

**西安交通大学**

**《计算方法B》上机题目**

**姓名： 彭仕锦**

**学院： 机械工程学院**

**学号： 3120301259**

**2021年1月6日**

1. 海底光缆铺设

海底光缆作为当代国际通信的重要手段，承担了90%的国际通信业务，是全球信息通信的主要载体。海底电缆工程被世界各国公认为复杂困难的大型工程。海底电缆的铺设也是比较困难的一项工作，在浅海区，如水深小于200米的海域缆线采用埋设，而在深海区则采用敷设方式。

在一次施工中，需要在海面宽度为5千米左右的海底沿近似直线铺设一条海底光缆。在铺设光缆之前需要对海底的地形进行初步探测，从而估计所需光缆的长度，为工程预算提供依据。

假设光缆在铺设时可以比较紧密地贴合海床，已通过探测得到一组探测点位置的海床的深度数据，相邻两个探测点按相距100米计算，如下表所示(单位:米)。针对以下数据，请用合适的方法拟合所测数据点,估算所需光缆长度的近似值，精确到小数点后一位小数，并作出铺设河底光缆的曲线图。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **探测点** | **深度** | **序号** | **探测点** | **深度** |
| 1 | 0 | 341.96 | 27 | 2600 | 328.92 |
| 2 | 100 | 311.10 | 28 | 2700 | 314.11 |
| 3 | 200 | 315.74 | 29 | 2800 | 319.84 |
| 4 | 300 | 324.49 | 30 | 2900 | 321.28 |
| 5 | 400 | 321.14 | 31 | 3000 | 333.84 |
| 6 | 500 | 333.09 | 32 | 3100 | 319.68 |
| 7 | 600 | 318.68 | 33 | 3200 | 321.20 |
| 8 | 700 | 338.88 | 34 | 3300 | 320.89 |
| 9 | 800 | 325.66 | 35 | 3400 | 325.90 |
| 10 | 900 | 309.84 | 36 | 3500 | 335.28 |
| 11 | 1000 | 328.62 | 37 | 3600 | 332.00 |
| 12 | 1100 | 321.80 | 38 | 3700 | 324.47 |
| 13 | 1200 | 320.74 | 39 | 3800 | 313.36 |
| 14 | 1300 | 308.30 | 40 | 3900 | 330.60 |
| 15 | 1400 | 328.45 | 41 | 4000 | 322.32 |
| 16 | 1500 | 312.62 | 42 | 4100 | 315.10 |
| 17 | 1600 | 317.83 | 43 | 4200 | 319.89 |
| 18 | 1700 | 327.40 | 44 | 4300 | 316.75 |
| 19 | 1800 | 323.75 | 45 | 4400 | 321.92 |
| 20 | 1900 | 313.48 | 46 | 4500 | 327.64 |
| 21 | 2000 | 316.66 | 47 | 4600 | 320.40 |
| 22 | 2100 | 340.19 | 48 | 4700 | 329.88 |
| 23 | 2200 | 332.40 | 49 | 4800 | 332.00 |
| 24 | 2300 | 319.20 | 50 | 4900 | 344.35 |
| 25 | 2400 | 334.64 | 51 | 5000 | 348.41 |
| 26 | 2500 | 311.43 |  |  |  |

本题首先要对所测量的 51 个点进行数据拟合，再对拟合的曲线进行线积分运算即可得到所需光缆长度的近似值。

数据近似的方法一般有两种，一个是多项式插值，另一种是分段插值。如果将所有的测量点都用上，采用多项式插值时，由于龙格现象的存在，随着点数的增大，有时会在两端产生激烈的震荡，使函数不收敛。因此，为了将所有的测量点都用上，本题应采用分段插值方法求解。而为了改善分段插值所引起的曲线的光滑性不够的问题，常采用较高的分段多项式改善曲线的光滑性质。故本题采用三次样条插值。

三次样条插值算法结构如下所示：

**算法**

2.11

1. -1

1. 将M矩阵中的元素个数存入m中
2. 将x中的元素个数存入r中

其中记在处的二阶导数，在自然样条插值中，。得到河底光缆曲线s(x)后，可利用第一类积分计算光缆长度l为：

**源程序：**

按照上述算法所编写的Matlab程序如下所示：

clc

clear

filename="sea2020.csv";

f=csvread(filename,1,0);

x=f(:,1); %等分点数组

X=0:0.1:5000; %插值点数组

y=f(:,2); %深度数组

n=length(x); %等分点数目

N=length(X); %插值点数目

%计算三次样条函数s(x)

