

**计算方法上机报告**

第四次上机报告

06220143顾豪阳



2022-6-4

东南大学

电子科学与工程学院

# 一、题目简介

# 二、理论分析

第一问利用最佳平方逼近公式求出拟合系数，先计算得到基函数的向量，通过求内积值，代入得到向量的线性组合，再求得y和各个基向量的内积，最后求解线性方程组得到拟合系数，然后利用题目中的拟合误差公式代入得出拟合误差进行比较。

第二问可以使用MATLAB自带的fit函数求出待定系数，并同上求出拟合误差。

# 三、数值实验过程

对于第一问，我们首先定义一个函数OptimalSquareApproximation(x,y,n)其中函数变量分别对应自变量x，应变量y，以及拟合次数n；通过书中P224页的公式求得拟合系数，随后在main函数中定义完x和y之后，代入不同的n，求得拟合后的系数，调用plot函数将其画在一张图表中进行比较，然后利用题目中的拟合误差公式代入得出拟合误差进行比较。

对于第二问，我们首先使用库中的fittype函数定义一个含有待定系数的函数，然后将其代入到库中的fit函数，两个函数的定义分别如下：

得到待定系数之后接下来的步骤就和第一问一样了。

# 四、程序代码与结果

**定义最佳平方逼近函数的代码：**

% 最佳平方逼近

% f为待逼近函数，n为最佳逼近多项式次数，[left,right]为积分区间

function c = OptimalSquareApproximation(x,y,n)

format long;

% 定义x^i \* f(x)的匿名函数用于后续求积分

rightfun = @(i) (x'.^i \* y);

% 定义左端的系数矩阵基本形式

leftfun = @(i,j) (x'.^i \* x.^j);

% 定义线性方程的左右两端矩阵

A = zeros(n+1,n+1);

b = zeros(n+1,1);

for p = 0:n

for q = 0:n

A(p+1, q+1) = leftfun(p,q); % 计算左端系数矩阵的每一项积分

end

% 计算右端矩阵的每一项

b(p+1,1) = rightfun(p);

end

c = A\b;

% disp(x);

end

**主函数的代码：**

x=2:11;

x=x';

y=[58;50;44;38;34;30;29;26;25;24];

%[n,m]=size(x);

c=OptimalSquareApproximation(x,y,2);

format long;

two = OptimalSquareApproximation(x,y,2); % 获取最佳二次逼近多项式系数

three = OptimalSquareApproximation(x,y,3); % 获取最佳三次逼近多项式系数

four = OptimalSquareApproximation(x,y,4); % 获取最佳三次逼近多项式系数

% 对多项式系数按从高次到低次排列

% disp(two);

% disp(three);

% disp(four);

two = flip(two);

three = flip(three);

four = flip(four);

x0 = 1:0.01:13;

figure;

%plot(x,y);

plot(x,y,'r',x0,polyval(two,x0),'g',x0,polyval(three,x0),'b',x0,polyval(four,x0),'m')

xlabel('x轴');

ylabel('y轴');

legend('原曲线','最佳二次逼近多项式','最佳三次逼近多项式','最佳四次逼近多项式');

e\_two=sum((polyval(two,x)-y).^2);%二次逼近多项式拟合误差

e\_three=sum((polyval(three,x)-y).^2);%三次逼近多项式拟合误差

e\_four=sum((polyval(four,x)-y).^2);%四次逼近多项式拟合误差

syms t;

f1=fittype('a+b/t','independent','t','coefficients',{'a','b'});

[cfun1,~]=fit(x,y,f1);

f2=fittype('a+b\*log(t)','independent','t','coefficients',{'a','b'});

[cfun2,~]=fit(x,y,f2);

f3=fittype('a\*exp(b\*t)','independent','t','coefficients',{'a','b'});

[cfun3,~]=fit(x,y,f3);

f4=fittype('1/(a+b\*t)','independent','t','coefficients',{'a','b'});

[cfun4,rsquare]=fit(x,y,f4);

figure;

plot(x,y,'r',x0,cfun1(x0),'g',x0,cfun2(x0),'b',x0,cfun3(x0),'m',x0,cfun4(x0),'c');

xlabel('x轴');

ylabel('y轴');

legend('原曲线','a+b/x','a+b\*ln(t)','a\*exp(b\*t)','1/(a+b\*t)');

e\_1=sum((cfun1(x)-y).^2);%a+b/x拟合误差

e\_2=sum((cfun2(x)-y).^2);%a+b\*ln(t)拟合误差

e\_3=sum((cfun3(x)-y).^2);%a\*exp(b\*t)拟合误差

e\_4=sum((cfun4(x)-y).^2);%1/(a+b\*t)拟合误差

disp([e\_two,e\_three,e\_four,e\_1,e\_2,e\_3,e\_4]);

disp(cfun1);

disp(cfun2);

disp(cfun3);

disp(cfun4);

