

实验名称：（有天安门升旗估算北京市的日照时长）

专业班级	软件三班
学号	2220202929
姓名	蔡博宇
实验日期	2021/12/1

1、实验环境

MATLAB2020b

2、实验目的和要求

实验目的和要求：由天安门升旗的时间和降旗时间来间接求得北京市的日照时间，并根据给出条件，求出某一事件的日照时间。

3、解题思路、代码

3.1 解题思路

对于本题，使用牛顿插值根据所给条件，构造插值函数，通过构造插值函数求得所求点的日照时长。

3.2 代码

Sheng_qi.m

```
clear;clc
load date.mat;
%% 数据初始化
[m,n] = size(date);
date1 = date(:,1);      % 日期 (date)
flag1_time = date(:,2); % 升旗时间 (min)
flag2_time = date(:,3); % 降旗时间 (min)
sun_time = zeros(m,1); % 日照时长 (min)
new_date = [8];         % 预测的日期 (date)
new_date_time = zeros(1,1);% 预测的结果 (min)
new_date2 = zeros(1,4); % 更新后的日期 (min)
new_sun_time2 = zeros(1,4);% 更新后的日落时间 (min)

%% 计算日照时间
```

```

for i = 1 : m
    sun_time(i) = flag2_time(i) - flag1_time(i);
end

%% 进行牛顿插值
[martix,new_sun_time] = fun1(date1,sun_time,new_date)
martix2 = [date1,martix]

new_date2 = [date1(1),
             new_date,
             date1(2:end)];
new_sun_time2 = [sun_time(1),
                 new_sun_time,
                 sun_time(2:end)]

%% 绘图
figure(1)
plot(date1,sun_time,'o-');grid on; %添加网格
hold on;
plot(new_date2,new_sun_time2,'o-');grid on; %添加网格
xlabel('日期(date)');
ylabel('日照时长(min)');
title('牛顿插值前后天安门广场某月内日照时长');
legend('插值前','插值后'); %图例

```

fun1.m

```

function [A,y]= fun1(X,Y,x)
%   Newton 插值函数
%   X 为已知数据点的 x 坐标
%   Y 为已知数据点的 y 坐标
%   x 为插值点的 x 坐标
%   函数返回 A 差商表
%   y 为各插值点函数值

n=length(X); m=length(x);
for t=1:m
    z=x(t); A=zeros(n,n);A(:,1)=Y';
    s1=0.0; y=0.0;
    for j=2:n
        for i=j:n
            A(i,j)=(A(i,j-1)- A(i-1,j-1))/(X(i)-X(i-j+1));// 相间插值
        end
    end
end

```

```

end
for k=1:n
    p=1.0;
    for j=1:k-1
        p=p*(z-X(j));
    end
    s1=s1+A(k,k)*p;
end
s2(t)=s1;
end
y=s2;
end

```

4、实验步骤

4.1 输入：

	1	2	3
1	1	369	1078
2	15	383	1056
3	22	391	1046

4.2 输出：

4.2.1 计算结果和插值矩阵

```

martix2 =

    1.0000    709.0000         0         0
   15.0000   673.0000   -2.5714         0
   22.0000   655.0000   -2.5714         0

new_sun_time2 =

    709
    691
    673
    655

