

《数值计算与最优化技术》实验报告

年级、专业、班级	2021 级计算机科学与技术(卓越)02 班	姓名	文红兵
实验题目	最小二乘法		
实验时间	2023 年 5 月 7 日	实验地点	DS3305
实验成绩	优秀/良好/中等	实验性质	<input type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input type="checkbox"/> 综合性
教师评价： <input type="checkbox"/> 算法/实验过程正确； <input type="checkbox"/> 源程序/实验内容提交； <input type="checkbox"/> 程序结构/实验步骤合理； <input type="checkbox"/> 实验结果正确； <input type="checkbox"/> 语法、语义正确； <input type="checkbox"/> 报告规范； 其他： <div>评价教师: 文静</div>			
实验目的 理解并熟练掌握最小二乘法。			

报告完成时间: 2023 年 5 月 7 日

1 实验项目内容

某地在某日的温度记录如下,共有 24 个数据点。

时间 (p.m.)	1	2	3	4	5	6	7	8	9	10	11	午夜
温度	66	66	65	64	63	63	62	61	60	60	59	58
时间 (a.m.)	1	2	3	4	5	6	7	8	9	10	11	正午
温度	58	58	58	58	57	57	57	58	60	64	67	68

表 1: 温度变化表格

寻找上表的 3 次多项式最小二乘曲线并画图。

注意:所有程序请用 python 语言实现。只提交本电子文档,注意本文件末尾的文件命名要求;源程序一节请用代码备注的方式说明你的算法和思路;实验结果一节需要提供测试结果截图并给出结果分析。

2 实验分析

最小二乘法三次多项式拟合:

$$y = a_3x^3 + a_2x^2 + a_1x + a_0$$

函数空间的基:

$$\{1, x, x^2, x^3\}$$

其中, $1, x, x^2, x^3$ 分别表示常数项、一次项、二次项和三次项

最小二乘法的法方程为:

$$X\beta = Y$$

其中, β 是待求解的系数向量。

3 实验结果

其中 X:

```
[[ 4 10 30 100]
 [ 10 30 100 354]
 [ 30 100 354 1300]
 [ 100 354 1300 4890]]
```

图 1: X

其中 Y :

Y矩阵是: [10 30 100 354]

图 2: Y

其中系数:

系数是: [6.63815669e-13 1.00000000e+00 4.59347375e-13 -6.04972862e-14]

图 3: 系数

其中拟合结果：

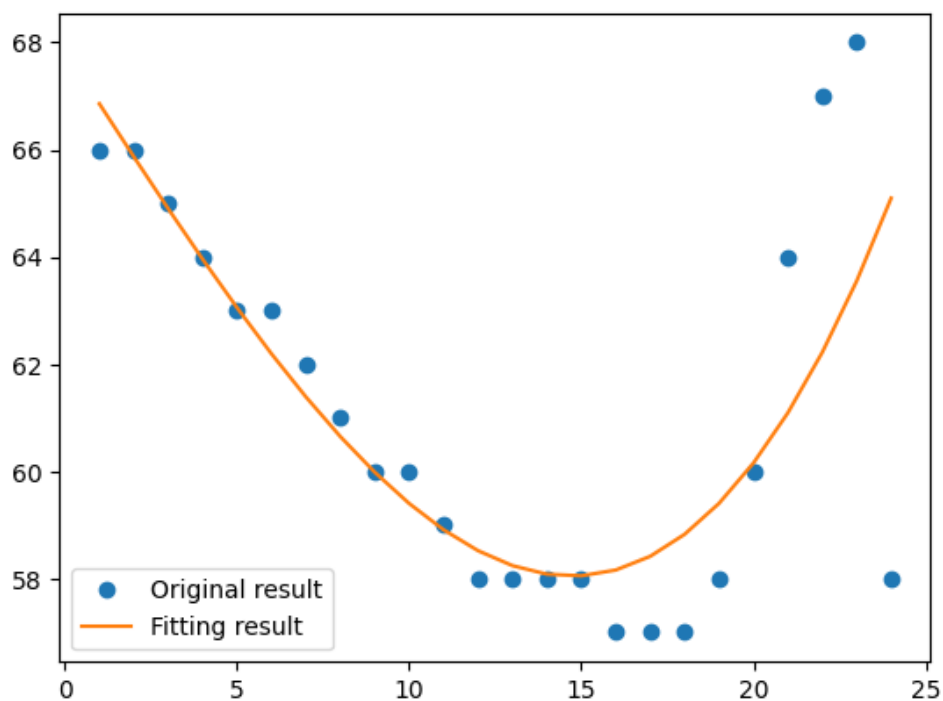


图 4: 拟合结果

A 最小二乘法代码

```
import numpy as np
import matplotlib.pyplot as plt

def Least_square_method(x, y, n):
    """
    最小二乘法拟合
    :param x: x->list, 自变量
    :param y: y->list, 因变量
    :param n: 拟合的次数
    :return: ret->list, 返回拟合的参数
    """

    # 构造X的矩阵
    X = []
    a = x
    ret = [len(x)]
    for i in range(1, 2 * n + 1):
        ret.append(sum(a))
        a = [num * a[j] for j, num in enumerate(x)]
    for i in range(n + 1):
        X.append(ret[i: i + n + 1])
```

```

X = np.array(X)
print("X矩阵是: \n", X)

# 构造y的矩阵
Y = []
a = y
for i in range(n + 1):
    Y.append(sum(a))
    a = [num1 * num2 for (num1, num2) in zip(a, x)]
Y = np.array(Y)
print("Y矩阵是: ", Y)

return np.linalg.solve(X, Y)

# 时间数据
x = np.arange(1, 25)

# 温度数据
y = np.array([66, 66, 65, 64, 63, 63, 62, 61, 60, 60, 59, 58, 58, 58, 58, 57, 57,
57, 58, 60, 64, 67, 68, 58])

# 3次多项式拟合
# 调用numpy的接口验证答案
p = np.polyfit(x, y, 3)
y_fit = np.polyval(p, x)
#自己写的多项式拟合
p1 = Least_square_method(x, y, 3)
y_fit2 = np.polyval(p1[:-1], x)

# 绘图
plt.plot(x, y, 'o', label='Original result')
plt.plot(x, y_fit, label='Reference result')
plt.legend()
plt.show()

plt.plot(x, y, 'o', label='Original result')
plt.plot(x, y_fit2, label='Fitting result')
plt.legend()
plt.show()

```