Numerical Analysis Project Design Report

Wu Jinhan 3220102328 *

(Information and Computing Science 2201), Zhejiang University

Due time: December 31, 2024

A. Programming Design

$$\begin{tikzcd} & & \text{class BSpline} & \\ \text{Spline} \arrow[r] & {} \arrow[ur] \arrow[r] \arrow[dr] & \text{class ppSpline} & \\ & & \text{class SplineConfigParams} \arrow[r] & \text{JSONload} \end{tikzcd}$$

class SplineConfigParams

The main implementation focuses on encapsulating spline parameters, with the core consisting of two constructor functions:

1. Manual Construction of Spline Parameters: One constructor allows the user to manually define the spline parameters.

2. JSON-Based Construction: The other constructor leverages ‘jsonspp‘ to read parameters from a JSON file, enabling the control of spline parameters through external configuration.

This dual approach ensures flexibility in defining and managing spline parameters. This approach ensures an efficient and systematic way to calculate spline values.

class ppSpline & BSpline

The main implementation methods of the spline class are all realized through these two classes. The core idea is as follows:

1. Spline Parameter Introduction: By introducing spline parameters, the process begins with constructing the parameter matrix.

2. Parameter Matrix Construction and Solving: The parameter matrix is self-constructed and solved to determine the coefficients.

3. Spline Value Calculation: When retrieving the value of the spline, back-substitution is performed using the obtained coefficients.

0.1 Auxiliary Class: PlotRW

The main functionality of this class is to export data from a vector and store the fitted data points in a CSV file. This allows the data to be easily read and processed by Python for further analysis or visualization.

*Electronic address: `3220102328@zju.edu.cn`

B. Mathematical Proof

PP-form Spline

I. Cubic Polynomial Expressions

We use the third derivative of $S(x)$, denoted as $S''(x) = M_j$ $(j = 0, 1, \ldots, n)$, to represent $S(x)$. Since $S(x)$ is a continuous function in the interval $[x_j, x_{j+1}]$, it can be expressed as:

$$S''(x) = M_j \frac{(x_{j+1} - x)^3}{6h_j} + M_j \frac{(x - x_j)^3}{6h_j}$$

For the second integral and the use of $S(x_j) = y_j$ and $S(x_{j+1}) = y_{j+1}$, the following cubic polynomial expressions are obtained:

$$S(x) = M_j \frac{(x_{j+1} - x)^3}{6h_j} + M_{j+1} \frac{(x - x_j)^3}{6h_j} + \left(y_j - \frac{M_j h_j^2}{6}\right) \frac{x_{j+1} - x}{h_j} + \left(y_{j+1} - \frac{M_{j+1} h_j^2}{6}\right) \frac{x - x_j}{h_j}, \quad j = 0, 1, \ldots, n-1$$

Here, M_j $(j = 0, 1, \ldots, n)$ is unknown. But based on the text book, we have following conclusion to determine M_j:

$$\mu_j M_{j-1} + 2M_j + \lambda_j M_{j+1} = d_j, \quad j = 1, 2, \ldots, n-1,$$

Where:

$$\mu_j = \frac{h_{j-1}}{h_{j-1} + h_j}, \quad \lambda_j = \frac{h_j}{h_{j-1} + h_j},$$
$$d_j = 6 \frac{f[x_j, x_{j+1}] - f[x_{j-1}, x_j]}{h_{j-1} + h_j} = 6f[x_{j-1}, x_j, x_{j+1}], \quad j = 1, 2, \ldots, n-1,$$

II. The construction of the matrix corresponding to the boundary conditions

For the complete boundary condition, two equations are derived:

$$2M_0 + M_1 = \frac{6}{h_0} \left(f[x_0, x_1] - f_0'\right),$$
$$M_{n-1} + 2M_n = \frac{6}{h_{n-1}} \left(f_n' - f[x_{n-1}, x_n]\right).$$

