

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра прикладної математики

СИСТЕМИ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ

Звіт із лабораторної роботи №1

з дисципліни:

«Програмне забезпечення ЕОМ»

Виконав: МОСКАЛЬ О. М.

Група: КМ-23

Керівник:

КОВАЛЬЧУК-ХІМЮК Л.О.

Київ — 2015

ЗМІСТ

ВСТУП.....	3
1 ОПИС МЕТОДУ.....	4
2 ПОРЯДОК ВИКОНАННЯ РОБОТИ.....	5
3 ВАРІАНТ ЗАВДАННЯ.....	6
4 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	7
4.1 БЛОК-СХЕМА АЛГОРИТМУ РОБОТИ ПРОГРАМИ.....	7
4.2 ПРИКЛАД РОБОТИ ПРОГРАМИ.....	8
5 ПЕРЕВІРКА РЕЗУЛЬТАТІВ.....	9
ВИСНОВКИ	
СПИСОК ЛІТЕРАТУРИ	
ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ	

ВСТУП

Метою даної лабораторної роботи є програмна реалізація методу Холецького розв'язання системи лінійних алгебраїчних рівнянь.

1 ОПИС МЕТОДУ

Згідно з [1], розкладом Холецкого прийнято називати будь-яке з наступних представлень додатно визначеної симетричної матриці P :

$$P = L L^T, \quad P = \bar{L} D \bar{L}^T, \quad P = U U^T, \quad P = \bar{U} D \bar{U}^T,$$

де L — нижня трикутна матриця з додатними елементами на діагоналі, U — верхня трикутна матриця з додатними елементами на діагоналі, \bar{L} — нижня трикутна матриця з одиничними елементами на діагоналі, D — діагональна матриця з додатними елементами на діагоналі, \bar{U} — верхня трикутна матриця з одиничними елементами на діагоналі.

ТЕОРЕМА 1.1 (Нижній трикутний розклад Холецкого). Симетрична матриця $P > 0$ має розклад $P = L L^T$, де L — нижня трикутна матриця. Цей розклад з додатними діагональними елементами в L можна отримати наступним алгоритмом.

Для $j = 1, \dots, n-1$ виконувати цикл утворений наступним впорядкованим набором виразів:

$$\begin{aligned} L(j, j) &= P(j, j)^{1/2}, \\ L(k, j) &= P(k, j) / L(j, j), & k &= j+1, \dots, \\ P(i, k) &= P(i, k) - L(i, j) L(k, j) & \begin{cases} k = j+1, \dots, n \\ i = k, \dots, n \end{cases} \end{aligned}$$

2 ПОРЯДОК ВИКОНАННЯ РОБОТИ

Виконання лабораторної роботи повинно проходити наступним чином:

- 1) Ознайомитися з теоретичними відомостями про задачу, методи та засоби її розв'язку.
- 2) Виконати розв'язок даної задачі.
- 3) Створити блок-схему алгоритму рішення задачі.
- 4) Виконати перевірку та оцінити точність методу вирішення.
- 5) Скласти звіт про виконання роботи.

3 ВАРІАНТ ЗАВДАННЯ

Варіант (5, v1): методом Холецького розв'язати систему лінійних алгебраїчних рівнянь.

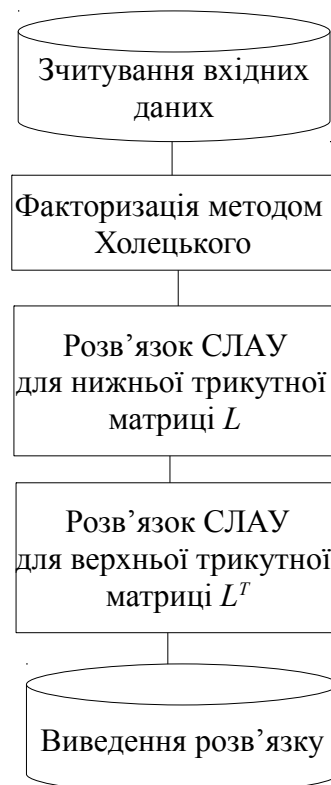
$$Ax=b$$
$$A=\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix} \quad b=\begin{pmatrix} 3 \\ 6 \\ 4 \\ 7 \end{pmatrix}$$

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

Для виконання поставленої задачі було розроблено програму на мові програмування C++(див. додаток А), яка реалізує метод Холецького для розв'язку системи лінійних алгебраїчних рівнянь.

Програма зчитує вхідні дані з текстового файлу та записує результат обчислень в текстовий файл. Ці файли вказуються користувачем, як параметри командного рядка. У файлі з вхідними даними користувач вказує розмірність системи, матрицю коефіцієнтів та стовпець вільних членів. У вихідному файлі міститиметься вектор розв'язок.

