

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет “Львівська політехніка”**  
**Інститут післядипломної освіти**



**ЗВІТ**

**Про виконання лабораторної роботи №2**  
**«Чисельні методи розв’язування системи лінійних**  
**рівнянь»**  
**з дисципліни «Чисельні методи»**

Виконав:  
слухач групи ПЗС-11  
Гринчук Тарас

Прийняла:  
ст. викл. Мельник Н. Б.

« » \_\_\_\_\_ 2014 р.

Σ \_\_\_\_\_

**Мета роботи:** Ознайомлення на практиці з прямими методами розв'язування систем лінійних алгебраїчних рівнянь.

## Система лінійних алгебраїчних рівнянь

Прямі (точні) методи дозволяють розв'язати систему рівнянь за скінчене число арифметичних операцій. Якщо всі операції виконуються точно (без помилок округлення), то розв'язок заданої системи також отримуємо точним. До прямих методів належать: метод послідовного виключення невідомих (метод Гауса та його модифікації: метод головного елемента, метод квадратного кореня, метод відображень та ін.), метод ортогоналізації, метод LU-розкладу.

[illegible]

У матричному вигляді система прийме вигляд:

$$\text{де } A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad \bar{B} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}, \quad \bar{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}. \quad (1.3)$$

$$\overline{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \text{ який перетворює матричне рівняння (2) у вірну числову тотожність.}$$

- сумісною, якщо вона має хоча б один ненульовий розв'язок;
- несумісною, якщо СЛАР не має жодного розв'язку;
- визначеною, якщо вона має тільки один розв'язок (випадок, коли  $m=n$ );
- невизначеною, якщо система має безліч розв'язків ( $m \neq n$ );
- виродженою, якщо головний визначник системи дорівнює нулю;
- неvirодженою, якщо головний визначник системи не дорівнює нулю.

Систему лінійних алгебраїчних рівнянь можна розв'язати на ЕОМ числовими методами, якщо вона сумісна, визначена, не вироджена.

Блок-схеми алгоритму Гауса з вибором головного елемента (рис. 1.1) та методу LU-розкладу (рис. 1.2):

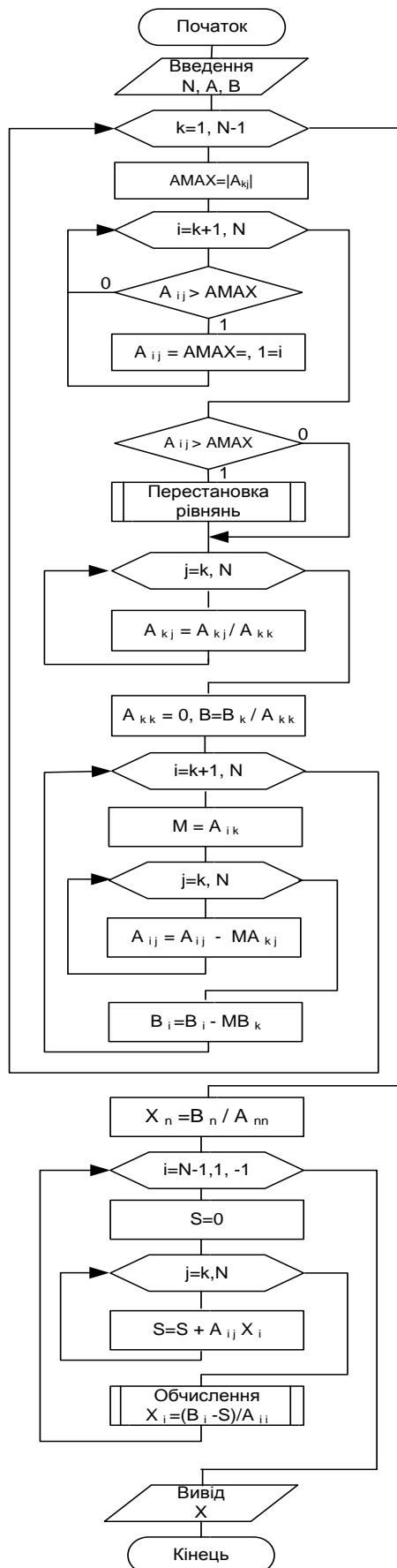


Рис. 1.1. Схема алгоритму розв'язання СЛАР методом Гауса з вибором головного елемента

