数值分析 code3 实验报告

张景浩 PB20010399

2023.3.22

1 问题介绍

给出函数

$$f(x) = e^x, x \in [0,1]$$

构造等距结点 $0 = t_0 < \cdots < t_N = 1$ 的两种样条插值函数 S：

1. 一次分片线性样条

2. 满足 $S'(0) = 1, S'(1) = e$ 的三次样条

并分别计算误差

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

其中，$x_{i-\frac{1}{2}}$ 是每个区间 $[t_{i-1}, t_i]$ 的中点。对 N=5,10,20,40 比较以上两中样条插值函数的误差结果，并利用公式 $ord = \frac{\ln(Error_{old}/Error_{now})}{\ln(N_{now}/N_{old})}$ 计算算法的收敛阶。

2 解决方法

（下面只给出算法的思路，具体代码实现见附录）

2.1 一次分片线性样条

算法思路参考《数值分析》$P277$。

由于在每一个区间 $[t_{i-1}, t_i]$ 上都有 $S(x) = a_i x + bi$，所以依据插值点的函数信息，即可计算出相对应的系数。

2.2 满足 $S'(0) = 1, S'(1) = e$ 的三次样条

算法思路参考《数值分析》$P278 \sim P280$ 以及课程 PPT。

首先由于多项式 $S_{i-1}(t_i) = y_i = S_i(t_i)$ $(1 \le i \le n-1)$ 有相同的插值，所以 S 是连续的。我们定义一组数 $z_i = S''(t_i)$，因为 S'' 在每个内结点上连续，所以显然，对于 $0 \le i \le n$，存在 z_i 且满足：

$$\lim_{x \to t_i^-} S''(x) = z_i = \lim_{x \to t_i^+} S''(x) \quad (1 \le i \le n-1) \qquad (1)$$

因为 S_i 是 $[t_i, t_{i+1}]$ 上的三次多项式，所以 S_i'' 也是 $[t_i, t_{i+1}]$ 上的一次函数：

$$S_i''(x) = \frac{z_i}{h_i}(t_{i+1} - x) + \frac{z_{i+1}}{h_i}(x - t_i) \qquad (2)$$

其中，$h_i = t_{i+1} - t_i$。将 (2) 式积分两次并结合插值条件 $S_i(t_i) = y_i$ 和 $S_i(t_{i+1}) = y_{i+1}$，我们得到：

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-t_i)^3 + (\frac{y_{i-1}}{h_i} - \frac{z_{i+1}h_i}{6})(x-t_i) + (\frac{y_i}{h_i} - \frac{z_i h_i}{6})(t_{i+1}-x) \qquad (3)$$

接下来通过 S' 的连续性来确定 $z_1, z_2, \cdots, z_{n-1}$，得到等式：

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = \frac{6}{h_i}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1}) \qquad (4)$$

令 $\mu_i = \frac{h_{i-1}}{h_i + h_{i-1}}, \lambda_i = \frac{h_i}{h_i + h_{i-1}} = 1 - \mu_i, g_i = \frac{6}{h_{i-1}+h_i}(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}})$，所以 (4) 式简化为：

$$\mu_i z_{i-1} + 2z_i + \lambda_i z_{i+1} = g_i$$

带入边界条件 $S'(0) = 1, S'(1) = e$ 后有：

$$g_0 = 2z_0 + z_1 = \frac{6}{h_0}(\frac{y_1 - y_0}{h_0} - 1)$$

$$g_n = z_{n-1} + 2z_n = \frac{6}{h_{n-1}}(e - \frac{y_n - y_{n-1}}{h_{n-1}})$$

解方程组：

$$\begin{bmatrix} 2 & 1 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & 1 & 2 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \\ g_n \end{bmatrix}$$

即可得到系数 $z_0, z_1, \cdots, z_n$，带入 (3) 式得三次样条函数。

3 编译环境及使用方法

本程用 matlab 编译，使用时直接调用 outcome.m 文件即可。

4 实验结果

N	Method(1)error	order	Method(2)error	order
5	1.230827×10^{-2}	-	1.090742×10^{-5}	-
10	3.232810×10^{-3}	1.928767	6.955865×10^{-7}	3.970937
20	8.285329×10^{-4}	1.964158	4.387129×10^{-8}	3.986881
40	2.097304×10^{-4}	1.982023	2.753775×10^{-9}	3.993794

