**吉林大学数学实验中心实验报告**

**2021年 11月 1日**

|  |
| --- |
| **课程名称：科学计算方法实验实验 题目：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Newton插值公式。**  **相关内容如下**  代码1 (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1    代码2 (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2    代码3 (C)  #include<stdio.h>  #include<math.h>  #define SIZE 20  double prod(double a[],int k);  int main()  {  //initial size n  int n;  printf("please initialize size n<20;\n");  scanf("%d",&n);    //input initx and inity  printf("please initialize x (ascending) (size=%d)\n",n);  double initx[SIZE];  for(int i=0;i<n;i++)  {  scanf("%lf",&initx[i]);  }    printf("please initialize y (size=%d)\n",n);  double inity[SIZE];  for(int i=0;i<n;i++)  {  scanf("%lf",&inity[i]);  }  //initialize x;  printf("input x where you want approximate\n");  double x;  scanf("%lf",&x);    //process x  for(int i=0;i<n;i++)  {  initx[i]=x-initx[i];  }    //initialize Ylist  double Ylist[SIZE][SIZE]={0};  for(int i=0;i<n;i++)  {  Ylist[i][0]=inity[i];  }  //Difference  for(int i=1;i<n;i++)  {  for(int j=0;j<n-i;j++)  {  Ylist[j][i]=-(Ylist[j+1][i-1]-Ylist[j][i-1])/(initx[j+i]-initx[j]);  }  }  double finaly=1;  finaly\*=Ylist[0][0];  for(int i=1;i<n;i++)  {  finaly+=prod(initx,i)\*Ylist[0][i];  }    printf("The value is %lf",finaly);  }  double prod(double a[],int k)  {  double product=1;  for(int i=0;i<k;i++)  {  product\*=a[i];  }  return product;  }  运行结果3 (C)    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      **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。 |