计 算 方 法

实验五 Gauss列主元消去法

|  |  |
| --- | --- |
| 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

# 实验报告五

# 题目（摘要）

利用Gauss列主元消去法、显式相对Gauss列主元消去法、隐式相对Gauss列主元消去法求解线性方程组。

输 入：；， ，

输 出：线性方程组的近似解，

# 前言（目的和意义）

目的： 利用Gauss列主元消去法、显式相对Gauss列主元消去法、隐式相对Gauss列主元消去法求解线性方程组。

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

高斯（Gauss）列主元消去法：对给定的阶线性方程组，首先进行列主元消元过程，然后进行回代过程，最后得到解或确定该线性方程组是奇异的。

如果系数矩阵的元素按绝对值在数量级方面相差很大，那么，在进行列主元消元过程前，先把系数矩阵的元素进行行平衡：系数矩阵的每行元素和相应的右端向量元素同除以该行元素绝对值最大的元素。这就是所谓的平衡技术。然后再进行列主元消元过程。

如果真正进行运算去确定相对主元，则称为显式相对Gauss列主元消去法；如果不进行运算，也能确定相对主元，则称为隐式相对Gauss列主元消去法。

显式相对Gauss列主元消去法：对给定的阶线性方程组，首先进行列主元消元过程，在消元过程中利用显式平衡技术，然后进行回代过程，最后得到解或确定该线性方程组是奇异的。

隐式相对Gauss列主元消去法：对给定的阶线性方程组，首先进行列主元消元过程，在消元过程中利用隐式平衡技术，然后进行回代过程，最后得到解或确定该线性方程组是奇异的。

