**吉林大学数学实验中心实验报告**

**2021年 10月 20日**

|  |
| --- |
| **课程名称：科学计算方法实验实验 题目：拉格朗日插值与牛顿插值**  **姓名：田泽禹 年级专业：2019**  **指导教师：王双**  **算法描述：编制通用Lagrange插值公式，与n次Newton向前、后插值计算公式。**  **相关内容如下**  代码1 (MATLAB, Lagrange)：输入初始插值点与函数值，输入任意x得到近似值  function [fx]=LagIntepo(x,initx,inity)  n=size(initx,2);  initx=x-initx;  coeff=zeros(1,n);  for i=1:n  coeff(i)=coef(initx,i);  end  fx=coeff\*inity';  end  function [ci]=coef(subx,i)  coefx=subx-subx(i);  coefx(i)=1;subx(i)=1;  ci=prod(subx./coefx);  end  运行结果1：    代码2 (C, Lagrange)：  #include<stdio.h>  #include<math.h>  #define Size 20  float prod(double a[]);  int main()  {  int n;  printf("please input integer n\n");  scanf("%d",&n);  double initx[Size],inity[Size];  double x;  printf("please input %d numbers as initial x\n",n+1);  for(int i=0;i<n+1;i++)  {  scanf("%lf",&initx[i]);  }  printf("please input %d numbers as initial y\n",n+1);  for(int i=0;i<n+1;i++)  {  scanf("%lf",&inity[i]);  }  printf("please input x that you want interpolate\n");  scanf("%lf",&x);  double coef[Size];  for(int i=0;i<n+1;i++)  {  initx[i]=x-initx[i];  }  for(int i=0;i<n+1;i++)  {  double temp=1;  for(int j=0;j<n+1;j++)  {  if(j!=i)  {  temp=temp\*initx[j]/(initx[j]-initx[i]);  }  }  coef[i]=temp;  }  double y=0;  for(int i=0;i<n+1;i++)  {  y+=coef[i]\*inity[i];  }  printf("The outcome is: %lf\n",y);  }  运行结果2；    代码3(Newton Forward) 用x^2实验插值效果,n为分点次，a为左端点，t变化0-1，h为间距  function [fxi]=NewtIntepoF(n,a,t,h)  xinit=a:h:a+n\*h;  yinit=fxreal(xinit);  ylist=zeros(n+1,n+1);  ylist(:,1)=yinit';  for k=1:n  ylist(:,k+1)=[ylist(2:n+1,k);0]-ylist(:,k);  end  coeff=ones(1,n+1);  for k=1:n  coeff(k+1)=(t-k+1)/k\*coeff(k);  end  fxi=coeff\*(ylist(1,:))';  end  function [fxr]=fxreal(x)  %fxr=1./(1+25\*x.^2);  fxr=x.^2;  end  运行结果3.    代码4(Newton Backward)  function [fxi]=NewtIntepoB(n,a,t,h)  xinit=a:h:a+n\*h;  yinit=fxreal(xinit);  ylist=zeros(n+1,n+1);  ylist(:,1)=yinit';  for k=1:n  ylist(:,k+1)=ylist(:,k)-[0;ylist(1:n,k)];  end  coeff=ones(1,n+1);  for k=1:n  coeff(k+1)=(t-k+1)/k\*coeff(k);  end  fxi=coeff\*(ylist(n+1,:))';  end  function [fxr]=fxreal(x)  %fxr=1./(1+25\*x.^2);  fxr=x.^2;  end  运行结果4    代码4(Newton General) initx，inity为初始值  function [fx]=NewtIntepo(x,initx,inity)  n=size(initx,2)-1;  initx=x-initx;  coeff=zeros(1,n+1);  ylist=zeros(n+1,n+1);  ylist(:,1)=inity';  for k=1:n  ylist(1:n+1-k,k+1)=(ylist(2:n+2-k,k)-ylist(1:n+1-k,k))./((initx(1:n+1-k)-initx(k+1:n+1))');  end  for i=2:n+1  coeff(i)=prod(initx(1,1:i-1));  end  coeff(1)=1;  fx=coeff\*ylist(1,:)';  end  function [ci]=coef(subx,i)  coefx=subx-subx(i);  coefx(i)=1;subx(i)=1;  ci=prod(subx./coefx);  end  运行结果4    代码5(Newton, syms)  function [fx]=NewtIntepoN(x,initx,inity)  n=size(initx,2)-1;  initx=x-initx  coeff=zeros(1,n+1);  ylist=zeros(n+1,n+1);  ylist(:,1)=inity';  for k=1:n  ylist(1:n+1-k,k+1)=(ylist(2:n+2-k,k)-ylist(1:n+1-k,k))./((initx(1:n+1-k)-initx(k+1:n+1))');  end  fx=ylist(1,1);  ylist(:,:)  for i=2:n+1  fx=fx+prodxx(initx,i-1)\*ylist(1,i);  end  end  function [ci]=coef(subx,i)  coefx=subx-subx(i);  coefx(i)=1;subx(i)=1;  ci=prod(subx./coefx);  end  function [y]=prodxx(initx,i)  y=1;  for k=1:i  y=initx(k)\*y;  end  end  运行结果5    下面我们将Newton General带入验证，即17.(2)    初始化n=5    取自变量    Y的估计值    n=10时      Lagrange 插值法 n=5 时      Lagrange 插值法 n=10 时      可见都远不准确  **总结**  无论是Lagrange插值公式，或者Newton插值公式，其都是多项式插值公式。Lagrange插值公式形式清晰，对称性明显，并且易于理解。但是在变样本数的插值过程中，却需要每一次重新计算系数。但是Newton插值仅需要在添加新的点后加上新的项，因此在此方面远远优于Lagrange插值。  但是两种插值方法都是多项式插值，对于全局上若使用，则在插值点较少时精度不足，插值点过多时产生剧烈震荡。因此如例题中[-1,1]的区间较大，插值误差无法得到有效控制。一般插值会在小区间上进行，然后拼成大定义域上的函数。得到准确的插值估计。 |