# 实验二 最小二乘法（4 课时）

## 一、实验目的

1.了解最小二乘拟合的基本原理和方法，注意与插值方法的区别。

2.掌握最小二乘法。

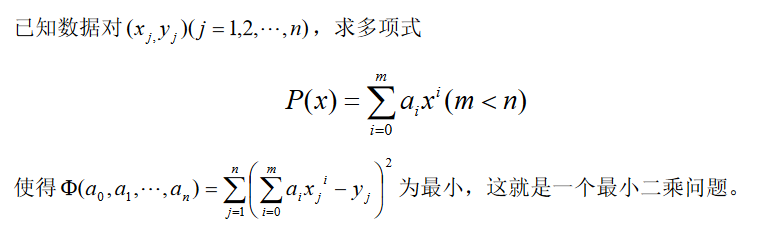
## 二、实验要求

1.掌握用 C 语言作最小二乘多项式拟合的方法。

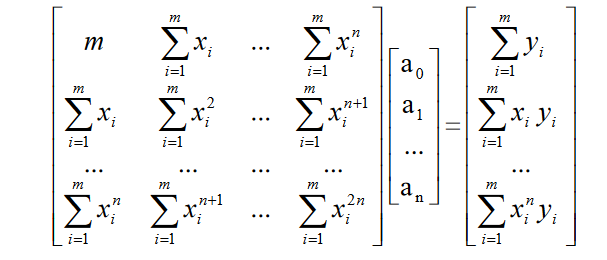
2.进一步加深对最小二乘法的理解。

## 三、实验原理

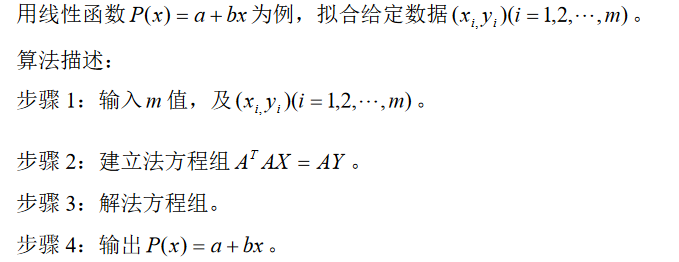
（一）最小二乘多项式拟合



（二）法方程组为



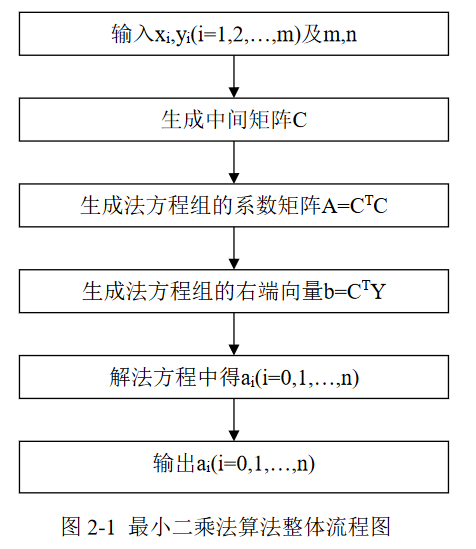
（三）最小二乘法计算步骤



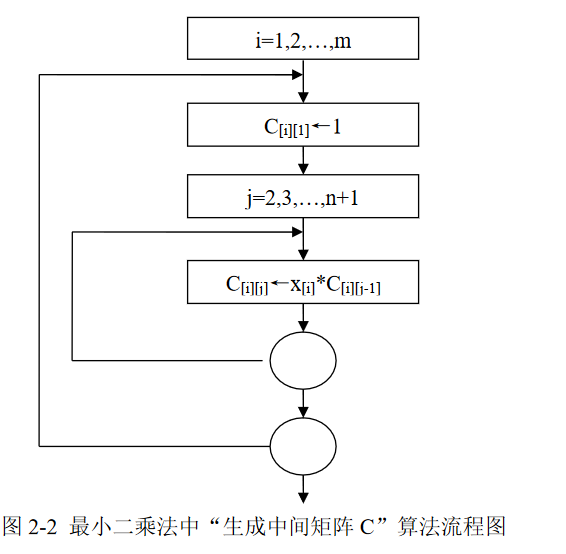
四、实验内容

（一）算法流程图

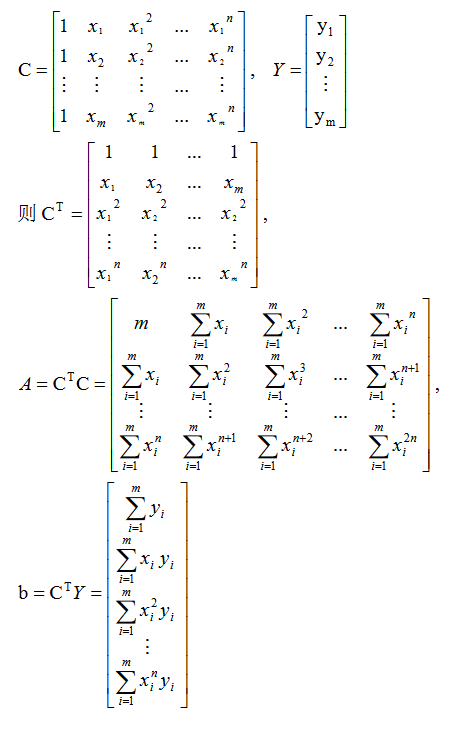
1.算法整体流程图（如图 2-1）



2.“生成中间矩阵 C”算法流程图（如图 2-2）

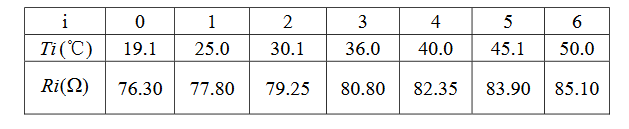


3.中间矩阵 C 的重要作用



（二）编程作业

测得铜导线在温度Ti (℃)时的电阻 Ri 如下表，求电阻 R 与温度 T 的近似函数关系

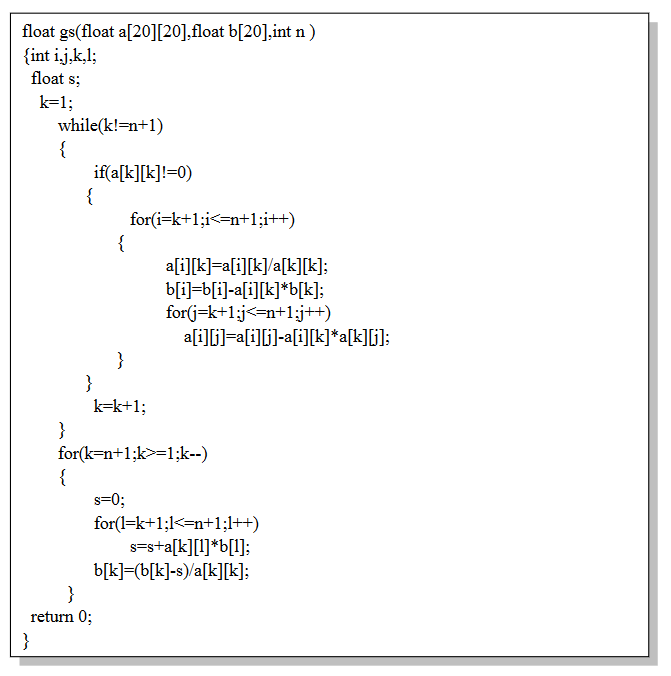


【提示】在进行程序实现时，务必注意中间矩阵的作用，以及非奇次线性方程组求

解问题！

为了实验的顺利完成，此处给出解非奇次线性方程组的高斯消元法的函数。请认真

阅读并理解。



实验主程序如下（请加上必要的注释）。

*#include* <bits/stdc++.h>  
  
*using namespace* std;  
*int* n, m;  
*double* x[10], y[10];  
  
*/\*  
7 1  
19.1 76.30  
25.0 77.80  
30.1 79.25  
36.0 80.80  
40.0 82.35  
45.1 83.90  
50.0 85.10  
 \*/  
int* main() {  
 *// 输入* cin >> m >> n;  
 *for* (*int* i = 1; i <= m; i++) {  
 cin >> x[i] >> y[i];  
 }  
 *// 生成中间矩阵C  
 double* C[m + 1][n + 2];  
 *for* (*int* i = 1; i <= m; i++) {  
 C[i][1] = 1;  
 *for* (*int* j = 2; j <= n + 1; j++) {  
 C[i][j] = x[i] \* C[i][j - 1];  
 }  
 }  
 *// 生成法方程组系数矩阵 A = C^T \* C  
