1. 由于该题目取的节点为八个等距节点，在x=0拐点处没有取节点，因此所有插值函数曲线在**对称轴附近偏差较大**。下面对各插值曲线其他的偏差进行分别讨论

拉格朗日插值，在两个边缘处震荡很大。

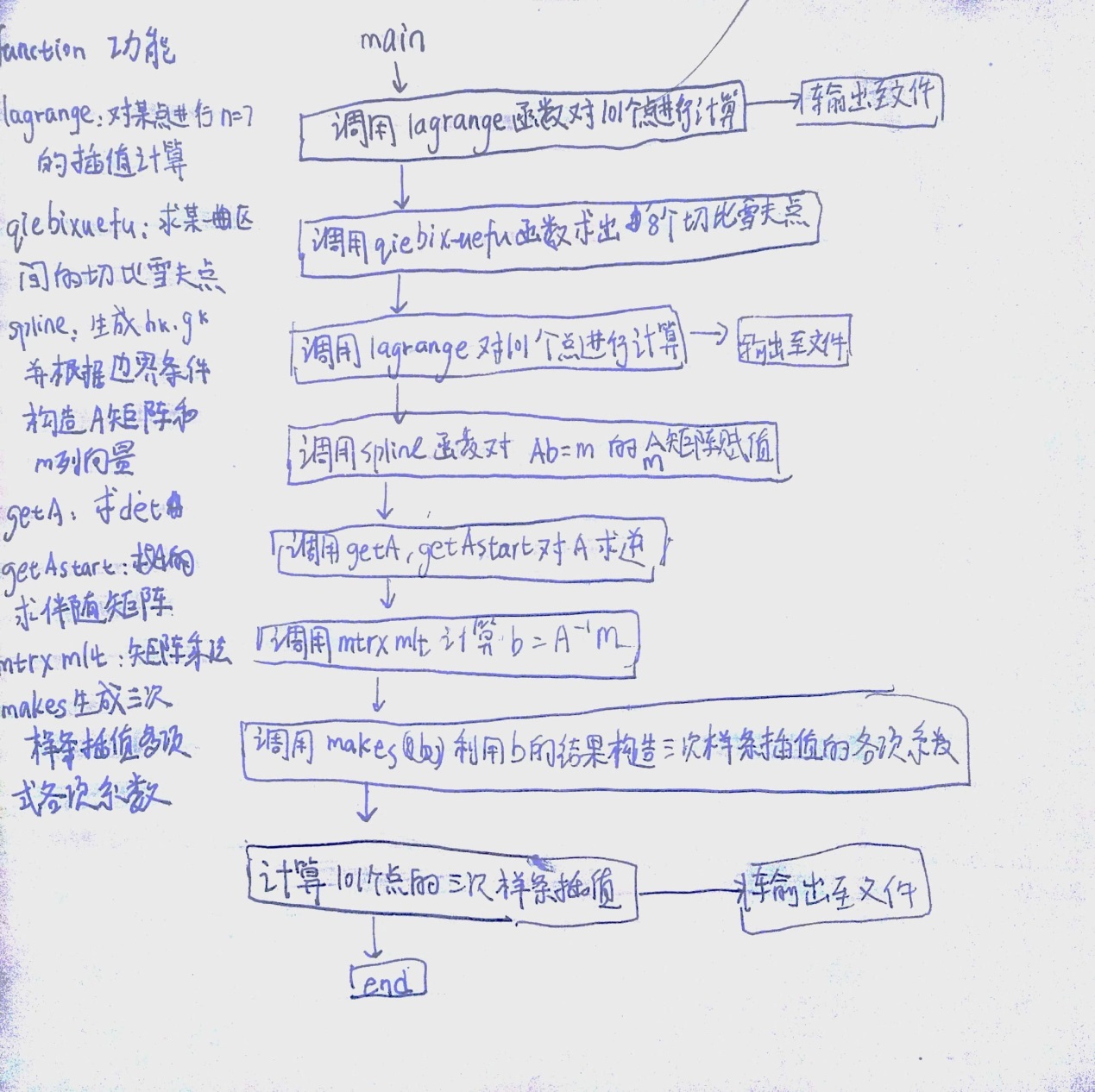
三次样条插值，逼近效果很好。

随后取了八个切比雪夫点，再通过拉格朗日进行插值，边缘震荡减小。

通过这个题目，我得到了进行插值的几点注意事项。

1. 为了更加准确，应将拐点等特殊点做为节点，这样可以保证函数大体的图像得以确定。若预计函数为偶函数，则最好取奇数个节点，这样可以保证在x=0处插值准确。
2. 选取切比雪夫点作为节点可以使得拉格朗日插值在边缘部分的震荡减小，提高精度。
3. 三次样条插值虽然更加精确，但是由于矩阵的介入使得计算量较大。由于做此题的时候还没有学到矩阵的计算，所以我采用的求解矩阵方程的方法是对方阵求逆，然后左右两边同时左乘方阵的逆，这样的方法由于阶数并不是特别大，时间复杂度导致的长时间计算略微有所体现。希望在学习完矩阵的有关计算的时候，可以对此实验的代码进行优化。

