

函数逼近与数据拟合上机实习报告

姓名:刘群 学号:2014211591 院系: 地学中心

Email: liu-q14@mails.tsinghua.edu.cn

(本文档由L^AT_EX编写)

1 问题的描述

设 $f(x) = x^2 \ln(2+x)$, $x \in [-1, 1]$, 试求出权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式. 另外用Chebyshev截断级数的办法和插值余项极小化方法分别给出近似最佳一致逼近的三次多项式, 并画图比较.

2 方法描述

在数值计算时我经常要计算数学函数的值, 因此构造出简单的可计算的近似公式就非常重要. 函数逼近就是要找这样一个函数的近似公式, 具体可表述如下:

设 F 为定义在区间 $[a, b]$ 上的某类函数组成的线性空间, Φ 为 F 的子集, 对于 $f \in F$, 我们要求 $p \in \Phi$, 使得 p 和 f 之差在某种度量意义下达到最小. 通常 F 为 $C[a, b]$, $L_2[a, b]$ 等, Φ 为简单的便于计算的函数集合, 通常取多项式、有理式及三角多项式等.

2.1 最佳平方逼近

设 f 为定义在 $[a, b]$ 上的实值函数,

$$\|f(x)\|_2 = \sqrt{\int_a^b \rho(x)[f(x)]^2 dx}$$

为函数 f 的2-范数, 其中 ρ 为 $[a, b]$ 上的权函数. 令

$$L_\rho^2[a, b] = \{f : \|f(x)\|_2 < \infty\}$$

这样的话, $L_\rho^2[a, b]$ 就是一个线性空间.

设 $\varphi_0, \varphi_1, \dots, \varphi_n$ 为 $L^2_\rho[a, b]$ 的 $n+1$ 个线性无关的函数, 记 $\Phi = \text{span}\{\varphi_0, \varphi_1, \dots, \varphi_n\}$, 即 Φ 为 $\varphi_0, \varphi_1, \dots, \varphi_n$ 的线性组合. 此时任取 $s \in \Phi$, 有

$$s(x) = \sum_{j=0}^n a_j \varphi_j(x), x \in [a, b]. \quad (1)$$

我们如果记 $s^* \in \Phi$ 是 f 在 Φ 中的最佳平方逼近, 则有

$$\|f - s^*\|_2 = \inf_{s \in \Phi} \|f - s\|_2 \quad (2)$$

事实上, 求 s^* 等价于求多元函数

$$F(a_0, a_1, \dots, a_n) = \int_a^b \rho(x) \left[\sum_{j=0}^n a_j \varphi_j(x) - f(x) \right]^2 dx$$

的极小值. 我们可以利用多元函数求极值的必要条件

$$\frac{\partial F}{\partial a_k} = 0, k = 0, 1, \dots, n.$$

进行求解, 化简之后即为

$$\sum_{j=0}^n (\varphi_k, \varphi_j) a_j = (f, \varphi_k), k = 0, 1, \dots, n. \quad (3)$$

其中,

$$(\varphi_k, \varphi_j) = \int_a^b \rho(x) \varphi_k(x) \varphi_j(x) dx, \quad (f, \varphi_k) = \int_a^b \rho(x) f(x) \varphi_k(x) dx.$$

方程(3)称为法方程, 写成矩阵的形式如下:

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{bmatrix} \quad (4)$$

可以看出, 系数矩阵是一个对称阵.

下面我们根据上面的介绍, 求解 $f(x) = x^2 \ln(2+x), x \in [-1, 1]$, 在权函数 $\rho(x) = 1$ 时在 $\mathcal{P}_3 = \text{span}\{1, x, x^2, x^3\}$ 中的最佳平方逼近三次多项式. 我们可以得到法方程矩阵中的各个元素如下:

