云南大学数学与统计学实验教学中心

实验报告

课程名称：数值计算方法实验	**学期**：2015—2016 学年第一学期	**成绩**：
指导教师： 李耀堂	**学生姓名**：金洋	**学生学号**：20131910023
实验名称： **最小二乘法拟合**		
实验编号：No.8	**实验日期**： **2015.12.25**	**实验学时**： **3**
学院： 数学与统计学院	**专业**： **信息与计算科学**	**年级**：2013 级

一、实验目的

掌握最小二乘法的原理及方法；

练习利用最小二乘法拟合给定的实验数据；

二、实验内容

函数 $y = f(x)$ 的一张函数表如下：

x	-1.00	-0.75	-0.50	-0.25	0	0.25
y	-0.2209	0.3295	0.8826	1.4392	2.003	2.5645

试球出用最下二乘法拟合的一次多项式。

三、实验环境

PC 计算机；

MATLAB R2014a；

四、实验方法：最小二乘法拟合

方法简述：

(1)最小二乘法拟合：

设给定实验数据如下：

x	x_1	x_2	$\cdots$	x_m
$f(x)$	$f(x_1)$	$f(x_2)$	$\cdots$	$f(x_m)$

如果用 n 次多项式 $\varphi_n(x) = \sum_{j=0}^{n} a_j x^j$ 插值(n+1<m)，即

$$\begin{cases} a_0 + a_1x_1 + \cdots + a_nx_1{}^n = y_1 \\ a_0 + a_1x_2 + \cdots + a_nx_2{}^n = y_2 \\ \cdots \\ a_0 + a_1x_m + \cdots + a_nx_m{}^n = y_m \end{cases}$$

由于 n+1<m，即方程个数多于未知数个数，一般来说，这是一个矛盾方程组，用一般解联立方程组的方法无法求解。

因此，不能要求 $\varphi_n(x_i) = f(x_i)(i = 1,2,\cdots,m)$ 精确成立，而仅仅要求多项式尽可能接近给定的数据，允许每个等式可以稍有偏差。

$$\begin{cases} a_0 + a_1x_1 + \cdots + a_nx_1{}^n - y_1 = \delta_1 \\ a_0 + a_1x_2 + \cdots + a_nx_2{}^n - y_2 = \delta_2 \\ \cdots \\ a_0 + a_1x_m + \cdots + a_nx_m{}^n - y_m = \delta_m \end{cases}$$

希望偏差向量 $\delta = (\delta_1, \delta_2, \cdots, \delta_m)^T$ 的模达到最小值，计算时采用使

$\|\delta\|_2^2 = \sum_{i=1}^{m} \delta_i{}^2 = \sum_{i=1}^{m} [\varphi_n(x_i) - f(x_i)]^2 = \sum_{i=1}^{m} [y_i - \sum_{j=0}^{n} a_i x_i{}^j]^2$ =min。

记 $x_i{}^j$ 为 x_{ij} (i=1, ⋯, m; j=0, ⋯, n),则 $\varphi(x_i)$ 可以表示为 $\varphi(x_i) = \sum_{j=0}^{n} a_j x_{ij}$；

故令 $X = (x_{ij}) \in R^{m*(n+1)}$, $y = (y_1, \cdots, y_m)^T$, $a = (a_0, \cdots, a_n)^T$

$$\therefore \sum_{i=1}^{m} [y_i - \sum_{j=0}^{n} a_i x_i{}^j]^2 = \|y - Xa\|_2^2$$

于是问题归结为求向量 $a = (a_0, \cdots, a_n)^T$，s.t. $\|y - Xa\|_2^2 = \min_{a \in R^{n+1}} \|y - Xa\|_2^2$,这就是矛盾方程组 Xa=y， $X = (x_{ij}) \in R^{m*(n+1)}$ 的最小二乘解。可以用 5.1 节的方法求解。

$X = \begin{pmatrix} 1 & x_1 & x_1{}^2 & \cdots & x_1{}^n \\ 1 & x_2 & x_2{}^2 & \cdots & x_2{}^n \\ \vdots & & & & \\ 1 & x_m & x_m{}^2 & \cdots & x_m{}^n \end{pmatrix}$,x 间各不相同，显然 X 是列满秩的，则正规方程

