

数值分析上机报告

使用三次样条插值多项式及用切比雪夫多项式零点构造牛顿插值多项式并观察高次多项式的龙格现象

陈开来

2022 年 4 月 9 日

目录

1 问题描述 1

2 牛顿插值 1

2.1 数学原理 1

2.2 算法实现 2

3 三次样条插值函数 2

3.1 数学原理 2

3.2 算法实现 3

4 chebyshev 插值 5

4.1 数学原理 5

4.2 算法实现 5

5 实验结果 6

5.1 图像绘制与分析 7

A 完整代码 8

1 问题描述

对于给定的函数

$$f(x) = \frac{1}{1 + 25x^2} (-1 \leq x \leq 1)$$

我们取其等距节点,在第二部分中我们构造 $f(x)$ 的牛顿插值多项式 $N_5(x), N_{10}(x)$ 在第三部分中我们将构造其三次样条插值函数 $S_{10}x$ 并将三种插值函数与 $f(x)$ 画在同一个坐标系内进行比较。第四部分我们将以 chebyshev 多项式的零点代替之前取的等距离点进行插值得到 $N_5(x), N_{10}(x)$ 并与准确解画图比较

2 牛顿插值

2.1 数学原理

在已知需要插值的 $n+1$ 个结点 x_i 所构成的列向量 x 和其对应值 y_i 所构成的列向量 y 的情况下我们关键是求出差商表

$$f = \begin{bmatrix} f[x_0] & & & & \\ f[x_1] & f[x_0, x_1] & & & \\ f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\ \vdots & \vdots & \vdots & \ddots & \\ f[x_n] & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \dots & f[x_0, \dots, x_n] \end{bmatrix}$$

递推的公式是 $f(k, i) = (f(k, i-1) - f(k-1, i-1)) / (x_i(k) - x_i(k+1-i)); (2 \leq i \leq n, i \leq k \leq n)$

最后代入

$$N_n(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_n)$$

即得

2.2 算法实现

```
function N = Newton(a,b)
```

%N是求得的插值多项式系数向量，a是插值点构成的列向量，b是插值点对应函数值的列向量

```
n=length(a);
```

```
df=zeros(n,n); %df储存差商表，并初始化
```

```
df(:,1)=b; %将b赋值给差商表第一列
```

```
syms x %定义一个参数x
```

```
N=df(1,1);
```

```
acc=1; %acc作为一个临时变量存储累加的式子
```

```
for i=2:n
```

```
    for k=i:n
```

```
        df(k,i)=(df(k,i-1)-df(k-1,i-1))/(a(k)-a(k+1-i));
```

```
    end
```

```
    acc=acc*(x-a(i-1));
```

```
    N=N+df(i,i)*acc;
```

```
end
```

```
N=expand(N); %将N展开
```

```
N=vpa(N,6); %将N中的分数保留为六位小数
```

```
end
```

3 三次样条插值函数

3.1 数学原理

由于三次样条插值多项式涉及到的参数较多，我们先将要用到的参数以表格形式给出

参数名	参数定义
$s(x)$	三次样条插值函数
$s_i(x)$	$[x_{i-1}, x_i]$ 上的三次样条插值函数
h_i	第 i 段插值区间长度 $x_i - x_{i-1}$
μ_i	$\mu_i = \frac{h_i}{h_i + h_{i+1}} (i = 2, 3, \dots, n-1)$
λ_i	$\lambda_i = 1 - \mu_i (i = 2, 3, \dots, n-1)$
d_i	$d_i = 6f[x_{i-1}, x_i, x_{i+1}] (i = 2, 3, \dots, n-1)$
M_i	$M_i = S''(x_i)$

我们用第一种边界条件, 即 $M_1 = f''(a), M_n = f''(b)$

插值表达式如下:

$$S_i(x) = \frac{(x_i - x)^3}{6h_i} M_{i-1} + \frac{(x_i - x_{i-1})^3}{6h_i} M_i + (y_{i-1} - \frac{h_i^2}{6} M_{i-1}) \frac{x_i - x}{h_i} + (y_i - \frac{h_i^2}{6} M_i) \frac{x - x_{i-1}}{h_i} (i = 2, 3, \dots, n)$$

