Отчёт по лабораторной работе №2

Денисов-Элерс Э.Ф, 5130904/40003, вариант: 34

1. Задание

Составить процедуру вычисления матрицы A вида

$$A = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ X_1^{N-1} & \cdots & X_N^{N-1} \end{pmatrix}$$

Используя её, решить несколько раз линейную систему $Az = b$, где $N = 5, 7, 9; b_k = 2^k + \cos(k)\,,\ k = 1, \ldots, N; x_k = k,\ k = 1, \ldots, N-1; x_n = 1.1, 1.01, 1.001, 1.0001$. Системы решать, используя программы **DECOMP** и **SOLVE**. Сделать необходимые выводы.

2. Описание решения

Язык: Java

Использованные методы из библиотеки:

- DECOMP – метод для осуществления LU-разложения.
- SOLVE– метод для решения двух систем с треугольными матрицами L и U (Ly=b и Ux=y).

3. Результаты

```
============================================================ N = 5 ============================================================
Xn = 1,1000: z=[1.25e+01, 1.74e+00, 3.72e-01, -8.05e-02, -1.20e+01], cond=5.3842e+04, flag=0
Xn = 1,0100: z=[1.02e+02, 1.44e+00, 4.42e-01, -9.08e-02, -1.01e+02], cond=4.2781e+05, flag=0
Xn = 1,0010: z=[9.93e+02, 1.41e+00, 4.49e-01, -9.18e-02, -9.93e+02], cond=4.1804e+06, flag=0
Xn = 1,0001: z=[9.91e+03, 1.41e+00, 4.50e-01, -9.19e-02, -9.91e+03], cond=4.1707e+07, flag=0
============================================================ N = 7 ============================================================
Xn = 1,1000: z=[5.13e+00, -1.97e+00, 3.53e+00, -1.91e+00, 5.93e-01, -8.05e-02, -2.74e+00], cond=1.9977e+07, flag=0
Xn = 1,0100: z=[2.46e+01, -2.08e+00, 3.58e+00, -1.94e+00, 6.00e-01, -8.13e-02, -2.22e+01], cond=1.3315e+08, flag=0
Xn = 1,0010: z=[2.20e+02, -2.09e+00, 3.59e+00, -1.94e+00, 6.00e-01, -8.14e-02, -2.17e+02], cond=1.2726e+09, flag=0
Xn = 1,0001: z=[2.17e+03, -2.09e+00, 3.59e+00, -1.94e+00, 6.00e-01, -8.14e-02, -2.17e+03], cond=1.2668e+10, flag=0
============================================================ N = 9 ============================================================
Xn = 1,1000: z=[-3.27e+00, -8.04e+00, 1.15e+01, -9.90e+00, 6.00e+00, -2.41e+00, 5.73e-01, -6.13e-02, 8.15e+00], cond=1.5077e+10, flag=0
Xn = 1,0100: z=[-5.94e+01, -7.60e+00, 1.12e+01, -9.67e+00, 5.87e+00, -2.36e+00, 5.62e-01, -6.02e-02, 6.41e+01], cond=7.5884e+10, flag=0
Xn = 1,0010: z=[-6.21e+02, -7.56e+00, 1.12e+01, -9.65e+00, 5.86e+00, -2.35e+00, 5.61e-01, -6.01e-02, 6.26e+02], cond=6.9175e+11, flag=0
Xn = 1,0001: z=[-6.24e+03, -7.56e+00, 1.12e+01, -9.65e+00, 5.86e+00, -2.35e+00, 5.61e-01, -6.01e-02, 6.24e+03], cond=6.8511e+12, flag=0
```