组 $X^T Xa = X^T y$ 的唯一解是 $Xa = y$ 的最小二乘解。

令 B= $X^T X = \begin{pmatrix} \sum 1 & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \vdots & & & & \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{pmatrix}$,C= $X^T y = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^n y_i \end{pmatrix}$

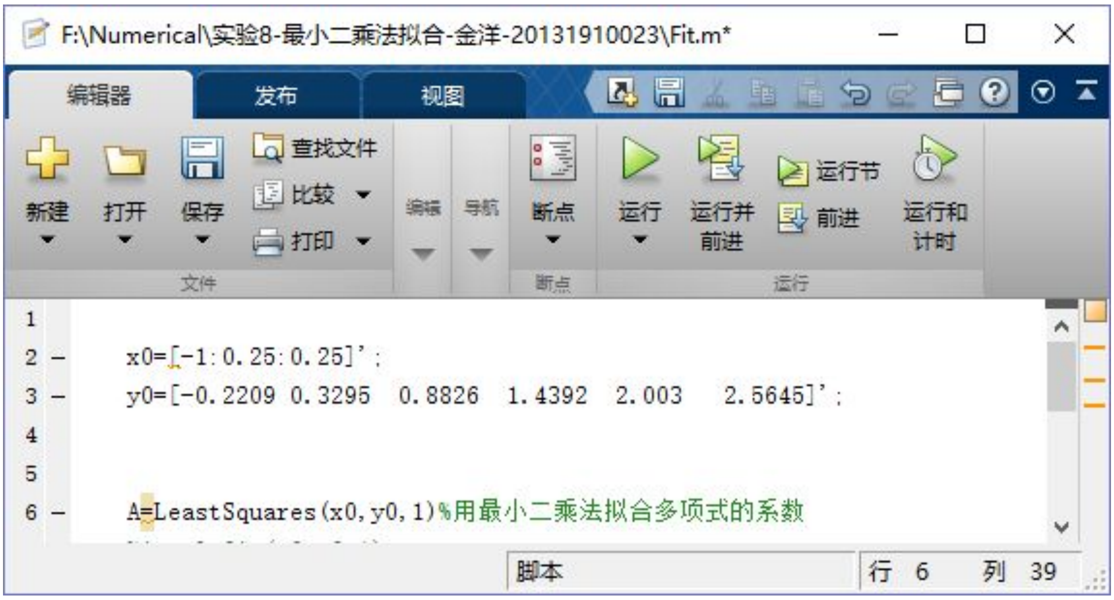
$\therefore a = B^{-1}C$

五、实验过程

1 实验步骤

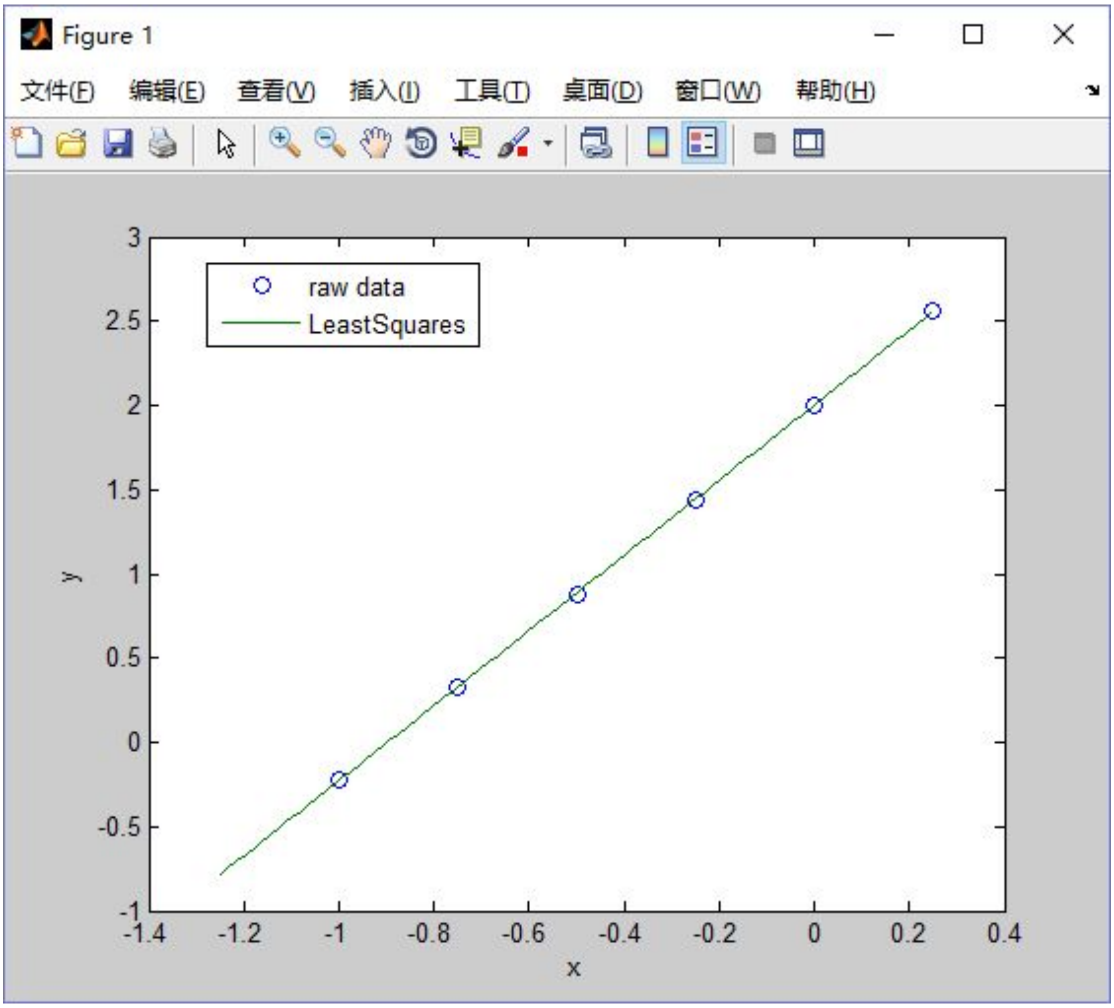
①编程：用 MATLAB 语言编出源程序。
②开机,键入所编程序源代码.
③调试程序, 修改错误至能正确运行.
④运行程序并输出计算结果.
⑤尝试改进

