

第四次上机作业

Anonymity 3220100000*

* College of Control Science and Engineering, Robotics Engineering

July 8, 2024

1 问题描述

考虑区间 $[-5, 5]$ 上的函数: $f(x) = \frac{x}{1+x^4}$

以 $x_i = -5 + \frac{10i}{n}, i = 0, 1, 2, \dots, n$ 为插值节点, 取不同的分点数目 $n = 2, 3, 4, \dots, 10$, 分别进行拉格朗日插值、分段线性插值, 并画出原函数 $f(x)$ 及插值多项式函数在 $[-5, 5]$ 上的图像, 比较并分析实验结果。

2 思路分析

2.1 拉格朗日插值

对插值节点 x_k , 构造一个 n 次多项式 $l_k(x)$, 使之满足条件:

$$l_k(x_i) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \quad (i = 0, 1, 2, \dots, n)$$

$x_i (i \neq k)$ 为 $l_k(x)$ 的零点, 故有

$$l_k(x) = a_k (x - x_0) (x - x_1) \dots (x - x_{k-1}) (x - x_{k+1}) \dots (x - x_n)$$

其中 a_k 为待定系数。可由条件 $l_k(x_k) = 1$ 确定, 于是得到:

$$l_k(x) = \frac{(x - x_0) (x - x_1) \dots (x - x_{k-1}) (x - x_{k+1}) \dots (x - x_n)}{(x_k - x_0) (x_k - x_1) \dots (x_k - x_{k-1}) (x_k - x_{k+1}) \dots (x_k - x_n)} = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{(x - x_i)}{(x_k - x_i)}$$

若 $f(x_i) = y_i$, 利用插值基函数 $l_i(x)$, 则得拉格朗日插值多项式:

$$L_n(x) = y_0 l_0(x) + y_1 l_1(x) + \dots + y_n l_n(x) = \sum_{i=0}^n l_i(x) y_i$$

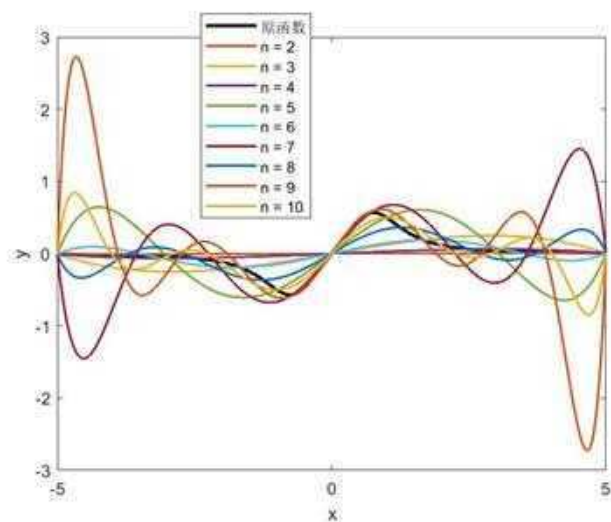
2.2 分段线性插值

把整个插值区间分成若干个小区间, 对给定区间 $[a, b]$ 做分割: $a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$ 在每个小区间 $[x_{i-1}, x_i]$ 上采用一次线性插值公式。在几何上就是用折线去近似替代曲线。

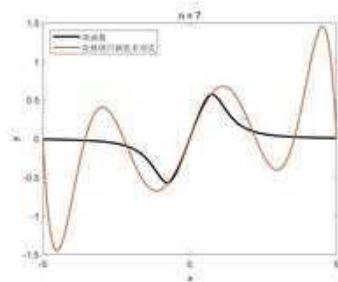
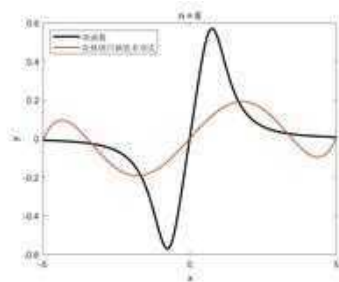
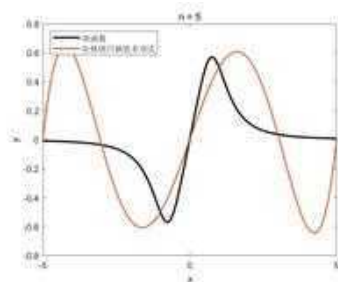
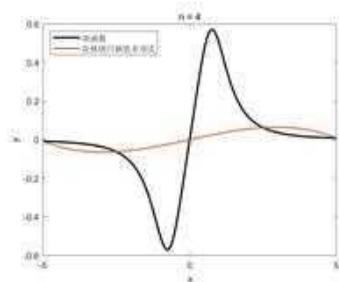
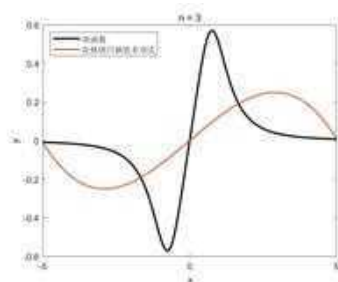
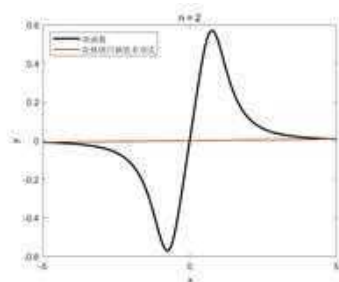
3 运行结果与分析