**结果分析：**

**第一问的结果：**

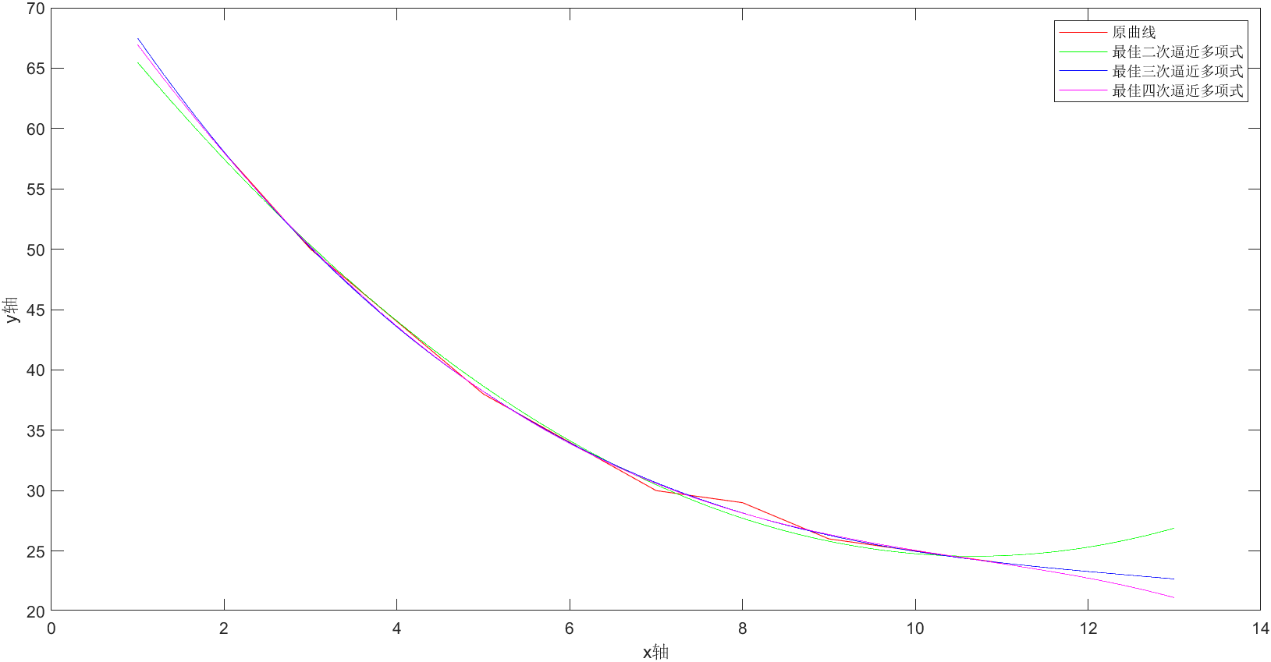
二次、三次、四次的拟合系数分别为:

74.326 -9.311 0.43561

78.542 -11.946 0.894 -0.02350

76.992 -10.613 0.52054 0.018162393160420 -0.001603

画出图形比较：



并计算拟合误差得到二次、三次、四次的误差分别为：

3.216666666666660 1.510256410256403 1.467948717948715

由此可以得到对于此题来说，平方逼近次数越高拟合误差越小。

**第二问的结果：**

四个拟合函数的系数分别为：

General model:

cfun1(t) = a+b/t

Coefficients (with 95% confidence bounds):

a = 18.16 (14.4, 21.92)

b = 87.33 (71.4, 103.3)

General model:

cfun2(t) = a+b\*log(t)

Coefficients (with 95% confidence bounds):

a = 72.14 (69.49, 74.79)

b = -20.76 (-22.21, -19.31)

General model:

cfun3(t) = a\*exp(b\*t)

Coefficients (with 95% confidence bounds):

a = 69.05 (63.27, 74.84)

b = -0.1088 (-0.1241, -0.09352)

General model:

