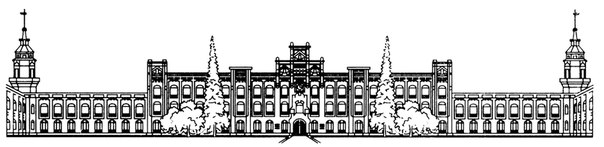
Національний технічний університет України «КПІ ім. Ігоря Сікорського»



Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота № 2

з дисципліни «Спеціальні розділи математики-2.  
Чисельні методи»

на тему

***“Методи розв’язання систем лінійних алгебраїчних рівнянь (СЛАР)”***

Виконав:

студент групи ІС-31

Коваль Богдан

Викладач:

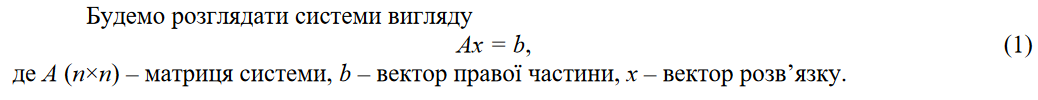
доц. Рибачук Л.В.

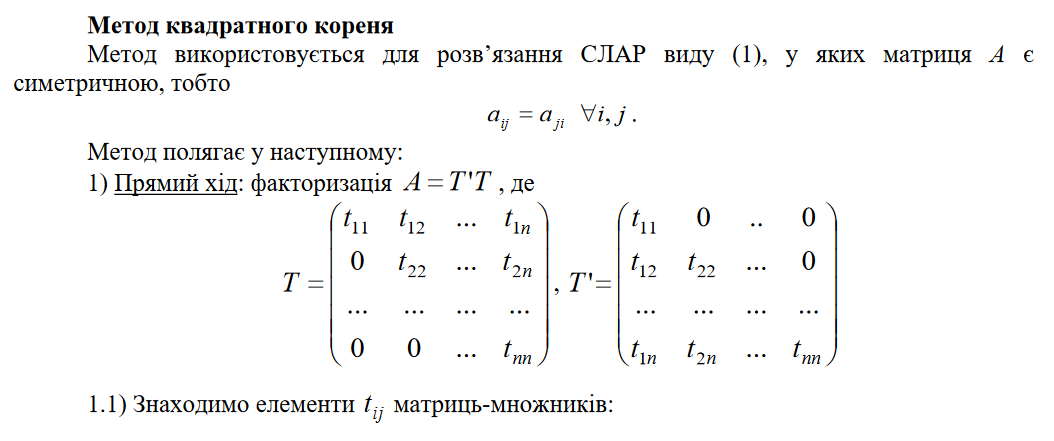
Київ – 2024

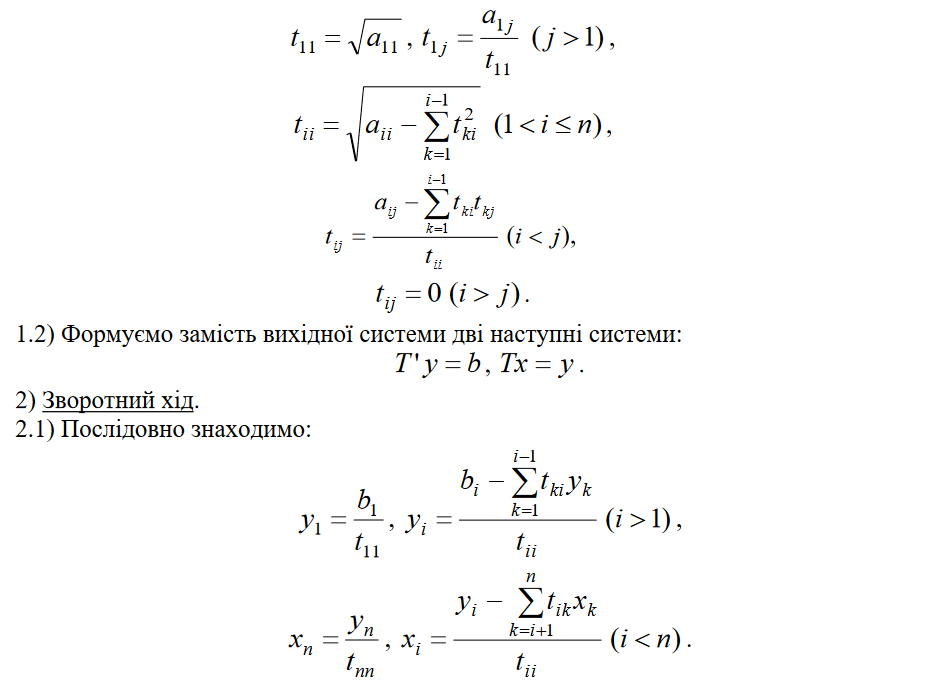
Зміст

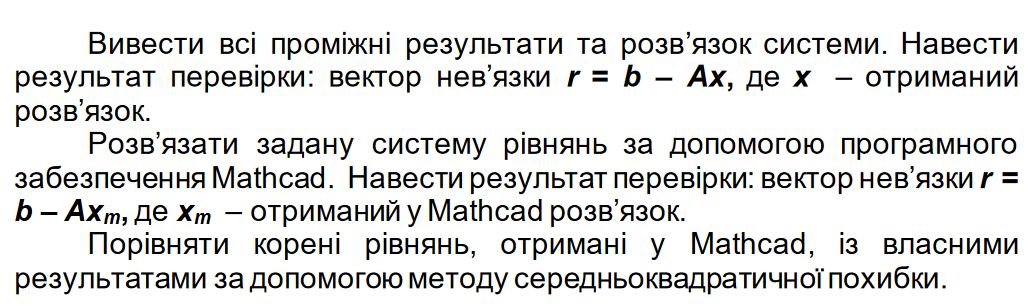
1. Постановка задачі
2. Алгоритм роботи програми
3. Опис роботи програми
4. Розв’язок
5. Висновок

