统计推断在数模转换系统中的应用

第 13 组　　李博 5140309112　王天逸 5140309108

摘要：本文结合统计方法，根据已知的少量数据实现对其余更多的未知数据的推断。文中分析了不同的拟合方法及其比较，选取了一种较优的拟合方式，并采用遗传算法，模拟一定种群大小的生物遗传进化方式。通过产生新解，不断比较优化，最终得到比较优化的解。
关键词：统计推断、插值拟合、遗传算法，matlab

Application of Statistical Inference in AD&DA Inverting System

Group number:13

ABSTRACT:

This report sets up a mathematical model is combined with statistical methods to infer the large amount of unknown data based on a small amount of known data. This report analysis different fitting methods to select an optimum fit and uses Genetic Algorithm to simulate the process of solid cooling. By constantly generating new solutions and optimizing them, we ultimately get more optimal solution.

Key words:

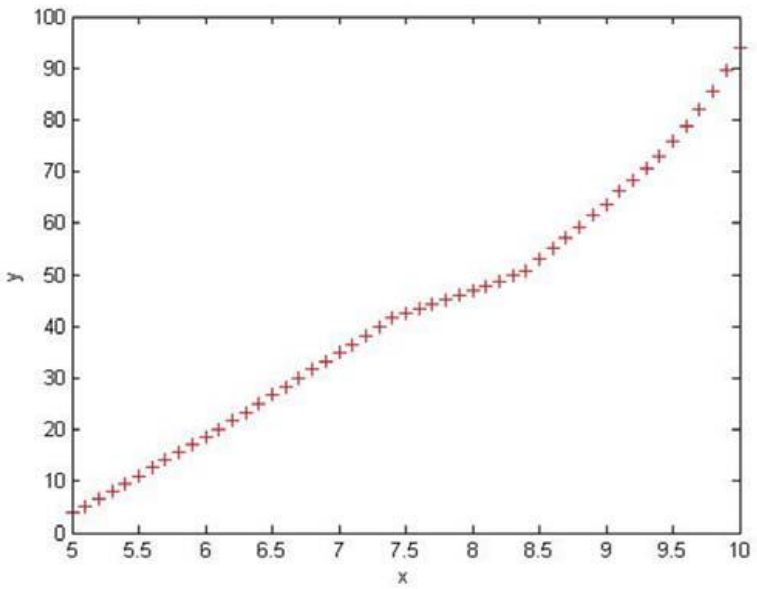
statistic Interference、interpolating fit、Genetic Algorithm

