**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Л А Б О Р А Т О Р Н А Р О Б О Т А № 5**

***Розв’язання систем лінійних алгебраїчних рівнянь***

Виконав:

Студент ФІОТ

Групи ІО – 31

Долинний Олександр

**Київ**

**2015**

**Лабораторна робота №5**

**1. Тема:** «Розв’язання систем лінійних алгебраїчних рівнянь»

**Мета:** Вивчити алгоритми методів розв'язання систем лінійних алгебраїчних рівнянь на ЕОМ.

**Завдання:** Відповідно до варіанту завдання розробити блок-схеми обчислення виразів для лінійного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу на алгоритмічній мові Pascal..

**2. Завдання:**

1. Ознайомитись з відомими точними чисельними методами розв'язання систем лінійних алгебраїчних рівнянь (СЛАР).

2. Скласти схему алгоритму розв’язання систем лінійних алгебраїчних рівнянь методом, вказаним викладачем.

3. Скласти програму розв'язання систем лінійних алгебраїчних рівнянь наалгоритмічній мові Pascal.

4. Користуючись даними відповідного варіанту (табл.1), розв’язати СЛАР на ЕОМ (початкові дані та результати роздрукувати).

5. Оформити звіт.

Звіт по лабораторній роботі повинен містити:

1) файл вихідного тексту програми;

2) файли результатів для тестового прикладу і для інтерполяції заданої функції;

3) опис алгоритму розрахунку (в текстовій формі та у вигляді блок-схеми) в

електронному та роздрукованому вигляді;

4) роздруківку файлів з коментарями;

5) загальні висновки за результатами роботи, що включають результати

тестування, отримані оцінки похибки результатів і обґрунтування цих оцінок.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер  варіанту | Матриця коефіцієнтів  системи | Стовпець  вільних членів | Примітка |
| 11 | 8,7 -3,1 1,8 2,2  2,1 6,7 -2,2 0  3,2 -1,8 -9,5 -1,9  1,2 2,8 -1,4 -9,9 | -9,7  13,1  6,9  25,1 | *x1=0.117*  *x2=1.714*  *x3=-0.622*  *x4=-1.948* |
|  |  |  |  |

**3. Лістинг програми:**

**Main**

package lab5;

public class Main {

public static void main(String[] args) {

Viewer viewer = new Viewer();

}

}

**Solver**

package lab5;

public class Solver {

private int dimension;

private double[][] factorArray;

private double[] absoluteArray;

private double[] solutionArray;

private boolean unicityOfSystem;

private boolean solvabilityOfSystem;

public Solver(int dimension, double[][] factorArray, double[] absoluteArray) {

setDimension(dimension);

setFactorArray(factorArray);

setAbsoluteArray(absoluteArray);

unicityOfSystem = true;

solvabilityOfSystem = true;

solutionArray = new double[dimension];

}

public void solve() {

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

//Перевірка рядку на 0

boolean flag = false;

for (int j = i; j < dimension; j++) {

if (factorArray[i][j] != 0) {

flag = true;

}

}

if (flag == false) {

if (absoluteArray[i] == 0) {

unicityOfSystem = false;

} else {

solvabilityOfSystem = false;

}

break;

}

//Пересув рядків

double temp = factorArray[i][i];

int tempN = i;

for (int j = i + 1; j < dimension; j++) {

if (Math.abs(factorArray[j][i]) > Math.abs(temp)) {

temp = factorArray[j][i];

tempN = j;

}

}

if (tempN != i) {

for (int j = i; j < dimension; j++) {

double buf = factorArray[i][j];

factorArray[i][j] = factorArray[tempN][j];

factorArray[tempN][j] = buf;

}

double buf = absoluteArray[i];

absoluteArray[i] = absoluteArray[tempN];

absoluteArray[tempN] = buf;

}

//Коефіцієнти і-го рядка

for (int j = i + 1; j < dimension; j++) {

factorArray[i][j] = factorArray[i][j] / factorArray[i][i];

}

absoluteArray[i] = absoluteArray[i] / factorArray[i][i];

factorArray[i][i] = 1;

//Віднімання коефіцієнтів

for (int j = 0; j < dimension; j++) {

if (j != i) {

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

factorArray[j][k] = factorArray[j][k] - factorArray[i][k] \* factorArray[j][i];