其中 M_i 满足三弯矩方程 $\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i (i = 2, 3, \dots, n-1)$

3.2 算法实现

```
function S=sanci(a, b, M1, Mn)%a是插值节点构成的向量, b是对应的值
n=length(a);
syms x
h=a(2:n)-a(1:n-1); % 求解向量h
mu=h(1:n-2)./(h(1:n-2)+h(2:n-1)); %
lambda=1-mu; % 求解向量lambda
h1=h(1:n-2); % 定义中间变量h1
h2=h(2:n-1); % 定义中间变量h2
d=6*((b(3:n)-b(2:n-1))./h2-(b(2:n-1)-b(1:n-2))./h1)./(h1+h2);
A=zeros(n-2, n-2); % 初始化系数矩阵A
for i=1:n-2
```

```

        A(i, i) = 2;
        if (i<n-2)
            A(i, i+1) = lambda(i); % 对角线上方
        end
        if (i>1)
            A(i, i-1) = mu(i); % 对角线下方
        end
    end
    %继续修改向量d
    d(1)=d(1)-mu(1)*M1;
    d(n-2)=d(n-2)-lambda(n-2)*Mn;
    M=[M1;A\d;Mn]; %求出向量M
    %根据Si(x)的表达式, 我们还需要额外定义一些变量
    x1=a(2:n)-x;
    x2=x-a(1:n-1);
    m1=M(1:n-1);
    m2=M(2:n);
    n1=b(1:n-1);
    n2=b(2:n);
    S=(x1.^3).*m1./(6*h)+(x2.^3).*m2./(6*h)+(n1-h.^2.*m1/6).*x1./h+(n2-h.^2.*m2/
    S=expand(S);
    S= vpa(S,6);
end

```

4 chebyshev 插值

4.1 数学原理

chebyshev 多项式 $T_k(x) = \cos(k \arccos(x)) (-1 \leq x \leq 1), k \in \mathbb{N}$

不难看出 $T_k(x)$ 在 $[-1, 1]$ 上有 k 个零点 $x_i = \cos(\frac{2i+1}{2k}\pi), i = 0, 1, \dots, k-1$

(chebyshev 多项式的极值)

$$\max_{-1 \leq x \leq 1} |p_n(x)| \geq \max_{-1 \leq x \leq 1} |2^{1-n} T_n(x)| = 2^{1-n}$$

这表明 chebyshev 在 ∞ 范数上的逼近性是最优的

代码实现也比较容易, 我们只需将牛顿插值多项式中的等距节点改为 chebyshev 多项式的零点即可

4.2 算法实现

```
%chebyshev零点构造N5(x)
x4=cos((1:2:11)*pi/12)
disp("T5 = ")
y=Newton(x4, f(x4));
disp(y)
plot(x4, subs(y, x4), 'm')
hold on

%chebyshev零点构造N10(x)
x5=cos((1:2:21)*pi/22);
disp("T10 = ")
y=Newton(x5, f(x5));
disp(y) % 输出T10的结果
plot(x5, subs(y, x5), 'k');
```

end

5 实验结果

得到的牛顿插值多项式:

$$N5 = 1.20192 * x^4 - 1.73077 * x^2 + 0.567308$$

$$\begin{aligned} N10 = & -220.942 * x^{10} + 494.91 * x^8 - 9.50648e - 14 * x^7 - 381.434 * x^6 - \\ & 8.50398e - 14 * x^5 + 123.36 * x^4 + 2.02021e - 14 * x^3 - 16.8552 * x^2 - 6.65938e - \\ & 16 * x + 1.0 \end{aligned}$$

得到的三次样条插值多项式:

