

《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 2 班				姓名	文红兵																																						
实验题目	插值法																																											
实验时间	2023 年 4 月 23 日			实验地点	DS3305																																							
实验成绩				实验性质	<input type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input type="checkbox"/> 综合性																																							
<p>教师评价：</p> <p><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；</p> <p><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；</p> <p>其他：</p> <p>评价教师签名：</p>																																												
<h2>一、实验目的</h2> <p>理解并熟练掌握三种插值法：拉格朗日插值法，牛顿插值法和三次样条插值法，并分析三种算法。</p>																																												
<h2>二、实验项目内容</h2> <p>1. 下表是水的表面张力对温度的函数数据：</p> <table><tr><td>$T(^{\circ}\text{C})$</td><td>0</td><td>20</td><td>40</td><td>60</td><td>80</td></tr><tr><td>$\sigma \times 10^3(\text{N/m})$</td><td>78.2</td><td>73.4</td><td>70.2</td><td>66.7</td><td>63.2</td></tr></table> <p>利用牛顿插值法求出穿过上述数据点的多项式，画出σ相对于 T 的关系图，并标出数据点的位置。</p> <p>2.</p> <p>1)使用拉格朗日插值法求出穿过下述数据点的多项式并画图,可以从中发现什么问题？</p> <table><tr><td>xi</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>yi</td><td>3.5</td><td>3.0</td><td>2.5</td><td>2.0</td><td>1.5</td><td>-2.4</td><td>-2.8</td><td>-3.2</td><td>-3.6</td><td>-4.0</td></tr></table> <p>2) 使用三次样条多项式对上述数据集重新进行插值并画图。</p>											$T(^{\circ}\text{C})$	0	20	40	60	80	$\sigma \times 10^3(\text{N/m})$	78.2	73.4	70.2	66.7	63.2	xi	1	2	3	4	5	6	7	8	9	10	yi	3.5	3.0	2.5	2.0	1.5	-2.4	-2.8	-3.2	-3.6	-4.0
$T(^{\circ}\text{C})$	0	20	40	60	80																																							
$\sigma \times 10^3(\text{N/m})$	78.2	73.4	70.2	66.7	63.2																																							
xi	1	2	3	4	5	6	7	8	9	10																																		
yi	3.5	3.0	2.5	2.0	1.5	-2.4	-2.8	-3.2	-3.6	-4.0																																		

报告创建时间：

三、实验过程或算法（源程序）

1、牛顿插值

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

def NewTon(data_x, data_yy, x):
    n = len(data_x)

    data_y = np.array(data_yy)
    temp = 1
    y = data_y[0]
    for j in range(1, n + 1):
        temp *= (x - data_x[j - 1])
        for i in range(n - j):
            data_y[i] = (data_y[i + 1] - data_y[i]) / (data_x[i + j]
- data_x[i])
        y += data_y[0] * temp
    return y

if __name__ == '__main__':
    data_x = [i for i in range(0, 81, 20)]
    data_y = [78.2, 73.4, 70.2, 66.7, 63.2]
    predict_x = np.linspace(min(data_x), max(data_x), 100)
    predict_y = np.array([NewTon(data_x, data_y, x) for x in
predict_x])
    plt.figure()
    plt.plot(predict_x, predict_y)
    plt.scatter(data_x, data_y)
    plt.show()
```

2、拉格朗日插值和三次样条插值

拉格朗日插值

```
import numpy as np
from scipy.interpolate import interp1d
import matplotlib.pyplot as plt

def Lagrange(data_x, data_y, x):
    n = len(data_x)
    ret = 0

    # Lagrange 函数处理
    for i in range(n):
        product_down = 1
        product_up = 1
        for j in range(n):
            if i == j:
                continue
            product_down *= data_x[i] - data_x[j]
            product_up *= x - data_x[j]
        ret += product_up / product_down * data_y[i]

