- 实验报告

姓名	
评分	

实验报告

课程名称：数值分析

课题名称：插值算法的实验与分析

专　　业：地球物理

姓　　名：王锴

班　　级：201145

完成日期：2016 年 11 月 13 日

目录

实验报告

一、实验名称

插值算法的实验与分析。

二、实验目的

（1） 认识高次插值的龙格现象；
（2） 掌握拉格朗日插值、牛顿插值、两点三次埃尔米特插值、分段线性插值及三次样条插值算法；
（3） 培养编程与上机调试能力；
（4） 熟悉 Matlab 软件环境。

三、实验要求

（1） 通过选取不同的函数进行多项式插值，分析高次插值的龙格现象；
（2） 通过编程实现拉格朗日插值、牛顿插值、两点三次埃尔米特插值、分段线性插值和三次样条插值，并分析它们的优缺点。

四、实验原理

（1） 龙格现象：

插值多项式余项公式

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

说明余项的大小既与插值节点的个数有关，也与函数$f(x)$的高阶导数有关。当插值多项式的次数变高时，有可能提高结果的准确程度，也有可能使误差更大，这时插值函数的图像在两端点处会出现激烈的震荡，这就是龙格现象。

（2） 拉格朗日插值算法：

两个插值点可求出一次插值多项式,而三个插值点可求出二次插值多项式。那么 n+1 个不同的节点可以构造一个次数不超过 n 的多项式函数，令

$$l_i(x)=\sigma_{ki}=\begin{cases}1,(i=k)\\0,(i\neq k)\end{cases},i=0,1,2\ldots n \quad (4.1)$$

$$L_n(x)=\sum_{k=0}^{n}l_k(x)\bullet y(x) \quad (4.2)$$

解（4.1）式得

$$l_n(x)=\frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} \quad (4.3)$$

则拉格朗日插值多项式为

$$L_n(x)=\sum_{k=0}^{n}\frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}\bullet y(x) \quad (4.4)$$

（3） 牛顿插值算法：

由线性代数知,任何一个不高于 n 次的多项式，都可以表示成函数

$$1,x-x_0,(x-x_0)(x-x_1),\dots,(x-x_0)(x-x_1)\dots(x-x_{n-1})$$

的线性组合，也就是说，可以把满足插值条件插值多项式，写成如下形式

$$N_n(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}) \quad (4.5)$$

解得

$$a_k=f[x_0,x_1,\dots,x_k]=\frac{f^{(k)}(\xi)}{k!} \quad (4.6)$$

这样求得的 $N_n(x)$ 就是牛顿插值多项式。

（4） 两点三次埃尔米特插值算法：

在许多实际应用中，不仅要求函数值相等，而且要求若干阶导数也相等，如机翼设计等。满足函数值相等且导数也相等的插值方法成为埃尔米特插值。其中两点三次埃尔米特插值是最为简单的一种类型。模仿拉格朗日插值多项式的思想，设

$$H_3(x)=y_0\alpha_0(x)+y_1\alpha_1(x)+m_0\beta_0(x)+m_1\beta_1(x) \quad (4.7)$$

其中$\alpha_0(x)$，$\alpha_1(x)$，$\beta_0(x)$，$\beta_1(x)$均为三次多项式，且满足

$$\begin{cases}\alpha_j(x_i)=\sigma_{ji},\ \alpha_j'(x_i)=0,\\ \beta_j(x_i)=0,\ \beta_j'(x_i)=\sigma_{ji},\ \mathrm{i,j}=0,1\end{cases} \quad (4.8)$$

解得

$$\begin{cases}\alpha_0(x)=\left(1+2\dfrac{x-x_0}{x_1-x_0}\right)\left(\dfrac{x-x_1}{x_0-x_1}\right)^2\\ \alpha_1(x)=\left(1+2\dfrac{x-x_1}{x_0-x_1}\right)\left(\dfrac{x-x_0}{x_1-x_0}\right)^2\\ \beta_0(x)=(x-x_0)\left(\dfrac{x-x_1}{x_0-x_1}\right)^2\\ \beta_1(x)=(x-x_1)\left(\dfrac{x-x_0}{x_1-x_0}\right)^2\end{cases} \quad (4.9)$$

这样求得的 $H_3(x)$ 就是两点三次埃尔米特插值多项式。

（5） 分段线性插值算法：

分段线性插值就是通过插值节点用折线段连接起来逼近$f(x)$。在几何上就是用折线替代曲线。容易得到

$$S(x)=\sum_{i=0}^{n}l_i(x)f(x_i) \quad (4.10)$$

其中

$$l_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{x_i-x_{i-1}}, & x_{i-1}\le x\le x_i\\ \dfrac{x-x_{i+1}}{x_i-x_{i+1}}, & x_i\le x\le x_{i+1}\\ 0, & x\epsilon[a,x_{i-1}]\cup[x_i,b]\end{cases}\quad(4.11)$$

这样求得的 $S(x)$ 就是分段线性插值多项式。

（6） 三次样条插值算法：

三次样条插值在每两个相邻节点之间都是不超过三次的多项式函数并且在整个定义域内存在连续的二阶导数。经过推导可以得到

$$S_i(x)=M_{i-1}\frac{(x_i-x)^3}{6h_i}+M_i\frac{(x-x_{i-1})^3}{6h_i}+\left(y_{i-1}-\frac{M_{i-1}}{6}h_i^2\right)\frac{(x_i-x)}{h_i}+\left(y_i-\frac{M_i}{6}h_i^2\right)\frac{(x-x_{i-1})}{h_i}\quad(4.12)$$

其中

$$\begin{cases}\mu_iM_{i-1}+2M_i+\lambda_iM_{i+1}=g_i\\ \mu_i=\dfrac{h_i}{h_i+h_{i+1}}\\ \lambda_i=1-\mu_i\\ g_i=\dfrac{6}{h_i+h_{i+1}}(f[x_i,x_{i+1}]-f[x_{i-1},x_i])\end{cases}\quad(4.13)$$

在第一类边界条件下

$$S'(x_0)=f'(x_0),S'(x_n)=f'(x_n)$$

第二类边界条件下

$$S''(x_0)=f''(x_0),S''(x_n)=f''(x_n)$$

可以解出M_i，从而解得三次样条插值多项式$S_i(x)$。

▪ 五、实验题目

（1） 选取不同的函数分析高次插值的龙格现象。

（2） 编程实现拉格朗日插值、牛顿插值、两点三次埃尔米特插值、分段线性插值和三次样条插值，并进行对比。

▪ 六、实验步骤

（1） 龙格现象：

① 定义插值节点数 n1，n2，计算得到插值节点 X1，X2，计算绘图时使用的横坐标 Xi；

② 用选定函数计算插值节点的值 Y1，Y2，同时计算原函数在 Xi 处的函数值 YY；

③ 调用函数 Lagrange()计算插值 Yi1，Yi2；

④ 在同一副图中绘制插值多项式曲线 Yi1，Yi2，及原函数曲线 YY。

（2） 拉格朗日插值：

① 编写 Lagrange()插值多项式函数，输入 x 为向量，表示全部的插值节点；y 为向量，表示插值节点处的函数值；xi 为向量，表示被估计函数自变量；yi 为向量，表示 xi 处的函数估计值。若 x 与 y 不等长，则报错；若 x 元素有重复，则报错；
② 定义插值节点 x，节点函数值 y，待求点 xi，绘制 x、y 散点图；
③ 调用函数 Lagrange()计算 yi，并输出；
④ 计算绘图横坐标 xi，调用函数 Lagrange()计算绘图纵坐标 yi 及原函数对应函数值 yy，在同一副图中绘制插值多项式曲线和原函数曲线。

（3） 牛顿插值：

① 编写 Newton()插值多项式函数，输入 X 为向量，表示全部的插值节点；Y 为向量，表示插值节点处的函数值；xi 为向量，表示被估计函数自变量；yi 为向量，表示 xi 处的函数估计值。若 X 与 Y 不等长，则报错；若 X 元素有重复，则报错；
② 定义插值节点 X，节点函数值 Y，待求点 xi，绘制 X、Y 散点图；
③ 调用函数 Newton()计算 yi，并输出；
④ 计算绘图横坐标 Xi，调用函数 Newton()计算绘图纵坐标 Yi 及原函数对应函数值 YY，在同一副图中绘制插值多项式曲线和原函数曲线；
⑤ 编写实时输入响应 GUI 界面，重复步骤①-④，以分析牛顿插值的承袭性。

（4） 两点三次埃尔米特插值：

① 编写两点三次 Hermite()插值多项式函数，输入 X 为向量，表示全部的插值节点；Y 为向量，表示插值节点处的函数值；M 为向量，表示插值点处的导数值；xi 为向量，表示被估计函数自变量；yi 为向量，表示 xi 处的函数估计值。若 X、Y、M 元素个数不为 2，则报错；若 X 元素有重复，则报错；
② 定义插值节点 X，节点函数值 Y，节点导数值 M，待求点 xi，绘制 X、Y 散点图；
③ 调用函数 Hermite()计算 yi，并输出；
④ 计算绘图横坐标 Xi，调用函数 Hermite()计算绘图纵坐标 Yi 及原函数对应函数值 YY，在同一副图中绘制插值多项式曲线和原函数曲线；

（5） 分段线性插值：

① 编写 fenDuan()插值多项式函数，输入 X 为向量，表示全部的插值节点；Y 为向量，表示插值节点处的函数值；xi 为向量，表示被估计函数自变量；yi 为向量，表示 xi 处的函数估计值。若 X 与 Y 不等长，则报错；若 X 元素有重复，则报错；
② 定义插值节点 X，节点函数值 Y，待求点 xi，绘制 X、Y 散点图；
③ 调用函数 fenDuan()计算 yi，并输出；
④ 计算绘图横坐标 Xi，调用函数 fenDuan()计算绘图纵坐标 Yi 及原函数对应函数值 YY，在同一副图中绘制插值多项式曲线和原函数曲线；

（6） 三次样条插值：

① 编写 Spline()插值多项式函数，输入 X 为向量，表示全部的插值节点；Y 为向量，表示插值节点处的函数值；M 为向量，表示边界条件，包含两个数值；xi 为向量，表示被估计函数自变量；s 为整数，1 表示第一类边界条件，2 表示第二类边界条件；yi 为向量，表示 xi 处的函数估计值。若 X 与 Y 不等长，则报错；若 M 元素个数不为 2，则报错；若 X 元素有重复，则报错；

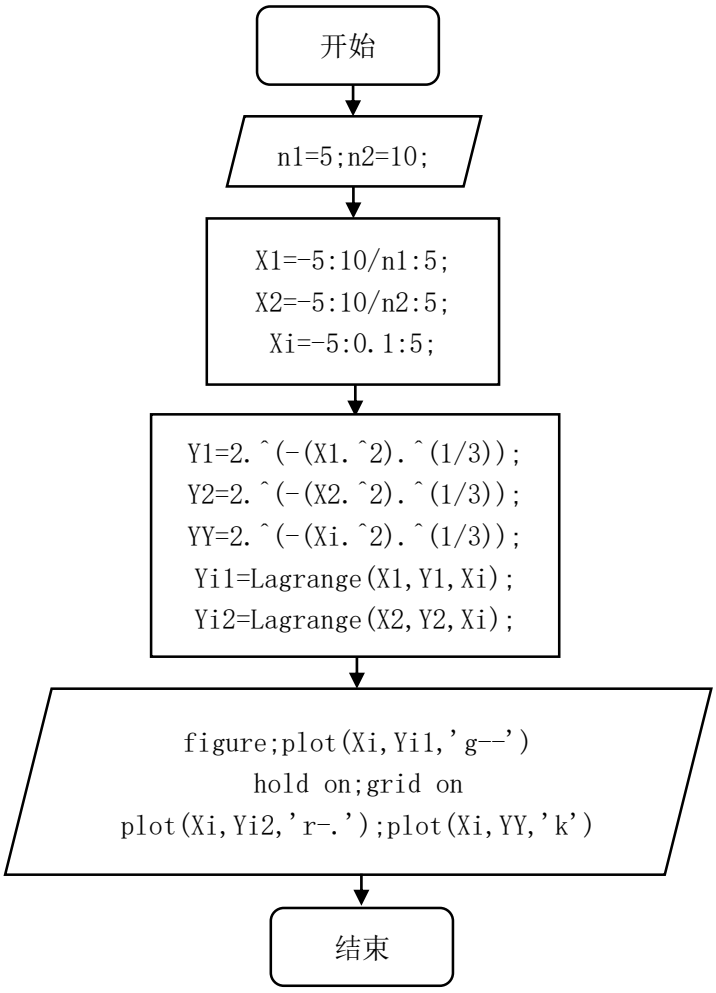
② 定义插值节点 X，节点函数值 Y，边界条件 M，待求点 xi，绘制 X、Y 散点图；

③ 调用函数 Spline()计算 yi，并输出；

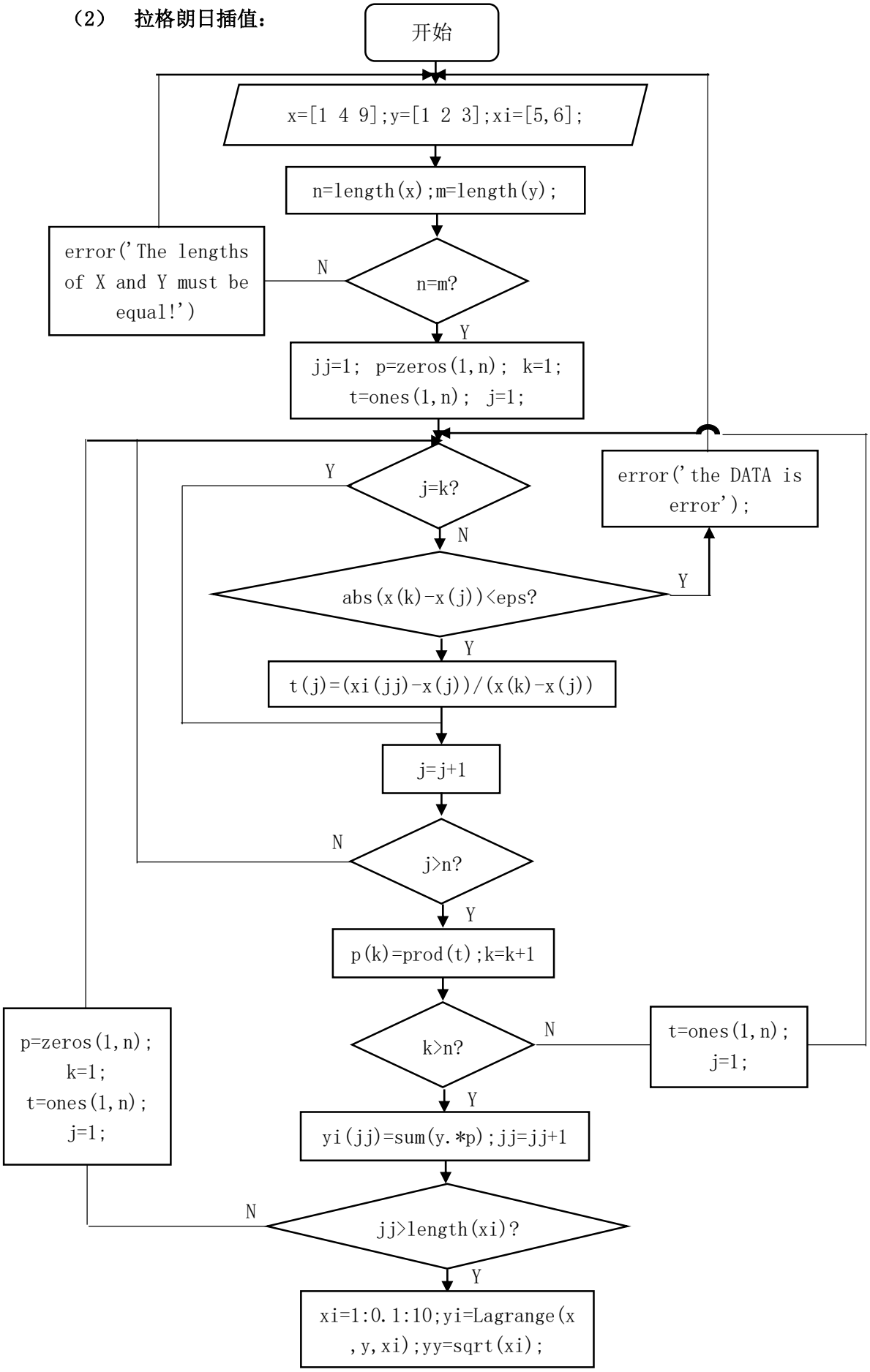
④ 计算绘图横坐标 Xi，调用函数 Spline()计算绘图纵坐标 Yi 及原函数对应函数值 YY，在同一副图中绘制插值多项式曲线和原函数曲线；

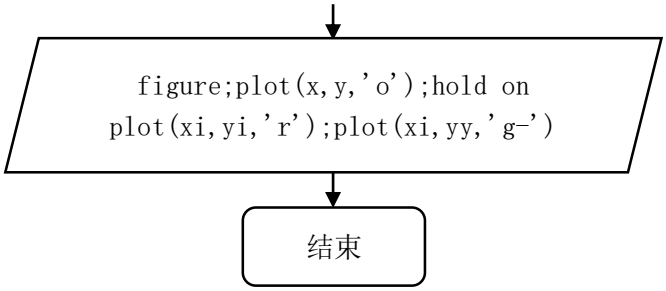
▪ 七、实验整体流程图或算法

（1） 龙格现象：

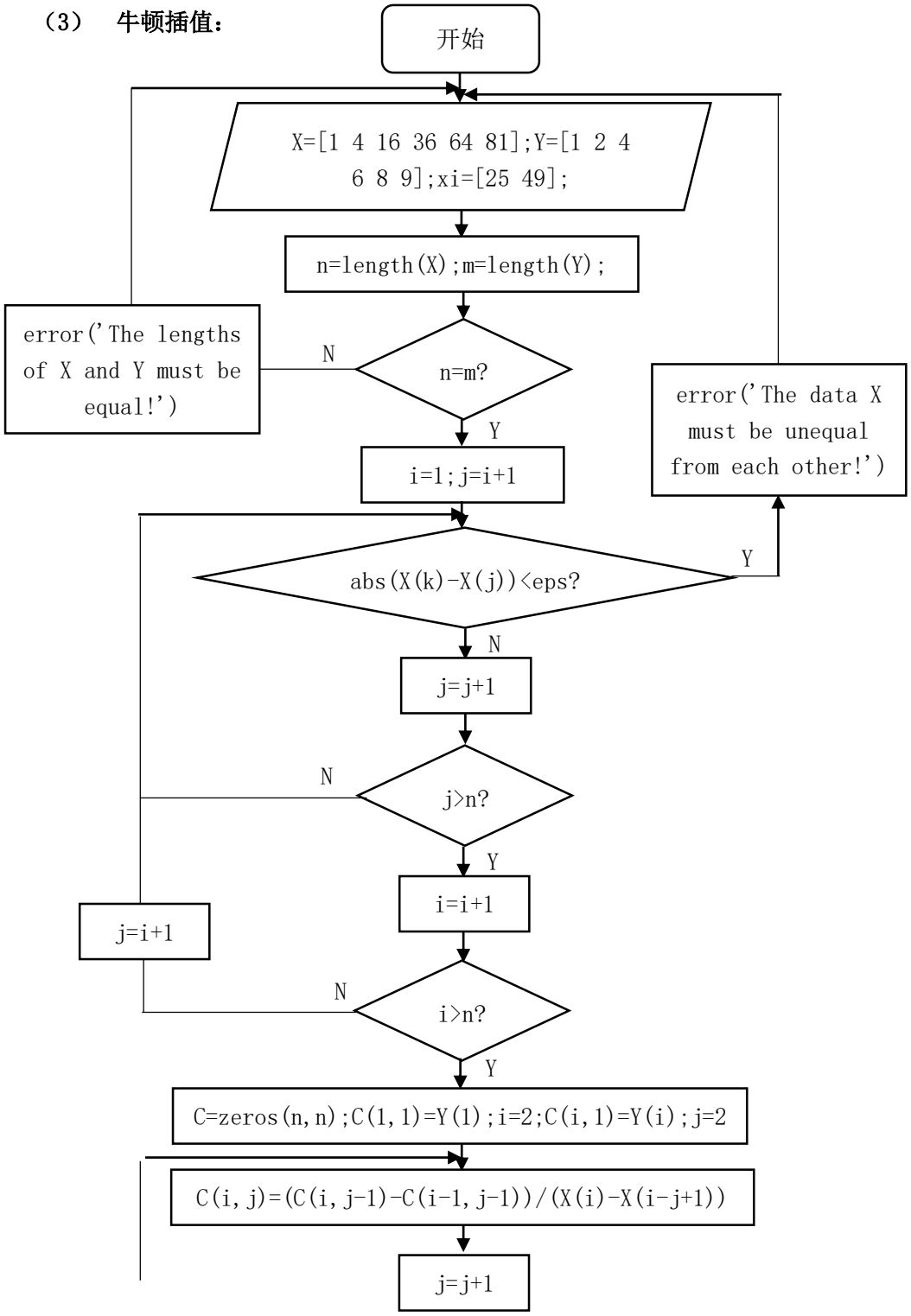


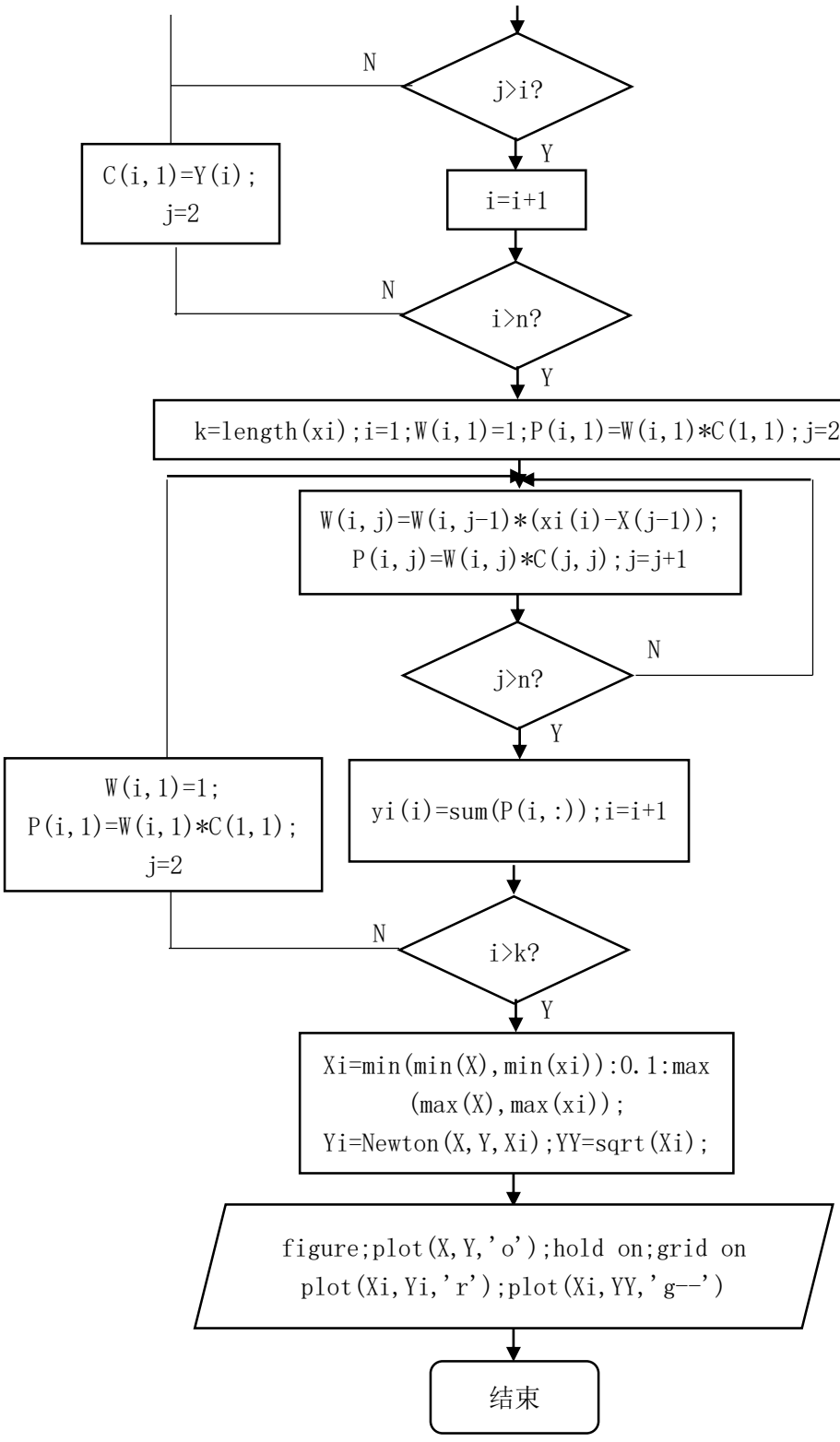
（2） 拉格朗日插值：


开始
x=[1 4 9];y=[1 2 3];xi=[5,6];
n=length(x);m=length(y);
n=m?
N
error('The lengths of X and Y must be equal!')
Y
jj=1; p=zeros(1,n); k=1;
t=ones(1,n); j=1;
j=k?
Y
N
error('the DATA is error');
abs(x(k)-x(j))<eps?
Y
Y
t(j)=(xi(jj)-x(j))/(x(k)-x(j))
j=j+1
j>n?
N
Y
p(k)=prod(t);k=k+1
k>n?
N
t=ones(1,n);
j=1;
Y
yi(jj)=sum(y.*p);jj=jj+1
jj>length(xi)?
N
Y
p=zeros(1,n);
k=1;
t=ones(1,n);
j=1;
xi=1:0.1:10;yi=Lagrange(x,y,xi);yy=sqrt(xi);