$$\begin{aligned}
(\varphi_0, \varphi_0) &= \int_{-1}^1 1 dx = 2, & (\varphi_0, \varphi_1) &= \int_{-1}^1 x dx = 0, & (\varphi_0, \varphi_2) &= \int_{-1}^1 x^2 dx = \frac{2}{3} \\
(\varphi_0, \varphi_3) &= \int_{-1}^1 x^3 dx = 0, & (\varphi_1, \varphi_1) &= \int_{-1}^1 x^2 dx = \frac{2}{3}, & (\varphi_1, \varphi_2) &= \int_{-1}^1 x^3 dx = 0 \\
(\varphi_2, \varphi_2) &= \int_{-1}^1 x^4 dx = \frac{2}{5}, & (\varphi_2, \varphi_3) &= \int_{-1}^1 x^5 dx = 0, & (\varphi_3, \varphi_3) &= \int_{-1}^1 x^6 dx = \frac{2}{7} \\
(f, \varphi_0) &= \int_{-1}^1 x^2 \ln(2+x) dx = 0.406948, & (f, \varphi_1) &= \int_{-1}^1 x^3 \ln(2+x) dx = 0.213537, \\
(f, \varphi_2) &= \int_{-1}^1 x^4 \ln(2+x) dx = 0.237508, & (f, \varphi_3) &= \int_{-1}^1 x^5 \ln(2+x) dx = 0.1534560
\end{aligned}$$

从而法方程变为

$$\begin{bmatrix} 2 & 0 & \frac{2}{3} & 0 \\ 0 & \frac{2}{3} & 0 & \frac{2}{5} \\ \frac{2}{3} & 0 & \frac{2}{5} & 0 \\ 0 & \frac{2}{5} & 0 & \frac{2}{7} \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0.406948 \\ 0.213537 \\ 0.237508 \\ 0.153460 \end{pmatrix} \quad (5)$$

可以解得 $a_0 = 0.0124894, a_1 = -0.0122489, a_2 = 0.572954, a_3 = 0.5542580$. 从而有 $f(x) = x^2 \ln(2+x), x \in [-1, 1]$, 在权函数 $\rho(x) = 1$ 时在 $\mathcal{P}_3 = \text{span}\{1, x, x^2, x^3\}$ 中的最佳平方逼近三次多项式为

$$p_3(x) = 0.0124894 - 0.0122489x + 0.572954x^2 + 0.5542580x^3 \quad (6)$$

2.2 Chebyshev 截断级数法

Chebyshev 多项式是在区间 $[-1, 1]$ 上的关于权函数 $\rho(x) = \frac{1}{\sqrt{1-x^2}}, x \in [-1, 1]$ 的正交多项式序列 T_n , 其表达式为

$$T_n(x) = \cos(n \arccos x), n \geq 0. \quad (7)$$

Chebyshev 多项式具有正交性, 具体来说, 有

$$(T_n, T_m) = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} T_n(x) T_m(x) dx = \begin{cases} 0, & n \neq m, \\ \frac{\pi}{2}, & n = m \neq 0, \\ \pi, & n = m = 0. \end{cases} \quad (8)$$

另外, 需要注意的是 Chebyshev 多项式具有递推公式

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (9)$$

其中, $T_0(x) = 1, T_1(x) = x$.

下面我们说一下如何用广义Fourier级数来进行函数的逼近. 如果 $f \in L_p^2[a, b], \psi_j \in L_p^2[a, b], j = 0, 1, \dots$, 为正交函数组, 则称

$$a_j = (f, \psi_j) = \int_a^b \rho(x) f(x) \psi_j(x) dx, j = 0, 1, \dots \quad (10)$$

为 f 的广义Fourier系数, 相应的级数 $\sum_{j=0}^{\infty} a_j \psi_j(x)$ 称为 f 的广义Fourier级数. 而函数的最佳平方逼近用正交函数组的表示的时候, 其结果正是广义Fourier级数的部分和, 即

$$S_n^*(x) = \sum_{j=0}^n \frac{1}{\|\psi_j\|_2^2} (f, \psi_j) \psi_j(x). \quad (11)$$

