Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №3

**На тему:**

«Розв’язування систем лінійних алгебраїчних рівнянь

методом Крамера та методом оберненої матриці»

з дисципліни «Чисельні методи»

**Лекторка:**

доцент каф. ПЗ

Мельник Н. Б.

**Виконав:**

ст. гр. ПЗ-11

Козира О.Ф.

**Прийняла:**

доцент каф. ПЗ

Мельник Н. Б.

« \_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

Львів – 2022

**Тема:** Розв’язування систем лінійних алгебраїчних рівнянь методом Крамера та методом оберненої матриці.

**Мета:** Ознайомлення на практиці з методом Крамера та методом оберненої матриці розв’язування систем лінійних алгебраїчних рівнянь.

**Теоретичні відомості**

**Метод Крамера** – використовується для розв’язання СЛАР яка містить n рівнянь та n невідомих, причому визначник матриці коефіцієнтів А не дорівнює нулю. Для знаходження коренів застосовують формулу , де це матриця A, в якій *i*-тий стовпець замінений стовпцем вільних членів (матриці B).

**Метод оберненої матриці** – полягає в отриманні рівняння з рівняння шляхом домноження його на . Для знаходження оберненої матриці потрібно транспонувати матрицю алгебраїчних доповнень та поділити її на визначник матриці А. Для знаходження алгебраїчних доповнень скористаємось формулою , де – це мінор, який отримують з матриці А викреслюванням *i*-го рядка та *j*-го стовпця.

**Індивідуальне завдання**

*Варіант 11*

Скласти програму розв’язування системи лінійних алгебраїчних рівнянь методом оберненої матриці та методом Крамера:

**Код функцій**

void kramer(double A[3][3], double B[3]) {

double Acopy[3][3], det = determinant(A);

cout << "\n Kramer method:\n\n";

if (abs(det) <= 0.000001) {

cout << " Cannot calculate, because determinant A is zero.\n";

return;

}

cout << " Determinant A: " << det << "\n\n";

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

for (int k = 0; k < 3; k++) {

Acopy[j][k] = A[j][k];

}

}

for (int j = 0; j < 3; j++) Acopy[j][i] = B[j];

cout << " " << i + 1 << "-th root: " << determinant(Acopy) / det << endl;

}

}

void matrix(double A[3][3], double B[3]) {

double AT[3][3], det = determinant(A);

cout << "\n Matrix method:\n\n";

if (abs(det) <= 0.000001) {

cout << " Cannot calculate, because determinant A is zero.\n";

return;

}

cout << " Determinant A: " << det << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

double M[2][2];

for (int o = 0, p = 0; o < 3; o++) {

if (o == i) continue;

for (int k = 0, l = 0; k < 3; k++) {

if (k == j) continue;

M[p][l] = A[o][k];

l++;

}

p++;

}

double m = M[0][0] \* M[1][1] - M[0][1] \* M[1][0];

AT[j][i] = pow(-1., i + j) \* m / det;

cout << "\n " << i << " " << j << " minor:\n " << M[0][0] << " " << M[0][1] << "\n " << M[1][0] << " " << M[1][1] << endl;

cout << " determinant of minor: " << m << "\n AT " << j << " " << i << " = " << AT[j][i] << endl;

}

}

cout << endl;

for (int i = 0; i < 3; i++) {

double r = AT[i][0] \* B[0] + AT[i][1] \* B[1] + AT[i][2] \* B[2];

cout << " " << i + 1 << "-th root: " << r << endl;

}

}

**Протокол роботи**

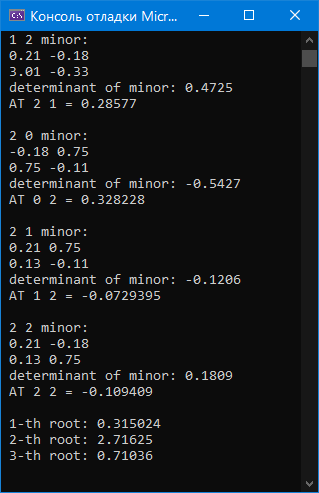
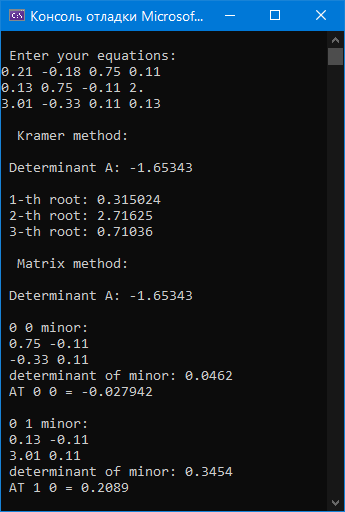


Рис.1. Робота програми

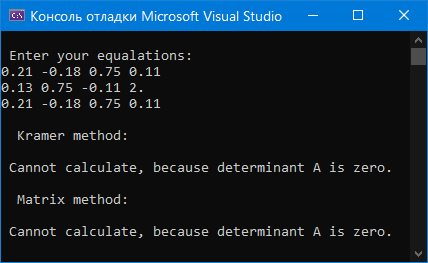


Рис.2. Введення некоректних даних

**Висновки**

Виконуючи лабораторну роботу №3, я навчився розв’язувати СЛАР методами Крамера та оберненої матриці, а також склав програму, яка їх розв’язує автоматично.