Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій

ЗВІТ

Про виконання лабораторної роботи № 3

«розв'язування систем лінійних алгебраїчних рівнянь методом Крамера та методом оберненої матриці»

з дисципліни «Чисельні методи»

Лектор:
доцент кафедри ПЗ
Мельник Н.Б.

Виконав:
студ. групи ПЗ-15
Марущак А. С.

Прийняв:
асистент кафедри ПЗ
Гарматій Г.Ю
«___» ______ 2022 р.
∑ = ____ ___________

Львів – 2022

Тема роботи: розв'язування систем лінійних алгебраїчних рівнянь методом Крамера та методом оберненої матриці.

Мета роботи: ознайомлення на практиці з методом Крамера та методом оберненої матриці розв'язування систем лінійних алгебраїчних рівнянь.

Теоретичні відомості

Математичні моделі багатьох процесів зводяться до розв'язування систем лінійних алгебраїчних рівнянь (СЛАР). Розв'язати деякі нелінійні задачі можна послідовним розвязуванням систем лінійних алгебраїчних рівнянь. Прямі (точні) методи дають змогу розв'язати систему лінійних алгебраїчних рівнянь за скінченну кількість арифметичних операцій. Якщо всі операції виконують точно (без похибок заокруглення), то розв'язок заданої системи також отримують точним. До прямих методів належать метод Крамера та метод оберненої матриці, які розглянемо нижче.

Систему лінійних алгебраїчних рівнянь записують у розгорнутій формі:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}$$

або у скороченому вигляді:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i\ , \qquad i = \overline{1,m},$$

або у матричній формі:

$$A \cdot X = B,$$

де

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & & & \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}, \quad \overline{X} = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix}, \quad \overline{B} = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_m \end{pmatrix},$$

A - прямокутна матриця коефіцієнтів системи розмірності m×n ; X – вектор-стовпець невідомих розмірності n; B - вектор-стовпець вільних членів розмірності m.

Розв'язком системи називають n-компонентний вектор-стовпець X, який перетворює співвідношення у правильну числову тотожність.

Метод Крамера

Розглянемо СЛАР, яка містить n рівнянь та n невідомих, причому визначник її не дорівнює нулеві. Для знаходження невідомих x_i застосовують формулу Крамера:

$$x_i = \frac{\det A_i}{\det A}, \quad i = \overline{1, n},$$

де det A - визначник матриці A, det A_i - визначник матриці A_i , яку отримують з матриці A шляхом заміни її i -го стовпця стовпцем вільних членів:

$$A_i = \begin{pmatrix} a_{11} & \dots & a_{1\,i-1} & b_1 & a_{1\,i+1} & \dots & a_{1n} \\ a_{21} & \dots & a_{2\,i-1} & b_2 & a_{2\,i+1} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{n\,i-1} & b_n & a_{n\,i+1} & \dots & a_{nn} \end{pmatrix}, \quad i = \overline{1, n}.$$

Алгоритм методу Крамера:

1. Отримати матрицю A коефіцієнтів системи та вектор-стовпець B вільних членів.
2. Знайти визначник матриці $\det A$.
3. Знайти визначник $\det A_i$, $i = \overline{1, n}$, де A_i – матриця, у якої і-й стовпець замінений на вектор B.
4. Визначити $x_i = \frac{\det A_i}{\det A}, i = \overline{1, n}$.

Метод оберненої матриці (матричний метод)

У лінійній алгебрі часто використовують матричний метод розв’язування систем лінійних алгебраїчних рівнянь. Цей метод ґрунтується на обчисленні оберненої матриці A^{-1} , яка існує лише при умові, коли визначник матриці A відмінний від нуля: det A $\neq$ 0.

Якщо обидві частини матричного рівняння $A \cdot X = B$ зліва домножити на матрицю A^{-1} , то отримаємо співвідношення:

$$A^{-1}AX = A^{-1}B.$$

Враховуючи, що добуток оберненої матриці на саму матрицю дає одиничну матрицю, а результатом добутку одиничної матриці E на матрицю-стовпець X є матриця-стовпець X, одержимо матричний розв'язок системи лінійних алгебраїчних рівнянь у вигляді:

$$X = A^{-1}B.$$

Для знаходження оберненої матриці A^{-1} необхідно застосувати алгоритм, який складається з таких пунктів:

1. Обчислення визначника det A матриці A коефіцієнтів системи . Якщо він не дорівнює нулеві, то продовжуємо розв'язувати СЛАР. Якщо det A = 0, то матриця A є виродженою і для неї не існує оберненої.

2. Знаходження алгебраїчних доповнень для елементів матриці A . Вони дорівнюють мінорам для відповідних елементів a_{ij} ($i, j = \overline{1, n}$), помноженим на$(-1)^{i+j}$, тобто

$$\overline{A_{ij}} = (-1)^{i+j} \cdot M_{ij},$$

а мінор M_{ij} отримують із матриці A шляхом викреслювання і -го рядка та j -го стовпця.

3. Формування матриці $\overline{A}$, елементами якої є алгебраїчні доповнення матриці A.

4. Транспонування матриці $\overline{A}$ і знаходження приєднаної матриці $\widetilde{A} = \overline{A}^T$.

5. Визначення оберненої матриці за формулою

$$A^{-1} = \frac{\widetilde{A}}{\det A}.$$

Алгоритм матричного методу:

1. Отримати матрицю A коефіцієнтів системи та вектор-стовпець B вільних членів.
2. Знайти обереную матрицю A^{-1} за вищенаведеним алгоритмом.
3. Вектор відповідей визначити як $X = A^{-1} \cdot B$.

Індивідуальне завдання

1. Ознайомитися з теоретичним матеріалом.
2. Скласти програму розв'язування системи лінійних алгебраїчних рівнянь методом оберненої матриці та методом Крамера.

$$6.\begin{cases} 0,95x_1 + 0,72x_2 - 1,14x_3 = 2,15 \\ 0,63x_1 + 0,24x_2 + 0,38x_3 = 0,74 \\ 1,23x_1 - 1,08x_2 - 1,16x_3 = 0,97 \end{cases}$$

Хід роботи

Для подальшої роботи я буду використовувати створену мною структуру даних SquareMatrix, що відображає квадратну матрицю, тобто таку, у якої к-ть рядків дорівнює к-ті стовпців. Ця структура даних передбачає можливості знаходження мінору, алгебраїчних доповнень, детермінанту, транспонованої та оберненої матриці, а також добутку матриці на число або вектор-стовпець. Покажемо деякі з цих функцій у вигляді коду:

Функція знаходження детермінанту:

```
double det(const SquareMatrix& matrix)
{
	if (matrix.size == 1) return matrix.data[0][0];

	double res = 0;
	for (int i = 0; i < matrix.size; i++)
	{
		res += matrix.data[0][i] * matrix.algebraicAddition(0, i);
	}
	return res;
}
```

Функція знаходження алгеб. доповнення:

```
double SquareMatrix::algebraicAddition(const int row, const int column) const
{
	if ((row + column) % 2 == 0) return this->minor(row, column);
	else return -this->minor(row, column);
}
```

Функція знаходження мінору:

```
double SquareMatrix::minor(const int row, const int column) const
{
	return det(this->withoutRowAndColumn(row, column));
}
```

Функція добутку матриці на вектор-стовпець:

```
std::vector<double> operator*(const SquareMatrix& matrix, const std::vector<double>& vector)
{
	if (matrix.size != vector.size()) throw std::string("Not equal sizes!");

	std::vector<double> result(matrix.size, 0);

	for (int i = 0; i < matrix.size; i++)
	{
		for (int j = 0; j < matrix.size; j++)
		{
			result[i] += matrix.data[i][j] * vector[j];
		}
	}
	return result;
}
```

Функція знаходження оберненної матриці:

```
SquareMatrix SquareMatrix::inversed() const
{
	double d = det(*this);
```

```
    if (d == 0) throw std::string("Inversed matrix doesn't exist");
    SquareMatrix result(this->size);
    for (int i = 0; i < result.size; i++)
    {
        for (int j = 0; j < result.size; j++)
        {
            result.data[i][j] = this->algebraicAddition(i, j);
        }
    }
    return (1 / d) * result.T();
}
```

Матричний метод:

Код функції:

```
std::vector<double> matrixMethod(const SquareMatrix& A, const std::vector<double>& B)
{
    if (A.matrixSize() != B.size()) throw std::string("Sizes not equal!"); // Перевіряємо на можливість перемножити матрицю і вектор-стовпець
    if(det(A) == 0) throw std::string("Solution can't be found!"); // Перевіряємо на рівність нулю детермінант

    std::vector<double> X(B.size()); // Створюємо вектор для запису відповідей

    SquareMatrix A_inv = A.inversed(); // Знаходимо оберену матрицю.

    std::cout << "Inversed matrix:\n" << A_inv << std::endl;
    X = A_inv * B; // Знаходимо його, як добуток оберненої матриці на вектор-стовпець вільних членів
    return X; // Повертаємо відповідь
}
```

Метод Крамера:

Код функції:

```
std::vector<double> kramerMethod(const SquareMatrix& A, const std::vector<double>& B)
{
    if (A.matrixSize() != B.size()) throw std::string("Sizes not equal!"); // Перевіряємо на можливість перемножити матрицю і вектор-стовпець

    double d = det(A); // Знаходимо детермінант початкової матриці
    if (d == 0) throw std::string("Solution can't be found!"); // Перевіряємо на рівність нулю детермінант

    std::vector<double> X(B.size()); // Створюємо вектор для запису відповідей

    std::cout << "det(A) = " << d << std::endl;

    for (int i = 0; i < A.matrixSize(); i++)
    {
        double detAi = det(A.withReplacedColumn(i, B)); // Створюємо матрицю Ai з заміненим стовпцем
        X[i] = detAi / d; // Знаходимо i-й корінь як частку детермінанта матриці Ai з заміненим стовпцем на детермінант початкової матриці
        std::cout << "det(A" << i + 1 << ") = " << detAi << "; X" << i + 1 << " = " << detAi << "/" << d << " = " << X[i] << std::endl;
    }

    return X; // Повертаємо відповідь
}
```

<u>Результат</u> виконання програми:

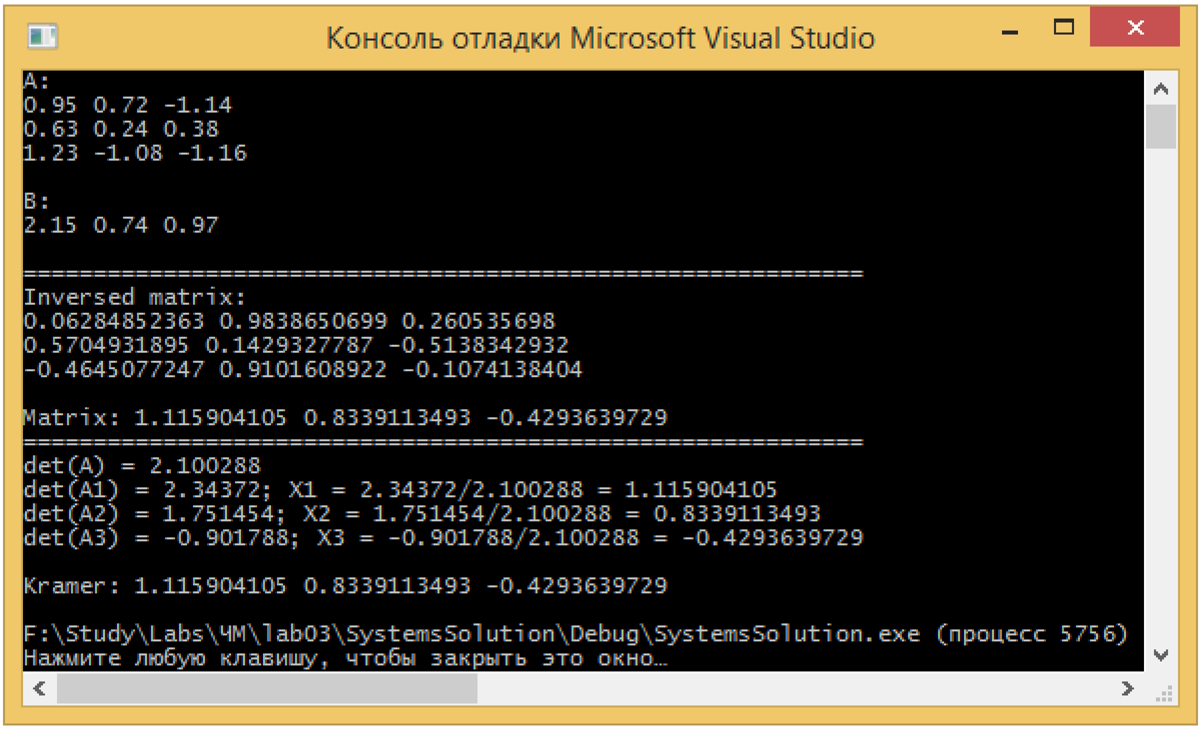

Рис 3.1. Результати виконання програми.

<u>Аналіз результатів</u>:

Як бачимо, метод Крамера і матричний метод дали однаковий результат, що свідчить, що він з високою вірогідністю є правильним. Його перевірка на калькуляторі підтверджує це.

<u>Висновок</u>:

Виконуючи цю лабараторну роботу ми ознайомились з 2ма методами розв'язування СЛАР: матричним та Крамера. За допомогою цих знань ми реалізували програму, що розв'язала наступну систему:

$$6.\begin{cases} 0{,}95x_1 + 0{,}72x_2 - 1{,}14x_3 = 2{,}15 \\ 0{,}63x_1 + 0{,}24x_2 + 0{,}38x_3 = 0{,}74 \\ 1{,}23x_1 - 1{,}08x_2 - 1{,}16x_3 = 0{,}97 \end{cases}$$

Ми отримали вектор дійсних чисел, , що відображає розв′язок цієї системи.

$$\mathrm{X} = (1.115904105, 0.8339113493, -0.4293639729)$$