M=y; %计算y的二阶差商并存储在矩阵M中

for k=2:3

for i=n:-1:k

M(i)=(M(i)-M(i-1))/(x(i)-x(i-k+1));

end

end

%计算三对角阵系数a,b,c及右端向量d

h(1)=x(2)-x(1);

for i=2:n-1

h(i)=x(i+1)-x(i);c(i)=h(i)/(h(i)+h(i-1));

a(i)=1-c(i);b(i)=2;d(i)=6\*M(i+1);

end

%选择自然边界条件

M(1)=0;M(n)=0;b(1)=2;b(n)=2;c(1)=0;a(n)=0;d(1)=0;d(n)=0;

%追赶法计算样条参数M

u(1)=b(1);

ynew(1)=d(1);

%追

for k=2:n

len(k)=a(k)/u(k-1);

u(k)=b(k)-len(k)\*c(k-1);

ynew(k)=d(k)-len(k)\*ynew(k-1);

end

M(n)=ynew(n)/u(n);

%赶

for k=n-1:-1:1

M(k)=(ynew(k)-c(k)\*M(k+1))/u(k);

end

for m=1:N

k=1;

for i=2:n-1

if X(m)<=x(i)

k=i-1;

break;

else

k=i;

end

end

%在各区间用三次样条插值函数计算X点处的值

h=x(k+1)-x(k);x1=x(k+1)-X(m);x2=X(m)-x(k);

s(m)=(M(k)\*(x1^3)/6+M(k+1)\*(x2^3)/6+(y(k)-(M(k)\*(h^2)/6))\*x1+(y(k+1)-(M(k+1)\*(h^2)/6))\*x2)/h;

end

%计算所需光缆长度

Length=0;

for i=2:N

Length=Length+sqrt((X(i)-X(i-1))^2+(s(i)-s(i-1))^2);

end

%河底光缆长度

fprintf("河底光缆长度=%.1fm\n",Length);

%绘制铺设河底光缆的曲线图

plot(x,y,'\*',X,s,'-')

xlabel('宽度/m');ylabel('深度/m');title('铺设河底光缆的曲线图');

grid on

计算结果：



采用三次样条插值函数拟合后的河底光缆曲线图：



2.新冠病毒疫情分析

新型冠状病毒(2019-nCoV)是从2019年年底开始在全世界范围内大规模流行的可引起感冒以及中东呼吸综合征（MERS）和严重急性呼吸综合征（SARS）等较严重疾病的一种恶性病毒，严重地威胁到了全人类的安全。虽然到目前为止，在中国境内已经得到了比较好的控制，但在其它国家依然没有得到有效的控制，该病毒仍在继续横行扩散，感染人数与死亡人数依然在日益增多，引起了更多人的关注和重视，希望尽早能将这一病毒有效控制，让人们能够在和平、健康的环境中生活的更加安逸。

现在，收集了全世界的新冠病毒每日的疫情数据（数据文件为alltime\_world.csv），以及美国每日的疫情数据（数据文件为alltime\_american.csv），见附2：

1. 请根据全球新冠病毒疫情数据，选用适当的函数进行拟合，得到每日确诊人数的变化趋势，试根据该拟合函数分析今年疫情在全世界的传播情况；
2. 根据该拟合函数，试说明如果在其它国家仍未采取更加有效的控制手段，任由病毒以目前情况传播的情况下，那么到2021年1月1日，全世界感染人数将会达到多少？同时，到这一天，美国的感染人数又将会达到多少？试对此结果做一评价。

数据字段名对照表：

|  |  |
| --- | --- |
| 字段名 | 含义 |
| Date | 日期 |
| total\_confirm | 累计确诊人数 |
| total\_suspect | 累计疑似人数 |
| total\_heal | 累计治逾人数 |
| total\_dead | 累计死亡人数 |
| total\_severe | 累计重症人数 |
| total\_input | 累计输入感染人数 |
| today\_confirm | 当日确认人数 |
| today\_suspect | 当日疑似人数 |
| today\_heal | 当日治逾人数 |
| today\_dead | 当日死亡人数 |
| today\_severe | 当日新增重症人数 |
| today\_storeConfirm | 较上一日现有确诊人数 |
| today\_input | 当日输入感染人数 |

