МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «Крымский федеральный университет имени В. И. Вернадского»
Физико-технический институт
Кафедра компьютерной инженерии и моделирования

Лабораторная работа №2
по курсу «Алгоритмы и методы вычислений»
на тему: «Решение систем
линейных алгебраических уравнений (СЛАУ)»

Выполнил:
студент 2 курса
группы ИВТ-202(1)
Шор К.А

Проверила:
старший преподаватель
кафедры компьютерной
инженерии и моделирования
Горская И.Ю.

Симферополь, 2022

Лабораторная работа № 2

Тема: Решение систем линейных алгебраических уравнений(СЛАУ)

Цель работы:

1. Изучить и научиться использовать на практике наиболее эффективные прямые и итерационные алгоритмы решения СЛАУ.

2. Написать программу, реализующую два метода численного решения СЛАУ в следующих комбинациях: метод исключения Гаусса с выборкой ведущего элемента, (или Гаусса-Жордана) и Гаусса-Зайделя, или метод группы градиентного (наискорейшего) спуска.

Перед выполнением лабораторной работы:

1. Изучена презентация лектора курса: «Решение систем линейных алгебраических уравнений (СЛАУ)», материалы доступны на One Drive.

2. Прочитаны соответствующие разделы книги: Каханер Д., Моулер К., Нэш С. Численные методы и математическое обеспечение: Пер. с англ.- М.: Мир, 1998.- 575 с., Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры.- М., Л.:«Наука» .- 1963г., А.Н.Тихонов и др. Регуляризирующие алгоритмы и априорная информация .- М.: «Наука» .- 1983.

В соответствии с индивидуальным заданием, решены следующие задачи:

1. Изучены и закреплены на практике наиболее эффективные прямые и итерационные алгоритмы решения СЛАУ;

2. Реализована программа решения СЛАУ метод Гаусса-Жордана с выбором ведущего элемента, а также методом Зейделя.

Ход работы

Вариант 12

$$12.\quad A = \begin{bmatrix} 3 & 1 & -1 \\ 1 & 5 & -1 \\ 2 & 0 & 3 \end{bmatrix}, \qquad b = \begin{bmatrix} -2 \\ 8 \\ 1 \end{bmatrix}$$

1. **Метод Гаусса-Жордана**

   1) Есть система и первое что с ней надо сделать это выбрать текущий элемент. Находим строку у которой этот элемент самый большой и закидываем её на верх. Тем сама мы выбираем рабочую строку.

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

   2) После чего приводим системы к треугольному виды по диагонали. Получим, в последней строке, одно уравнение с одной неизвестной. Находим эту неизвестную и начинаем возвращаться назад.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ \quad a_{22}^{(1)}x_2 + \ldots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad a_{mn}^{(m-1)}x_n = b_m^{(1)} \end{cases}$$

   3) Метод Обратного хода.

2. Метод Гаусса-Зейделя

Метод Гаусса – Зейделя, в свою очередь, основан на итерационном методе, вычисляющим приближения $X^{(k+1)}$, которые вычисляются по предыдущему приближению $X^{(k)}$ путём подстановки компонент $X^{(k)}$ в правую часть уравнений системы.

1) Приводим систему к виду

$$\left.\begin{aligned} x_1 &= c_{11}x_1 + c_{12}x_2 + \ldots + c_{1n}x_n + f_1, \\ x_2 &= c_{21}x_1 + c_{22}x_2 + \ldots + c_{2n}x_n + f_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ x_n &= c_{n1}x_1 + c_{n2}x_2 + \ldots + c_{nn}x_n + f_n, \end{aligned}\right\}$$

Или же в матричном вводе

$$X = CX + F,$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{pmatrix}, \quad C = \begin{pmatrix} c_{11} & c_{12} & .. & c_{1n} \\ c_{21} & c_{22} & .. & c_{2n} \\ .. & .. & .. & .. \\ c_{n1} & c_{n2} & .. & c_{nn} \end{pmatrix}, \quad F = \begin{pmatrix} f_1 \\ f_2 \\ .. \\ f_n \end{pmatrix}.$$

2) Берём за нулевое приближение произвольный вектор
3) Строи векторы

$$X^{(k+1)} = CX^{(k)} + F, \; k = 0,1,2,..$$

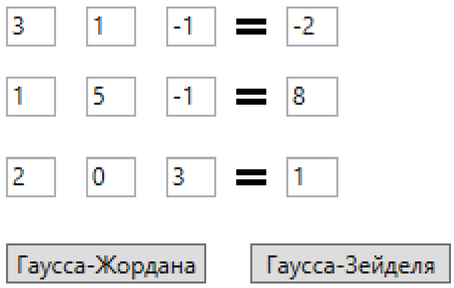
4) Проверяем условие сходимости

$$\sum_{j=1}^{n} |c_{ij}| < 1; \; i = 1,\ldots,n, \text{ или } \sum_{i=1}^{n} |c_{ij}| < 1; \; j = 1,\ldots,n, \text{ или } \sum_{i,j=1}^{n} |c_{ij}|^2 < 1.$$

Перейдём к разбору программной реализации данных методов

В качестве языка разработки приложений использовался C# в интегрируемой среде разработки Visual Studio Community 2019.