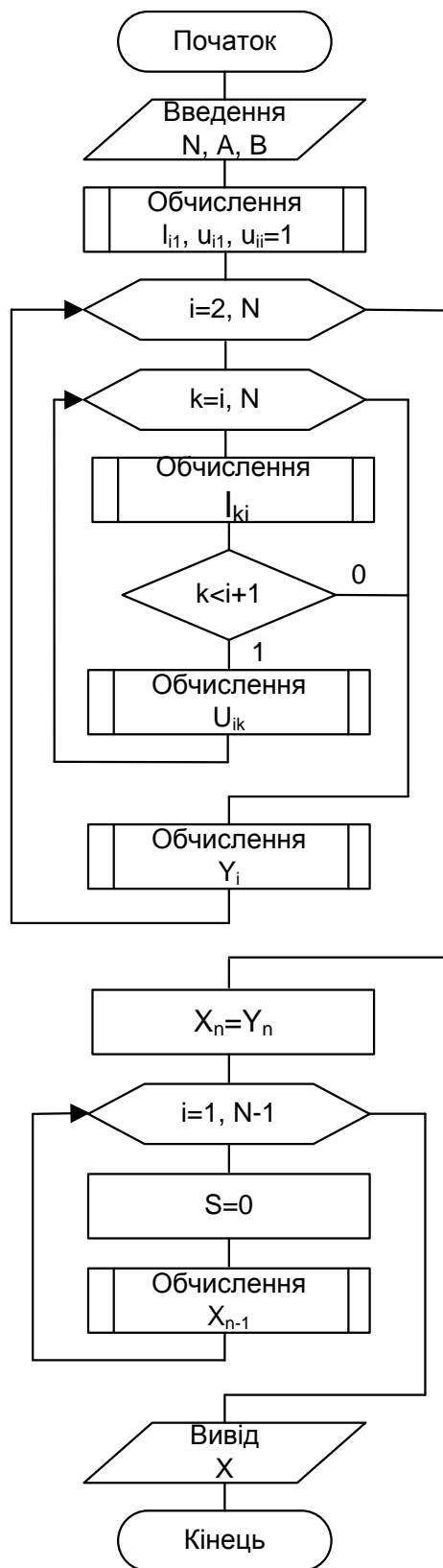


Рис. 1.2. Схема алгоритму розв'язання СЛАР методом LU-розкладу

## 2. Хід роботи

**Завдання (варіант 5).** Написати програму розв'язку системи лінійних алгебраїчних рівнянь у відповідності до варіанту:

- 1) методом Гауса з вибором головного елемента;
- 2) методом LU-розкладу.

$$\begin{cases} 0,92x_1 - 0,83x_2 + 0,62x_3 = 2,15 \\ 0,24x_1 - 0,54x_2 + 0,43x_3 = 0,62 \\ 0,73x_1 - 0,81x_2 - 0,67x_3 = 0,88 \end{cases}$$

### Вимоги до звіту

Звіт до лабораторної роботи повинен містити такі структурні елементи:

1. Титульний аркуш.
2. Тема.
3. Мета.
4. Короткі теоретичні відомості.
5. Алгоритм розв'язку СЛАР.
6. Текст програми з коментарями.
7. Вигляд реалізованої програми.
8. Висновки.

### Вимоги до програми

Програма має передбачати наступні можливості:

1. Автоматичний розв'язок СЛАР відповідними методами.
2. Ввід вхідних даних вручну: задати порядок СЛАР, елементи матриці коефіцієнтів та вільні члени системи.
3. Передбачити можливість некоректного введення даних.
4. Передбачити вивід покрокового виконання для кожного методу.