**1 Постановка задачі**



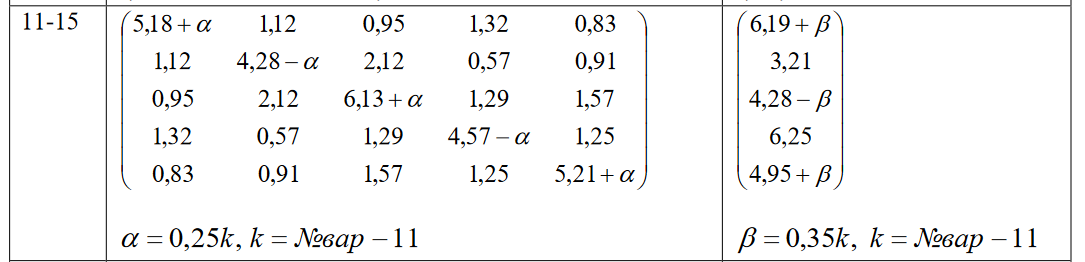




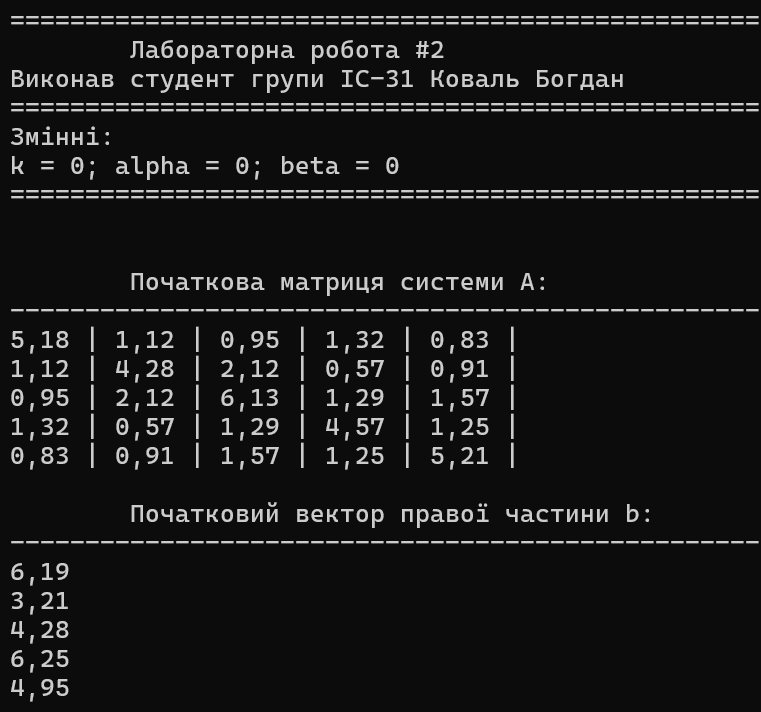
******

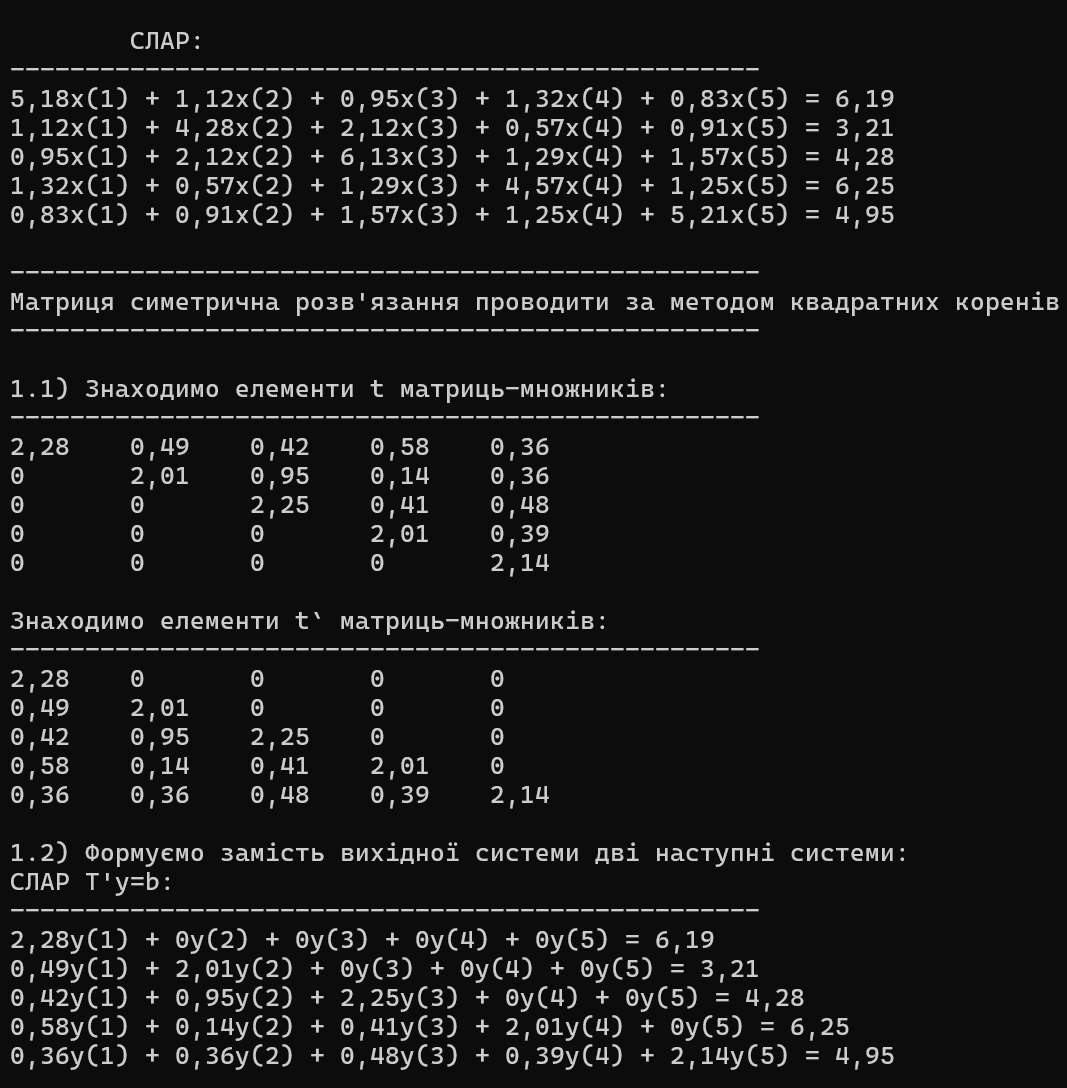