```

4.2.2 插值图像

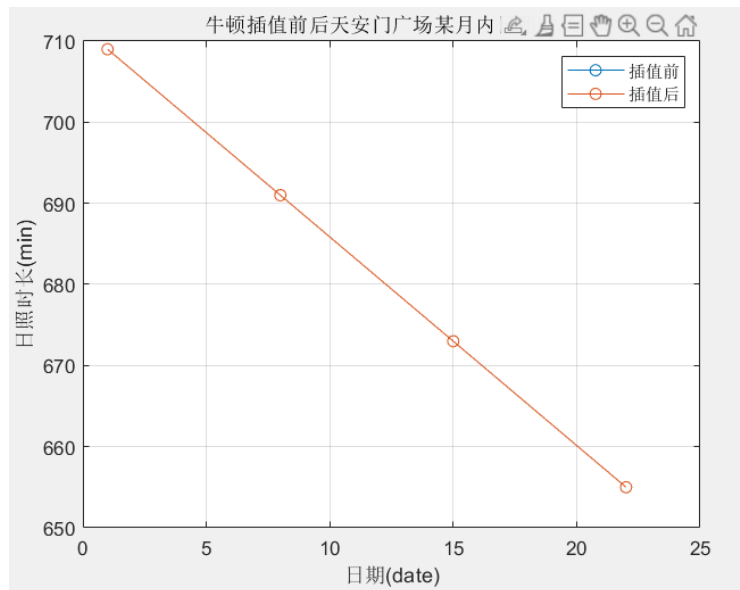


图 1 北京日照时长插值图像

5、讨论和分析

对算法及实验结果进行分析讨论。

由计算给出的插值前后图像几乎重合可得出牛顿插值算法的优越性,以及在小范围内插值效果的准确性。

实验名称：（黄河排沙问题）

专业班级	软件工程三班
学号	2220202929
姓名	蔡博宇
实验日期	2021/12/04

1、实验环境

MATLAB R2020b

2、实验目的和要求

由每次观测的时间点和水流量和含沙量，计算（1）排沙任何时刻的排沙量和排沙总量
（2）确定排沙量和水量的关系

3、解题思路、代码

3.1 解题思路

对于问题一，根据排沙量和时间变化的规律，建立插值模型，利用所写的牛顿插值求出插值函数（1），然后利用 matlab 中 **spline 函数**和 **pchip 函数**来进行三次样条插值和埃尔米特插值来求得高精度的插值函数（2）和（3），并通过积分求得总排沙量，依据各插值函数的拟合优度 **R** 进行排序，选取最优插值函数，并依据此进行积分求得总排砂量。

对于问题二，根据排沙量和时刻的函数图像，可看出排沙量和时刻并不成线性关系，而又进一步计算得到，根据最高点的排沙量可以将数据分成两断，分别建立拟合模型。由散点图的绘制可知，此猜想成立。然后对由此根据两段数据分别计算排沙量和水流量的函数表达式，由计算得出第一段近似于二次函数并求得,第二段近似于四次函数，并用 **polyfit 函数**和 **polyval 函数**求得其表达式。

3.2 代码

huang_he_sha.m

```
clc;clear;
load data.mat
%% 数据预处理
% 根据观测次数 以时间节点为轴
% 建立时刻与观测次数的关系式  $t(i) = 3600 \times (12 \times i - 4)$ 
% 由于给出了每次观测的水流量  $v(i)$ 和含沙量  $c(i)$  计算排沙量  $y(i)=v(i)*c(i)$ 
```

% 由于给出了计算结果 本次实验直接使用预处理训练集

```
[m,n] = size(data);
i = data(1,:) % 节点
t = data(2,:); % 时刻 (h)
y = data(3,:) % 排沙量(t)
new_i = 1:1:24;
new_t = 3600 .* (21 .* new_i - 4);
```

%% 插值函数 求任意时刻的排沙量

```
p1 = pchip(t,y,new_t); %分段三次埃尔米特插值
```

```
p2 = spline(t,y,new_t); %三次样条插值
```

```
plot(t,y,'o',new_t,p1,'r-',new_t,p2,'b-');
```

```
legend('样本点','三次埃尔米特插值','三次样条插值','Location','SouthEast') %标注显示在东南方向
```

%% 利用三次样条插值函数计算总排沙量 解决(1)

```
t1 = t(1); t2 = t(end);
```

```
pp = spline(t,y);
```

```
pp2 = fn2fm(pp,'pp');
```

```
pp2
```

```
T1 = quadl(@(tt)fnval(pp,tt),t1,t2) %% 计算总排沙量 3.2.2 问题 (2) 的求解
```

huang_he_sha2.m

```
clc;clear;
```

```
load data.mat
```

%% 数据初始化

```
[m,n] = size(data);
```

```
i = data(1,:) % 节点
```

```
t = data(2,:); % 时刻 (h)
```

```
y = data(3,:) % 排沙量(t)
```

%% 绘图 观察大致关系

```
figure(1)
```

```
plot(t,y,'b-');
```

```
xlabel('时刻(h)');
```

```
ylabel('排沙量(t)');
```

```
title('排沙量和时刻的关系');
```

```

%% 分段绘图
% 由观察知排沙量和时刻并非是线性关系,而是由两线性部分组成
% 因此 将问题分成两部分: 增长 + 降低
t1 = 460800; % 由观察知分段点位前端最高值 记录此时 x 值
index = find(t == t1);
% 分别画出两段的散点图观察
figure(2)
subplot(1,2,1)
plot(t(1:index),y(1:index),'*');
xlabel('时刻(h)');
ylabel('排沙量(t)');
title('排沙量和时刻的关系');
subplot(1,2,2)
plot(t(index:end),y(index:end),'*');
xlabel('时刻(h)');
ylabel('排沙量(t)');
title('排沙量和时刻的关系');

```

huang_he_sha_2.m

```

clc;clear;
load data.mat
load data1.mat

%% 数据初始化
[m,n] = size(data);
i = data(1,:) % 节点
t = data(2,:); % 时刻 (h)
y = data(3,:) % 排沙量(t)
v = data1(1,:); % 水流量(m^3/s)
c = data1(2,:) % 含沙量(t)

%% 模型求解
% 由散点图知一二段大致成线性关系
% 根据所给值分段建立模型
y = c .* v; % 计算排沙量
format long e
t1 = 460800; % 由观察知分段点位前端最高值 记录此时 x 值
index = find(t == t1);

%% 第一段模型拟合

