

数值实验报告

杜鸿宇

2016141211049

指导老师：朱瑞

目录

加权样条的原理和构造方法:.....	3
参数的选择:	5
本论文中的 B-spline 介绍:	6
控制多边形:.....	8
论文实例模拟:	10
第一张图 (M 值):	10
第二张图 (M 值):	11
第三张图 (M 值):	11
第四张图 (M 值):	11
不同结点比较:	12
第一张图:.....	12
第二张图:.....	12
第三张图:.....	13
第四张图:.....	13
加权样条分析 (包含边界分析和权值不同分析):.....	14
第一张图:	14
第二张图:	16
第三张图:	17
第四张图:	18
连续性分析:	19
不加权重原图模拟代码:.....	19
第一张图:.....	19
第二张图:.....	21
第三张图:.....	23
第四张图:.....	26
不加权重结点比较图:	28
第一张图:	28
第二张图:	29
第三张图:	29
第四张图:	30
加权重代码原图 (包含对不同结点比较, 不同权值不再另附代码, 只需在源代码中给出新的 a 和 c 的值即可, 首先给出 dividif 的代码):	30
第一张图:.....	31
第二张图:.....	32
第三张图:.....	34
第四张图:.....	36
控制多边形代码:	38

本篇论文主要介绍加权样条的构造方法，性质，以及关于它的一些实例。接下来我将在本报告中重述加权样条的构造过程和这些性质。

加权样条的原理和构造方法：

加权样条强调的是为了保证插值出来图像的单调性和凹凸性，所以构造方法将从这一点入手。

首先给出一些证明过程中符号的表示：

$(x_i, f_i), i = 0, \dots, N+1$ 代表 $N+2$ 个插值点

$f[x_i, x_{i+1}] = (f_{i+1} - f_i) / h_i, h_i = x_{i+1} - x_i, i = 0, \dots, N$ 是两点的差商

单调性和凹凸性的满足分别对应于：

$f[x_i, x_{i+1}] \geq 0, i = 0, \dots, N$ 和 $f[x_{i-1}, x_i, x_{i+1}] \geq 0, i = 1, \dots, N$ ；

我们规定权的符号为：

$$w(x), 0 < m \leq w(x) \leq M ;$$

它是满足以下这个微分多点边值问题的解：

$$\frac{d^2}{dx^2} \left(w(x) \frac{d^2 S}{dx^2} \right) = 0, x \in (x_i, x_{i+1}), i = 0, \dots, N, S \in C^k[a, b], k \geq 1 ;$$

接下来我们考虑在区间 $[a, b]$ 上的初始边值条件，分成以下的几种情况：

第一类： $S'(a) = f'_0$ 和 $S'(b) = f'_{N+1}$ ；

第二类： $S''(a) = f''_0$ 和 $S''(b) = f''_{N+1}$ ；

周期型： $S^{(r)}(a) = S^{(r)}(b), r = 0, 1, w_0 S''(x_0^+) = w_N S''(x_{N+1}^-)$ ；

Not-a-knot： $S'''(x_i^-) = S'''(x_i^+)$ ；

我们再给出一些记号：

$$M_i = w_{i-1} S''(x_i^-) = w_i S''(x_i^+), i = 1, \dots, N ;$$

$$M_0 = w_0 S''(x_0^+), M_{N+1} = w_N S''(x_{N+1}^-);$$

那么通过对转角的运算我们得到:

$$S(x) = f_i(1-t) + f_{i+1}t - t(1-t)\frac{h_i^2}{6w_i}[(2-t) + (1+t)M_{i+1}];$$

其中:

$$t = (x - x_i) / h_i, h_i = x_{i+1} - x_i;$$

为了找到 M_i , 我们对上式求导, 这样就可以得到:

$$S'(x) = f[x_i, x_{i+1}] - \frac{h_i}{6w_i}[(2-6t+3t^2)M_i + (1-3t^2)M_{i+1}];$$

又因为:

$$S'(x_i^-) = S'(x_i^+);$$

所以我们可以得到:

$$\frac{h_{i-1}}{w_{i-1}}M_{i-1} + 2\left(\frac{h_{i-1}}{w_{i-1}} + \frac{h_i}{w_i}\right)M_i + \frac{h_i}{w_i}M_{i+1} = 6\delta_i f, i = 1, \dots, N;$$

其中:

$$\delta_i f = f[x_i, x_{i+1}] - f[x_{i-1}, x_i];$$

接下来我们考虑三次样条插值的加权法。

我们定义在区间 $[x_i, x_{i+1}]$ 上:

$$m_i = S'(x_i), i = 0, \dots, N+1;$$

这样我们就可以得到:

$$S(x) \equiv S_i(x) = f_i(1-t)^2(1+2t) + f_{i+1}t^2(3-2t) + m_i h_i t(1-t)^2 - m_{i+1} h_i t^2(1-t), x \in [x_i, x_{i+1}]$$

通过二次求导就可以得到:

;

$$\text{再由条件: } S''_i(x) = \frac{2}{h_i}(3(1-2t)f[x_i, x_{i+1}] - (2-3t)m_i - (1-3t)m_{i+1})$$

$$w_{i-1}S''(x_i^-) = w_iS''(x_i^+);$$

得到:

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = 3\lambda_i f[x_{i-1}, x_i] + 3\mu_i f[x_i, x_{i+1}], i = 1, \dots, N;$$

其中:

$$\lambda_i = \frac{w_{i-1} h_i}{w_{i-1} h_i + w_i h_{i-1}}, \mu_i = 1 - \lambda_i;$$

与一般样条的区别就在于这个三转角公式中是含有权值的。

接着, 给出两个误差估计公式, 证明过程省略。

若: $S \in C^1[a, b]$, 且在第一类边界条件的情况下, 若满足 $S'(x_0) = f'_0$ 和 $S'(x_{N+1}) = f'_{N+1}$,

那么对于 $f \in C^2[a, b]$ 和 $f \in C^4[a, b]$, 分别满足:

$$\|S^{(r)}(x) - f^{(r)}(x)\|_c \leq C_r \bar{h}^{-2-r} \|f''\|_c, r = 0, 1, \quad C_0 = 13/48, C_1 = 0.86229, \bar{h} = \max_i h_i;$$

$$\|S^{(r)}(x) - f^{(r)}(x)\|_c \leq C_r \bar{h}^{-2-r} \|f''\|_c + \tilde{C}_r \bar{h}^{-4-r} \|f^{IV}\|_c, r = 0, 1;$$

$$C_0 = 1/8, C_1 = 1/2, \tilde{C}_0 = 5/384, \tilde{C}_1 = 1/24.$$

接着我们将看到这篇论文的核心思想: 如何通过 $w(x)$ 来控制单调性和凹凸性。

参数的选择:

我们给出如下的两个定理:

第一: 若 $S \in C^1[a, b]$ 且有第一类边界条件: $S'(x_0) = f'_0$ 和 $S'(x_{N+1}) = f'_{N+1}$, 如果如下三个式子是成立的:

$$\begin{aligned} 0 \leq f'_0 \leq 3f[x_0, x_1], 0 \leq f'_{N+1} \leq 3f[x_N, x_{N+1}], \\ \lambda_i f[x_{i-1}, x_i] \leq (1 + \lambda_i) f[x_i, x_{i+1}], \\ \mu_i f[x_i, x_{i+1}] \leq (1 + \mu_i) f[x_{i-1}, x_i], i = 1, \dots, N \end{aligned};$$

那么: $S'(x) \geq 0, x \in [a, b]$, S 在 $[a, b]$ 上是单调的。

第二: 若三次权重样条 $S \in C^1[a, b]$ 满足第二类边界条件, 那么当:

$$f''_0 \geq 0, f''_{N+1} \geq 0, 2d_i - \mu_i d_{i-1} - \lambda_i d_{i+1} \geq 0, i = 1, \dots, N;$$

$$\text{其中: } d_0 = 2w_0 f''_0, d_{N+1} = 2w_N f''_{N+1}, \text{ 那么: } S''(x) \geq 0, x \in [a, b]$$

S 是凹凸的。

我们给出满足上述两个条件的一个权重 $w(x)$:

$$w(x) = [1 + C_i (f[x_i, x_{i+1}])^2]^{-\alpha_i}, C_i \geq 1, \alpha_i \geq 0, i = 0, \dots, N;$$

为了能够很好的给出一组所求的 C_i 和 α_i ，我们需要 $w(x)$ 满足如下的公式：

$$\frac{w_{i-1}}{w_i} \frac{h_i}{h_{i-1}} \geq \frac{f[x_i, x_{i+1}]}{f[x_{i-1}, x_i]}, \quad \frac{w_i}{w_{i-1}} \frac{h_{i-1}}{h_i} \geq \frac{f[x_{i-1}, x_i]}{f[x_i, x_{i+1}]}, \quad i = 1, \dots, N$$

这个公式是由之前的式子得到的：

$$\begin{aligned} 0 \leq f'_0 \leq 3f[x_0, x_1], 0 \leq f'_{N+1} \leq 3f[x_N, x_{N+1}], \\ \lambda_i f[x_{i-1}, x_i] \leq (1 + \lambda_i) f[x_i, x_{i+1}], \\ \mu_i f[x_i, x_{i+1}] \leq (1 + \mu_i) f[x_{i-1}, x_i], i = 1, \dots, N \end{aligned}$$

它可以保证第一类边值条件和第二类边值条件情况下的单调性。而保证凸性的式子又可以化为：

$$\begin{aligned} 0 \leq w_0 f''_0 \leq d_1, 0 \leq w_N f''_{N+1} \leq d_N, \\ d_{i-1} / 2 \leq d_i \leq 2d_{i-1}, i = 2, \dots, N. \end{aligned}$$

