统计推断在数模转换中的应用

——寻求校准工序的优化方案

35 组　卫琛尧 5130309230，刘承天 5130309231

摘要：本论文在一系列实验结果数据的基础上，探究统计推断在数模、模数转换中的应用。报告中利用 469 个观测样本的 51 个观测点，采用选取特征点，运用随机函数和遗传算法，不断寻求使总成本最小的最优函数。

关键词：统计推断，曲线拟合，三次样条及埃尔米特插值，遗传算法

Application of Statistic Model in A/D and D/A Converse System

ABSTRACT: The paper is to explore the application of statistic model in model in module-digital-analog conversion system on basis of a series of experimental data. The report uses 469 samples and 51 sets of data to pick characteristic points to find a consensus function which makes the total cost the least by means of random selection and genetic algorithm.

Key words: Statistic Model, Curve Fitting, Spine and Pchip Interpolation, Genetic Algorithm

1 引言

在工程实践与科学实验中经常会遇到一些由多个变量组成的系统，当需要知道这些变量两两之间的关系时，观测点个数的选择，观测点的选择，拟合函数的选择往往对成本有着极大的影响。通常采取的方法是通过多种不同曲线拟合，算出多种不同的方案，并根据单位总成本的大小来选出最优的方案

1.1 问题的提出

本次课题是为某型产品内部的一个监测模块，寻求校准工序的最优方案。

2 寻找最优解的方案设计

2.1 分段取点算法

获得全局最优解最显而易见的方案是将所有可能的性逐一进行计算，也就是一共要计算 C_{51}^{7}=115775100 次，假设每次运算用时为 0.01 秒，由于有 469 个样本，算出总成本需要 6284.55 天，笔记本显然是无法做到的，因此分段取点不失为一种较暴力算法来说较为优化的算法。

表 2-1 观测点选取范围

观测点序号	选取区间
1	[1,3]
2	[9,11]
3	[17,19]
4	[25,27]
5	[33,35]
6	[41,43]
7	[49,51]

利用编程来找到成本最小的解，从而大大减少了运算时间，提高了运算的效率。然而分

段取点无法得出成本的最小值，在随后的实验中，我们放弃了这种算法，全部使用遗传算法进行计算。

2.2 遗传算法

2.2.1 遗传算法介绍

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法是从代表问题可能潜在的解集的一个种群开始的，而一个种群则由经过基因编码的一定数目的个体组成。每个个体实际上是染色体带有特征的实体。染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现是某种基因组合，它决定了个体的形状和外部表现。

遗传算法的主要步骤包括：选择（selection），选择父代中优秀的个体直接传递给下一代，即保留父代。交叉（crossover），以两个父代的基因为基础，进行交叉重组，生成下一代，好的基因会有更大的概率传给子辈。变异（mutation），每个父代个体都有一定的概率发生随机的基因突变。 评价基因好坏的依据是适应度。在遗传算法的步骤中起着突出作用的其实是变异。正因为有了变异，算法才存在在全局范围内寻找到最优解的可能性。但是遗传算法需要通过反复试验，才有可能给出最优解。