**2 Алгоритм роботи програми**

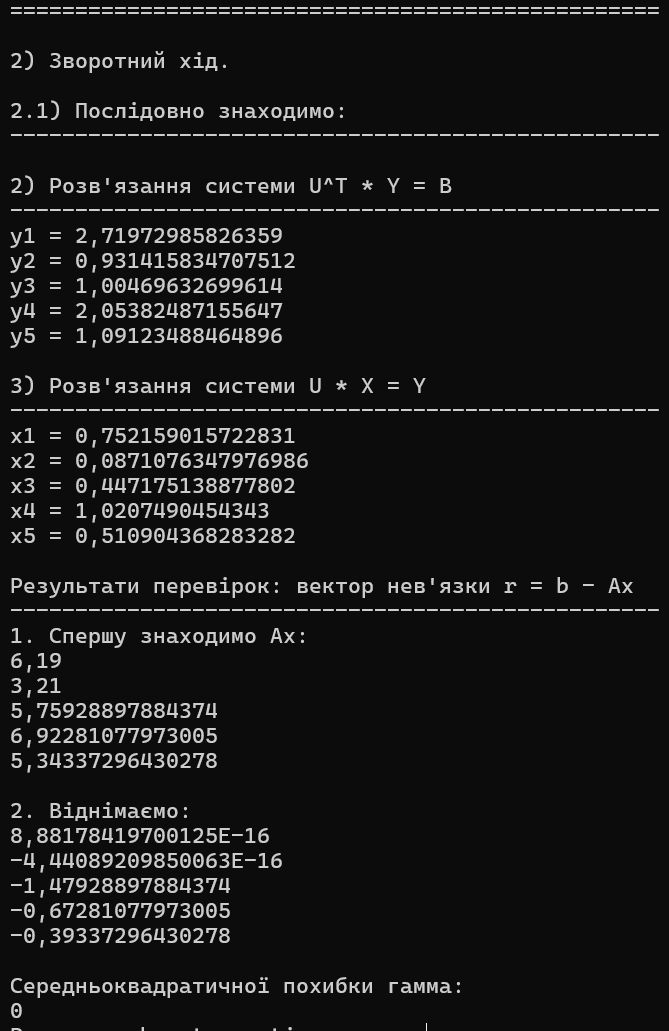
Для вирішення СЛАР поставленої задачі необхідно використати метод квадратного кореня.