即为：

$$\begin{aligned} 0 < f''_0 < 6\delta_1 f / h_0, 0 < f''_{N+1} < 6\delta_N f / h_N, \\ \frac{w_0}{w_1} \frac{h_1}{h_0} \leq \frac{6\delta_1 f}{h_0 f''_0} - 1, \\ \frac{1}{2} \frac{\delta_i f}{\delta_{i-1} f} - 1 \leq \frac{w_{i-1}}{w_i} \frac{h_i}{h_{i-1}} \leq 2 \frac{\delta_i f}{\delta_{i-1} f} - 1, i = 2, \dots, N, \\ \frac{w_N}{w_{N-1}} \frac{h_{N-1}}{h_N} \leq \frac{6\delta_N f}{h_N f''_{N+1}} - 1. \end{aligned}$$

这样也保证了凸性。由上面这些式子综合起来的 a 和 c 就是符合题意的 a 和 c 。接着用递归算法就可以求出 a 和 c 的所有值。本文后面的例子中所用参数是经过多次调试之后得到的，满足不等式要求。而递归算法的具体实现过程受编程水平影响没能给出。

本论文中的 B-spline 介绍：

B 样条中对于 $0, \dots, N+1$ 个点，我们需要考虑如下的点： $-3, -2, -1, 0, \dots, N, N+1, N+2, N+3, N+4$ 这是实际起作用的点，注意这只是点的下标而不是真正意义上的点。接着我们有如下的性质：

$$\begin{aligned} B_i(x) > 0, x \in (x_{i-2}, x_{i+2}), \\ B_i(x) \equiv 0, x \notin (x_{i-2}, x_{i+2}), \\ \sum_{j=-1}^{N+2} B_j(x) \equiv 1, x \in [a, b]. \end{aligned}$$

加上权重之后，式子可以写为：

$$S(x)=[S(x_i)-\phi_i(x)M_i](1-t)+[S(x_{i+1})-\psi_i(x_{i+1})M_{i+1}]t+\phi_i(x)M_i+\psi_i(x)M_{i+1},$$

$$\phi_i(x)=\frac{(x_{i+1}-x)^3}{6h_iw_i},\psi_i(x)=\frac{(x-x_i)^3}{6h_iw_i},$$

$$z_j^r\equiv z_j^{(r)}(x_j)=\psi_{j-1}^{(r)}(x_j)-\phi_j^{(r)}(x_j),r=0,1;y_j=x_j-\frac{z_j}{z_j},$$

因此我们就得到:

$$B_i(x)=\begin{cases}\psi_{i-2}(x)M_{i-1,B_i},x\in[x_{i-2},x_{i-1}),\\ \frac{x-y_{i-1}}{y_i-y_{i-1}}+\phi_{i-1}(x)M_{i,B_i}+\psi_{i-1}(x)M_{i,B_i},x\in[x_{i-1},x_i),\\ \frac{y_{i+1}-x}{y_{i+1}-y_i}+\phi_i(x)M_{i,B_i}+\psi_i(x)M_{i+1,B_i},x\in[x_i,x_{i+1}),\\ \phi_{i+1}(x)M_{i+1,B_i},x\in[x_{i+1},x_{i+2}),\\ 0,其它\end{cases}$$

其中:

$$M_{j,B_i}=\frac{y_{i+1}-y_{i-1}}{z_jw_{i-1}(y_j)},j=i-1,i,i+1,\\ w_{i-1}(x)=(x-y_{i-1})(x-y_i)(x-y_{i+1}),$$

接着我们又得到:

$$B_{j,1}(x)=\begin{cases}\psi_j''(x),x_j\leq x<x_{j+1},\\ \phi_{j+1}''(x),x_{j+1}\leq x<x_{j+2},\\ 0,其它\end{cases},$$

其自身拥有相加唯一性和递归性:

$$\sum_{j=0}^N w_j[B_{j-1,1}(x)+B_{j,1}(x)]\equiv 1,x\in[a,b],\\ B_{j,k}(x)=\int_{x_j}^x \frac{B_{j,k-1}(\tau)}{c_{j,k-1}}d\tau-\int_{x_{j+1}}^x \frac{B_{j+1,k-1}(\tau)}{c_{j+1,k-1}}d\tau,\\ c_{j,k-1}=\int_{x_j}^{x_{j+k}} B_{j,k-1}(\tau)d\tau,j=i-2,i-k+1,k=2,3\\ c_{j,1}=z_{j+1}',j=i-2,i-1,i;c_{j,2}=y_{j+2}-y_{j+1},j=i-2,i-1$$

通过求导:

$$B'_{j,k}(x) = B_{j,k-1}(x)/c_{j,k-1} - B_{j+1,k-1}(x)/c_{j+1,k-1}, k = 2, 3$$

我们得到：

$$B_{j,2} = \begin{cases} \psi'_j(x)/c_{j,1}, x_j \leq x < x_{j+1}, \\ 1 + \phi'_{j+1}(x)/c_{j,1} - \psi'_{j+1}(x)/c_{j+1,1}, x_{j+1} \leq x < x_{j+2}, \\ -\phi'_{j+2}(x)/c_{j+1,1}, x_{j+2} \leq x < x_{j+3}, \\ 0, \text{其它} \end{cases}$$

当 b 已知的时候我们得到：

$$S(x) = b_i + \tilde{\Delta}_i b(x - y_i) + c_i \phi_i(x) + c_{i+1} \psi_i(x)$$

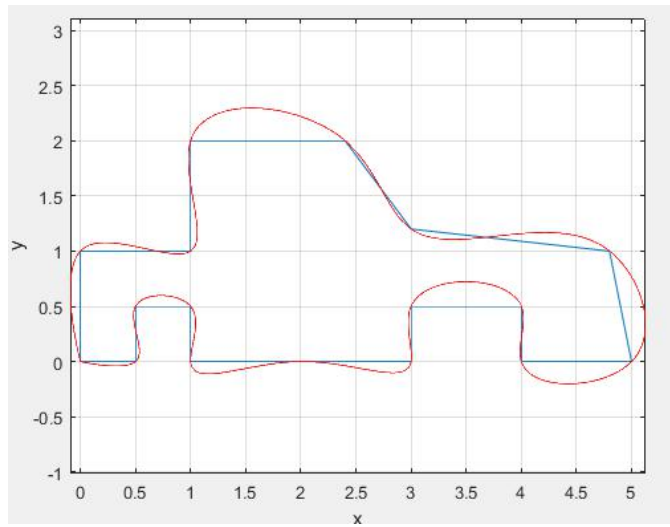
其中：

$$c_j = \frac{\tilde{\Delta}_j b - \tilde{\Delta}_{j-1} b}{c_{j-1,1}}, j = i, i+1, \tilde{\Delta}_j b = \frac{b_{j+1} - b_j}{c_{j-1,2}}$$

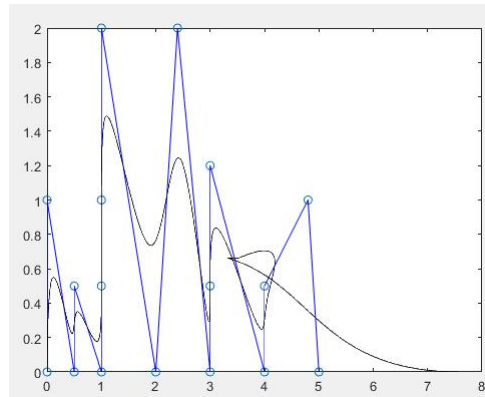
$$b_j = f_j - \frac{1}{c_{j-1,1}} [\psi_{j-1}(x_j) \frac{f_{j+1} - f_j}{h_j} - \phi_j(x_j) \frac{f_j - f_{j-1}}{h_{j-1}}]$$

控制多边形：

以下是控制多边形不加权重的具体图像，第一个图是本人自己编程独立完成，后面三个图在同学允许且帮助之下完成：

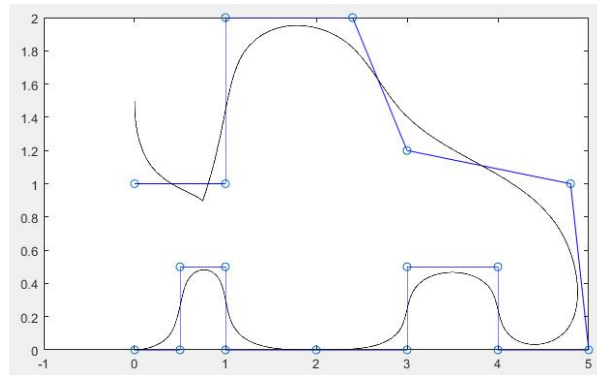



```
x=[0.0 0.0 0.5 0.5 1.0 1.0 1.0 1.0 2.0 2.4 3.0 3.0 3.0 4.0 4.0 4.8 5.0];
y=[0.0 1.0 0.0 0.5 0.0 0.5 1.0 2.0 0.0 2.0 0.0 0.5 1.2 0.0 0.5 1.0 0.0];
```

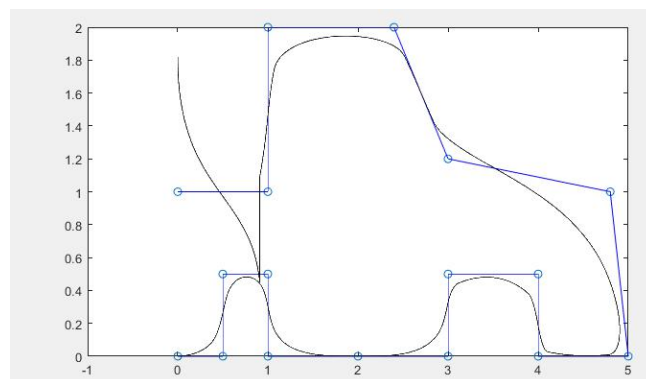


这是一张改变结点之后出错的图像。

```
x=[0.0 0.5 0.5 1.0 1.0 2.0 3.0 3.0 4.0 4.0 5.0 4.8 3.0 2.4 1.0 1.0 0.0];
y=[0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.5 0.5 0.0 0.0 1.0 1.2 2.0 2.0 1.0 1.0];
w=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
```



