计算物理第四次作业

李柏轩 20300200004

October 12, 2022

1 题目 1：函数插值

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 $\frac{1}{1+25x^2}$ within $[-1, 1]$.

Compare the results with the cubic spline interpolation.

1.2 程序描述

本程序按照题目要求，分别使用 Newton 插值法以及 cubic splines 插值法对两个函数进行插值，输出插值函数图像以及误差图像。

Newton 插值法的原理公式为

$$y = a_0 + \sum_{i=1}^{n} \left(a_i \prod_{j=0}^{i-1} (x - x_j) \right) \tag{1}$$

主要使用数组差分方法来计算 Eq. 1中的系数，计算过程如 Fig. 1所示，其中 $f[x_n, ..., x_0] = \frac{f[x_n,...,x_1]-f[x_{n-1},...,x_0]}{x_n - x_0}$。由此得到多项式插值结果。

Cubic Splines 插值方法使用分段函数进行插值，在每两个数据点之间都采用三次多项式，形式如 Eq. 2所示。

$$\begin{aligned} f_i(x) =& \frac{f''(x_i)}{6(x_{i+1} - x_i)}(x_{i+1} - x)^3 + \frac{f''(x_{i+1})}{6(x_{i+1} - x_i)}(x - x_i)^3 \\ &+ \left[\frac{f''(x_i)}{x_{i+1} - x_i} - \frac{f''(x_i)(x_{i+1} - x_i)}{6} \right] (x_{i+1} - x) \\ &+ \left[\frac{f''(x_{i+1})}{x_{i+1} - x_i} - \frac{f''(x_{i+1})(x_{i+1} - x_i)}{6} \right] (x - x_i), i = 1, 2, ..., n-1 \end{aligned} \tag{2}$$

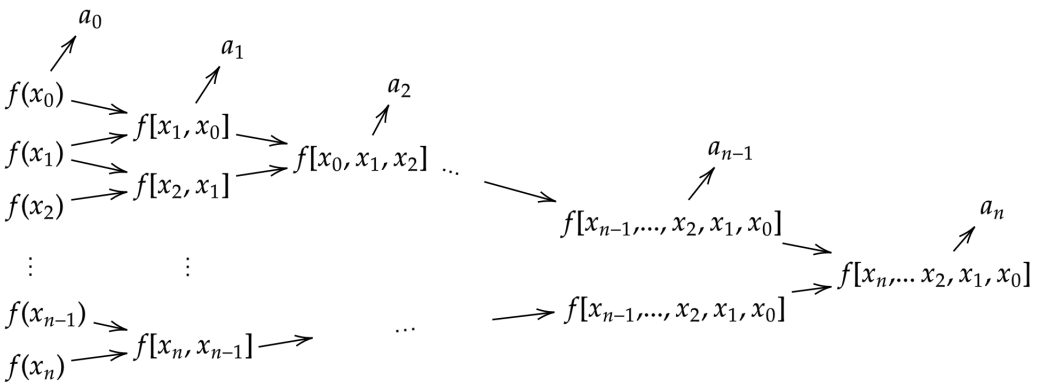

Figure 1: Newton 插值法的计算示意图

Eq. 2中未知数为 $f''(x_i), i = 1, 2, ...n$。根据自然边界条件与一阶导数连续的限制：

$$f''(x_0) = 0$$

$$f''(x_n) = 0$$

$$(x_i - x_{i-1})f''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f''(x_i) + (x_{i+1} - x_i)f''(x_{i+1}) = \frac{6}{x_{i+1} - x_i}[f(x_{i+1} - f(x_i)] + \frac{6}{x_i - x_{i-1}}[f(x_{i-1} - f(x_{i-1}) \tag{3}$$

得到三对角矩阵。采用 Thomas 算法（见课件，具体计算过程不在伪代码中列出）求解线性方程组，即得到插值函数。因为题目中两个函数都为均匀采样，故可以一定程度上简化运算。

本程序源文件为 Interpolation.py，运行依赖 Python 第三方库 Numpy 和 Matplotlib。在终端进入当前目录，使用命令 python -u Interpolation.py 运行本程序。运行后依次输出 Fig. 3,Fig. 4,Fig. 5,Fig. 6所示图像（需要关闭当前图像才能显示下一张图像），并在控制台输出插值函数的各未知系数。

1.3 伪代码

求解 Newton 插值法系数的伪代码如 Alg.1所示。求解 Cubic splines 插值法系数的伪代码如 Alg.2所示。

Algorithm 1 Calculating the coefficients of Newton interpolation method

Input: The number of sample points n, the y coordinates of sample points Y, and the step of sampling.

