计算方法作业一

姓名　龚小航　　学号　**PB18151866**　　日期　**2020.10.18**

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

$a)$ 用节点多项式

$$\ell(x)=\prod_{k=0}^{n}(x-x_k) \tag{1}$$

表示 Lagrange 插值基函数：

先写出拉格朗日插值基函数以及节点多项式的导数在 $x=x_j$ 处的值：

$$\ell_j(x)=\frac{\prod_{k\neq j}(x-x_k)}{\prod_{k\neq j}(x_j-x_k)};\qquad \ell'(x_j)=\frac{d}{dx}\ell(x)_{x=x_j}=\prod_{k\neq j}(x_j-x_k) \tag{2}$$

对比发现恰好 $\ell'(x_j)$ 就是拉格朗日插值基函数的分母。由此可得它们的关系：

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

由此可以推导出重心插值公式的第一形式：

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

其中利用了替换 $\lambda_j=1/\ell'(x_j)$,λ_j 称为插值权重。

$b)$ 证明所有的 Lagrange 插值基函数的和为 1：

观察公式 (1) 的形式，可取被插函数 $f(x)=1$, 在这个函数上插值，无论选取几个插值点，都有：

$$p(x)=\sum_{j=0}^{n}f_j\ell_j(x)=\sum_{j=0}^{n}\ell_j(x)=1 \tag{5}$$

这是由于 n 次插值多项式的误差为：

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i)=0 \tag{6}$$

由于 f 是常数，一阶及以上的导数都为 0。

再由此推出重心插值公式的第二形式：

$$\begin{aligned}
p(x) &= \sum_{j=0}^{n} f_j \ell_j(x) = \sum_{j=0}^{n} f_j \frac{\prod_{k\neq j}(x-x_k)}{\prod_{k\neq j}(x_j-x_k)} \\
&= \prod_{i=0}^{n}(x-x_i)\sum_{j=0}^{n}\frac{f_j}{(x-x_j)\prod_{k\neq j}(x_j-x_k)} \\
&= \frac{\prod_{i=0}^{n}(x-x_i)\sum_{j=0}^{n}\frac{f_j\lambda_j}{(x-x_j)}}{\sum_{j=0}^{n}\ell_j(x)} = \frac{\sum_{j=0}^{n}\frac{f_j\lambda_j}{(x-x_j)}}{\sum_{j=0}^{n}\frac{\prod_{k\neq j}(x-x_k)}{\prod_{i=0}^{n}(x-x_i)\prod_{k\neq j}(x_j-x_k)}} \\
&= \frac{\sum_{j=0}^{n}\frac{f_j\lambda_j}{(x-x_j)}}{\sum_{j=0}^{n}\frac{1}{(x-x_j)\prod_{k\neq j}(x_j-x_k)}} \\
&= \frac{\sum_{j=0}^{n}\frac{f_j\lambda_j}{(x-x_j)}}{\sum_{j=0}^{n}\frac{\lambda_j}{(x-x_j)}} \qquad (7)
\end{aligned}$$

$c)$ 即需要证明的是：

$$x_j = cos(\frac{j\pi}{n}), j = 0, 1, 2, \cdot\cdot\cdot, n \qquad (8)$$

$$\prod_{k=0,k!=j}^{n}\frac{1}{x_j-x_k} = \begin{cases} \frac{2^{n-2}}{n} & j=0 \\ \frac{2^{n-1}}{n}(-1)^j & 0<j<n \\ \frac{2^{n-2}}{n}(-1)^n & j=n \end{cases}$$

为证明这个关系式，由结果的特点，可以利用数学归纳法。证明 $j = j$ $j = j+1$ 之间的关系即可：

先计算 $j = 0$ 的情况：$x_0 = cos(0) = 1$

$$\begin{aligned}
\prod_{k=1}^{n}(1-cos\frac{k\pi}{n}) &= \prod_{k=1}^{n-1}(sin^2\frac{k\pi}{2n}) = \prod_{k=1}^{n-1}(sin\frac{k\pi}{2n}cos\frac{(n-k)\pi}{2n}) \qquad (9)\\
&= \prod_{k=1}^{n-1}cos(\frac{k\pi}{2n})sin(\frac{(n-k)\pi}{2n}) \quad (k\Rightarrow n-k) \qquad (10)\\
&= (\prod_{k=1}^{n-1}cos(\frac{k\pi}{2n})sin(\frac{k\pi}{2n})cos(\frac{(n-k)\pi}{2n})sin(\frac{(n-k)\pi}{2n}))^{\frac{1}{2}} \qquad (11)\\
&= \frac{1}{2^{n-1}}\prod_{k=1}^{n-1}sin(\frac{k\pi}{n}) \qquad (12)
\end{aligned}$$