遗传算法的步骤如图 2-1 所示。

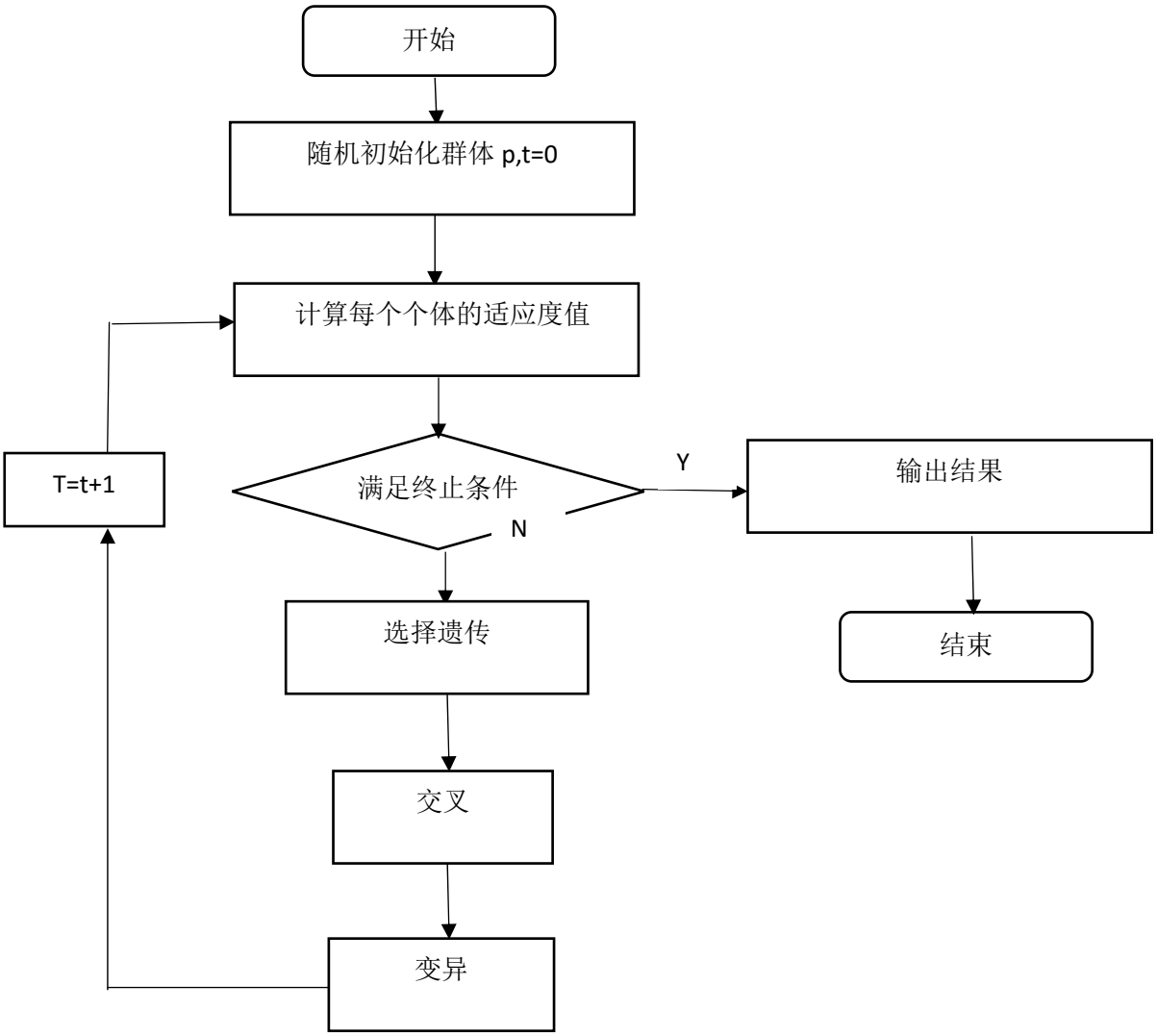

图 2-1 遗传算法流程图

常用的编码方式有二进制编码、实数编码、矩阵编码、树形编码等等，在这个实验中，我们选择十进制编码。

2.2.2 参数设置

popsize=100(种群数量，构成繁殖池)

pc=0.8（交叉的概率）

pm=0.08（变异的概率）

cut条件为：个体得分为种群内个体最小得分的1.3倍。若cut<30，则cut=30

loopnum=100（遗传代数）

3 拟合曲线方案设计

3.1 概述

无论是分段取点法、遗传算法还是模拟退火算法，其目的都是寻找某种拟合方式下对成本计算的最优解，因此对于某种给定的拟合方式，不同的算法在时间复杂度上会有所差异，但计算出的成本都会无限接近该拟合方式的最低成本。从而，选择对于实验数据最佳的拟合方案，而不是寻找最优解的算法（GA、SA），才是得出最低成本的关键。对于该实验样本，我们分析了多项式拟合、分段三次埃尔米特插值拟合与三次样条插值拟合。

3.2 多项式拟合法

低次多项式拟合的优点在于拟合速度较快，运算简单，原理更加简洁。

其明显缺点在于由少数点确定的多项式在整体的适应度上表现不佳，曲线与实际数据偏差较大,且高次拟合速度较慢。

3.3 分段三次 Hermite 插值拟合法

许多实际插值问题中，为使插值函数能更好地和原来的函数重合，不但要求二者在节点上函数值相等，而且还要求相切，对应的导数值也相等，甚至要求高阶导数也相等。这类插值称作切触插值，或埃尔米特(Hermite)插值。满足这种要求的插值多项式就是埃尔米特插值多项式。Matlab 中直接提供了 interp1(x_premea,y0_premea,x,'pchip')命令可以进行分段三次 Hermite 插值拟合。