Output: The coefficients of Newton interpolation method $A = [a_0, ..., a_n]$

1: The first row of $F \leftarrow Y$

2: **for** i in range$(0, n-1)$ **do**

3: The first $n - i - 1$ elements in the $i + 1$th row of $F \leftarrow \frac{\text{the difference of the } i \text{ th row of } F}{\text{step} \times (i+1)}$

4: **end for**

5: **return** the first column of F

Algorithm 2 Calculating the coefficients of Cubic splines interpolation method

Input: The number of sample points n, the y coordinates of sample points Y, and the step of sampling.

Output: The coefficients of Cubic splines interpolation method $f''(x_i), i = 1, 2, ...n$

1: Get coefficients matrix A and b with Eq. 3

2: solve the linear equation with Thomas Algorithm and get the solution X

3: **return** X

1.4 输入输出示例

程序在控制台输出的两种方法的待求系数如 Fig.2和 Table.1,Table.2所示。

```
coefficients of Newton methods are: [ 1.00000000e+00 -1.72767915e-01 -4.65094866e-01  8.20724393e-02
  3.12762139e-02 -6.22353351e-03 -6.73588529e-04  1.79930720e-04
  4.13516274e-06 -2.63252636e-06]
f''(x_i)= [ 0.         -1.21997819 -0.70122561 -0.52490505 -0.16881588  0.16881588
  0.52490505  0.70122561  1.21997819  0.        ]
coefficients of Newton methods are: [  0.03846154   0.10601955   0.29513907   1.03451217   3.14532398
 -19.76308633  37.31208049 -38.44404401  21.62477475   0.        ]
f''(x_i)= [  0.          1.13172288  -0.98522269  14.6269341  -15.0619018
 -15.0619018   14.6269341   -0.98522269   1.13172288   0.        ]
```

Figure 2: 控制台输出的两种方法的待求系数

Table 1: $f(x) = \cos x$ 插值参数

a_i	1	-0.1728	-0.4651	0.0821	0.0313	-0.0062	-0.0007	0.0002	0.000004	-0.000003
$f''(x_i)$	0	-1.2200	-0.7012	-0.5249	-0.1688	0.1688	0.5249	0.7012	1.2200	0

Table 2: $f(x) = \frac{1}{1+25x^2}$ 插值参数

a_i	0.0385	0.1060	0.2951	1.0345	3.1453	-19.7631	37.3121	-38.4440	21.6248	0
$f''(x_i)$	0	1.1317	-0.9852	14.6269	-15.0619	-15.0619	14.6269	-0.9852	1.1317	0

对于 $f(x) = \cos x$ 在 $[0, \pi]$ 间使用两种方法的插值结果如 Fig. 3，两种方法的误差如 Fig. 4所示。可以看到 Newton 方法在误差图中几乎看不出误差，而 Cubic Splines 方法则在两端点处由于其较强的假定（二阶导为 0）而具有较大的误差。可以看出，当函数性质较好，没有很陡峭的变化时，由于 Newton 法使用的多项式阶数较高且没有额外假定，具有更好的插值效果。

对于 $f(x) = \frac{1}{1+25x^2}$ 在 $[-1, 1]$ 间使用两种方法的插值结果如 Fig. 5，两种方法的误差如 Fig. 6所示。可以看到误差 Newton 法插值误差较大，而 Cubic Splines 法误差较小。这是因为 Newton 插值法的误差 $\epsilon_n(x) = f(x) - p_x(x) = p_{n+1}(x) - p_n(x) = f[x, x_n, ..., x_0](x - x_0)...(x - x_n) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)...(x - x_n)$，$f^{(n)}(\xi)$ 的函数图像如 Fig. 7所示，高阶导数在 $[-1, 1]$ 间数值会越来越大，导致越采用高阶多项式插值，插值误差会越大。