### 3. Текст програми методу Гауса з вибором головного елемента

```
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>

int const n = 3;
//якщо 0 - то ввід вхідних даних з клавуатури
int const testMode = 1;

void mainElem( int k, double mas[] [n + 1], int n, int xx[] ) {
    int i, j, i_max = k, j_max = k;
    double temp;

    //Шукаємо максимальний по модулю елемент
    for ( i = k; i < n; i++ )
        for ( j = k; j < n; j++ )
            if ( fabs( mas[i_max] [j_max] ) < fabs( mas[i] [j] ) )
            {
                i_max = i;
                j_max = j;
            }

    //Переставляємо рядки
    for ( j = k; j < n + 1; j++ ) {
        temp = mas[k] [j];
        mas[k] [j] = mas[i_max] [j];
        mas[i_max] [j] = temp;
    }

    //Переставляємо стовбці
    for ( i = 0; i < n; i++ ) {
        temp = mas[i] [k];
        mas[i] [k] = mas[i] [j_max];
        mas[i] [j_max] = temp;
    }

    //Враховуємо зміну порядку коренів
    i = xx[k];
    xx[k] = xx[j_max];
    xx[j_max] = i;
}

int main(void) {
    setlocale(LC_ALL, "Ukrainian");
    double mas[n] [n + 1];
    double x[n]; //Корені системи
    //Відповідає за порядок коренів
    int xx[n];
    int i, j, k;

    if(testMode) {
        mas[0][0] = 0.92;
        mas[0][1] = -0.83;
        mas[0][2] = 0.62;

        mas[1][0] = 0.24;
        mas[1][1] = -0.54;
        mas[1][2] = 0.43;

        mas[2][0] = 0.73;
        mas[2][1] = -0.81;
        mas[2][2] = -0.67;
```

```

        mas[0][n] = 2.15;
        mas[1][n] = 0.62;
        mas[2][n] = 0.88;

    } else {
        printf( "Введіть систему:\n" );
        for ( i = 0; i < n; i++ )
            for ( j = 0; j < n + 1; j++ )
                scanf( "%lf", &mas[i][j]);
    }

    //виведення введеної системи
    printf( "Система:\n" );
    for ( i = 0; i < n; i++ ) {
        for ( j = 0; j < n + 1; j++ )
            printf( "%7.2f ", mas[i][j] );
        printf( "\n" );
    }

    //спочатку всі корені по порядку
    for ( i = 0; i < n + 1; i++ ) xx[i] = i;

    //Прямий хід
    for ( k = 0; k < n; k++ )
    { //на якій позиції повинен стояти головний елемент
      mainElem( k, mas, n, xx ); //Встановлення головного елемента
      if ( fabs( mas[k][k] ) < 0.0001 )
      {
          printf( "Система не має жодного рішення !" );
          return ( 0 );
      }
      for ( j = n; j >= k; j-- )
          mas[k][j] /= mas[k][k];
      for ( i = k + 1; i < n; i++ )
          for ( j = n; j >= k; j-- )
              mas[i][j] -= mas[k][j] * mas[i][k];
    }

    //Зворотній хід
    for ( i = 0; i < n; i++ )
        x[i] = mas[i][n];
    for ( i = n - 2; i >= 0; i-- )
        for ( j = i + 1; j < n; j++ )
            x[i] -= x[j] * mas[i][j];

    //Вивід результату
    printf( "Результат:\n" );
    for ( i = 0; i < n; i++ )
        for ( j = 0; j < n; j++ )
            if ( i == xx[j] )
            { //розтавлення коренів по порядку
                printf( "%f\n", x[j] );
                break;
            }
    }

    _getch();
    return 0;
}