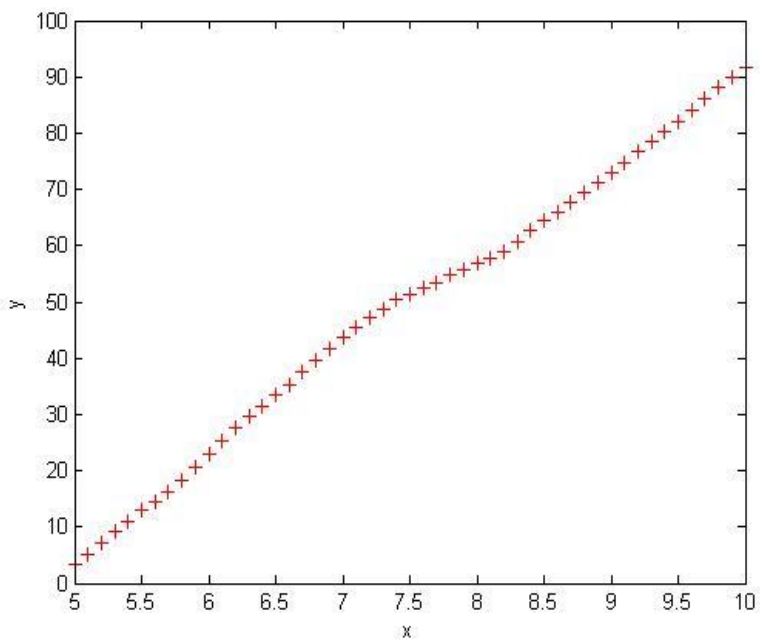
1. 引言

在工程实践中往往会设计许多的测量工具，这些工具主要由传感器和电子信号接收器组成，为了对这些批量生产的工具用一个科学而合理的方法来定标，使其在应用时能够达到一定的精度，但是由于工具的测量数据往往十分多，单靠人工一个一个的进行定标虽然会具有相当高的精度，但是在非精密仪器的情况下，人们希望能够找到一个省时省力的方法来对这些仪器进行定标，让其达到事前给的预期测量精度。显然对于每一单个“工具”都需要根据这个工具产生的某些数据点的反馈进行合理的模拟得出此工具的测量输入-输出关系式，在本文中对曲线插值进行讨论，本文中采用常见的启发式搜索——遗传算法，对数据进行筛选。

2. 课题要求

已获得 400 个样品的测定数据（标准样本库），下面是 2 个样品的数据的绘图实例





1. X-Y 特性曲线是单调递增的。

2. X 取值在 5 – 10 之间；Y 取值大致在 0 – 100 之间。

3. 大致都可以分为首（左）、中、尾（右）三段，三段都不是完全线性的，有一定弯曲度。

4. 中段的斜率小于首段和尾段的斜率，且中段的起点位置和长度都带有随机性。

5. 本课题只需针对 X=5.0,5.1,5.2,……,9.9,10.0，讨论定标问题一是指曲线必须通过若干已知离散数据点的一种拟合。

3. 成本计算

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

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

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

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

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。

$$C=\frac{1}{M}\sum_{i=1}^{M}S_i \qquad (3)$$

总成本较低的校准方案，认定为较优方案。

4 遗传算法

4.1 遗传算法简介

遗传算法（Genetic Algorithm）是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。它是由美国的 J.Holland 教授 1975 年首先提出，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。它是现代有关智能计算中的关键技术。

它的基本运算过程如下：

4.1.2 适应度计算

遗传算法是一种概率性搜索算法，但是它并非等概率无目的地任意搜索，而是通过适应度函数来进行选择。适应度大的个体存活进行下一轮遗传。所谓的适应度大就是此方案较符合问题解决方案的标准。因此适应度函数的设计直接影响遗传收敛性与收敛速度。若收敛速度过小，则收敛性较弱；若收敛速度过大，则过早收敛，种群缺少多样性，即满足条件进入下一轮遗传的可行解数目过少。

4.1.3 选择

选择操作决定如何从当前种群中选取个体作为下一代种群的父代个体。不同的选择策略导致的选择压力也不一样。选择压力即最佳个体被选中的概率与平均中概率的比值。压力越大，选中最佳个体的概率越高，但收敛速度较快，容易过早收敛。较小的选择压力则会导致时间上的损失。常用的选择方法有基于比例的适应度分配方法和基于排名的适应度分配方法。

4.1.4 交叉

交叉是指将父代个体的部分结构加以替换重组而产生新的个体。其目的是在下一代获得优良个体，提高遗传算法的搜索能力。一般交叉不能破坏太多个体编码，同时又要产生一些较好的新编码。另外交叉算法要结合个体编码一起考虑。本课题中采用单点交叉作为交叉算法。

4.1.5 变异

变异是遗传算法中保持生物多样性的一种方法。它以一定概率选择某一基因进行改变（对于二进制编码来说就是 1 变 0 或 0 变 1）。变异的概率可以给定，也可以采取自适应取得。通过这种方法可以更广泛地进行基因选择，而且又不过多改变优良的基因编码，是提供多种基因选择的一种基础途径。

4.2 遗传算法在此数学模型中的应用

1）编码

最常用是采用二进制编码，即用一串 51 个 0/1 二进制数构成的符号串来表示一个个体，在初始化过程中，通过一定的概率保证每个样本中至少有两个测试点为 1。

如：p[1]=[0,0,0,0,1,0,0,1,0,……,1,0]

2）解码

对于一个二进制数构成的符号串，1 表示在这个个体中取该值，0 表示不取，然后取出 X=[5.0:1:10]中对应的 x 值。

3）选择

选择过程是利用解码后求得的各个体对应的平均成本，取其倒数在样本中所占比例作为该样本适应度，淘汰一些较差的个体而选出一些比较优良的个体，以进行下一步的交叉和变异操作。

4）交叉

采用单点交叉的方法来实现交叉算子，即按选择概率 PC 在两两配对的个体编码串中随机设置一个交叉点，然后在该点相互交换两个配对个体的部分基因，从而形成两个新的个体。

5）变异

对于二进制的基因串而言，变异操作就是按照变异概率随机选择变异点，在变异点处将其位取反即可。

6）遗传算法（ GA)对应流程图见附录

5.实验程序各步骤分析

5.1 拟合方法

5.1.1 三次样条插值法

5.1.1.1 方法概述

三次样条插值法一种非线性插值法，它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。三次样条插值法每次选取相邻的 4 个点确定一条三次曲线，再取出中间两个点之间的三次曲线作为样条，然后借助样条来计算出插值点的估计值。实际计算时还需要引入边界条件才能完成计算。边界通常有自然边界（边界点的导数为 0），夹持边界（边界点导数给定），非扭结边界（使两端点的三阶导与这两端点的邻近点的三阶导相等）。

5.1.1.2 优缺点分析

优点：插值所得曲线为光滑曲线，且曲线的导数曲线依然光滑。插值所得曲线通过每一个数据点，使得数据点的估计值等于观察值，无需对其另外处理。而且真实 X-Y 曲线的形态，对插值所得曲线与真实 X-Y 曲线的吻合度影响较小。

缺点：运算量较大，速度较慢。

5.1.1.3 MATLAB 实现

MATLAB 中实现插值的语句为：

```
Y1=interp1(X, Y, datax(i,1:maxSize),'spline') ;
```

其中 X 和 Y 分别为数据点的 X 值和 Y 值，列表 datax(i,1:maxSize) 为所有数据点的 x 制，Y1 为所有数据点 Y 的估测值列表。MATLAB 中将非扭结边界条件作为默认的边界条件。

5.1.2 多项式拟合

5.1.2.1 方法概述

多项式拟合是一种最为基本的拟合方式。对给定数据（xi,yi)在给定的函数类Φ中，求 p(x)∈Φ，使误差 ri=p(xi)-yi 为最小，这样求得的函数 p(x)就是拟合出来多项式函数。

5.1.2.2 优缺点分析

优点:运算速度快，易于理解，可以很方便的进行操作和分析。拟合的效果可能产生极其符合实验点的拟合函数。

缺点:拟合的精度不能有平均的保证。真实 X-Y 曲线的形状对拟合曲线的拟合精度影响较大。

5.1.2.3 MATLAB 实现

MATLAB 中的实现语句为：

```
p=polyfit(X, Y, 3);
Y1=polyval (p, dataX (i, :));
```

其中 X 和 Y 分别为数据点的 X 值和 Y 值，dataX(i,:)为所有数据点的 X，Y1 为所有数据点 Y 的估测值列表。

5.1.2 比较讨论

多项式拟合方法虽然快，但不够精准，而三次样条插值法运算量大，但是求得的拟合曲线更为精准，因而本次试验选取三次样条插值法。

5.3 初始化方法

遗传算法中的初始种群由随机方式产生。即以 51 个 0,1 随机数组成一个编码来表示一种取点方案。这样重复 100 次，得到 100 组编码，以此作为遗传的初始种群。对问题的解做一些估算，易知拥有最小平均成本的取点方案所取点数不会超过 10，因此在随机构造初始种

群时可调整其概率，使初始种群中个体取点方案编码的“1”的数目平均值在 10 左右，这样可以减少循环次数，提高程序效率。

5.2 适应度计算

本次试验适应度取值为个体的平均成本的倒数在种群中所占的比例。并且是每个个体的适应度参与到交叉互换的过程中。

5.3 自然选择方法

对每个个体进行成本计算后，排序后，将成本最大的 3 位个体“杀死”，然后换之以成本最小的 3 位个体。

5.4 交叉互换方法

根据自然选择中个体依据成本的排序，用适应度的一定倍数与随机数相乘，我们尝试过大于 0.5 的个体可以参与交叉互换或小于 0.5 的个体参与交叉互换，将自己的性征遗传给下一代，两次实验结果相似，因此本次课题采用大于 0.5 的个体参与交叉互换，交叉互换采用单点交叉法。