程序流程图如下：



此程序的运行结果输出至文件，所以并没有显式结果的截图

下图为函数图像，图像由matlab 读取运行结果的文件画出

代码为

A=xlsread('C:\lagrange.xlsx');

B=xlsread('C:\qiebixuefu.xlsx');

C=xlsread('C:\lab1.xlsx');

a1=A(:,1);

b1=A(:,2);

a2=B(:,1);

b2=B(:,2);

a3=C(:,1);

b3=C(:,2);

x=-5:0.1:5;

y=1./(1+x.\*x);

plot(x,y,'k');

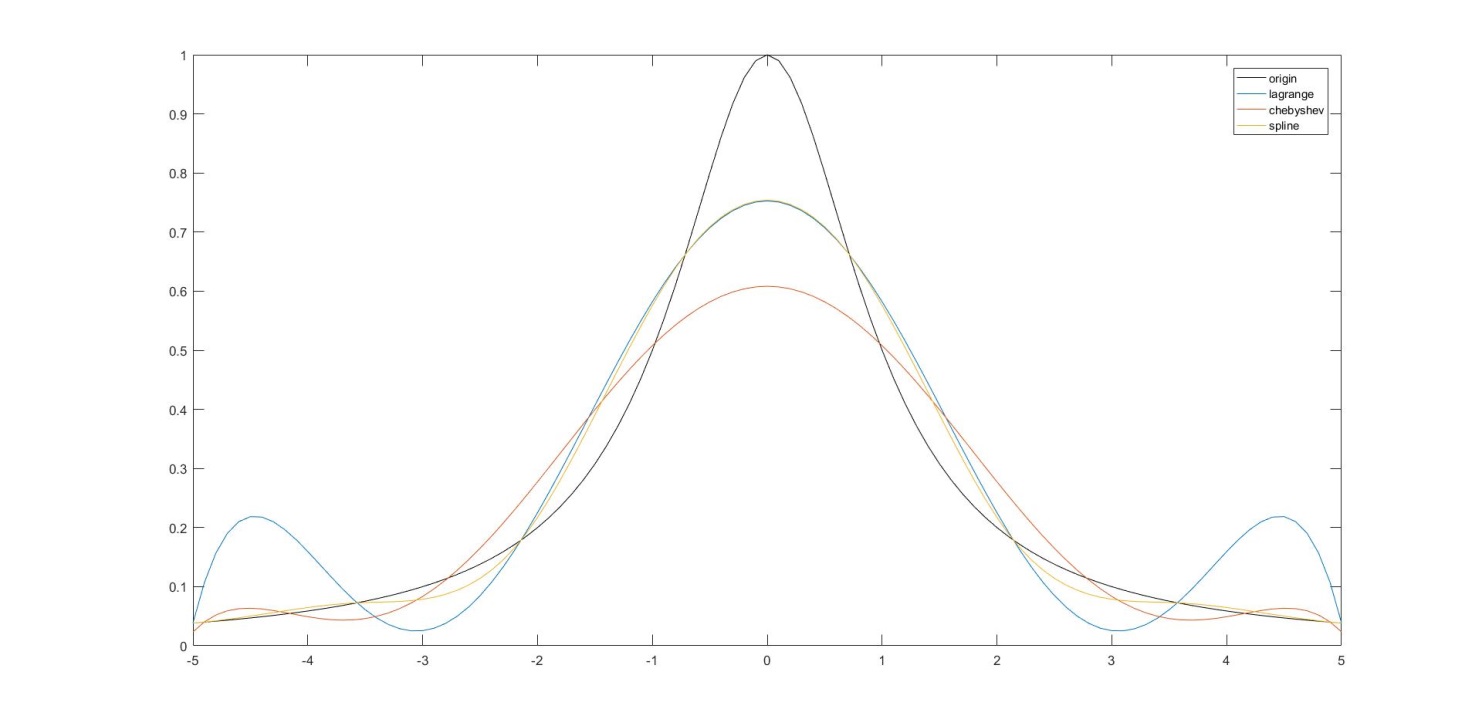
hold on;

plot(a1,b1);

plot(a2,b2);

plot(a3,b3);

legend('origin','lagrange','chebyshev','spline');