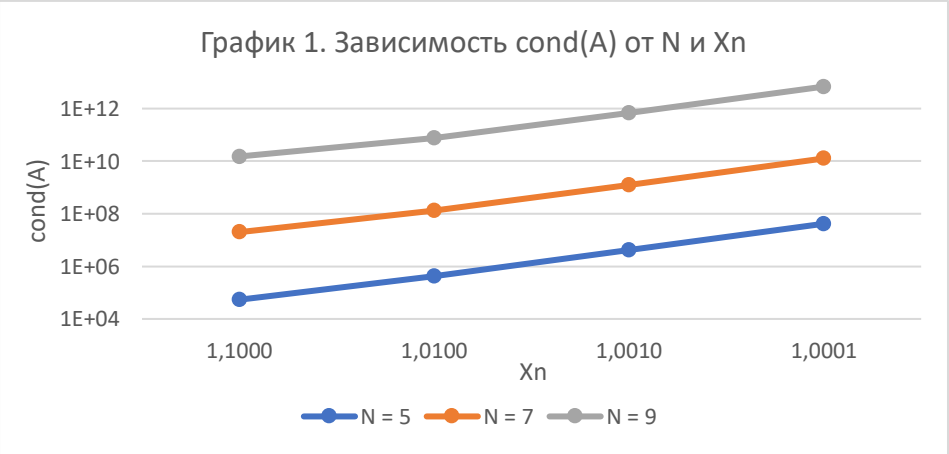
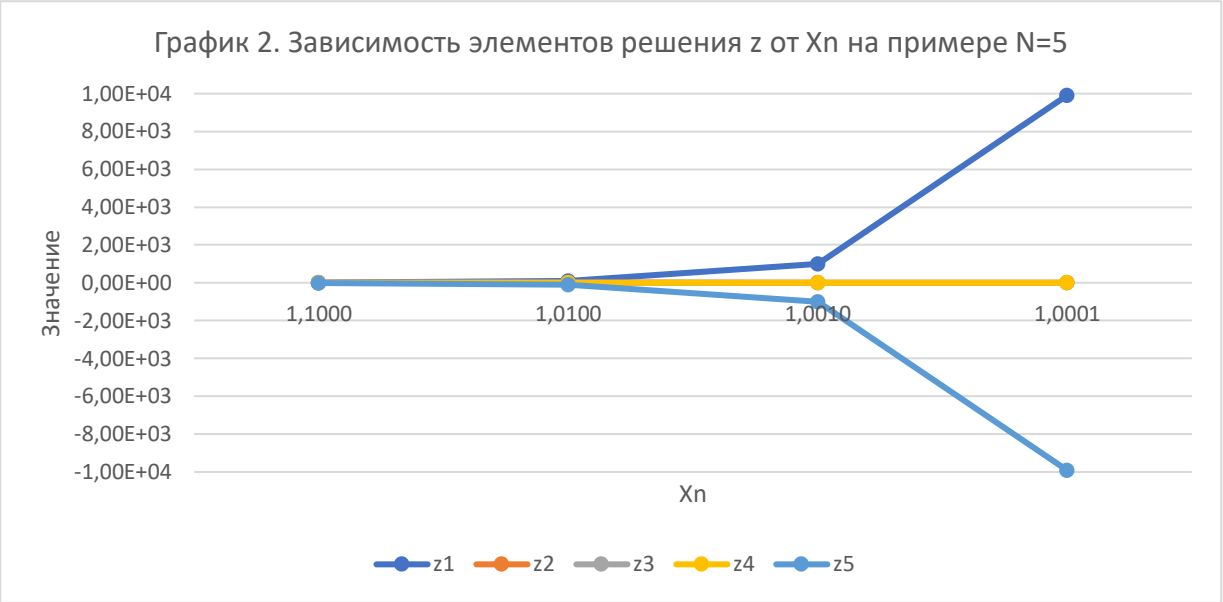


График 1. Зависимость cond(A) от N и Xn



4. Анализ результатов

При решении системы $Az = b$ (1) с различными размерностями матрицы A ($N = 5, 7, 9$) и значениями последнего узла $x_n = 1.1,\ 1.01,\ 1.001,\ 1.0001$ виден сильный рост числа обусловленности $cond(A)$ как при увеличении N, так и при приближении x_n к единице. На графике 1 видно, что оба этих параметра влияют на $cond(A)$, вызывая его увеличение. Программа DECOMP при этом отработала корректно, так как флаг равен нулю. Следовательно, рост $cond(A)$ обусловлен свойствами самой матрицы.