 double* A[n + 2][n + 2];  
 *for* (*int* i = 1; i <= n + 1; i++) {  
 *for* (*int* j = 1; j <= n + 1; j++) {  
 *for* (*int* k = 1; k <= m; k++) {  
 A[i][j] += C[k][i] \* C[k][j];*//C^T \* C* }  
 }  
 }  
 *// 生成法方程组右端向量 b = C^T \* Y  
 double* b[n + 2];  
 *for* (*int* i = 1; i <= n + 1; i++) {  
 *for* (*int* k = 1; k <= m; k++) {  
 b[i] += C[k][i] \* y[k];*//C^T \* Y* }  
 }  
 *// 使用高斯消元法解非齐次线性方程组 AX=b  
 for* (*int* k = 1; k <= n + 1; k++) {  
 *if* (A[k][k] == 0) *continue*;  
 *for* (*int* i = k + 1; i <= n + 1; i++) {  
 A[i][k] = A[i][k] / A[k][k];  
 b[i] = b[i] - A[i][k] \* b[k];  
 *for* (*int* j = k + 1; j <= n + 1; j++)  
 A[i][j] = A[i][j] - A[i][k] \* A[k][j];  
 }  
 }  
 *for* (*int* k = n + 1; k >= 1; k--) {  
 *double* s = 0;  
 *for* (*int* l = k + 1; l <= n + 1; l++)  
 s = s + A[k][l] \* b[l];  
 b[k] = (b[k] - s) / A[k][k];  
 }  
 *// 输出结果* cout << "a = ";  
 *for* (*int* i = 1; i <= n + 1; i++) {  
 cout << b[i] << ", ";  
 }  
 *// 对结果进行验算,计算误差* cout << "\nR(t) = " << b[1] << " + " << b[2] << "t\n";  
 *double* diff = 0;  
 *for* (*int* i = 1; i <= m; i++) {  
 *double* r = b[1] + b[2] \* x[i];  
 *double* d = abs(r - y[i]) \* 100 / y[i];  
 cout << "R(" << x[i] << ") = " << r << " ;\t 相对误差为" << d << "%\n";  
 diff += d;  
 }  
 cout << "平均误差:" << diff / m << "%\n";  
}

拟合结果:

