数学与科学学院

《数值计算》 课程设计 1

学 号：S321150006

专 业：核科学与技术

学生姓名：陈久涛

任课教师：杨丽宏

完成时间：2021年10月15日

插值多项式的振荡现象

### 一、实验内容

给定函数 *f* (*x*) =  ， *x*[− 1, 1] ，完成以下工作：

1、利用 11个等距节点做 10 次 Lagrange 插值多项式*L*(*x*) ，在同一坐标系下绘出 *f* (*x*) 和*L*(*x*) 的图像，观察发生的现象。若将插值多项式的次数提高到20，结果又将如何？

2、随机产生 19个(− 1, 1) 之间的实数，再加上区间的两个端点， 构造 20次 Lagrange插值多项式*L*(*x*)，在同一坐标系下绘出 *f* (*x*)和*L*(*x*)的图像，该插值函数是否振 荡？

3、利用11个等距节点计算 *f* (*x*)的分段二次插值函数，并绘图，观察发生的现象。

4、利用 n=10,20,40 次切比雪夫多项式的零点为插值节点，构造 Lagrange插值多项式，在同一坐标系下绘出*f* (*x*)和*L*(*x*)的图像，观察插值多项式是否振荡，并对观察到的现象给予简单的解释。

### 二、数学原理

**1、拉格朗日插值法**

一般地，若已知y=f(x)在互不相同 n+1 个点x0,x1,…,xn处的函数值y0，y1,…yn(即该函数过（x0,y0）,(x1,y1),…(xn,yn) 这n+1个点），则可以考虑构造一个过这n+1个点的、次数不超过n的多项式y=Pn(x) ,使其满足：

Pn（xk）=yk,k=0,1,…,n (\*)

要估计任一点ξ，ξ≠xi,i=0,1,2,...,n,则可以用Pn（ξ）的值作为准确值f(ξ)的近似值，此方法叫做“插值法”。称式（\*）为插值条件（准则），含xi(i=0,1,...,n)的最小区间[a,b]，其中a=min{x0,x1,...,xn}，b=max{x0,x1,...,xn}。

在平面上有（x0,y0）,(x1,y1),…(xn-1,yn-1) 共n个点，现作一条函数f(x) 使其图像经过这n个点。

作法：设集合Dn 是关于点(x,y)的角标的[集合](https://baike.baidu.com/item/%E9%9B%86%E5%90%88/2908117" \t "_blank)，Dn={0,1,…,n-1} ，作n个多项式

pj(x),j∈Dn。对于任意k∈Dn,都有pk（x）,Bk={i|i≠k,i∈Dn}

使得

 Pk(x)是n-1次[多项式](https://baike.baidu.com/item/%E5%A4%9A%E9%A1%B9%E5%BC%8F" \t "_blank)，且满足，pk（xm）=0并且pk（xk）=1

最后可得

形如上式的插值多项式称为拉格朗日（Lagrange）插值多项式。

**2、分段线性插值法**

给定区间[a,b]，将其分割成a=x0<x1<…<xn=b,已知函数f=y（x）在这插值结点的函数值为yk=f（xk）k=（0,1,…,n）求一个分段函数Ik（x），使其满足：

(1) Ih(xk)=yk,k=(0,1,..,n);

(2) 在每个区间[xk,xk+1]上，Ih(x)是个一次函数。

易知，Ih(x)是个折线函数，在每个区间[xk,xk+1]上，k=(0,1,…,n)

于是，在[a,b]上是连续的，但其一阶导数是不连续的。

于是即可得到如下分段线性插值函数：

### 三、程序设计及结果

1、利用 11个等距节点做 10 次 Lagrange 插值多项式*L*(*x*) ，在同一坐标系下绘出 *f* (*x*) 和*L*(*x*) 的图像，观察发生的现象。若将插值多项式的次数提高到20，结果又将如何

（1）首先建立lagr.m文件，（见附录一）

编写拉格朗日插值多项式

（2）.建立homework1-1.m文件（见附录二） 此文件首先输入：

x0=[-1:0.01:1];

y0=1./(1+25\*x0.^2);

plot(x0,y0);

axis([-1.1,1.1,-0.5,1.1]);

hold on;