Gauss列主元消去法

1对，做1.1—1.3，消元过程

1.1 寻找最小的正整数，和。如果，输出奇异标志，停机；

1.2 如果，那么交换两行；

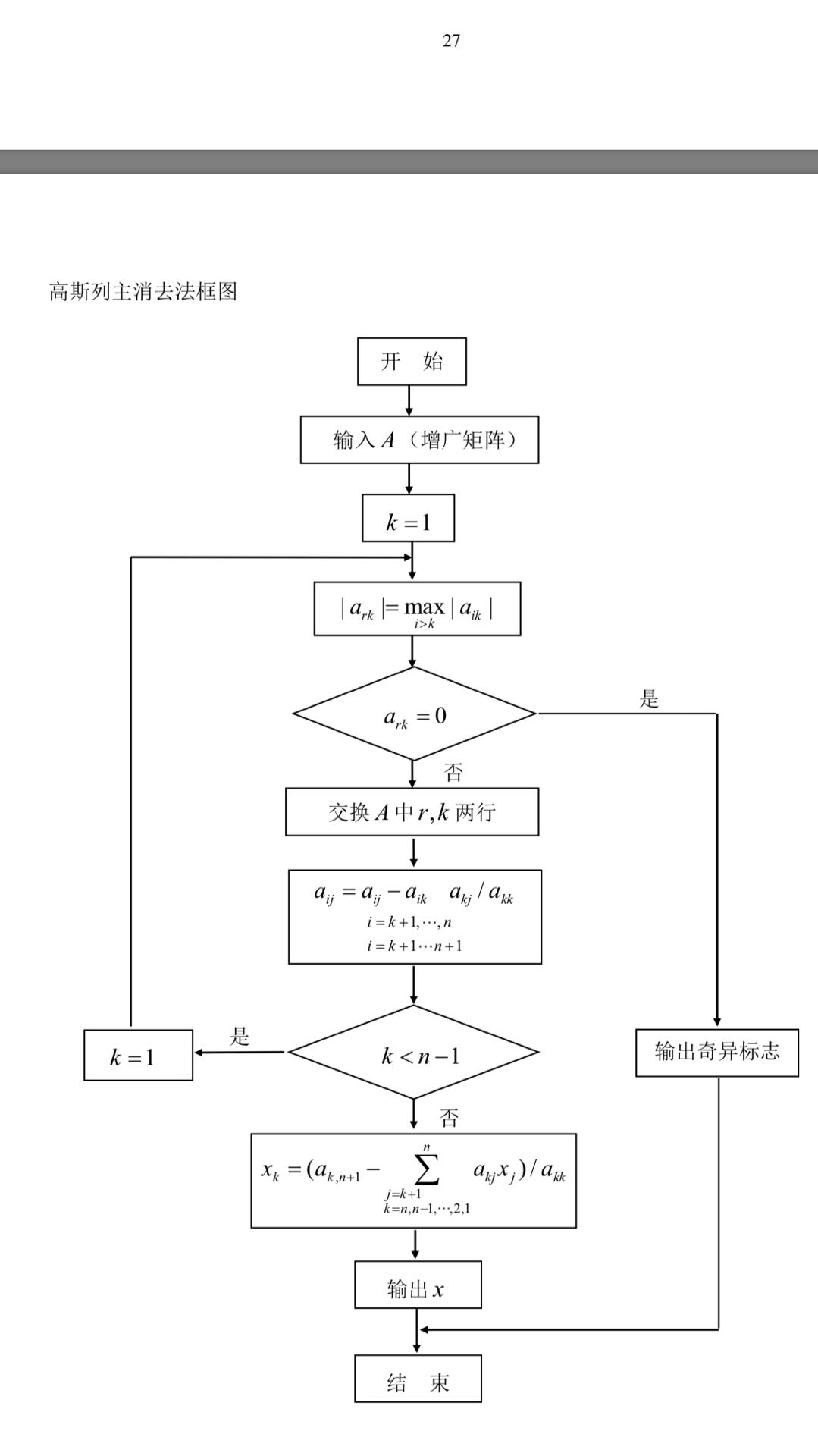
1.3 对，记，计算

2. 如果输出奇异标志，停机；

3. 置 ，回代过程

4. 对，置

# 程序设计流程



#include <cmath>

#include <cstdio>

#include <iostream>

using namespace std;

#define N 10

int n;

double a[N][N], b[N], x[N];

int main() {

    scanf("%d", &n);

    for (int i = 1; i <= n; i++)

        for (int j = 1; j <= n; j++) scanf("%lf", &a[i][j]);

    for (int i = 1; i <= n; i++) scanf("%lf", &b[i]);

    for (int k = 1; k < n; k++) {

        int p = k;

        double maxabs = fabs(a[k][k]);

        for (int j = k + 1; j <= n; j++)

            if (fabs(a[j][k]) - maxabs > 0) {

                p = j;

                maxabs = fabs(a[j][k]);

            }

        if (a[p][k] == 0) {

            printf("Singular");

            return 0;

        }

        if (p != k) {

            double tmp;

            for (int j = 1; j <= n; j++) {

                tmp = a[p][j];

                a[p][j] = a[k][j];

                a[k][j] = tmp;

            }

            tmp = b[p];

            b[p] = b[k];

            b[k] = tmp;

        }

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

            double m\_ik = a[i][k] / a[k][k];

            for (int j = k + 1; j <= n; j++) a[i][j] -= a[k][j] \* m\_ik;

            b[i] -= b[k] \* m\_ik;

        }

    }

    if (a[n][n] == 0) {

        printf("Singular");

        return 0;

    }

    x[n] = b[n] / a[n][n];

    for (int k = n - 1; k >= 1; k--) {

        double sigma = 0.0;

        for (int j = k + 1; j <= n; j++) sigma += a[k][j] \* x[j];

        x[k] = (b[k] - sigma) / a[k][k];

    }

    for (int i = 1; i <= n; i++) printf("%lf\t", x[i]);

    return 0;

}

# 实验结果、结论与讨论

(1)

解得

准确结果为

(2)

解得

准确结果为

(3)

解得

准确结果为

(4)

解得

准确结果为

## 思考题：计算实验1、实验2的各个题目说明：对什么类型的线性方程组三种方法是一致的？

在各主元不是非常小、且数量级差别不大的时候，三种方法结果一致。