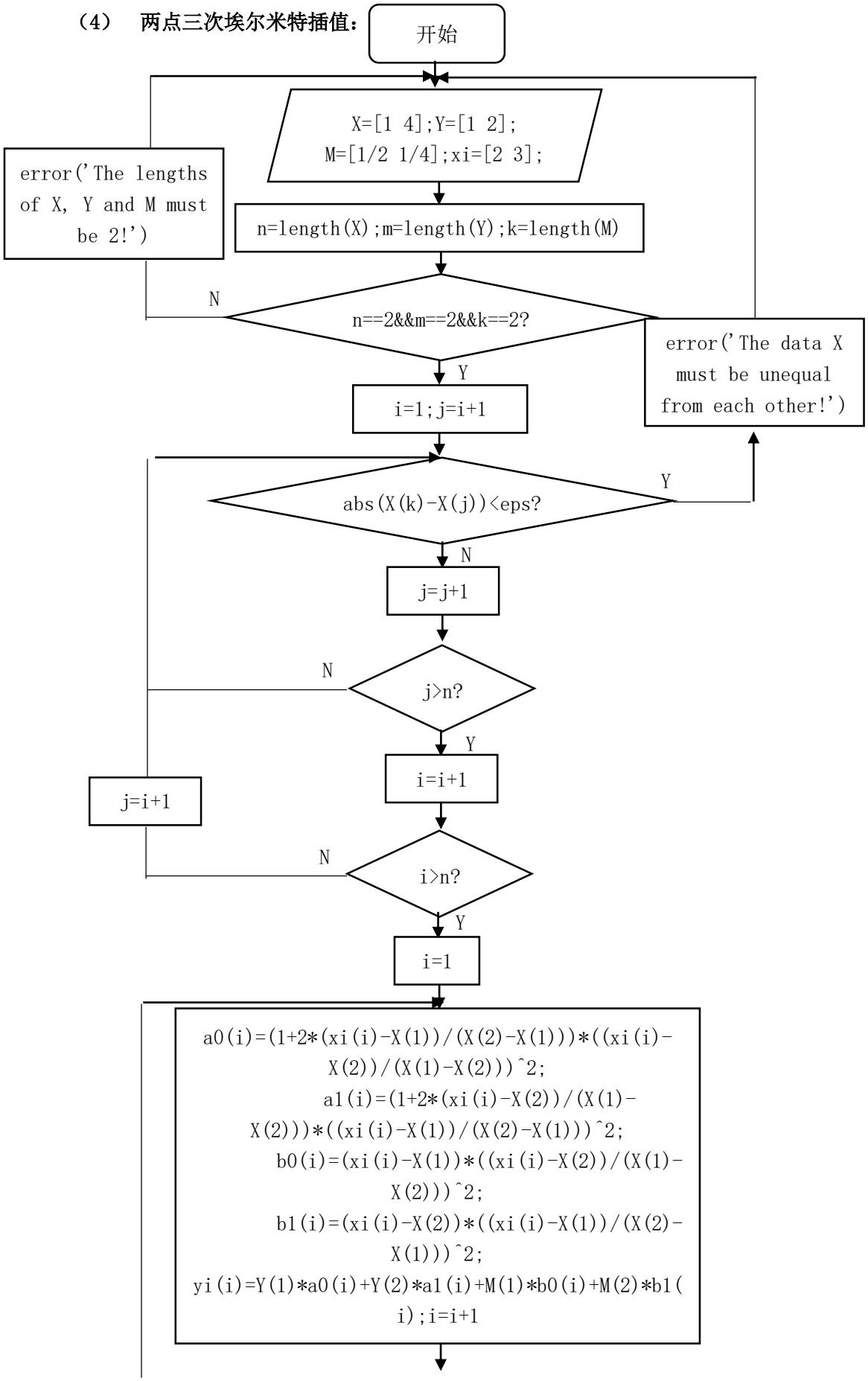

figure;plot(x,y,'o');hold on
plot(xi,yi,'r');plot(xi,yy,'g-')
结束

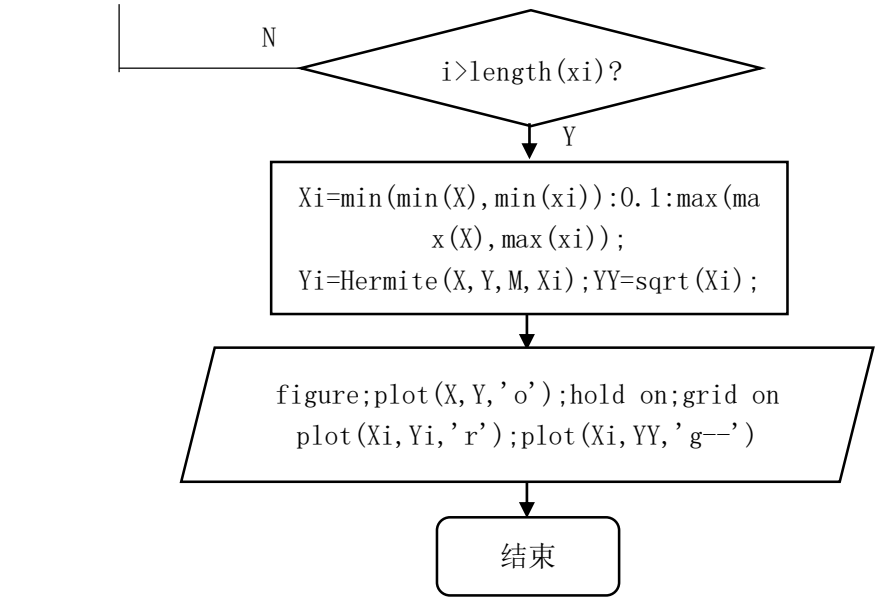
（3） 牛顿插值：


开始
X=[1 4 16 36 64 81];Y=[1 2 4 6 8 9];xi=[25 49];
n=length(X);m=length(Y);
n=m?
N
error('The lengths of X and Y must be equal!')
Y
i=1;j=i+1
abs(X(k)-X(j))<eps?
Y
error('The data X must be unequal from each other!')
N
j=j+1
j>n?
N
Y
i=i+1
j=i+1
i>n?
N
Y
C=zeros(n,n);C(1,1)=Y(1);i=2;C(i,1)=Y(i);j=2
C(i,j)=(C(i,j-1)-C(i-1,j-1))/(X(i)-X(i-j+1))
j=j+1

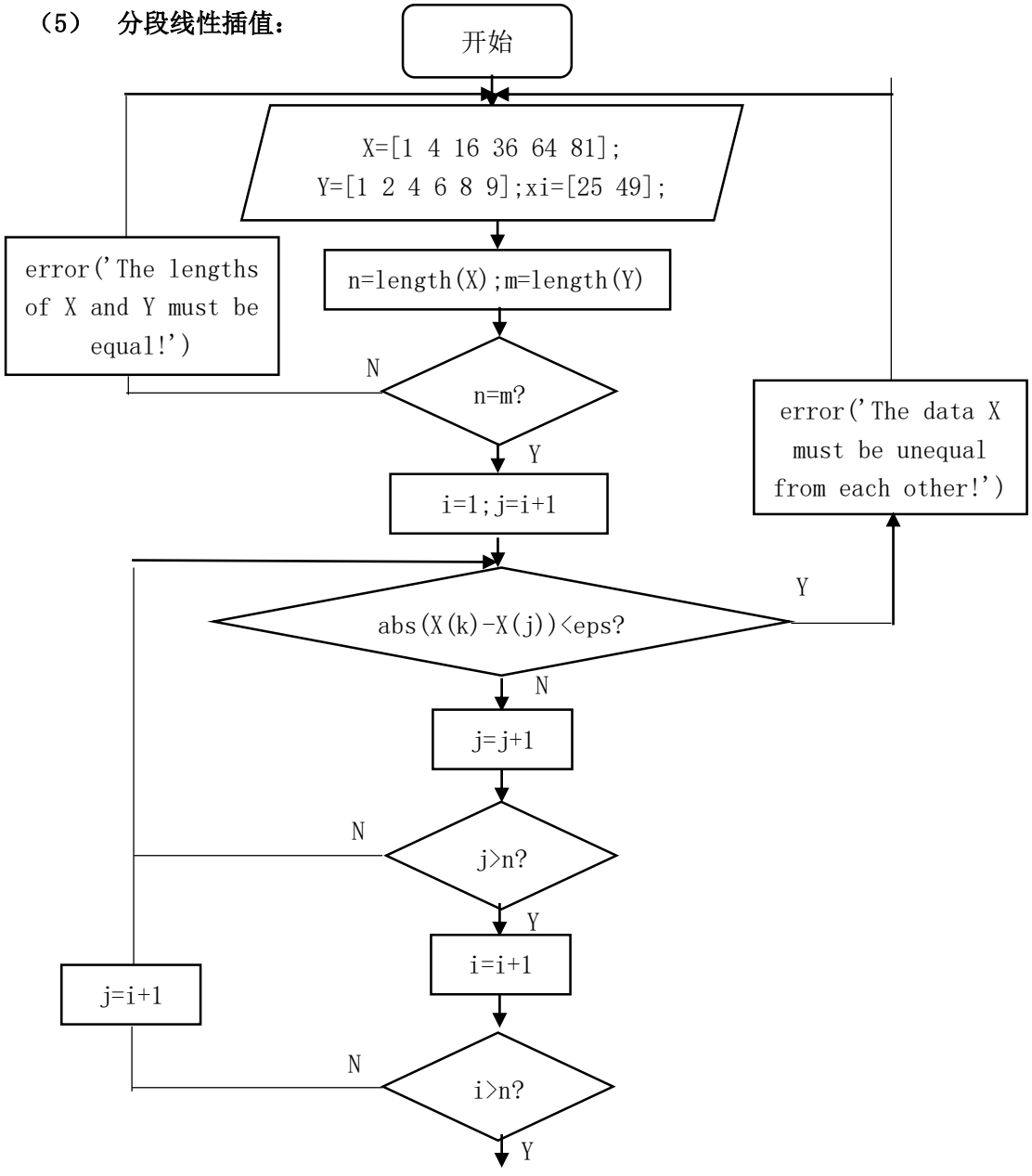

j>i?
N
Y
C(i,1)=Y(i);
j=2
i=i+1
i>n?
N
Y
k=length(xi);i=1;W(i,1)=1;P(i,1)=W(i,1)*C(1,1);j=2
W(i,j)=W(i,j-1)*(xi(i)-X(j-1));
P(i,j)=W(i,j)*C(j,j);j=j+1
j>n?
N
Y
W(i,1)=1;
P(i,1)=W(i,1)*C(1,1);
j=2
yi(i)=sum(P(i,:));i=i+1
i>k?
N
Y
Xi=min(min(X),min(xi)):0.1:max
(max(X),max(xi));
Yi=Newton(X,Y,Xi);YY=sqrt(Xi);
figure;plot(X,Y,'o');hold on;grid on
plot(Xi,Yi,'r');plot(Xi,YY,'g--')
结束