If we set $\lambda_0 = 1, d_0 = \frac{6}{h_0}\left(f[x_0, x_1] - f_0'\right), \mu_0 = 1, d_n = \frac{6}{h_{n-1}}\left(f_n' - f[x_{n-1}, x_n]\right)$, then equations can be written in matrix form:

$$\begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}$$

For the natural boundary condition, the end point equations are directly obtained:

$$M_0 = f_0'', \quad M_n = f_n''.$$

If we set $\lambda_0 = \mu_0 = 0, d_0 = 2f_0'', d_n = 2f_n''$, then equations can be written in matrix form like above.

For the periodic boundary condition, we obtain:

$$M_0 = M_n, \quad \lambda_n M_1 + \mu_n M_{n-1} + 2M_n = d_n,$$

Where:

$$\lambda_n = \frac{h_0}{h_{n-1} + h_0}, \quad \mu_n = 1 - \lambda_n,$$
$$d_n = \frac{6(f[x_0, x_1] - f[x_{n-1}, x_n])}{h_0 + h_{n-1}}.$$

Then we get the matrix form

$$\begin{pmatrix} 2 & \lambda_1 & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ \lambda_n & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}$$

B-Spline(3 degrees)

I. Node construction

Based on the textbook, we have:

$$S_n^{n-1}(t) = \sum_{i=1}^{N-2} a_i B_i^n(t),$$

and the value of each S node is determined by the following basis splines:

$$\begin{aligned}
S(t_0) \quad &\Rightarrow \quad B_{-n}^n, \dots, B_{-1}^n, \\
S(t_1) \quad &\Rightarrow \quad B_{1-n}^n, \dots, B_0^n, \\
&\vdots \\
S(t_{N-1}) \quad &\Rightarrow \quad B_{N-n-1}^n, \dots B_{N-2}^n,
\end{aligned}$$

To this end, we extend the nodes: the nodes in the forward extension are decreased by 1 incrementally, while the nodes in the backward extension are increased by 1 incrementally, in order to maintain the monotonic increase of the nodes.

II. The boundary conditions

First, we have the differentiation formula for the B-spline basis functions of order B

$$\frac{d}{dx} B_i^n(x) = \frac{nB_i^{n-1(x)}}{t_{i+n} - t_i} - \frac{nB_{i+1}^{n-1(x)}}{t_{i+n+1} - t_{i+1}}.$$

Subsequently, we can derive the first-order and second-order derivatives corresponding to the boundary conditions

$$\begin{aligned}
S''(t_0) &= \frac{6}{t_1 - t_{-1}} \left(\frac{a_{-3} - a_{-2}}{t_1 - t_{-2}} - \frac{a_{-2} - a_{-1}}{t_2 - t_{-1}} \right) B_{-1}^1(t_0), \\
S''(t_{N-1}) &= \frac{6}{t_N - t_{N-2}} \left(\frac{a_{N-4} - a_{N-3}}{t_N - t_{N-3}} - \frac{a_{N-2} - a_{N-3}}{t_{N+1} - t_{N-2}} \right) B_{N-2}^1(t_{N-1}), \\
S'(t_0) &= \frac{3(-a_{-3} + a_{-2})}{t_1 - t_{-2}} B_{-2}^2(t_0) - \frac{3(a_{-2} - a_{-1})}{t_2 - t_{-1}} B_{-1}^2(t_0), \\
S'(t_{N-1}) &= \frac{-3(a_{N-4} + a_{N-3})}{t_N - t_{N-3}} B_{N-3}^2(t_{N-1}) - \frac{3(a_{N-2} - a_{N-3})}{t_{N+1} - t_{N-2}} B_{N-2}^2(t_{N-1}).
\end{aligned}$$

Subsequently, based on the definitions of the three types of periodic conditions, the solution matrix for the coefficients a_i can be easily obtained.

Acknowledgement