计 算 方 法

实验四 Newton迭代法

|  |  |
| --- | --- |
| 姓名 | 郭茁宁 |
| 学号 | 1183710109 |
| 院系 | 计算机科学与技术学院 |
| 专业 | 软件工程 |

哈尔滨工业大学

# 实验报告四

# 题目（摘要）

利用牛顿迭代法求的根

输 入：初值，精度，最大迭代次数

输 出：方程根的近似值或计算失败标志

# 前言（目的和意义）

目的：利用牛顿迭代法求的根

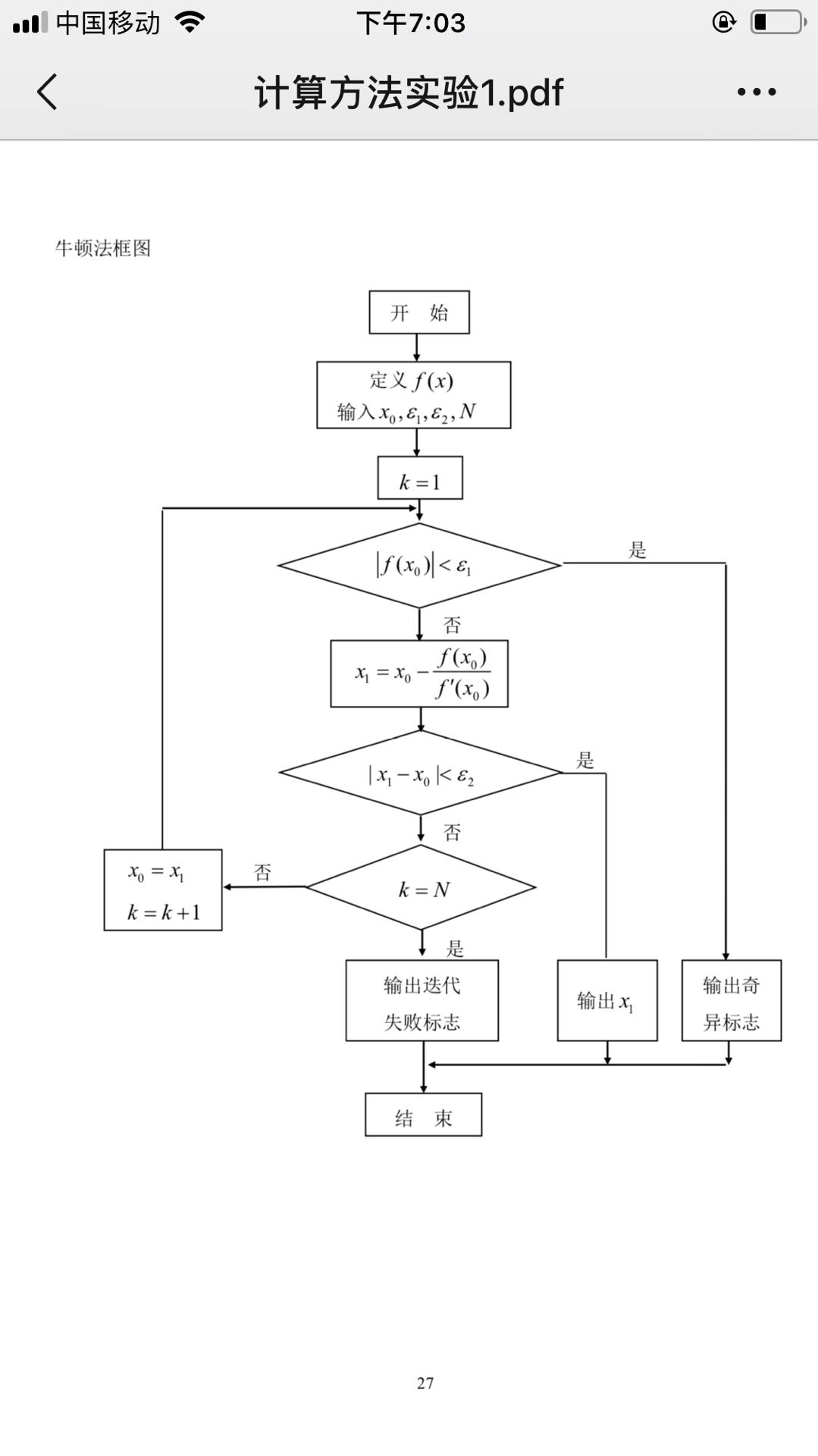
意义：学习根据实际问题建立的数学模型，针对数学模型的特点确定适当的计算方法，编制出计算机能够执行的计算程序，输入计算机，进行调试，完成运算等数值计算的过程。不只会套用教科书中的标准程序进行数值计算，独立地将学过的数值算法编制成计算机程序，灵活应用已经掌握的算法求解综合性较大的课题。理解数值计算程序结构化的思想，提高编程能力，加深对“计算方法”课程内容的理解和掌握，进一步奠定从事数值计算工作的基础。具体可以利用所掌握的“高级语言”顺利地编制出计算机程序，上机实习，完成实验环节的教学要求。不简单地套用现成的标准程序完成实验题目，把重点放在对算法的理解、程序的优化设计、上机调试和计算结果分析上，达到实验课的目的。

# 数学原理

求非线性方程的根，牛顿迭代法计算公式

一般地，牛顿迭代法具有局部收敛性，为保证迭代收敛，要求，对充分小的，。如果，，，那么，对充分小的，当时，由牛顿迭代法计算出的收敛于，且收敛速度是2阶的；如果，，，那么，对充分小的，当时，由牛顿迭代法计算出的收敛于，且收敛速度是1阶的；