C语言实现代码如下（复制过来注释成了乱码，建议从附件lab1.cpp查看，用VS查看会报错，因为VS中的math.h并没有定义M\_PI变量）

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

#define \_USE\_MATH\_DEFINES

#define N 8

#define eps 1e-6

double a1[N]={-5.0,-5+10.0/7,-5+20.0/7,-5+30.0/7,5-30.0/7,5-20.0/7,5-10.0/7,5.0};

double q[N]={0};//ÓÃÓÚ±£´æÇÐ±ÈÑ©·ò²åÖµµã

double s[101]={0};//ÓÃÓÚ¼ÆËã²åÖµ½á¹ûµÄ±£´æ

double g[N]={0};//´æ·ÅgkµÄÖµ

double ans[N]={0};//¾ØÕó·½³ÌÓÒ±ß×ó³Ë×ó±ß·½ÕóµÄÄæ£¬½á¹û±£´æÖÁ´Ë£¬¼´ÎªM¾ØÕóµÄÖµ

double runge(double x);

double ss[N-1][4]={0};//´æ·ÅÈý´ÎÑùÌõ²îÖµ¶àÏîÊ½ÏµÊý

double h[N-1]={0};//´æ·Åhk

int DinV(double A[N][N]);

inline void swap(double &a,double &b){double c=a;a=b;b=c;};

void makes(double a[N-1][4],double an[])//Éú³É²åÖµº¯ÊýµÄÏµÊý

{

int i,j;

for (i=0;i<N;i++){

a[i][0]=(an[i+1]-an[i])/6/h[i];

a[i][1]=an[i]/2-3\*a[i][0]\*a1[i];

a[i][2]=((runge(a1[i+1])-runge(a1[i]))/h[i]-(2\*an[i]+an[i+1])/6\*h[i])\

+3\*a[i][0]\*a1[i]\*a1[i] -an[i]\*a1[i];

a[i][3]=runge(a1[i])-a1[i]\*((runge(a1[i+1])-runge(a1[i]))/h[i]-(2\*an[i]+an[i+1])/6\*h[i])\

+an[i]/2\*a1[i]\*a1[i]-a[i][0]\*pow(a1[i],3);

}

}

void mtrxmlt(double n[N][N],double m[N])//¾ØÕó³Ë·¨

{

int i=0,j=0;

for (i=0;i<N;i++)

for (j=0;j<N;j++)

ans[i]+=n[i][j]\*m[j];

}

double getA(double arcs[N][N],int n)//¾ØÕó:°´µÚÒ»ÐÐÕ¹¿ªµÝ¹é¼ÆËã|A|

{

if(n==1)

return arcs[0][0];

double ans = 0;

double temp[N][N];

int i,j,k;

for(i=0;i<n;i++){

for(j=0;j<n-1;j++)

for(k=0;k<n-1;k++)

temp[j][k] = arcs[j+1][(k>=i)?k+1:k];

double t = getA(temp,n-1);

if(i%2==0)

ans += arcs[0][i]\*t;

else

ans -= arcs[0][i]\*t;

}

return ans;

}

void getAStart(double arcs[N][N],int n,double ans[N][N])//¾ØÕó£º¼ÆËãÃ¿Ò»ÐÐÃ¿Ò»ÁÐµÄÃ¿¸öÔªËØËù¶ÔÓ¦µÄÓà×ÓÊ½£¬×é³ÉA\*

{

if(n==1){

ans[0][0] = 1;

return;

}

int i,j,k,t;

double temp[N][N];

for(i=0;i<n;i++){

for(j=0;j<n;j++){

for(k=0;k<n-1;k++)

for(t=0;t<n-1;t++)

temp[k][t] = arcs[k>=i?k+1:k][t>=j?t+1:t];

ans[j][i] = getA(temp,n-1);

if((i+j)%2 == 1)

ans[j][i] = - ans[j][i];

}

}

}

void spline(double mtrx[N][N]){

double lumda[N-1]={0},miu[N-1]={0};

int i=0,j=0;

while(i<N-1){

h[i]=a1[i+1]-a1[i];

i++;

