Лабораторная работа №1

Факультет прикладной математики и информатики

Кафедра вычислительной математики

**«Метод Гаусса»**

**Преподаватель:** Никифоров Иван Васильевич

**Студент:** Шелег Владислава Михайловна

Минск, 2016

**Постановка задачи**

1. Реализовать программу решения системы линейных алгебраических уравнений (СЛАУ) вида методом Гаусса с выбором главного элемента по столбцу.

где ,(для данного варианта ),

а – одномерный вектор правой части.

**Теория**

Пусть дана система уравнений

Прямой ход метода Гаусса состоит в том, чтобы привести данную систему к виду

с помощью следующих рекуррентных формул:

.

Обратный ход метода Гаусса позволяет последовательно найти неизвестные, используя соотношения

**Листинг программы**

(Программа реализована на языке Java)

**package** vma\_lab1;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**public** **class** lab1 {

**public** **static** **void** main(String[] args) **throws** IOException {

**double**[][] Matrix = { { 7, 2.5, 2, 1.5, 1 }, { 2.5, 8, 2.5, 2, 1.5 },

{ 2, 2.5, 9, 2.5, 2 }, { 1.5, 2, 2.5, 10, 2.5 },

{ 1, 1.5, 2, 2.5, 11 }

};

**double**[] Column = { 1, 1, 1, 1, 1 };

BufferedReader in = **new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("Enter coefficient:");

**int** lambda = Integer.*parseInt*(in.readLine());

in.close();

**for** (**int** i = 0; i < Matrix.length; i++){

Matrix[i][i] += lambda;

}

//Print matrix

**for** (**int** i = 0; i < Matrix.length; i++) {

**for** (**int** j = 0; j < Matrix[i].length; j++) {

System.***out***.printf("%5.4f ", Matrix[i][j]);

}

System.***out***.printf("| %5.4f\n", Column[i]);

}

System.***out***.println();

**double**[] x = *Solve*(Matrix, Column);

System.***out***.println("x:");

**for** (**double** X : x) {

System.***out***.printf("%.4f ", X);

}

}

**private** **static** **double**[] Solve(**double**[][] mat, **double**[] col) {

**for** (**int** i = 0; i < col.length; i++) {

**int** maximum = i;

**for** (**int** j = i + 1; j < col.length; j++) {

**if** (Math.*abs*(mat[j][i]) > Math.*abs*(mat[maximum][i])) {

maximum = j;

}

}

**double**[] temp1 = mat[i];

mat[i] = mat[maximum];

mat[maximum] = temp1;

**double** temp2 = col[i];

col[i] = col[maximum];

col[maximum] = temp2;

// a straight course

**for** (**int** j = i + 1; j < col.length; j++) {

**double** tmp = mat[j][i] / mat[i][i];

col[j] -= tmp \* col[i];

**for** (**int** k = i; k < col.length; k++) {

mat[j][k] -= tmp \* mat[i][k];

}

}

}

// reversal

**double**[] x = **new** **double**[col.length];

**for** (**int** i = col.length - 1; i >= 0; i--) {

**double** sum = 0;

**for** (**int** j = i + 1; j < col.length; j++) {

sum += mat[i][j] \* x[j];

}

x[i] = (col[i] - sum) / mat[i][i];

}

**return** x;

}

}

**Результаты**