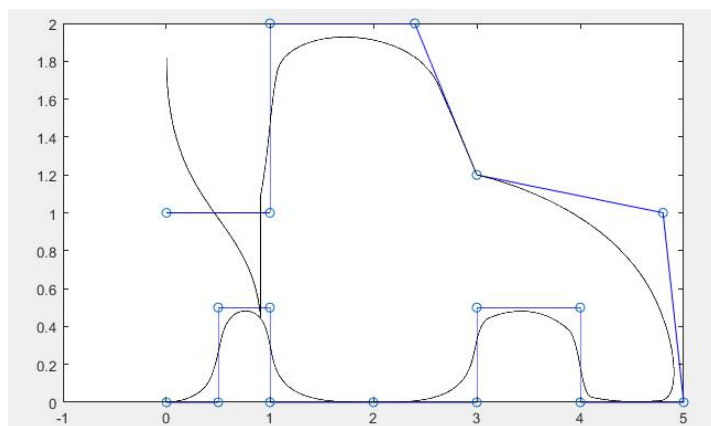
```
x=[0.0 0.5 0.5 1.0 1.0 2.0 3.0 3.0 4.0 4.0 5.0 4.8 3.0 2.4 1.0 1.0 0.0];
y=[0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.5 0.5 0.0 0.0 1.0 1.2 2.0 2.0 1.0 1.0];
w=[1 1 1 1 1 1 1 8 1 40 1 1 40 1 8 1 1];
```



```

x=[0.0 0.5 0.5 1.0 1.0 2.0 3.0 3.0 4.0 4.0 5.0 4.8 3.0 3.0 2.4 1.0 1.0 0.0];
y=[0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 1.0 1.2 1.2 2.0 2.0 1.0 1.0];
w=[1 1 1 1 1 1 1 8 1 40 1 1 40 40 1 8 1 1];

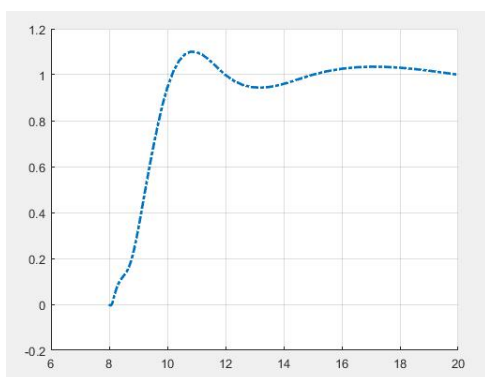
```



论文实例模拟：

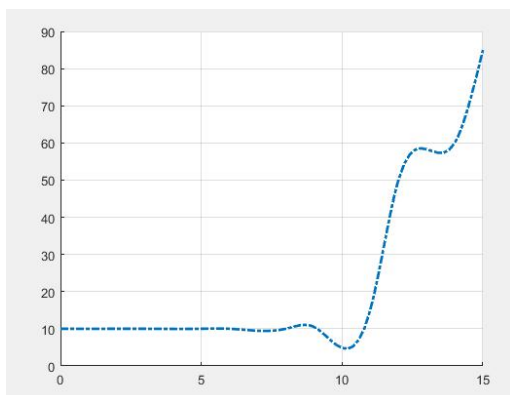
我们给出以上四张图的具体数值结果（在不加权重的情况下的数值结果）由于这是一般的三弯矩方程得到的图像，所以我们只给出四张图片相对应的 M 的值，具体函数式可由分段函数直接表示出来。而边值条件只取题目中已给的边值条件，不同边值条件的图像比较将在下面给出。代码也将在后面给出。

第一张图（ M 值）：



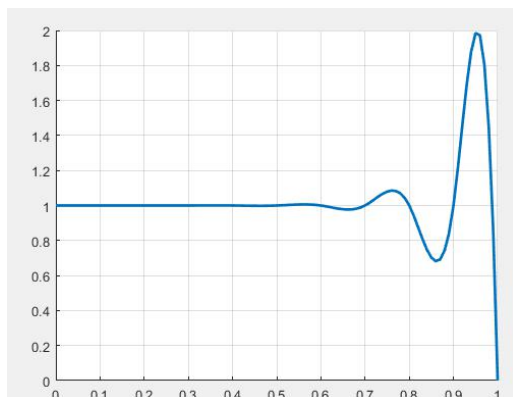
Variable	Value
M0	0.000000
M1	1.593931
M2	-0.4063234
M3	0.2844177
M4	-0.2553947E-01
M5	-0.1041731
M6	0.1923138E-01
M7	-0.3655883E-02
M8	0.000000

第二张图（M 值）：



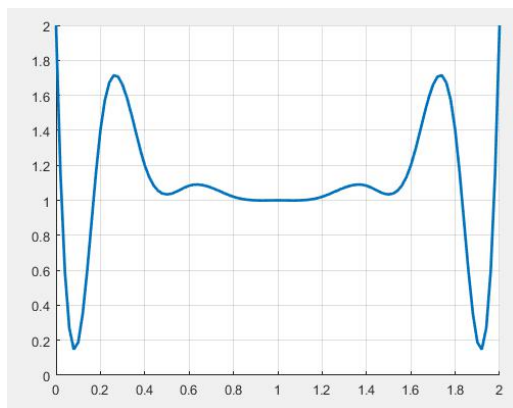
Variable	Value
M0	0.000000
M1	0.5176154E-02
M2	-0.3137063E-01
M3	0.9109422E-01
M4	-0.4883943
M5	1.413026
M6	-2.511600
M7	7.671943
M8	-8.316404
M9	6.120995
M10	0.000000

第三张图（M 值）：



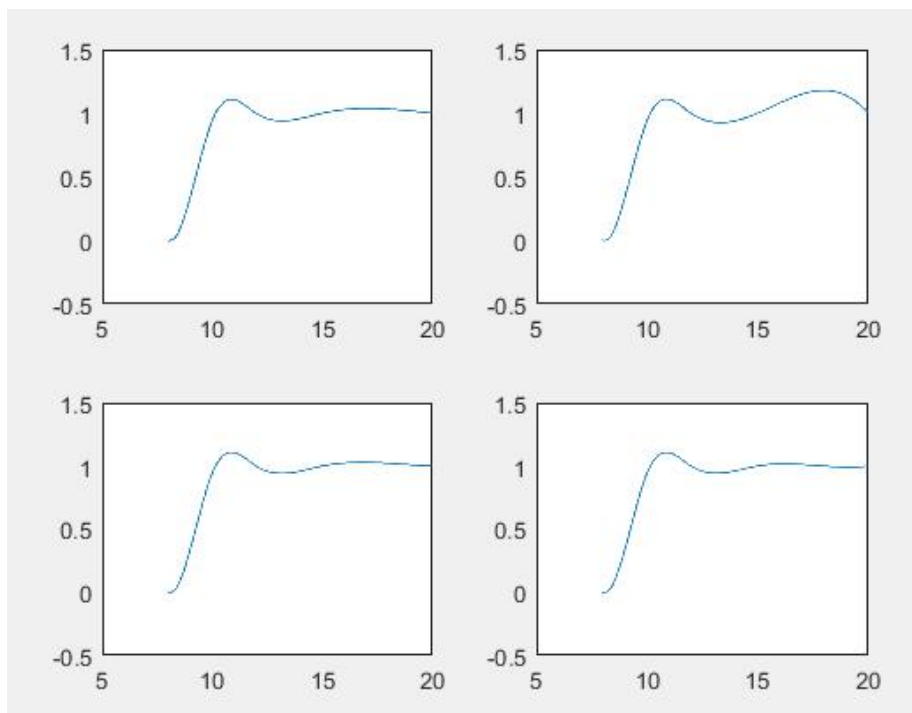
Variable	Value
M0	0.4648776E-02
M1	0.1859510E-01
M2	-0.7902919E-01
M3	0.2975217
M4	-1.111057
M5	4.146708
M6	-15.47578
M7	57.75639
M8	-215.5498
M9	804.4428
M10	-3102.221