```

```

for j = 1:2
    nihe1{j} = polyfit(v(1:index),y(1:index),j); %拟合多项式 系数从高到低
    yhat1{j} = polyval(nihe1{j},v(1:index)); %求预测值
    %以下求误差平方和与剩余标准差
    remse1(j) = sqrt(sum((y(1:index)-yhat1{j}).^2)/(10-j)); % 求剩余标准差
end
nihe1{:}
cha1 = remse1

%% 第二阶段拟合
% 以下是第二阶段的拟合
for j = 1:3
    nihe2{j} = polyfit(v(index+1:end),y(index+1:end),(j+1)); % 这里使用细胞数组
    yhat2{j} = polyval(nihe2{j},v(index+1:end));
    remse2(j) = sqrt(sum((y(index+1:end)-yhat2{j}).^2)/(11-j)); % 求剩余标准差
end
nihe2{:} % 显示细胞数组的所有元素
cha2 = remse2
format

```

4、实验步骤

4.1（1）排沙任何时刻的排沙量和排沙总量

4.1.1 输入：

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
19	20	21	22	23	24												
28800	72000	115200	158400	201600	244800	288000	331200	374400									
417600	460800	504000	547200	590400	633600	676800	720000	763200									
806400	849600	892800	936000	979200	1022400												
57600	114000	157500	187000	207000	235200	250000	265200	286200									
302400	312800	307400	306800	300000	271400	231000	160000	111000									
91000	54000	45500	30000	8000	4500												

4.1.2 输出：

（1）插值函数拟合与预测图像

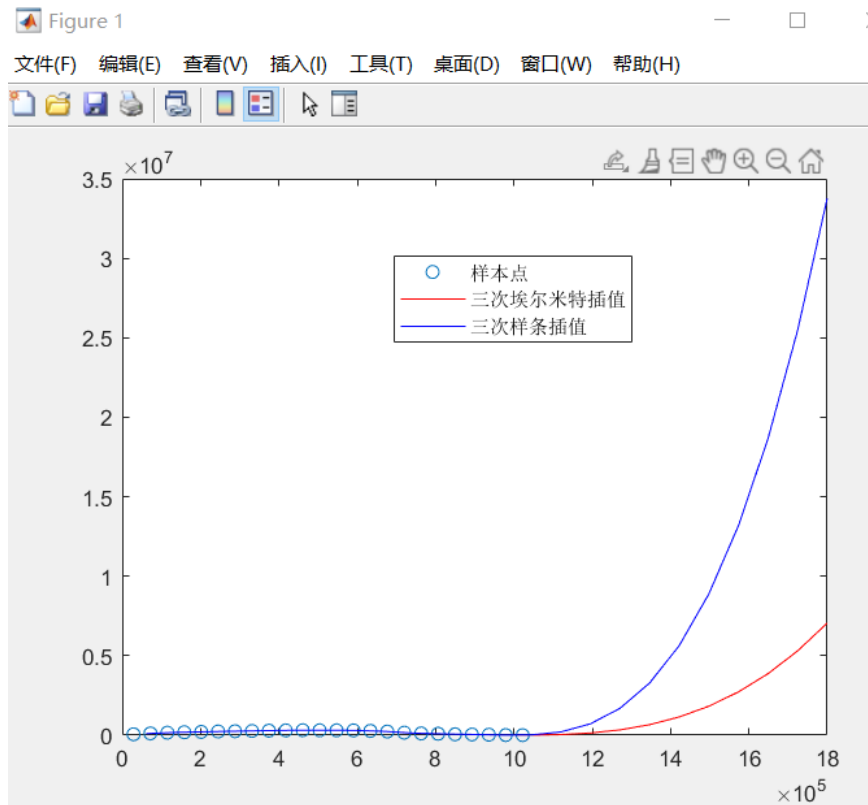


图 1 插值图像

(2) 插值函数与总排沙量结果

```
pp2 =

包含以下字段的 struct:

    form: 'pp'
  breaks: [1×24 double]
   coefs: [23×4 double]
  pieces: 23
   order: 4
    dim: 1

T =

1.8440e+11
```

4.2 (2) 确定排沙量和水量的关系

4.2.1 输入:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
19	20	21	22	23	24												

28800	72000	115200	158400	201600	244800	288000	331200	374400
417600	460800	504000	547200	590400	633600	676800	720000	763200
806400	849600	892800	936000	979200	1022400			