下面我们就用Chebyshev正交函数组来做 $f(x)$ 的最佳平方逼近. 利用(11)式和Chebyshev多项式的递推表达式(9), 我们有

$$S_n^*(x) = \sum_{j=0}^n \frac{1}{\|T_j\|_2^2} (f, T_j) T_j(x). \quad (12)$$

具体到这个问题上, 我们只是选择3次多项式进行逼近, 因此这里 $n = 3$. 从而有

$$T_n(n \leq 3) = \begin{cases} T_0(x) = 1, \\ T_1(x) = x, \\ T_2(x) = 2x^2 + 1, \\ T_3(x) = 4x^3 - 3x \end{cases}$$

利用Matlab编程, 我们可以得到每项前面的系数, 因此最后的表达式为

$$\begin{aligned} S_n^*(x) &= 0.2940T_0(x) + 0.4051T_1(x) + 0.2754T_2(x) + 0.1405T_3(x) \\ &= 0.2940 + 0.4051x + 0.2754(2x^2 + 1) + 0.1405(4x^3 - 3x) \end{aligned}$$

2.3 插值余项极小化方法

我们知道, 在等距节点的插值时, 比如说Lagrange插值法进行插值时, 当次数比较高时, 往往会出现Runge现象, 因此我们需要想办法加以克服. 在上一次实验中, 我们采用了分段低次插值的办法解决这个问题. 这里我们换一种思路, 即选用非等距节点进行Lagrange插值, 具体来说, 我们所选的节点 x_0, x_1, \dots, x_n 是Chebyshev 正交多项式 T_{n+1} 的根(由于这里的区间与标准形式一样, 都是 $[-1, 1]$, 因此我们不需要进行坐标转换).

我们知道 $T_{n+1}(x)$ 在 $[-1,1]$ 内有 $n+1$ 个不同的零点, 即

$$x_{i+1} = \cos \frac{2i+1}{2(n+1)}\pi, \quad i = 0, 1, \dots, n. \quad (13)$$

然后我们按照Lagrange插值公式进行计算, 即

$$\tilde{L}_n(x) = \sum_{i=0}^n f(x_i)l_i(x) = \sum_{i=0}^n x_i^2 \ln(x_i+2)l_i(x) \quad (14)$$

其中

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

3 方案设计

我们通过编写Matlab 程序来对各种逼近方法进行求解. 在程序approx.m中, 第一部分关于权函数 $\rho(x) = 1$ 时的最佳平方逼近的三次多项式的求解, 我们调用了Matlab内置函数int和自定义符号函数通过循环来求解定积分, 这样就可以通过循环将(4)式所示的法方程的矩阵方程中的各个量求出来, 然后就可以调用Matlab的左除运算求解出系数 a_0, a_1, a_2, a_3 . 求解出系数之后, 我们可以将其代入最佳逼近函数的表达式中, 这样就得到了最佳逼近函数. 然后就可以画图对其与原函数的图像进行比较.

在程序的第二部分, 我们主要是求解了用Chebyshev级数来逼近函数. 我们主要利用递推关系式(9)来求每一项Chebyshev多项式, 然后按照(10)式和(11)式来求解级数的系数, 另外注意到在(11)式中已经做了归一化处理. 按照循环可以将上述系数和Chebyshev多项式一一求解出来, 最后把它们在 $[-1, 1]$ 的函数值求出来即可. 同样, 我们绘制了这种方法与原函数的图像. 在编写这个程序的过程中, 在求内积时由于要用到一些积分, 因此我就直接调用了Matlab内置的函数quad来求解.

在求解插值余项最小化方法的逼近时, 我首先编写程序求解出Chebyshev多项式的根, 然后将这些根作为插值节点传到Lagrange插值的子程序Lagrange.m中去, 这样Lagrange.m程序就会返回相应的插值结果. 在主程序中, 我利用返回的结果进行画图分析.

最后, 我把三种方法的结果的误差与原函数进行了比较, 绘制了绝对误差的图像, 由于 $f(x) = x^2 \ln(x+2)$ 在 $x=0$ 处为0, 因此不便计算相对误差, 故只算了绝对误差.