}

absoluteArray[j] = absoluteArray[j] - absoluteArray[i] \* factorArray[j][i];

}

}

}

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

solutionArray[i] = absoluteArray[i];

}

}

public int getDimension() {

return dimension;

}

public void setDimension(int dimension) {

this.dimension = dimension;

}

public double[][] getFactorArray() {

return factorArray;

}

public void setFactorArray(double[][] factorArray) {

this.factorArray = factorArray;

}

public double[] getAbsoluteArray() {

return absoluteArray;

}

public void setAbsoluteArray(double[] absoluteArray) {

this.absoluteArray = absoluteArray;

}

public double[] getSolutionArray() {

return solutionArray;

}

public boolean isUnicityOfSystem() {

return unicityOfSystem;

}

public boolean isSolvabilityOfSystem() {

return solvabilityOfSystem;

}

}

**Viewer**

package lab5;

import javax.swing.\*;

import javax.swing.table.TableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Viewer {

public static int FRAME\_WIDTH = 600;

public static int FRAME\_HEIGHT = 300;

double[][] factoryArray = new double[4][4];

double[] absoluteArray = new double[4];

JPanel panel = new JPanel();

GridLayout g = new GridLayout(3, 1);

JPanel panela = new JPanel();

GridLayout ga = new GridLayout(4, 1);

JPanel panela0 = new JPanel();

JPanel panela1 = new JPanel();

JPanel panela2 = new JPanel();

JPanel panela3 = new JPanel();

JPanel panelb = new JPanel();

JPanel panelc = new JPanel();

JButton task = new JButton("Обчислити");

JButton taskA = new JButton("Значення за замовченням");

JButton taskC = new JButton("Стерти значення");

JLabel labelA = new JLabel("Безліч коренів");

JLabel labelB = new JLabel("Немає коренів");

JTable table = new JTable(4, 1);

JTextField fielda00 = new JTextField(5);

JLabel labela00 = new JLabel(" x1 + ");

JTextField fielda01 = new JTextField(5);

JLabel labela01 = new JLabel(" x2 + ");

JTextField fielda02 = new JTextField(5);

JLabel labela02 = new JLabel(" x3 + ");

JTextField fielda03 = new JTextField(5);

JLabel labela03 = new JLabel(" x4 = ");

JTextField fielda0 = new JTextField(5);

JTextField fielda10 = new JTextField(5);

JLabel labela10 = new JLabel(" x1 + ");

JTextField fielda11 = new JTextField(5);

JLabel labela11 = new JLabel(" x2 + ");

JTextField fielda12 = new JTextField(5);

JLabel labela12 = new JLabel(" x3 + ");

JTextField fielda13 = new JTextField(5);

JLabel labela13 = new JLabel(" x4 = ");

JTextField fielda1 = new JTextField(5);

JTextField fielda20 = new JTextField(5);

JLabel labela20 = new JLabel(" x1 + ");

JTextField fielda21 = new JTextField(5);

JLabel labela21 = new JLabel(" x2 + ");

JTextField fielda22 = new JTextField(5);

JLabel labela22 = new JLabel(" x3 + ");

JTextField fielda23 = new JTextField(5);

JLabel labela23 = new JLabel(" x4 = ");

JTextField fielda2 = new JTextField(5);

JTextField fielda30 = new JTextField(5);

JLabel labela30 = new JLabel(" x1 + ");

JTextField fielda31 = new JTextField(5);

JLabel labela31 = new JLabel(" x2 + ");

JTextField fielda32 = new JTextField(5);

JLabel labela32 = new JLabel(" x3 + ");

JTextField fielda33 = new JTextField(5);

JLabel labela33 = new JLabel(" x4 = ");