3.4 三次样条插值拟合法

三次样条插值是通过一系列形值点的一条光滑曲线,数学上通过求解三弯矩方程组得出曲线函数组的过程。

三次样条函数：

定义:函数 $S(x)\in C^2[a,b]$，且在每个小区间$[x_j, x_{j+1}]$上是三次多项式，其中 $a<x_0<x_1<\cdots<x_n=b$ 是给定节点，则称 S(x)是节点 x_0，x_1，$x_2\cdots x_n$上的三次样条函数。

若在节点 x_j上给定函数值 $y_j=f(x_j)(j=0,1,\cdots,n)$，并成立 $S(x_j)=y_j(j=0,1,\cdots,n)$，则称 S(x)为三次样条插值函数。

Matlab 中直接提供了 interp1(x_premea,y0_premea,x,'spline')命令可以进行三次样条插值拟合。

3.5 pchip 与 spline 拟合的对比

查阅 matlab 中的 pchip.m 可知：

spine 提供的函数 s(x)的构建方法和 pchip 里面的函数 p(x)完全相同，只不过在 X(j)处的斜率的选择方法不一样，

spline 函数的 s(x)在 X(j)的二阶导数 D^2s(x)也是连续的，这导致了如下结果：

spline 更加光滑，也就是说，D^2s(x)是连续的。

如果数据是一个光滑函数的值，则 spline 更加精确。

如果数据不是光滑的，则 pchip 没有 overshoots，也不太震荡（less oscillation）。

样条比 pchip 光滑，样条的两阶导数连续，而 pchip 一阶导数连续。不连续的两阶导数隐含着不连续的曲率。人的眼睛可以检测出图形上曲率的不连续。另一方面，pchip 是保形状的，而样条不一定保形状。

考虑到实验曲线的形状，我们认为 pchip 的拟合结果优于 spline 的拟合结果。

对于本次实验数据，其拟合差异直观地如图 3-3 所示。

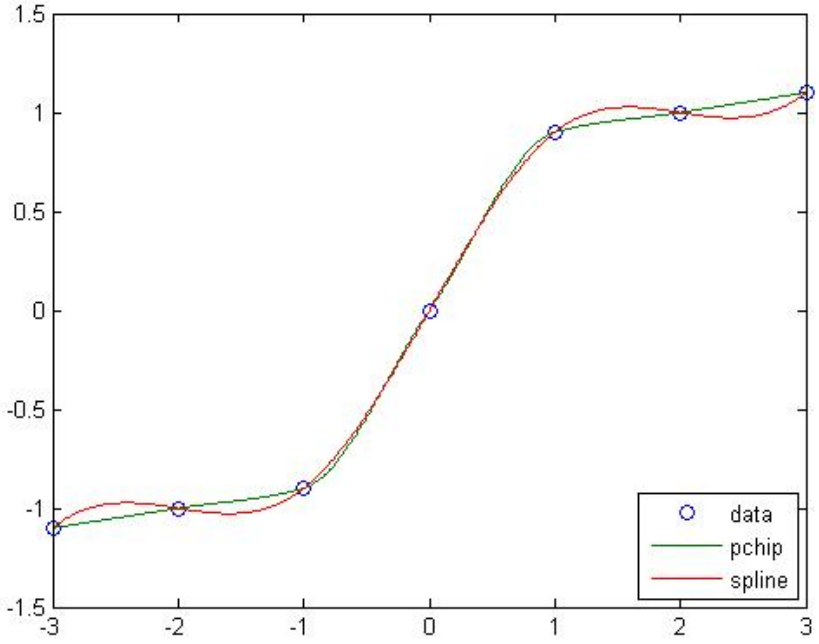

图 3-3 pchip 与 spline 拟合的对比图

4 结论

4.1 多项式拟合结果

在多项式拟合次数选取中，我们取出 7 个点对所有样本分别进行不同次数的拟合。

表 3-1 多项式拟合所得结果

拟合次数	最优解	最低定标成本
2	6 19 25 31 40 45 46	257.0235
3	3 10 19 27 35 43 49	132.0800
4	4 13 23 31 38 46 50	116.5107
5	3 7 17 26 34 43 50	107.3881
6	3 8 13 27 40 44 51	117.3230

