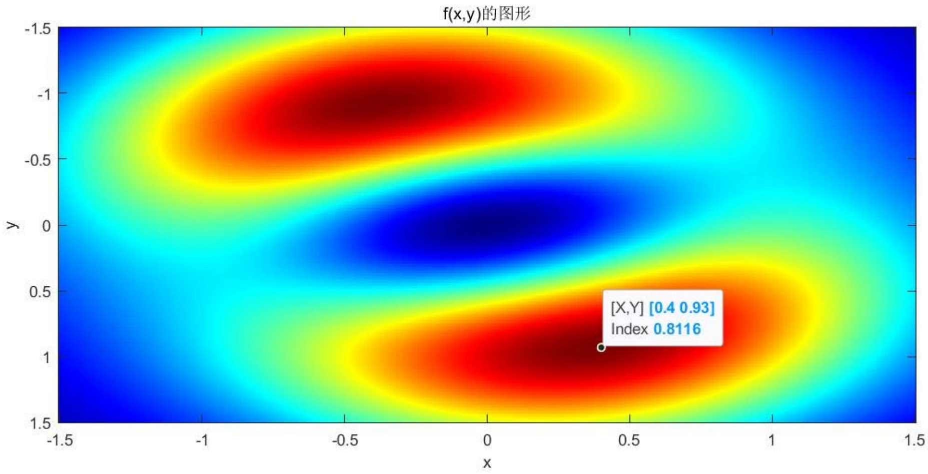
```
    function maxVal = q3_3(n)
        % 通过 Monte Carlo 法求 f(x,y)最大值
        x = unifrnd(-1.5, 1.5, 1, n);
        y = unifrnd(-1.5, 1.5, 1, n);
        f_xy = (x.^2 + 2 * y.^2 + x .* y) .* exp(-x.^2 - y.^2);
        maxVal = max(f_xy);

        % 通过 imagesc 绘制 f(x,y)的图形
        all_x = -1.5:0.01:1.5;
        all_y = -1.5:0.01:1.5;
        [X, Y] = meshgrid(all_x, all_y);
        Z = (X.^2 + 2 * Y.^2 + X .* Y) .* exp(-X.^2 - Y.^2);
```

```
        % colormap('jet')
        imagesc(all_x, all_y, Z); hold on
        % waterfall(all_x, all_y, Z); hold on
    end
```

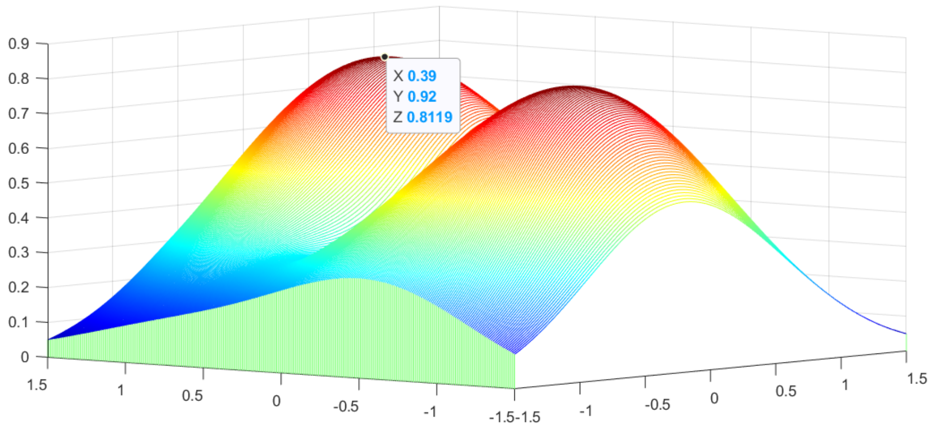
实验结果与分析：

通过$10^7$次的Monte Carlo实验，得到最大值为max $= 0.8119$

通过下面图形可以看出，最大值为 0.8119，两者是一致的。




$f(x,y) = (x^2 + 2y^2 + xy)e^{-x^2-y^2}$ 的颜色图




$f(x,y) = (x^2 + 2y^2 + xy)e^{-x^2-y^2}$ 的*waterfall*图