4 计算结果及其分析

下面是我们采用不同的逼近方式所得到的结果, 如Figure 1 所示. 从中可以看出, 无论是权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式, 还是Chebyshev截断级数方法, 还是插值余项极小化方法, 都给出了很好的近似结果. 特别需要说明的是, 采用插值余项极小化方法后, 没有产生Lagrange插值所出现的振荡现象, 因此可以说, 这种方法更好. 同时我们需要指出的是这几种方法还是有些差别的, 比如说普通的权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式需要计算的量比较多, 比较麻烦. 但是当我们选用Chebyshev截断级数的方法时, 我们只需要计算法方程矩阵主对角线上的元素即可, 不需要进行太多的计算. 采用插值余项极小化方法, 我们所需要的节点是固定的, 如果另外添加节点也是不可能的.

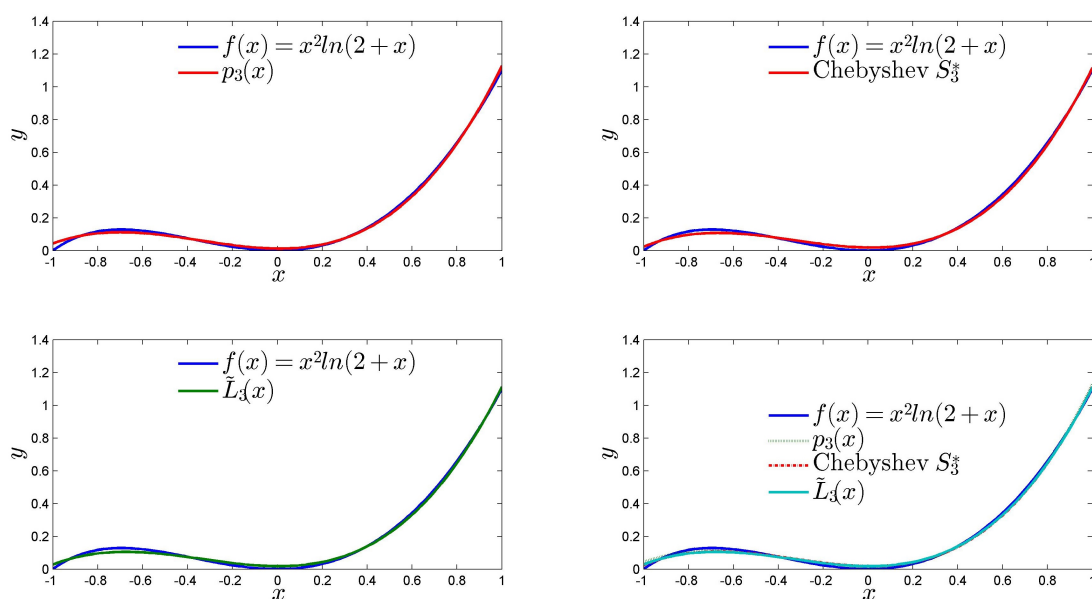


Figure 1: 各种方法的近似结果比较

Figure 2 是几种方法的绝对误差的图像, 从中可以看出权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式、Chebyshev截断级数方法和插值余项极小化方法这三种方法的误差还是很小的, 而且, 三者的误差也差不多在同一个量级上, 只是权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式在端点处的误差稍大一些.