第四张图（M 值）：



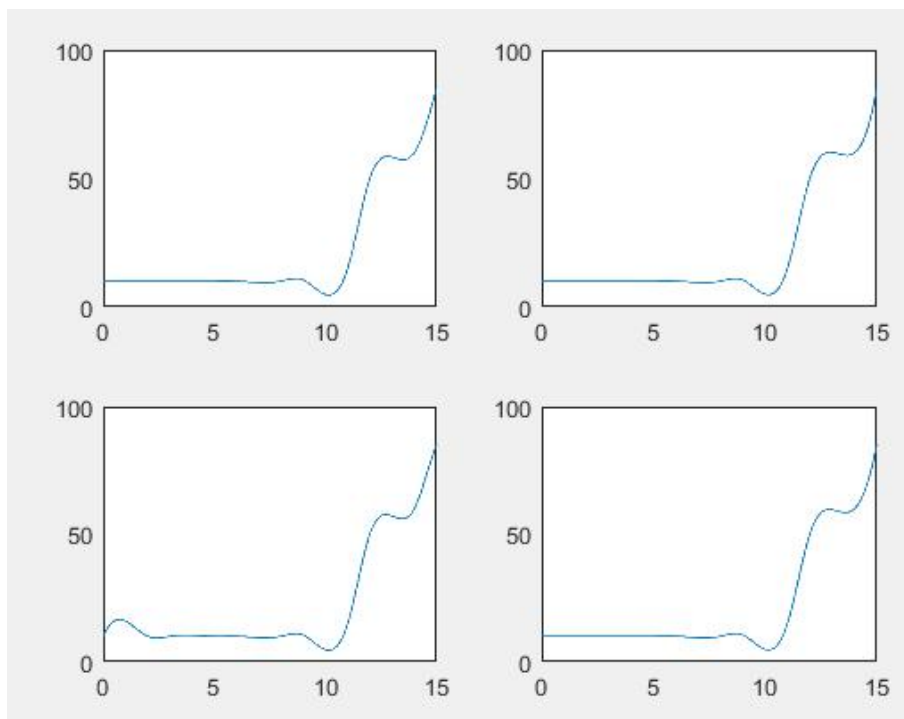
Variable	Value
M0	812.5943
M1	-215.1886
M2	58.15994
M3	-15.36382
M4	4.625350
M5	-2.060175
M6	4.625350
M7	-15.36382
M8	58.15994
M9	-215.1886
M10	812.5943

接下来考虑在不同边界条件下的图像，即不同端点条件对插值结果的影响：
第一张图：



从左到右从上到下依次为：自然型、Not-a-knot 型、周期型和第一类型边界条件下的图像。

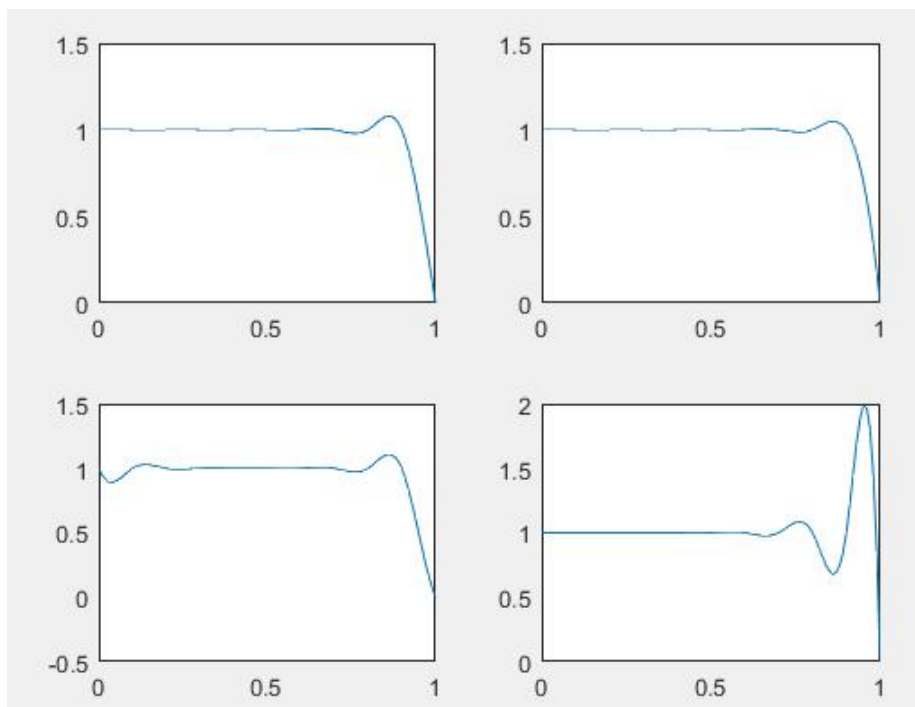
第二张图：



从左到右从上到下依次为：自然型、Not-a-kont 型、周期型和第一类型边界条件下的图

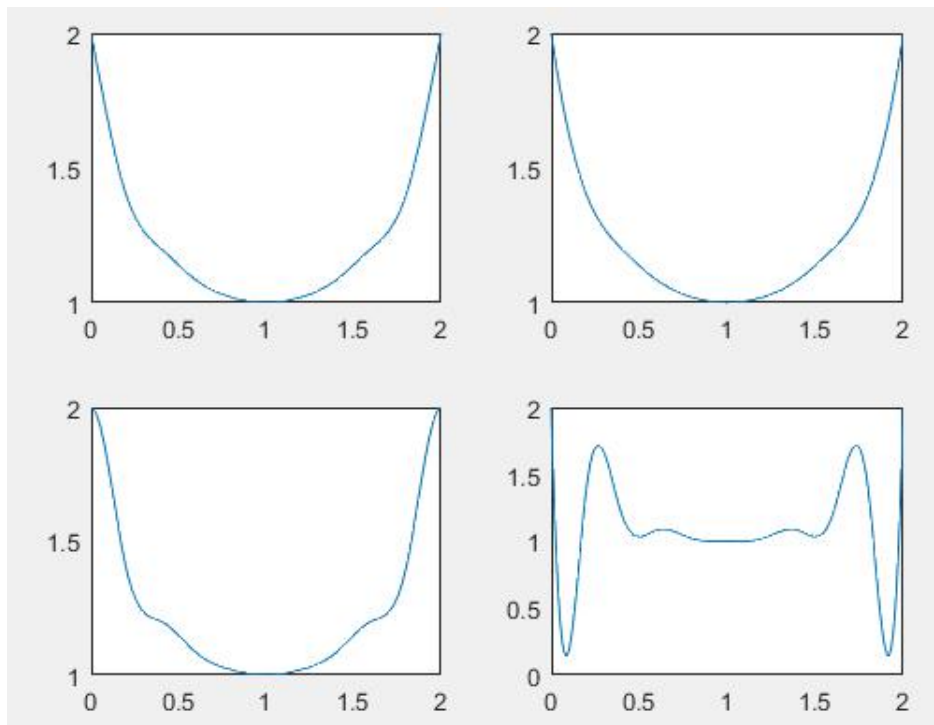
像。

第三张图：



从左到右从上到下依次为：自然型、Not-a-kont 型、周期型和第一类型边界条件下的图像。

第四张图：



从左到右从上到下分别是：自然边界条件，not-a-knot 条件，周期边界条件和第一类边界条件。

加权样条分析：

由于加权样条的参数 a_i 和 C_i 未知，所以我也做到尽可能的去估计它。为了配合上替的代码在接下来的几张图中的运算，代码中出现的 $a = [2, 4, 6, 8, 10]$ 表示的不是 a 取值为 2, 4, 6, 8, 10, 而是前后两项之差才是真正的 a 值。

第一张图：

我们取：

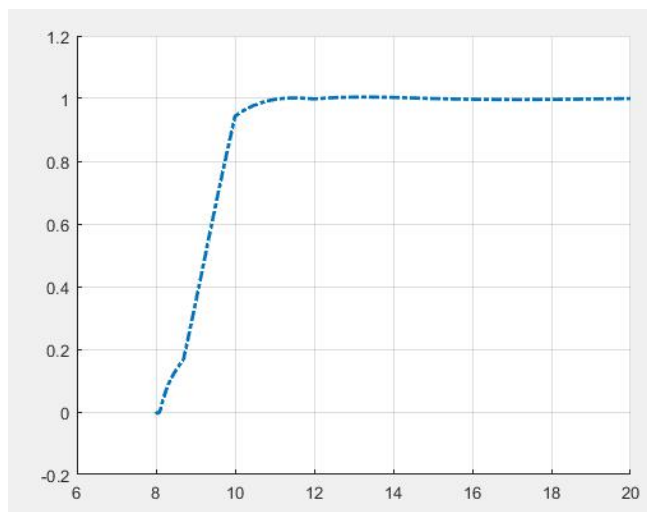
```
a=[5 10 15 18 19 20 25 27 28];
c=[0 0 0 1 4 6 12 14 16];
```

因此得到：

$$a_0 = 5, a_1 = 5, a_2 = 5, \dots, a_8 = 1$$

$$c_0 = 0, c_1 = 0, c_2 = 0, \dots, c_8 = 2,$$

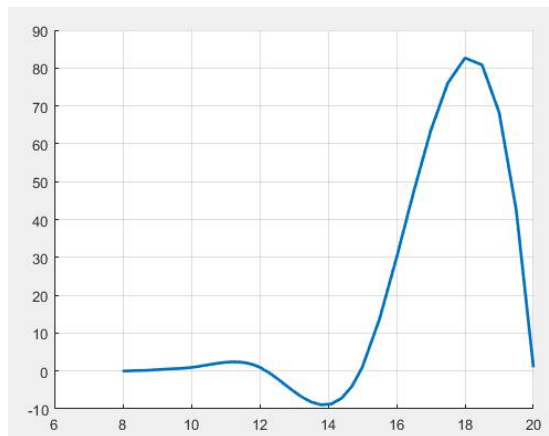
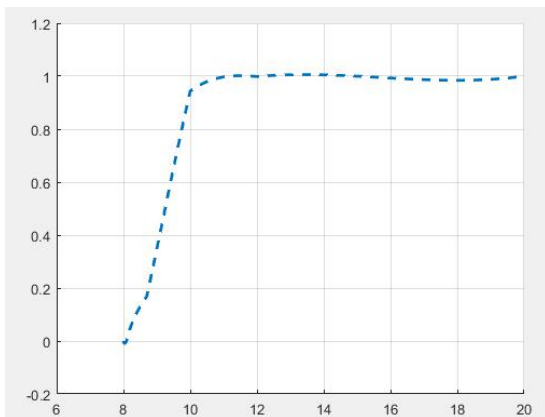
因此我们得到第一张图为：