1. 实现思想：

首先，通过matlab中自带的文件操作函数对csv文件进行处理，从而获取文件信息。此题已知全世界新冠的每日确诊人数，比较适合用最小二乘法的方法来进行函数拟合。而通过观察前307天全球每日确诊人数的增长趋势，可以认为采用指数函数或者多项式函数来进行最小二乘法拟合更加合适。

算法

* 1. [形成矩阵]
  2. [变换到]

2. [解三角方程]

2.1

2.2

2.2.1

3. [计算误差]

**源程序：**

按照上述算法所编写的Matlab程序如下所示：

clc

clear

W=csvread("alltime\_world.csv",1,0);

%%

%世界新冠数据

no=W(:,1); %天数

confirm=W(:,3); %世界每日确证人数

dead=W(:,4); %世界每日死亡人数

heal=W(:,5); %世界每日治愈人数

newadd=W(:,6); %每日新增确诊人数

%%

%采用指数函数y=a0\*e^(a1\*x)来拟合世界每日确诊人数

%两边取对数便得ln(y)=ln(a0)+a1\*x,g1(x)=1,g2(x)=x,z=ln(y),a=[ln(a0),a1]'

for i=1:307

g(i,1)=1;

g(i,2)=no(i);

z(i,1)=log(confirm(i));

end

%求解方程g.'ga=g.'z

G=g.'\*g;

Z=g.'\*z;

a=inv(G)\*Z; %得到系数ln(a0),a1

a0=exp(a(1));

a1=a(2);

%得到拟合的指数函数，并将其画出

y=a0\*exp(a1.\*no);

subplot(1,2,1)

plot(no,y,'-')

hold on

plot(no,confirm,'x')

grid on

title('指数拟合y=a0\*exp(a1\*x)');

legend('指数拟合','真实数据');

xlabel('天数');

ylabel('每日确诊人数');

%%

%采用多项式函数y=a0+a1\*x+a2\*x^2+a3\*x^3来拟合，g1(x)=1,g2(x)=x,g3(x)=x^2,g4(x)=x^3

%a=[a0,a1,a2,a3].'

for i=1:307

g1(i,1)=1;

g1(i,2)=no(i);

g1(i,3)=no(i)\*no(i);

g1(i,4)=no(i)\*no(i)\*no(i);

z1(i,1)=confirm(i);

end

%求解方程g1.'g1a=g1.'z

G1=g1.'\*g1;

Z1=g1.'\*z1;

a1=inv(G1)\*Z1;%得到a0,a1,a2,a3

%得到拟合的指数函数，并将其画出

y1=a1(1)+a1(2).\*no+a1(3).\*no.\*no+a1(4).\*no.\*no.\*no;

subplot(1,2,2)

plot(no,y1,'r-')

hold on

plot(no,confirm,'b-x')

grid on

title('多项式拟合y=a0+a1\*x+a2\*x^2+a3\*x^3');

legend('多项式拟合','真实数据');

xlabel('天数');

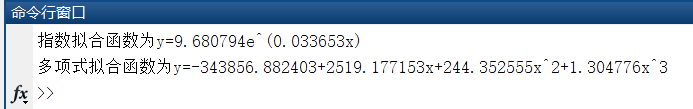
ylabel('每日确诊人数');

