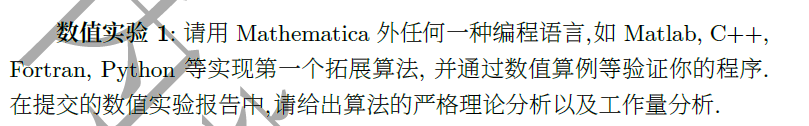
## Chpter1上机报告

1. **题目**

Chapter1上机选择2：Horner算法的拓展1，求多项式在某一点的导数值

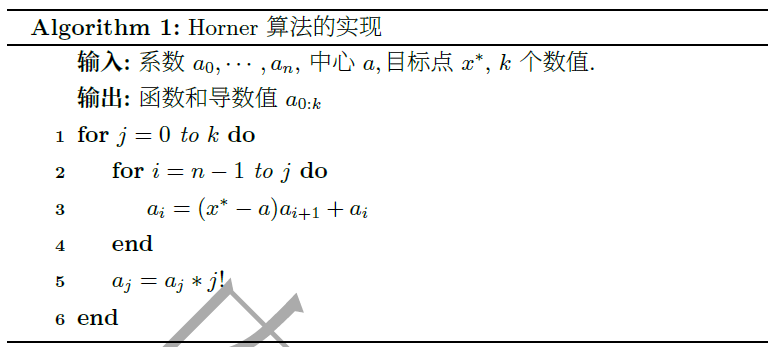


1. **分析及解法**

由上机文档中的讨论可知函数在某一点的k阶导数的导数值可以由下式得到：



其中*f*n-k是对函数*f*进行Horner算法后，每次取其商多项式构成新的函数进行Horner算法，持续n-k次得到的商多项式，且每次进行horner算法后系数的数量会减少一个。因此可以得到其实现的过程如下：



共需要n+(n-1)+…+(n-k)=kn+0.5k(k+1)次加法，n+0.5k(k+1)+k+1次乘法

1. **程序以及运行结果 （采用C++）**

#include<iostream>

using namespace std;

int\* hornerRule(int list[],int m,int x0, int a); //Horner法则求多项式值

int jiecheng(int m) //求阶乘

{

int k=1;

for(;m>1;m--)

{k = m\*k;}

return k;

}

int main()

{

int i,n,x0,a;

int list[]= {1, 2, 3, 4, 5}; //存放系数an

n = sizeof(list) / sizeof(\*list)-1; //注意已经减1了

cout<<"x0 的值:"<<endl;

cin>>x0;

cout<<"a 的值"<<endl;

cin>>a;

int\* m = hornerRule(list,n,x0,a);

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

{

cout << \*(m + i) << endl; //输出导数值

}

system("pause");

return 0;

}

int\* hornerRule(int list[],int n,int x0, int a)

{

int i,j,k; //k表示第k阶导数的值

cout<<"需要求导数的阶数k：";

cin>>k;

for(j=0;j<=k;j++)

{

for(i=n-1;i>=j;i--)

{

list[i]=(x0-a)\*list[i+1]+list[i]; //bn的函数形式

}

list[j]=list[j]\*jiecheng(j);

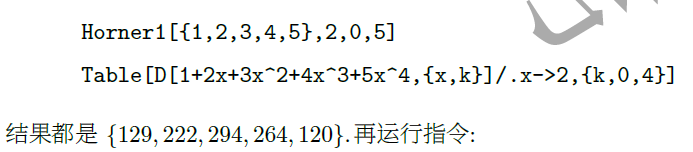
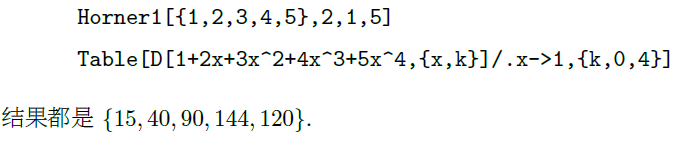
}

return list;

}

**运行结果：**

**采用上机文档中的两个算例进行验证，均满足要求**

 ****

1. list[]= {1, 2, 3, 4, 5}（2）list[]= {1, 2, 3, 4, 5}

x0=2 a=0 k=5 x0=2 a=1 k=5

