作业一

姓名李田所学号 PB18114514 日期 2020 年 10 月 23 日

第一题 重心插值公式 (barycentric interpolation formula)

课堂上我们已经讨论过了基于 $n+1$ 个插值点 $\{x_j\}_{j=0}^n$ 的 Lagrange 插值多项式:

$$p(x) = \sum_{j=0}^{n} f_j \ell_j(x) \tag{1}$$

此处，$f_j = f(x_j)$。Lagrange 插值基函数 (Lagrange polynomial)

$$\ell_j(x) = \frac{\prod_{k \neq j} (x - x_k)}{\prod_{k \neq j} (x_j - x_k)} \tag{2}$$

满足

$$\ell_j(x_k) = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases}$$

(a) 由节点多项式定义

$$\ell(x) = \prod_{k=0}^{n} (x - x_k)$$

可以推出

$$\ell'(x) = \sum_{i=1}^{n} \prod_{k \neq i} (x - x_k)$$

和

$$\ell'(x_j) = \prod_{k \neq i} (x_j - x_k)$$

计算可得

$$\frac{\ell(x)}{\ell'(x_j)(x - x_j)} = \frac{\prod_{k \neq j}(x - x_k)}{\prod_{k \neq j}(x_j - x_k)} = \ell_j(x) \tag{3}$$

将 (3) 代入原多项式

$$p(x) = \sum_{j=0}^{n} f_j \ell_j(x) = \sum_{j=0}^{n} f_j \frac{\ell(x)}{\ell'(x_j)(x - x_j)} = \ell(x) \sum_{j=0}^{n} \frac{\lambda_j}{x - x_j} f_j \tag{4}$$

(b) 不妨设 $f(x) \equiv 1$，利用误差的余项关系式

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k) \tag{5}$$

可得

$$\sum_{j=0}^{n} \ell_j(x) = 1 \tag{6}$$

利用 (6) 可以计算

$$1 = \sum_{j=0}^{n} \ell_j(x) = \sum_{j=0}^{n} \frac{\ell(x)}{\ell'(x_j)(x - x_j)} \tag{7}$$

即

$$\ell(x) = \frac{1}{\sum_{j=0}^{n} \frac{1}{\ell'(x_j)(x-x_j)}} = \frac{1}{\sum_{j=0}^{n} \frac{\lambda_j}{x-x_j}} \tag{8}$$

将 (8) 代入 (4)，得

$$p(x) = \sum_{j=0}^{n} \frac{\lambda_j f_j}{x - x_j} / \sum_{j=0} n \frac{\lambda_j}{x - x_j} \tag{9}$$

(c) 我没整出来淦再查下别的书

(d) 运行 Matlab 程序如下：

```
        LW = 'linewidth'; lw = 2;


n = 5000;
x = zeros(n+1,1);%%x1插值点5001个
m = 10000;
xx = linspace(-1, 1, m)';%%xx等距取样点10001个
F = @(x)tanh(20*sin(12.*x))+0.02*exp(3.*x).*sin(300.*x);
f = F(x);
p1 = zeros(m, 1);%%分子
p2 = zeros(m, 1);%%分母
p  = zeros(m, 1);%%总函数
%%求重心插值公式
for j=1:n+1
    x(j)=cos((j-1)*pi/n);
end
p1 = 0.5*(F(1)./(xx-1)+((-1)^n)*F(-1)./(xx+1));
p2 = 0.5*(1./(xx-1)+((-1)^n)./(xx+1));
for k = 2: n
        p1 = p1 + ((-1)^(k-1))*F(x(k-1))./(xx-x(k-1));
        p2 = p2 + ((-1)^(k-1))./(xx-x(k-1));
```

```
end
p = p1 ./ p2;

figure(1)
plot(xx, F(xx), 'k', LW, lw), hold on
plot(xx, p,'b', LW, lw)
legend('exact', 'interpolant', 'location', 'nw')

figure(2)
plot(2)
semilogy(xx, abs(F(xx) - p), 'k', LW, lw), hold on
legend('error', 'error bound', 'location', 'se')
```

程序运行结果为:

第二题 MATLAB 程序显示如下:

```
for q=6:12
n = 2^q;
x = linspace(-1, 1, n+1);
F = @(x)exp(3.*cos(pi.*x));
f = F(x);
%%
h = diff(x);
df = diff(f);
lambda = h(2:n) ./ (h(2:n) + h(1:n-1));
d = 6 * ( df(2:n) ./ h(2:n) - df(1:n-1)
         ./ h(1:n-1) ) ./ (h(2:n) + h(1:n-1));
mu = 1-lambda;
%%
%第一类边界条件
M0 = 0;
Mn = 0;
A1 = diag(2*ones(n-1,1)) + diag(lambda(1:n-2), 1)
                            + diag(mu(2:n-1), -1);
D1 = [d(1) - mu(1)*M0; d(2:n-2); d(n-1)
                            - lambda(n-1)*Mn];
```

```
    M1 = A1\D1;
    M1 = [M0; M1; Mn];
    %%
    figure(1)
    CubicSpline(x, F, h, M1, q);hold on
    end
%%函数
function S = CubicSpline(x, F, h, M, q )
LW = 'linewidth'; lw = 2;
n = size(x) - 1;
f = F(x);
for k = 1:n
    m = 4;
    xx = linspace(x(k), x(k+1), m)';
    S = ( (x(k+1)-xx).^3*M(k) + (xx-x(k)).^3*M(k+1) ) / (6*h(k)) +...
        ( (x(k+1)-xx)*f(k) + (xx-x(k))*f(k+1) ) / h(k) -...
        h(k) * ( (x(k+1)-xx)*M(k) + (xx-x(k))*M(k+1) ) / 6;
    subplot(2,1,1)
    p1 = plot(xx, F(xx), 'k', LW, lw); hold on
    p2 = plot(xx, S, 'r', LW, lw); hold on

    error = abs(F(xx) - S);
end
mmax = max(error);
subplot(2,1,2)
    p3 = loglog(q, mmax,'o');hold on
```

输出结果:

其他两个边界条件类似时间问题还没写完程序

第三题 MATLAB 程序显示如下:

```
    x = [-0.7 -0.5 0.25 0.75];
y = [0.99 1.21 2.57 4.23];
y1 = log(y);
M = zeros(2,2);
uncol = zeros(2,1);
```

```
col = zeros(2,1);
xx = linspace(-1,1,1000);
%%线性拟合
M(1,1) = 4;
for i=1:1:4
M(1,2) = M(1,2) + x(i);
M(2,1) = M(2,1) + x(i);
M(2,2) = M(2,2) + (x(i))^2;
col(1) = col(1) + y1(i);
col(2) = col(2) + x(i)*y1(i);
end
uncol = inv(M)*col;
F =@(x) exp(uncol(1))*exp(uncol(2)*x);

figure(1)
plot(x, y, 'o'); hold on
plot(xx,F(xx));
```

输出图像: