统计推断在数模转换系统中的应用

组号 07　姓名：邵冰洁 学号：5140309453　　姓名：张珂新 学号：5140309452

摘要：本课程以某产品内部传感器部件的输入输出特性为研究对象，通过采样进行曲线拟合。若对每个样品都进行所有 51 组数据的完整测定，过于耗费时间且成本高。若测量数据过少，又可能导致拟合精度降低。为了在定标成本和拟合精度间找到平衡点，我们对拟合方式和算法都进行了探索。确定采用三次样条插值的方法和遗传算法为该模块的批量生产设计一种成本合理的传感特性校准方案。

关键词：拟合，三次样条插值法， 遗传算法

1 引言

随着现代科技的不断发展，传感器已经得到了广泛地应用，要在保证质量的前提下大批量生产传感器，那么在制作仍少不了校准和定标的步骤。但在实际运用中，我们很难找到具体两个物理变量的实际函数关系。因此，定标的工序为一般是先生产一定批量的电子产品，测得若干组数据，再进行一定的数据处理得到最佳的函数关系，以便达到最终定标目的。[2] 普遍意义上大多数输入输出都呈现非线性关系，通过数学推导并不可靠，因此在定标的时候要进行一定数量的数据拟合。监测的物理量与相关的变量的选取也是影响最终定标校准的一个不可忽略的因素，也有一定的附加成本，减少测定的数量即可以节约大量生产成本。本课题选择测定点的组合方案相当于解一个组合优化问题，但是备选的方案数量巨大，是典型的 NP 难问题。目前的计算机尚难对该问题进行简单穷举（暴力穷举）的方式进行解决。因此本课题中，我们选取其中的遗传算法进行问题求解。

2 传感器校准的相关原理

2.1 传感部件特性[1]

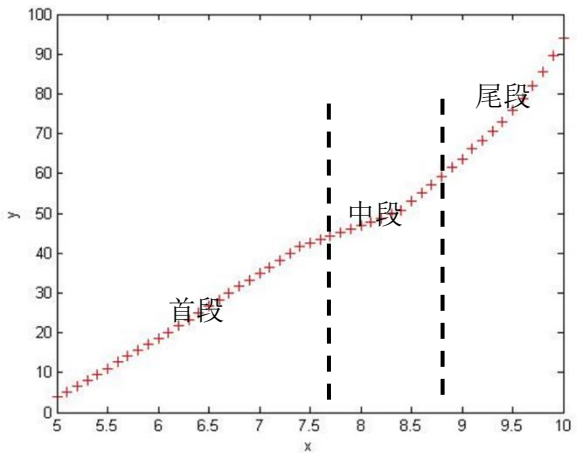


图 2.1 传感特性图示

一个传感部件个体的输入输出特性大致如图 2 所示，有以下主要特征：

- Y 取值随 X 取值的增大而单调递增；
- X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；
- 不同个体的特性曲线形态相似但两两相异；
- 特性曲线按斜率变化大致可以区分为首段、中段、尾段三部分，中段的平均斜率小于首段和尾段；
- 首段、中段、尾段单独都不是完全线性的，且不同个体的弯曲形态有随机性差异；
- 不同个体的中段起点位置、终点位置有随机性差异。

2.2 成本计算[1]

为评估和比较不同的校准方案，特制定以下成本计算规则。

- 单点定标误差成本

$$s_{i,j}=\begin{cases}0 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.4\\ 0.1 & \text{if } 0.4<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.6\\ 0.7 & \text{if } 0.6<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.8\\ 0.9 & \text{if } 0.8<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 1\\ 1.5 & \text{if } 1<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 2\\ 6 & \text{if } 2<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 3\\ 12 & \text{if } 3<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 5\\ 25 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|>5\end{cases} \tag{1}$$

单点定标误差的成本按式（1）计算，其中 $y_{i,j}$ 表示第 i 个样本之第 j 点 Y 的实测值，$\hat{y}_{i,j}$ 表示定标后得到的估测值（读数），该点的相应误差成本以符号 $s_{i,j}$ 记。

- 单点测定成本

实施一次单点测定的成本以符号 q 记。本课题指定 q=12。

- 某一样本个体的定标成本

$$S_i=\sum_{j=1}^{51}s_{i,j}+q\cdot n_i \tag{2}$$

对样本 i 总的定标成本按式（2）计算，式中 n_i 表示对该样本个体定标过程中的单点测定次数。

- 校准方案总成本

按式（3）计算评估校准方案的总成本，即使用该校准方案对标准样本库中每个样本个体逐一定标，取所有样本个体的定标成本的统计平均。

$$C=\frac{1}{M}\sum_{i=1}^{M}S_i \tag{3}$$

总成本较低的校准方案，认定为较优方案。

2.3 三次样条插值法[2]

三次样条插值法对于中间部分的相邻 4个点采用三次曲线拟合，并在中间两点处做出拟合曲线；对于边界部分的 2个点，选取与之靠近的三个点采用二次曲线拟合，并在靠近边界的两点处做出拟合曲线。如此操作，在断点处斜率和曲率都将连续。选取7个相同的点，通过三次样条插值拟合得到一条拟合曲线，对比曲线中的每一个点与实验测的数据，通过评测函数计算其成本。

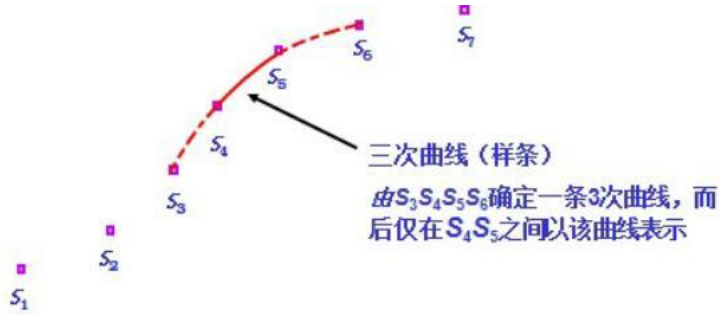


图 2.3 三次样条插值曲线拟合

2.4 遗传算法[2]

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定;具有内在的隐并行性和更好的全局寻优能力;采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。

遗传算法是解决搜索问题的一种通用算法，对于各种通用问题都可以使用。搜索算法的共同特征为：

（1）首先组成一组候选解；

（2）依据某些适应性条件测算这些候选解的适应度；

（3）根据适应度保留某些候选解，放弃其他候选解；

（4）对保留的候选解进行某些操作，生成新的候选解。

在遗传算法中，上述几个特征以一种特殊的方式组合在一起，：基于染色体群的并行搜索，带有猜测性质的选择操作、交换操作和突变操作。这种特殊的组合方式将遗传算法与其它搜索算法区别开来。

3 研究流程

3.1 研究方法框图

每组数据共 51 个点，减少测量点数势必可以直接降低定标成本，但同时，也会导致所

绘出的曲线与实际测量得到的曲线相差较大。所以，我们实验的方向是在减少测量点的基础上，同时保证所拟合出来的曲线与实际曲线很接近。
实验时，假定选取 7 个点来推定所有点分布情况时，定标成本最小。在后续实验对比中，我们还将找出实际情况下，选取几个点成本最优，以及在什么条件下能更好地降低成本。
研究方法框图如下：

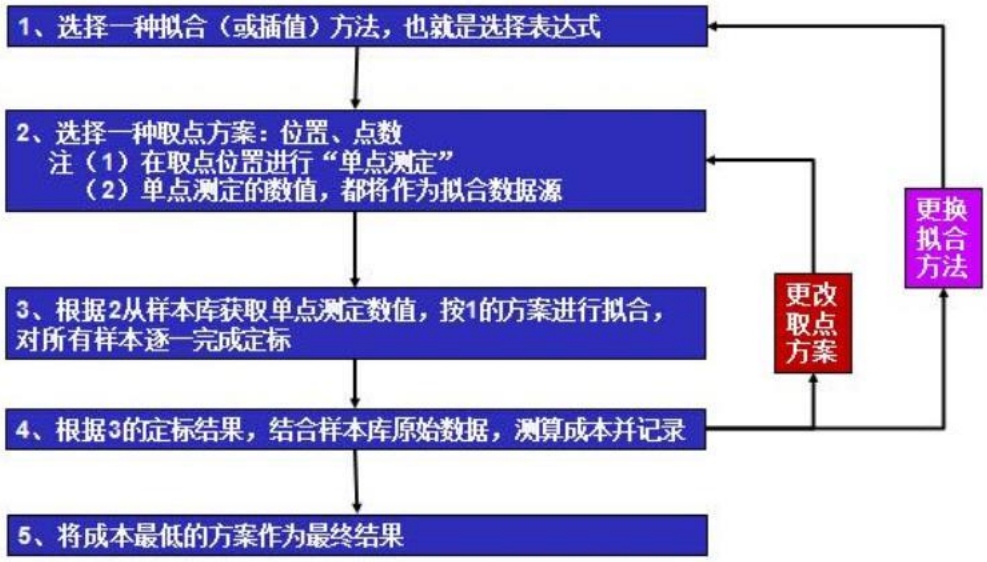


图 3.1 研究方法框图[1]

3.2 对样本数据的观察

选择样本 1,118,236,352 进行描点绘图，得到图 3.2(a)，图 3.2(b)，图 3.2(c)，图 3.2(d)

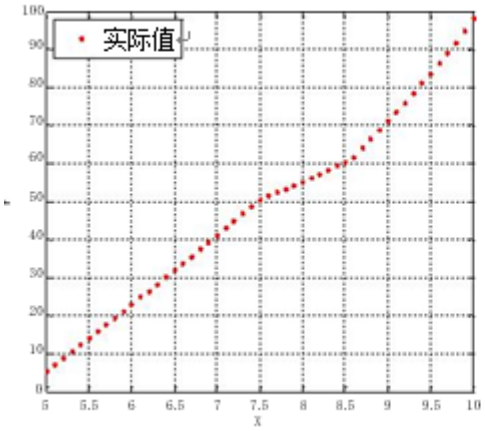


图 3.2(a) 样本 1 的 Y-X 图

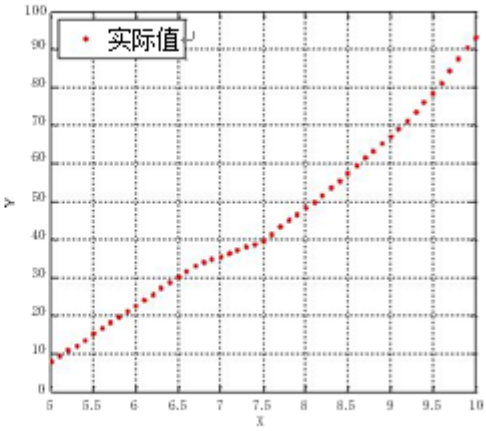
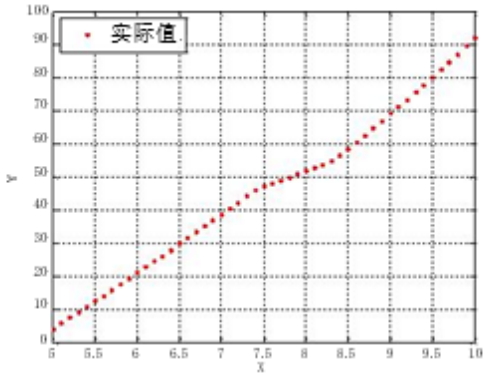


图 3.2(b) 样本 118 的 Y-X 图



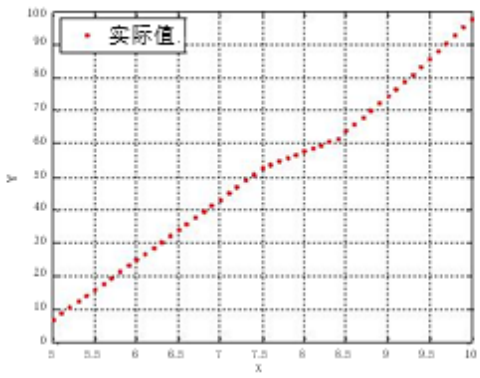


图 3.2(c)　样本 236 的 Y-X 图　　图 3.2(d) 样本 352 的 Y-X 图

通过对以上 4 幅图进行观察，发现，曲线明显有两个拐点，所以宜采用三次多项式或者三次样条的方法进行拟合。

4 结果讨论

4.1 初步运行结果

设定的目标成本为 95，程序运行 54 代后，得到结果，对应的最低成本的 7 个点为【　2　8　20　26　33　43　49】。

4.2 最终结果

4.2.1 三次样条

由于取点的多少影响到三次样条插值的精确性以及和源数据的拟合程度，即是取得点数越多，与原数据拟合程度越好，从而误差成本越小，但是取点越多会造成测定成本的相应提高，所以在误差成本变化不大的情况下要考虑到越少取点测定成本越小，这样才能有效降低总成本。故我们决定改变基因条数即测试点个数来重复进行实验。进一步确定在改变基因条数之后最小平均成本的变化以及最佳取点个数。下表是改变取点个数的实验结果。

表1不同取点个数及成本

取点个数	最佳取点方案	测定成本	误差成本	最小成本
6	3　10　22　31　43　50	72	21.680	93.680
7	3　8　20　26　35　44　50	84	10.985	94.9850
8	2　10　18　26　33　36　45　50	96	7.3255	103.3255

由表格中可以看出当取点个数发生变化时，测定成本逐渐攀升，而误差成本则降，他们的总和最小平均成本出现一个向下再向上的性态。可见，当取点个数为 6 测定成本 72 的时候最小平均成本最低可达到 93.680。

4.2.2 三项式拟合

三次项使用 （%p=polyfit(temp_x,temp_y,3); %f=polyval(p,Xdata(j,:)))

6 个样本点，最小成本：96.7275，选取点： 2 12 16 29 35 49

7 个样本点，最小成本：95.3113，选取点：1 13 22 24 29 38 49

4.2.3 统计分析

对所得方案进行统计分析，为了研究最终方案的效果，分别测量每个样本的误差成本，绘图如下，反映了普遍效果。

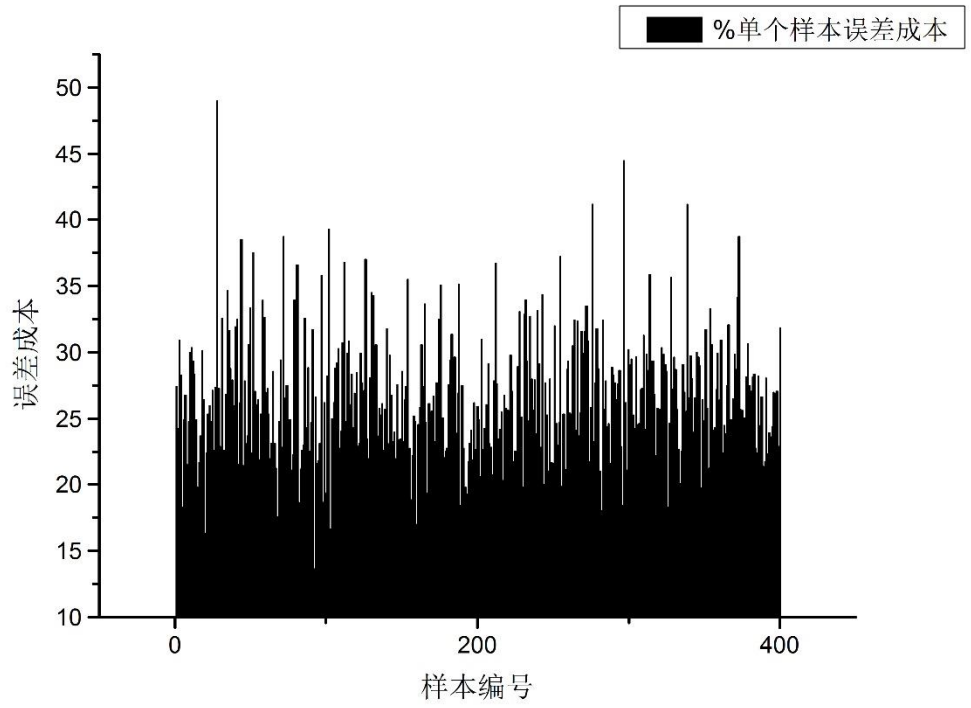


图 4.2 误差成本图

从上图可以看出来，单个样本的误差成本大致分布比较均匀，极个别的点误差成本过大。

本次实验，我们以遗传算法为主，配合三次样条插值和三次多项式拟合，对如何降低测定成本进行了探讨。最终得到以下结论：

1、对取点数不同的情况，进行多次实验后，发现取六个点时成本较低，得到的最优解为以下六点：3 10 22 31 43 50。也就是说，在以后的测试过程中，只需选这些点为测试点，利用三次样条插值，即可较为准确的估计其他点的数据。

2、三次样条插值与三次多项式拟合相比，三次样条插值拟合结果更优。因此在测试时应选择三次样条插值拟合。而且与三次多项式拟合相比，三次样条插值能够提高拟合的精确度。在一些需要控制误差成本的工程中，三次样条插值拟合方法更加优越。

3、本次实验我们加强对于 matlab 的掌握，了解了启发式搜索方法原理及应用，同时结合实际应用进行了一些改进。在这些过程中收获良多。

5 参考文献

[1]上海交通大学电子工程系.统计推断在数模转换中的应用课程讲义，ftp://202.120.39.248/

[2] 百度百科。网址：http://baike.baidu.com/

6【附录】遗传算法代码

%main 函数，使用三次样条插值的遗传算法

```
global pop;              %取观察点种群
global fitnessTable;     %成本列表
global DATAX;          %样本电压 X 列表
global DATAY;           %样本物理量 Y 列表
global len;            %样本组数量

popSize=100;             %种群大小
codeSize=6;              %观察点数量
crossRate=0.8;           %交叉概率
mutateRate=0.02;          %变异概率
maxGeneration=1000;      %最大子代数
aimScore=100;            %目标成本
minScore=200;              %最低成本

rand('state',sum(100*clock));

readData();
init(popSize,codeSize,51);
fitnessTable=zeros(popSize,1);
for i=1:popSize
    fitnessTable(i)=fitness(pop(i,:));
end
currMin=min(fitnessTable) %currMin 表示当前种群中的最低成本

if currMin<minScore
    minScore=currMin;%minScore 表示目前为止所有运行过的种群出现过的最低成本
    minScoreLocation=find(fitnessTable==currMin);
    minScoreMethod=pop(minScoreLocation,: %minScoreMethod 表示上述最低成本对应的
选取点
end
minScore
minScoreMethod
if minScore<aimScore
    return;
end
G=1;%G 表示当前种群的子代数
while G<=maxGeneration
```

```
    selection(popSize,codeSize);
    cross(popSize,codeSize,crossRate);
    mutation(popSize,codeSize,mutateRate);
    for i=1:popSize
        fitnessTable(i)=fitness(pop(i,:));
    end
    currMin=min(fitnessTable)

    if currMin<minScore
        minScore=currMin;
        minScoreLocation=find(fitnessTable==currMin);
        minScoreMethod=pop(minScoreLocation,:);
    end
    minScore
    minScoreMethod
    if minScore<aimScore
        break;
    end
    G
    G=G+1;
end
%init 函数
function init(popSize,codeSize,maxNum)
global pop;
poparr=zeros(popSize,codeSize);
for i=1:popSize
    for j=1:codeSize
        poparr(i,j) = randi(maxNum);
        k=1;
        while (k<=j-1)
            if (poparr(i,j) == poparr(i,k))
                poparr(i,j) = randi(maxNum);
                k=0;
            end
            k=k+1;
        end
    end
end
pop=sort(poparr,2);
end
%负责读入数据的 readData 函数
function readData()
origin=xlsread('20150915dataform.csv');
global DATAX;
```

```
global DATAY;
global len;
len=length(origin)/2;
DATAX=origin(1:2:(len)*2-1,:);
DATAY=origin(2:2:(len)*2,:);
End
%fitness 函数，计算适应度
function [score] = fitness(method)
global DATAX;
global DATAY;
global len;                    %样本组容量
codeSize=length(method);     %codeSize=7
sumN=0;
X=zeros(1,codeSize);
Y=zeros(1,codeSize);
for i=1:len
    n=0;
    for j=1:codeSize
        X(j)=DATAX(i,method(j));
        Y(j)=DATAY(i,method(j));
    end
    Y1=interp1(X,Y,DATAX(i,:),'spline');
    yTable=abs(Y1-DATAY(i,:));
    for j=1:51
        dif=yTable(j);
        if dif<=0.5
            n=n+0;
        elseif dif<=1
            n=n+0.5;
        elseif dif<=2
            n=n+1.5;
        elseif dif<=3
            n=n+6;
        elseif dif<=5
            n=n+12;
        elseif dif>5
            n=n+25;
        end
    end
    sumN=sumN+n;
end
score=sumN/len+72;
end
%selection 函数，选出优胜的后代
```

```
function [] = selection(popSize,codeSize)
global pop;
global fitnessTable;
fitnessSum=zeros(popSize,1);
fitnessSum(1)=1/fitnessTable(1);
for i=2:popSize
    fitnessSum(i)=fitnessSum(i-1)+1/fitnessTable(i);
end
popNew=zeros(popSize,codeSize);
for i=1:popSize
    r=rand()*fitnessSum(popSize);
    left=1;
    right=popSize;
    mid=round((left+right)/2);
    while 1
        if r>fitnessSum(mid)
            left=mid;
        else
            if r<fitnessSum(mid)
                right=mid;
            else
                popNew(i,:)=pop(mid,:);
                break;
            end
        end
        mid=round((left+right)/2);
        if (mid==left)||(mid==right)
            popNew(i,:)=pop(right,:);
            break;
        end
    end
end
pop=popNew;
%mutation 函数，模拟变异
function mutation(popSize,codeSize,mutateRate)
global pop;
for i=1:popSize
    r=rand();
    if r<mutateRate
        r=randi(codeSize);
        if rand()>0.5
            x=1;
        else
            x=-1;
```

```
        end
        p=pop(i,r)+x;
        if (r==1)
            if (p>=1)&&(pop(i,r+1)~=p)
                pop(i,r)=p;
            end
        elseif (r==codeSize)
            if (p<=51)&&(pop(i,r-1)~=p)
                pop(i,r)=p;
            end
        else
            if (pop(i,r-1)~=p)&&(pop(i,r+1)~=p)
                pop(i,r)=p;
            end
        end
        %checkPop(i,codeSize);
    end
end
end
%cross 函数，模拟交叉
function [] = cross(popSize,codeSize,crossRate)
global pop;
for i=1:2:popSize
    r=rand;
    if r>crossRate
        continue;
    end
    m=randi([2,codeSize]);
    p=pop(i,m:codeSize);
    pop(i,m:codeSize)=pop(i+1,m:codeSize);
    pop(i+1,m:codeSize)=p;
    checkPop(i,codeSize);
    checkPop(i+1,codeSize);
end
end
%checkPop 函数
function checkPop(n,codeSize)
global pop;
pop(n,:)=sort(pop(n,:));
flag=0;
for j=2:codeSize
    if pop(n,j-1)==pop(n,j)
        pop(n,j)=randi(51);
        flag=1;
```

```
    end
end
if flag
    checkPop(n,codeSize);
end
end
%提取400个样本的成本
global DATAX;         %样本电压X列表
global DATAY;          %样本物理量Y列表
global len;           %样本组数量


codeSize=6;              %观察点数量


rand('state',sum(100*clock));

readData();
%init(popSize,codeSize,51);
%fitnessTable=zeros(popSize,1);
X=zeros(1,codeSize);
Y=zeros(1,codeSize);
yy=zeros(1,51);
method=[3 10 22 31 43 50];
for i=1:len
    n=0;
    for j=1:codeSize
        X(j)=DATAX(i,method(j));
        Y(j)=DATAY(i,method(j));
    end
    Y1=interp1(X,Y,DATAX(i,:),'spline');
    yTable=abs(Y1-DATAY(i,:));
    for j=1:51
        dif=yTable(j);
        n=n+dif;
    end
    yy(i)=n;
end
fid = fopen('sdata.txt','wt');
fprintf(fid,'%g\t',yy);
fclose(fid);
```