输出为函数*f* (*x*) = 图像

接着调用lagr.m文件的lagrange插值函数，

设置插值多项式次数为10

得出结果如图1.1

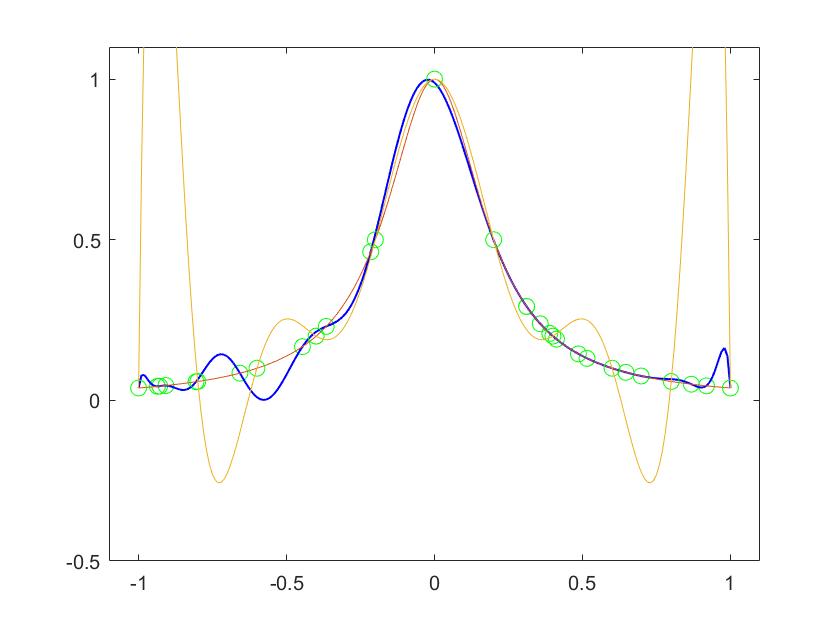


图1.1

将次数提高到20，输出结果如图1.2：

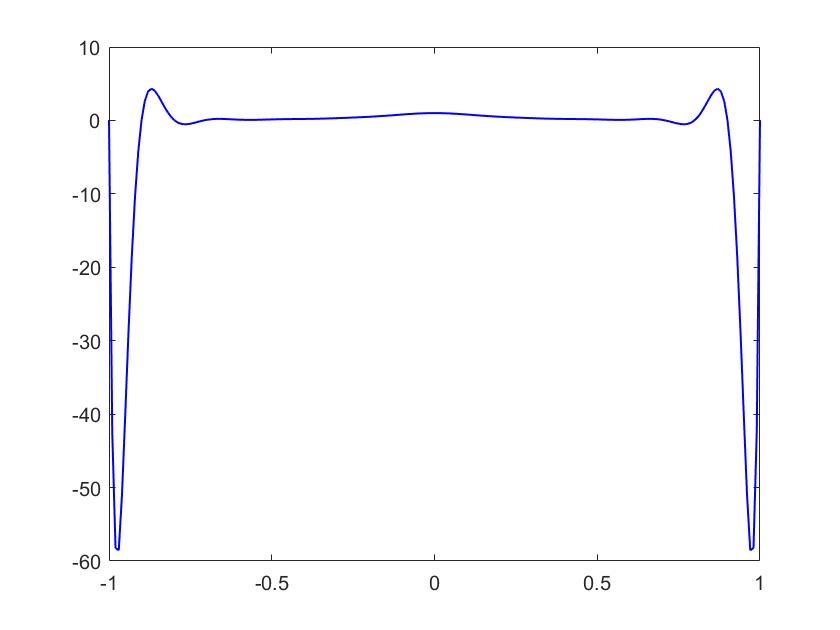


图1.2

图像分析：对于等距划分来说节点数越多，最大误差越大，可是越靠近中间的误差越少。越接近两个端点的误差越大、及龙格现象明显。当节点数很大时，最大误差的来源只与两个端点的误差有关。

2、随机产生 19个(− 1, 1) 之间的实数，再加上区间的两个端点， 构造 20次 Lagrange插值多项式*L*(*x*)，在同一坐标系下绘出 *f* (*x*)和*L*(*x*)的图像，该插值函数是否振 荡？

在 homework1-2.m（见附录三）文件中输入代码，生成随机数

rand\_x=2\*rand(1,19)-1;

x=[-1,rand\_x,1];