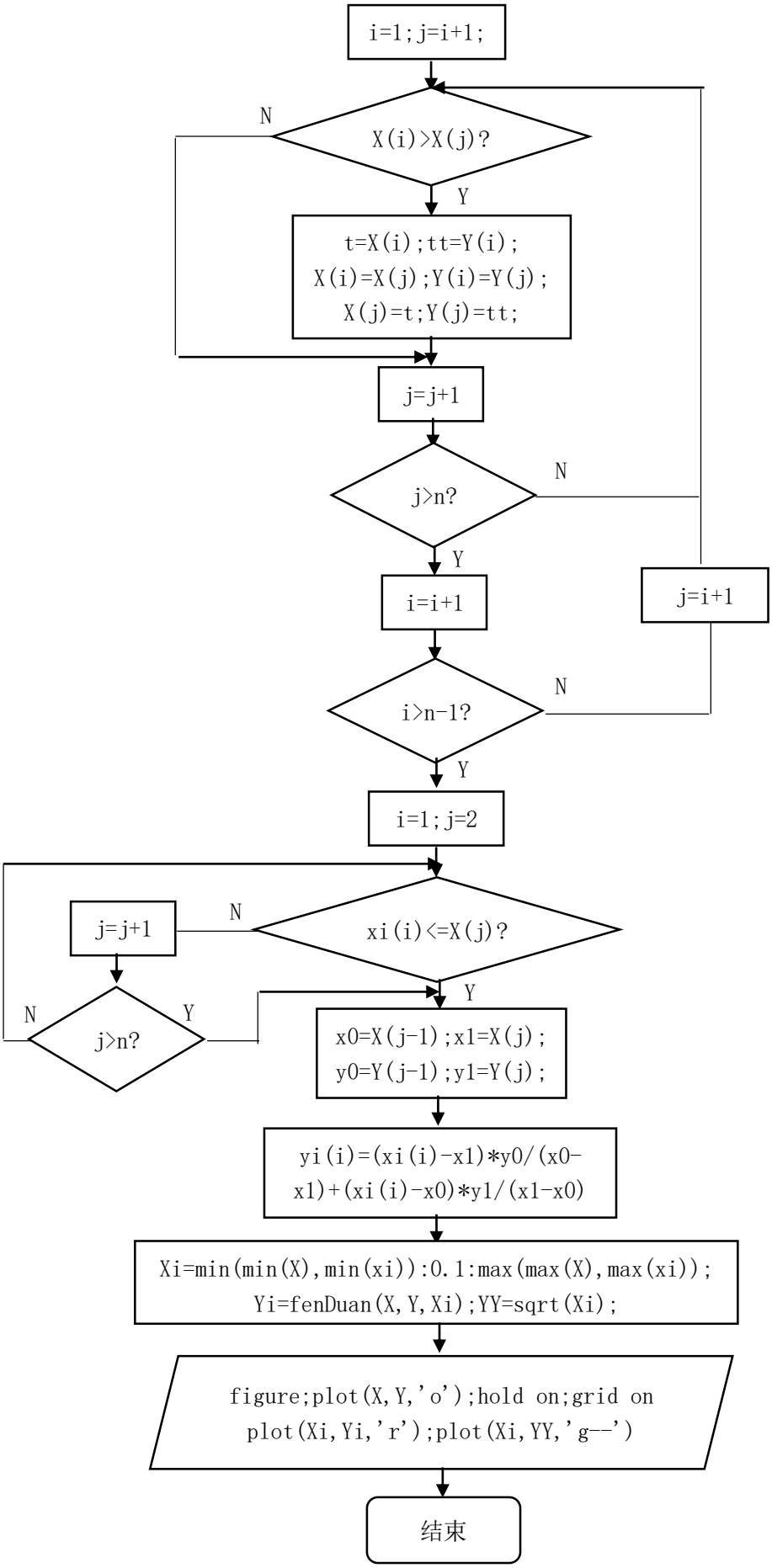
（4） 两点三次埃尔米特插值：



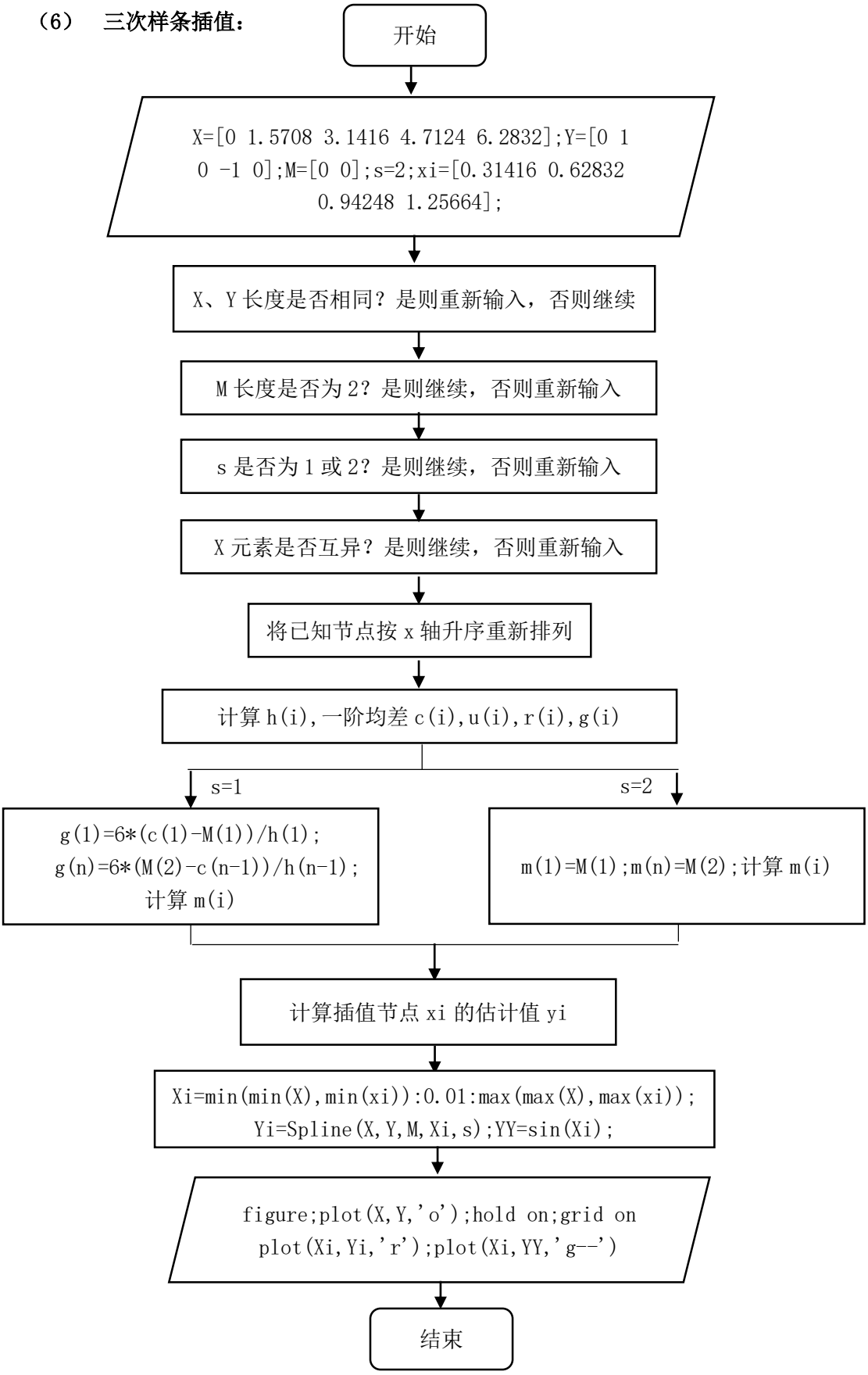

N
i>length(xi)?
Y
Xi=min(min(X),min(xi)):0.1:max(max(X),max(xi));
Yi=Hermite(X,Y,M,Xi);YY=sqrt(Xi);
figure;plot(X,Y,'o');hold on;grid on
plot(Xi,Yi,'r');plot(Xi,YY,'g--')
结束

（5） 分段线性插值：


开始
X=[1 4 16 36 64 81];
Y=[1 2 4 6 8 9];xi=[25 49];
n=length(X);m=length(Y)
N
n=m?
error('The lengths of X and Y must be equal!')
Y
i=1;j=i+1
Y
abs(X(k)-X(j))<eps?
error('The data X must be unequal from each other!')
N
j=j+1
N
j>n?
Y
i=i+1
j=i+1
N
i>n?
Y


i=1;j=i+1;
X(i)>X(j)?
N
Y
t=X(i);tt=Y(i);
X(i)=X(j);Y(i)=Y(j);
X(j)=t;Y(j)=tt;
j=j+1
j>n?
N
Y
i=i+1
j=i+1
i>n-1?
N
Y
i=1;j=2
j=j+1
xi(i)<=X(j)?
N
Y
j>n?
N
Y
x0=X(j-1);x1=X(j);
y0=Y(j-1);y1=Y(j);
yi(i)=(xi(i)-x1)*y0/(x0-
x1)+(xi(i)-x0)*y1/(x1-x0)
Xi=min(min(X),min(xi)):0.1:max(max(X),max(xi));
Yi=fenDuan(X,Y,Xi);YY=sqrt(Xi);
figure;plot(X,Y,'o');hold on;grid on
plot(Xi,Yi,'r');plot(Xi,YY,'g--')
结束

（6） 三次样条插值：


开始
X=[0 1.5708 3.1416 4.7124 6.2832];Y=[0 1 0 -1 0];M=[0 0];s=2;xi=[0.31416 0.62832 0.94248 1.25664];
X、Y 长度是否相同？是则重新输入，否则继续
M 长度是否为 2？是则继续，否则重新输入
s 是否为 1 或 2？是则继续，否则重新输入
X 元素是否互异？是则继续，否则重新输入
将已知节点按 x 轴升序重新排列
计算 h(i),一阶均差 c(i),u(i),r(i),g(i)
s=1
s=2
g(1)=6*(c(1)-M(1))/h(1);
g(n)=6*(M(2)-c(n-1))/h(n-1);
计算 m(i)
m(1)=M(1);m(n)=M(2);计算 m(i)
计算插值节点 xi 的估计值 yi
Xi=min(min(X),min(xi)):0.01:max(max(X),max(xi));
Yi=Spline(X,Y,M,Xi,s);YY=sin(Xi);
figure;plot(X,Y,'o');hold on;grid on
plot(Xi,Yi,'r');plot(Xi,YY,'g--')
结束

八、程序代码

（1） 龙格现象：

```
n1=5;%第一个插值多项式次数
n2=10;%第二个插值多项式次数
X1=-5:10/n1:5;%第一个插值多项式节点 X 值
X2=-5:10/n2:5;%第二个插值多项式节点 X 值
Xi=-5:0.1:5;%原函数 X 值
% Y1=(1+X1.^2).^(-1); Y2=(1+X2.^2).^(-1); YY=(1+Xi.^2).^(-1);%实验函数 1
% Y1=(1+X1.^4).^(-1); Y2=(1+X2.^4).^(-1); YY=(1+Xi.^4).^(-1);%实验函数 2
% Y1=exp(-X1.^2); Y2=exp(-X2.^2); YY=exp(-Xi.^2);%实验函数 3
Y1=2.^(-(X1.^2).^(1/3)); Y2=2.^(-(X2.^2).^(1/3)); YY=2.^(-(Xi.^2).^(1/3));%实验函数 4
Yi1=Lagrange(X1,Y1,Xi);%第一个插值多项式在整个区间上的函数值
Yi2=Lagrange(X2,Y2,Xi);%第二个插值多项式在整个区间上的函数值
figure;
plot(Xi,Yi1,'g--')%第一个插值多项式函数图形，绿色
hold on;grid on
plot(Xi,Yi2,'r-.')%第二个插值多项式函数图形，红色
plot(Xi,YY,'k')%原函数图形，黑色
```

（2） 拉格朗日插值：

① Lagrange 插值多项式函数：

```
function yi=Lagrange(x,y,xi)
% Lagrange 插值多项式，其中
% x 为向量，全部的插值节点；
% y 为向量，插值节点处的函数值；
% xi 为向量，被估计函数自变量；
% yi 为向量，xi 处的函数估计值
n=length(x);m=length(y);%X、Y 长度
if n~=m % 输入的插值点与它的函数值应有相同的个数
    error('The lengths of X and Y must be equal!');%报错
    return;
end
for jj=1:length(xi)%对每个被估计函数自变量做计算
    p=zeros(1,n);%单项式初始化
    for k=1:n%计算每个单项式
        t=ones(1,n);%单项式的因式初始化
        for j=1:n%计算每个因式
            if j~=k%输入的插值节点必须互异
                if abs(x(k)-x(j))<eps
```

```
                    error('the DATA is error');%报错
                    return
                end
                t(j)=(xi(jj)-x(j))/(x(k)-x(j));%计算因式
            end
        end
        p(k)=prod(t);%计算单项式
    end
    yi(jj)=sum(y.*p);%计算函数估计值
end
```