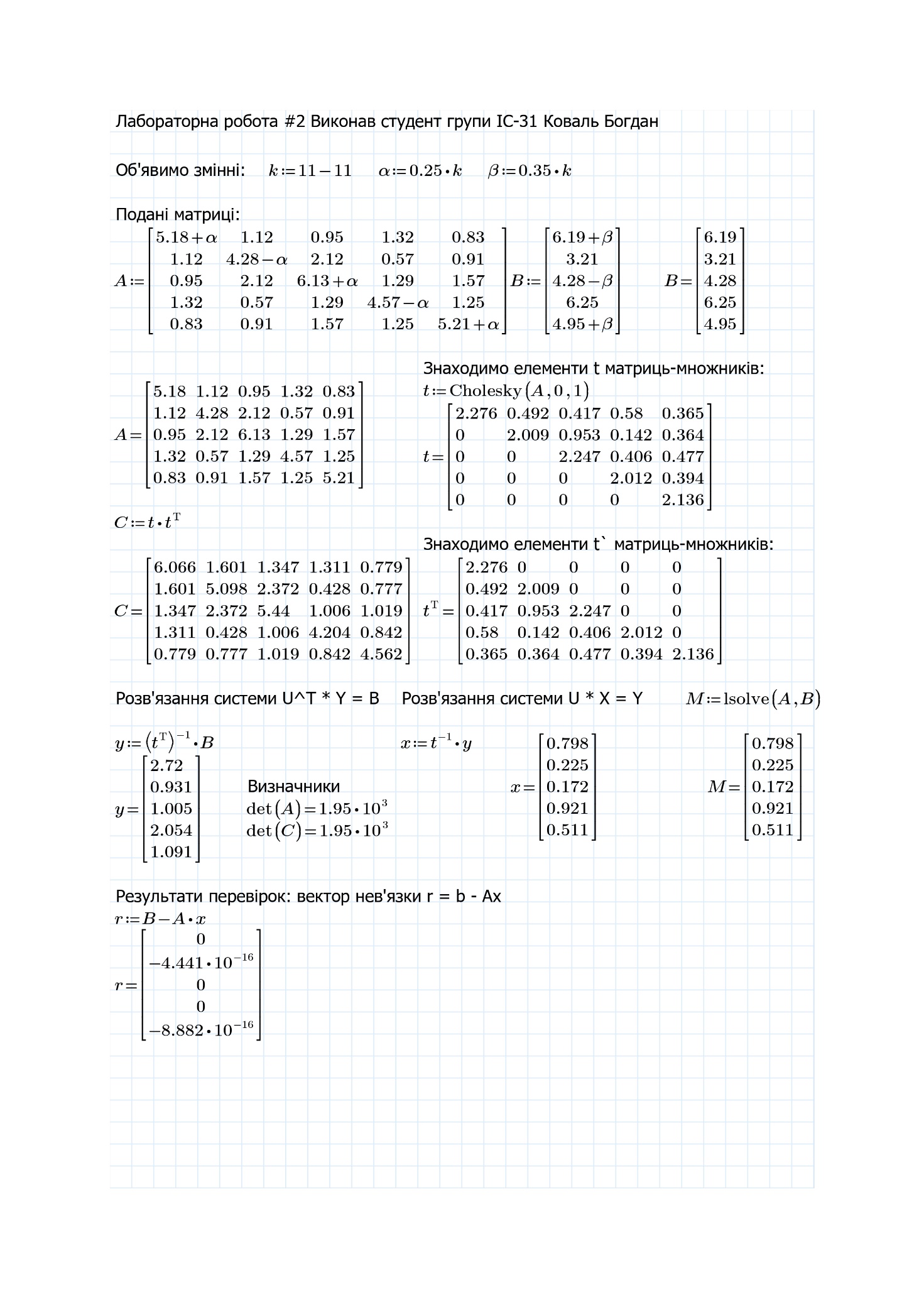
1. **Введення даних**:
   * Коефіцієнти матриці системи лінійних рівнянь.
   * Вектор вільних членів.
2. **Перевірка можливості застосування методу квадратного кореня**:
   * Метод квадратного кореня може бути застосований, якщо матриця системи є симетричною та додатньо визначеною.
3. **Факторизація матриці системи**:
   * Розкладання матриці системи на добуток верхньої трикутної (або нижньої) та її транспонованої.
   * Для цього можна використовувати метод Холецького або інші методи факторизації.
4. **Розв'язання системи лінійних рівнянь**:
   * Розв'язання двох послідовних систем з треугольними матрицями, що здійснюється методом зворотнього ходу.
   * Один раз для системи з верхньою трикутною матрицею та ще раз для транспонованої системи з нижньою трикутною матрицею.
5. **Виведення результатів**:
   * Виведення вектора невідомих, який є розв'язком системи.

**Проміжні результати та кінцевий результат;**

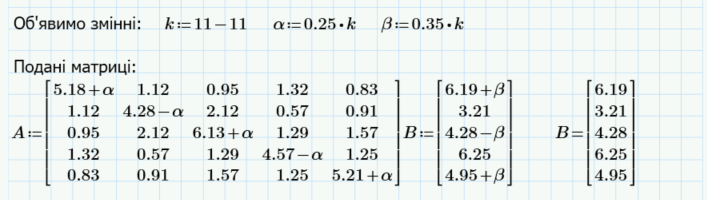


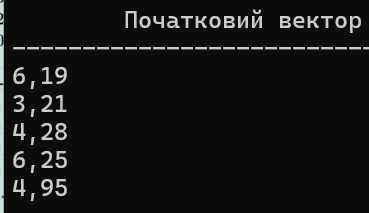
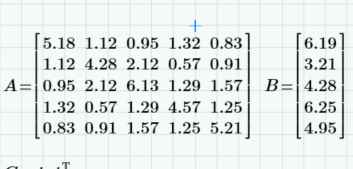