再进行如第一题计算流程，得出结果如图2.1

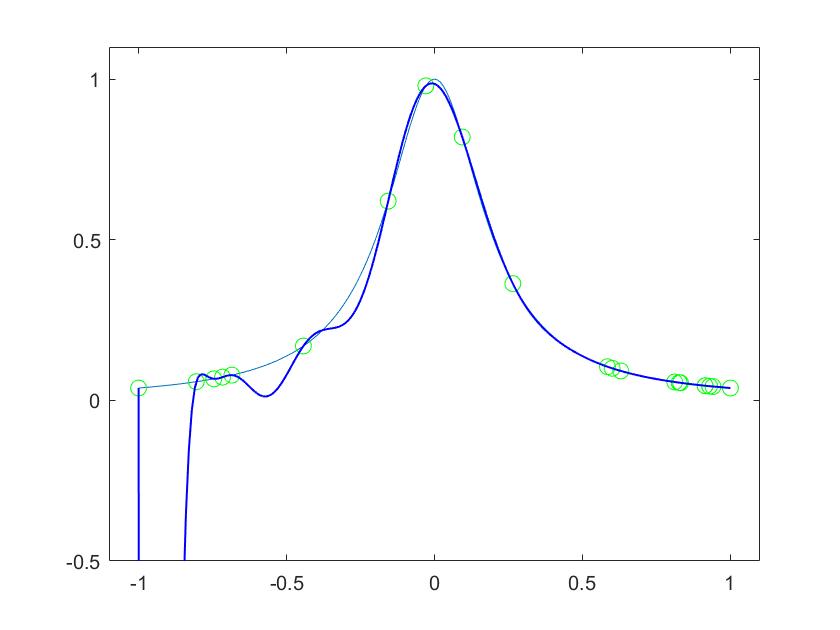


图2.1

图像分析：对于随机生成的节点构造的高次拉格朗日插值多项式，其插值值与真实值偏差非常大，存在较大的震荡。

3、利用11个等距节点计算 f(x)的分段二次插值函数,并绘图，观察发生的现象。

如homework1-3.m（见附录四）文件所示，首先绘制f(x)图像，

x=[-1:0.2:1];

y=1./(1+25\*x.^2); 设置11个插值节点，

随机创建插值节点并且进行分段二次插值，

结果如图3.1所示

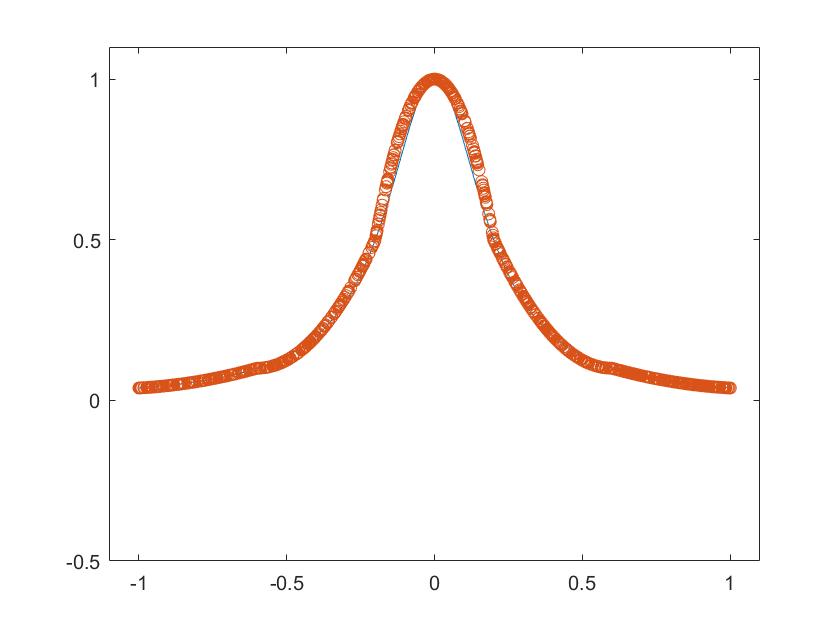


图3.1

图像分析：整体来说误差很小，两边误差几乎可以忽略不计，越靠近中间误差越大。

4、利用 n=10,20,40 次切比雪夫多项式的零点为插值节点，构造 Lagrange插值多项式，在同一坐标系下绘出*f* (*x*)和*L*(*x*)的图像，观察插值多项式是否振荡，并对观察到的现象给予简单的解释。