5 次方拟合的结果最优。

4.2 spline 拟合结果

点集为： 3 9 18 24 32 43 50

最小值： 95.4648

4.3 pchip 拟合结果

点集为： 3 10 18 25 31 39 50

最小值为: 92.3454

5 拓展

5.1 用 pchip 探究更少的拟合点

5.1.1 6 点拟合与 5 点拟合

6 点拟合结果

点集为: 4 13 21 28 37 48

最小值为: 86.5320

5 点拟合结果

点集为: 4 16 28 37 49

最小值为: 85.1898

5.1.1 讨论

这个结果，一方面可以说明，pchip 在拐点的拟合的确比较符合曲线特性,在 5 个点时的拟合远远优于 spline 的拟合结果。另一方面，通过分析每个样本每一个点的误差，发现依然有不小的差异。因此从实际应用角度考虑，取点的成本仍然应该降低，误差的成本应该更加大一些。

6 参考文献

[1] 张至涌.精通 Matlab R2011a[M] 北京:北京航空航天大学出版社,2011 年 11 月

[2] 上海交大电子工程系. 统计推断在数模转换系统中的应用课程讲义 [EB/OL].ftp://202.120.39.248.

7 附录

6.1 师生交流部分

在师生交流那天，对于遗传算法以及模拟退火算法（振荡某点观察成本，一定取成本更小的点，一定概率取成本变大的点）的步骤事实上已经了解的非常清楚了，然而一直有个 bug 没找到，回去以后找了很久才发现是在 for 的后面把冒号打成分号了，这个错误十分隐蔽，而且结果程序居然还可以运行。所以对师生交流那天，因为没有输出结果导致无法充分交流感到十分遗憾。

不过当时虽然已经理解了遗传算法，但是拟合依然用的是 polyfit，对于三次样条插值拟合不是十分了解，成本远远大于 100。后来在老师的鼓励下查阅了相关资料和 interp1 内各种拟合的数学背景，基本弄懂了 interp1 里 6 种拟合的功能。

6.2 spline 与 pchip 的比较代码

```
x = -3:3;
y = [-1.1 -1 -0.9 0 0.9 1 1.1];
t = -3:.01:3;
plot(x,y,'o',t,[pchip(x,y,t); spline(x,y,t)])
legend('data','pchip','spline',4)
```

6.3 分段取点法

```
function p = algorithm2( )

p=zeros(2,7);
p(1,1)=10000;
x1=zeros(1,7);
y1=zeros(1,7);
y3=zeros(1,51);
grade=0;y=0;
x=csvread('d:\\20141010dataform.csv');

for d1=1:1:3
    for d2=9:1:11
        for d3=17:1:19
            for d4=25:1:27
                for d5=33:1:35
                    for d6=41:1:43
                        for d7=49:1:51
                            for k=1:2:937
                                x1=[x(k,d1) x(k,d2) x(k,d3) x(k,d4) x(k,d5) x(k,d6) x(k,d7)];
                                y1=[x(k+1,d1)  x(k+1,d2)  x(k+1,d3)  x(k+1,d4)  x(k+1,d5)
x(k+1,d6) x(k+1,d7)];
                                y3=mycurvefitting(x1,y1);


                                for n=1:1:51
                                    if (abs(y3(n)-x(k+1,n)))<=0.5
                                        grade=grade;
                                    elseif (abs(y3(n)-x(k+1,n)))<=1
                                        grade=grade+0.5;
                                    elseif (abs(y3(n)-x(k+1,n)))<=2
                                        grade=grade+1.5;
                                    elseif (abs(y3(n)-x(k+1,n)))<=3
                                        grade=grade+6;
                                    elseif (abs(y3(n)-x(k+1,n)))<=5
                                        grade=grade+12;
                                    else grade=grade+25;
                                    end
                                end
                                y=y+grade+12*7;
                                grade=0;
                            end
                            y=y/469;                %返回C=(∑Si)/
                            if p(1,1)>y
                                p(1,1)=y;
```

```
                                    p(2,:)=[d1,d2,d3,d4,d5,d6,d7];
                                end
                                y=0;
                            end
                        end
                    end
                end
            end
        end
    end
end
disp(y);
disp([d1,d2,d3,d4,d5,d6,d7]);
end
```