② 调用：

```
x=[1 4 9];%插值多项式节点 X 值
y=[1 2 3];%插值多项式节点 Y 值
xi=[5,6];%待插值点 X 值
yi=Lagrange(x,y,xi)%计算待插值点 Y 值
xi=1:0.1:10;%绘制函数图像的 X 值
yi=Lagrange(x,y,xi);%绘制插值多项式函数图像的 Y 值
yy=sqrt(xi);%绘制原函数图像的 Y 值
figure;plot(x,y,'o')%所有节点，圆圈
hold on
plot(xi,yi,'r--')%插值多项式函数图形，红色
plot(xi,yy,'g')%原函数图形，绿色
```

（3） 牛顿插值：

① Newton 插值多项式函数：

```
function yi=Newton(X,Y,xi)
% Newton 插值多项式，其中
% X 为向量，全部的插值节点；
% Y 为向量，插值节点处的函数值；
% xi 为向量，被估计函数自变量；
% yi 为向量，xi 处的函数估计值
n=length(X);m=length(Y);%X、Y 长度
if n~=m %输入的插值点与它的函数值应有相同的个数
    error('The lengths of X and Y must be equal!');%报错
    return;
end
for i=1:n %输入的插值节点必须互异
    for j=i+1:n
        if abs(X(i)-X(j))<eps %计算机所能识别的精度
            error('The data X must be unequal from each other!');%报
错
            return;
        end
```

```
    end
end
C=zeros(n,n);%初始化均差矩阵
C(1,1)=Y(1);
for i=2:n %计算均差矩阵
    C(i,1)=Y(i);
    for j=2:i
        C(i,j)=(C(i,j-1)-C(i-1,j-1))/(X(i)-X(i-j+1));
    end
end
k=length(xi);%被估计点的个数
for i=1:k %第 i 个被估计点
    W(i,1)=1;%单项式的连乘项
    P(i,1)=W(i,1)*C(1,1);%每个单项式
    for j=2:n %第 j 个单项式
        W(i,j)=W(i,j-1)*(xi(i)-X(j-1));%计算连乘项
        P(i,j)=W(i,j)*C(j,j);%计算单项式
    end
    yi(i)=sum(P(i,:));%单项式叠加，计算估计值
end
```

② 调用：

```
X=[1 4 16 36 64 81];%插值多项式节点 X 值
Y=[1 2 4 6 8 9];%插值多项式节点 Y 值
xi=[25 49];%待插值点 X 值
yi=Newton(X,Y,xi)%计算待插值点 Y 值
Xi=min(min(X),min(xi)):0.1:max(max(X),max(xi));%绘制函数图像的 X 值
Yi=Newton(X,Y,Xi);%绘制插值多项式函数图像的 Y 值
YY=sqrt(Xi);%绘制原函数图像的 Y 值
figure;plot(X,Y,'o')%所有节点，圆圈
hold on;grid on
plot(Xi,Yi,'r--')%插值多项式函数图形，红色
plot(Xi,YY,'g')%原函数图形，绿色
```

（4） 两点三次埃尔米特插值：

① 两点三次 Hermite 插值多项式函数：

```
function yi=Hermite(X,Y,M,xi)
% 两点三次 Hermite 插值多项式，其中
% X 为向量，全部的插值节点；
% Y 为向量，插值节点处的函数值；
% M 为向量，插值点处的导数值；
% xi 为向量，被估计函数自变量；
% yi 为向量，xi 处的函数估计值。
n=length(X);m=length(Y);k=length(M);%X、Y、M 的长度
```

```
if n==2&&m==2&&k==2 %X、Y、M 的长度应为 2
    for i=1:n %输入的插值节点必须互异
        for j=i+1:n
            if abs(X(i)-X(j))<eps %计算机所能识别的精度
                error('The data X must be unequal from each
other!');%报错
                return;
            end
        end
    end
    for i=1:length(xi) %计算每个被估计点的参数和函数值
        a0(i)=(1+2*(xi(i)-X(1))/(X(2)-X(1)))*((xi(i)-X(2))/(X(1)-
X(2)))^2;%参数 a0
        a1(i)=(1+2*(xi(i)-X(2))/(X(1)-X(2)))*((xi(i)-X(1))/(X(2)-
X(1)))^2;%参数 a1
        b0(i)=(xi(i)-X(1))*((xi(i)-X(2))/(X(1)-X(2)))^2;%参数 b0
        b1(i)=(xi(i)-X(2))*((xi(i)-X(1))/(X(2)-X(1)))^2;%参数 b1
        yi(i)=Y(1)*a0(i)+Y(2)*a1(i)+M(1)*b0(i)+M(2)*b1(i);%估计值
    end
else
    error('The lengths of X, Y and M must be 2!');%报错
    return;
end
```

② 调用：

```
X=[1 4];%插值多项式节点 X 值
Y=[1 2];%插值多项式节点 Y 值
M=[1/2 1/4];%插值多项式节点导数值
xi=[2 3]; %待插值点 X 值
yi=Hermite(X,Y,M,xi)%计算待插值点 Y 值
Xi=min(min(X),min(xi)):0.1:max(max(X),max(xi));%绘制函数图像的 X 值
Yi=Hermite(X,Y,M,Xi);%绘制插值多项式函数图像的 Y 值
YY=sqrt(Xi);%绘制原函数图像的 Y 值
figure;plot(X,Y,'o')%所有节点，圆圈
hold on;grid on
plot(Xi,Yi,'r--')%插值多项式函数图形，红色
plot(Xi,YY,'g')%原函数图形，绿色
```

（5） 分段线性插值：

① fenDuan 插值多项式函数：

```
function yi=fenDuan(X,Y,xi)
% fenDuan 插值多项式，其中
% X 为向量，全部的插值节点；
% Y 为向量，插值节点处的函数值；
```

```
% xi 为向量，被估计函数自变量；
% yi 为向量，xi 处的函数估计值
n=length(X);m=length(Y);%X、Y 长度
if n~=m %输入的插值点与它的函数值应有相同的个数
    error('The lengths of X and Y must be equal!');%报错
    return;
end
for i=1:n %输入的插值节点必须互异
    for j=i+1:n
        if abs(X(i)-X(j))<eps %计算机所能识别的精度
            error('The data X must be unequal from each other!');%报
错
            return;
        end
    end
end
for i=1:n-1 %将所有节点按 X 升序排序
    for j=i+1:n
        if X(i)>X(j)
            t=X(i);tt=Y(i);
            X(i)=X(j);Y(i)=Y(j);
            X(j)=t;Y(j)=tt;
        end
    end
end
for i=1:length(xi)%计算每个被估计点
    for j=2:n %寻找被估计点所在区间
        if xi(i)<=X(j)
            x0=X(j-1);x1=X(j);%区间端点
            y0=Y(j-1);y1=Y(j);%区间端点函数值
            break;
        end
    end
    yi(i)=(xi(i)-x1)*y0/(x0-x1)+(xi(i)-x0)*y1/(x1-x0);%计算估计值
end
```

② 调用：

```
X=[1 4 16 36 64 81];%插值多项式节点 X 值
Y=[1 2 4 6 8 9];%插值多项式节点 Y 值
xi=[25 49];%待插值点 X 值
yi=fenDuan(X,Y,xi)%计算待插值点 Y 值
Xi=min(min(X),min(xi)):0.1:max(max(X),max(xi));%绘制函数图像的 X 值
Yi=fenDuan(X,Y,Xi);%绘制插值多项式函数图像的 Y 值
YY=sqrt(Xi);%绘制原函数图像的 Y 值
```

```
figure;plot(X,Y,'o')%所有节点，圆圈
hold on;grid on
plot(Xi,Yi,'r--')%插值多项式函数图形，红色
plot(Xi,YY,'g')%原函数图形，绿色
```

（6） 三次样条插值：

① Spline 插值多项式函数：

```
function yi=Spline(X,Y,M,xi,s)
% Spline 插值多项式，其中
%  X 为向量，全部的插值节点；
%  Y 为向量，插值节点处的函数值；
%  M 为向量，边界条件，包含两个数值；
% xi 为向量，被估计函数自变量；
%  s 为整数，1 表示第一类边界条件，2 表示第二类边界条件；
% yi 为向量，xi 处的函数估计值
n=length(X);%X 的长度
if length(Y)~=n %输入的插值点与它的函数值应有相同的个数
    error('The lengths of X and Y must be equal!');%报错
    return;
elseif length(M)~=2 %M 的长度应为 2
    error('The length of M must be 2!');%报错
    return;
elseif s~=1&&s~=2 %s 的值应为 1 或 2
    error('s must be 1 or 2!');%报错
    return;
end
for i=1:n %输入的插值节点必须互异
    for j=i+1:n
        if abs(X(i)-X(j))<eps %计算机所能识别的精度
            error('The data X must be unequal from each other!');%报
错
            return;
        end
    end
end
for i=1:n-1 %将所有节点按 X 升序排序
    for j=i+1:n
        if X(i)>X(j)
            t=X(i);tt=Y(i);
            X(i)=X(j);Y(i)=Y(j);
            X(j)=t;Y(j)=tt;
        end
    end
end
```

```
for i=2:n
    h(i-1)=X(i)-X(i-1);%计算 h(i)
    c(i-1)=(Y(i)-Y(i-1))/h(i-1);%计算均差 c(i)
end
for i=1:n-2
    u(i)=h(i)/(h(i)+h(i+1));%计算 u(i)
    r(i)=1-u(i);%计算 r(i)
    g(i+1)=6*(c(i+1)-c(i))/(h(i)+h(i+1));%计算 g(i)
end
if s==1 %第一类边界条件
    g(1)=6*(c(1)-M(1))/h(1);
    g(n)=6*(M(2)-c(n-1))/h(n-1);
    A=zeros(n,n);%初始化系数矩阵
    A(1,1)=2;A(1,2)=1;
    A(n,n-1)=1;A(n,n)=2;
    for i=2:n-1%系数矩阵赋值
        A(i,i-1)=u(i-1);
        A(i,i)=2;
        A(i,i+1)=r(i-1);
    end
    for i=1:n%赋值常数项矩阵
        G(i,1)=g(i);
    end
    m=A\G;%求解 m
else  %第二类边界条件
    m(1)=M(1);m(n)=M(2);
    A=zeros(n-2,n-2);%初始化系数矩阵
    A(1,1)=2;A(1,2)=r(1);
    A(n-2,n-3)=u(n-2);A(n-2,n-2)=2;
    for i=2:n-3%系数矩阵赋值
        A(i,i-1)=u(i);
        A(i,i)=2;
        A(i,i+1)=r(i);
    end
    gg(1)=g(2)-u(1)*M(1);gg(n-2)=g(n-1)-r(n-2)*M(2);
    for i=1:n-4
        gg(i+1)=g(i+2);
    end
    for i=1:n-2%赋值常数项矩阵
        G(i,1)=gg(i);
    end
    mm=A\G;
    m(2:n-1)=mm;%求解 m
end
```

```
for i=1:length(xi)%计算每个被估计点
    for j=2:n %寻找被估计点所在区间
        if xi(i)<=X(j)
            x0=X(j-1);x1=X(j);%区间端点
            m0=m(j-1);m1=m(j);%区间端点二阶导数值
            y0=Y(j-1);y1=Y(j);%区间端点函数值
            h1=h(j-1);%区间长度
            break;
        end
    end
    yi(i)=(m0*((x1-xi(i))^3)/(6*h1))+(m1*((xi(i)-x0)^3)/(6*h1))+...
        ((y0-m0*(h1^2)/6)*(x1-xi(i))/h1)+((y1-m1*(h1^2)/6)*(xi(i)-
x0)/h1);%计算估计值
end
```

② 调用：

```
X=[0 1.5708 3.1416 4.7124 6.2832];%插值多项式节点 X 值
Y=[0 1 0 -1 0];%插值多项式节点 Y 值
M=[0 0];%插值多项式边界条件
s=2;%插值多项式边界条件类型
xi=[0.31416 0.62832 0.94248 1.25664];%待插值点 X 值
yi=Spline(X,Y,M,xi,s)%计算待插值点 Y 值
Xi=min(min(X),min(xi)):0.01:max(max(X),max(xi));%绘制函数图像的 X 值
Yi=Spline(X,Y,M,Xi,s);%绘制插值多项式函数图像的 Y 值
YY=sin(Xi);%绘制原函数图像的 Y 值
figure;plot(X,Y,'o')%所有节点，圆圈
hold on;grid on
plot(Xi,Yi,'r--')%插值多项式函数图形，红色
plot(Xi,YY,'g')%原函数图形，绿色
```