Использовался следующий интерфейс

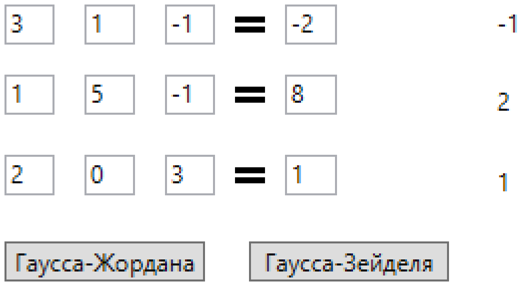


Для ввода данных используется представление текстовых полей в матричном виде, также написан алгоритм их считывания с последующей записью в массив

Листинг-1 Функция решения СЛАУ методом Гаусса-Жордана:

```
public static double[] Metod_GaussJordan(double[,] matrixA, double[] vectorB)
{
    // Копирование матрицы
    double[,] A_copy = new double[3, 3];
    for (int i = 0; i < matrixA.GetLength(0); i++)
    {
        for (int j = 0; j < matrixA.GetLength(1); j++)
        {
            A_copy[i, j] = matrixA[i, j];
        }
    }

    //Прямой ход
    for(int k = 0; k < matrixA.GetLength(0); k++)
    {
        //Нули на диагонали
        if (A_copy[k, k] == 0)
        {
            int nul = k;
            for (int i = k + 1; i< matrixA.GetLength(1); i++)
            {
                if (A_copy[i,k] != 0)
                {
                    nul = i;
                    MessageBox.Show("Диагональ");
                    break;

                }
            }
            if (nul != k)
            {
                // Выбор рабочей точки
                double[,] new_copy = new double[3, 3];
                for (int i = 0; i < matrixA.GetLength(0); i++)
                {
                    for (int j = 0; j < matrixA.GetLength(1); j++)
                    {
                        new_copy[i, j] = matrixA[i, j];
                    }
                }
                for (int j = 0; j < matrixA.GetLength(1); j++)
                {
                    new_copy[nul, j] = matrixA[k, j];
                    new_copy[k, j] = matrixA[nul, j];
                }
                double temp = vectorB[k];
                vectorB[k] = vectorB[nul];
                vectorB[nul] = temp;
                for (int i = 0; i < matrixA.GetLength(0); i++)
                {
                    for(int j = 0; j < matrixA.GetLength(1); j++)
                    {
                        A_copy[i, j] = new_copy[i, j];
                    }
                }
            }
            else
            {
```

```
                MessageBox.Show("Ошибка при выборе рабочей точки");
            }

        }

        //Приведение к теругольному ввиду
        for (int i = 0; i < matrixA.GetLength(0); i++)
        {
            matrixA[k, i] /= A_copy[k, k];
        }
        vectorB[k] /= A_copy[k, k];
        for (int i = k + 1; i < matrixA.GetLength(0); i++)
        {
            double K = matrixA[i, k] / matrixA[k, k];
            for (int j = 0; j < matrixA.GetLength(1); j++)
            {
                matrixA[i, j] -= matrixA[k, j] * K;
            }
            vectorB[i] -= vectorB[k] * K;
        }
        for (int i = 0; i < matrixA.GetLength(0); i++)
        {
            for (int j = 0; j < matrixA.GetLength(1); j++)
            {
                A_copy[i, j] = matrixA[i, j];
            }
        }
    }

    // Обратный ход
    for (int k = matrixA.GetLength(0) - 1; k > -1; k--)
    {
        for (int i = matrixA.GetLength(1) - 1; i > -1; i--)
        {
            matrixA[k, i] /= A_copy[k, k];
        }
        vectorB[k] /= A_copy[k, k];
        for (int i = k - 1; i > -1; i--)
        {
            double K = matrixA[i, k] / matrixA[k, k];
            for (int j = matrixA.GetLength(1) - 1; j > -1; j--)
            {
                matrixA[i, j] -= matrixA[k, j] * K;
            }
            vectorB[i] -= vectorB[k] * K;
        }
    }

    return vectorB;

}
```


3
1
-1
=
-2
-1
1
5
-1
=
8
2
2
0
3
=
1
1
Гаусса-Жордана
Гаусса-Зейделя

Листинг-2 функция решения СЛАУ методом Зейделя

```
public static double[] Metod_Iteration(double[,] matrixA, double[] vectorB)
{
    double[] X = new double[3];
    double[,] C = new double[3, 3];
    double[] F = new double[3];


    //Приведение к виду
    for (int i = 0; i < C.GetLength(0); i++)
    {
        for (int j = 0; j < C.GetLength(1); j++)
        {
            if (j != i)
            {
                C[i, j] = -(matrixA[i, j] / matrixA[i, i]);
            }
            else
            {
                C[i, j] = 0;
            }
        }
        F[i] = vectorB[i] / matrixA[i, i];
    }
    //Интегрирование
    int n = 0;
    while (n < 100)
    {
        for (int i = 0; i < C.GetLength(0); i++)
        {
            double x = 0;
            for (int j = 0; j < C.GetLength(1); j++)
            {
                x += C[i, j] * X[j];
            }
            x += F[i];
            X[i] = x;
        }
        n++;
    }

    //Проверка сходимости
    for (int i = 0; i < C.GetLength(0); i++)
    {
        double E = 0;
        for (int j = 0; j < C.GetLength(1); j++)
        {
            if (i == j) continue;
            E += Math.Abs(C[i, j]);
        }
        if (!(E < 1))
        {
            MessageBox.Show("Условие сходимсти");
            break;
        }

        if (matrixA[i, i] == 0)
        {
            MessageBox.Show("Недопустимые нули на диагонали");
```

```
                    break;
                }
            }
            return X;
        }
```