

# 计算物理作业 4

李明钰 22307110156

2024 年 10 月 24 日

## 1 题目 1: Newton 插值法和三次样条插值法

### 1.1 题目描述

Newton interpolation of:

1. 10 equal spacing points of  $\cos(x)$  within  $[0, \pi]$
2. 10 equal spacing points  $1/(1 + 25x^2)$  within  $[-1, 1]$ .

Compare the results with the cubic spline interpolation.

### 1.2 程序描述

先生成十个样本点  $x_{1,i} = 0 + \frac{\pi}{10}i, y_{1,i} = \cos(x_{1,i}) \quad i = 0, 1, \dots, 9$  然后利用牛顿插值法 (算法1)、插值顺序反向后进行牛顿插值法、拉格朗日插值法 (算法2)、三次样条插值法 (算法3) 分别进行插值, 得到插值后的函数  $f_{1,NI}$ 、 $f_{1,invNI}$ 、 $f_{1,LI}$ 、 $f_{1,CSI}$ , 并画出函数图像进行对比, 然后画出了原函数  $f_1 = \cos(x)$  减去上述插值得到函数的图像, 用以了解各插值方法的误差。

对第二题重复上述操作, 但是不再测试插值反向后牛顿插值法的情况。

### 1.3 伪代码

Newton 插值法的伪代码如下所示

---

#### Algorithm 1 Newton 插值法

---

- 1: **Input:** 样本点  $x_{\text{sample}}$  (自变量数组), 样本点  $y_{\text{sample}}$  (函数值数组)
- 2: **Output:** Newton 插值多项式的函数  $P(x)$
- 3: 设  $n \leftarrow \text{len}(x_{\text{sample}}) - 1$
- 4: 初始化  $N$  为零矩阵, 大小为  $n \times \text{len}(x_{\text{sample}})$
- 5: 初始化  $a$  为零向量, 大小为  $n$
- 6: **Step 1:** 计算第一个  $a_0$  的值
- 7:  $a_0 \leftarrow \frac{y_{\text{sample},1} - y_{\text{sample},0}}{x_{\text{sample},1} - x_{\text{sample},0}}$
- 8:  $N_0(x) \leftarrow y_{\text{sample},0} + a_0 \cdot (x - x_{\text{sample},0})$
- 9: **Step 2:** 迭代计算  $a_i$  的值 ( $i = 1, 2, \dots, n - 1$ )
- 10: **for**  $i = 1$  **to**  $n - 1$  **do**

```

11:  设  $N_{\text{part}} \leftarrow 1$ 
12:  for  $j = 0$  to  $i$  do
13:       $N_{\text{part}} \leftarrow N_{\text{part}} \cdot (x_{\text{sample}} - x_{\text{sample},j})$ 
14:  end for
15:  计算  $a_i$ 
16:   $a_i \leftarrow \frac{y_{\text{sample},i+1} - N_{i-1}(x_{\text{sample},i+1})}{N_{\text{part}}(x_{\text{sample},i+1})}$ 
17:  更新  $N_i(x)$ 
18:   $N_i(x) \leftarrow N_{i-1}(x) + a_i \cdot N_{\text{part}}(x)$ 
19: end for
20: Step 3: 定义 Newton 插值多项式函数  $P(x)$ 
21:  $P(x) \leftarrow y_{\text{sample},0}$ 
22: for  $i = 0$  to  $n - 1$  do
23:     设  $y_{\text{part}} \leftarrow 1$ 
24:     for  $j = 0$  to  $i$  do
25:          $y_{\text{part}} \leftarrow y_{\text{part}} \cdot (x - x_{\text{sample},j})$ 
26:     end for
27:      $P(x) \leftarrow P(x) + a_i \cdot y_{\text{part}}$ 
28: end for
29: return  $P(x)$ 

```

---

Lagrange 插值法伪代码如下所示

---

#### Algorithm 2 Lagrange 插值法

---

```

1: Input: 样本点  $x_{\text{sample}}$  (自变量数组), 样本点  $y_{\text{sample}}$  (函数值数组)
2: Output: Lagrange 插值多项式的函数  $P(x)$ 
3: 设  $n \leftarrow \text{len}(x_{\text{sample}})$ 
4: function A( $x, j$ )
5:      $A \leftarrow 1$ 
6:     for  $i = 0$  to  $n - 1$  do
7:         if  $i \neq j$  then
8:              $A \leftarrow A \cdot \frac{(x - x_{\text{sample},i})}{(x_{\text{sample},j} - x_{\text{sample},i})}$ 
9:         end if
10:    end for
11:    return  $A$ 
12: end function
13: function LAGRANGEINTERPOLATEDFUNC( $x$ )
14:      $y \leftarrow 0$ 
15:     for  $j = 0$  to  $n - 1$  do
16:          $y \leftarrow y + A(x, j) \cdot y_{\text{sample},j}$ 
17:     end for
18:     return  $y$ 
19: end function