输入已知点的数据如下：



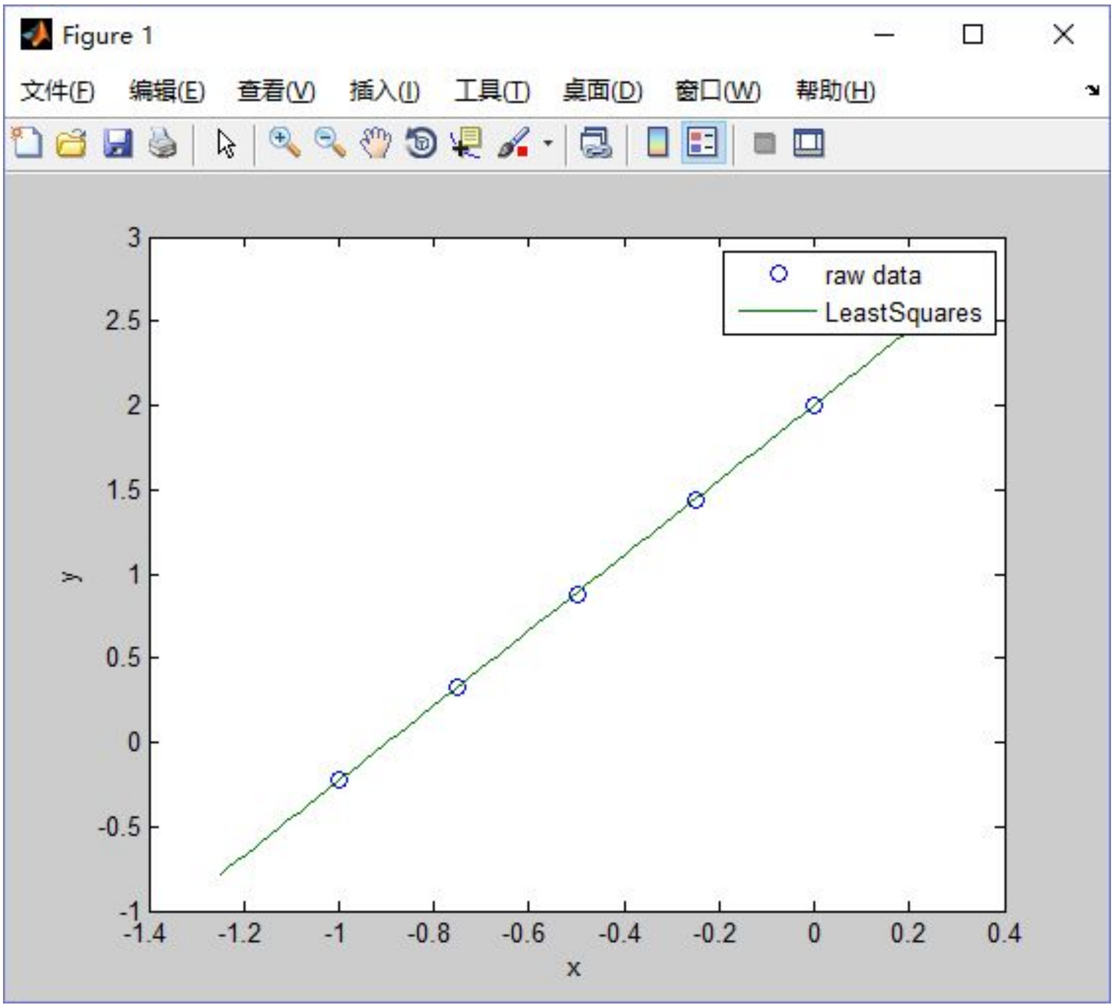
运行结果

①采用如上四中方法运行结果：



即 y=2.2290x+2.0022；

②使用 MATLAB 内置最小二乘法拟合运行结果如下：



即 y=2.2290x+2.0022，结果与①相同；

2 关键代码及其解释

①

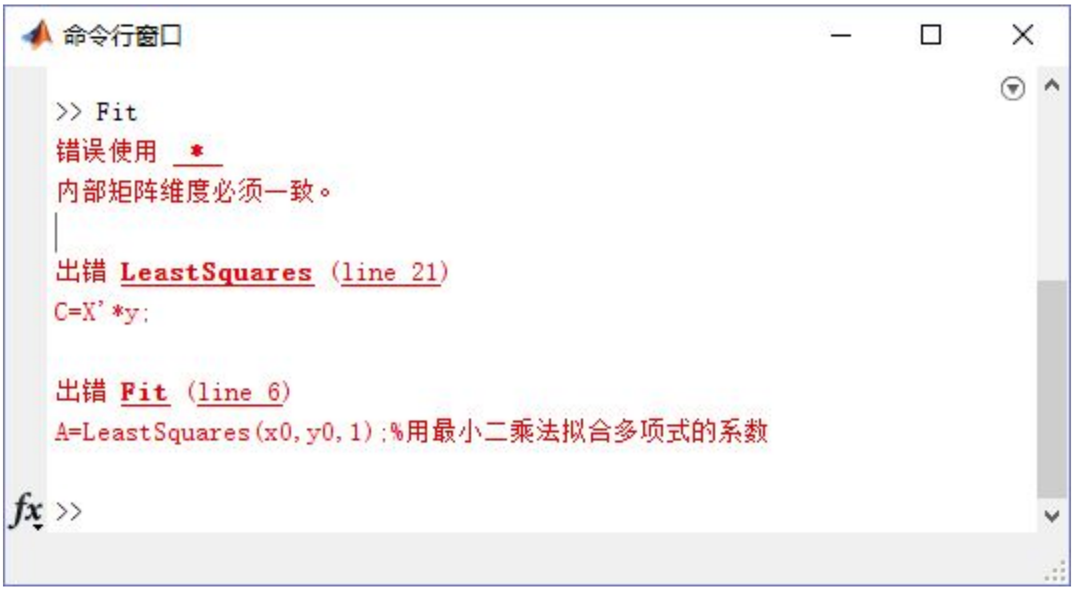
```
%A=LeastSquares(x0,y0,1)%用最小二乘法拟合多项式的系数
A=polyfit(x0,y0,1)

x=[-1.25:0.01:0.25];
y=polyval(A,x);%拟合函数函数值

plot(x0,y0,'o',x,y);
```