5 结论

函数的最佳平方逼近是非常有效的一种数值计算方法, 它可以很好的去近似原函数. 通过

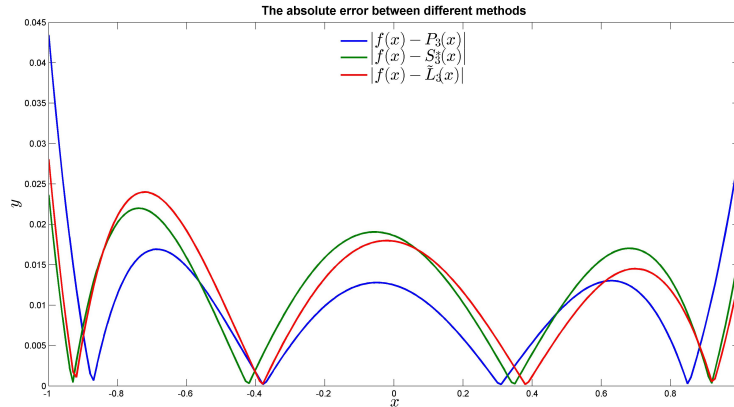


Figure 2: 各种方法的绝对误差

我们本次上机实习可以看出,无论是权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式, Chebyshev 截断级数方法, 还是插值余项极小化方法, 都给出了原函数很好的近似结果. 特别需要说明的是, 采用插值余项极小化方法后, 没有产生Lagrange插值所出现的振荡现象, 因此可以说, 这种方法比Lagrange插值方法更好, Lagrange插值方法的不稳定性主要是由于等距节点造成的, 因此这里我们选择Chebyshev多项式的根就很好的解决了插值节点等距所带来的问题. 同时我们需要指出的是这几种方法还是有些差别的, 比如说普通的权函数 $\rho(x) = 1$ 的最佳平方逼近三次多项式需要计算的量比较多, 比较麻烦. 但是当我们选用Chebyshev 截断级数的方法时, 我们只需要计算法方程矩阵主对角线上的元素即可, 不需要进行太多的计算. 采用插值余项极小化方法, 我们所需要的节点时固定的, 如果另外添加节点也是不可能的. 从误差的角度看,这三种方法的误差都比较小.

6 附录: Matlab程序

(1) approx.m

```
% 计算 $f(x)=x^2 \ln(2+x)$ 的最佳平方逼近三次多项式权函数,  $\rho(x)=1$ 
% Author: Qun Liu
% Time: 2014-12-11
clear;
clc;
close all;
% 定义多项式的阶数为3( $n=4$ ).
n = 3;
N = n+1;
% 函数的坐标值x
xx = -1:0.01:1;
% 函数 $f(x)$ 的值
f = xx.^2 .* log(2+xx);
```

```

%% 计算权函数为  $\rho(x)$  的最佳平方逼近三次多项式=1
b = zeros(N,1);
A = zeros(N, N);
% 定义符号变量, 用于求定积分
syms x
% 定义符号函数, 要逼近的函数  $f$ 
F = x^2*log(2+x);
for i = 1:N
    for j = i:N
        A(i,j) = int(x^(i-1)*x^(j-1), -1, 1);
        if i~=j
            A(j,i) = A(i,j);
        end
    end
    b(i) = int(F*x^(i-1), -1, 1);
end
% 最佳平方逼近三次多项式的系数
a = A\b;

%% 最佳平方逼近三次多项式
y1=0;
for i = 1:N
    y1 = y1 + a(i).* xx.^(i-1);
end
figure,
set(gcf,'outerposition',get(0,'screensize'));
subplot(2,2,1),
plot(xx,f,'-',xx,y1,'r-', 'linewidth', 2);
%plot(xx, [f;y1], 'linewidth', 2);
h = legend('f(x)=x^2ln(2+x)', 'p-3(x)'); %Cubic\ polynomial\ of\ best\ square\
approximation\
%h = legend('f(x)=x^2ln(2+x)', 'Cubic\ polynomial\ of\ best\ square\ approximation\
p-3(x)');
set(h, 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold', 'Box', 'off', 'Location', 'Best');
%title 原函数与最佳平方逼近三次多项式的比较(' ', 'FontSize', 14, 'FontWeight', 'bold');
hx = xlabel('x');
hy = ylabel('y');
set([hx,hy], 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold')
%% Chebyshev 截断级数
syms x
T0 = sym('1');
T1 = x;
% a 是广义傅里叶级数的系数
a2 = zeros(N, 1);
% 为权函数  $\rho$ 
rho = 1/sqrt(1-x^2);
%a2(1) = int(rho*F*T0, -1, 1)/pi;
a2(1) = quad(eval(['@(x)', vectorize(rho*F*T0)]), -1, 1)/pi;