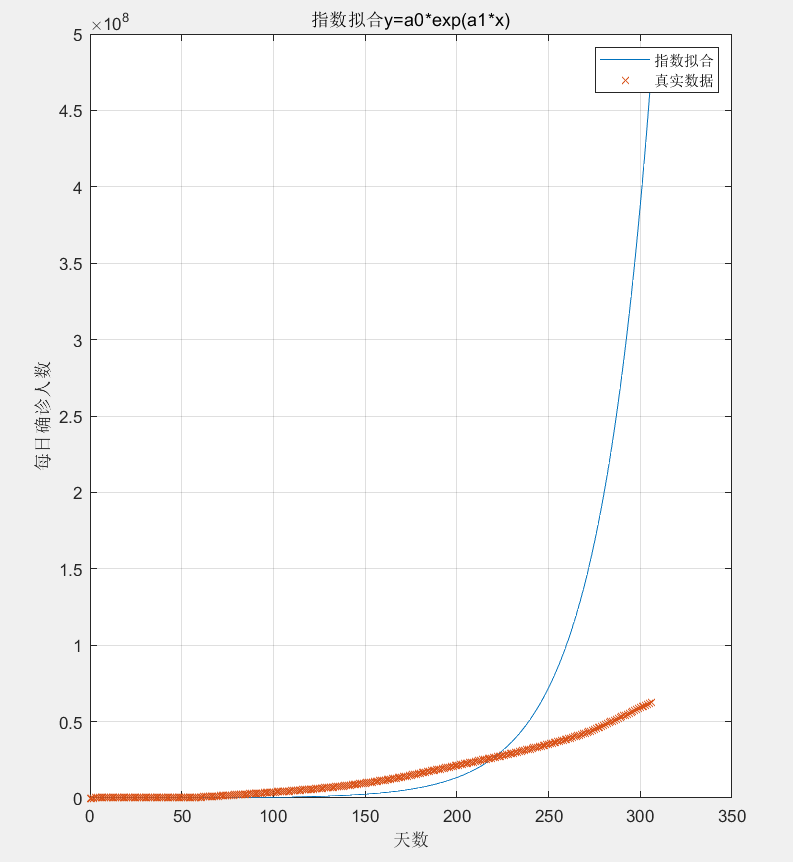
%%

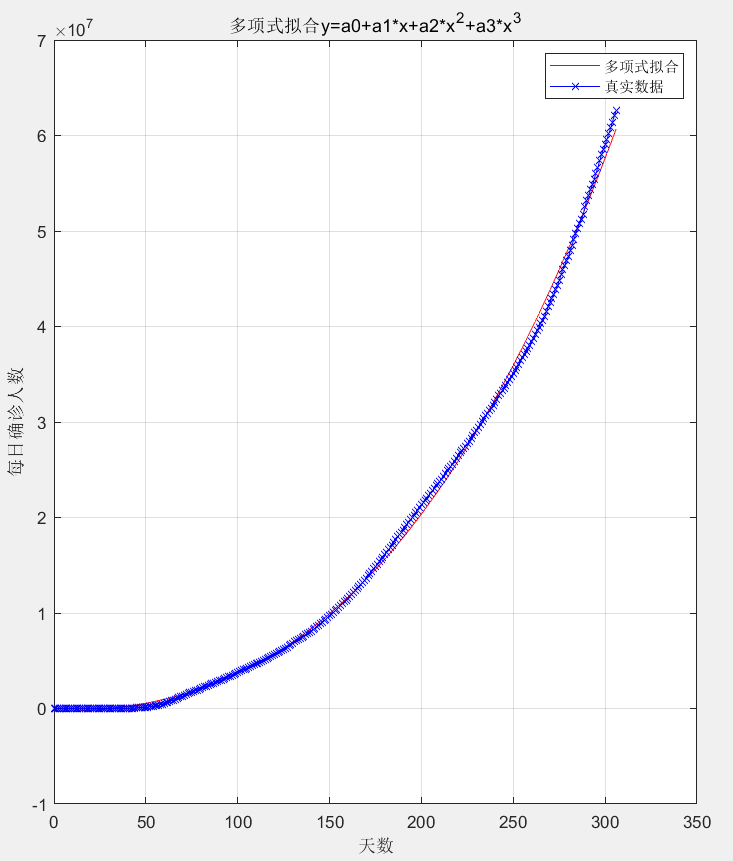
fprintf('指数拟合函数为y=%fe^(%fx)\n',a(1),a(2));

fprintf('多项式拟合函数为y=%f+%fx+%fx^2+%fx^3\n',a1(1),a1(2),a1(3),a1(4));

计算结果：







可以很明显的看到，采用指数函数拟合的结果会在后期增长过快，完全的超过了实际的变化速度，而采用三次多项式拟合的结果则十分贴合世界每日确诊人数。

虽然从拟合的结果来看，世界范围内的病毒传播速度尚还没有达到指数级的增长速度，但是也已经呈现三次多项式

的增长趋势。

很明显恒成立

这意味着每天的确诊人数的增长速度在逐步加快，说明世界范围内的疫情还没有得到有效的控制，疫情形势不容乐观。

1. 由(1)拟合得到的三次多项式当x=0时，对应日期为2020年1月28日，所以2021年1月1日对应于x=339，所以y(339)= 79423133.29956

取整，得到如果在其它国家仍未采取更加有效的控制手段，任由病毒以目前情况传播的情况下，那么到2021年1月1日，全世界感染人数将会达到79423134人。

对于美国的累计确诊人数进行三次多项式的最小二乘法拟合，matlab源程序如下所示：

clc

clear

A=xlsread("alltime\_american.csv");