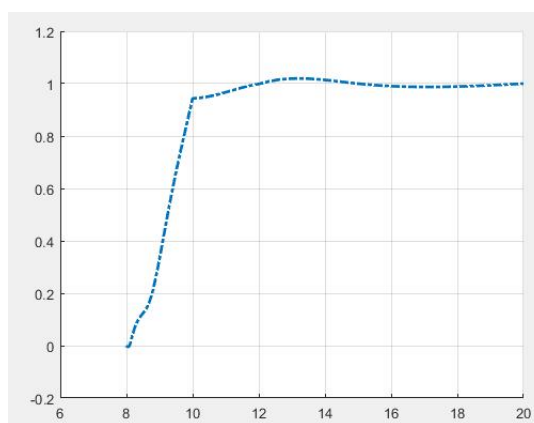
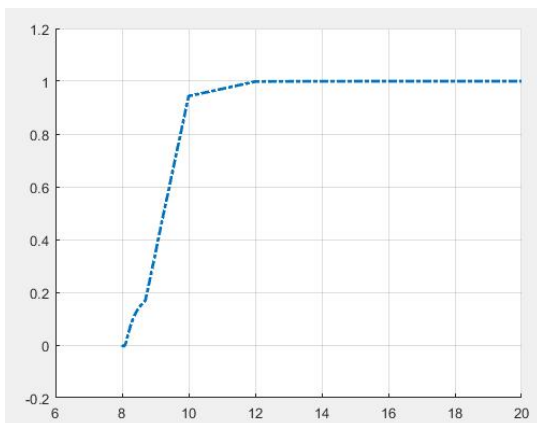
这张图和原文中的图基本相同。根据题意，这是在自然边界条件下得到的图像。接下来在数据不变的情况下给出另外两个边值条件下的图像：

Not-a-Knot:

第一类边界条件:



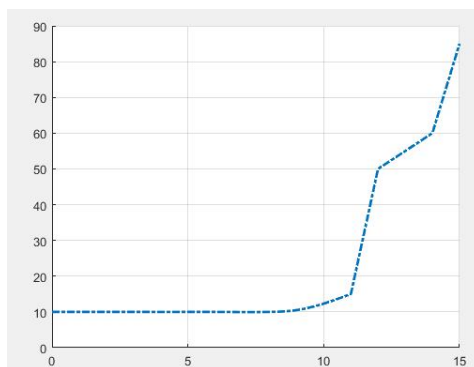
接着把权值改为: $a=[0 \ 0 \ 15 \ 18 \ 19 \ 22 \ 25 \ 25 \ 25]$; $c=[0 \ 0 \ 0 \ 0 \ 4 \ 11 \ 12 \ 14 \ 16]$;
和 $a=[0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 3 \ 12 \ 14]$; $c=[0 \ 0 \ 0 \ 0 \ 0 \ 2 \ 4 \ 9 \ 12]$;



可以看出改变权值虽然对图像整体没有太大改观，但是能够很好地改变原图像的光滑性，而改变插值边界条件就会大幅度地改动原来图像的单调性和凹凸性。

第二张图:

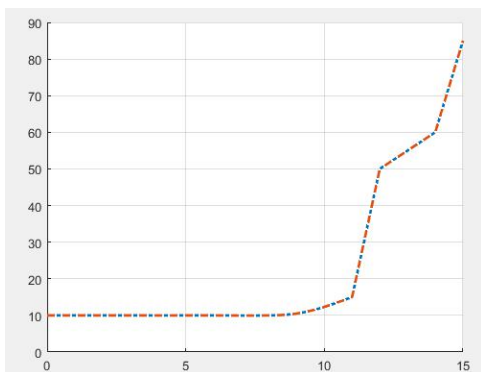
$a=[0\ 0\ 0\ 0\ 9\ 11\ 12\ 12\ 17\ 22\ 24];c=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 4\ 12\ 21\ 23];$



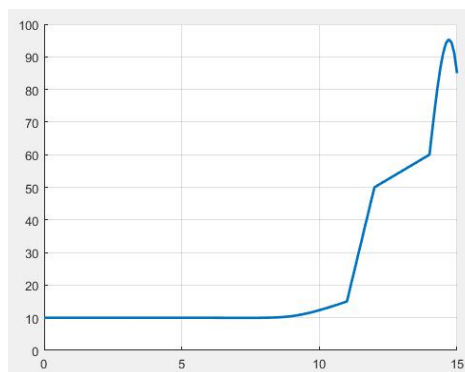
此图与原论文的图近乎一致。

改变边界条件:

Not-a-knot:

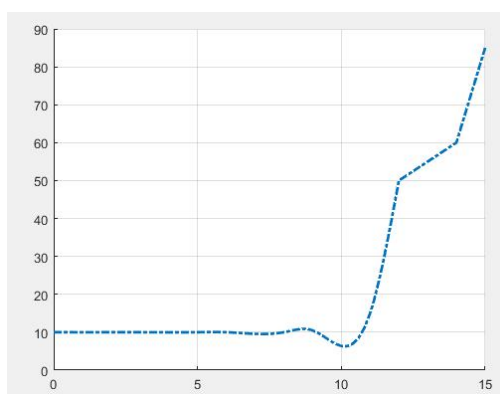
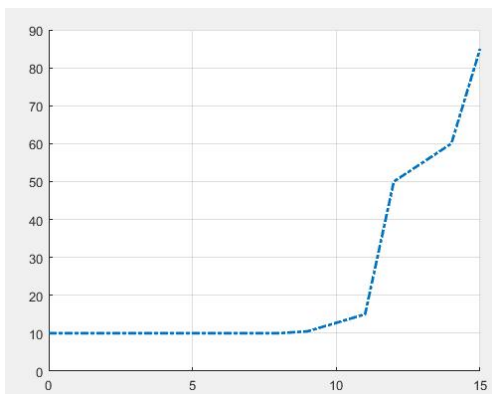


第一类边界条件:



改变权重值之后可能得到:

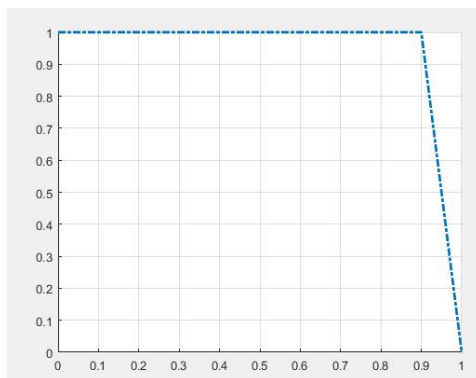
$a=[0\ 0\ 0\ 0\ 0\ 1\ 12\ 12\ 17\ 22\ 24];c=[0\ 0\ 0\ 1\ 2\ 2\ 3\ 10\ 12\ 22\ 100];$
 $a=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 3\ 5];c=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 5\ 5];$



我们可以看出不同边值结点和不同权值都可以改变原图像的单调性和凹凸性。

第三张图:

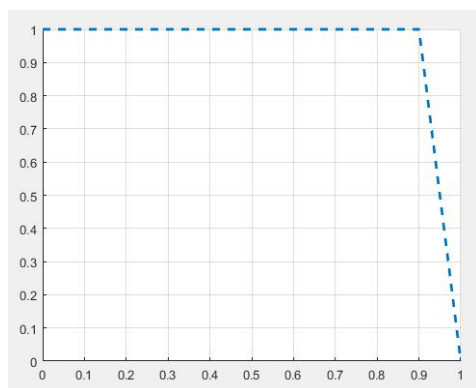
在第二类边界条件下, 我们取: $a=[0 \ 0 \ 0 \ 2 \ 5 \ 11 \ 11 \ 12 \ 15 \ 20 \ 25]$; $c=[0 \ 0 \ 0 \ 0 \ 11 \ 12 \ 18 \ 25 \ 30 \ 35 \ 40]$; 得到:



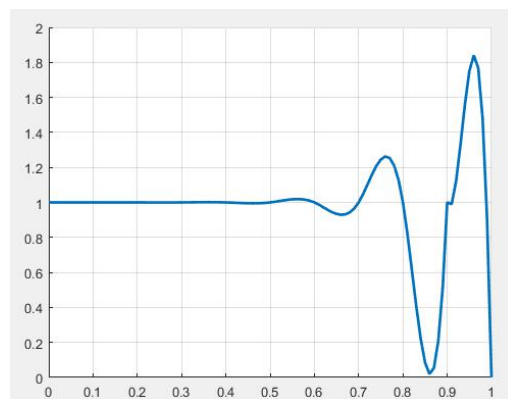
这个图像与原图近似相等。

改变边界条件可以得到:

Not-a-Knot:

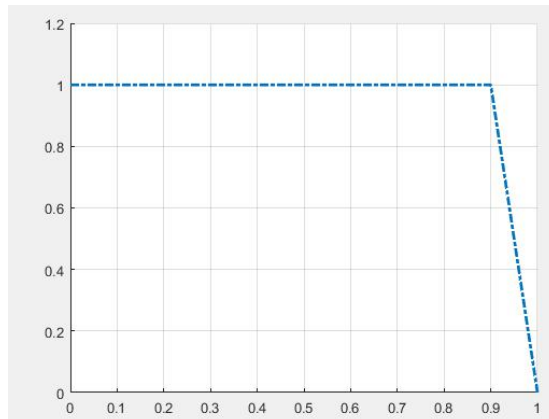
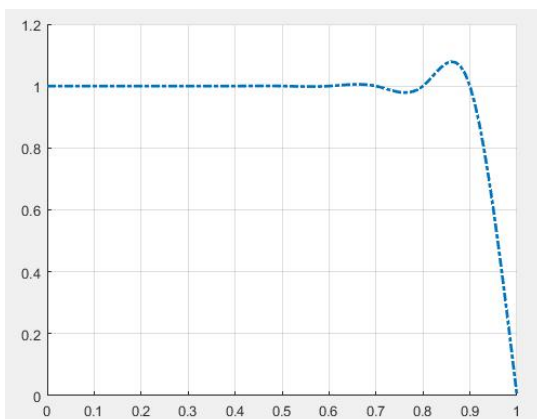