```



```

a2(2) = quad(eval(['@(x)', vectorize(rho*F*T1)]), -1, 1)/(pi/2);
% 是函数返回值y2
y2 = 0;
y2 = y2 + a2(1)*1 + a2(2)*xx;
for i = 3:N
    T2 = 2*x*T1-T0;
    T = eval(['@(x)', vectorize(T2)]);
    a2(i) = quad(eval(['@(x)', vectorize(rho*F*T2)]), -1, 1)/(pi/2);
    %y2 = y(2) + a(i)*subs(T2, xx);
    T0 = T1;
    T1 = T2;
    y2 = y2 + a2(i)*T(xx);
end
subplot(2,2,2)
plot(xx, f, '- ', xx, y2, 'r-', 'linewidth', 2);
%plot(xx, [f;y1], 'linewidth', 2);
h = legend('f(x)=x^2ln(2+x)', 'Chebyshev_S3');
set(h, 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold', 'Box', 'off', 'Location', 'Best');
%title 原函数与(截断级数Chebyshev(n=3)的比较', 'FontSize', 14, 'FontWeight', 'bold');
hx = xlabel('x');
hy = ylabel('y');
set([hx, hy], 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold');
%% 插值余项极小化法
k = 0:n;
xn = cos((2*k+1)./(2*N).*pi);
y3 = Lagrange(n, xn, xx);
subplot(2,2,3),
plot(xx, f, xx, y3, 'linewidth', 2);
%plot(xx, [f;y1], 'linewidth', 2);
h = legend('f(x)=x^2ln(2+x)', '$\tilde{L}_3(x)$');
set(h, 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold', 'Box', 'off', 'Location', 'Best');
%title 原函数与插值余项极小化法('n=3)的比较', 'FontSize', 14, 'FontWeight', 'bold');
hx = xlabel('x');
hy = ylabel('y');
set([hx, hy], 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold');
%% 绘制三种方法对比的图像
subplot(2,2,4)
plot(xx, f, '- ', xx, y1, ': ', xx, y2, '-.', xx, y3, 'linewidth', 2);
%plot(xx, [f;y1], 'linewidth', 2);
h = legend('f(x)=x^2ln(2+x)', '$p_3(x)$', 'Chebyshev_S3', '$\tilde{L}_3(x)$');
%Cubic\ polynomial\ of\ best\ square\ approximation\
set(h, 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold', 'Box', 'off', 'Location', 'Best');
hx = xlabel('x');
hy = ylabel('y');
set([hx, hy], 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold');
set(gcf, 'PaperPositionMode', 'auto') % Use screen size

```

```

% 打印图片
print -djpeg -r300 approx.jpeg;

%% 误差
% 绝对误差
figure ,
set(gcf, 'outerposition', get(0, 'screensize'));
plot(xx, abs([f-y1; f-y2; f-y3]), 'LineWidth', 2)
h = legend('$|f(x)-P_3(x)|$', '$|f(x)-S_3^*(x)|$', '$|f(x)-\tilde{L}_3(x)|$');
set(h, 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold', 'Box', 'off', 'Location', 'Best')
title('The absolute error between different methods', 'FontSize', 14, 'FontWeight', 'bold');
hx = xlabel('$x$');
hy = ylabel('$y$');
set([hx, hy], 'Interpreter', 'Latex', 'FontSize', 14, 'FontWeight', 'bold')
set(gcf, 'PaperPositionMode', 'auto') % Use screen size
% 打印图片
print -djpeg -r300 approxabserror.jpeg;

```

(2) Lagrange.m

```

function f=Lagrange(n, xn, x)
% 插值函数 Lagrange, 为插值节点的个数  $n$ , 为要求的函数点的值  $x$ 
% Qun Liu 2014-12-13

f = zeros(size(x));
for i = 1:n+1
    li = 1;
    for j = 1:n+1
        if i~=j
            %li = li * (x(k)-xn(j))/(xn(i)-xn(j));
            li = li .* (x-xn(j))/(xn(i)-xn(j));
        end
    end
    %f(k)= f(k)+li*1/(1+25*xn(i)^2);
    f = f+ li*xn(i)^2*log(xn(i)+2);
end

```