6.3 遗传算法

```
function [] = testOfMine( )
data = csvread('d:\20141010dataform.csv');
loopnum=100; %% 遗传代数，交叉变异次数,可改，范围30-100。
points=7;
tic
final=mainfun(loopnum,data,points);
disp('points at: ');
disp(final(1,1:7));
disp(['with score ' num2str(final(8))]);
toc
end


function [finalpop] = mainfun(loopnum,data,points)
popsize=100;  %% 每次繁殖交叉的个体数，构成繁殖池，必为偶数
pc=0.8;   %% 交叉的概率，可改，在0.6-0.9之间。
pm=0.08;  %% 变异的概率，可改，在0.01-0.1之间，较小。
pop=creationfun(popsize,points);
bestfit=10000;
bestnum=1;
fitvalue=zeros(1,popsize);  %存每个个体的得分
finalpop=ones(1,points+1);
record=zeros(1,loopnum);

for i=1:1:loopnum
    for j=1:popsize %检查是否有重复点
        for k=1:points
            pop(j,:)=check(k,pop(j,:),points);
        end
```

```
    end
    pop=sort(pop')';
%     disp(pop);
    for j=1:popsize
        fitvalue(j)=fitnessfun(pop(j,:),data,points);    %输入j个体得分
    end
    record(i)=max(fitvalue);
    k=find(fitvalue==min(fitvalue));
%     disp('pop is');disp(pop);
    disp('代数为：');disp(i);
    disp('点集为');disp(pop(k(1),:));
    disp('最小值为:');disp(min(fitvalue));
    disp('平均值为:');disp(mean(fitvalue));
    disp('最大值为:');disp(max(fitvalue));

    [pop,cut]=selectfun(pop,fitvalue,popsize);  %对pop的个体按得分升序排
列
%     disp('seletion ended,pop is');
%     disp(pop);

    %重新打乱
    for mess=1:popsize
        temp1=ceil(rand*popsize);
        temp2=ceil(rand*popsize);
        pop(temp1,:)=pop(temp2,:);
        pop(temp2,:)=pop(temp1,:);
    end
%     disp(pop);
%     disp('addpop ended');
%     pause;
    pop=crossfun(pc,pop,data,points);
    pop=mutationfun(pm,pop,loopnum,i,points);
    for j=1:popsize %检查是否有重复点
        for k=1:points
            pop(j,:)=check(k,pop(j,:),points);
        end
    end
end
disp('end1');
for k=1:popsize
    fitvalue(k)=fitnessfun(pop(k,1:points),data,points);
    if bestfit<fitvalue(k)
        bestfit=fitvalue(k);
        bestnum=k;
```

```
    end
end
finalpop(1,1:points)=pop(bestnum,1:points);
finalpop(points+1)=max(fitvalue);
end


function pop = creationfun(popsize,points)
pop=ones(popsize,points);
for i=1:popsize
    for j=1:points
        pop(i,j)=ceil(rand*51);
    end
    for k=1:points
        pop(i,:)=check(k,pop(i,:),points);
    end
end
% disp('creation ended');
% pause;
end


function y=fitnessfun(x,data,points)  %x是pop
row=size(data,1);  %row为data的行数
x1=zeros(1,points);  %取点x的位置
y1=zeros(1,points);  %y值的大小
y3=zeros(1,51);  %存拟合取点的51个点的Y值
grade=0;  %分数
y=0;  %%总分

for k=1:2:row    %对每行X
    for i=1:points
        x1(i)=data(k,x(i)); %取出pop中的7个点
        y1(i)=data(k+1,x(i)); %取出pop中7个点对应的Y
    end
    numberOfX=[5.0:0.1:10.0];
    y3= mycurvefitting( x_premea,y0_premea )
    for n=1:1:51
        if (abs(y3(n)-data(k+1,n)))<=0.5
            grade=grade;
        elseif (abs(y3(n)-data(k+1,n)))<=1
            grade=grade+0.5;
        elseif (abs(y3(n)-data(k+1,n)))<=2
            grade=grade+1.5;
```

```
        elseif (abs(y3(n)-data(k+1,n)))<=3
            grade=grade+6;
        elseif (abs(y3(n)-data(k+1,n)))<=5
            grade=grade+12;
        else grade=grade+25;
        end
    end
    y=y+grade+12*points;