再计算这个求和式：

$$\prod_{k=1}^{n-1} sin(\frac{k\pi}{n}) = \prod_{k=1}^{n-1}(\frac{e^{\frac{ik\pi}{n}} - e^{\frac{-ik\pi}{n}}}{2i}) = \frac{1}{(2i)^{n-1}}\prod_{k=1}^{n-1}(e^{\frac{ik\pi}{n}} - e^{\frac{-ik\pi}{n}}) \tag{13}$$

$$= \frac{1}{(2i)^{n-1}}(\prod_{k=1}^{n-1} e^{\frac{-ik\pi}{n}})(\prod_{k=1}^{n-1}(e^{\frac{2ik\pi}{n}} - 1)) \tag{14}$$

$$= \frac{1}{(2i)^{n-1}} e^{-i\frac{(n-1)\pi}{2}} \prod_{k=1}^{n-1}(\xi^k - 1) \tag{15}$$

$$(f(x) = \prod_{k=1}^{n-1}(x - \xi^k) = \frac{x^n - 1}{x - 1} = \Sigma_{k=0}^{n-1}\xi^k) \tag{16}$$

$$= \frac{1}{(2i)^{n-1}} e^{-i\frac{(n-1)\pi}{2}}(-1)^{n-1} * n \tag{17}$$

$$= \frac{n}{2^{n-1}} \tag{18}$$

$$\Longrightarrow \prod_{k=1}^{n} \frac{1}{1 - cos(\frac{k\pi}{n})} = \frac{2^{n-2}}{n}$$

(19)

再计算其递推式：

即需要证明 $j = j + 1$ 时与 $j = j$ 时相比仅多出一个负号。

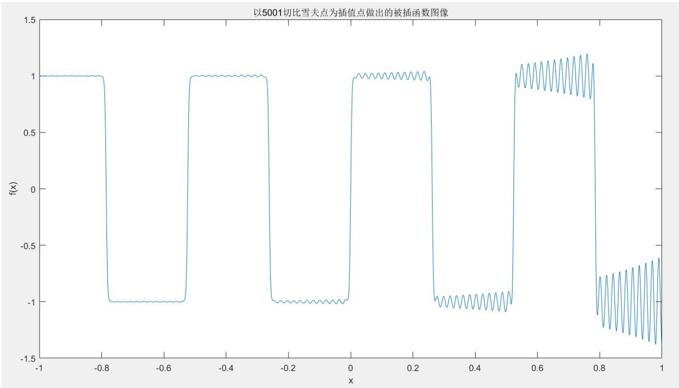
未证完整

d) 代码如下所示：

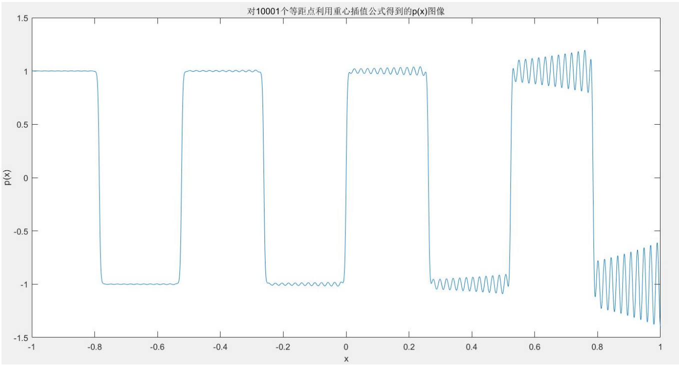
```
clear, clc, clf
LW = 'linewidth'; lw = 2;
format long;

n = 5000;
x = zeros(1,5001);
for i=1:n+1
   x(i) = cos((i-1)*pi/n);
end
F = @(x) tanh(20.*sin(12.*x))+0.02.*exp(3.*x)
         .*sin(300.*x);
f = F(x);

figure(1)
plot(x,F(x));
title('以5001切比雪夫点为插值点做出的被插函数图像')
xlabel('x')
ylabel('f(x)')


t = linspace(-1, 1, 2*n+1)';
vart=check(t);

figure(2)
plot(t,vart);
title('对10001个等距点利用重心插值公式得到的p(x)图像')
xlabel('x')
ylabel('p(x)')

figure(3)
semilogy(t,abs(vart-F(t)));
title('10001个点的|F(x)-p(x)|')
xlabel('x')
ylabel('log R')
```

```matlab
function G = check(t)   %%做出10001个点得到的p(x)
i = 1;
     n = 5000;
    x = zeros(1,5001);
    for i=1:n+1
      x(i) = cos((i-1)*pi/n);
    end
    F = @(x) tanh(20.*sin(12.*x))+0.02.*exp(3.*x)
             .*sin(300.*x);

     r1=(-F(x(1))./(t-x(1)))./2+((-1).^(n+1)
        .*F(x(n+1))./(t-x(n+1)))./2;
     r2=((-1)./(t-x(1)))./2+((-1).^(n+1)
         ./(t-x(n+1)))./2;
     for i = 2 : n
          r1=r1+(((-1).^i.*F(x(i))./(t-x(i))));
          r2=r2+(((-1).^i./(t-x(i))));
     end
     G = r1./r2;
end
```