表 1: 两种样条函数的误差和收敛阶

5 总结

从表 1 中可以看出，不论是一次样条函数还是三次样条函数对于目标函数都有很好的拟合效果，并且随着插值结点的增加，与原函数之间的误差也在减小。相比之下，三次样条函数的拟合性

要远好于一次样条函数，基本上原函数的误差比一次样条函数的小了 4 个数量级左右。从收敛阶上来看，二者的收敛阶都随着插值结点个数的增加而增大，而三次样条函数的收敛阶大约是一次样条函数的两倍左右。结合本次实验结果和前两次实验来看，样条函数克服了多项式插值可能出现的震荡现象，具有较好的数值稳定性和收敛性。

A Computer Code

Here we include the computer code.

```
% @一次样条函数
% @输入t为等距结点，x为自变量，输出S是一次样条函数在x处的值
function S=linearSplineInter(t,x)
N=length(t);
%% 构造一次样条函数系数
a=zeros(1,N-1);b=zeros(1,N-1);
for i=1:N-1
    a(i)=(exp(t(i+1))-exp(t(i)))/(t(i+1)-t(i));
    b(i)=exp(t(i))-t(i)*(exp(t(i+1))-exp(t(i)))/(t(i+1)-t(i));
end
%% 计算一次样条函数
for i=1:N-1
    if x>=t(i)&&x<=t(i+1)
        S=a(i)*x+b(i);
        break;
    end
end
end

% @三次样条函数
% @输入t为等距结点，x为自变量，输出S是三次样条函数在x处的值
function S=cubicSplineInter(t,x)
N=length(t);
%% 计算三次样条函数在结点处的二阶导数
h=zeros(1,N-1);b=zeros(1,N-1);
for i=1:N-1
    h(i)=t(i+1)-t(i);
    b(i)=6*(exp(t(i+1))-exp(t(i)))/h(i);
end
u=ones(1,N-1);v=ones(1,N-1);g=zeros(1,N);
for i=1:N-2
    u(i)=h(i)/(h(i+1)+h(i));v(i+1)=1-u(i);
    g(i+1)=(b(i+1)-b(i))/(h(i+1)+h(i));
end
g(1)=(b(1)-6*1)/h(1);
g(N)=(exp(1)*6-b(N-1))/h(N-1);
m=ones(1,N)*2;
M=diag(m)+diag(u,-1)+diag(v,1);
```

```
z=M\g.';
%% 构造三次样条函数
A=zeros(1,N-1);B=zeros(1,N-1);C=zeros(1,N-1);
for i=1:N-1
    A(i)=(z(i+1)-z(i))/(6*h(i));
    B(i)=z(i)/2;
    C(i)=-h(i)*z(i+1)/6-h(i)*z(i)/3+(exp(t(i+1))-exp(t(i)))/h(i);
end

for i=1:N-1
    if x>=t(i)&&x<=t(i+1)
        S=exp(t(i))+(x-t(i))*(C(i)+(x-t(i))*(B(i)+(x-t(i))*A(i)));
        break;
    end
end
end

% @误差计算函数
% @输入结点t，输出两种样条函数的最大误差
function [err_1,err_2]=errorFunc(t)
N=length(t);
x=zeros(1,N-1);y_1=zeros(1,N-1);y_2=zeros(1,N-1);
for i=1:N-1
    x(i)=(t(i)+t(i+1))/2;
    y_1(i)=abs(exp(x(i))-linearSplineInter(t,x(i)));
    y_2(i)=abs(exp(x(i))-cubicSplineInter(t,x(i)));
end
err_1=max(y_1);err_2=max(y_2);
end

% @程序脚本
N=5;
err=zeros(1,8);
i=1;
%% 计算两种样条函数的误差
while N<=40
    t=linspace(0,1,N+1);
    [err(i),err(i+4)]=errorFunc(t);
    N=N*2;i=i+1;
end
%% 计算收敛阶
ord=zeros(1,8);
for i=2:4
    ord(i)=log(err(i-1)/err(i))/log(2);ord(i+4)=log(err(i+3)/err(i+4))/log(2);
end
N=5;
for i=1:4
```

```
    fprintf('%d & %e & %f & %e & %f\t',N,err(i),ord(i),err(i+4),ord(i+4));
    fprintf('\\');
    fprintf('\\');
    fprintf('\n');
    N=N*2;
end
```