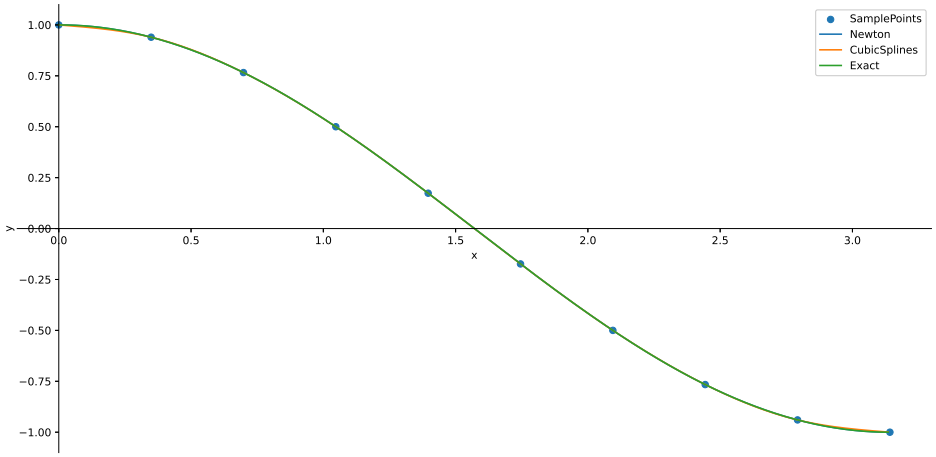

Figure 3: $f(x) = \cos x$ 在 $[0, \pi]$ 间使用两种方法的插值结果

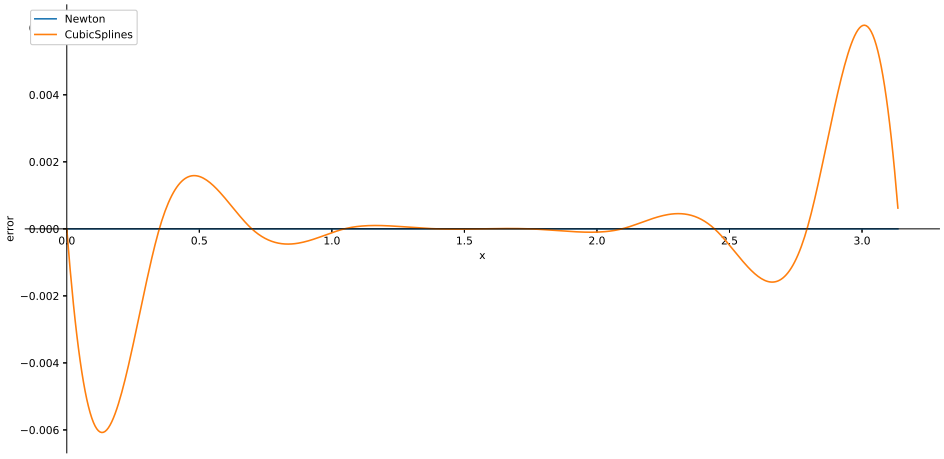

Figure 4: $f(x) = \cos x$ 在 $[0, \pi]$ 间使用两种方法的插值误差

2 题目 2：数据拟合

2.1 题目描述

1. Compute a least-squares, straight-line fit to the given data using $T(x) = a_0 + a_1 x$;

2. Compute a least-squares, parabolic-line fit to the given data using $T(x) = b_0 + b_1 x + b_2 x^2$

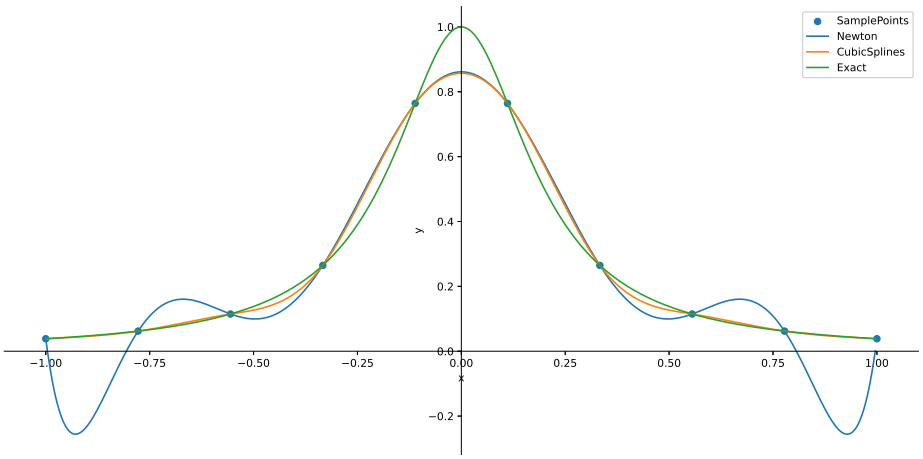

Figure 5: $f(x) = \frac{1}{1+25x^2}$ 在 $[-1, 1]$ 间使用两种方法的插值结果

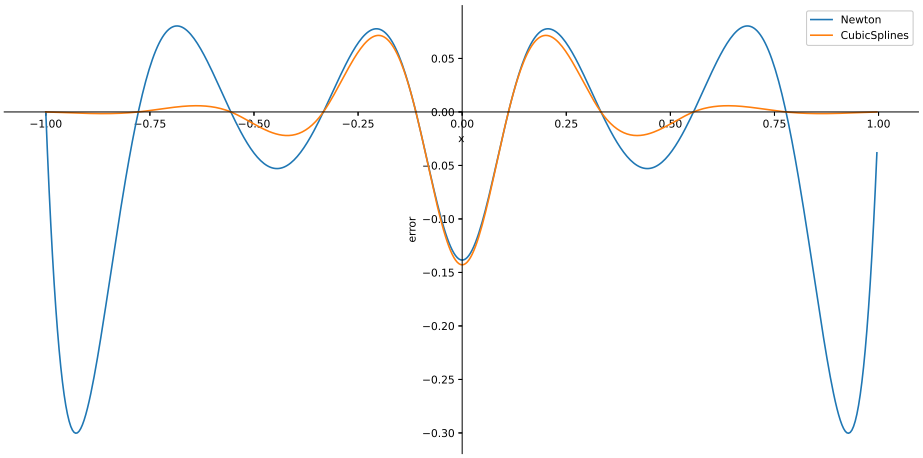

Figure 6: $f(x) = \frac{1}{1+25x^2}$ 在 $[-1, 1]$ 间使用两种方法的插值误差

2.2 程序描述

本程序按要求给出给定数据 X, T 的线性和二次曲线的最小二乘估计，输出得到的多项式系数、拟合函数图像以及残差图。

对于线性拟合 $y = a_0 + a_1 x$，直接代入公式

$$a_0 = \frac{c_0 c_3 - c_1 c_2}{c_0^2 - (n+1)c_1}, a_1 = \frac{c_0 c_2 - (n+1)c_3}{c_0^2 - (n+1)c_1} \tag{4}$$