```

20: **return** LAGRANGEINTERPOLATEDFUNC( $x$ )

三次样条插值的伪代码如下所示

---

**Algorithm 3** Cubic Spline 插值法伪代码

---

```
1: Input: 样本点  $x_{\text{sample}}$  (自变量数组), 样本点  $y_{\text{sample}}$  (函数值数组)
2: Output: Cubic Spline 插值多项式的函数  $P(x)$ 
3: 设  $n \leftarrow \text{len}(x_{\text{sample}})$ 
4: 设  $h \leftarrow \text{np.diff}(x_{\text{sample}})$ 
5: 设  $\delta_y \leftarrow \text{np.diff}(y_{\text{sample}})$ 
6: Step 1: 构造三对角矩阵  $A$  和向量  $b$ 
7: 初始化  $A$  为  $n \times n$  零矩阵
8: 初始化  $b$  为大小为  $n$  的零向量
9: 设  $A[0, 0] \leftarrow 1$  ▷ 自然边界条件  $S''(x_0) = 0$ 
10: 设  $A[n-1, n-1] \leftarrow 1$  ▷ 自然边界条件  $S''(x_n) = 0$ 
11: for  $i = 1$  to  $n-2$  do
12:    $A[i, i-1] \leftarrow h[i-1]$ 
13:    $A[i, i] \leftarrow 2 \cdot (h[i-1] + h[i])$ 
14:    $A[i, i+1] \leftarrow h[i]$ 
15:    $b[i] \leftarrow 6 \cdot \left( \frac{\delta_y[i]}{h[i]} - \frac{\delta_y[i-1]}{h[i-1]} \right)$ 
16: end for
17: Step 2: 求解二阶导数向量  $M$ 
18: 设  $M \leftarrow$  线性方程组  $A\vec{M} = \vec{b}$  的解
19: Step 3: 计算每段的样条系数
20: 初始化  $parameter$  为  $n-1 \times 4$  的零矩阵
21: for  $i = 0$  to  $n-2$  do
22:    $parameter[i, 0] \leftarrow y_{\text{sample}, i}$ 
23:    $parameter[i, 1] \leftarrow \frac{\delta_y[i]}{h[i]} - \frac{h[i]}{6} \cdot (M[i+1] + 2 \cdot M[i])$ 
24:    $parameter[i, 2] \leftarrow \frac{M[i]}{2}$ 
25:    $parameter[i, 3] \leftarrow \frac{M[i+1] - M[i]}{6 \cdot h[i]}$ 
26: end for
27: function CUBICSPLINEINTERPOLATEDFUNC( $x$ )
28:   for  $i = 0$  to  $n-2$  do
29:     if  $x_{\text{sample}, i} \leq x \leq x_{\text{sample}, i+1}$  then
30:        $y \leftarrow parameter[i, 0] + parameter[i, 1] \cdot (x - x_{\text{sample}, i})$ 
31:        $y \leftarrow y + parameter[i, 2] \cdot (x - x_{\text{sample}, i})^2$ 
32:        $y \leftarrow y + parameter[i, 3] \cdot (x - x_{\text{sample}, i})^3$ 
33:       return  $y$ 
34:     end if
35:   end for
36: end function
37: return CUBICSPLINEINTERPOLATEDFUNC( $x$ )
```

---

## 1.4 运行结果

运行得到的结果如下图所示，可以发现，对  $f_1(x) = \cos(x)$  多项式插值效果更好，而对  $f_2(x) = \frac{1}{1+25x^2}$  样条插值法效果更好。