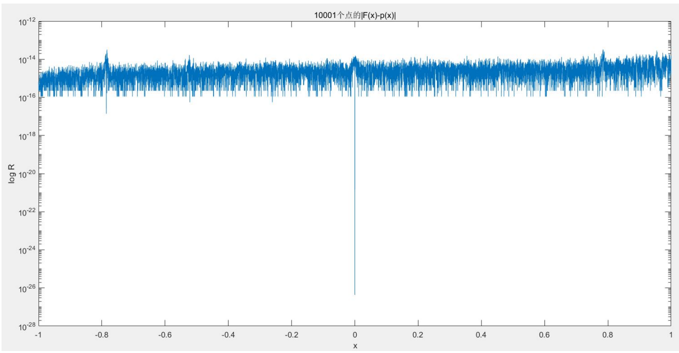
由题意做出 5001 个切比雪夫点，将它们用作插值点构造标准函数 $F(x)$，再在 $[-1,1]$ 上均匀取 10001 个点作重心插值得到插值函数 $p(x)$。最后做出 10001 个点与其误差组成的 y 轴半对数图。结果如下所示：



(a) 利用 5001 个切比雪夫插值点作出的原函数 f(x)



(b) 用 10001 个等距点重心插值出的 p(x)



(c) |p(x)-F(x)|

图 1: 第一题 d 问插图

第二题 MATLAB 程序显示如下:

```
clear, clc, clf
LW = 'linewidth'; lw = 2;
format long;

[p61,p62,p63]=T2(6);
[p71,p72,p73]=T2(7);
[p81,p82,p83]=T2(8);
[p91,p92,p93]=T2(9);
[p101,p102,p103]=T2(10);
[p111,p112,p113]=T2(11);
[p121,p122,p123]=T2(12);

y1 = [p61,p71,p81,p91,p101,p111,p121];
y2 = [p62,p72,p82,p92,p102,p112,p122];
y3 = [p63,p73,p83,p93,p103,p113,p123];
k = 6:1:12;
x = 2.^k;

figure(1)
pic1=loglog(x,y1,'k');hold on
pic2=loglog(x,y2,'g');hold on
pic3=loglog(x,y3,'r');hold on
legend([pic1,pic2,pic3], '第一类边界条件',
        '第二类边界条件','第三类边界条件')
title('第二题误差图')
xlabel('log n')
ylabel('log max{|S-F(x)|}')


function [p1,p2,p3]=T2(k)
    n = 2^k;

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];
%%
%第二类边界条件
m0 = 0;
mn = 0;
lambda2 = [1; lambda];
mu2 = [mu; 1];
d0 = 6 * ( df(1) / h(1) - m0 ) / h(1);
dn = 6 * ( mn - df(n) / h(n) ) / h(n);
D2 = [d0; d; dn];
A2 = diag(2*ones(n+1,1)) + diag(lambda2, 1)
     + diag(mu2, -1);
M2 = A2\D2;
%%
%第三类边界条件
lambda0 = h(1) / (h(1) + h(n));
lambda3 = [lambda0; lambda(1:n-2)];
mu0 = 1 - lambda0;
```

```
d0 = 6 * (df(1) ./ h(1) - df(n) ./ h(n))
     / (h(1) + h(n));
D3 = [d0; d];
A3 = diag(2*ones(n,1)) + diag(lambda3, 1)
     + diag(mu, -1);
A3(1, n) = mu0;
A3(n, 1) = lambda(n-1);
M3 = A3\D3;
M3 = [M3; M3(1)];

%%
p1 = CubicSpline(x, F, h, M1);
%display(p1);

p2 = CubicSpline(x, F, h, M2);
%display(p2);

p3 = CubicSpline(x, F, h, M3);
%display(p3);
end


function result = CubicSpline(x, F, h, M)
n = size(x) - 1;
f = F(x);
result = 0;
for k = 1:n
    m = 6;
    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;
```

```
        resulti=max(abs(S-F(xx)));
        if(resulti>result)
            result=resulti;
        end
    end

    end
```