}

for (i=1;i<N-1;i++){

lumda[i]=h[i]/(h[i]+h[i-1]);

miu[i]=h[i-1]/(h[i]+h[i-1]);

g[i]=6\*(runge(a1[i+1])-2\*runge(a1[i])+runge(a1[i-1]))/20\*7/10\*7;

}

g[0]=6/h[0]\*((runge(a1[1])-runge(a1[0]))/h[0]-10/pow(26,2));

g[7]=6\*(-(runge(a1[7])-runge(a1[6]))/h[6]-10/pow(26,2))/10\*7;

mtrx[0][1]=1;

mtrx[N-1][N-2]=1;

for(i=0;i<N;i++)

mtrx[i][i]=2;

for(i=1;i<N-1;i++){

mtrx[i][i-1]=miu[i];

mtrx[i][i+1]=lumda[i];

}

}

double qiebixuefu(double s,double e,double i,double n){

return (s+e)/2+(e-s)/2\*cos((2\*i+1)/(2\*n+2)\*M\_PI);

}

double runge(double x){

return 1/(1+pow(x,2));

}

double lagrange(double a,double t1[]){

int i=0,j=0;

double count=0,multi=1;

while(i<8){

j=0;

multi=1;

while(j<N){

if(i!=j)

multi\*=(a-t1[j])/(t1[i]-t1[j]);

j++;

}

multi\*=runge(t1[i]) ;

count+=multi;

i++;

}

return count;

}

int main(){

FILE \*fp;

int i=0;

/\*À­¸ñÀÊÈÕ²åÖµ\*/

while(i<101){

s[i]=lagrange(i\*0.1-5,a1);

i++;

}

fp=fopen("C:\\lagrange.txt","w+");

for (i=0;i<101;i++)

fprintf(fp,"%lf\t%lf\n",i\*0.1-5,s[i]);

fclose(fp);

for(i=0;i<8;i++)//ÇóÇÐ±ÈÑ©·òµã

q[i]=qiebixuefu(-5,5,i,7);

for (i=0;i<101;i++)//ÇÐ±ÈÑ©·òÀ­¸ñÀÊÈÕ²åÖµ

s[i]=lagrange(i\*0.1-5,q);

fp=fopen("C:\\qiebixuefu.txt","w+");

for (i=0;i<101;i++)

fprintf(fp,"%lf\t%lf\n",i\*0.1-5,s[i]);

fclose(fp);

/\* Èý´ÎÑùÌõ²åÖµ\*/

double arcs[N][N]={0};//ÒªÇóÄæµÄ¾ØÕó

double astar[N][N]={0};//±£´æ°éËæ¾ØÕó

double m[N]={0};

int j;

spline(arcs) ;

double a = getA(arcs,N);//¿ªÊ¼ÇóÄæ¾ØÕó£¬½«½á¹û±£´æÖÁinverse

if(abs(a-0)<eps)

printf("can not transform!\n");

else

getAStart(arcs,N,astar);

double inverse[N][N];

for(i=0;i<N;i++)

for(j=0;j<N;j++)

inverse[i][j]=astar[i][j]/a;

mtrxmlt(inverse,g);

makes(ss,ans);

double ii=0;

for(i=0;i<101;i++)

{

s[i]=0;

ii=-5+i\*0.1;

for(j=0;j<N-1;j++){

if(a1[j]<=ii&&ii<=a1[j+1]){

s[i]=ss[j][0]\*pow(ii,3)+ss[j][1]\*pow(ii,2)+ss[j][2]\*ii+ss[j][3];

break;

}

}

}

fp=fopen("C:\\lab1.txt","w+");