▪ 九、运行结果及实验结果分析

（1） 龙格现象：

① 考察函数

$$f_1(x)=\frac{1}{1+x^2},-5\le x\le 5$$

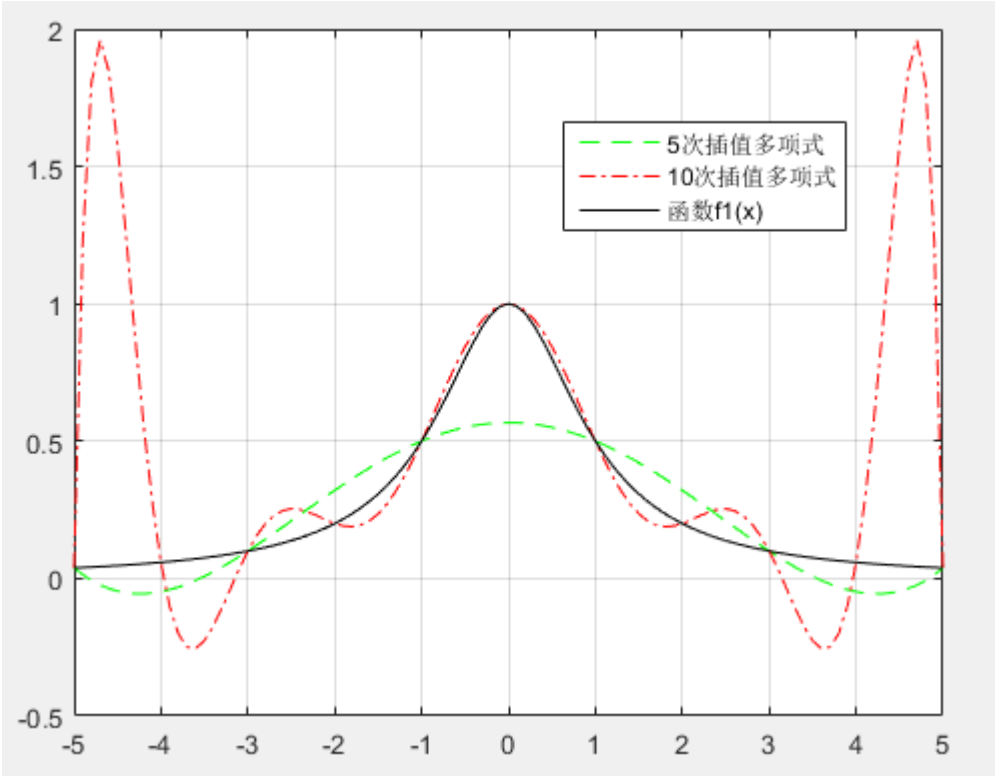


图 9.1 函数 f1(x)不同次数插值多项式对比

② 考察函数

$$f_2(x) = \frac{1}{1+x^4}, -5 \le x \le 5$$

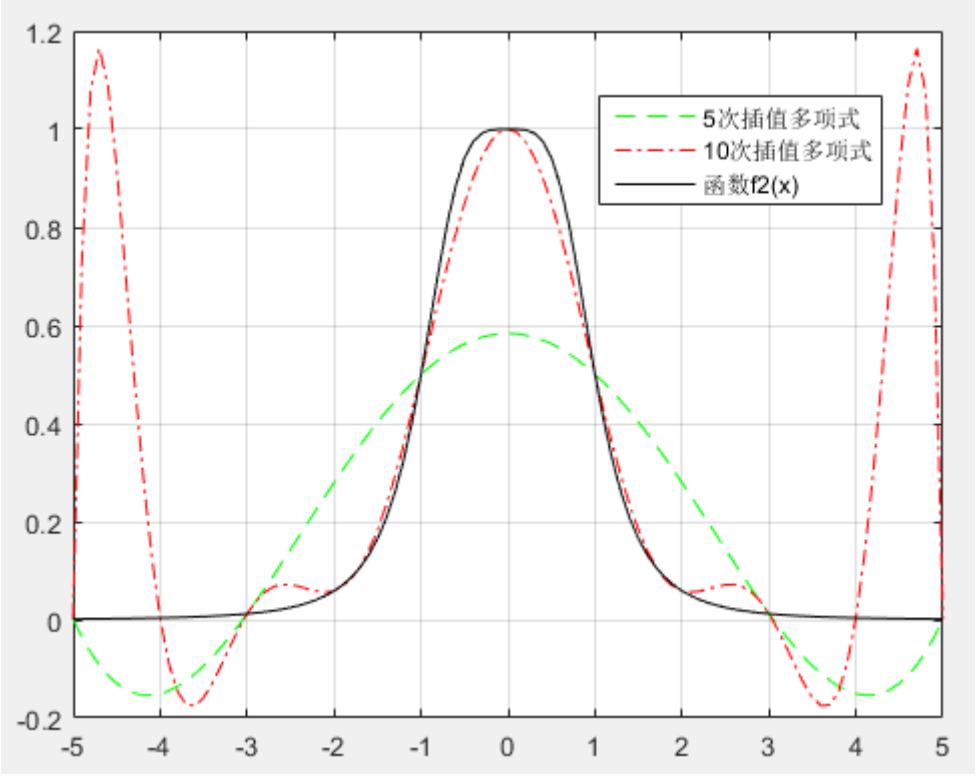


图 9.2 函数 f2(x)不同次数插值多项式对比

③ 考察函数

$$f_3(x) = e^{-x^2}, -5 \le x \le 5$$

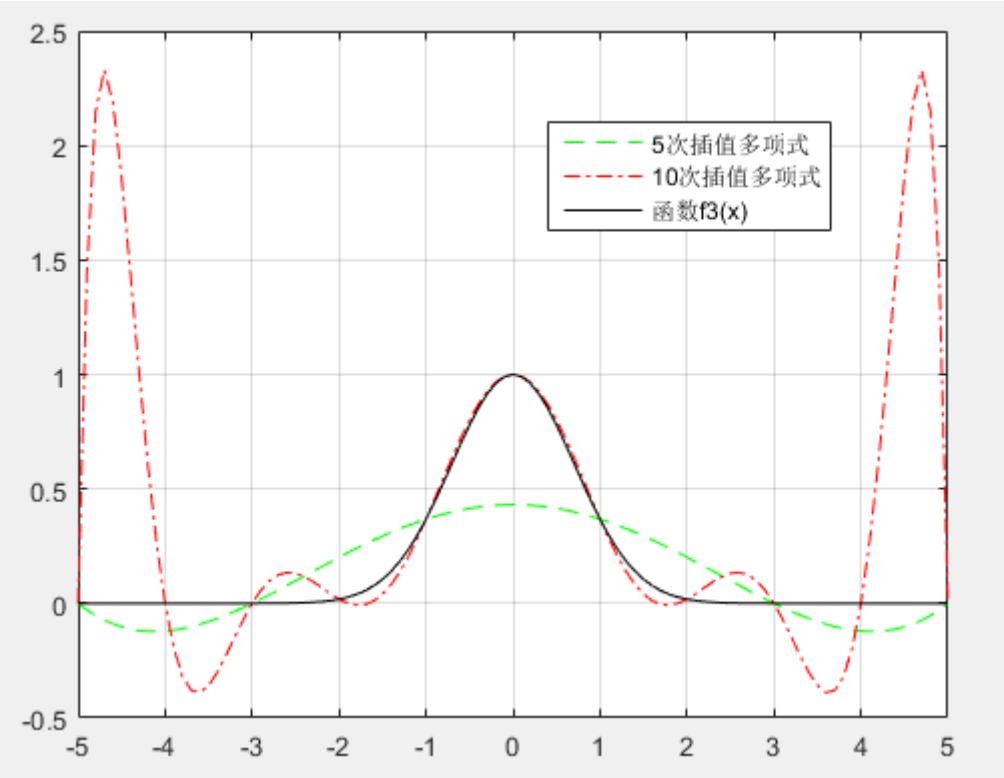


图 9.3 函数 f3(x)不同次数插值多项式对比

④ 考察函数

$$f_4(x) = 2^{-x^{2/3}}, -5 \le x \le 5$$

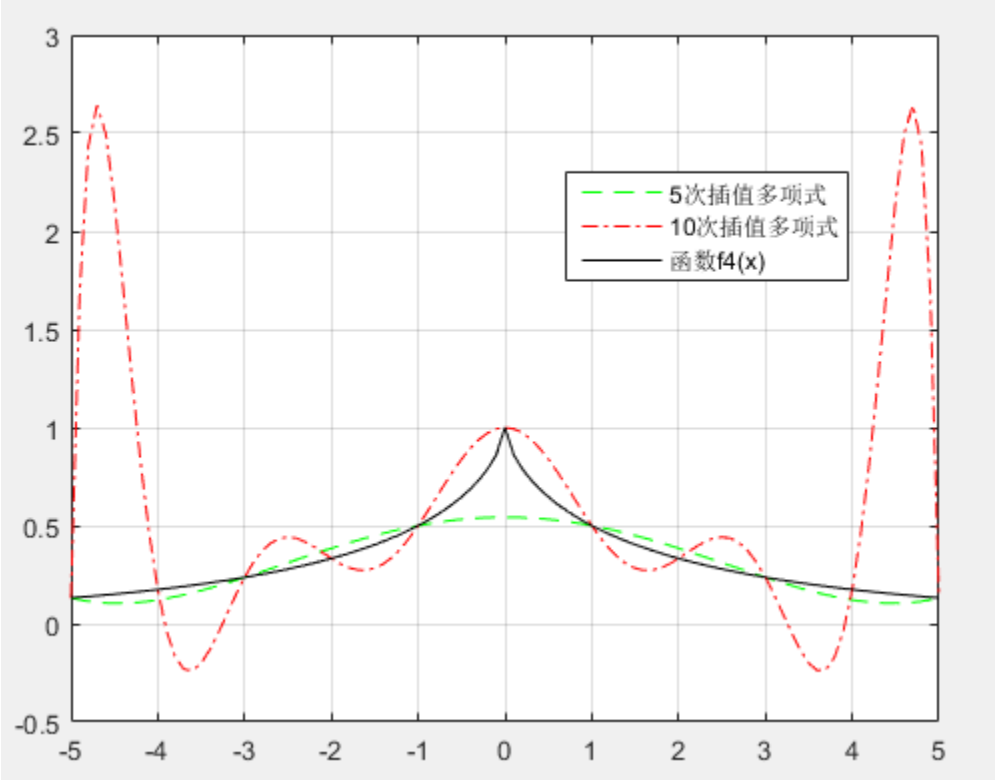


图 9.4 函数 f4(x)不同次数插值多项式对比

图 9.1、9.2、9.3、9.4 显示，当 n 增大时，n 次插值多项式在两端会出现激烈的震荡，这说明了在大范围内使用高次插值，逼近的效果往往是不理想的。另外，从舍入误差来看，高次插值误差的传播比较严重，在一个节点上产生的舍入误差会在计算的过程中不断扩大，并传播到其他节点上。因此，次数太高的高次插值多项式并不实用，因为节点数增加时，计算量增大了，但插值函数的精度并未提高。

（2） 拉格朗日插值：

输入：x=[1 4 9];y=[1 2 3];xi=[5,6];

输出：yi =[2.2667 2.5000]