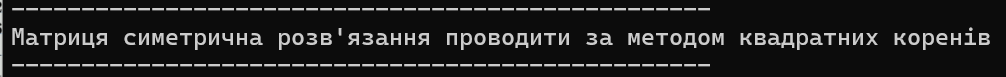


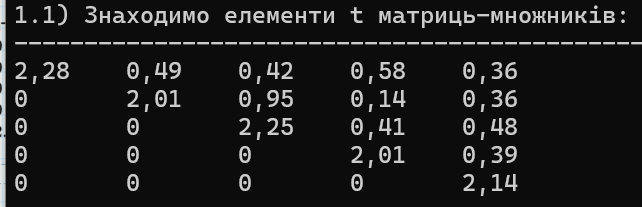
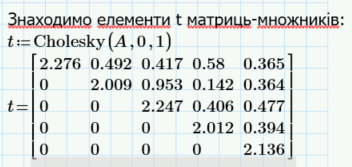


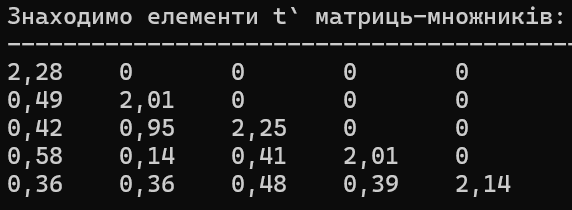
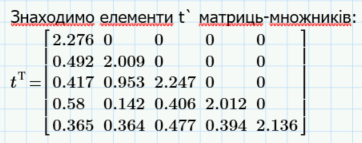
**Порівняння власного розв’язку та розв’язку, отриманого у Mathcad.**

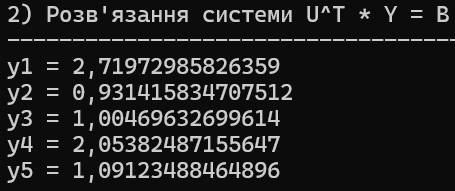
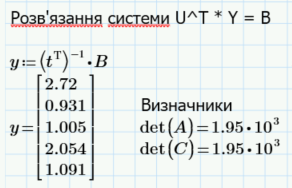


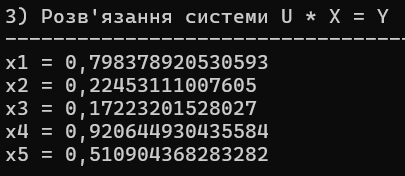
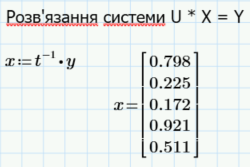


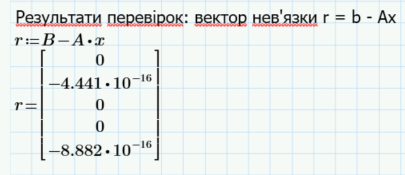
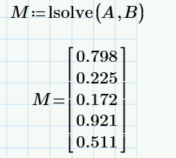


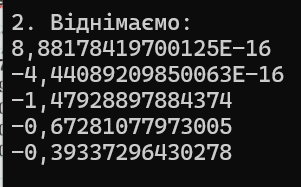
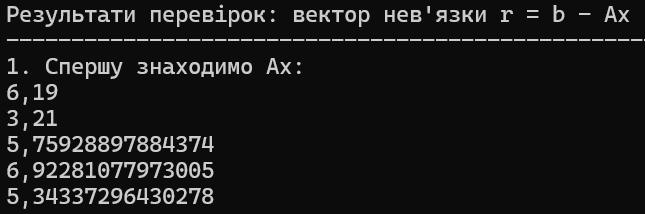










****

**Лістинг програми:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

namespace lab\_2

{

internal class Program

{

private static readonly int j;

static void PrintSLAR(double[,] Matrix, double[,] Vector, string tmp\_unk = "x")

{

for (int i = 0; i < Matrix.GetLength(0); i++)