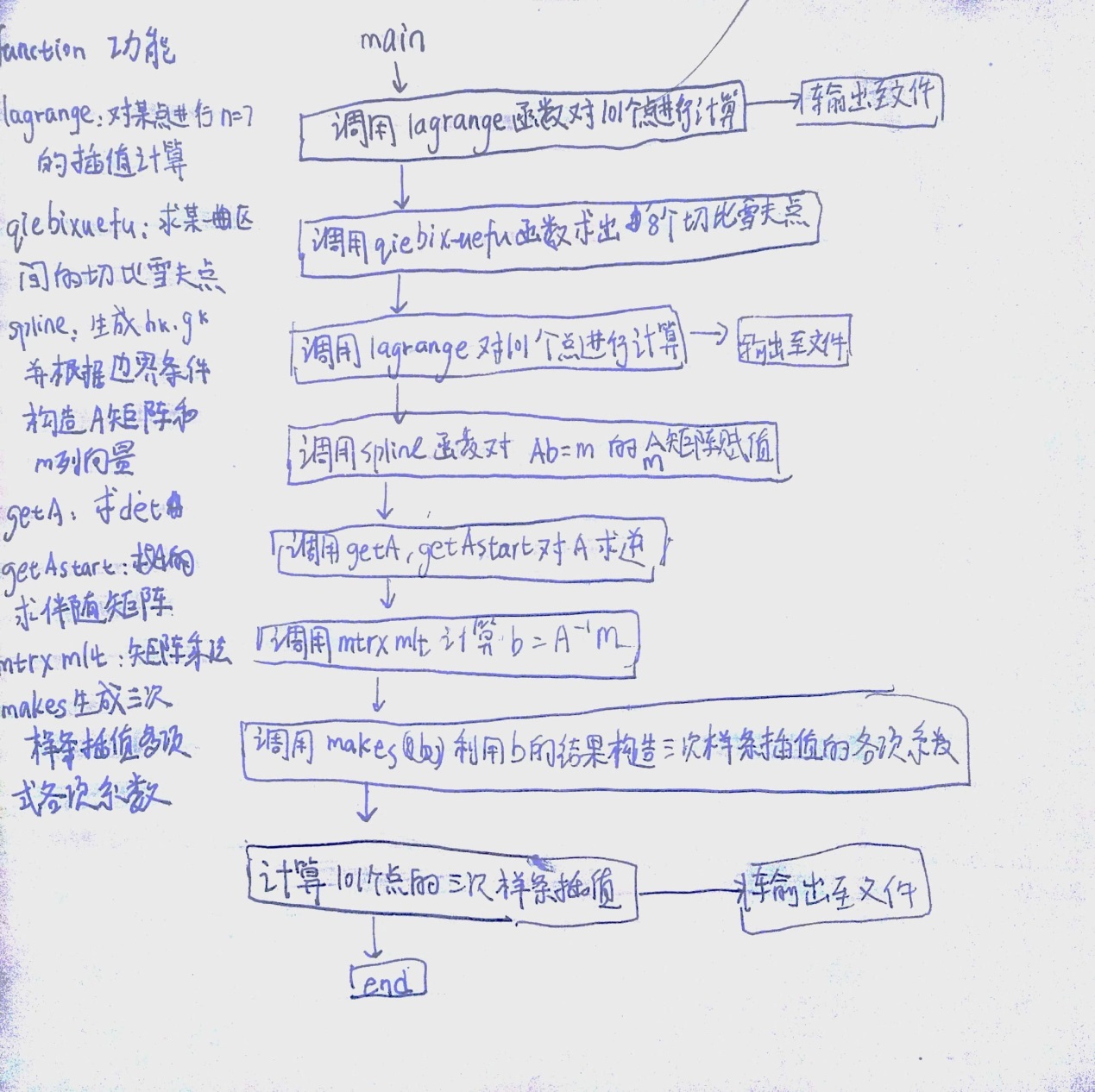
for(i=0;i<101;i++)

fprintf(fp,"%lf\t%lf\n",-5+0.1\*i,s[i]);

