Міністерство освіти і науки, молоді та спорту України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**про виконання лабораторної роботи № 3**

«**«***Прямі методи розв’язування СЛАР. Матричний метод. Метод Крамера*»

**з дисципліни «Чисельні методи»**

**Лектор:**

доцент кафедри ПЗ

Мельник Н.Б.

**Виконала:**

студ. групи ПЗ-12

Ліхтарчик С.Й.

**Прийняла:**

доцент кафедри ПЗ

Мельник Н.Б.

«\_\_\_» \_\_\_\_\_\_ 2020 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2020р.

**Тема роботи:** Прямі методи розв’язування СЛАР. Матричний метод. Метод Крамера.

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

**Теоретичні відомості**

Прямі методи дозволяють розв’язати систему рівнянь за скінчене число арифметичних операцій. Якщо всі операції виконуються точно (без помилок заокруглення), то розв’язок заданої системи також отримуємо точним.

*Системою лінійних алгебраїчних рівнянь* (СЛАР) називають систему вигляду

|  |  |
| --- | --- |
| , | (1) |

де  – невідомі,  – вільні члени системи,  – коефіцієнти системи.

У матричному вигляді система прийме вигляд:

|  |  |
| --- | --- |
|  | (2) |
| де |  |
| , , . | (3) |

*Розв’язком системи* (2) називається *n*-компонентний вектор-стовбець ,

який перетворює матричне рівняння (2) у вірну числову тотожність.

Якщо матриця A невироджена (неособлива), тобто її визначник відмінний від нуля , то система (1) має єдиний розв’язок.

Розв’язком системи лінійних алгебраїчних рівнянь є будь-яка сукупність [дійсних чисел](http://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B9%D1%81%D0%BD%D1%96_%D1%87%D0%B8%D1%81%D0%BB%D0%B0) , яка при підстановці в кожне рівняння системи перетворює його в [тотожність](http://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D1%82%D0%BE%D0%B6%D0%BD%D1%96%D1%81%D1%82%D1%8C).

Якщо система має хоча б один розв’язок, то вона називається сумісною, і несумісною, якщо не має жодного. Сумісна система називається визначеною, якщо вона має єдиний розв’язок, і невизначеною, якщо вона має безліч розв’язків.

Якщо всі вільні члени , система лінійних алгебраїчних рівнянь називається однорідною. Однорідна система має очевидний розв'язок, у якому всі . Цей розв'язок прийнято називати тривіальним. Відмінні від тривіального розв'язки існують тільки тоді, коли матриця A [вироджена](http://uk.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D0%BE%D0%B4%D0%B6%D0%B5%D0%BD%D0%B0_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D1%8F).

**Метод оберненої матриці**

У лінійній алгебрі звичайно використовують спосіб розв’язання СЛАР, що ґрунтується на обчисленні оберненої матриці A-1. Дійсно, якщо помножити обидві частини рівняння (3) на матрицю A-1, то розв’язок отримаємо у вигляді

,

.

Елементи оберненої матриці  можна обчислити за формулою

,

де  - алгебраїчне доповнення елемента  матриці A і  - визначник цієї матриці. Тоді для знаходження всіх її елементів потрібно знайти значення  визначників порядку . Ця задача настільки трудомістка, що її дуже важко розв’язати для =10.

Метод Крамера

Менш трудомістким є метод Крамера, у якому

**,**

де матриця  формується з матриці *A* заміною її -го стовпця на стовпець *B* вільних членів

.

Для розв’язання СЛАР з  невідомими треба обчислити +1 визначників порядку . Це також дуже трудомістка операція (для СЛАР порядку  треба виконати  операцій).

**Індивідуальне завдання**

Написати програму розв’язку системи лінійних алгебраїчних рівнянь у відповідності до варіанту:

1. матричним методом;
2. методом Крамера.

**Хід роботи**

Код програми:

Header.h

#pragma once

#include <iostream>

#include <iomanip>

using namespace std;

void printArray(double array[3][3]);

void printFullArray(double array[3][3], double res[3]);

double findDeterminant(double array[3][3]);

double findMinor(double array[3][3], int i, int j);

void copyArray(double array1[3][3], double array2[3][3]);

void copyArrayAndSwapColumn(double array1[3][3], double array2[3][3], double res[3], int z);

void cramerMethod(double array[3][3], double res[3], bool printSteps);

void inverseMatrix(double array[3][3]);

void transpose(double array[3][3]);

void multiplyArray(double array[3][3], double multVal);

void matrixMethod(double array[3][3], double res[3], bool printSteps);

Source.cpp

#include "header.h"

void printArray(double array[3][3])

{

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

{

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

{

cout << setw(10) << showpos << array[i][j] << " ";

}

cout << endl;

}

cout << endl;

}

void printFullArray(double array[3][3], double res[3])

{

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

{

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

{

cout << showpos << array[i][j] << " ";

}

cout << "| " << res[i] << endl;

}

cout << endl;

}

double findDeterminant(double arr[3][3])

{

double det = 0;

det += arr[0][0] \* arr[1][1] \* arr[2][2];

det += arr[0][1] \* arr[1][2] \* arr[2][0];

det += arr[1][0] \* arr[2][1] \* arr[0][2];

det -= arr[0][2] \* arr[1][1] \* arr[2][0];

det -= arr[0][0] \* arr[1][2] \* arr[2][1];

det -= arr[0][1] \* arr[1][0] \* arr[2][2];

return det;