第一类边界条件:



再比较不同权值下的函数图像:

$a=[0 \ 0 \ 1 \ 3 \ 5 \ 11 \ 11 \ 12 \ 15 \ 20 \ 25]$; $c=[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$; 和 $a=[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 9 \ 10]$; $c=[0 \ 0 \ 0 \ 0 \ 0 \ 11 \ 12 \ 15 \ 27 \ 38 \ 40]$;



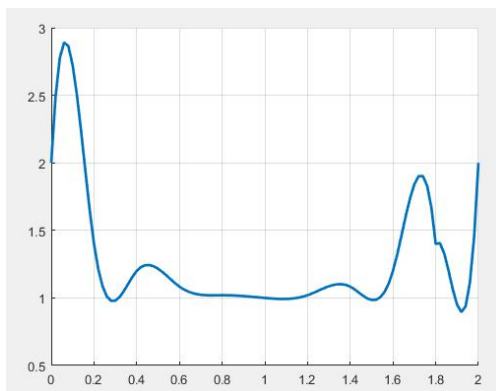
可以得到边界条件改变有较大影响而权值不同在第二类条件下对局部凸型和单调性有影响。

第四张图：

我们选择： $a=[4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 7]$ ； $c=[2\ 2\ 2\ 4\ 5\ 6\ 7\ 8\ 11\ 11\ 12]$ ；
根据题意公式：

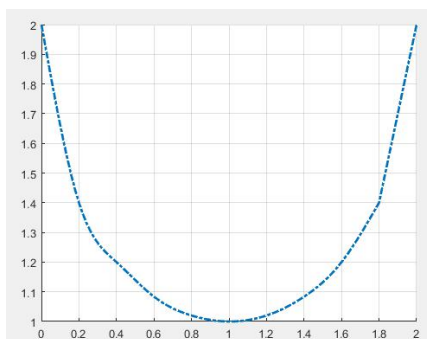
$$M_0 = 3w_0\delta_1/h_0, M_1 = 3w_N\delta_N f/h_N$$

我们得到第一类边界条件下两端点的值都为 30，于是我们得到如下的图像：

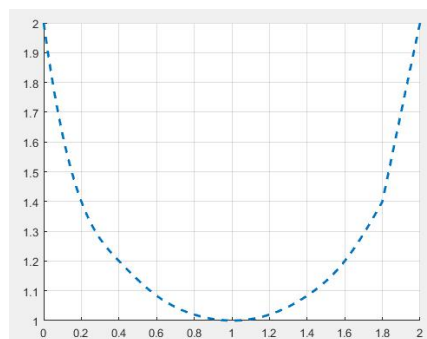


此图像在左半部分于原图有较大出入，已经实验多次但仍无法得到最理想的效果。
不同边界条件下的比较：

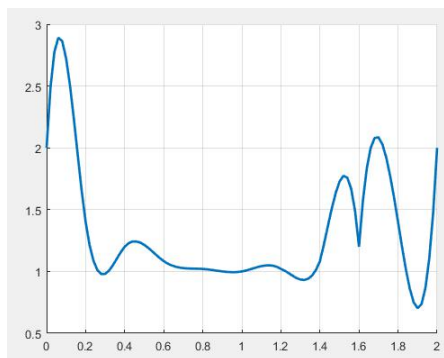
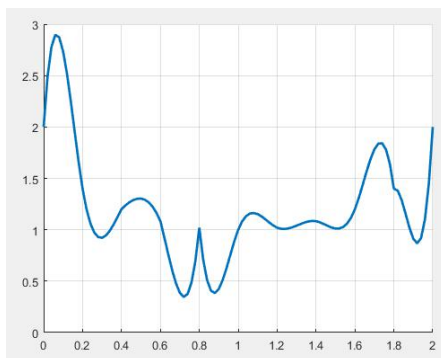
自然边界条件：



Not-a-Knot：



改变图像权重，在第一类边界条件下可以得到在 $a=[3\ 3\ 3\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 5]$ ；
 $c=[2\ 2\ 2\ 4\ 5\ 6\ 7\ 8\ 11\ 11\ 12]$ ；和 $a=[4\ 4\ 4\ 4\ 4\ 4\ 4\ 5\ 5\ 9\ 10]$ ；
 $c=[2\ 2\ 2\ 2\ 3\ 3\ 7\ 10\ 11\ 11\ 11]$ ；下的图像：



可以看出改变边值条件会虽然图像单调性和凸型得到保证但图像类似于抛物线已经失去了图像本身的意义。另一方面改变权重就会得到一系列奇怪的图像，原因是权值的取法也有一定的限制，文章最后我将结合这一点做出说明。

连续性分析：

通过以上的图像容易知道，所有的图像无论加权与否或者改变边值条件与否都是连续的函数，但是连续的程度有着明显的差别：即光滑性。不加权的情况下，函数图像的光滑性明显要强于加权样条插值的情况。但是另一方面，加权样条的单调性和凸型有着更好的保障，也能让读者更加直观的理解函数本身。所以整体来说连续性是非常良好的。

不加权重原图模拟代码：

第一张图：

```
function HW2018_6_3(x,y,type)
%x=[4.00 4.5 5.0 5.5 6.0 6.5];
x=[7.99 8.09 8.19 8.7 9.2 10 12 15 20];
y=[0 0.0000276429 0.0437498 0.169183 0.469428 0.94374 0.998636 0.999916
0.999994];
n=length(x);
m=length(y);
if n~=m
    return;
end
%计算 newton 差商
newtondiff=divdif(x,y);
%计算步长
h=x(2:end)-x(1:end-1);
lambda=h(1:end-1)./(h(1:end-1)+h(2:end));
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
```

```

A(end,end)=1;
%构造自然样条和 Not-A-Knot 的右端项
b=[0;6*newtondiff(3:end,3);0];
else
%根据一型样条的边界条件修正系数矩阵的第一行和最后一行
A(1,1:2)=[2,1];
A(end,end-1:end)=[1,2];
%构造一型样条的右端项
m0=-1;
m1=-2;
b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);...
6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b;
%绘图
linetype=['- ','-.','--'];
nn=10;
pointx=(0:nn) *(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))+...
    +(y(k)-M(k)*h(k).^2/6).*(x(k+1)-tmpx)./h(k)+(y(k+1)-M(k+1)*h(k).^2/6)*(t
    mpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);
M 求解的 matlab 代码和 Lingo 代码:
x=[7.99,8.09,8.19,8.7,9.2 10 12 15 20];
y=[0 0.0000276429 0.0437498 0.169183 0.469428 0.94374 0.998636 0.999916
0.999994];
n=length(x)-1;
H=diff(x);

```

```

lambda=H(1:n-1) ./ (H(1:n-1)+H(2:n))
Y1=diff(y) ./ diff(x);
Y2=diff(Y1) ./ (x(3:end)-x(1:end-2))
model:
m0=0;
0.5*m0+2*m1+0.5*m2=2.9847;
0.1639*m1+2*m2+0.8361*m3=-0.3136;
0.505*m2+2*m3+0.495*m4=0.351;
0.3846*m3+2*m4+0.6154*m5=-0.0058;
0.2857*m4+2*m5+0.7146*m6=-0.2019;
0.4*m5+2*m6+0.6*m7=-0.0054;
0.375*m6+2*m7+0.625*m8=-0.0001;
m8=0;
@free(m0);
@free(m1);
@free(m2);
@free(m3);
@free(m4);
@free(m5);
@free(m6);
@free(m7);
@free(m8);

```

第二张图:

```

function HW2018_6_3(type)
%x=[4.00 4.5 5.0 5.5 6.0 6.5];
x=[0 2 3 5 6 8 9 11 12 14 15];
y=[10 10 10 10 10 10 10.5 15 50 60 85];
n=length(x);
m=length(y);
if n~=m
    return;
end
%计算 newton 差商
newtondiff=divdif(x,y);
%计算步长
h=x(2:end)-x(1:end-1);
lambda=h(1:end-1) ./ (h(1:end-1)+h(2:end));
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';

```

```

A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
    A(end,end)=1;
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
else
    %根据一型样条的边界条件修正系数矩阵的第一行和最后一行
    A(1,1:2)=[2,1];
    A(end,end-1:end)=[1,2];
    %构造一型样条的右端项
    m0=-1;
    m1=-2;
    b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);...
        6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b
%绘图
linetype=['- ','-.','--'];
nn=10;
pointx=(0:nn)'+(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))+...
        (y(k)-M(k)*h(k).^2/6)*(x(k+1)-tmpx)/h(k)+(y(k+1)-M(k+1)*h(k).^2/6)*(t
        mpx-x(k))/h(k);

```

```

        xx=[xx;tmpx];
        yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);
M求解的matlab代码和Lingo代码:
x=[0 2 3 5 6 8 9 11 12 14 15];
y=[10 10 10 10 10 10 10.5 15 50 60 85];
w=[2 2.4 5.6 3.4 5.6 5 2 9 12 23 2.3];
n=length(x)-1;
H=diff(x);
G=diff(w);
lambda=(G(2:n).*H(1:n-1))./( (G(2:n).*H(1:n-1))+H(2:n).*G(1:n-1))
y2=diff(diff(y)./diff(x))./(x(3:end)-x(1:end-2)) %二阶差商
model:
m0=0;
0.67*m0+2*m1+0.33*m2=0;
0.33*m1+2*m2+0.67*m3=0;
0.67*m2+2*m3+0.33*m4=0;
0.33*m3+2*m4+0.67*m5=0;
0.67*m4+2*m5+0.33*m6=1.67;
0.33*m5+2*m6+0.67*m7=0.5833;
0.67*m6+2*m7+0.33*m8=10.9167;
0.33*m7+2*m8+0.67*m9=-10;
0.67*m8+2*m9+0.33*m10=6.67;
m10=0;
@free(m0);
@free(m1);
@free(m2);
@free(m3);
@free(m4);
@free(m5);
@free(m6);
@free(m7);
@free(m8);
@free(m9);
@free(m10);