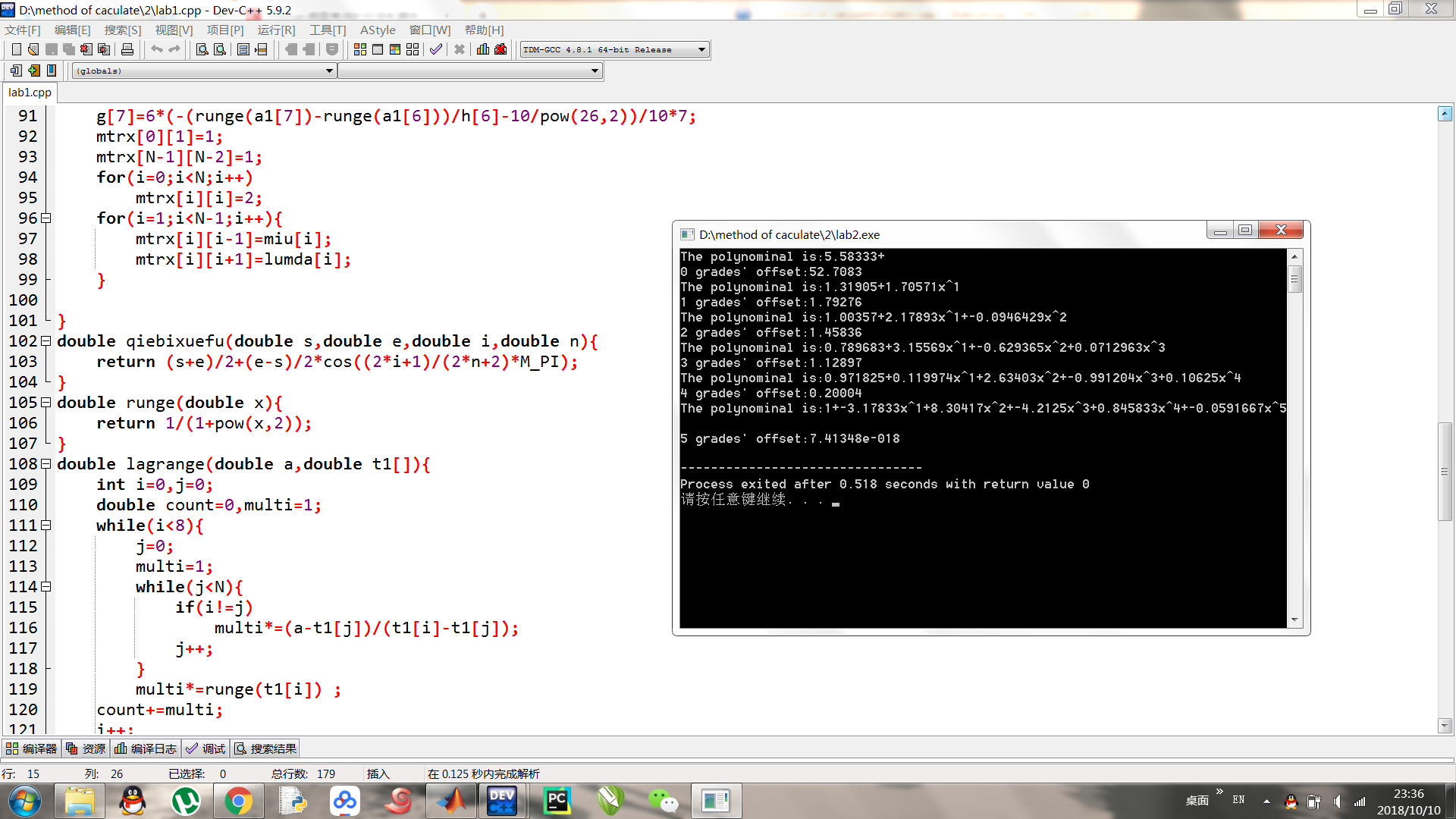
fclose(fp);