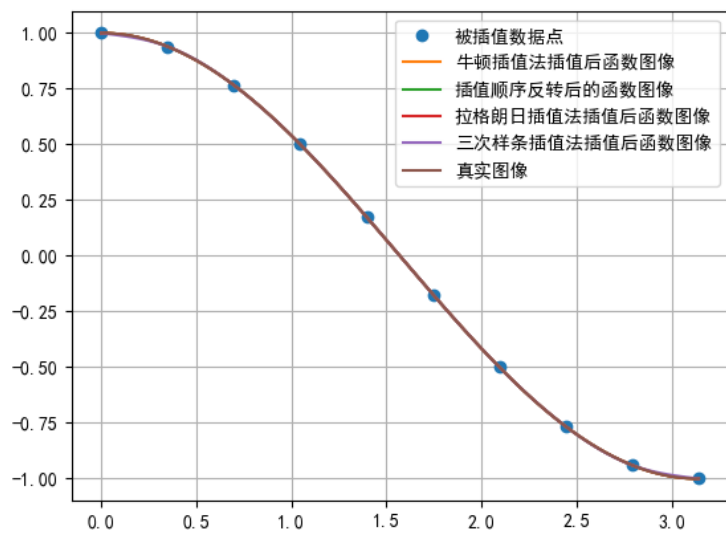


图 1:  $f_1 = \cos(x)$  与各插值函数图像

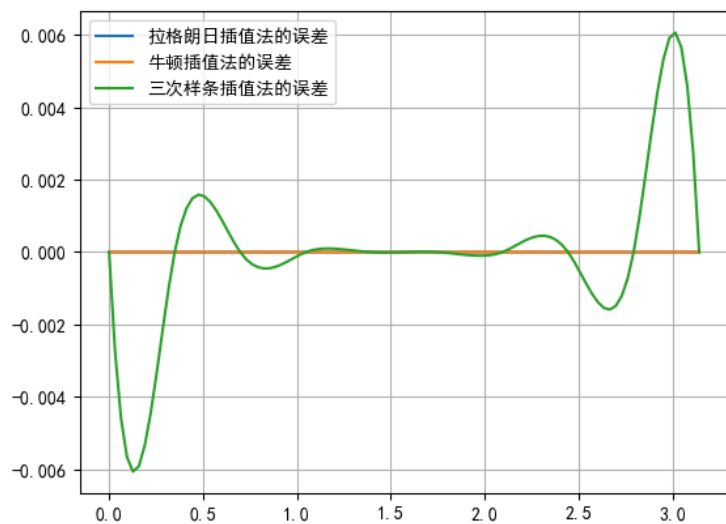


图 2:  $f_1(x) = \cos(x)$  与各插值函数误差

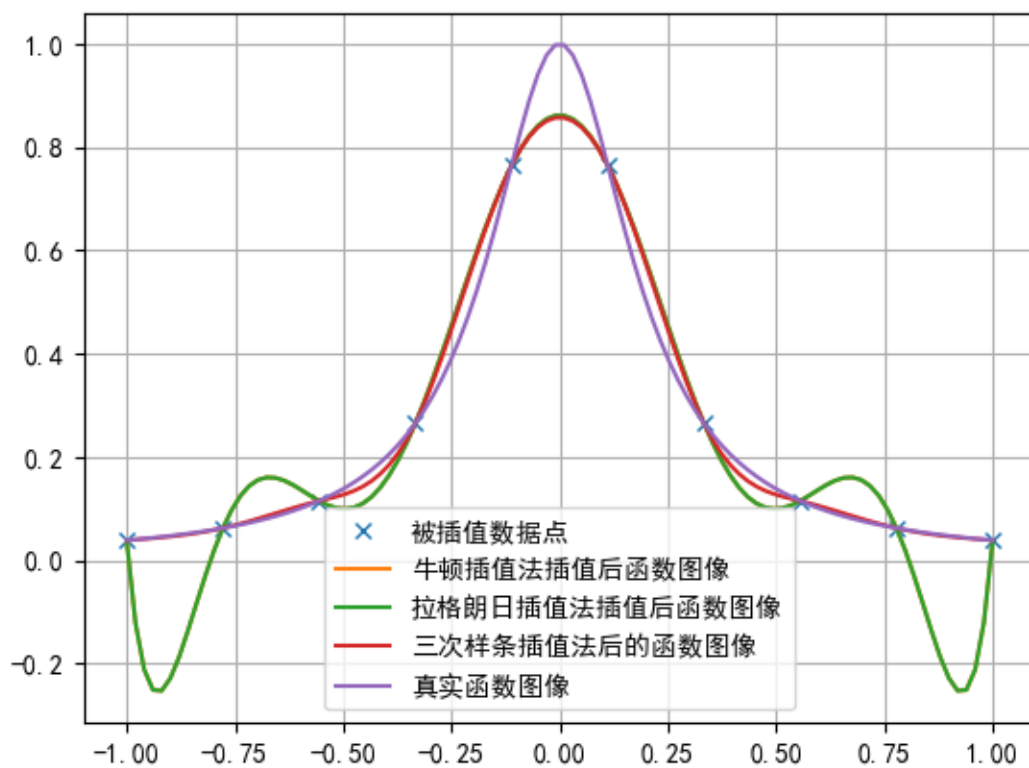


图 3:  $f_2(x) = \frac{1}{1+25x^2}$  与各插值函数图像

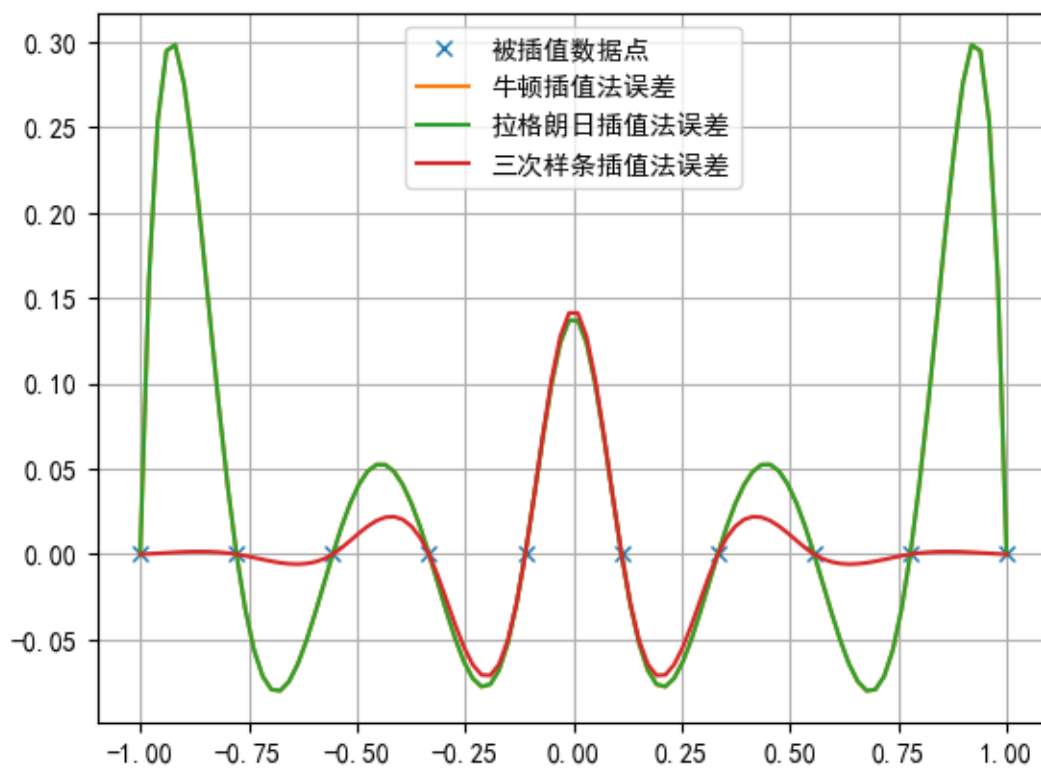


图 4:  $f_2(x) = \frac{1}{1+25x^2}$  与各插函数误差

## 1.5 附加题证明

牛顿插值法是根据  $n+1$  个点  $(x_i, y_i), i = 1, 2, \dots, n+1$  得到一个  $n$  次多项式  $P_n(x) = \sum_{i=0}^n a_i x^i$ , 实际上是一个解  $(n+1)$  元线性方程组的过程, 事实上, 过这  $n+1$  个点的  $n$  次多项式只有一个形式, 因此不管如何改变牛顿插值法的插入顺序其结果都相同且与拉格朗日插值法得到结果相等。