JTextField fielda3 = new JTextField(5);

public Viewer() {

JFrame frame = new JFrame();

frame.setTitle("П'ята лабораторна робота");

frame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);

labelA.setVisible(false);

labelB.setVisible(false);

table.setVisible(false);

panel.setBackground(Color.white);

panel.setLayout(g);

panela.setLayout(ga);

panela0.add(fielda00);

panela0.add(labela00);

panela0.add(fielda01);

panela0.add(labela01);

panela0.add(fielda02);

panela0.add(labela02);

panela0.add(fielda03);

panela0.add(labela03);

panela0.add(fielda0);

panela1.add(fielda10);

panela1.add(labela10);

panela1.add(fielda11);

panela1.add(labela11);

panela1.add(fielda12);

panela1.add(labela12);

panela1.add(fielda13);

panela1.add(labela13);

panela1.add(fielda1);

panela2.add(fielda20);

panela2.add(labela20);

panela2.add(fielda21);

panela2.add(labela21);

panela2.add(fielda22);

panela2.add(labela22);

panela2.add(fielda23);

panela2.add(labela23);

panela2.add(fielda2);

panela3.add(fielda30);

panela3.add(labela30);

panela3.add(fielda31);

panela3.add(labela31);

panela3.add(fielda32);

panela3.add(labela32);

panela3.add(fielda33);

panela3.add(labela33);

panela3.add(fielda3);

panela.add(panela0);

panela.add(panela1);

panela.add(panela2);

panela.add(panela3);

panelc.add(table);

panelb.add(task);

panelb.add(taskA);

panelb.add(taskC);

panelc.add(labelA);

panelc.add(labelB);

panel.add(panela);

panel.add(panelb);

panel.add(panelc);

task.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

labelA.setVisible(false);

labelB.setVisible(false);

table.setVisible(false);

factoryArray[0][0] = Double.parseDouble(fielda00.getText());

factoryArray[0][1] = Double.parseDouble(fielda01.getText());

factoryArray[0][2] = Double.parseDouble(fielda02.getText());

factoryArray[0][3] = Double.parseDouble(fielda03.getText());

factoryArray[1][0] = Double.parseDouble(fielda10.getText());

factoryArray[1][1] = Double.parseDouble(fielda11.getText());

factoryArray[1][2] = Double.parseDouble(fielda12.getText());

factoryArray[1][3] = Double.parseDouble(fielda13.getText());

factoryArray[2][0] = Double.parseDouble(fielda20.getText());

factoryArray[2][1] = Double.parseDouble(fielda21.getText());

factoryArray[2][2] = Double.parseDouble(fielda22.getText());

factoryArray[2][3] = Double.parseDouble(fielda23.getText());

factoryArray[3][0] = Double.parseDouble(fielda30.getText());

factoryArray[3][1] = Double.parseDouble(fielda31.getText());

factoryArray[3][2] = Double.parseDouble(fielda32.getText());

factoryArray[3][3] = Double.parseDouble(fielda33.getText());

absoluteArray[0] = Double.parseDouble(fielda0.getText());

absoluteArray[1] = Double.parseDouble(fielda1.getText());

absoluteArray[2] = Double.parseDouble(fielda2.getText());

absoluteArray[3] = Double.parseDouble(fielda3.getText());

Solver solver = new Solver(4, factoryArray, absoluteArray);

solver.solve();

if (solver.isSolvabilityOfSystem() == false) {

labelB.setVisible(true);

} else {

if (solver.isUnicityOfSystem() == false) {

labelA.setVisible(true);

} else {

table.setVisible(true);

double[] solution = solver.getSolutionArray();

TableModel model = table.getModel();

for (int i = 0; i < 4; i++)

model.setValueAt(solution[i], i, 0);

}

}

}

});

taskA.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

fielda00.setText("8.7");

fielda01.setText("-3.1");

fielda02.setText("1.8");

fielda03.setText("2.2");

fielda0.setText("-9.7");

fielda10.setText("2.1");

fielda11.setText("6.7");

fielda12.setText("-2.2");

fielda13.setText("0");

fielda1.setText("13.1");

fielda20.setText("3.2");

fielda21.setText("-1.8");

fielda22.setText("-9.5");

fielda23.setText("-1.9");

fielda2.setText("6.9");

fielda30.setText("1.2");

fielda31.setText("2.8");

fielda32.setText("-1.4");

fielda33.setText("-9.9");

fielda3.setText("25.1");

}

});

taskC.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

fielda00.setText("");

fielda01.setText("");

fielda02.setText("");

fielda03.setText("");

fielda0.setText("");

fielda10.setText("");

fielda11.setText("");

fielda12.setText("");

fielda13.setText("");

fielda1.setText("");

fielda20.setText("");

fielda21.setText("");

fielda22.setText("");

fielda23.setText("");

fielda2.setText("");

fielda30.setText("");

fielda31.setText("");

fielda32.setText("");

fielda33.setText("");

fielda3.setText("");