    return ret

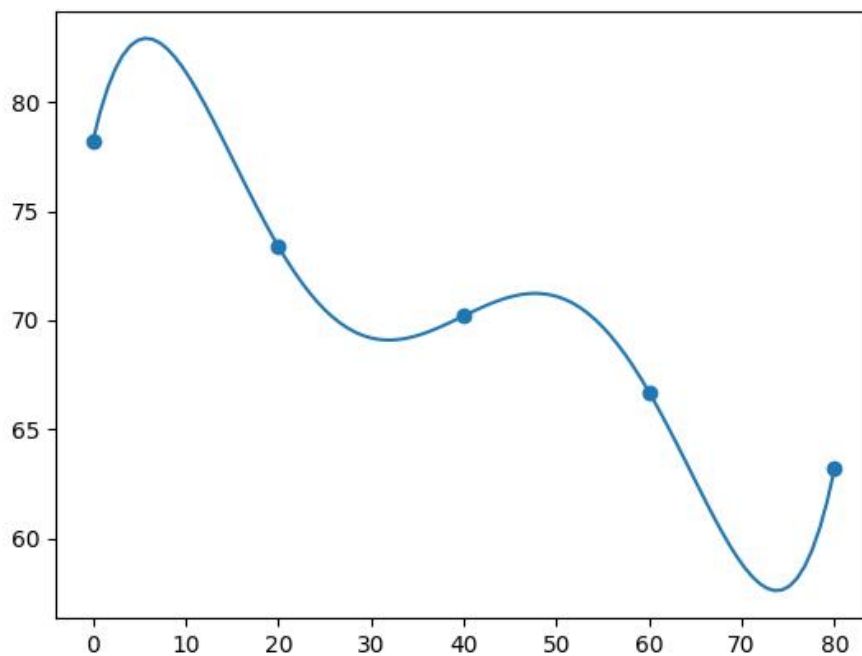
if __name__ == '__main__':
    data_x = [i for i in range(1, 11, 1)]
    data_y = [3.5, 3.0, 2.5, 2.0, 1.5, -2.4, -2.8, -3.2, -3.6, -4.0]
    predict_x = np.linspace(min(data_x), max(data_x), 100)
    predict_y = np.array([Lagrange(data_x, data_y, x) for x in
predict_x])
    plt.figure()
    plt.plot(predict_x, predict_y)
    plt.scatter(data_x, data_y)
    plt.show()
```

三次样条插值

```
func = interp1d(data_x, data_y, kind='cubic')
predict_x = np.linspace(min(data_x), max(data_x), 100)
predict_y = np.array([func(x) for x in predict_x])
plt.figure()
plt.plot(predict_x, predict_y)
plt.scatter(data_x, data_y)
plt.show()
```

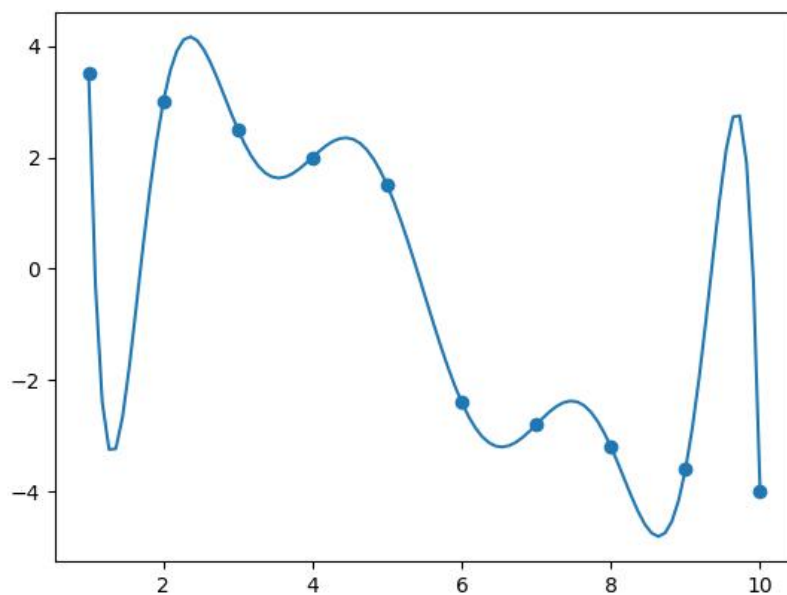
四、实验结果及分析和（或）源程序调试过程

1、牛顿插值

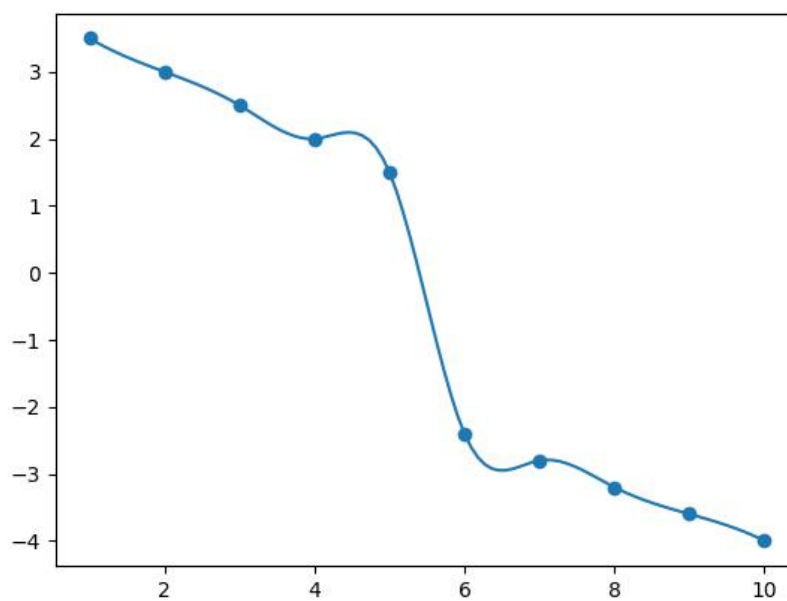


2、

拉格朗日插值



三次样条插值



分析：

使用拉格朗日插值法进行插值可以明显地看到出现了龙格现象，边缘的插值结果偏差极大。

相对的，使用三次样条插值并没有出现龙格现象，而且曲线更加光滑