    grade=0;
end
y=y/(row/2); %返回C=(ΣSi)/M
end


function  [newpop,cut]= selectfun(pop,fitvalue,popsize)  %% 选出池中优秀个体。
[px,py]=size(pop);
newpop=ones(px,py);
[value,index]=sort(fitvalue);   %对fitvalue排序，value是排序后的值，index是它们原先所在的位置
for i=1:popsize
    newpop(i,:)=pop(index(i),:);%% 按升序排个体。
end
cut=0;  %% 去掉的后代数。
for i=1:popsize
    if ((value(i)-value(1))/value(1))>0.3
        break;
    end
end
cut=popsize-i+1;
if cut<30
    cut=30;
end
% disp('cut=');
% disp(cut);
total=0;
for i=1:1:(popsize-cut)
    total=total+1/value(i);
end
% disp('total=');
% disp(total);
posibility=ones(1,popsize-cut);
for i=1:1:(popsize-cut)
    posibility(i)=(1/value(i))/total;
```

```
end
% disp('posibility=');
% disp(posibility);

for s=1:cut
    temp=rand;
    for i=1:1:(popsize-cut)
        temp=temp-posibility(i);
        if temp<0
            break
        end
    end
    newpop(popsize-cut+s,:)=newpop(i,:);
    %disp('i=');
    %disp(i);
end
%disp('selection ended');
end


function [ newpop ] = mutationfun(pm,pop,loop,k,points) %变异
px=size(pop,1);
newpop=ones(size(pop));
for i=1:px
    newpop(i,:)=pop(i,:);
    if(rand<pm)
        mpoint=ceil(rand*points);     %变异点的位置
        newpop(i,:)=mutatep(mpoint,pop(i,:),loop,k);
    end
    newpop(i,:)=sort(newpop(i,:));
end
% disp('mutation ended');
end


function [ newrow ] = mutatep(a,row,loop,k)
temp=0;
while(temp==0) %%防止有相同的点序号
    temp=row(a);
    while(temp==row(a))
        temp=round(row(a)+(1-2*rand)*51*(1-k/1.07/loop));  %% 随着繁殖次
数增加，变异变化量减小。
        if(temp>51)
            temp=temp-51;
```

```
        elseif(temp<1)
            temp=temp+51;
        end
    end
    row(a)=temp;
end
newrow=row;
end
function [newpop] = crossfun(pc,pop,data,points)  %% 个体之间进行交叉。
[px,py]=size(pop);
newpop=ones(px,py);
for i=1:2:(px-1)
    if(rand<pc)  %% 以pc概率交叉。
        cpoint=ceil(rand*points);  %产生介于1-7的随机交叉点
temp2=fitnessfun(pop(i,1:points),data,points)+fitnessfun(pop(i+1,1:po
ints),data,points);
        newpop(i,:)=[pop(i,1:cpoint),pop(i+1,cpoint+1:py)];     %交换基
因
        newpop(i+1,:)=[pop(i+1,1:cpoint),pop(i,cpoint+1:py)];
        for j=i:i+1 %检查是否有重复点
            for k=1:points
                newpop(i,:)=check(k,pop(i,:),points);
                newpop(i+1,:)=check(k,pop(i+1,:),points);
            end
        end
temp1=fitnessfun(newpop(i,1:points),data,points)+fitnessfun(newpop(i+
1,1:points),data,points);
        if(temp1<temp2)%% 若变差了，不接受。
            newpop(i,:)=pop(i,:);  %% 赋原值。
            newpop(i+1,:)=pop(i+1,:);   %下一个个体
        end
    else
        newpop(i,:)=pop(i,:);
        newpop(i+1,:)=pop(i+1,:);
    end
end
% disp('cross ended');
% pause;
end

function [newrow]=check(a,row,points) %%防止7点中有重复的.
temp=row(a);%%把row行的第a个值赋给temp
for i=1:points
    if ((a~=i)&(row(i)==temp))
```

```
        temp=ceil(rand*50+1);
        i=1;
        row(a)=temp;
    end
end
newrow=sort(row);
end


function y1 = mycurvefitting( x_premea,y0_premea )
x=[5.0:0.1:10.0];
% 将你的定标计算方法写成指令代码
%     f=polyfit(x_premea,y0_premea,4);
%     y1=polyval(f,5:0.1:10);
y1=interp1(x_premea,y0_premea,x,'pchip');
end
```