Плохая обусловленность матрицы может стать причиной большого изменения решения при малом изменении исходных данных. Это наглядно показано на графике 2. При $x_n \to 1$ наблюдается резкий рост крайних компонент ($z1, z5$). Их значения приведены в таблице:

	$x_n = 1.1$	$x_n = 1.0001$
z1	$1.25 \cdot 10^1$	$9.91 \cdot 10^3$
z5	$-1.20 \cdot 10^1$	$-9.91 \cdot 10^3$

Заметим, что матрица A является матрицей Вандермонда. Для неё характерен сильный рост $cond(A)$ при увеличении размерности. Анализ, проведенный выше, подтверждает это свойство. Следовательно, использование матрицы Вандермонда большой размерности при построении интерполяционного полинома может привести к «вычислительной катастрофе» из-за превышения критического числа обусловленности. В такой ситуации рекомендуется выбрать более устойчивый метод. Например, можно использовать ортогональные полиномы Чебышёва или Лежандра, либо выполнить сплайн-интерполяцию.

Приложение 1. Код программы

Lab2

```
package com.edwin.lab2;

import com.edwin.sources.decompSolve.Decomp;
import com.edwin.sources.decompSolve.Solve;

public class Lab2 {
    final static int[] nToCheck = new int[]{5, 7, 9};
    final static double[] xNToCheck = new double[]{1.1, 1.01, 1.001, 1.0001};

    public static void main(String[] args) {
        for (var n: nToCheck) {
            System.out.println("=".repeat(64) + " N = " + n + " " +
"=".repeat(64));
            for (var xN: xNToCheck) {
                var solution = calculate(n, xN);
                System.out.print("Xn = " + String.format( "%.4f", xN ) + ":
");
                System.out.println(solution);
            }
        }
    }

    public static Solution calculate(int n, double xN) {
        var xRow = generateXRow(n, xN);
        var a = fillVandermonde(xRow);
        var pivot = new int[n];
        var decomp = new Decomp(n, n, a, 0.0, pivot, 0);
        decomp.decomp();
        var b = generateB(n);
        var solve = new Solve(n, n, a, b, pivot);
        solve.solve();

        return new Solution(b, decomp.getCond(), decomp.getFlag());
    }

    private static double[][] fillVandermonde(double[] x) {
        int n = x.length;
        double[][] a = new double[n][n];

        for (int i = 0; i < n; i++)
            a[0][i] = 1;

        for (int i = 1; i < n; i++)
            for (int j = 0; j < n; j++)
                a[i][j] = a[i-1][j] * x[j];

        return a;
    }

    private static double[] generateXRow(int n, double xN) {
        double[] row = new double[n];
        for (int k = 1; k <= n - 1; k++)
            row[k - 1] = k;
        row[n - 1] = xN;

        return row;
    }
```

```
    private static double[] generateB(int n) {
        double[] b = new double[n];
        for (int k = 1; k <= n; k++)
            b[k - 1] = Math.pow(2, k) + Math.cos(k);

        return b;
    }
      }
```

Solution

```
package com.edwin.lab2;

public record Solution(double[] z, double cond, int flag) {
    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("z=[");
        for (int i = 0; i < z.length; i++) {
            if (i > 0) sb.append(", ");
            sb.append(String.format("%.2e", z[i]));
        }
        sb.append("], cond=").append(String.format("%.4e", cond)).append(",
flag=").append(flag);

        return sb.toString();
    }
}
```