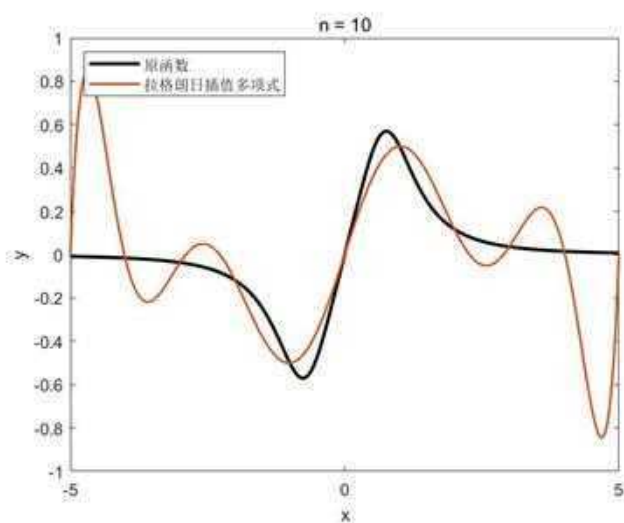
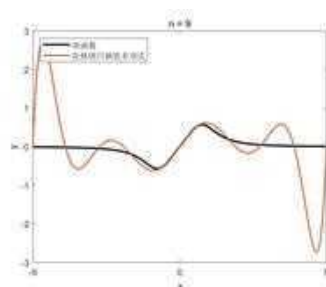
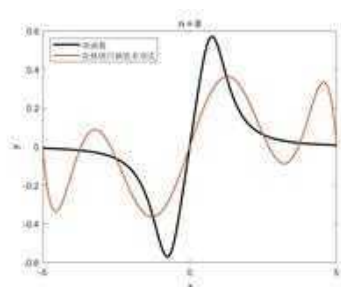
3.1 拉格朗日插值

原函数 $f(x)$ 及插值多项式函数在 $[-5, 5]$ 上的图像:



由于函数过多，因此分开绘制图像:



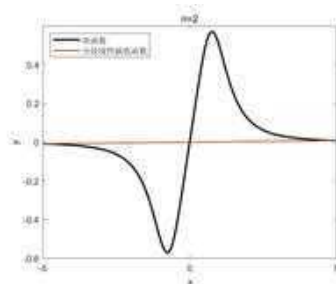
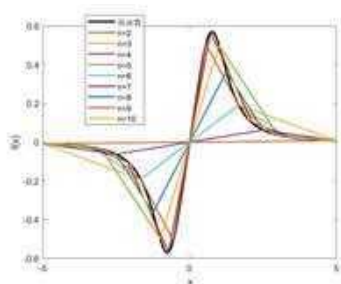


