МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий

Дисциплина:

«Алгоритмы и структуры данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Тема: «Реализация решения систем линейных алгебраических уравнений методом Гаусса на языке С++»

Выполнила:

Студентка гр. N32511

Синюта А. А.________

Проверил:

Дата:________________

Оценка:_____________

Чернов Р. И.________

Санкт-Петербург

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Задание:

Реализация решения систем линейных алгебраических уравнений методом Гаусса на языке C++.

Цель:

Нахождение оптимального алгоритма, реализующего решение систем линейных алгебраических уравнений методом Гаусса.

Задачи:

1. Изучить теорию по решению систем линейных алгебраических уравнений методом Гаусса;
2. Разработать алгоритм с наименьшей асимптотической сложностью;
3. Построить блок-схему алгоритма;
4. Реализовать разработанный алгоритм программой на языке C++;
5. Проверить корректность работы программы.

1 ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

1.1 Теория

Алгоритм решения СЛАУ методом Гаусса подразделяется на два этапа:

1. На первом этапе осуществляется так называемый прямой ход, когда путём элементарных преобразований над строками систему приводят к ступенчатой или треугольной форме, либо устанавливают, что система несовместна. Для этого среди элементов первого столбца матрицы выбирают ненулевой, перемещают содержащую его строку в крайнее верхнее положение, делая эту строку первой. Далее ненулевые элементы первого столбца всех нижележащих строк обнуляются путём вычитания из каждой строки первой строки, домноженной на отношение первого элемента этих строк к первому элементу первой строки. После того, как указанные преобразования были совершены, первую строку и первый столбец мысленно вычёркивают и продолжают, пока не останется матрица нулевого размера. Если на какой-то из итераций среди элементов первого столбца не нашёлся ненулевой, то переходят к следующему столбцу и проделывают аналогичную операцию.
2. На втором этапе осуществляется так называемый обратный ход, суть которого заключается в том, чтобы выразить все получившиеся базисные переменные через небазисные и построить фундаментальную систему решений, либо, если все переменные являются базисными, то выразить в численном виде единственное решение системы линейных уравнений. Эта процедура начинается с последнего уравнения, из которого выражают соответствующую базисную переменную (а она там всего одна) и подставляют в предыдущие уравнения, и так далее, поднимаясь по «ступенькам» наверх. Каждой строчке соответствует ровно одна базисная переменная, поэтому на каждом шаге, кроме последнего (самого верхнего), ситуация в точности повторяет случай последней строки.

1.2 Блок-схема алгоритма

Блок-схема алгоритма представлена на рисунке 1.