（1）创建Chebyshev.m文件

定义切比雪夫多项式

function x=Chebyshev(n)

for i=1:n+1

x(i)=cos((2\*(i-1)+1)\*pi/(2\*(1+n)));

end

end

（2）创建homework1-4.m（见附录五）文件，

分别设置n=10，20，40

x10=Chebyshev(10);

x20=Chebyshev(20);

x40=Chebyshev(40);

创建需要的插值点并利用上面的切比雪夫插值节点对这些点进行插值计算，输出结果如图4.1：

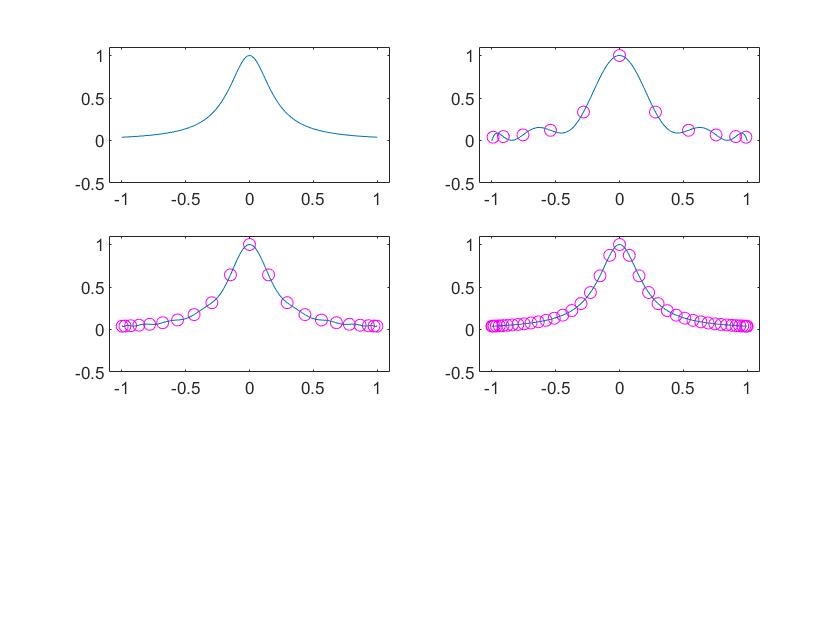


图4.1

图像分析：使用切比雪夫多项式插值节点构造的高次拉格朗日插值多项式，可以改善这种情况，提高插值精度。切比雪夫多项式的零点选择的次数在20次和40次时较为合适，龙格现象基本不存在，误差存在但图像基本吻合。

### 四、结果分析

1、由图可以看出当插值节点数为10时，误差较小，将次数提高到20时，误差变得很大，可以得出当对一个函数在一定区间上取等距节点时，高次Lagrange插值多项式随着次数n的增大，龙格现象越明显。

2、由结果图可看出该插值函数没有发生振荡。

3、由结果图发现相比拉格朗日插值法，利用分段二次插值法对f(x)进行 11个等距节点的插值拟合效果更好，误差更小。可以得出分段线性插值在每个小区间上相对于原函数都有很强的收敛性（舍入误差影响不大），以同样的节点数使用分段线性插值多项式更贴近于被插函数，并且随着次数n的增大，插值函数越贴近被插函数，能很好的解决龙格现象，数值稳定性好等优点。

4、插值多项式没有发生振荡，并且插值节点数越多，误差越小，越贴近被插函数。

### 五、心得体会

通过此次的插值章节习题的训练，加深了我对插值概念的理解和应用，更重要的是训练了我通过使用计算机数据工具，结合面向过程的编程思想，学会解决一些基本的数学问题，锻炼了自己编程的能力，并通过编程加深了对插值概念的理解。

附录：

function p= lagr\_fun(x,y,x0)

%x，y--原节点数据的坐标

%x0--运用插值要计算的点横坐标

n=length(x);

l=ones(1,n);

%根据拉格朗日插值公式含义写出其基函数

for i=1:n

for j=1:n

if i~=j

l(i)=l(i)\*(x0-x(j))/(x(j)-x(i));

end

end

end

p=sum(l.\*y);

end

%%\*\*\*\*\*\*\*\*\*\*\*########作业1.1.1############%%

x0=[-1:0.01:1];

y0=1./(1+25\*x0.^2); %%绘制f(x)图像

plot(x0,y0);

axis([-1.1,1.1,-0.5,1.1]);

hold on;

x=[-1:0.2:1]; %[-1 1]均等分为10份，11个点设为已知插值节点

y=1./(1+25\*x.^2); %11个已知插值节点对应的函数值

plot(x,y,'go', 'MarkerSize',8) %在绘图窗口标出原数据点

X=[-1:0.01:1]; %创建要插值点的

m=length(X);