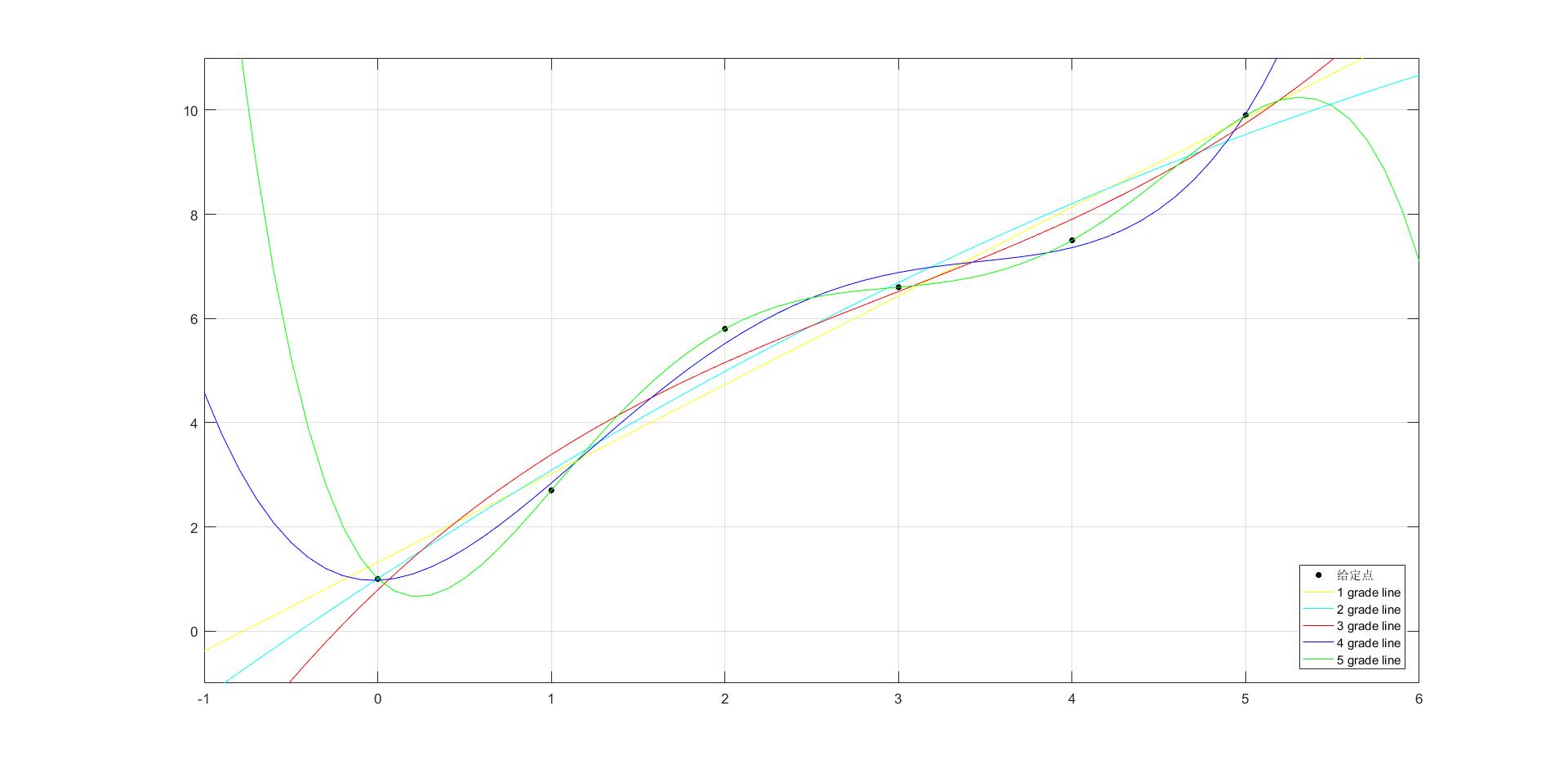
}

2.该题通过多项式来拟合几个三点的函数曲线，分别对n=1,2,3,4,5,进行拟合

流程图如下

由程序运行结果可得每一拟合曲线的方程和对应误差，可知5阶多项式拟合的误差最小





原始点和拟合曲线图由MATLAB画出，程序代码如下(部分汉字复制来变成乱码)

由于水平有限，0阶拟合直线没有画出，应为y=5.58的水平直线

close all;clc

A=xlsread('D:\method of caculate\2\lab2.xlsx');

a=A(1:6,:);

plot(a(:,1:1),a(:,2:2),'k.','Markersize',15);

axis([-1 6 -1 11]);

hold on;

x=-1:0.1:6;

y0=5.58;

y1=1.31905+1.70571\*x;

y2=1.00357+2.17893\*x.^1-0.0946429\*x.^2;

y3=0.789683+3.15569\*x.^1-0.629365\*x.^2+0.0712963\*x.^3;

y4=0.971825+0.119974\*x.^1+2.63403\*x.^2-0.991204\*x.^3+0.10625\*x.^4;

y5=1-3.17833\*x.^1+8.30417\*x.^2-4.2125\*x.^3+0.845833\*x.^4-0.0591667\*x.^5;

plot(x,y1,'y');

plot(x,y2,'c');

plot(x,y3,'r');

plot(x,y4,'b');

plot(x,y5,'g');

legend({'¸ø¶¨µã','1 grade line','2 grade line'...

,'3 grade line','4 grade line','5 grade line'},'Location','SouthEast');

grid on;

实验二的C语言实现为（注释复制过来成了乱码，建议从附件lab2.cpp查看）

#include<iostream>

#include<iomanip>

#include<cmath>

#define N 6

using namespace std;

double xy[N][2]={0,1,1,2.7,2,5.8,3,6.6,4,7.5,5,9.9};

void makematrix(int n);

double sumright(double a[N][2],double i);

double sumleft(double a[N][2],double i);

void mtrxmlt(double n[N][N],double m[N]);

double getA(double arcs[N][N],int n);

void getAStart(double arcs[N][N],int n,double ans[N][N]);

void mtrxmlt(double n[N][N],double m[N],double ans[]);

double offset(double a[],int n);

main(){

int i=1;

while(i<N+1){

makematrix(i);//µ÷ÓÃ¸Ãº¯Êý Çó²»Í¬½×ÊýµÄÄâºÏº¯Êý

i++;