②

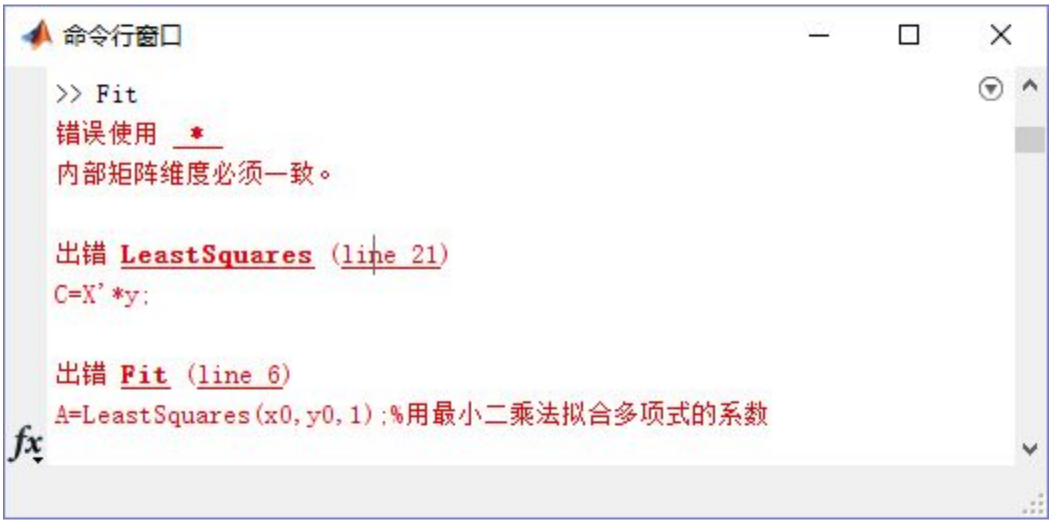
```
for i=1:n
   X(:,i+1)=X(:,i).*x;
end

B=X'*X;
C=X'*y;

%求解法方程 X'XA=X'y
A=B\C;
A=A(n+1:-1:1)';
```