%%

%美国新冠数据

num=A(:,1); %天数

total\_confirm=A(:,3); %累计确诊

total\_healtotal\_heal=A(:,5); %累计治愈

total\_dead=A(:,6); %累计死亡

today\_confirm=A(:,9); %当日确认人数

today\_heal=A(:,11); %当日治愈人数

today\_dead=A(:,12); %当日死亡人数

%%

%采用多项式函数y=a0+a1\*x+a2\*x^2+a3\*x^3来拟合，g1(x)=1,g2(x)=x,g3(x)=x^2,g4(x)=x^3

%a=[a0,a1,a2,a3].'

for i=1:281

g1(i,1)=1;

g1(i,2)=num(i);

g1(i,3)=num(i)\*num(i);

g1(i,4)=num(i)\*num(i)\*num(i);

z1(i,1)=total\_confirm(i);

end

%求解方程g1.'g1a=g1.'z

G1=g1.'\*g1;

Z1=g1.'\*z1;

a1=inv(G1)\*Z1;%得到a0,a1,a2,a3

%得到拟合的指数函数，并将其画出

y1=a1(1)+a1(2).\*num+a1(3).\*num.\*num+a1(4).\*num.\*num.\*num;

plot(num,y1,'r-')

hold on

plot(num,total\_confirm,'b-x')

grid on

title('多项式拟合y=a0+a1\*x+a2\*x^2+a3\*x^3');

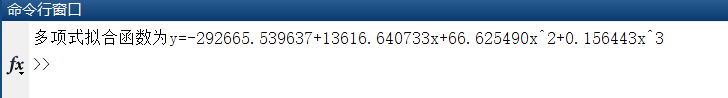
legend('多项式拟合','真实数据');

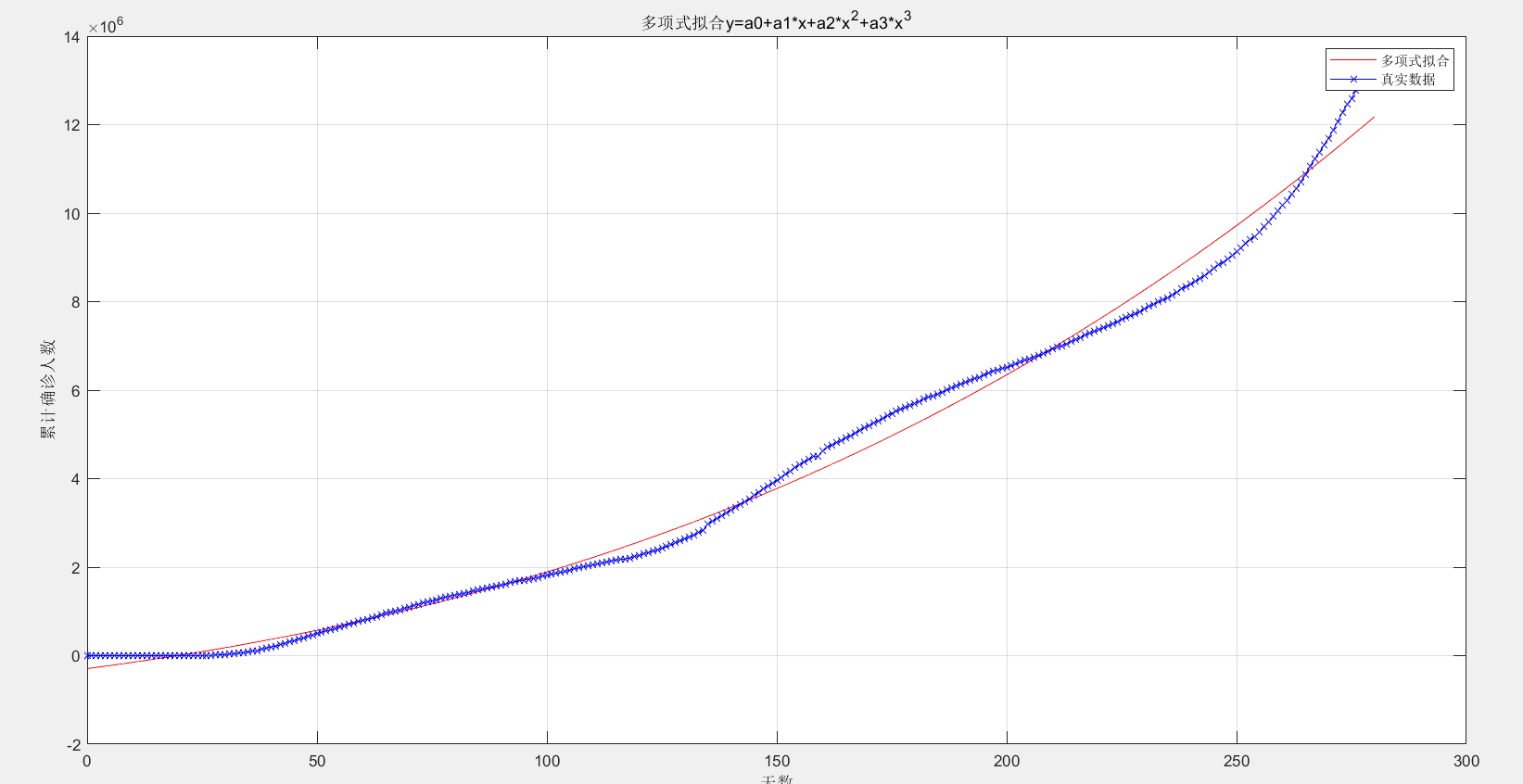
xlabel('天数');