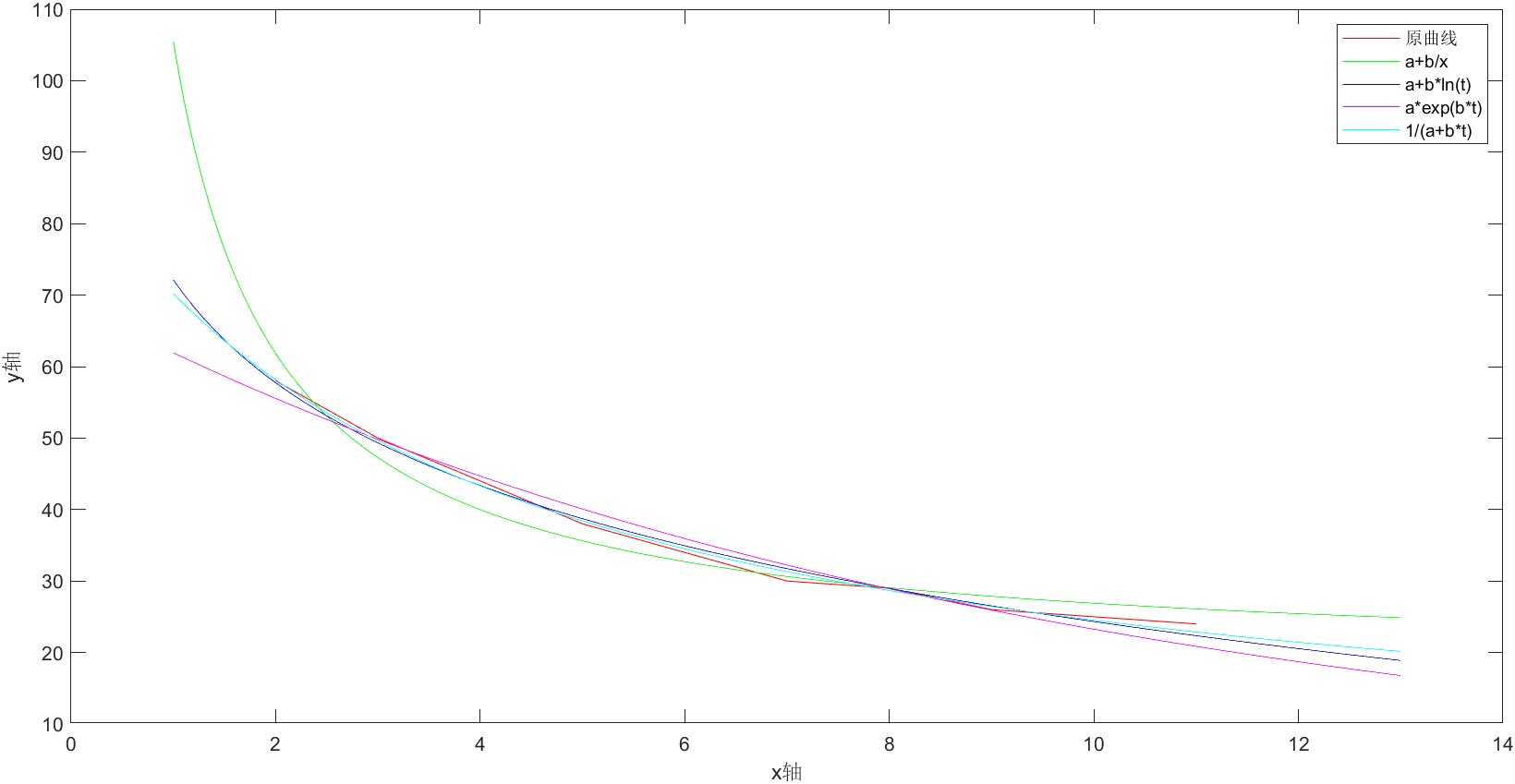
cfun4(t) = 1/(a+b\*t)

Coefficients (with 95% confidence bounds):

a = 0.0113 (0.01062, 0.01197)

b = 0.002948 (0.002784, 0.003112)

画出图形比较：



并计算拟合误差得到题中四种拟合函数的误差依次为：

57.302 8.779 32.511 4.573

由此可以得到对于此题来说，第二种、第四种拟合函数在对应取值的误差更小，即：

**a+b\*ln(x) 和 1/(a+b\*x)**

而第三、第四种误差相对较大，即：

**a+b/x 和 a\*exp(b\*x)**

# 五、结论和感想

这个算例代码量还是相对略大一些，因此一开始选用封装相对较好的MATLAB进行编程，对于第一问来说无非就是矩阵的运算，我们在编程时需要考虑如何使自己编写的代码可移植性更强，定义的函数可实现功能更多；对于第二问来说，我们需要对MATLAB本身的库有一定的掌握，因此需要我们有较强的上网检索的能力，对于不同函数来说，对不同数据的拟合效果也是不同的，可以在拟合之前计算一下其拟合误差再来选用合适的拟合函数。