计算方法第一次上机报告

电子科学与工程学院 06220143 顾豪阳

# 一、题目简介

请用Mathematica外任何一种编程语言，如MATLAB，C++，Fortran，Python等实现第一个拓展算法，并通过数值算例等验证你的程序，在提交的实验报告中，请给出算法的严格理论分析以及工作量分析。

第一个拓展算法：求多项式在一点处的导数值。

# 二、理论分析

对于多项式：

利用Horner算法，可以按照下面的形式进行运算：

由上式可得：

代入第一条式子可以得出：

记

则有：

对上式求导：

代入可得：

因此只需求解。

通过观察又可以得出中的系数即为第一次求多项式值得出的。也就是说再用一次Horner算法即可得到导数值

# 三、数值实验过程

以书本P58案例为例：

用Horner算法计算。

**解：**首先

利用：

求解出：

由此得到了。

用代替，用代替，且其值等于。

得出结果为876.

# 四、程序代码与结果

#include<iostream>

#include<cstring>

using namespace std;

void Horner(int \*p,int x,int a,int n);//求解f(x)

void Horner\_derivation(int \*p,int x,int a,int n);//求解f'(x)

void Horner\_plus(int \*p,int x,int \*a,int n);//当a为Newton型基底（顺便做下拓展2）

int main(){

//书本P58例7

int p[]={2,0,1,2,9,7};//系数升序排列

int n=sizeof(p)/sizeof(int);

Horner(p,3,1,n); //Horner(系数数组p,自变量x,展开的点a,系数数组长度n)

Horner\_derivation(p,3,1,n);//同上

//书本P62例8

int a[]={1,2,3};

int p1[]={1,9,8,4};

Horner\_plus(p1,-1,a,sizeof(p1)/sizeof(int));

system("pause");

return 0;

}

void Horner(int \*p,int x,int a,int n)

{

int pnew[100];//C++麻烦的数组拷贝

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

pnew[i]=p[i];

}

int xnew=x;

int i;

for(i=n-2;i>=0;i--){

pnew[i]=pnew[i+1]\*(xnew-a)+pnew[i];

}

cout<<"f("<<xnew<<")的值为：";

cout<<pnew[0]<<endl;

}

void Horner\_derivation(int \*p,int x,int a,int n)

{

int pnew[100];//C++麻烦的数组拷贝

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

pnew[i]=p[i];

}

int xnew=x;

int i;

for(i=n-2;i>=0;i--){

pnew[i]=pnew[i+1]\*(xnew-a)+pnew[i];

}

for(int j=n-2;j>=1;j--){

pnew[j]=pnew[j+1]\*(xnew-a)+pnew[j];

}

cout<<"f'("<<x<<")的值为：";

cout<<pnew[1]<<endl;

}

void Horner\_plus(int \*p,int x,int \*a,int n){

int pnew[100];//C++麻烦的数组拷贝

int anew[100];

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

pnew[i]=p[i];

}

for(int i=0;i<n-1;i++){

anew[i]=a[i];

}

int xnew=x;

int i;

for(i=n-2;i>=0;i--){

pnew[i]=pnew[i+1]\*(xnew-a[i])+pnew[i];

}

cout<<"f("<<xnew<<")的值为：";

cout<<pnew[0]<<endl;

for(int j=n-2;j>=1;j--){

pnew[j]=pnew[j+1]\*(xnew-a[j-1])+pnew[j];

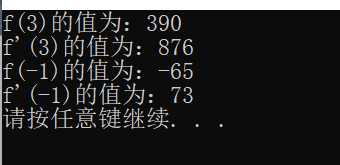
}

cout<<"f'("<<xnew<<")的值为：";

cout<<pnew[1]<<endl;

}

实验结果算得：



和书上的结论吻合。

# 五、对实验的分析

优点：

运用C++进行运算，电脑程序运行效率高。

采用了书本中“覆盖”的实现方式，节省存储。

并且代码可移植性强，通过上面的求一阶导函数可以发现，反复实现Horner算法的代码即可求解多阶导数。

通过传递数组，轻松就能实现newton基底的horner算法。

缺点：

C++的数组存储有限，不能把数组作为形参传入函数，只能传指针。因此可迭代的容量有限。

计算效率高不可避免就会导致代码编写麻烦，而且传的参数过多。

封装性太差，不方便做后续的拓展。

# 六、结论和感想

Horner算法中的就是商多项式的系数，可以通过对再用一次Horner算法得到导数值，而经过多次Horner算法可以得到的高阶导数值。