本题可以通过自定义函数，通过输入 k 的值来确定在一、二、三类边界条件下 $[x_[k], x_{k+1}]$ 上误差的最大值并返回三个值。在函数外的脚本主体部分比较简单，只负责将 $k = 6:1:12$ 时函数的返回值都储存在一个变量内，再将其作与 n 的关系图即可。脚本运行结果如下图所示：

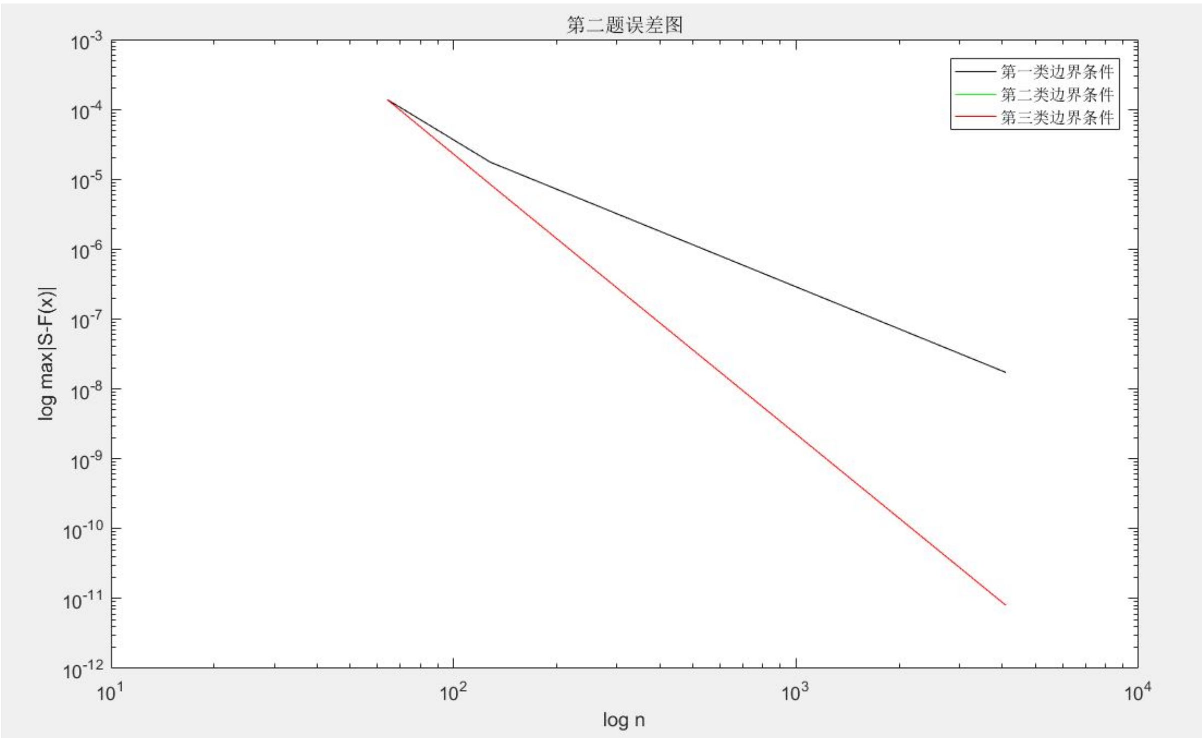


图 2: 第二题误差图

其中第二类和第三类边界条件下误差差值不是非常明显。在这幅图中，线的斜率表示当 n 的规模以 2^n 增大时，误差随插值点规模减小的速率。

第三题　对下列数据进行形如 $y = ae^{bx}$ 形式的最小二乘拟合：

x_i	-0.70	-0.50	0.25	0.75
y_i	0.99	1.21	2.57	4.23

由课本 53 页的例子，多项式拟合的方法可以推广到其他类型的函数拟合。对于本题中的形式 $y = ae^{bx}$，先对其取对数，令 $\hat{y}_i = ln(y_i)$，然后对 $(x_i, \hat{y}_i), i = 1, 2, \cdots, m$ 作形如 $ln\varphi(x) = lna + bx$ 的线性拟合。即求 $c_0 = lna, c_1 = b$ 使得 $\hat{Q}(c_0, c_1) = \sum_{i=1}^{4}(c_0 + c_1 x_i - \hat{y}_i)$ 达到最小值。
直接在 matlab 中调用函数解决这个线性拟合问题:

图 3: 第三题求解代码

即得到结果 $b = 1.0020, lna = 0.6918 \Rightarrow a = 1.9972$。即拟合结果为：

$$y = 1.9972e^{1.0020x} \tag{20}$$

所得函数的误差 2 范数已在图中给出，为：

$$s = 3.7880 * 10^{-5} \tag{21}$$