例如：

第 a 个取点方案 pop[a][51]=[0,0,0,1,1,0,0,……,1,0]

第 b 个取点方案 pop[b][51]=[1,1,0,0,0,1,1,……,0,0]

第 q=5 个位置交换

则 pop[a][51]=[0,0,0,1,1,1,1,……,0,0]；

Pop[b][51]=[1,1,0,0,0,0,1,……,1,0]

5.5 变异方法

设定变异概率为 0.1，因为种群较少，变异概率太低的话，无法产生更加优良的个体。每个个体所产生的随机数如果小于 0.1，该个体有机会变异，对一个体重的每一个测试点所对应的随机数，如果小于 0.02，该点就会变异。

6. 实验结果及感想

运行时间：大约 15-20 分钟。

运行 4 次后得到的结果：

2	9	19	26	33	43	50	95.1147
2	9	20	27	34	44	50	95.2219
2	9	20	26	34	43	50	95.2375
3	10	20	27	34	44	50	95.2437

感想：

1）但是我们发现使用三次样条插值的方法，似乎成本存在一个极限值 95。如果需要降低成本，只有进一步优化算法或者是采用其他曲线的表达形式。

2）遗传算法的计算量非常的庞大，因为计算机进行三次样条插值的过程耗时比较大，我们测算出 MATLAB 通过我们遗传算法设置的初始参数，初始种群 100 以及交配 200 代使得电脑需要运行成千上万次上述过程，导致总运行时间非常的长，有时时间达到 20 分钟。

3）交叉用的是单点法，直接交叉两段基因，所以不利于搜索的全面性，易陷入局部极值，即最终一代里出现大量与最优先完全相同的个体。

4）在选择下一代群体时，最佳个体的生存机会将显著增加，最差个体的生存机会将被剥夺，低适值个体淘汰太快容易使算法收敛于局部最优解，这可能是成本存在极限值的原因。

5）在大半个学期的学习实践过程中，我们小组遇到了许多的困难，也学习了许多的知识，同样也收获了许多。我们小组在学习 MATLAB 的用法和对遗传算法的理解上上花费了很多的时间，

之后随着对 MATLAB 和遗传算法的逐渐熟悉，我们小组的工作也进展的越来越顺利，我们首先使用了三次多项式拟合的方式，但结果不是很理想，然后更换了拟合方法，但是程序仍然有很大 BUG，最后经过面谈时老师的教导，我们完善了算法修复了许多逻辑 BUG，并取消了取第一个点和最后一个点的限制，并得到了显著地成效，经过 200 代的遗传，最后我们小组得到了较为满意的最优解，最优解为[2，9，19，26，33，43，50]，最优成本为 95.1147.