# 程序设计流程



#include <cmath>

#include <cstdio>

#include <iostream>

using namespace std;

double x, e1, e2;

int n;

double f(double *x*) { return cos(x) - x; }

double df(double *x*) { return -sin(x) - 1; }

int main() {

    scanf("%lf%lf%lf%d", &x, &e1, &e2, &n);

    for (int i = 1; i <= n; i++) {

        double F = f(x), DF = df(x);

        if (fabs(F) < e1) {

            printf("%lf", x);

            return 0;

        }

        if (fabs(DF) < e2) {

            printf("Failed");

            return 0;

        }

        double x1 = x - F / DF;

        double tol = fabs(x - x1);

        if (tol < e1) {

            printf("%lf", x1);

            return 0;

        }

        x = x1;

    }

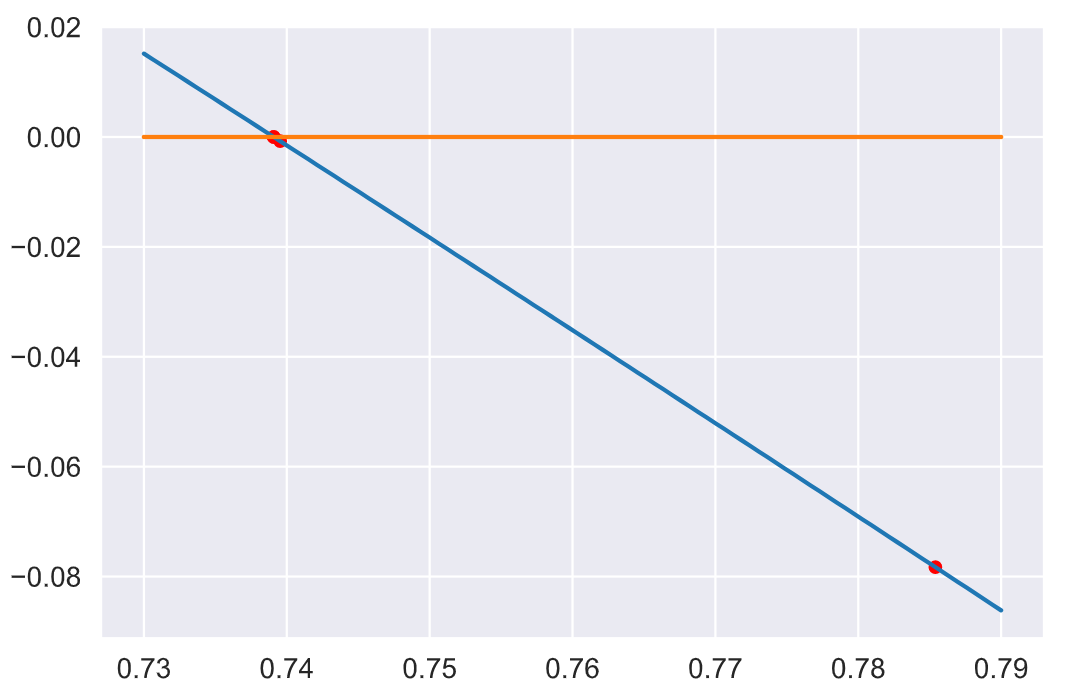
    printf("Failed");

    return 0;

}

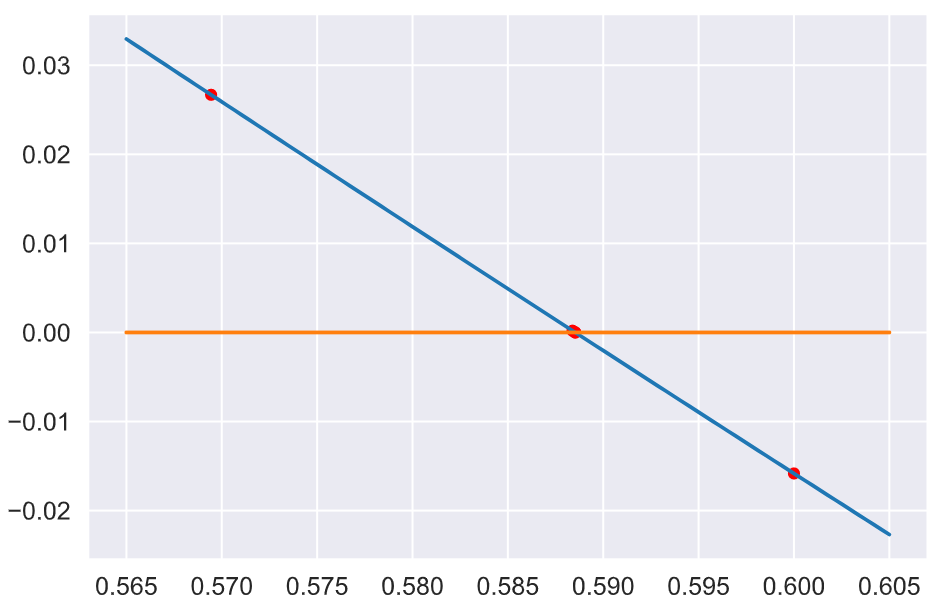
# 实验结果、结论与讨论

1. ，，，，



解得

1. ，，，，



解得

## 思考题：对实验1 确定初值的原则是什么？实际计算中应如何解决？

在的根附近任取一点作为初始近似根。

由于Newton法具有局部收敛性，所以在实际问题中，当提供了接近于根的初始值位置即可使用，其可确保收敛。当初值难以确定时，迭代序列就不一定收敛，要通过如二分法等算法确定近似值，再将该近似值作为初值，保证收敛性。