3 调试过程

①



②

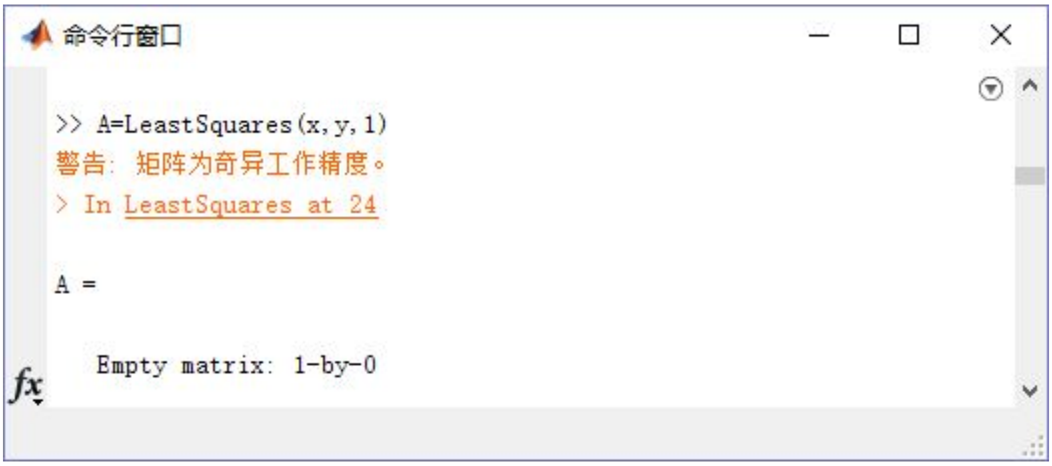


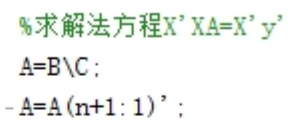
③

④生成 **X** 矩阵时出错

```
for i=1:n
    X(i+1)=X(i,:).*x';
end
```

⑤将 **A** 中系数需从高次到低次排，之中发生错误





六、实验总结

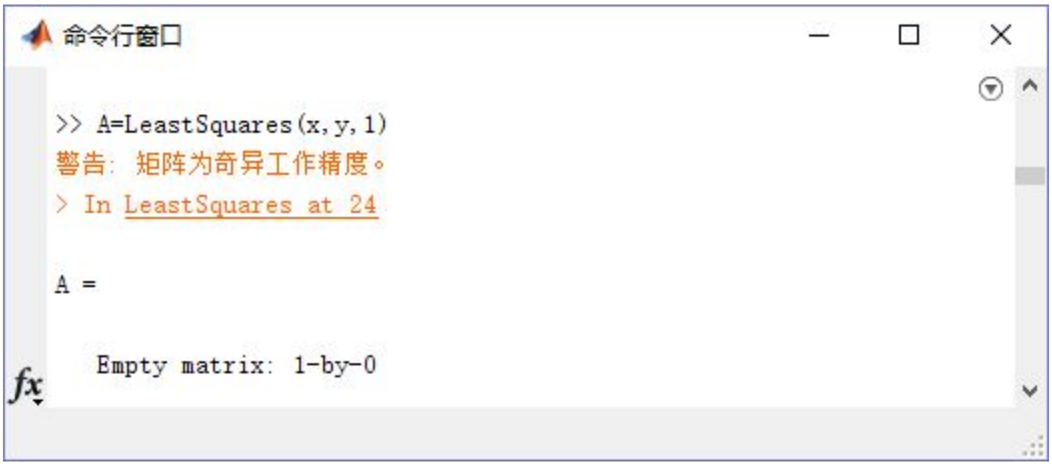
1. 遇到的问题及解决过程

①多次提示矩阵维度错误，后发现在生成 X 矩阵时有问题：

```
for i=1:n
    X(i+1)=X(i,:).*x;
End
```

应改为 `X(:,i+1)=X(:,i).*x;`

② 解决①后，发现产生的图像为一条直线，且提示



后发现在：

```
%求解法方程 X'XA=X'y
A=B\C;
A=A(n+1:1)';
```

应该改为 `A=A(n+1:-1: 1)';`

2．产生的错误及原因分析

①MATLAB 计算中，会检验矩阵的维度来确定运算是否合理，故在前期要仔细检测输入数据的正确性；

②提取向量 A 的全部元素时可以使用 A(1:1:n)或 A(1:n),但是要逆向提取矩阵的元素时应该用 A(n:-1:1), 不能使用 A(n:1)；

3．体会和收获。

①当已知点个数多于拟合曲线次数时，联立时会发现方程个数多于未知数个数，一般来说，这是一个矛盾方程组，用一般解联立方程组的方法无法求解。此时如果用高次插值多项式来代替原本设定的函数，反而不合适。

因此，我们不要求 $\varphi_n(x_i)=f(x_i)(i=1,2,\cdots,m)$ 精确成立，而仅仅要求多项式尽可能接近给定的数据，允许每个等式可以稍有偏差。