{

for (int j = 0; j < Matrix.GetLength(1); j++)

{

Console.Write(Math.Round(Matrix[i, j], 2) + $"{tmp\_unk}({j + 1})");

Console.Write((j < Matrix.GetLength(0) - 1) ? " + " : "");

}

Console.WriteLine(" = " + Math.Round(Vector[i, 0], 2));

}

}

static void PrintMatrix(double[,] Matrix)

{

for (int i = 0; i < Matrix.GetLength(0); i++)

{

for (int j = 0; j < Matrix.GetLength(1); j++)

{

Console.Write(Math.Round(Matrix[i, j], 2) + "\t");

}

Console.WriteLine();

}

}

static void PrintSep()

{

Console.WriteLine(string.Concat(Enumerable.Repeat("=", 50)));

}

static void PrintSepT()

{

Console.WriteLine(string.Concat(Enumerable.Repeat("-", 50)));

}

static void Main(string[] args)

{

PrintSep();

Console.WriteLine("\tЛабораторна робота #2");

Console.WriteLine("Виконав студент групи IC-31 Коваль Богдан");

PrintSep();

double alpha, k, beta;

bool Flag\_symmetry = true;

k = 11 - 11;

alpha = 0.25 \* k;

beta = 0.35 \* k;

Console.WriteLine("Змiннi:");

Console.WriteLine($"k = {k}; alpha = {alpha}; beta = {beta}");

PrintSep();

Console.WriteLine("\n\n\tПочаткова матриця системи А:");

PrintSepT();

double[,] Matrix\_A =

{

{5.18 + alpha, 1.12, 0.95, 1.32, 0.83},

{1.12, 4.28 - alpha, 2.12, 0.57, 0.91},

{0.95, 2.12, 6.13 + alpha, 1.29, 1.57},

{1.32, 0.57, 1.29, 4.57 - alpha, 1.25 },

{0.83, 0.91, 1.57, 1.25, 5.21 + alpha },

};

double[,] Vector\_b =

{

{6.19 + beta},

{3.21},

{4.28 - beta},

{6.25},

{4.95 + beta},

};

for (int i = 0; i < Matrix\_A.GetLength(0); i++)

{

for (int j = 0; j < Matrix\_A.GetLength(1); j++)

{

Console.Write(Math.Round(Matrix\_A[i, j], 2) + " | ");

if (Matrix\_A[i, j] != Matrix\_A[j, i]) Flag\_symmetry = false;

}

Console.WriteLine();

}

Console.WriteLine("\n\tПочатковий вектор правої частини b:");

PrintSepT();

PrintMatrix(Vector\_b);

Console.WriteLine("\n\tCЛАР:");

PrintSepT();

PrintSLAR(Matrix\_A, Vector\_b);

Console.WriteLine();

PrintSepT();

Console.WriteLine((Flag\_symmetry) ? "Maтриця симетрична розв’язання проводити за методом квадратних коренiв" : "Матриця не симетрична використати метод Гауса");

PrintSepT();

Console.WriteLine();

double[,] Matrix\_T = new double[Matrix\_A.GetLength(0), Matrix\_A.GetLength(1)];

Matrix\_T[0, 0] = Math.Sqrt(Matrix\_A[0, 0]);

double[,] Matrix\_Ttr = new double[Matrix\_A.GetLength(0), Matrix\_A.GetLength(1)];

Console.WriteLine("1.1) Знаходимо елементи t матриць-множникiв:");

PrintSepT();

for (int i = 0; i < Matrix\_A.GetLength(0); i++)

{

for (int j = i; j < Matrix\_A.GetLength(1); j++)

{

if (j > 0)

{

Matrix\_T[0, j] = Matrix\_A[0, j] / Matrix\_T[0, 0];

}

if (0 < i && i <= Matrix\_A.GetLength(0))

{

double tmp = 0;

for (int h = 0; h <= i - 1; h++)

{

tmp += (Math.Pow(Matrix\_T[h, i], 2));

}

Matrix\_T[i, i] = Math.Sqrt(Matrix\_A[i, i] - tmp);

}

if (i < j)

{

double temp = 0;

for (int p = 0; p <= i - 1; p++)