6)最后，衷心感谢老师在讲座和面谈中的悉心教导，这门课使我们受益匪浅。

7. 参考资料

《统计推断讲座 1_课程动员》上海交通大学电子工程系 2015 年 9 月

《统计推断讲座 2_问题的提出和基本求解思路》上海交通大学电子工程系 2015 年 9 月

《统计推断讲座 3：问题的求解途径》上海交通大学电子工程系 2015 年 9 月

《MATLAB 教程》　罗建军 电子工业出版社

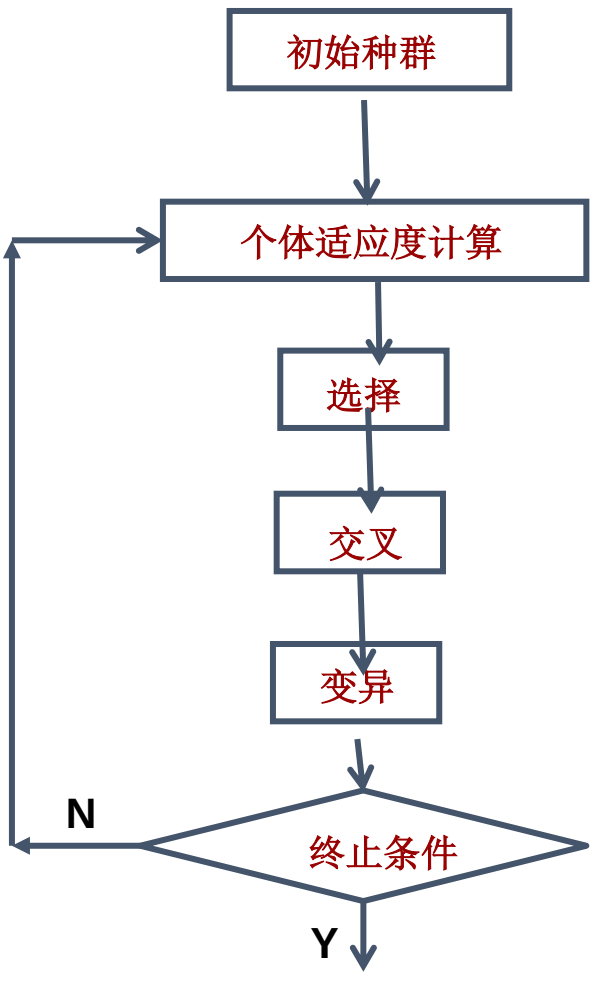
《演化程序遗传算法和数据编码》　【美】Z.米凯利维茨　科学出版社

遗传算法_百度百科

http://baike.baidu.com/link?url=O0b4P0D0fdVtkhWRtVDs_ZzsdRYeuBtNk5rONQ1Sh5S7EYYeYy3C-JYRFWQ6dN2Atbt6QIIdIaqAwplb4Ci9dq

附录（一）：

遗传算法图解：



附录（二）：

Matlab 程序：

Main 函数:

```
global data;
global datax;
global datay;
global maxSize;
global popsize;
global sampleSize;
global cost;
global fitable;
global pop;
global generation;
popsize = 100;
generation = 200;
cost = zeros(popsize,1);%初始化样本成本
fitable = zeros(popsize,1);%初始化样本适应度
data = csvread('dataform.csv');%打开数据库
[sampleSize,maxSize] = size(data);%保存数据库大小
for i=1:(sampleSize/2)%读入数据库数据
    datax(i,:) = data(2*i-1,:);
    datay(i,:) = data(2*i,:);
end
initalize();%初始化种群
select();%初代自然选择
for i=1:generation
    cross();%交叉
    mutate();%变异
    select();%自然选择
end
[~,mm] = sort(cost);%最后一代成本排序
min = pop(mm(1),:);%找出最小成本
log = logical(min);%最小成本对应个体解码
pp = (1:1:51);
minpop = pp(log);
disp(minpop);
disp(cost(mm(1)));
```

初始化种群：

```
function initalize()
%初始化种群
global popsize;
```

```
global pop;
global maxSize;
pop = zeros(popsize,maxSize);%定义种群大小
for i=1:popsize
    tmp = rand(1,maxSize);
    for j=1:maxSize
        if(tmp(j)<0.2)
            pop(i,j) = 1;%初始化个体基因
        end
    end
end
End
```

自然选择：

```
function select()
%优胜劣汰
global pop;
global popsize;
global maxSize;
global cost;
global fitable;
global nn;
for k=1:popsize %计算各个成本
    num = 0;
    for j=1:maxSize
        if(pop(k,j)==1)
            num = num + 1;
        end
    end
    apop = pop(k,:);
    cost(k) = sampleCost(apop,num);
end
[~,nn] = sort(cost);%成本排序
for i=(int8(popsize*0.99)+1):popsize %“杀死”成本较高的个体
        pop(nn(i),:) = pop(nn(i-int8(popsize*0.99)),:);
        cost(nn(i)) = cost(nn(i-int8(popsize*0.99)));
end
sumofth = 0;
for k=1:popsize
    fitable(k) = 1/cost(k);
    sumofth = sumofth + fitable(k);
end
for k=1:popsize  %适应度计算
    fitable(k) = fitable(k)/sumofth;
```

```
end
end
```