}

});

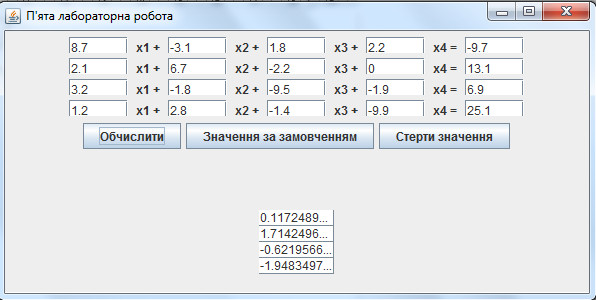
frame.add(panel);

frame.setVisible(true);

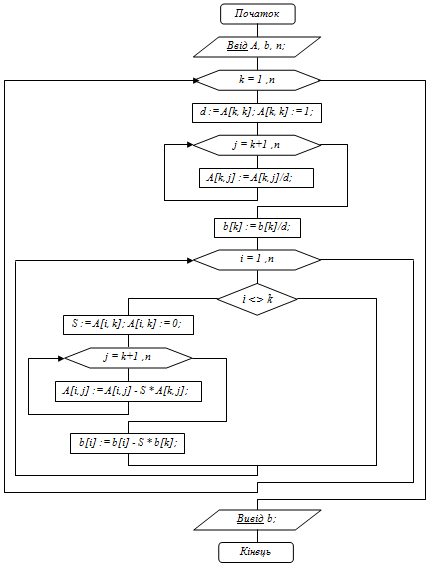
}

}

**4. Приклад**



**5. Опис алгоритму**

****Метод Жордана-Гаусса являється однією з модифікацій методу Гаусса і знаходження розв'язку системи лінійних алгебраїчних рівнянь з допомогою даного методу зводиться до перетворення вихідної системи до системи з одиничною або діагональною матрицею. Тобто основна відмінність між методом Гаусса і методом Жордана-Гаусса полягає в тому, що при реалізації останнього, елементи матриці обнулюються як під, так і над головною діагоналлю, а значення діагональних елементів стають рівними одиниці. В результаті даного перетворення елементи вектора вільних членів являтимуться шуканим розв'язком системи.

Розглянемо даний метод більш детально. Для цього запишемо систему лінійних рівнянь наступного вигляду:

метод Жордана-Гаусса

Обчислювальна схема методу Жордана-Гаусса складається з Метод Жордана-Гаусса циклів, в кожному з яких послідовно з допомогою Метод Жордана-Гаусса-го рядка виключаються елементи при невідомій Метод Жордана-Гаусса в кожному рядку матриці коефіцієнтів, крім Метод Жордана-Гаусса-го. Дана схема реалізується з допомогою наступних кроків:

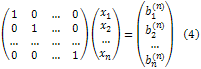
1. Перевіряємо, щоб наш рядок не містив лише нульових елементів.
2. Для досягнення найбльшої точності результату елемент Метод Жордана-Гаусса повинен бути більшим за всі наступні елементи у k-му стовпчику, інакше робимо пересув стовпчиків.
3. Виконуємо ділення Метод Жордана-Гаусса-го рядка матриці та стовпця вільних членів на елемент Метод Жордана-Гаусса. Даний крок описується з допомогою наступних формул:

Метод Жордана-Гаусса

1. На другому кроці, від Метод Жордана-Гаусса-го рядка матриці та Метод Жордана-Гаусса-го рядка стовпця вільних членів віднімаємо Метод Жордана-Гаусса-й помножений на елемент Метод Жордана-Гаусса:

Метод Жордана-Гаусса

В результаті виконання Метод Жордана-Гаусса циклів отримаємо систему з одиничною матрицею коефіцієнтів:



З допомогою даної системи вектор невідомих легко знаходиться за наступною формулою:Метод Жордана-Гаусса.

Зауваження: необхідною і достатньою умовою при використанні методу Жордана-Гаусса являється відмінність від нуля всіх ведучих елементів матриці, тобто Метод Жордана-Гаусса.

1. **Висновки**

**У ході лабораторної роботи було використано метод Жордана-Гауса для обчислення систем лінійних алгебраїчних рівнянь з чотирма невідомими та знайдена помилка у методичних вказівках до лабораторної роботи.**