②拟合的程序实现上，可以将推理的成果转化为代码，也可以直接使用 MATLAB 的内置函数 polyfit()来实现。

③曲线拟合使偏差最小到最后转化为求解矛盾方程组的问题，而后者是我们已经研究过的。故转化思想需要我们在学习中着重培养。

七、程序源代码：

①主函数

其中调用拟合函数时也可以直接使用 MATLAB 的内置函数 polyfit()，经过试验，两者功能一致

Fit.m

```
x0=[-1:0.25:0.25]';
y0=[-0.2209 0.3295  0.8826  1.4392  2.003   2.5645]';



A=LeastSquares(x0,y0,1)%用最小二乘法拟合多项式的系数
%A=polyfit(x0,y0,1)

x=[-1.25:0.01:0.25];
y=polyval(A,x);%拟合函数函数值

plot(x0,y0,'o',x,y);
xlabel('x');
ylabel('y');
legend('raw data','LeastSquares');%标注
```

②最小二乘法拟合函数

LeastSquares.m

```
function A=LeastSquares(x,y,n)
%Input  -x m*1 矩阵,已知点的横坐标
%       -y m*1 矩阵,已知点的纵坐标
%       -n  最小二乘法拟合多项式的次数

%Output -A  1*(n+1)矩阵，多项式的系数矩阵 A=(an,…,a0);

%求已知点个数
m=length(x);

%初始化 A,X
A=zeros(n+1,1);
X=zeros(m,n+1);
X(:,1)=1;

for i=1:n
   X(:,i+1)=X(:,i).*x;
end

B=X'*X;
C=X'*y;

%求解法方程 X'XA=X'y
A=B\C;
A=A(n+1:-1:1)';
```

八、教师评语