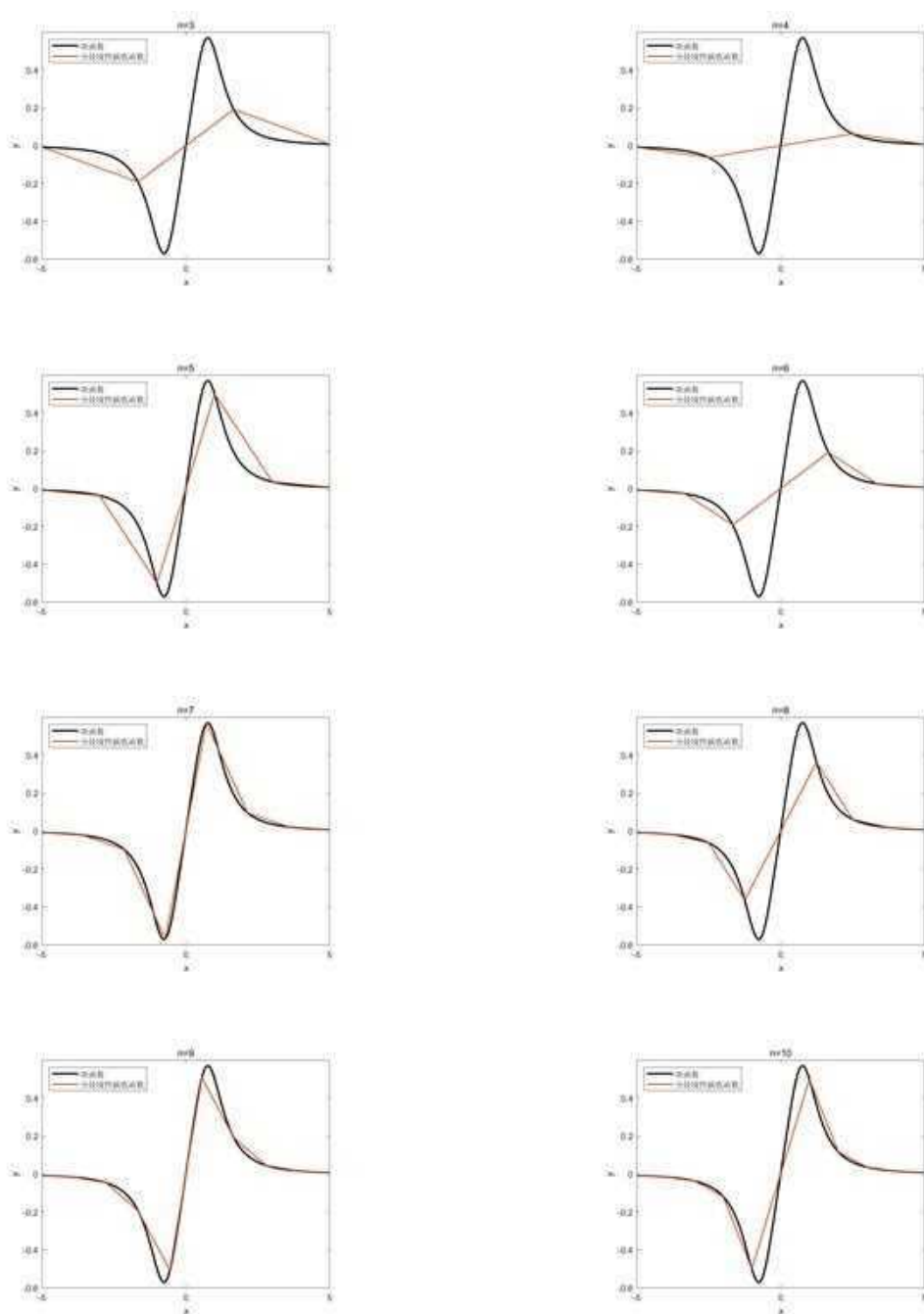
由以上图像直观分析知，高次插值的拟合程度更高，符合规律。

同时，可以发现偏差较大的地方均出现在原函数较为平缓之处，这是因为拉格朗日插值引入的幂函数存在波动性，对于斜率较小的函数图像拟合度很低。

3.2 分段线性插值

原函数 $f(x)$ 及插值多项式函数在 $[-5, 5]$ 上的图像：





直观分析以上图像，整体上符合，插值多项式的次数 n 的越高，区间逐渐减小，插值函数逐渐逼近原函数，误差在不断减小。

但是由于目标函数的特殊性，原函数 $f(x) = \frac{x}{1+x^4}$ 的极值点为 $x = \frac{1}{\sqrt[3]{3}}$ 和 $x = -\frac{1}{\sqrt[3]{3}}$ ，其近似值为 0.75925 和 -0.75925，在 $n = 7$ 时与取到的 $x_3 = -0.714285$ 和 $x_4 = 0.714285$ 的值很相近，而两个极值点间的函数图像又接近线性，因此上述拟合是在 $n = 7$ 时的拟合程度最高。

3.3 小结

对比拉格朗日插值法和分段线性插值法，可知拉格朗日插值法得到的插值函数是光滑的，在插值节点处导数连续，部分区间上准确性不错；分段线性插值法，随着插值函数次数的增加会更