这一结果也可以由对  $f_1(x)$  的插值结果验证

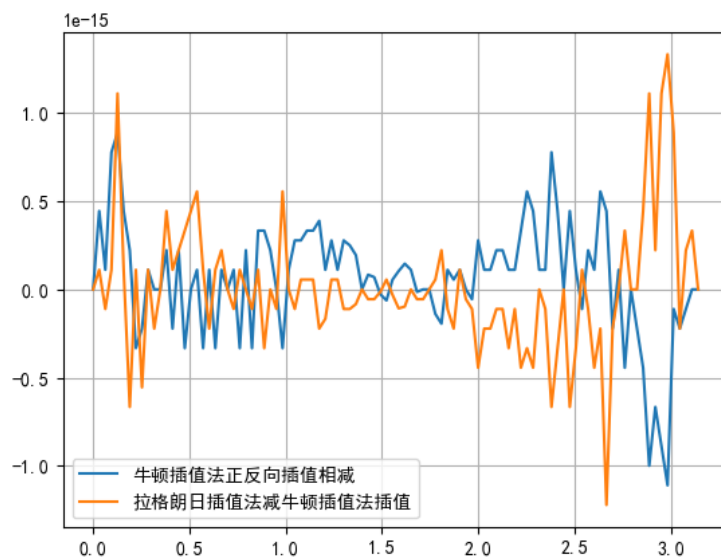


图 5: 多项式插值相减.png

## 2 题目 2: 线性拟合与抛物线拟合

### 2.1 题目描述

The table below gives the temperature  $T$  along a metal rod whose ends are kept at fixed constant temperatures.

The temperature is a function of the distance  $x$  along the rod.

$x_i$ (cm)	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
$T_i$ (°C)	14.6	18.5	36.6	30.8	59.2	60.1	62.2	79.4	99.9

1. Compute a least-squares, straight-line fit to these data using  $T(x) = ax + b$
2. Compute a least-squares, parabolic-line fit to these data using  $T(x) = a + bx + cx^2$

### 2.2 程序描述

利用线性和最小二乘法, 分别计算除了线性拟合和抛物线拟合的结果, 并计算出来相应的  $R^2$ 。

### 2.3 伪代码

---

#### Algorithm 4 Straight Line Fit

---

```
1: function STRAIGHTLINEFIT(x, y)
2:    $n \leftarrow \text{len}(x)$ 
3:    $a \leftarrow \frac{\sum(x \cdot y) - n \cdot \text{mean}(y) \cdot \text{mean}(x)}{\sum(x^2) - n \cdot (\text{mean}(x))^2}$ 
```

```

4:    $b \leftarrow \text{mean}(y) - a \cdot \text{mean}(x)$ 
5:   return  $a, b$ 
6: end function

```

---

#### Algorithm 5 Parabolic Line Fit

---

```

1: function PARABOLICLINEFIT( $x, y$ )
2:    $n \leftarrow \text{len}(x)$ 
3:    $X \leftarrow \text{zeros}(n, 3)$ 
4:   for  $i = 0$  to  $n - 1$  do
5:     for  $j = 0$  to  $2$  do
6:        $X[i][j] \leftarrow x[i]^j$ 
7:     end for
8:   end for
9:    $a \leftarrow (\text{inv}(X^T \cdot X) \cdot (X^T \cdot y))$ 
10:  return  $a[0], a[1], a[2]$ 
11: end function

```

---

### 2.4 运行结果

```

1 print("线性拟合得到结果为y=()x+(),R方为{}".format(a1,b1,R_squared1))
2 print("抛物线拟合得到结果为y=()x+()x^2+(),R方为{}".format(a2,b2,c2,R_squared2))
✓ 0.0s
线性拟合得到结果为y=10.07333333333327x+0.888888888889284,R方为0.9411125465748046
抛物线拟合得到结果为y=8.261904761906393+6.051688311688025x+0.40216450216452726x^2,R方为0.9488127480227583

```

图 6: 拟合结果

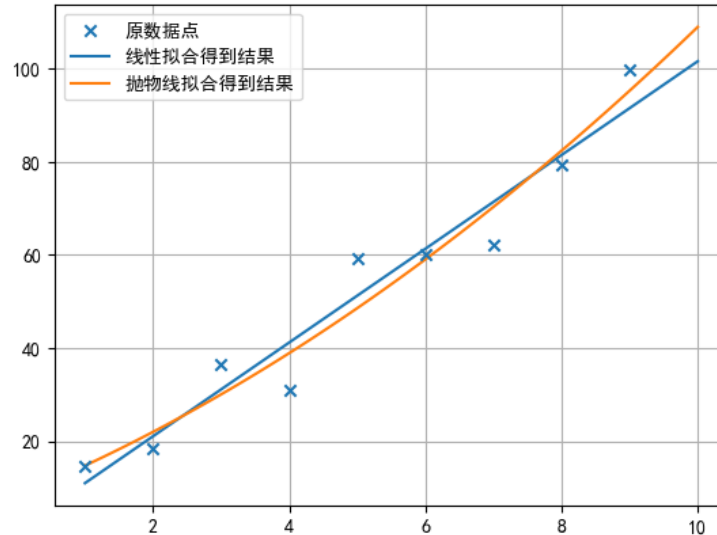


图 7: 拟合图像