

**学 生 实 验 报 告**

**（理工类）**



课程名称： 计算方法 专业班级： 15软件工程（Z）

学生学号： 1512001066 学生姓名： 吴跟强

所属院部： 软件工程学院 指导教师：

**20 17 ——20 18 学年 第 1 学期**

金陵科技学院教务处制

**实验报告书写要求**

实验报告原则上要求学生手写，要求书写工整。若因课程特点需打印的，要遵照以下字体、字号、间距等的具体要求。纸张一律采用A4的纸张。

**实验报告书写说明**

实验报告中一至四项内容为必填项，包括实验目的和要求；实验仪器和设备；实验内容与过程；实验结果与分析。各院部可根据学科特点和实验具体要求增加项目。

**填写注意事项**

（1）细致观察，及时、准确、如实记录。

（2）准确说明，层次清晰。

（3）尽量采用专用术语来说明事物。

（4）外文、符号、公式要准确，应使用统一规定的名词和符号。

（5）应独立完成实验报告的书写，严禁抄袭、复印，一经发现，以零分论处。

**实验报告批改说明**

实验报告的批改要及时、认真、仔细，一律用红色笔批改。实验报告的批改成绩采用百分制，具体评分标准由各院部自行制定。

**实验报告装订要求**

实验批改完毕后，任课老师将每门课程的每个实验项目的实验报告以自然班为单位、按学号升序排列，装订成册，并附上一份该门课程的实验大纲。

实验项目名称： 方程求根 实验学时： 2

同组学生姓名： 实验地点： 1514

实验日期： 11.28 实验成绩：

批改教师： 批改时间：

一、实验目的和要求

实验目的：

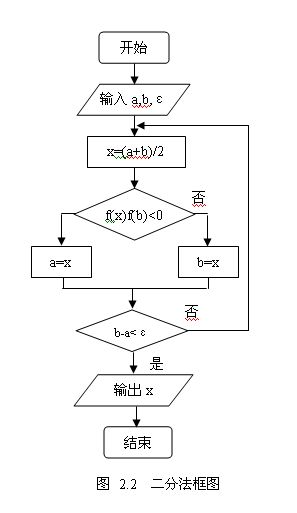
1. 通过对二分法、牛顿法、割线法作编程练习，进一步体会它们各自不同的特点；
2. 了解二分法，切线法，割线法。
3. 能熟练运用二分法，牛顿法进行方程求根
4. 通过上机调试运行，对方程求根的几种方法程序进行改进。

实验要求：

1. 上机前作好充分准备，包括复习编程所需要的语言工具。
2. 上机时要遵守实验室的规章制度，爱护实验设备。
3. 记录调试过程及结果，记录并比较与手工运算结果的异同。
4. 程序调试完后，须由实验辅导教师在机器上检查运行结果。
5. 给出本章实验单元的实验报告。

二、实验仪器和设备

1. 硬件设备：IBM PC以上计算机，有硬盘和一个软驱、单机和网络环境均可。
2. 软件环境： C语言运行环境。



三、实验原理、方法

**二分算法计算步骤：**

（1）输入有根区间的端点a、b及预先给定的精度ε；

（2）计算中点x=(a+b)/2；

（3）若f(x)f(b)<0，则a=x，转向下一步；否则b=x，转向下一步；

（4）若b-a<ε，则输出方程满足精度要求的根x，结束；否则转向步骤（2）。

**迭代法：**

图 2.3 迭代法框图

开始

输入

x0, ε,N

k=1

结束

是

x1=(x0)

≤ε

k<N?

否

否

k=k+1

x0=x1

是

输出迭代失败标志

输出近似根x1

图2 牛顿法框图

开始

输入x0, å,N

k=1

=0?

否

是

结束

是

≤å

k<N?

否

否

k=k+1

x0=x1

是

输出奇异标志

输出迭代失败标志

输出近似根x1



**牛顿法：**

牛顿迭代法是一种逐步线性化方法，即将非线性方程f(x)=0的求根问题归结为计算一系列线性方程的根。

设xk是方程f(x)=0的一个近似根，将f(x)在xk处作一阶泰勒展开，即

f(x)≈f(xk)+f′(xk)(x- xk)

于是得到如下的近似方程

f(xk)+f′(xk)(x- xk)=0 （2.7）

设f′(xk)≠0，则式（2.7）的解为



取x作为原方程的新的近似根xk+1，即令

　　　k=0,1,2, … 　　 （2.8）

则称式（2.8）为牛顿迭代公式。用牛顿迭代公式（2.8）求方程近似根的方法称为牛顿迭代法，简称牛顿法，又称切线法。

四、实验内容

1. 以方程：x3-0.2x2-0.2x-1.2=0为例，编写程序求方程的根
2. 编写二分法、迭代法、牛顿法程序，分析运行结果。
3. 对用这两种方法求解出的根进行对比分析

五、实验过程

二分法程序：

#include "stdio.h"

#include "math.h"

double f(double x)

{

double sum;

sum=x\*x\*x-0.2\*x\*x-0.2\*x-1.2;

return sum;

}

double judge(double a,double b,double c)

{

double x;

x=(a+b)/2;

while(fabs(b-a)>=c)

{

double flag=f(x)\*f(b);

if(flag<0)

a=x;

else

b=x;

x=(a+b)/2;

}

return x;

}

int main()

{

double a,b,c,x=0.0;

printf("请输入有根区间a、b及预先给定的精度c：");

scanf("%lf,%lf,%lf",&a,&b,&c);

x=judge(a,b,c);

printf("满足精度要求的根x=%lf",x);

return 0;

}

牛顿迭代法程序：

#include "stdio.h"

#include "math.h"

double f(double x)

{

double sum;

sum=x\*x\*x-0.2\*x\*x-0.2\*x-1.2;

return sum;

}

double f1(double x)

{

double sum1;

sum1=3\*x\*x-0.4\*x-0.2;

return sum1;

}

int main()

{

double x0,c,x1;

int N=0,i=1;

printf("请输入x0、N及预先给定的精度c：");

scanf("%lf,%d,%lf",&x0,&N,&c);

while(f1(x0)!=0&&fabs(x1-x0)>c&&i<N)

{

x0=x1;

x1=x0-f(x0)/f1(x0);

i++;

}

if(fabs(x1-x0)<=c)

printf("满足精度要求的根x=%lf",x1);

if(i==N)

printf("不符合要求");

if(f1(x0)==0)

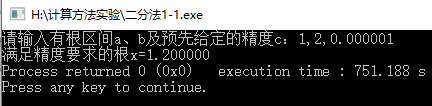
printf("x=%lf",x0);

return 0;

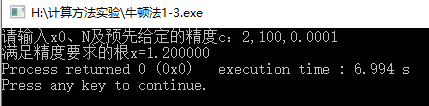
}

六、实验结果与分析

二分法程序结果：



牛顿迭代法程序结果：



对用这两种方法求解出的根进行对比分析

答：结果都是1.2这个根，根据运行时间来看，牛顿迭代法比二分法时间要节省很多，效率更高。

七、实验感想

根据这次实验，对二分法和牛顿迭代法更加的清楚了，对方法的运用更加熟悉，也知道了方法的使用会影响结果的精确度，牛顿法比二分法考虑的要全面些。