第二次实验

1. 差商和牛顿插值多项式

*f* (*x*)在的一阶差商为：

一般使用下面符号表示*f* (*x*)在*xi*、*xj*上的一阶差商：



推广可得，*f* (*x*) 在的*n*阶差商为：



差商具有如下性质：



* **而牛顿插值多项式可表示为：**



显然，要得到牛顿插值多项式只要计算1阶、2阶、…、*n*阶差商即可

* 差商有两种算法：

**（1）方法1**

用上面**的性质**计算。即：



**（1）方法2**

用下面的表格计算。



利用*f* (*x*)在的函数值先计算一阶差商，再用一阶差商计算二级差商，……

例如计算二姐差商，用公式 依次类推。

* 实验题目1

编程实现牛顿插值法，并用后面的例子中的数据验证。

已知为插值点，求牛顿插值法在*xx*位置的近似值。

【利用**方法1**的算法】

//k阶差商，即求*ai*

double ChaShang(double \*x, double \*y, int k)

{

定义k阶差商结果f=0; //差商初始值

for(int i=0;i<k+1;i++)

{

定义子项初始值temp=y[i]; //第i子项分子

利用temp和x坐标计算**第i子项**数据; //即方法1公式累加和中的**第i子项**

将**第i子项**累加到f;

}

return f;

}

double Newton(double \*x, double \*y,double xx, int n){

定义前n项数据result=0;

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

计算i阶差商，即求*ai*; //系数

计算存于temp；

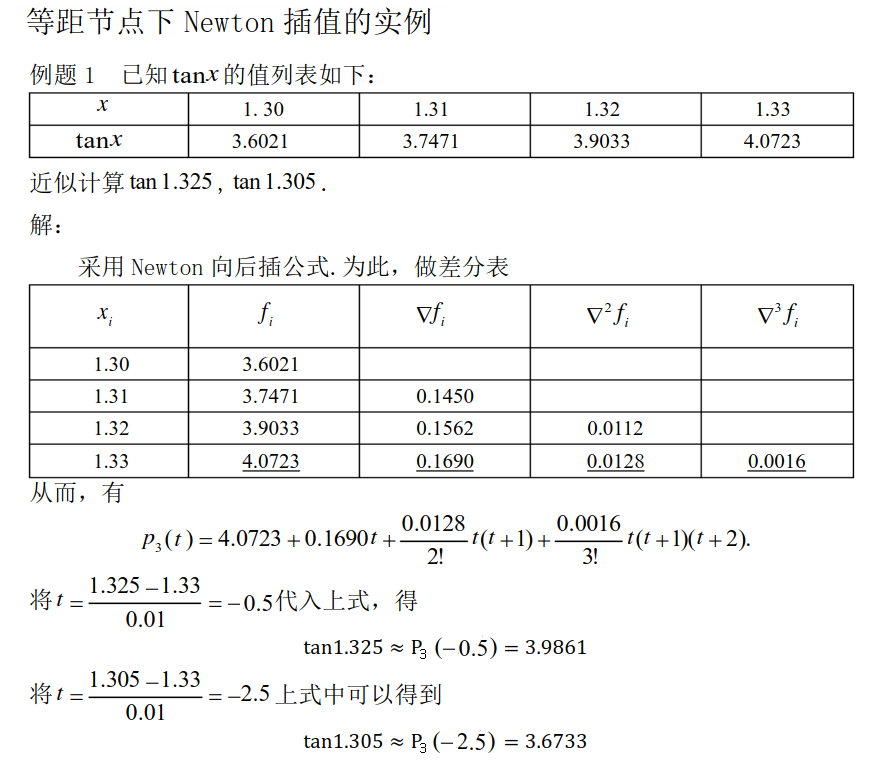
将牛顿插值多项式的一项*ai\**temp累加到result中;

}

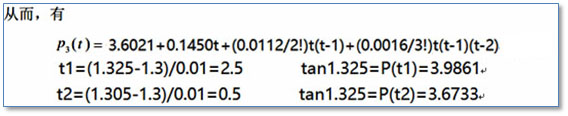
return result;

}

本题目也可以利用方法2计算差商，方法1、方法2实现一种即可。



以上例子是以最后点1.33为基准做的P(t)，和课本上稍有不同，下面是按教材方法写的P(t)，按对角线元素计算系数。两个P(t)计算结果是一样的.



1. 二分法求解方程的根

二分法的基本思想是逐步将非线性方程*f* (*x*)=0的有根区间二分，通过判断函数值的符号，逐步对半缩小有根区间，直到区间缩小到容许误差范围之内，然后取区间的中点为根的近似值。

* 实验题目2

用二分法求解5sin(x)-x/2-1=0

【算法】

double f(float x)

{

定义方程左侧表达式;

}

int main()

{

定义变量a,b;// 输入求解区间

定义求解区间中点mid;

定义中间变量fa,fmid,fb;

输入求解区间a和b;

if(f(a)\*f(b)>0) {

异常处理退出;

}

//利用二分法求解

while (a和b区间是否足够小)

{

计算a和b的中间点mid;

if (a和min的函数值同号)

a=mid;

else

b=mid;

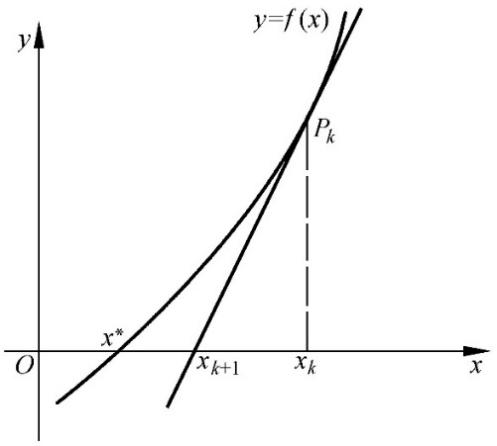
}

输出计算结果;

}

1. 牛顿迭代法求方程的根

设是根的某个近似值，过曲线上横坐标为的点作切线，并将该切线与轴的交点的横坐标作为的新的近似值。如图所示。



下列公式是牛顿迭代方程：

， 

* 实验题目3

用牛顿迭代法求e*x* + *x* = 0等于0.5附近的根。

【算法】

float f(float x){

定义*f* (*x*)表达式;

}

float df(float x){

定义表达式*f* (*x*)的导数;

}

int main(void)

{

定义变量x,x0;

假设从0.5开始迭代;

do{

x0=x;

根据牛顿迭代法计算下一次结果;

}while(判断x-x0是否足够小);

输出计算结果;

return 0 ;

}

1. 一元线性回归（选作）

将书上一元线性回归的程序输入电脑运行。对程序中for循环的作用加注释。

【算法】

int main()

{

定义数组数据x和y;

定义数据个数count;

定义中间其他变量;

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

{

对x轴样本数据求和;

对y周样本数据求和;

}

对x轴样本数据求平均值;

对y周样本数据求平均值;

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

{

根据公式计算Lxy;

根据公式计算Lxx;

}

// 计算系数

b=Lxy/Lxx;

a=y\_avg-b\*x\_avg;

输出计算机结果;

}