其中 $c_0 = \sum_{i=0}^{n} x_i, c_1 = \sum_{i=0}^{n} x_i^2, c_2 = \sum_{i=0}^{n} y_i, c_3 = \sum_{i=0}^{n} x_i y_i$。

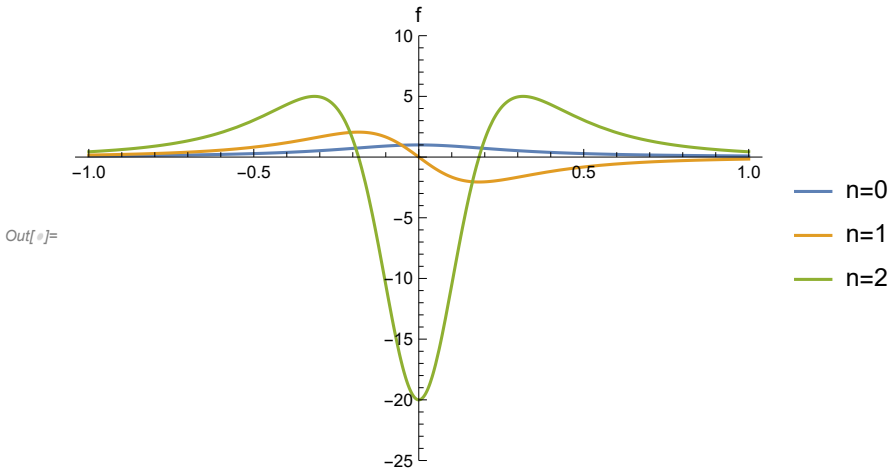

Figure 7: $f^{(n)}(\xi)$ 的函数图像（$n = 0, 1, 2$）

对于二次曲线拟合 $y = b_0 + b_1 x + b_2 x^2 = b_0 + b_1 X_1 + b_2 X_2$，最小二乘法的结果为

$$\begin{pmatrix} n & \sum_{i=1}^{n} X_{i1} & \sum_{i=1}^{n} X_{i2} \\ \sum_{i=1}^{n} X_{i1} & \sum_{i=1}^{n} X_{i1}^2 & \sum_{i=1}^{n} X_{i1}X_{i2} \\ \sum_{i=1}^{n} X_{i2} & \sum_{i=1}^{n} X_{i1}X_{ii2} & \sum_{i=1}^{n} X_{i2}^2 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} Y_i \\ \sum_{i=1}^{n} X_{i1}Y_i \\ \sum_{i=1}^{n} X_{i2}Y_i \end{pmatrix} \tag{5}$$

。对系数矩阵进行 QR 分解，得到该线性方程组的解为 $x = R^{-1}Q^T b$。

本程序源文件为 Fitting.py，运行依赖 Python 第三方库 Numpy 和 Matplotlib。在终端进入当前目录，使用命令 python -u Fitting.py 运行本程序。运行后先在控制台输出线性和二次曲线拟合函数的解析形式，之后输出 Fig.9所示的拟合函数图像以及 Fig.10所示的残差图。