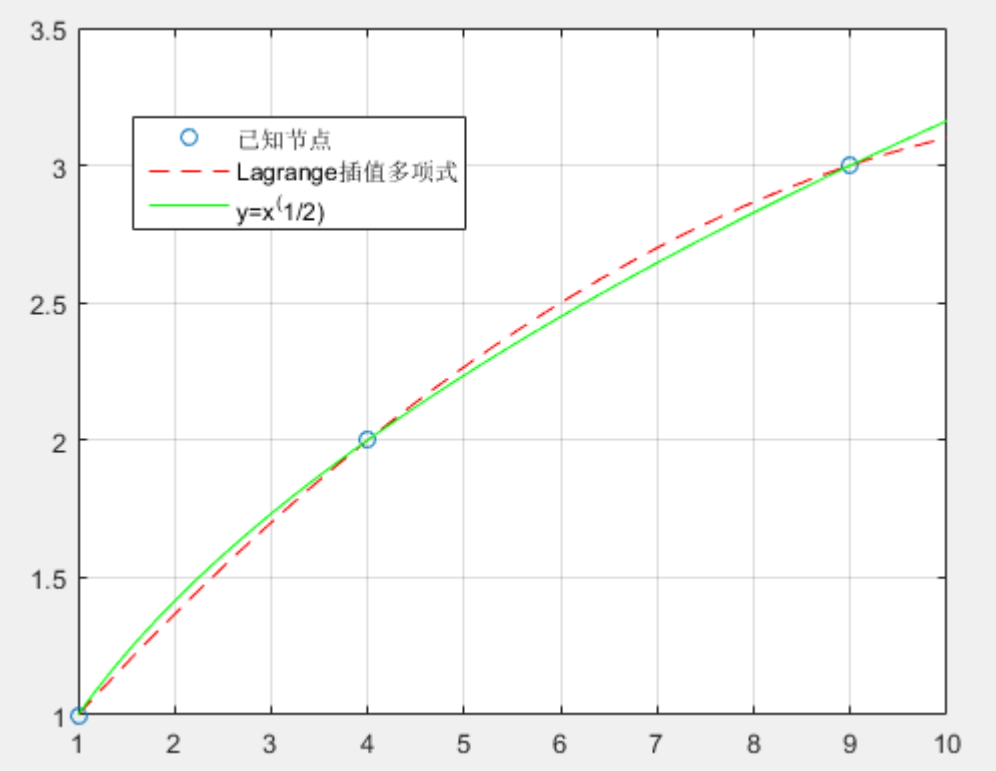


图 9.5 拉格朗日插值结果

拉格朗日插值多项式结构对称，使用方便。从图 9.5 可以看出其插值结果较好，其插值多项式的余项为

$$R_2(x)=\frac{f^{(3)}(\xi)}{3!}\omega_3(x)\le 0.004(x-1)(x-2)(x-3)$$

但由于使用基函数构成的插值，这样要增加一个节点时，所有的基函数必须全部重新计算，不具备承袭性，还造成计算量的浪费。

（3） 牛顿插值：

输入：X=[1 4 16 36 64 81]; Y=[1 2 4 6 8 9]; xi=[25 49];

输出：yi =[4.8384 7.3587]

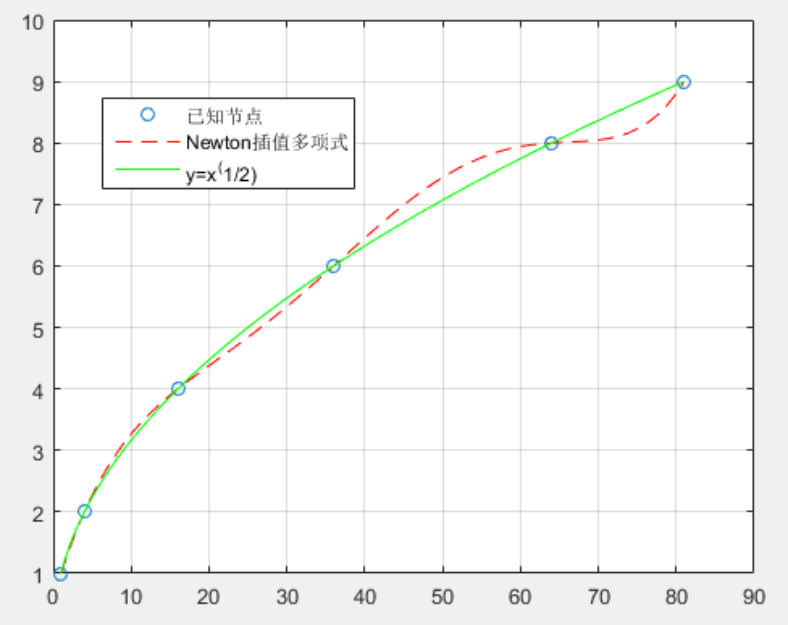


图 9.6 牛顿插值结果

牛顿插值多项式是插值多项式的另一种表示形式，与 Lagrange 多项式相比它不仅克服了“增加一个节点时整个计算工作重新开始”的缺点，且可以节省乘除法运算次数，同时在 Newton 插值多项式中用到差分与差商等概念，又与数值计算的其他方面有密切的关系。从图 9.6 可以看出其插值结果较好，其插值多项式的余项为

$$R_5(x) = \frac{f^{(6)}(\xi)}{6!}\omega_3(x)$$

但当 n 增大时，可能会出现龙格现象。

（4） 两点三次埃尔米特插值：

输入：X=[1 4]; Y=[1 2]; M=[1/2 1/4]; xi=[2 3];

输出：yi =[1.4259 1.74077]

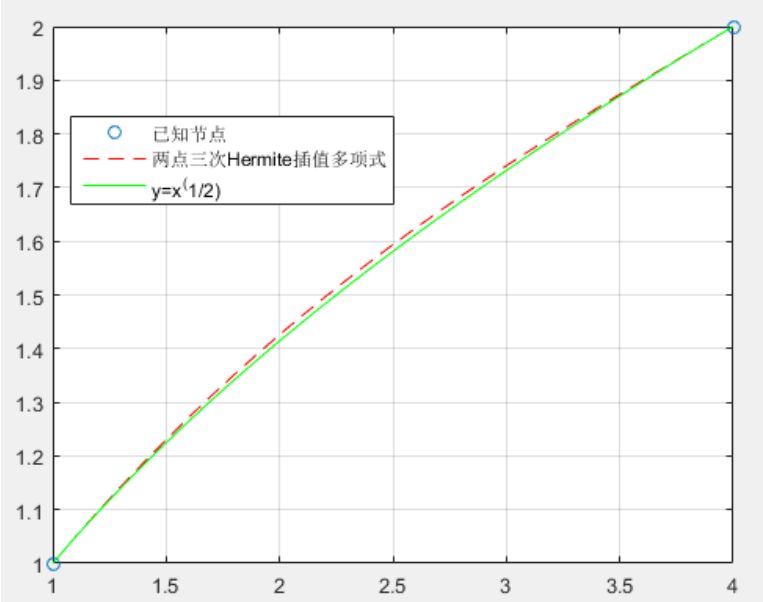


图 9.7 两点三次埃尔米特插值结果

两点三次 Hermite 插值多项式给出了满足函数值相等且导数也相等的插值方法，这样可以使得插值结果更加满足求解需要。从图 9.7 可以看出其插值结果较好，其插值多项式的余项为

$$R_3(x) = \frac{f^{(4)}(\xi)}{4!}(x-1)(x-4)$$

但此方法的限定条件较多，通用性不是很好。

（5） 分段线性插值：

输入：X=[1 4 16 36 64 81]; Y=[1 2 4 6 8 9];M=[1/2 1/4]; xi=[25 49];

输出：yi =[4.9000 6.9286]

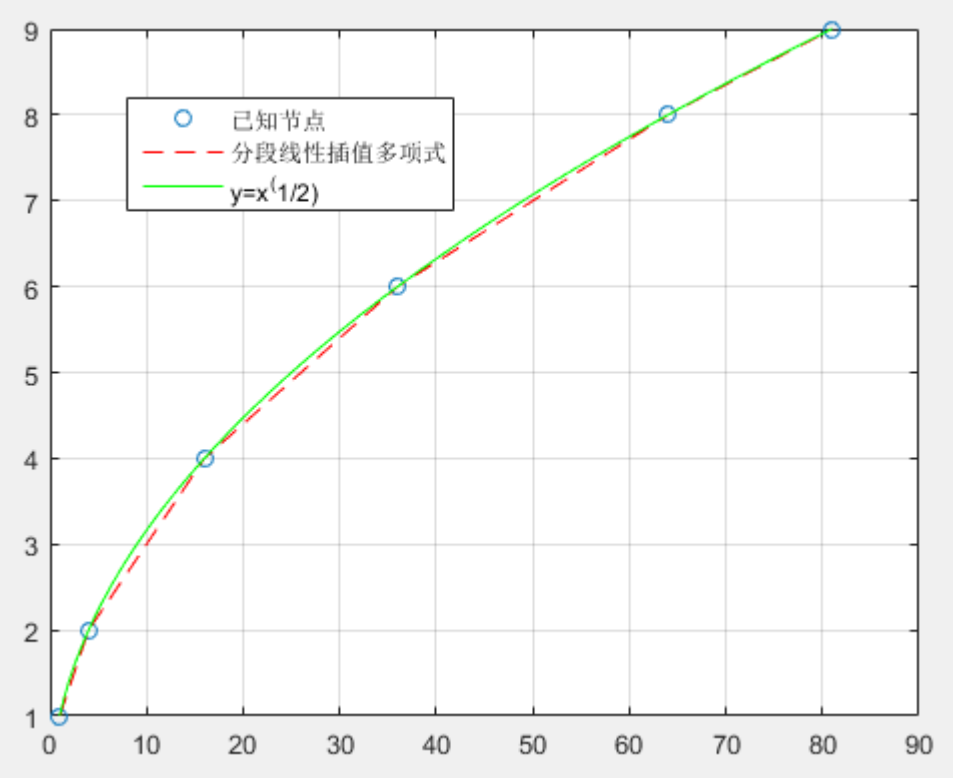


图 9.8 分段线性插值结果

分段线性插值就是通过插值节点用折线段连接起来逼近 f(x)，在几何上就是用折线替代曲线，此方法在节点较多、较密的情况下有很好的收敛性，在一定程度上避免了龙格现象的出现。从图 9.8 可以看出其插值结果较好，其插值多项式的余项为

$$R(x) \le \frac{h^2}{8} \max_{x_i \le x \le x_{i+1}} |f''(x)|$$

但当节点较少、较疏时，分段线性插值结果较差，同时，因为函数使用折线连接的，故此方法插值的多项式不具有圆滑性。

（6） 三次样条插值：

输入：X=[0 1.5708 3.1416 4.7124 6.2832]; Y=[0 1 0 -1 0]; M=[0 0]; s=2; xi=[0.31416 0.62832 0.94248 1.25664];

输出：yi =[0.2960 0.5680 0.7920 0.9440]

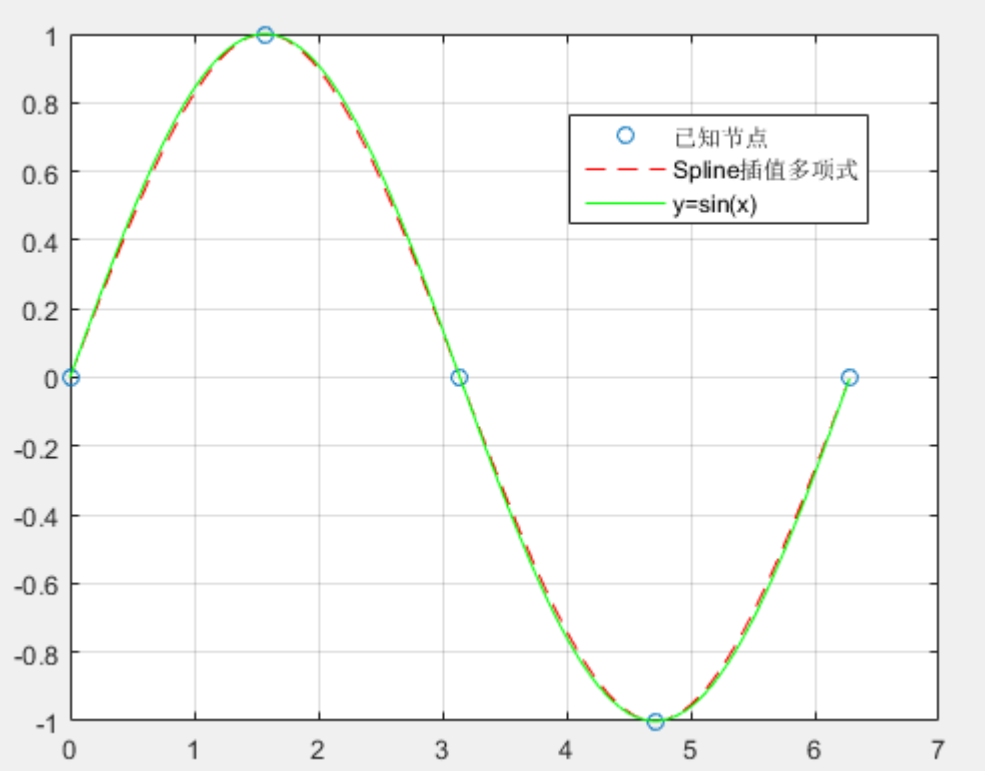


图 9.9 三次样条插值结果

三次样条绘制的曲线不仅有很好的光滑度，而且当节点逐渐加密时，其函数值在整体上能很好地逼近被插函数，相应的导数值也收敛于被插函数的导数， 不会发生龙格现象。从图 9.9 可以看出其插值结果极好，其插值多项式的余项相应的也较小。因此三次样条在计算机辅助设计中有广泛的应用。