```

第三张图:

```

function HW2018_6_3(type)
%x=[4.00 4.5 5.0 5.5 6.0 6.5];
x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];

```

```

y=[1 1 1 1 1 1 1 1 1 1 0];
n=length(x);
m=length(y);
if n~=m
    return;
end
%计算 newton 差商
newtondiff=dividif(x,y);
%计算步长
h=x(2:end)-x(1:end-1);
lambda=h(1:end-1)./(h(1:end-1)+h(2:end));
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
    A(end,end)=1;
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
else
    %根据一型样条的边界条件修正系数矩阵的第一行和最后一行
    A(1,1:2)=[2,1];
    A(end,end-1:end)=[1,2];
    %构造一型样条的右端项
    m0=0;
    m1=-100;
    b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);...
        6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b
%绘图
linetype=['- ' ; '-.' ; '--'];

```



```

nn=10;
pointx=(0:nn)'*(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))...

    +(y(k)-M(k)*h(k).^2/6).*(x(k+1)-tmpx)./h(k)+(y(k+1)-M(k+1)*h(k).^2/6)*(t
    mpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);
M 求解的 matlab 代码和 Lingo 代码:
x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
y=[1 1 1 1 1 1 1 1 1 1 0];
n=length(x)-1;
H=diff(x);
lambda=H(1:n-1)./(H(1:n-1)+H(2:n))
Y1=diff(y)./diff(x);
Y2=diff(Y1)./(x(3:end)-x(1:end-2))
model:
4*m0-m1=0;
0.5*m0+2*m1+0.5*m2=0;
0.5*m1+2*m2+0.5*m3=0;
0.5*m2+2*m3+0.5*m4=0;
0.5*m3+2*m4+0.5*m5=0;
0.5*m4+2*m5+0.5*m6=0;
0.5*m5+2*m6+0.5*m7=0;
0.5*m6+2*m7+0.5*m8=0;
0.5*m7+2*m8+0.5*m9=0;
0.5*m8+2*m9+0.5*m10=-50;
2*m10+m9=-5400;
@free(m0);
@free(m1);
@free(m2);

```

```

@free(m3);
@free(m4);
@free(m5);
@free(m6);
@free(m7);
@free(m8);
@free(m9);
@free(m10);

```

第四张图:

```

function HW2018_6_3(type)
%x=[4.00 4.5 5.0 5.5 6.0 6.5];
x=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2];
y=[2 1.4 1.2 1.0835 1.0202 1 1.0202 1.0835 1.2 1.4 2];
n=length(x);
m=length(y);
if n~=m
    return;
end
%计算 newton 差商
newtondiff=divdif(x,y);
%计算步长
h=x(2:end)-x(1:end-1);
lambda=h(1:end-1)./(h(1:end-1)+h(2:end));
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
    A(end,end)=1;
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
else

```

```

%根据一型样条的边界条件修正系数矩阵的第一行和最后一行
A(1,1:2)=[2,1];
A(end,end-1:end)=[1,2];
%构造一型样条的右端项
m0=-50;
m1=50;
b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);...
    6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b
%绘图
linetype=['- ' ; '-.' ; '--'];
nn=10;
pointx=(0:nn) *(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))+
    (y(k)-M(k)*h(k).^2/6)*(x(k+1)-tmpx)/h(k)+(y(k+1)-M(k+1)*h(k).^2/6)*(tmpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);
M求解的matlab代码和Lingo代码:
x=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2];
y=[2 1.4 1.2 1.0835 1.0202 1 1.0202 1.0835 1.2 1.4 2];
n=length(x)-1;
H=diff(x);
lambda=H(1:n-1)./(H(1:n-1)+H(2:n))
Y1=diff(y)./diff(x);
Y2=diff(Y1)./(x(3:end)-x(1:end-2))
model:
-0.4*m0-0.2*m1=-282;

```

```

0.5*m0+2*m1+0.5*m2=5;
0.5*m1+2*m2+0.5*m3=1.0437;
0.5*m2+2*m3+0.5*m4=0.665;
0.5*m3+2*m4+0.5*m5=0.5387;
0.5*m4+2*m5+0.5*m6=0.505;
0.5*m5+2*m6+0.5*m7=0.5387;
0.5*m6+2*m7+0.5*m8=0.665;
0.5*m7+2*m8+0.5*m9=1.0437;
0.5*m8+2*m9+0.5*m10=5;
0.4*m10+0.2*m9=282;
@free (m0);
@free (m1);
@free (m2);
@free (m3);
@free (m4);
@free (m5);
@free (m6);
@free (m7);
@free (m8);
@free (m9);
@free (m10);

```

不加权重结点比较图:

第一张图代码:

```

x=[7.99 8.09 8.19 8.7 9.2 10 12 15 20];
y=[0 0.0000276429 0.00437498 0.169183 0.469428 0.943740 0.998636 0.999916
0.999994];
pp1=csape(x,y,'variational');
pp2=csape(x,y,'not-a-knot');
pp3=csape(x,y,'periodic');
pp4=csape(x,y,'complete');
xx=7.99:0.01:20;
yy1=ppval(pp1,xx);
yy2=ppval(pp2,xx);
yy3=ppval(pp3,xx);
yy4=ppval(pp4,xx);
subplot(2,2,1)
plot(xx,yy1)
subplot(2,2,2)
plot(xx,yy2)
subplot(2,2,3)

```

```

plot(xx,yy3)
subplot(2,2,4)
plot(xx,yy4)

```

第二张图代码:

```

x=[0 2 3 5 6 8 9 11 12 14 15];
y=[10 10 10 10 10 10 10.5 15 50 60 85];
pp1=csape(x,y,'variational');
pp2=csape(x,y,'not-a-knot');
pp3=csape(x,y,'periodic');
pp4=csape(x,y,'complete');
xx=0:0.01:15;
yy1=ppval(pp1,xx);
yy2=ppval(pp2,xx);
yy3=ppval(pp3,xx);
yy4=ppval(pp4,xx);
subplot(2,2,1)
plot(xx,yy1)
subplot(2,2,2)
plot(xx,yy2)
subplot(2,2,3)
plot(xx,yy3)
subplot(2,2,4)
plot(xx,yy4)

```

第三张图代码:

```

x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
y=[1 1 1 1 1 1 1 1 1 1 0];
pp1=csape(x,y,'variational');
pp2=csape(x,y,'not-a-knot');
pp3=csape(x,y,'periodic');
pp4=csape(x,y,'complete',[0 -100]);
xx=0:0.01:1;
yy1=ppval(pp1,xx);
yy2=ppval(pp2,xx);
yy3=ppval(pp3,xx);
yy4=ppval(pp4,xx);
subplot(2,2,1)
plot(xx,yy1)
subplot(2,2,2)
plot(xx,yy2)

```

```

subplot(2,2,3)
plot(xx,yy3)
subplot(2,2,4)
plot(xx,yy4)

```

第四张图代码:

```

x=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2];
y=[2 1.4 1.2 1.0835 1.0202 1 1.0202 1.0835 1.2 1.4 2];
pp1=csape(x,y,'variational');
pp2=csape(x,y,'not-a-knot');
pp3=csape(x,y,'periodic');
pp4=csape(x,y,'complete');
xx=0:0.01:2;
yy1=ppval(pp1,xx);
yy2=ppval(pp2,xx);
yy3=ppval(pp3,xx);
yy4=ppval(pp4,xx);
subplot(2,2,1)
plot(xx,yy1)
subplot(2,2,2)
plot(xx,yy2)
subplot(2,2,3)
plot(xx,yy3)
subplot(2,2,4)
plot(xx,yy4)

```

加权重代码原图（包含对不同结点比较，不同权值不再另附代码，只需在源代码中给出新的 a 和 c 的值即可，首先给出 dividif 的代码）：

dividif:

```

function [d]=dividif(x,y,w)
[n m]=size(y);
if n==1
    n=m;
    y=y';
end
n=n-1;
d=sparse(n+1,n+1);
d(:,1)=y';
for j=2:n+1
    for i=j:n+1

```

```

d(i,j)=(d(i-1,j-1)-d(i,j-1))*w(i-1)*w(i)/(x(i-j+1)*w(i)-x(i)*w(i-j+1));
    end
end

```

第一张图:

```

function HW2018_6_3(type)
x=[7.99 8.09 8.19 8.7 9.2 10 12 15 20];
y=[0 0.0000276429 0.0437498 0.169183 0.469428 0.94374 0.998636 0.999916
0.999994];
a=[5 10 15 18 19 20 25 27 28];
c=[0 0 0 1 4 6 12 14 16];
n=length(x);
m=length(y);
if n~=m
    return;
end

```

%计算步长

```

h=x(2:end)-x(1:end-1);
g=y(2:end)-y(1:end-1);
e=c(2:end)-c(1:end-1);
f=a(2:end)-a(1:end-1);
w=(1+e(1:end).*((g(1:end)./h(1:end)).^(2))).^(-f(1:end));
lambda=[w(2:end).*h(1:end-1)]./[w(1:end-1).*h(2:end)+w(2:end).*h(1:end-1)];
w=[1,w(1:end),1];