2.3 伪代码

Algorithm 3 Calculating the coefficients of the parabolic fitting function

Input: The sample points X, Y

Output: The coefficients of the parabolic fitting function b_0, b_1, b_2

1: $A, b \leftarrow$the expression in Eq.5

2: $Q, R \leftarrow \mathrm{qr}(A)$

3: **return** $R^{-1}Q^T b$

2.4 输入输出示例

对给定数据分布使用线性函数和二次函数进行拟合的解析形式如 Fig.8所示，结果和残差分别如 Fig.9和 Fig.10所示。

线性拟合结果为 $y = 0.5818 + 10.1218x$，二次曲线拟合结果为 $8.2619 + 6.0517x + 0.4022x^2$。可以看出两者拟合残差分布在 ± 10 的范围内，拟合结果可以反映数据趋势。

```
Linear:y= 0.5818181818182171 + 10.121818181818178 x
Parabolic:y= 8.261904761902034 + 6.05168831168945 x+ 0.40216450216440114 x^2
```

Figure 8: 拟合结果解析形式

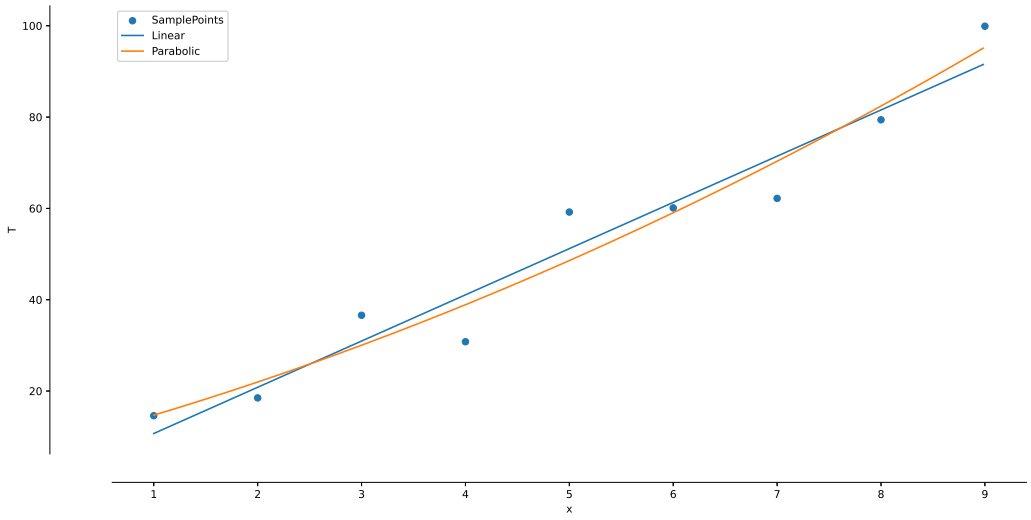

Figure 9: 线性函数和二次函数拟合结果

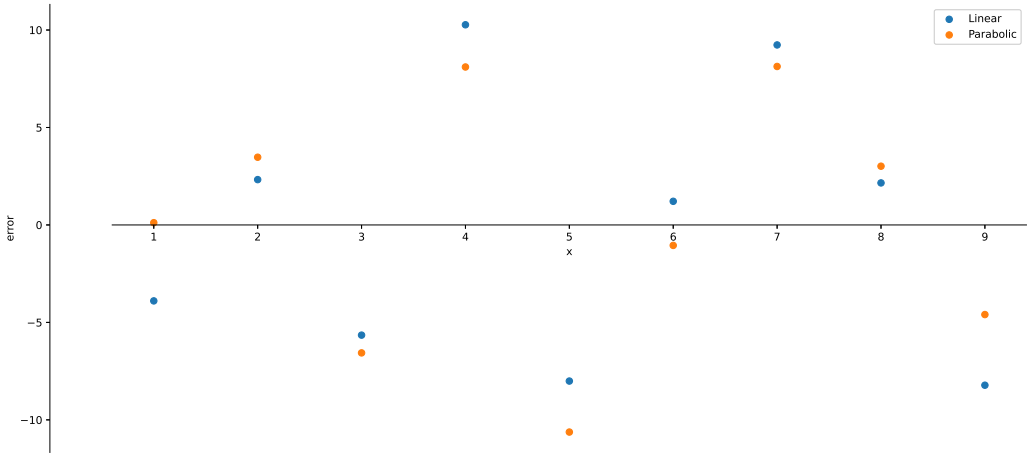

Figure 10: 线性函数和二次函数拟合残差