但此方法计算复杂度较大。

▪ 十、拓展：实时输入处理的牛顿插值方法的实现

从第九节的分析中我们得到一个结论：牛顿插值方法克服了“增加一个节点时整个计算工作重新开始”的缺点，从而节省乘除法运算次数。因此，我们可以使用牛顿插值算法编制一个实时输入处理的插值算法。此过程中使用 matlab GUI 编程实现，其核心代码如下：

①function testGui_OpeningFcn(hObject, eventdata, handles, varargin)%全局信息

clear;clc

global C;global X;global Y;%定义全局变量，C 为均差矩阵，X、Y 为输入点的信息

②function pushbutton1_Callback(hObject, eventdata, handles)%按钮“添加”事件

global X;global Y;global C;%使用全局变量

n=str2double(get(handles.edit5,'String'));%获取当前节点数

n=n+1;%增加一个节点

X(n)=str2double(get(handles.edit1,'String'));%获取节点 X 值

```
    Y(n)=str2double(get(handles.edit2,'String'));%获取节点 Y 值
    f=1;
    for i=1:n-1 %X 的元素必须互异
        for j=i+1:n
            if X(i)==X(j)
                f=0;
                n=n-1;
                errordlg('The data X must be unequal from each other!','错
误'); %报错
                break;
            end
        end
        if f==0
            break;
        end
    end
    if n==1 %实时处理均差矩阵
        C(1,1)=Y(1);%第一个点时直接赋值
    else
        C(n,1)=Y(n);%处理一个点的均差
        for i=2:n
            C(n,i)=(C(n,i-1)-C(n-1,i-1))/(X(n)-X(n-i+1));
        end
    end
    set(handles.edit5,'String',n);%实时显示节点数
```

③function pushbutton2_Callback(hObject, eventdata, handles)%按钮“计算插值”事件

```
    global C;global X;%使用全局变量
    n=str2double(get(handles.edit5,'String'));%获取当前节点数
    xi=str2double(get(handles.edit3,'String'));%获取所求插入点
    W(1)=1;%单项式第一个因式
    P(1)=W(1)*C(1,1);%第一个单项式
    for i=2:n %处理每个单项式
        W(i)=W(i-1)*(xi-X(i-1));%连乘项
        P(i)=W(i)*C(i,i);% 第 i 个单项式
    end
    yi=sum(P);%单项式叠加
    set(handles.edit4,'String',yi);%输出插值结果
```

④function pushbutton3_Callback(hObject, eventdata, handles)%按钮“绘图”事件

```
  cla(handles.axes1)%清除绘图区
  global X;global Y;%使用全局变量
```

```
n=str2double(get(handles.edit5,'String'));%获取当前节点数
plot(handles.axes1,X(1:n),Y(1:n),'o')%绘制节点
hold on;grid on
Xi=min(X(1:n)):0.1:max(X(1:n));%计算绘图使用的 X 值
Yi=Newton(X(1:n),Y(1:n),Xi); %计算绘图使用的 Y 值
plot(Xi,Yi,'r')%绘制插值多项式函数图像
```

程序运行结果如下：

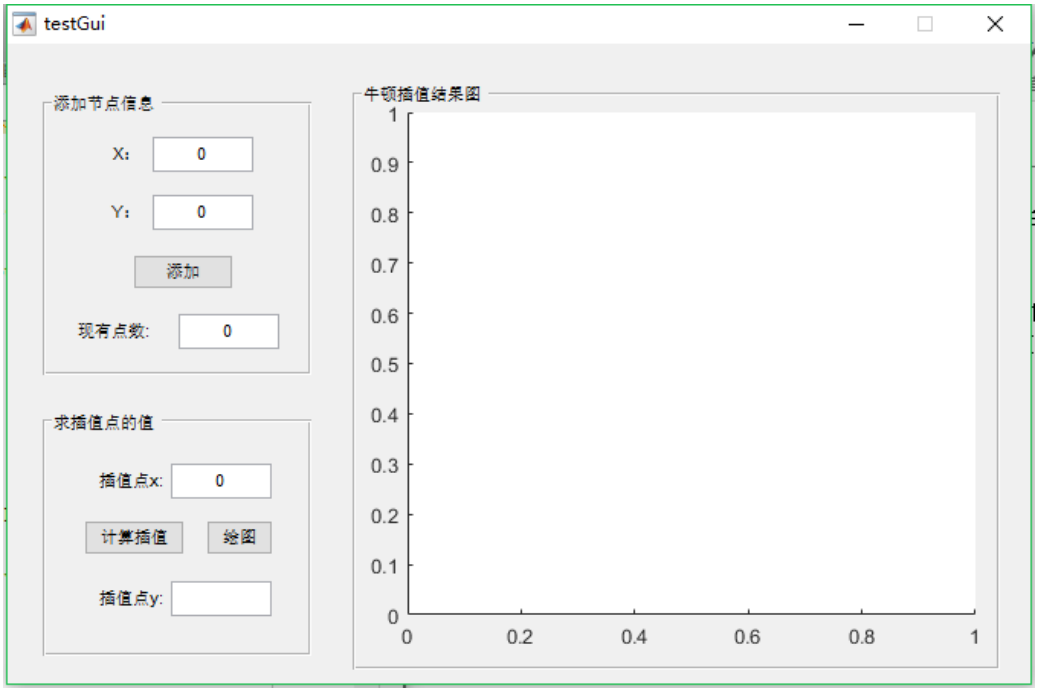


图 10.1 实时输入处理的牛顿插值方法的实现

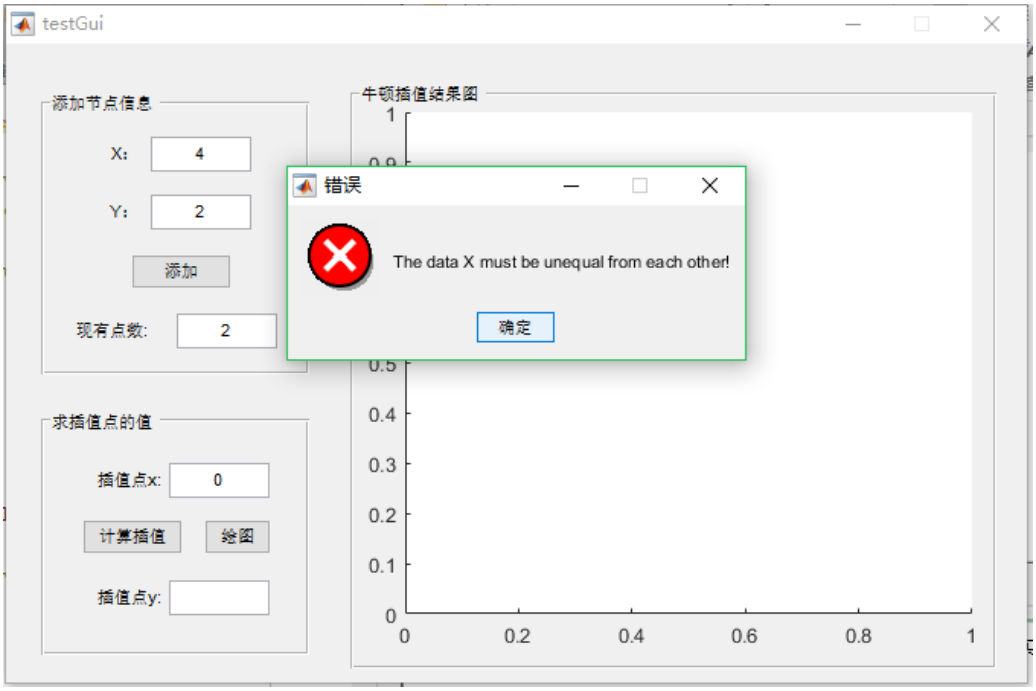


图 10.2 错误提示

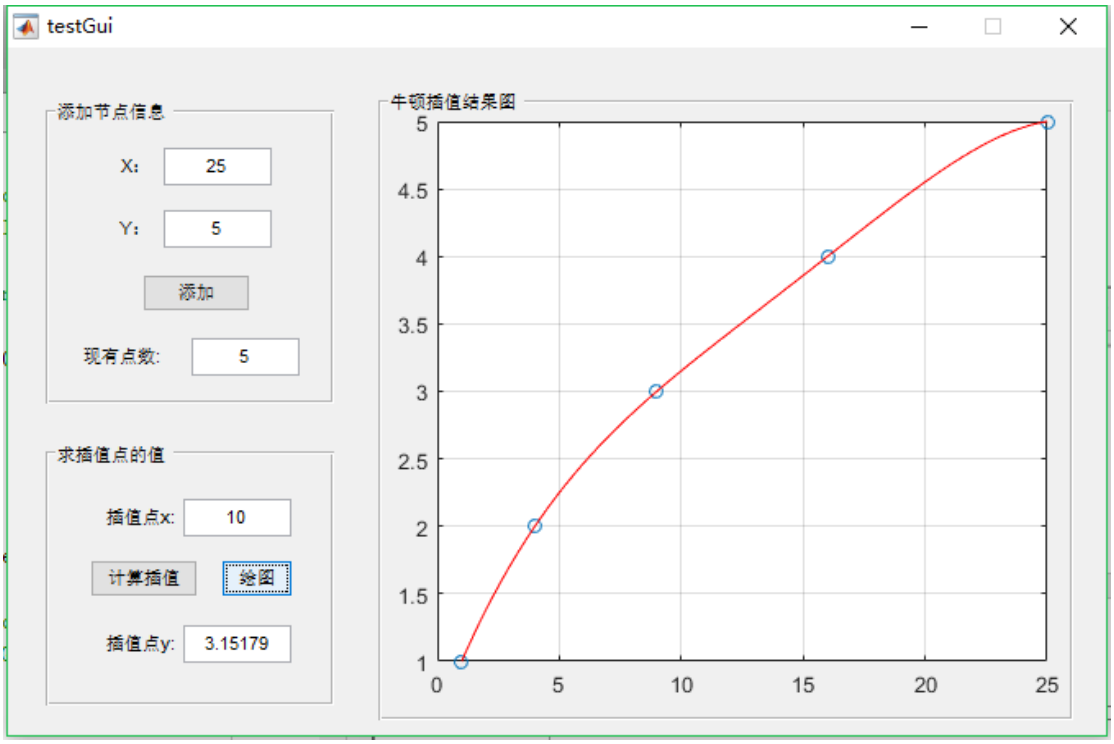


图 10.3 计算插值及绘图

此方法可以实现实时数据输入处理的要求，同时不需要重复计算，每添加一个节点，则绘图区可更新一副插值多项式曲线图，计算效率优于拉格朗日插值算法。

▪ 十一、实验体会

本次实验量较大，主要在于五种插值算法的实现。通过本次实验，我深入理解了龙格现象的含义及原理，认识到在一个固定区间内，当节点数增多并且没有导数值限制的时候，插值多项式函数将会产生剧烈震荡，在两端的估计值误差将会增大。

在五种插值算法分析的过程中，我得到了以下结论：①拉格朗日插值多项式结构对称，使用方便，但不具备承袭性，且会出现龙格现象；②牛顿插值有效解决了节点实时输入的处理，但仍会出现龙格现象；③两点三次 Hermite 插值多项式给出了满足函数值相等且导数也相等的插值方法，这样可以使得插值结果更加满足求解需要，但此方法的限定条件较多，通用性不是很好；④分段线性插值在节点较多、较密的情况下有很好的收敛性，在一定程度上避免了龙格现象的出现，但当节点较少、较疏时，分段线性插值结果较差，且不具有圆滑性；⑤三次样条绘制的曲线有很好的光滑度，能很好地逼近被插函数，相应的导数值也收敛于被插函数的导数，不会发生龙格现象，但此方法计算复杂度较大。

最后，在分析算法的基础上，我实现了实时输入处理的牛顿插值方法，这种方法不需要重复计算。

总而言之，我在这次实验收获丰富，加深了我对插值算法的理解与掌握，最后，要感谢付老师的谆谆教诲与帮助。