求解样条插值小区间上三次多项式系数算法

Illusionna

22:57, Wednesday 21st February, 2024

Contents

I. 源

算法出自于这本书：Burden, Annette and Burden, Richard and Faires, J. Numerical Analysis, 9th ed. (2010).

样条插值原理见下面网页，网址太长了，笔者拆开后注意某些文件夹名称中含空格，如果依然搜不到，直接搜主页，然后自行找找.

https://Illusionna.readthedocs.io/zh/latest/projects/

Mathematics/Numerical Analysis/三点三次自然样条插值/Spline.html

主页：https://Illusionna.readthedocs.io

以下所有算法符号以及代码按照网页原理中统一.

II. 算法

Input: $\vec{\boldsymbol{x}} = (x_0, x_1, x_2, \ldots, x_n)$, $\vec{\boldsymbol{y}} = (y_0, y_1, y_2, \ldots, y_n)$, 点列按照自变量从小到大已经排序好了.

Output: $\forall k,\ a_k, b_k, c_k, d_k$.

$$\forall x \in [x_{k-1}, x_k], \quad s_k(x) = a_k + b_k x + c_k x^2 + d_k x^3, \quad k = 1, 2, \ldots, n$$

Step 1: 计算步长并存储在步长数组里.

$$\text{for } k,\ \text{stepArray}_k = \vec{\boldsymbol{x}}_{k+1} - \vec{\boldsymbol{x}}_k$$

Step 2: 计算 α 并存在 αArray 数组里.

$$\text{for } k,\ \alpha\text{Array}_k = \frac{3}{\text{stepArray}_k} \times (\vec{\boldsymbol{y}}_{k+1} - \vec{\boldsymbol{y}}_k) - \frac{3}{\text{stepArray}_{k-1}} \times (\vec{\boldsymbol{y}}_k - \vec{\boldsymbol{y}}_{k-1})$$

Step 3: 给 lArray, μArray, zArray 数组首索引元素设置初值.

$$l\text{Array}_0 = 1, \quad \mu\text{Array}_0 = 0, \quad z\text{Array}_0 = 0$$

Step 4: 对 lArray, μArray, zArray 数组更新.

$$\text{for } k,\ l\text{Array}_k = 2(\vec{\boldsymbol{x}}_{k+1} - \vec{\boldsymbol{x}}_{k-1}) - \text{stepArray}_{k-1} \times \mu\text{Array}_{k-1}$$

$$\mu\text{Array}_k = \frac{\text{stepArray}_k}{l\text{Array}_k}$$

$$z\text{Array}_k = \frac{\alpha\text{Array}_k - \text{stepArray}_{k-1} \times z\text{Array}_{k-1}}{l\text{Array}_k}$$

Step 5: 再引入 cArray 数组并给尾索引赋值.

$$l\text{Array}_{n+1} = 1, \quad z\text{Array}_{n+1} = 0, \quad c\text{Array}_{n+1} = 0$$

Step 6: 再引入 bArray, dArray 数组并计算多项式各次项系数.

$$\text{for } t = n, n-1, \ldots, 0$$

$$c\text{Array}_t = z\text{Array}_t - \mu\text{Array}_t \times c\text{Array}_{t+1}$$

$$b\text{Array}_t = \frac{\vec{\boldsymbol{y}}_{t+1} - \vec{\boldsymbol{y}}_t}{\text{stepArray}_t} - \frac{\text{stepArray}_t(c\text{Array}_{t+1} + 2 \times c\text{Array}_t)}{3}$$

$$dArray_t = \frac{cArray_{t+1} - cArray_t}{3 \times stepArray_t}$$

Step 7: 最后打印 $\vec{y}$, bArray, cArray, dArray.

III. C 代码

```c
/*
System ——> Linux & gcc8.1.0
File ————> NaturalCubicSpline.c
Author ——> Illusionna
Create ——> 2024/2/21 22:16:30
'''
—*— Encoding: UTF—8 —*—
*/


# include <stdio.h>

int main(){
    printf("\033[H\033[J");
    // ********************************************************
    double X[] = {3, 4.5, 7, 9};
    double Y[] = {2.5, 1, 2.5, 0.5};
    // ********************************************************
    int i, j;
    int lengthX = sizeof(X) / sizeof(X[0]);
    int lengthY = sizeof(Y) / sizeof(Y[0]);
    if (lengthX == lengthY){
        int n = lengthX - 1;
        double stepArray[n], alphaArray[n], lArray[n+1], muArray[n
            +1], zArray[n+1], cArray[n+1], bArray[n], dArray[n];
        // Step 1.
```

```
26          for (i=0; i<n; ++i){
27              stepArray[i] = X[i+1] - X[i];
28          }
29          // Step 2.
30          for (i=1; i<n; ++i){
31              alphaArray[i] = (3 * (Y[i+1] - Y[i]) / stepArray[i]) - (3
                    * (Y[i] - Y[i-1]) / stepArray[i-1]);
32          }
33          // Step 3.
34          lArray[0] = 1;
35          muArray[0] = 0;
36          zArray[0] = 0;
37          // Step 4.
38          for (i=1; i<n; ++i){
39              lArray[i] = 2 * (X[i+1] - X[i-1]) - stepArray[i-1] *
                    muArray[i-1];
40              muArray[i] = stepArray[i] / lArray[i];
41              zArray[i] = (alphaArray[i] - stepArray[i-1] * zArray[i
                    -1]) / lArray[i];
42          }
43          // Step 5.
44          lArray[n] = 1;
45          zArray[n] = 0;
46          cArray[n] = 0;
47          // Step 6.
48          for (j=n-1; j>=0; --j){
49              cArray[j] = zArray[j] - muArray[j] * cArray[j+1];
50              bArray[j] = ((Y[j+1] - Y[j]) / stepArray[j]) - (
                    stepArray[j] * (cArray[j+1] + 2 * cArray[j]) / 3);
51              dArray[j] = (cArray[j+1] - cArray[j]) / (3 * stepArray[j
                    ]);
52          }
```

```
53            // Information.
54            printf("The coefficients of each interval cubic polynomial:\n
                  ");
55            printf("%2s %8s %8s %8s %8s\n", "k", "ak", "bk", "ck", "dk
                  ");
56            for (i=0; i<n; ++i){
57                printf("%2d %9.5f %8.5f %9.5f %9.5f\n", i+1, Y[i],
                      bArray[i], cArray[i], dArray[i]);
58            }
59        }
60        else{
61            // Warning...
62            printf("\033[31mError\033[0m: length of X is\033[31m %d
                  \033[0m, but length of Y is\033[31m %d\033[0m.\n",
                  lengthX, lengthY);
63            printf("\033[33mCheck array!\033[0m");
64        }
65        return 0;
66    }
```