数值分析实验报告 - Code 3

Chase Young

2024 年 3 月 14 日

1 实验目的

对函数 $f(x)=\mathrm{e}^{x}, x\in[0,1]$ 构造等距节点的样条插值函数，对以下两种类型的样条函数：

(1) 一次分片线性样条

(2) 满足 $S'(0)=1, S'(1)=\mathrm{e}$ 的三次样条

并计算如下误差：

$$\max_i\left\{\left|f\left(x_{i-\frac{1}{2}}\right)-S\left(x_{i-\frac{1}{2}}\right)\right|, i=1,2,\ldots,N\right\}$$

这里 $x_{i-\frac{1}{2}}$ 为每个小区间的中点。对 $N=5,10,20,40$ 比较以上两组节点的结果。并利用公式

$$\mathrm{Ord}=\frac{\ln\left(\mathrm{Error}_{\mathrm{old}}/\mathrm{Error}_{\mathrm{now}}\right)}{\ln\left(\mathrm{N}_{\mathrm{now}}/\mathrm{N}_{\mathrm{old}}\right)}$$

计算算法的收敛阶。

2 实验方法

分别根据一次、三次样条插值的公式，构造样条插值函数。其中一次样条函数为：

$$S_i(x)=\frac{y_{i+1}-y_i}{x_{i+1}-x_i}(x-x_i)+y_i, x\in[x_i,x_{i+1}), i=0,1,2,\ldots,N$$

具体实现见函数:

- **function** s = splineOrder1(x, xx, yy)

对于三次样条函数，记 $m_i=S''(x_i), i=0,1,\ldots,N$。令 $h_i=x_{i+1}-x_i$，根据插值条件 $S_i(x_{i+1})=S_{i+1}(x_{i+1})$，一阶、二阶导数的连续性 $S'_{i+1}(x_{i+1})=S'_i(x_{i+1}), S''_{i+1}(x_{i+1})=S''_i(x_{i+1})$，以及固定边界条件 $S'(0)=1, S'(1)=\mathrm{e}$，可得关于 $m_0,m_1,\ldots,m_N$ 的线性方程组：

$$\begin{pmatrix} 2h_0 & h_0 & 0 & 0 & 0 & \ldots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & \ldots & 1 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & 2h_{n-1} \\ 0 & 0 & 0 & 0 & 0 & h_{n-1} & 2h_{n-1} \end{pmatrix}\cdot\begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{pmatrix}=\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{pmatrix} \quad (1)$$

其中 $h_i = x_{i+1} - x_i, v_0 = \frac{y_1 - y_0}{h_0} - 1, v_n = \mathrm{e} - \frac{y_n - y_{n-1}}{h_{n-1}}, v_i = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}, i = 1, 2, \ldots, n-1.$

根据上式，求解出 $m_0, m_1, \ldots, m_n$ 后，先判断给定的 x 在哪个区间 $[x_i, x_{i+1})$ 中，然后根据下式计算

$$S_i\left(x\right) = \frac{m_i}{6h_i}\left(x_{i+1} - x\right)^3 + \frac{m_{i+1}}{6h_i}\left(x - x_i\right)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{m_{i+1}h_i}{6}\right)\left(x - x_i\right) + \left(\frac{y_i}{h_i} - \frac{m_i h_i}{6}\right)\left(x_{i+1} - x\right)$$

即可得到插值函数在给定点 x 处的函数值。

具体实现见函数

- **function** mm = get2diff(xx, yy, m0, mn)
- **function** s = splineOrder3(x, xx, yy, mm)

3 实验结果

运行脚本 code_3.m，计算不同结点数量时，线性样条和三次样条插值的误差和收敛阶，如表1所示。

n	Method (1) error	order	Method (2) error	order
5	0.012308267	-	0.000010907	-
10	0.003232810	1.928766838	0.000000696	3.970936521
20	0.000828533	1.964158039	0.000000044	3.986880794
40	0.000209730	1.982022667	0.000000003	3.993794127