L=zeros(1,m); %插值点拉格朗插值后的值--空向量

for i=1:m

L(i)=lagr\_fun(x,y,X(i)); %计算出需要插值点的对应的值

end

plot(X,L);

legend('原函数图像','插值节点','十一点十次插值多项式');

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*&&&&&&&&作业1.1.2\*\*\*\*\*\*\*\*\*\*\*\*&&&&&%%

figure

x=[-1:0.1:1]; %[-1 1]均等分为20份，21个点设为已知的插值节点

y=1./(1+25\*x.^2); %21个已知插值节点对应的函数值

plot(x,y,'go', 'MarkerSize',8) %在绘图窗口标出原数据点

X=[-1:0.01:1]; %创建需要的插值点

m=length(X);

L=zeros(1,m); %插值点拉格朗插值后的值--空向量

for i=1:m

L(i)=lagr\_fun(x,y,X(i)); %计算出需要插值点的对应的值

end

plot(X,L,'-b','LineWidth',1);

legend('原函数图像','插值节点','二一点二十次插值多项式');

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*%%%%%%作业1.2\*\*\*\*\*\*\*\*\*\*\*&&&&%%

x0=[-1:0.01:1];

y0=1./(1+25\*x0.^2); %%绘制f(x)图像

plot(x0,y0);

axis([-1.1,1.1,-0.5,1.1]);

hold on;

rand\_x=2\*rand(1,19)-1; %产生19个（-1,1）的随机数--作插值节点

x=[-1,rand\_x,1]; %21个插值节点（附加端点）

y=1./(1+25\*x.^2);

plot(x,y,'go', 'MarkerSize',8);

X=[-1:0.01:1]; %创建需要的插值点

m=length(X);

L=zeros(1,m); %插值点拉格朗插值后的值--空向量

for i=1:m

L(i)=lagr\_fun(x,y,X(i)); %计算出需要插值点的对应的值

end

plot(X,L,'-b','LineWidth',1);

legend('原函数图像','插值节点','随机二一点二十次插值多项式');

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*&&&&&&&&&&--作业1.3---\*\*\*\*\*\*\*\*\*&&&&&&--分段二次拉格朗插值\*\*\*\*\*\*\*\*\*%%

x0=[-1:0.01:1];

y0=1./(1+25\*x0.^2); %%绘制f(x)图像

plot(x0,y0);

axis([-1.1,1.1,-0.5,1.1]);

hold on;

%差值节点

x=[-1:0.2:1]; %分成10等份，11个插值节点

y=1./(1+25\*x.^2); %11个插值节点对应的函数值

rand\_x=2\*rand(1,1000)-1; %随机创建插值点（-1,1）

y0=zeros(1,1000); %承接计算的二次差值点值

for i=1:1000

if(rand\_x(i)<=-0.6 && rand\_x(i)>=-1)

y0(i)=y(1)\*(rand\_x(i)-x(2))\*(rand\_x(i)-x(3))/(x(1)-x(2))/(x(1)-x(3))+...

y(2)\*(rand\_x(i)-x(1))\*(rand\_x(i)-x(3))/(x(2)-x(1))/(x(2)-x(3))+...

y(3)\*(rand\_x(i)-x(1))\*(rand\_x(i)-x(2))/(x(3)-x(1))/(x(3)-x(2));

elseif(rand\_x(i)<=-0.2 && rand\_x(i)>-0.60)

y0(i)=y(3)\*(rand\_x(i)-x(4))\*(rand\_x(i)-x(5))/(x(3)-x(4))/(x(3)-x(5))+...

y(4)\*(rand\_x(i)-x(3))\*(rand\_x(i)-x(5))/(x(4)-x(3))/(x(4)-x(5))+...

y(5)\*(rand\_x(i)-x(3))\*(rand\_x(i)-x(4))/(x(5)-x(4))/(x(5)-x(3));

elseif(rand\_x(i)<=0.2 && rand\_x(i)>-0.20)

y0(i)=y(5)\*(rand\_x(i)-x(6))\*(rand\_x(i)-x(7))/(x(5)-x(6))/(x(5)-x(7))+...