$$\begin{aligned} S10 = & 0.119274 * x^3 + 0.463081 * x^2 + 0.644325 * x + 0.338981 & x \in [-1, -0.8] \\ & 0.952866 * x^3 + 2.4637 * x^2 + 2.24482 * x + 0.76578 & x \in (-0.8, -0.6] \\ & 0.820393 * x^3 + 2.22525 * x^2 + 2.10175 * x + 0.737165 & x \in (-0.6, -0.4] \\ & 13.4126 * x^3 + 17.3359 * x^2 + 8.14602 * x + 1.54307 & x \in (-0.4, -0.2] \\ & -54.4709 * x^3 - 23.3942 * x^2 - 2.69784e - 15 * x + 1.0 & x \in (-0.2, 0] \\ & 54.4709 * x^3 - 23.3942 * x^2 + 2.69784e - 15 * x + 1.0 & x \in (0, 0.2] \\ & -13.4126 * x^3 + 17.3359 * x^2 - 8.14602 * x + 1.54307 & x \in (0.2, 0.4] \\ & -0.820393 * x^3 + 2.22525 * x^2 - 2.10175 * x + 0.737165 & x \in (0.4, 0.6] \\ & -0.952866 * x^3 + 2.4637 * x^2 - 2.24482 * x + 0.76578 & x \in (0.6, 0.8] \\ & -0.119274 * x^3 + 0.463081 * x^2 - 0.644325 * x + 0.338981 & x \in (0.8, 1] \end{aligned}$$

得到的 chebyshev 插值多项式:

$$\begin{aligned} T5 = & -4.59755e - 16 * x^5 + 0.711567 * x^4 - 1.42109e - 14 * x^3 - 1.09581 * \\ & x^2 + 0.444089 \\ T10 = & -46.6329 * x^{10} + 3.3088e - 14 * x^9 + 130.106 * x^8 - 8.53011e - 14 * \\ & x^7 - 133.445 * x^6 + 7.00218e - 14 * x^5 + 61.443 * x^4 - 2.01241e - 14 * x^3 - \end{aligned}$$

$$12.4765 * x^2 + 9.78343e - 16 * x + 1.0$$

5.1 图像绘制与分析

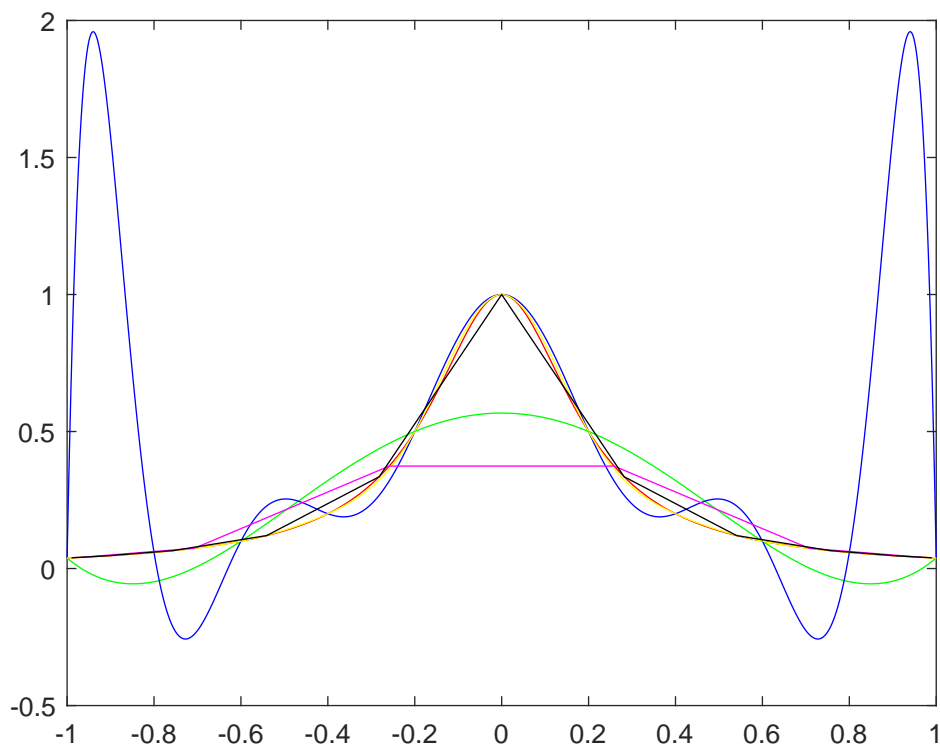


图 1: $f(x)$ 红 N_5 绿 N_{10} 蓝 S_{10} 黄 T_5 紫 T_{10} 黑

我们可以发现: $N_5(x)$ 由于插值节点较少, 插值次数较低, 拟合效果不佳; $N_{10}(x)$ 在 $[-0.2, 0.2]$ 附近拟合较好, 但在靠近-1 和 1 附近出现了明显的龙格现象