4.1 Блок-схема алгоритму роботи програми



4.2 Приклад роботи програми

```
mosk@lenovo:~/Labs/eom/lab1$ ./lab1 --help
Purpose: solving system of linear equations using Cholesky decomposition
Usage: lab1 -i input.txt -o output.txt
Options:
  --input, -i input.txt    set input file name
  --output, -o output.txt  set output file name
  --help, -h               print this help

Input file format:
First line:  1 number N(dimension)
Next N lines: N numbers(Matrix)
Next N line: 1 number(coefficient vector)
mosk@lenovo:~/Labs/eom/lab1$ cat input.txt
4
4 3 2 1
3 6 4 2
2 4 6 3
1 2 3 4
3
6
4
7
mosk@lenovo:~/Labs/eom/lab1$ ./lab1 -i input.txt -o output.txt
mosk@lenovo:~/Labs/eom/lab1$ cat output.txt
5.55112e-17
1
-1
2
```


5 ПЕРЕВІРКА РЕЗУЛЬТАТІВ

Використовуючи математичний пакет MATLAB було розв'язано систему рівнянь зазначену в розділі 3. (рис. 5.1).

```
>> A = [4,3,2,1;3,6,4,2;2,4,6,3;1,2,3,4]

A =
     4     3     2     1
     3     6     4     2
     2     4     6     3
     1     2     3     4

>> b = [3;6;4;7]

b =
     3
     6
     4
     7

>> x = linsolve(A,b)

x =
     0
     1
    -1
     2

>>
```

Рисунок 5.1 - Розв'язок знайдений за допомогою MATLAB

Судячи з результату отриманого за допомогою MATLAB можна зробити висновок, що розв'язок отриманий використовуючи дану програмну реалізацію методу Холецкого є вірним.

ВИСНОВКИ

У цій лабораторній роботі було розглянуто та програмно реалізовано метод Холецького розв'язку системи лінійних алгебраїчних рівнянь.

СПИСОК ЛИТЕРАТУРЫ

1. Вычислительные методы алгебры и оценивания : учебное пособие / И. В. Семушин. — Ульяновск : УлГТУ, 2011. — 366 с.

ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <getopt.h>
#include <cmath>

using namespace std;

class matrix
{
    double *array;
public:
    const int dim;
    matrix(int n): dim(n)
    {
        array = new double[n*n];
    }
    double& operator()(int i, int j)
    {
        return array[i*dim + j];
    }
    ~matrix()
    {
        delete[] array;
    }
};

void cholesky(matrix &in, matrix &out)
{
    for (int i = 0; i < in.dim; i++)
        for (int j = 0; j < in.dim; j++)
            out(i, j) = 0.;

    for (int j = 0; j < in.dim; j++)
    {
        out(j, j) = sqrt(in(j, j));
        for (int k = j + 1; k < in.dim; k++)
            out(k, j) = in(k, j) / out(j, j);

        for (int k = j + 1; k < in.dim; k++)
            for (int i = k; i < in.dim; i++)
                in(i, k) = in(i, k) - out(i, j)*out(k, j);
    }
}

std::vector<double>
solve_lower_triangle_linear_system(matrix &A, const std::vector<double> &b)
{
    std::vector<double> x(A.dim, 0);
    for (int i = 0; i < A.dim; i++)
    {
        double b_i = b[i];
        for (int j = 0; j < i; j++)
            b_i -= A(i, j) * x[j];
        x[i] = b_i / A(i, i);
    }
    return x;
}

std::vector<double>
solve_upper_triangle_linear_system(matrix &A, const std::vector<double> &b)
{
    std::vector<double> x(A.dim, 0);
    for (int i = A.dim - 1; i >= 0; i--)
    {
        double b_i = b[i];
        for (int j = i + 1; j < A.dim; j++)
            b_i -= A(i, j) * x[j];
        x[i] = b_i / A(i, i);
    }
}
```

```

    return x;
}

void print_usage()
{
    cout << "Purpose: solving system of linear equations using Cholesky decomposition" << endl
        << "Usage: lab1 -i input.txt -o output.txt" << endl
        << "Options:" << endl
        << "  --input, -i input.txt    set input file name" << endl
        << "  --output, -o output.txt  set output file name" << endl
        << "  --help, -h              print this help" << endl
        << endl
        << "Input file format:" << endl
        << "First line:   1 number N(dimension)" << endl
        << "Next N lines: N numbers(Matrix)" << endl
        << "Next N line:  1 number(coefficient vector)" << endl;
}

int main(int argc, char **argv)
{
    std::string input_file_name = "input.txt";
    std::string output_file_name = "output.txt";
    struct option options[] = {
        {"input", required_argument, NULL, 'i'},
        {"output", required_argument, NULL, 'o'},
        {"help", no_argument, NULL, 'h'},
        {0,0,0,0}
    };
    int opt;
    while ( (opt = getopt_long(argc, argv, "i:o:h", options, NULL)) != -1 )
    {
        if (opt == 'i')
            input_file_name = optarg;
        else if (opt == 'o')
            output_file_name = optarg;
        else
        {
            print_usage();
            return 0;
        }
    }
    ifstream in(input_file_name.c_str());
    ofstream out(output_file_name.c_str());

    int dimension;
    in >> dimension;

    matrix A(dimension), U(dimension);
    std::vector<double> b(dimension);
    for (int i = 0; i < dimension; i++)
        for (int j = 0; j < dimension; j++)
            in >> A(i, j);
    for (int i = 0; i < dimension; i++)
        in >> b[i];

    for (int i = 0; i < dimension; i++)
        for (int j = i; j < dimension; j++)
            if (A(i, j) != A(j, i))
            {
                cout << "Input matrix is not symetric" << endl;
                return 0;
            }
    cholesky(A, U);
    vector<double> b2 = solve_lower_triangle_linear_system(U, b);
    for (int i = 0; i < dimension; i++)
        for (int j = 0; j <= i; j++)
            std::swap(U(i, j), U(j, i));
    vector<double> x = solve_upper_triangle_linear_system(U, b2);

    for (int i = 0; i < dimension; i++)
        out << x[i] << std::endl;

    return 0;
}

```