```

%计算 newton 差商

```

newtondiff=dividif(x,y,w);
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';

```

```

if type==3

```

%根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行

```

A(1,(1:3))=90*[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];

```

```

A(end,(end-2:end))=3*[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];

```

%构造自然样条和 Not-A-Knot 的右端项

```

b=[0;6*newtondiff(3:end,3);0];

```

```

elseif type==2

```

%根据自然边界条件修正系数矩阵的第一行和最后一行

```

A(1,1)=1;
A(end,end)=1;
%构造自然样条和 Not-A-Knot 的右端项
b=[0;6*newtondiff(3:end,3);0];
else
    %根据一型样条的边界条件修正系数矩阵的第一行和最后一行
    A(1,1:2)=[2,1];
    A(end,end-1:end)=[1,2];
    %构造一型样条的右端项
    m0=0;
    m1=-100;
    b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);
        6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b;
%绘图
linetype=['- ' ;'-.' ;'--'];
nn=10;
pointx=(0:nn) *(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))...
        +(y(k)-M(k)*h(k).^2/6).*(x(k+1)-tmpx)./h(k)+(y(k+1)-M(k+1)*h(k)^2/6)*(t
        mpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);

```

第二张图:

```

x=[0 2 3 5 6 8 9 11 12 14 15];
y=[10 10 10 10 10 10 10.5 15 50 60 85];
a=[0 0 0 0 9 11 12 12 17 22 24];

```



```

c=[0 0 0 0 0 0 0 4 12 21 23];
n=length(x);
m=length(y);
if n~=m
    return;
end

%计算步长
h=x(2:end)-x(1:end-1);
g=y(2:end)-y(1:end-1);
e=c(2:end)-c(1:end-1);
f=a(2:end)-a(1:end-1);
w=(1+e(1:end).*((g(1:end)./h(1:end)).^(2))).^(-f(1:end));
lambda=[w(2:end).*h(1:end-1)]./[w(1:end-1).*h(2:end)+w(2:end).*h(1:end-1)];
w=[1,w(1:end),1];
%计算 newton 差商
newtondiff=dividif(x,y,w);
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=90*[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=3*[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
    A(end,end)=1;
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
else
    %根据一型样条的边界条件修正系数矩阵的第一行和最后一行
    A(1,1:2)=[2,1];
    A(end,end-1:end)=[1,2];
    %构造一型样条的右端项
    m0=0;
    m1=-100;

```

```

        b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);
          6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b;
%绘图
linetype=['- ','-.','--'];
nn=10;
pointx=(0:nn) *(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))...
    +(y(k)-M(k)*h(k).^2/6).*(x(k+1)-tmpx)./h(k)+(y(k+1)-M(k+1)*h(k)^2/6)*(t
    mpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);

```

第三张图:

```

function HW2018_6_3(type)
x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
y=[1 1 1 1 1 1 1 1 1 1 0];
a=[0 0 0 2 5 11 11 12 15 20 25];
c=[0 0 0 0 11 12 18 25 30 35 40];
n=length(x);
m=length(y);
if n~=m
    return;
end

%计算步长
h=x(2:end)-x(1:end-1);
g=y(2:end)-y(1:end-1);

```

```

e=c(2:end)-c(1:end-1);
f=a(2:end)-a(1:end-1);
w=(1+e(1:end).*(g(1:end)./h(1:end)).^(2))).^(-f(1:end));
lambda=[w(2:end).*h(1:end-1)]./[w(1:end-1).*h(2:end)+w(2:end).*h(1:end-1)];
w=[1,w(1:end),1];
%计算 newton 差商
newtondiff=dividif(x,y,w);
A=sparse(n,n);
A(1:n+1:end)=2;
A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=90*[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=3*[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
    A(end,end)=1;
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
else
    %根据一型样条的边界条件修正系数矩阵的第一行和最后一行
    A(1,1:2)=[2,1];
    A(end,end-1:end)=[1,2];
    %构造一型样条的右端项
    m0=0;
    m1=-100;
    b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);
        6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b;
%绘图
linetype=['-';'-.-';'--'];
nn=10;
pointx=(0:nn)'+(x(2:end)-x(1:end-1))/nn;
figure(1);

```

```

hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))...

    +(y(k)-M(k)*h(k).^2/6).*(x(k+1)-tmpx)./h(k)+(y(k+1)-M(k+1)*h(k).^2/6)*(t
    mpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);

```

第四张图:

```

function HW2018_6_3(type)
x=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2];
y=[2 1.4 1.2 1.0835 1.0202 1 1.0202 1.0835 1.2 1.4 2];
a=[4 4 4 4 4 4 4 4 4 4 7];
c=[2 2 2 4 5 6 7 8 11 11 12];
n=length(x);
m=length(y);
if n~=m
    return;
end

```

%计算步长

```

h=x(2:end)-x(1:end-1);
g=y(2:end)-y(1:end-1);
e=c(2:end)-c(1:end-1);
f=a(2:end)-a(1:end-1);
w=(1+e(1:end).*((g(1:end)./h(1:end)).^(2))).^(-f(1:end));
lambda=[w(2:end).*h(1:end-1)]./[w(1:end-1).*h(2:end)+w(2:end).*h(1:end-
1)];
w=[1,w(1:end),1];
%计算 newton 差商
newtondiff=dividif(x,y,w);
A=sparse(n,n);
A(1:n+1:end)=2;

```

```

A(2:n+1:(n-2)*n)=lambda';
A(2*n+2:n+1:end)=(1-lambda)';
if type==3
    %根据 NOT-A-KNOT 条件修正系数矩阵的第一行和最后一行
    A(1,(1:3))=90*[1/h(1),-(1/h(2)+1/h(1)),1/h(2)];
    A(end,(end-2:end))=3*[1/h(n-2),-(1/h(n-2)+1/h(n-1)),1/h(n-1)];
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
elseif type==2
    %根据自然边界条件修正系数矩阵的第一行和最后一行
    A(1,1)=1;
    A(end,end)=1;
    %构造自然样条和 Not-A-Knot 的右端项
    b=[0;6*newtondiff(3:end,3);0];
else
    %根据一型样条的边界条件修正系数矩阵的第一行和最后一行
    A(1,1:2)=[2,1];
    A(end,end-1:end)=[1,2];
    %构造一样条的右端项
    m0=w(1)*30;
    m1=w(10)*30;
    b=[6*(newtondiff(2,2)-m0)/h(1);6*newtondiff(3:end,3);
        6*(m1-newtondiff(end,2))/h(end)];
end
%求解弯矩
M=A\b;
%绘图
linetype=['- ','-.','--'];
nn=10;
pointx=(0:nn)'+(x(2:end)-x(1:end-1))/nn;
figure(1);
hold on;
grid on;
xx=[];
yy=[];
for k=1:n-1
    tmpx=pointx(:,k)+x(k);

    tmpy=M(k)*(x(k+1)-tmpx).^3./(6*h(k))+M(k+1)*(tmpx-x(k)).^3./(6*h(k))...

```

```

+ (y(k)-M(k)*h(k).^2/6).*(x(k+1)-tmpx)./h(k)+(y(k+1)-M(k+1)*h(k)^2/6)*(t
mpx-x(k))/h(k);
    xx=[xx;tmpx];
    yy=[yy;tmpy];
end
plot(xx,yy,linetype(type,:), 'LineWidth',2);

```

控制多边形代码:

```

clear
k=3;
x=[0,0.5,0.5,1.0,1.0,2.0,3.0,3.0,4.0,4.0,5.0,4.8,3.0,2.4,1.0,1.0,0,0;
    0,0,0.5,0.5,0,0,0,0.5,0.5,0,0,1.0,1.2,2.0,2.0,1.0,1.0,0];
x=x';
[n,m]=size(x);
%-----弦长参数化-----
u(k+n)=0;
plot(x(:,1),x(:,2));
for i=1:n-1
u(k+i+1)=u(k+i)+sqrt((x(i+1,1)-x(i,1))^2+(x(i+1,2)-x(i,2))^2);
end;
L=u(n+k);
for i=1:n
u(k+i)=u(k+i)/L;
end;
for i=1:3
u(k+i+n)=1;
end
%控制多边形
hold on
%-----反求 n+2 个控制点-----
%首位重节点 v1=v2
%首位与控制多边形相切
A=zeros(n+2);
A(1,1)=1;A(1,2)=-1;
A(2,2)=1;
A(n+2,n+1)=-1;A(n+2,n+2)=1;
A(n+1,n+1)=1;
for i=3:n

```

```

for j=0:2
A(i,i+j-1)=Base(i+j-1,k,u,u(i+2));
end
end
e=0;
for i=1:m
e(n+2,i)=0;
end
for i=1:n
e(i+1,:)=x(i,:);
end
d=inv(A)*e;
hold on
x=0;y=0;down=0;
for j=1:(n-1)
uu=(u(j+3)):0.0005:u(j+4);
for kk=1:length(uu)
down=down+1;

x(down)=d(j,1)*Base(j,3,u,uu(kk))+d(j+1,1)*Base(j+1,3,u,uu(kk))+d(j+2,1)
*Base(j+2,3,u,uu(kk))+d(j+3,1)*Base(j+3,3,u,uu(kk));

y(down)=d(j,2)*Base(j,3,u,uu(kk))+d(j+1,2)*Base(j+1,3,u,uu(kk))+d(j+2,2)
*Base(j+2,3,u,uu(kk))+d(j+3,2)*Base(j+3,3,u,uu(kk));
end
end
axis('equal');
plot(x,y,'red');
xlabel('x');ylabel('y');
grid on

```