ylabel('累计确诊人数');

fprintf('多项式拟合函数为y=%f+%fx+%fx^2+%fx^3\n',a1(1),a1(2),a1(3),a1(4));

拟合结果：





对于美国累计确诊人数的拟合函数，当x=0时，对应的日期为2020年2月8号，则2021年的1月1日，对应的x=328。

取整，y1(328)=16861920

即在2021年1月1日，通过拟合函数得到的美国累计确诊人数预测结果为16861920人。

根据Worldometers世界实时统计数据显示，截至香港时间1月1日8时47分，美国新冠肺炎累计确诊病例突破2042万例，达到20,426,035例，比预估值多了3564115确诊人数，相对误差为17.4%，这说明采用三次多项式拟合得到的结果精度上还有欠缺，但也可以说明从2020年11月29日到2021年1月1日这一个月时间里面，美国政府对于本土疫情的防控做得很不好，使得疫情传播加重。

截至北京时间2021年1月1日9时01分，全球累计确诊新冠肺炎（COVID-19）病例超过8377万例，新增716,065例，达到83,771,267例，比拟合模型预估的79423134人要多了4,348,133人，相对误差为5.2%。说明采用三次多项式拟合得到的结果精度上还有欠缺。

考虑到三次多项式拟合的结果不是很好，采用四次多项式来对数据进一步拟合。得到美国累计确诊人数的拟合函数为：

此时预测2021年1月1日的结果为：y1(328)= 21085088.41，取整为21085089，预测结果与实际值差了659,054，相对误差为3.2%，较三次多项式拟合的结果要好了很多。

拟合得到世界累计确诊人数的拟合函数为：

预测2021年1月1日的结果为：y(339)= 83862564.9585，取整为83862565，此时与实际值差距为91,298，相对误差为0.11%，证明采用四次多项式插值得到的拟合函数与实际的世界范围内的新冠累计确证人数的拟合程度非常高。

3. 大规模稀疏线性方程组的求解

在大数据应用分析中，大规模稀疏线性方程组的求解问题日益普遍。

线性方程组求解方法一般是以高斯消去法、列主元高斯消去法、迭代法为主。当线性方程组中的系数矩阵是严格对角占优矩阵时，直接使用高斯消去法就可以得到比较准确的解。

为了使数据在各模块之间方便快速传送，可以将方程组的系数矩阵和右端常量一起存储在数据文件中，通过文件在单机中各模块之间或在分布式系统中各主机之间进行传送。在进行求解时，从文件中读出系数矩阵和右端常量，然后使用高斯消去法进行求解。

本例中的数据文件为二进制文件形式，其存贮效率和读写效率更高，其存储结构见附件3。

本题数据文件共有4个，其说明如下表所示，并且各数据文件中的系数矩阵均为严格对角占优的带状矩阵：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **数据文件** | **规模** | **类型** | **根** | **说明** |
| 1 | data20201.dat | 20阶 | 非压缩格式 | 1.314 | 用于测试程序 |
| 2 | data20202.dat | 20阶 | 压缩格式 | 1.314 | 用于测试程序 |
| 3 | data20203.dat | 2000阶左右 | 非压缩格式 | 待解 | 用于程序求解 |
| 4 | data20204.dat | 50000阶左右 | 压缩格式 | 待解 | 用于程序求解 |