```

#### 4. Текст програми методу LU-розкладу

```
#include <iostream>
#include <conio.h>
#include <locale>

using namespace std;

int const N = 3;

//якщо 0 - то ввід вхідних даних з клавуатури
int const testMode = 1;

double a[N][N], b[N][N], L[N][N],
       U[N][N], x[N][N], y[N][N];

//зчитування масиву з клавіатури
void input(double a[N][N], int n, int m) {
    for(int i=0; i < n; i++)
        for(int j = 0; j < m; j++) {
            cout << "a[" << i << "][" << j << "]: ";
            cin >> a[i][j];
        }
}

//виведення матриці на екран
void display(double a[N][N], int n, int m) {
    for(int i=0; i < n; i++) {
        for(int j = 0; j < m; j++)
            cout << a[i][j] << "\t";
        if(m > 1) cout<<endl;
    }
}

//обчислення матриць L та U, де  $U \cdot L \cdot x = b$ 
void calculateLU () {
    for(int i = 0; i < N; i++) U[0][i] = a[0][i];
    for(int i = 1; i < N; i++) L[i][0] = a[i][0] / U[0][0];

    U[1][1] = a[1][1] - L[1][0] * U[0][1];
    L[2][1] = (a[2][1] - L[2][0] * U[0][1]) / U[1][1];
    U[1][2] = a[1][2] - L[1][0] * U[0][2];
    U[2][2] = a[2][2] - L[2][1] * U[1][2] - L[2][0] * U[0][2];

    for(int i = 0; i < N; i++) L[i][i] = 1;
}

//Обчислення  $Y = U \cdot x$ 
void calculateY() {
    y[0][0] = b[0][0];
    y[1][0] = b[1][0] - L[1][0] * y[0][0];
    y[2][0] = b[2][0] - L[2][0] * y[0][0] - L[2][1] * y[1][0];
}

//Обчислення X
void calculateX() {
    x[2][0] = y[2][0] / U[2][2];
    x[1][0] = (y[1][0] - U[1][2] * x[2][0]) / U[1][1];
    x[0][0] = (y[0][0] - U[0][2] * x[2][0] - U[0][1] * x[1][0]) / U[0][0];
}

void main() {
    setlocale(LC_ALL, "Ukrainian");
    if(testMode) {
        a[0][0] = 0.92;
```



```

        a[0][1] = -0.83;
        a[0][2] = 0.62;

        a[1][0] = 0.24;
        a[1][1] = -0.54;
        a[1][2] = 0.43;

        a[2][0] = 0.73;
        a[2][1] = -0.81;
        a[2][2] = -0.67;

        b[0][0] = 2.15;
        b[1][0] = 0.62;
        b[2][0] = 0.88;

    } else {
        cout<<"Введіть матрицю A:\n";
        input(a, N, N);
        cout<<"\nВведіть стовпець B:\n";
        input(b, N, 1);
    }

    //Вивід a та b
    cout<<"A:\n";
    display(a, N, N);
    cout<<"\nB:\n";
    display(b, N, 1);

    //обчислення L та U
    calculateLU();

    //Вивід L та U
    cout<<"\n\nL:\n";
    display(L, N, N);
    cout<<"\nU:\n";
    display(U, N, N);

    //обчислення Y та X
    calculateY();
    calculateX();

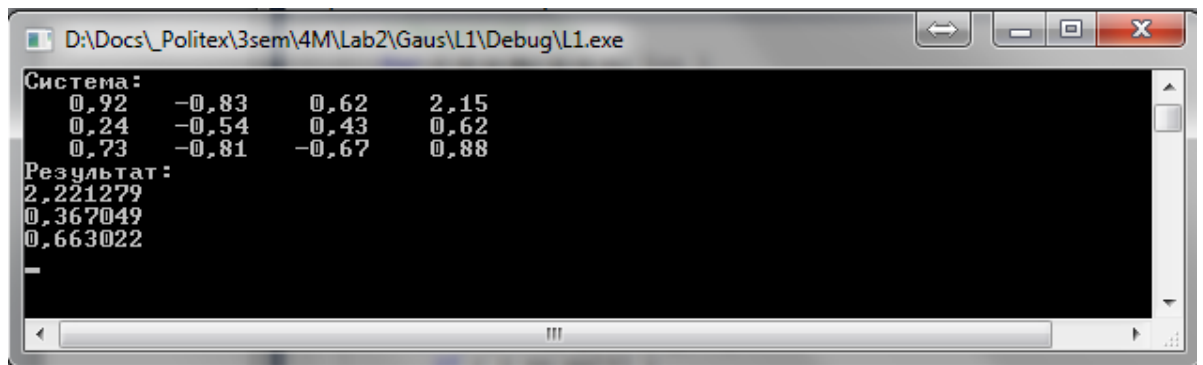
    //Вивід Y та X
    cout<<"\n\nY:\n";
    display(y, N, 1);
    cout<<"\n\nX:\n";
    display(x, N, 1);

    _getch();
}