{

temp += (Matrix\_T[p, i] \* Matrix\_T[p, j]);

}

Matrix\_T[i, j] = (Matrix\_A[i, j] - temp) / (Matrix\_T[i, i]);

}

if (i > j)

{

Matrix\_T[i, j] = 0;

}

}

}

PrintMatrix(Matrix\_T);

Console.WriteLine("\nЗнаходимо елементи t` матриць-множникiв:");

PrintSepT();

for (int i = 0; i < Matrix\_Ttr.GetLength(0); i++)

{

for (int j = 0; j < Matrix\_T.GetLength(1); j++)

{

Matrix\_Ttr[i, j] = Matrix\_T[j, i];

Console.Write(Math.Round(Matrix\_Ttr[i, j], 2) + "\t");

}

Console.WriteLine();

}

Console.WriteLine("\n1.2) Формуємо замiсть вихiдної системи двi наступнi системи:");

Console.WriteLine("CЛАР T'y=b:");

PrintSepT();

PrintSLAR(Matrix\_Ttr, Vector\_b, "y");

double[] Vector\_y = new double[Matrix\_T.GetLength(0)];

double tempSum;

Console.WriteLine('\n');

PrintSep();

Console.WriteLine("\n2) Зворотний хiд.\n");

Console.WriteLine("2.1) Послiдовно знаходимо:");

PrintSepT();

Console.WriteLine("\n2) Розв'язання системи U^T \* Y = B");

PrintSepT();

for (int i = 0; i < Vector\_b.GetLength(0); i++)

{

if (i >= 1)

{

tempSum = 0;

for (int j = 0; j <= i - 1; j++)

{

tempSum += Matrix\_Ttr[i, j] \* Vector\_y[j];

}

Vector\_y[i] = (Vector\_b[i, 0] - tempSum) / Matrix\_Ttr[i, i];

}

else

{

Vector\_y[i] = Vector\_b[0, i] / Matrix\_Ttr[i, i];

}

Console.WriteLine($"y{i + 1} = {Vector\_y[i]}");

}

double[] Vector\_x = new double[Matrix\_T.GetLength(0)];

int n = Matrix\_T.GetLength(0) - 1;

Console.WriteLine("\n3) Розв'язання системи U \* X = Y");

PrintSepT();

for (int i = n; i >= 0; i--)

{

if (i < n)

{

tempSum = 0;

for (int j = i + 1; j <= n; j++)

{

tempSum += Matrix\_T[i, j] \* Vector\_x[j];

}

Vector\_x[i] = (Vector\_y[i] - tempSum) / Matrix\_T[i, i];

}

else

{

Vector\_x[i] = Vector\_y[i] / Matrix\_T[i, i];

}

}

for (int i = 0; i < Vector\_x.GetLength(0); i++)

{

Console.WriteLine($"x{i + 1} = {Vector\_x[i]}");

}

Console.WriteLine("\nРезультати перевiрок: вектор нев’язки r = b – Ax");

PrintSepT();

double sumElem;

int LenghtVec = Vector\_x.GetLength(0);

double[] Vector\_tmp = new double[LenghtVec];

Console.WriteLine("1. Спершу знаходимо Ax:");

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

{

sumElem = 0;

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

{

sumElem += Matrix\_A[i, j] \* Vector\_x[j];

}

Vector\_tmp[i] = sumElem;

Console.WriteLine(Vector\_tmp[i]);

}

Console.WriteLine("\n2. Вiднiмаємо:");

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

{

Console.WriteLine(Vector\_b[i, 0] - Vector\_tmp[i]);

}

double fault;

double tmp\_suma = 0;

int nLenght = Vector\_x.GetLength(0);

double[] Vector\_xMad = new double[] { 0, -4.441 \* Math.Pow(10, -16), 0, 0, -8.882 \* Math.Pow(10, -16) };

Console.WriteLine("\nСередньоквадратичної похибки гамма:");

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

{

tmp\_suma += Math.Pow((Vector\_x[i] - Vector\_xMad[i]), 2);

}

fault = Math.Sqrt((1 / nLenght) \* tmp\_suma);

Console.WriteLine(fault);

}

}

}