对于压缩格式的方程组，在进行求解时可以只对带状区域中的元素进行处理，这样可以大大减少计算量。

（1）请试编写一个统一的程序，实现从4个数据文件中读入方程组的数据，再使用高斯消去法进行求解。

（2）针对本专业中所碰到的实际问题，提炼一个使用方程组进行求解的例子，并对求解过程进行分析、求解。

1. **实现思想**

首先，通过利用matlab中自带的文件操作函数对dat文件进行处理，从而获取文件信息。对压缩带状方程组和非压缩带状方程组分别采用不同方法进行处理。

消去法的中心思想是降维，即将求解的n元方程组的问题转化为先解n-1元方程组，一旦次n-1元方程组的解取得，则剩余的一个未知量自然可以求得。通过这样逐步减少未知量的个数，从而对多元方程组进行求解。消去法包含两个主要步骤：消去和回代。在小区的过程中通过适当的交换方程的顺序保证小区过程能够顺利进行及计算解的精确度，交换的原则是使第k步消去过程中产生的数中绝对值最大的一个换到(k,k)位置而成为第k步消去的主元，即列主元Gauss消去法。

**算法**

* 1. 找满足的下标

输出错误的信息；stop

1. [回代过程]

2.2.1

**源程序**

按照上述算法所编写的Matlab程序如下所示：

clc

clear

%%

%文件操作,读取系数矩阵

[filename,pathname]=uigetfile('\*.dat','选择数据文件'); %选择读取数据文件

num=6; %输入稀疏矩阵文件的个数

name=strcat(pathname,filename); %连接字符构造文件路径

file=fopen(name,'r');

head=fread(file,num,'uint'); %读取二进制头文件

id=dec2hex(head(1)); %读取标识符

fprintf('文件标识符为%s\n',id);

ver=dec2hex(head(2)); %读取版本号

fprintf('文件版本号为%s\n',ver);

n=head(4); %读取阶数

fprintf('矩阵A的阶数%d\n',n);

q=head(5); %上带宽

fprintf('矩阵A的上带宽%d\n',q);

p=head(6); %下带宽

fprintf('矩阵A的下带宽%d\n',p);

dist=4\*num;

fseek(file,dist,'bof'); %把句柄值转向第七个元素开头处

[A,count]=fread(file,inf,'float'); %读取二进制文件，获取系数矩阵

fclose(file); %关闭二进制头文件

tic; %计时开始

%%

%对非压缩带状矩阵求解

if ver=='102'

a=zeros(n,n);

%提取系数矩阵A

for i=1:n

for j=1:n

a(i,j)=A((i-1)\*n+j);

end

end

%提取矩阵b

b=zeros(n,1);

for i=1:n

b(i)=A(n\*n+i);

end

%列主元Gauss消去法

%消去过程

for k=1:n-1

m=k;

%寻找第k步消去的主元

for i=k+1:n

if abs(a(m,k))<abs(a(i,k))

m=i;

end

end

if a(m,k)==0

disp('An error occured')

return

end

%交换元素位置

for j=1:n

temp=a(k,j);

a(k,j)=a(m,j);

a(m,j)=temp;

temp=b(k);

b(k)=b(m);

b(m)=temp;

end

for i=k+1:n

a(i,k)=a(i,k)/a(k,k);

for j=(k+1):n

a(i,j)=a(i,j)-a(i,k)\*a(k,j);

end

b(i)=b(i)-a(i,k)\*b(k);

end

end

%回代过程

x=zeros(n,1);

x(n)=b(n)/a(n,n);

for k=n-1:-1:1

x(k)=(b(k)-sum(a(k,k+1:n)\*x(k+1:n)))/a(k,k);

end

%对压缩带状矩阵求解

elseif ver=='202'

m=p+q+1;

a=zeros(n,m);

%提取系数矩阵A

for i=1:n

for j=1:m

a(i,j)=A((i-1)\*m+j);

end

end

%提取矩阵b

b=zeros(n,1);

for i=1:n

b(i)=A(n\*m+i);

end

%列主元Gauss消去法

%消去过程

for k=1:(n-1)

if a(k,p+1)==0

disp('An error occured');

break;

end

length1=n;

if(k+p)<n

length1=k+p;

end

for i=(k+1):length1

a(i,(k+p-i+1))=a(i,(k+p-i+1))/a(k,p+1);

for j=(k+1):(k+q)

a(i,j+p-i+1)=a(i,j+p-i+1)-a(i,k+p-i+1)\*a(k,j+p-k+1);