```

## 5. Результати виконання програм

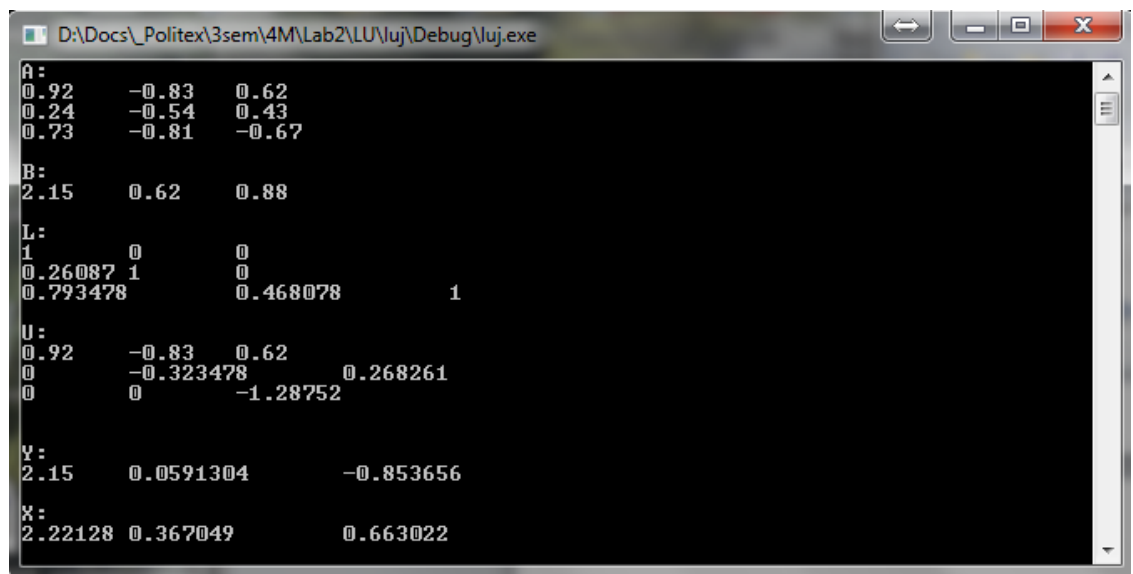
Запустимо програму методу Гауса на виконання (рис. 5.1):



```
D:\Docs\_Politex\3sem\4M\Lab2\Gaus\L1\Debug\L1.exe
Система:
  0,92   -0,83   0,62   2,15
  0,24   -0,54   0,43   0,62
  0,73   -0,81  -0,67   0,88
Результат:
2,221279
0,367049
0,663022
-
```

Рис. 5.1. Метод Гауса

Запустимо програму методу LU-розкладу на виконання (рис. 5.2):



```
D:\Docs\_Politex\3sem\4M\Lab2\LU\luj\Debug\luj.exe
A:
  0,92   -0,83   0,62
  0,24   -0,54   0,43
  0,73   -0,81  -0,67
B:
  2,15   0,62   0,88
L:
  1      0      0
  0,26087 1      0
  0,793478 0,468078 1
U:
  0,92   -0,83   0,62
  0      -0,323478 0,268261
  0      0      -1,28752
Y:
  2,15   0,0591304   -0,853656
X:
  2,22128 0,367049   0,663022
```

Рис. 5.2. Проміжні результати та реакція на помилки

Як бачимо ці дві програми повернули однаковий результат.

## **6. ВИСНОВКИ**

В ході даної лабораторної роботи я оволодів прямими методами розв'язування систем лінійних алгебраїчних рівнянь, а саме:

- 1)методом Гауса з вибором головного елемента;
- 2)методом LU-розкладу.