}

}

double offset(double a[],int n){

double num[6]={0},dvt=0;

int i=0,j=0;

while(j<6){//ÇóÃ¿Ò»µãµÄÄâºÏÖµ

for(i=0;i<n;i++){

num[j]+=a[i]\*pow((double)j,i);

}

//cout<<j<<" "<<setw(8)<<num[j]<<endl;;

j++;

}

i=0;

while(i<6){//¶ÔÃ¿Ò»µã²úÉúµÄÎó²îÆ½·½ÇóºÍ

dvt+=pow((num[i]-xy[i][1]),2);

i++;

}

return dvt;

}

void makematrix(int n){

/\*¹¹Ôìn½×¾ØÕó·½³Ì\*/

double sqare[N][N]={0},vector[N]={0};

sqare[0][0]=N;

int a,b;

for(a=0;a<n;a++){

for(b=a;b<n;b++){

sqare[a][b]=sumleft(xy,a+b);

sqare[b][a]=sqare[a][b];

}

vector[a]=sumright(xy,a);

}

/\*½â·½³Ì£¬·½³ÌÁ½±ß×ó³Ë·½ÕóµÄÄæ\*/

double inverse[N][N]={0},cmp[N][N]={0},ans[N]={0};//ans´æ·ÅÇó½â³öµÄÁÐÏòÁ¿

getAStart(sqare,n,cmp);

double det=getA(sqare,n);

for(a=0;a<n;a++){

for(b=0;b<n;b++){

inverse[a][b]=cmp[a][b]/det;

}

}

mtrxmlt(inverse,vector,ans);

/\*ÇóÎó²î\*/

double result=offset(ans,n);

cout<<"The polynominal is:";

cout<<ans[0]<<"+";

for(a=1;a<n;a++)

a==n-1?cout<<ans[a]<<"x^"<<a:cout<<ans[a]<<"x^"<<a<<"+";

cout<<endl;

cout<<n-1<<" grades' offset:"<<result<<endl;

}

double sumleft(double a[N][2],double i){//ÇóºÍº¯Êý£¬½á¹ûÊÇ·½³Ì×ó±ßµÄ·½ÕóµÄÔªËØ

int j;

double val=0;

for(j=0;j<N;j++)

val+=pow(xy[j][0],i);

return val;

}

double sumright(double a[N][2],double i){//ÇóºÍº¯Êý£¬½á¹ûÊÇ·½³ÌÓÒ±ßµÄÁÐÏòÁ¿µÄÔªËØ

int j;

double val=0;

for(j=0;j<N;j++)

val+=pow(xy[j][0],i)\*xy[j][1];

return val;

}

void mtrxmlt(double n[N][N],double m[N],double ans[])//¾ØÕó³Ë·¨

{

int i=0,j=0;

for (i=0;i<N;i++)

for (j=0;j<N;j++)

ans[i]+=n[i][j]\*m[j];

}

double getA(double arcs[N][N],int n)//¾ØÕó:°´µÚÒ»ÐÐÕ¹¿ªµÝ¹é¼ÆËã|A|

{

if(n==1)

return arcs[0][0];

double ans = 0;

double temp[N][N];

int i,j,k;

for(i=0;i<n;i++){

for(j=0;j<n-1;j++)

for(k=0;k<n-1;k++)

temp[j][k] = arcs[j+1][(k>=i)?k+1:k];

double t = getA(temp,n-1);

if(i%2==0)

ans += arcs[0][i]\*t;

else

ans -= arcs[0][i]\*t;

}

return ans;

}

void getAStart(double arcs[N][N],int n,double ans[N][N])//¾ØÕó£º¼ÆËãÃ¿Ò»ÐÐÃ¿Ò»ÁÐµÄÃ¿¸öÔªËØËù¶ÔÓ¦µÄÓà×ÓÊ½£¬×é³ÉA\*

{

if(n==1){

ans[0][0] = 1;

return;

}

int i,j,k,t;

double temp[N][N];

for(i=0;i<n;i++){

for(j=0;j<n;j++){

for(k=0;k<n-1;k++)

for(t=0;t<n-1;t++)

temp[k][t] = arcs[k>=i?k+1:k][t>=j?t+1:t];

ans[j][i] = getA(temp,n-1);

if((i+j)%2 == 1)

ans[j][i] = - ans[j][i];

}

}

}