57600	114000	157500	187000	207000	235200	250000	265200	286200
302400	312800	307400	306800	300000	271400	231000	160000	111000
91000	54000	45500	30000	8000	4500			

28800	72000	115200	158400	201600	244800	288000	331200	374400
417600	460800	504000	547200	590400	633600	676800	720000	763200
806400	849600	892800	936000	979200	1022400			

57600	114000	157500	187000	207000	235200	250000	265200	286200
302400	312800	307400	306800	300000	271400	231000	160000	111000
91000	54000	45500	30000	8000	4500			

4.2.2 输出:

4.2.2.1.排沙量与时刻总图

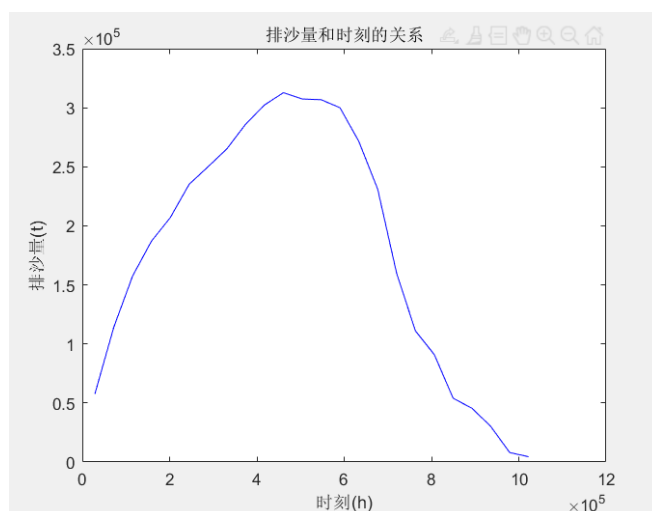


图 2 排沙量与时刻关系

4.2.2.2.两段排沙量和时刻的散点图

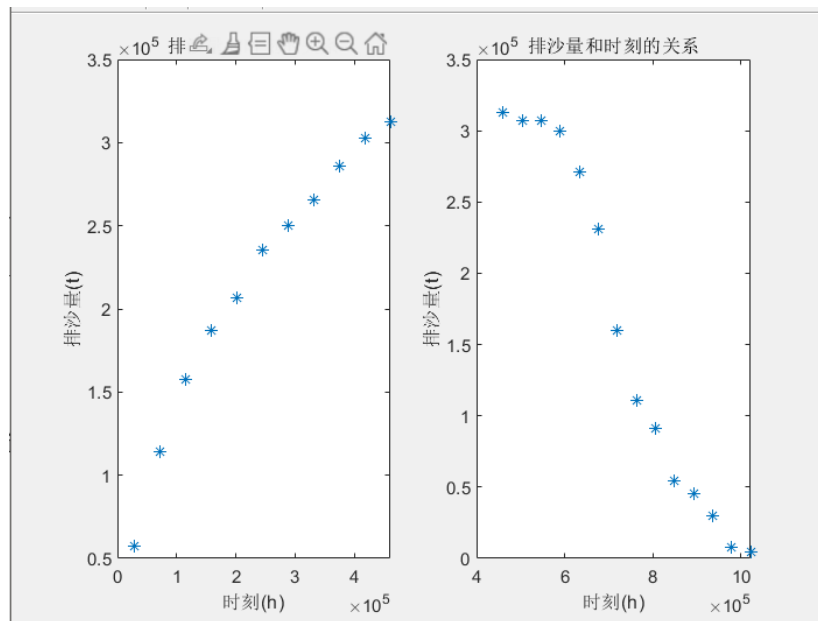


图 3 排沙量和时刻散点图

4.2.2.3.模型计算结果

模型一

```
ans =

    3.363672727272728e+05    -1.922017745454548e+10

ans =

    3.525533108866447e-01    1.6375717171715e+05    -4.672191999999971e+09

cha1 =

    6.509260742501897e+09    1.112915505619032e+09
```

模型二

```
ans =

列 1 至 3

    -1.371159323346807e-11    4.631069869004235e-05    -5.744293648171392e+01

列 4 至 5

    3.055461085308336e+07    -5.701680678576287e+12

cha2 =

    1.889011712691622e+10    1.008622248546389e+10    7.577821433773317e+09
```

5、讨论和分析

对于第二问的结果，发现对于拟合函数 `polyfit` 对于第一段数据，每次循环中所减少的误差以近 6 倍减少，而对于第二段数据前两次拟合效果较好，而在第三次拟合时误差出现显著飞跃，由此得出此函数在正相关型数据中，误差将越来越小，而在负相关型数据中，误差可能随着拟合次数的增加而显著增加，当然本实验计算可能存在偶然性，进一步结论需要更多数据进行多次实验。