end

b(i)=b(i)-a(i,k+p-i+1)\*b(k);

end

end

%回代过程

x=zeros(n,1);

x(n)=b(n)/a(n,p+1);

sum=0;

for k=(n-1):-1:1

sum=b(k);

length2=n;

if(k+q)<n

length2=k+q;

end

for j=(k+1):length2

sum=sum-a(k,j+p-k+1)\*x(j);

end

x(k)=sum/a(k,p+1);

sum=0;

end

end

%%

%输出方程组的解

disp('方程组的解为：')

x

toc; %计时结束

**计算结果：**

1. **对data20201文件进行求解，来验证程序对非压缩格式的稀疏矩阵解的正确性。**

文件标识符为id=C0A8708

文件版本号为ver=102

矩阵A的阶数n=20

矩阵A的上带宽q=3

矩阵A的下带宽p=3

方程组的解为：

1.314003 1.313997 1.314000 1.313999 1.314000 1.314001 1.314001 1.314001 1.313999 1.313998 1.314000 1.314001 1.314001 1.313999 1.313998 1.314002 1.314001 1.313999

1.313999 1.314001

时间已过 0.016224 秒。

计算结果都十分接近1.314，符合要求

1. **对data20202文件进行求解，来验证程序对压缩格式的稀疏矩阵解的正确性。**

文件标识符为id=C0A8708

文件版本号为ver=202

矩阵A的阶数n=20

矩阵A的上带宽q=3

矩阵A的下带宽p=3

方程组的解为：

1.314000 1.313999 1.313999 1.314000 1.314000 1.314000 1.314000 1.313999 1.314001

1.314001 1.313999 1.313999 1.314001 1.314002 1.313999 1.314001 1.313998 1.313999

1.314001 1.314002

时间已过 0.014677 秒。

计算结果都十分接近1.314，符合要求

所以，通过（1）（2）可以直到，程序能够正确的求解压缩和非压缩形式的稀疏矩阵，接下来就可以用程序去计算data20203和data20204了。

1. **对data20203文件进行求解。**

文件标识符为id=C0A8708

文件版本号为ver=102

矩阵A的阶数n=2048

矩阵A的上带宽q=4

矩阵A的下带宽p=4

因为解的数目实在太多了，这里只统计解的个数，而不列出每个解了。以下面的程序进行统计：

sum=0;

for i=1:n

if(abs(x(i)-2.078)<10^-5)

sum=sum+1;

end

end

fprintf('解的数目sum=%d',sum);

**得到的结果为：**

方程组的解为：

解的数目sum=2048

时间已过 145.551021 秒。

由此，可得data20203的稀疏矩阵解为2.078

1. **对data20204文件进行求解。**

文件标识符为id=C0A8708

文件版本号为ver=202

矩阵A的阶数n=51520

矩阵A的上带宽q=4

矩阵A的下带宽p=4

因为解的数目实在太多了，这里只统计解的个数，而不列出每个解了。以下面的程序进行统计：

sum=0;

for i=1:n

if(abs(x(i)-2.077)<10^-5)

sum=sum+1;

end

end

fprintf('解的数目sum=%d',sum);

**得到的结果为：**

方程组的解为：

解的数目sum=51520

时间已过 0.089041 秒。

由此，可得data20203的稀疏矩阵解为2.077

**实际问题求解：**

化学方程式表示化学反应中的消耗和产生的物质的量。配平化学方程式就是必须找出一组数使得方程式左右两端的各类原子的总数对应相等。一个方法就是建立能够描述反应过程中每种原子数目的向量方程，然后找出该方程组的最简的正整数解。下面利用此思路来配平如下化学反应方程式：

其中x1，x2,…,x6均为正整数。

上述化学反应式中包含5中不同的原子（K, Mn, O, S, H），于是在R5中为每一种反应物和生成物构成如下向量:

:[1 1 4 0 0]T : [0 1 4 1 0]T [0 0 1 0 2]T

[0 1 2 0 0]T [2 0 4 1 0]T [0 0 4 1 2]T

其中，每一个向量的各个分量依次表示反应物和生成物中K, Mn, O, S, H的原子数目。

为了配平化学方程式，系数x1，x2,…,x6必须满足方程组：

由此得到齐次线性方程组：

通过列主元的Gauss消去法，可以得到齐次线性方程组的通解：

由于化学方程式通常取最简的正整数，因此在通解中取c=1即得配平后的化学方程式：