}

double findMinor(double array[3][3], int i0, int j0)

{

double min[4];

double minor;

int z = 0;

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

{

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

{

if (i != i0 && j != j0)

{

min[z] = array[i][j];

z++;

}

}

}

minor = min[0] \* min[3] - min[1] \* min[2];

return minor;

}

void copyArray(double array1[3][3], double array2[3][3])

{

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

{

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

{

array2[i][j] = array1[i][j];

}

}

}

void copyArrayAndSwapColumn(double array1[3][3], double array2[3][3], double res[3], int z)

{

copyArray(array1, array2);

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

{

array2[j][z] = res[j];

}

}

void cramerMethod(double array[3][3], double res[3], bool printSteps)

{

if (printSteps) cout << "\nCramer method was launched... " << endl;

double roots[3];

double array2[3][3];

for (int z = 0; z < 3; ++z)

{

copyArrayAndSwapColumn(array, array2, res, z);

if (printSteps) cout << "Determinant " << noshowpos << z + 1 << " = " << findDeterminant(array2) << endl;

roots[z] = findDeterminant(array2) / findDeterminant(array);

}

cout << "\nCramer method gave a result: " << endl;

cout << "x1 = " << roots[0] << "; x2 = " << roots[1] << "; x3 = " << roots[2] << endl;

}

void inverseMatrix(double array[3][3])

{

double array2[3][3];

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

{

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

{

array2[i][j] = findMinor(array, i, j);

if ((i + j) % 2)

{

array2[i][j] = -array2[i][j];

}

}

}

copyArray(array2, array);

}

void transpose(double array[3][3])

{

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

{

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

{

if (i < j)

{

double temp = array[i][j];

array[i][j] = array[j][i];

array[j][i] = temp;

}

}

}

}

void multiplyArray(double array[3][3], double multVal)

{

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

{

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

{

array[i][j] \*= multVal;

}

}

}

void matrixMethod(double array[3][3], double res[3], bool printSteps)

{

if (printSteps) cout << "\nMatrix method was launched... " << endl;

double multValue = 1 / findDeterminant(array);

inverseMatrix(array);

transpose(array);

multiplyArray(array, multValue);

cout << "Inversed matrix: " << endl;

printArray(array);

double roots[3]{ 0 };

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

{

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

{

roots[i] += array[i][j] \* res[j];

}

}

cout << "Matrix method gave a result: " << endl;

cout << noshowpos << "x1 = " << roots[0] << "; x2 = " << roots[1] << "; x3 = " << roots[2] << endl;

}

NM\_lab\_3.cpp

#include "header.h"

int main()

{

cout << "Do you want to enter new coefs or use default one?" << endl;

cout << "1 - Yes" << endl;

cout << "0 - No" << endl;

bool newValueEntered;

while (cout << "So, your choice: " && !(cin >> newValueEntered))

{

std::cin.clear(); //clear bad input flag

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n'); //discard input

std::cout << "Invalid input; please re-enter.\n";

}

double indecesArray[3][3] = { {0.46, 1.72, 2.53},

{1.53, -2.23, -1.83},

{0.75, 0.86, 3.72 } };

double resultArray[3] = { 2.44, 2.83, 1.06 };

if (newValueEntered)

{

cout << "Enter new coefficients: " << endl;

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

{

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

{

cin >> indecesArray[i][j];

}

}

cout << "Enter new rvalues: " << endl;

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

{

cin >> resultArray[i];

}

}

cout << "\nDo you want to see the steps?" << endl;

cout << "1 - Yes" << endl;

cout << "0 - No" << endl;

while (cout << "So, your choice: " && !(cin >> newValueEntered))

{

std::cin.clear(); //clear bad input flag

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n'); //discard input

std::cout << "Invalid input; please re-enter.\n";

}

cout << "\nHere is yout matrix: " << endl;

printFullArray(indecesArray, resultArray);

cout << "Determinant: " << findDeterminant(indecesArray) << endl;

cramerMethod(indecesArray, resultArray, newValueEntered);

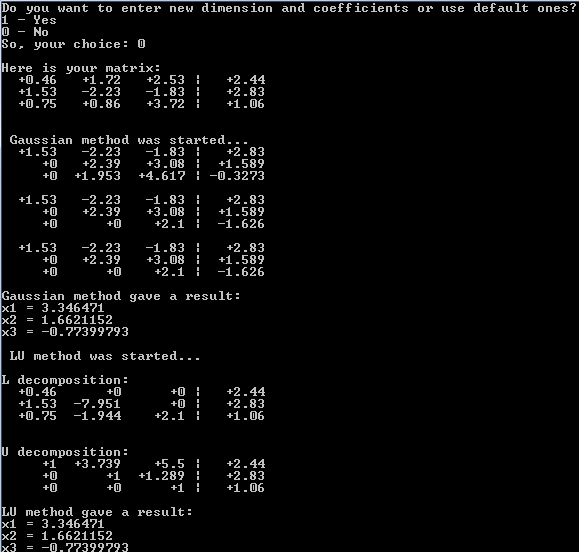
matrixMethod(indecesArray, resultArray, newValueEntered);

system("pause");

return 0;

}

Результат програми:



**Висновки**

Під час виконання лабораторної роботи я ознайомився з двома методами розв’язування систем лінійних аналітичних рівнянь, а саме з методом Крамера та матричним методом. Обидва методи реалізував програмно, використавши їх для розв’язування заданої системи рівнянь.