加逼近原函数，准确性更加高，但不能保证节点处插值函数导数的连续，因此图像也不光滑，因而不能满足某些工程技术上曲线光滑的要求。

4 实验心得

在编程过程中，我更加直观地体会到了拉格朗日插值法的缺点——无承袭性，即在为了提高精度需增加结点时，所有的基函数都要重新计算，延续性不好。

而分段低次插值函数公式简单，只要区间充分小，就能保证误差要求。同时它具有局部特性，修改某个节点 x_i 的值，仅在相邻区间受到影响。

总体来说，本次实验加深了我对于拉格朗日插值法和分段线性插值法两种插值方法的理解，理解了不同方法有不同适用情况这一道理。

5 源代码

5.1 拉格朗日插值混合图像

```
1 % 定义函数
2 f = @(x) x./(1+x.^4);
3
4 % 区间和分点数目
5 a = -5;
6 b = 5;
7 n_values = 2:10;
8
9 % 绘制原函数
10 x = linspace(a, b, 1000);
11 y = f(x);
12 plot(x, y, 'k-', 'LineWidth', 2);
13 hold on;
14
15 % 对不同的分点数进行插值
16 for n = n_values
17     % 计算插值节点
18     xi = linspace(a, b, n+1);
19
20     % 计算拉格朗日插值多项式
21     L = zeros(size(x));
22     for i = 1:length(xi)
23         li = ones(size(x));
24         for j = 1:length(xi)
25             if j ~= i
26                 li = li .* (x - xi(j)) / (xi(i) - xi(j));
```

```

27         end
28     end
29     L = L + f(xi(i)) * li;
30 end
31
32 % 绘制插值多项式
33 plot(x, L, 'LineWidth', 1.5);
34 end
35
36 % 添加图例和标签
37 legend('原函数', 'n = 2', 'n = 3', 'n = 4', 'n = 5', 'n = 6', 'n = 7',
        'n = 8', 'n = 9', 'n = 10', 'Location', 'Northwest');
38 xlabel('x');
39 ylabel('y');

```

5.2 拉格朗日插值单独图像

```

1 % 定义函数
2 f = @(x) x./(1+x.^4);
3
4 % 区间和分点数目
5 a = -5;
6 b = 5;
7 n_values = 2:10;
8
9 % 绘制原函数
10 x = linspace(a, b, 1000);
11 y = f(x);
12 plot(x, y, 'k-', 'LineWidth', 2);
13 hold on;
14
15 % 对不同的分点数进行插值
16 for n = n_values
17     % 计算插值节点
18     xi = linspace(a, b, n+1);
19
20     % 计算拉格朗日插值多项式
21     L = zeros(size(x));
22     for i = 1:length(xi)
23         li = ones(size(x));
24         for j = 1:length(xi)

```

```

25         if j ~= i
26             li = li .* (x - xi(j)) / (xi(i) - xi(j));
27         end
28     end
29     L = L + f(xi(i)) * li;
30 end
31
32 % 绘制插值多项式
33 plot(x, L, 'LineWidth', 1.5);
34 end
35
36 % 添加图例和标签
37 legend('原函数', 'n = 2', 'n = 3', 'n = 4', 'n = 5', 'n = 6', 'n = 7',
38         'n = 8', 'n = 9', 'n = 10', 'Location', 'Northwest');
39 xlabel('x');
40 ylabel('y');

```

5.3 分段线性插值混合图像

```

1 % 定义原函数
2 f = @(x) x./(1+x.^4);
3
4 % 分点数目
5 n_values = 2:10;
6
7 % 定义插值区间
8 x_interp = linspace(-5, 5, 1000);
9
10 % 绘制原函数图像
11 figure;
12 plot(x_interp, f(x_interp), 'k-', 'LineWidth', 2);
13 hold on;
14
15 for i = 1:length(n_values)
16     n = n_values(i);
17     % 计算插值节点
18     x_nodes = -5 + (10*(0:n)/n);
19     y_nodes = f(x_nodes);
20
21     % 执行分段线性插值
22     y_interp = interp1(x_nodes, y_nodes, x_interp, 'linear');

```

```

23
24     % 绘制插值多项式函数图像
25     plot(x_interp, y_interp, 'LineWidth', 1.5);
26 end
27
28 % 添加图例和标签
29 legend('原函数', 'n=2', 'n=3', 'n=4', 'n=5', 'n=6', 'n=7', 'n=8', 'n=9',
        'n=10', 'Location', 'northwest');
30 xlabel('x');
31 ylabel('f(x)');
32
33 % 设定图形范围
34 xlim([-5, 5]);
35 ylim([-0.6, 0.6]);

```

5.4 分段线性插值单独图像

```

1 % 定义原函数
2 f = @(x) x./(1+x.^4);
3
4 % 分点数目
5 n_values = 2:10;
6
7 % 定义插值区间
8 x_interp = linspace(-5, 5, 1000);
9
10 for i = 1:length(n_values)
11     n = n_values(i);
12     % 创建新的图像
13     figure;
14     % 绘制原函数图像
15     plot(x_interp, f(x_interp), 'k-', 'LineWidth', 2);
16     hold on;
17     % 计算插值节点
18     x_nodes = -5 + (10*(0:n)/n);
19     y_nodes = f(x_nodes);
20
21     % 执行分段线性插值
22     y_interp = interp1(x_nodes, y_nodes, x_interp, 'linear');
23
24     % 绘制插值多项式函数图像

```



```
25     plot(x_interp, y_interp, 'LineWidth', 1.5);
26
27     % 添加图例和标签
28     legend('原函数', '分段线性插值函数', 'Location', 'northwest');
29     xlabel('x');
30     ylabel('y');
31
32     % 设定图形范围
33     xlim([-5, 5]);
34     ylim([-0.6, 0.6]);
35     % 添加标题
36     title(['n=' num2str(n)]);
37 end
```