y(6)\*(rand\_x(i)-x(5))\*(rand\_x(i)-x(7))/(x(6)-x(5))/(x(6)-x(7))+...

y(7)\*(rand\_x(i)-x(5))\*(rand\_x(i)-x(6))/(x(7)-x(6))/(x(7)-x(5));

elseif(rand\_x(i)<=0.6 && rand\_x(i)>0.20)

y0(i)=y(7)\*(rand\_x(i)-x(8))\*(rand\_x(i)-x(9))/(x(7)-x(8))/(x(7)-x(9))+...

y(8)\*(rand\_x(i)-x(7))\*(rand\_x(i)-x(9))/(x(8)-x(7))/(x(8)-x(9))+...

y(9)\*(rand\_x(i)-x(7))\*(rand\_x(i)-x(8))/(x(9)-x(8))/(x(9)-x(7));

else

y0(i)=y(9)\*(rand\_x(i)-x(10))\*(rand\_x(i)-x(11))/(x(9)-x(10))/(x(9)-x(11))+...

y(10)\*(rand\_x(i)-x(9))\*(rand\_x(i)-x(11))/(x(10)-x(9))/(x(10)-x(11))+...

y(11)\*(rand\_x(i)-x(9))\*(rand\_x(i)-x(10))/(x(11)-x(10))/(x(11)-x(9));

end

end

scatter(rand\_x,y0);

legend('原函数图像','分段二次插值散点');

%%\*\*\*\*\*\*\*\*\*\*\*&&&&&&&&---作业1.4切比雪夫零点插值----\*\*\*\*\*\*\*&&&&&%%

x10=Chebyshev(10);

x20=Chebyshev(20); %10,20,40次切比雪夫多项式的零点--插值节点

x40=Chebyshev(40);

y10=1./(1+25\*x10.^2);

y20=1./(1+25\*x20.^2); %1计算插值节点对应的函数值

y40=1./(1+25\*x40.^2);

%创建需要的插值点，并利用上面的切比雪夫插值节点对这些插值点进行插值计算

X=[-1:0.01:1]; %创建需要的插值点

m=length(X);

L10=zeros(1,m); %插值点拉格朗插值后的值--空向量

for i=1:m

L10(i)=lagr\_fun(x10,y10,X(i)); %计算出需要插值点的对应的值

end

L20=zeros(1,m); %插值点拉格朗插值后的值--空向量

for i=1:m

L20(i)=lagr\_fun(x20,y20,X(i)); %计算出需要插值点的对应的值

end

L40=zeros(1,m); %插值点拉格朗插值后的值--空向量

for i=1:m

L40(i)=lagr\_fun(x40,y40,X(i)); %计算出需要插值点的对应的值

end

x0=[-1:0.01:1];

y0=1./(1+25\*x0.^2);

subplot(3,2,1);

plot(x0,y0);

axis([-1.1,1.1,-0.5,1.1]);

legend('原函数图像');

subplot(3,2,2);

plot(x10,y10,'mo',X,L10);

axis([-1.1,1.1,-0.5,1.1]);

legend('插值节点','L10图像');

subplot(3,2,3);

plot(x20,y20,'mo',X,L20);

axis([-1.1,1.1,-0.5,1.1]);

legend('插值节点','L20图像');

subplot(3,2,4);

plot(x40,y40,'mo',X,L40);

axis([-1.1,1.1,-0.5,1.1]);

legend('插值节点','L40图像');

%%\*\*\*\*\*\*\*\*\*\*\*&&&&&&&&---作业1.5三次样条插值--函数调用----\*\*\*\*\*\*\*&&&&&%%

x=[-1:0.01:1];

y=1./(1+25\*x0.^2); %%绘制f(x)图像

plot(x0,y0);

axis([-1.1,1.1,-0.5,1.1]);

hold on;

x0=[-1:0.2:1];

y0=1./(1+25\*x0.^2);

X=-1:0.01:1;

Y=interp1(x0,y0,X,'spline');

plot(x0,y0,'ro',X,Y,'b');

xlabel('三次样条插值');

legend('原函数图像','插值节点','三次样条插值');

function x=Chebyshev(n)

for i=1:n+1

x(i)=cos((2\*(i-1)+1)\*pi/(2\*(1+n)));

end

end