分段低次插值多项式 $S_{10}(x)$ 在整个区间上都拟合得很好

chebyshev 插值多项式 $T_5(x)$ 由于次数较低拟合效果不好, 但随着次数增加, 到了 $T_{10}(x)$, 在整个区间上都拟合得较好, 且有效消除了高次插值公式的龙格现象

A 完整代码

```
function main
a = -1;
b = 1;
x = (a:0.01:b);
plot(x, f(x), 'r'); % 画出f(x)图像
hold on
x=(a:0.001:b)
xx = (a:2/5:b);
y = Newton(xx, f(xx));
class(y)
disp("N5 = ")
disp(y) % 输出N5插值结果
plot(x, subs(y, x), 'g'); % N5图像
hold on

xx = (a:1/5:b)';
y = Newton(xx, f(xx));
disp("N10 = ")
disp(y) % 输出N10插值结果
plot(x, subs(y, x), 'b'); % N10图像
hold on

xxx= (a:2/10:b)';
y = sanc1(xxx, f(xxx), fff(a), fff(b));
disp("S10 = ")
disp(y)
```

```
for i = 1 : 10
    l = xxx(i);
    r = l + 1/5;
    x = l:0.01:r;
    plot(x, subs(y(i), x), 'y') % S10图像
    hold on
end
```

```
x4=cos((1:2:11)*pi/12)
disp("T5 = ")
y=Newton(x4, f(x4));
disp(y)
plot(x4, subs(y, x4), 'm')
hold on %chebyshev零点构造N5(x)
```

```
x5=cos((1:2:21)*pi/22);
disp("T10 = ")
y=Newton(x5, f(x5));
disp(y) % 输出T10的结果
plot(x5, subs(y, x5), 'k'); %chebyshev零点构造N10(x)
end
```

```
function N = Newton(a,b)
```

%N是求得的插值多项式系数向量，a是插值点构成的列向量，b是插值点对应函数值的列

```
n=length(a);
```

```
df=zeros(n,n);
```

```
df(:,1)=b;
```

```

syms x
N=df(1,1);
acc=1;
for i=2:n
    for k=i:n
        df(k,i)=(df(k,i-1)-df(k-1,i-1))/(a(k)-a(k+1-i));
    end
    acc=acc*(x-a(i-1));
    N=N+df(i,i)*acc;
end
N=expand(N);
N=vpa(N,6);
end

```

```

function y = f(x)
    y = 1 ./ (1 + 25 * x.^2);
end
function y = fff(x)
    y = 50 * (75*x.*x-1) ./ (1+25*x.*x)^3;
end

```

```

function S=sanci(a, b, M1, Mn)%a是插值节点构成的向量，b是对应的值
n=length(a);
syms x
h=a(2:n)-a(1:n-1); % 求解向量h
mu=h(1:n-2) ./ (h(1:n-2) + h(2:n-1)); %
lambda= 1-mu; % 求解向量lambda

```

```

h1 = h(1:n-2); % 定义中间变量h1
h2 = h(2:n-1); % 定义中间变量h2
d = 6*((b(3:n)-b(2:n-1))./h2-(b(2:n-1)-b(1:n-2))./h1)./(h1+h2);
A = zeros(n-2, n-2); % 初始化系数矩阵A
for i=1:n-2
    A(i, i) = 2;
    if (i<n-2)
        A(i, i+1) = lambda(i); % 对角线上方
    end
    if (i>1)
        A(i, i-1) = mu(i); % 对角线下方
    end
end
%继续修改向量d
d(1)=d(1)-mu(1)*M1;
d(n-2)=d(n-2)-lambda(n-2)*Mn;
M=[M1;A\d;Mn]; %求出向量M
%根据Si(x)的表达式, 我们还需要额外定义一些变量
x1=a(2:n)-x;
x2=x-a(1:n-1);
m1=M(1:n-1);
m2=M(2:n);
n1=b(1:n-1);
n2=b(2:n);
S=(x1.^3).*m1./(6*h)+(x2.^3).*m2./(6*h)+(n1-h.^2.*m1/6).*x1./h+(n2-h.^2.*m2./h);
S=expand(S);
S= vpa(S,6);

```

end