表 1: 不同结点数量下，线性样条插值和三次样条插值的误差及收敛阶

从上述结果中可以看出：

(1) 从误差角度来看，两种插值方法的误差都随着结点数量的增加而降低；对于相同的结点数量，三次样条插值的误差比线性插值的误差更小；

(2) 从收敛阶数来看，线性样条插值的收敛阶趋近于 2，三次样条插值的收敛阶趋近于 4；

4 后续讨论

从上述实验结果可以看出，无论是从收敛阶数来看，还是从插值误差来看，对于 $f\left(x\right) = \mathrm{e}^x, x \in [0, 1]$，三次样条插值的表现均比线性样条插值更加出色。

A 代码

本部分包含了主要用到的代码。

splineOrder1.m

```
function s = splineOrder1(x, xx, yy)
    % spline of order 1
    n = length(xx) - 1;
```

```
    idx = floor(x * n) + 1;
    s = (yy(idx+1) - yy(idx)) * (x - xx(idx)) / (xx(idx+1) - xx(idx)) + yy(idx);
end
```

get2diff.m

```
function mm = get2diff(xx, yy, m0, mn)
    % compute S'' for spline of order 3
    n = length(xx) - 1;
    h = 1:n;
    for i = 1:n
        h(i) = xx(i+1) - xx(i);
    end
    A = zeros(n+1, n+1);
    A(1,1) = 2*h(1);
    A(1,2) = h(1);
    A(2,1) = h(1);
    for i = 2:n
        A(i,i) = 2*(h(i-1) + h(i));
        A(i,i+1) = h(i);
        A(i+1,i) = h(i);
    end
    A(n+1,n+1) = 2*h(n);
    v = (1:n+1)';
    v(1) = (yy(2) - yy(1))/h(1) - m0;
    for i = 2:n
        v(i) = (yy(i+1) - yy(i))/h(i) - (yy(i) - yy(i-1))/h(i-1);
    end
    v(n+1) = mn - (yy(n+1) - yy(n))/h(n);
    v = 6.*v;
    mm = A\v;
end
```

splineOrder3.m

```
function s = splineOrder3(x, xx, yy, mm)
    % spline of order 3
    n = length(xx) - 1;
    i = floor(x*n) + 1;
    h = xx(i+1) - xx(i);
    s = mm(i) * (xx(i+1) - x)^3 / (6*h) + mm(i+1) * (x - xx(i))^3 / (6*h) ...
    + (yy(i+1)/h - mm(i+1)*h/6) * (x - xx(i)) + (yy(i)/h - mm(i)*h/6) * (xx(i+1) - x);
end
```

code_3.m

```
% code_3
NList = [5 10 20 40];
error1 = 1:length(NList);
error2 = 1:length(NList);
order1 = 1:length(NList);
```

```
order2 = 1:length(NList);
for i = 1:length(NList)
    N = NList(i);
    xx = (0:N) / N;
    yy = exp(xx);

    xTest = xx(2:end) - 0.5/N;
    yTest = exp(xTest);

    ss = 1:length(xTest);
    for j = 1:length(xTest)
        ss(j) = splineOrder1(xTest(j), xx, yy);
    end
    error1(i) = max(abs(yTest - ss));

    mm = get2diff(xx, yy, 1, exp(1));
    for j = 1:length(xTest)
        ss(j) = splineOrder3(xTest(j), xx, yy, mm);
    end
    error2(i) = max(abs(yTest - ss));
end

for i = 2:length(NList)
    order1(i) = log(error1(i-1)/error1(i)) / log(NList(i)/NList(i-1));
    order2(i) = log(error2(i-1)/error2(i)) / log(NList(i)/NList(i-1));
end

fprintf("N = %d\n", NList(1));
fprintf("Max error of method (1) is %.9f\n", error1(1));
fprintf("Max error of method (2) is %.9f\n", error2(1));
fprintf("Order of method (1) is ---\n");
fprintf("Order of method (2) is ---\n");
for i = 2:length(NList)
    fprintf("N = %d\n", NList(i));
    fprintf("Max error of method (1) is %.9f\n", error1(i));
    fprintf("Max error of method (2) is %.9f\n", error2(i));
    fprintf("Order of method (1) is %.9f\n", order1(i));
    fprintf("Order of method (2) is %.9f\n", order2(i));
end
```