交叉互换：

```
function cross()
%交叉产生子代
global pop;
global popsize;
global maxSize;
global fitable;
global nn;
global generation;
crossra = rand(popsize,1)*(300-generation);%随着个体优化程度增大，交叉概率降低
i = 1;
while(i<=popsize)
    while(i<=popsize&&(fitable(nn(i))*crossra(i))>0.5)%选择第一个交叉对象
        i = i + 1;
    end
    cross1 = i;
    i = i + 1;
    while(i<=popsize&&(fitable(nn(i))*crossra(i))>0.5)%选择第二个交叉对象
        i = i + 1;
        if(i==popsize+1)
           break;
        end
    end
    cross2 = i;
    if(cross1<=popsize&&cross2<=popsize)
        crosssta = randi(50) + 1;%选择交叉位置
        tmp = pop(nn(cross1),crosssta:maxSize);
        pop(nn(cross1),crosssta:maxSize) = pop(nn(cross2),crosssta:maxSize);
        pop(nn(cross2),crosssta:maxSize) = tmp;
    end
    i = i + 1;
end
end
```

变异：

```
function mutate()
%子代变异
global maxSize;
global pop;
global popsize;
mutpro = 0.1;
```

```
for i=1:popsize
    tmp1 = rand();%个体变异概率
    if tmp1<=mutpro
        tmp2 = rand(1,maxSize);%个体内基因变异概率
        for j=1:maxSize
            if(tmp2(j)<0.02)
                if(pop(i,j)==1)
                    pop(i,j) = 0;
                else
                    pop(i,j) = 1;
                end
            end
        end
    end
end
end
```

个体对应成本：

```
function money = sampleCost(method,n)
%样本的平均定标成本
global datax;
global datay;
global maxSize;
global sampleSize;
money = 0;
for i=1:sampleSize/2
    log = logical(method);%解码
    x = datax(i,log);
    y1 = datay(i,log);
    y2 = datay(i,:);
    z = interp1(x,y1,datax(i,1:maxSize),'spline');%插值拟合
    for j=1:maxSize;
        money = money + pointCost(y2(j),z(j));
    end
end
money = money / (sampleSize/2) + 12 * n;%平均定标成本
end
```

单点测定成本：

```
function y = pointCost(x1,x2)
%单点定标误差成本
if abs(x1-x2)<=0.4
    y = 0;
```

```
elseif abs(x1-x2)<=0.6
    y = 0.1;
elseif abs(x1-x2)<=0.8
    y = 0.7;
elseif abs(x1-x2)<=1
    y = 0.9;
elseif abs(x1-x2)<=2
    y = 1.5;
elseif abs(x1-x2)<=3
    y = 6;
elseif abs(x1-x2)<=5
    y = 12;
else
    y = 25;
end
end
```

答案检验函数：

```
%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%

my_answer=[ 3, 12, 22, 31, 43, 50 ];%把你的选点组合填写在此
my_answer_n=size(my_answer,2);

% 标准样本原始数据读入
minput=dlmread('20150915dataform.csv');
[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample,npoint);
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
for i=1:nsample
    x(i,:)=minput(2*i-1,:);
    y0(i,:)=minput(2*i,:);
end
my_answer_gene=zeros(1,npoint);
my_answer_gene(my_answer)=1;


% 定标计算
index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample
    % 请把你的定标计算方法写入函数 mycurvefitting
```

```
    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

% 成本计算
Q=12;
errabs=abs(y0-y1);

le0_4=(errabs<=0.4);
le0_6=(errabs<=0.6);
le0_8=(errabs<=0.8);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-le0_8)+1.5*(le2_0-le1_
0)+6*(le3_0-le2_0)+12*(le5_0-le3_0)+25*g5_0;
si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
cost=sum(si)/nsample;

% 显示结果
fprintf('\n 经计算，你的答案对应的总体成本为%5.2f\n',cost);

function y1 = mycurvefitting( x_premea,y0_premea )

x=[5.0:0.1:10.0];

y1=interp1(x_premea,y0_premea,x,'spline');

end
```