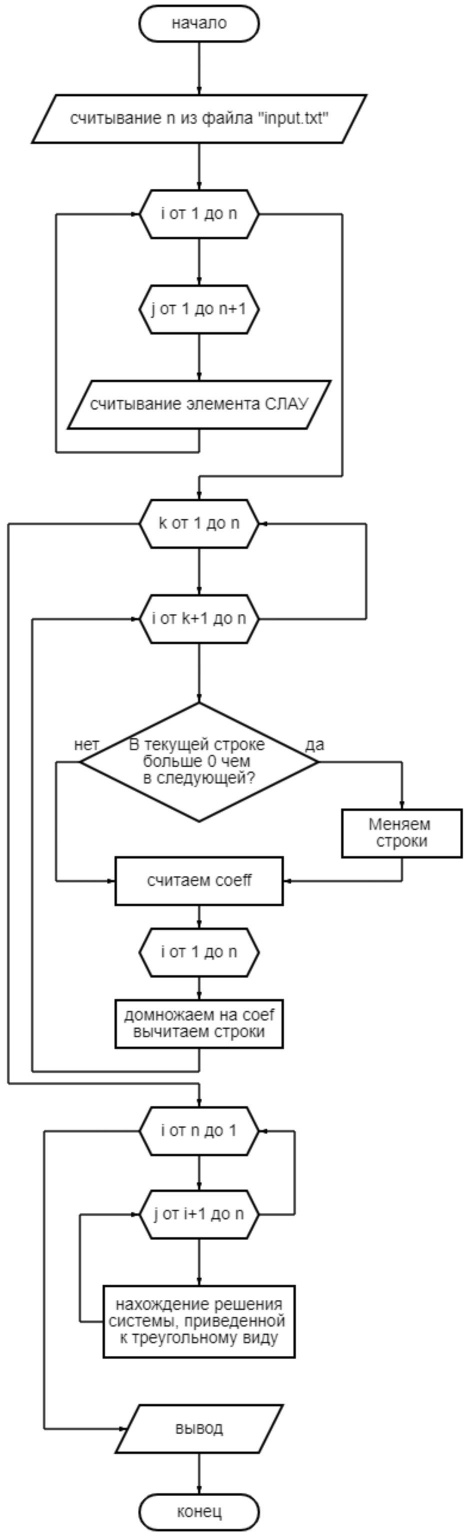

Рисунок 1 - Блок-схема алгоритма

1.3 Листинг программы

```
//1. Элементарные структуры данных. Работа с памятью. ЛР:
// Реализация решения систем линейных алгебраических уравнений методом Гаусса на языке C++;

#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <string>

using namespace std;

int main()
{
    ifstream fin("input3.txt");
    vector<vector<int>> matrix;

    string line;
    while (getline(fin, line)) {
        vector<int> row;
        istringstream iss(line);
        int value;
        while (iss >> value) {
            row.push_back(value);
        }
        matrix.push_back(row);
    }

    // Выводим матрицу
    cout << "Matrix:" << endl;
    for (const auto& row : matrix) {
        for (const auto& value : row) {
            cout << value << " ";
        }
        cout << endl;
    }
    cout << endl;

    size_t numRows = matrix.size();
    size_t numCols = matrix[0].size();

    // Проверка наличия решений
    if (numCols != numRows + 1) {
        cout << "The system has no solutions." << endl;
        return 0;
    }

    int n = numRows;
    vector<vector<double>> A(n, vector<double>(n));
    vector<double> b(n);
    vector<double> x(n);

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            A[i][j] = matrix[i][j];
        }
```

```
        b[i] = matrix[i][n];
    }

    // Приведение матрицы коэффициентов к ступенчатому виду
    for (int k = 0; k < n; k++)
    {
        for (int i = k + 1; i < n; i++)
        {
            if (A[k][k] == 0)
            {
                swap(A[k], A[k + 1]);
                swap(b[k], b[k + 1]);
                continue;
            }
            double coeff = A[i][k] / A[k][k];
            for (int j = k; j < n; j++)
            {
                A[i][j] -= coeff * A[k][j];
            }
            b[i] -= coeff * b[k];
        }
    }

    // Проверка единственного решения
    for (int i = 0; i < n; i++)
    {
        if (A[i][i] == 0)
        {
            if (b[i] != 0)
            {
                cout << "The system has no solutions." << endl;
                return 0;
            }
            cout << "The system has an infinite number of solutions." << endl;
            return 0;
        }
    }

    // Нахождение решения системы
    for (int i = n - 1; i >= 0; i--)
    {
        double sum = 0;
        for (int j = i + 1; j < n; j++)
        {
            sum += A[i][j] * x[j];
        }
        x[i] = (b[i] - sum) / A[i][i];
    }

    // Вывод решения
    cout << "System solution:";
    for (int i = 0; i < n; i++)
    {
        cout << endl << "x" << i + 1 << " = " << x[i];
    }
    cout << endl;
    return 0;
}
```

1.4 Результаты работы программы

Matrix	System solution
1 2 3 3 3 5 7 0 1 3 4 1	x1 = -4 x2 = -13 x3 = 11
75 30 35 -90 30 75 20 50 35 20 70 -60	x1 = -1.53463 x2 = 1.41206 x3 = -0.493274
1 2 3 1 4 5 6 2	The system has no solutions.
1 2 1 2 4 3	The system has no solutions.
1 2 3 1 4 5 6 2 7 8 9 3	The system has an infinite number of solutions.

Файл “input.txt” с входными данными представлен на рисунке 2.1:

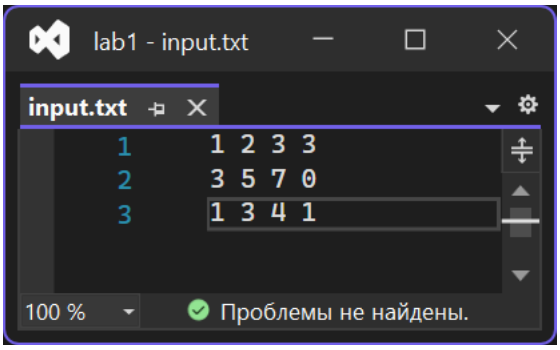

Рисунок 2.1 – Результат работы программы (Входные данные)

Вывод в консоль с выходными данными представлен на рисунке 2.2:

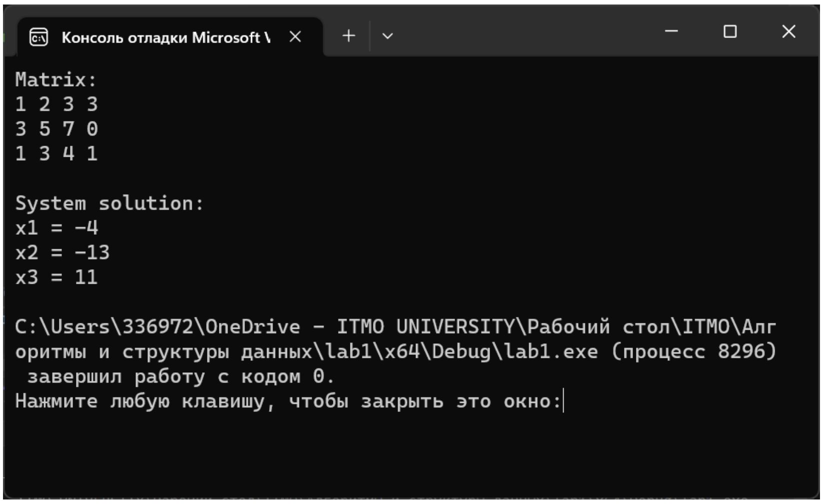

Рисунок 2.2 – Результат работы программы (Выходные данные)

1.5 Проверка результатов

Выполнение проверки с помощью сайта https://mathdf.com/equ/ru/. Результат представлен на рисунке 3.

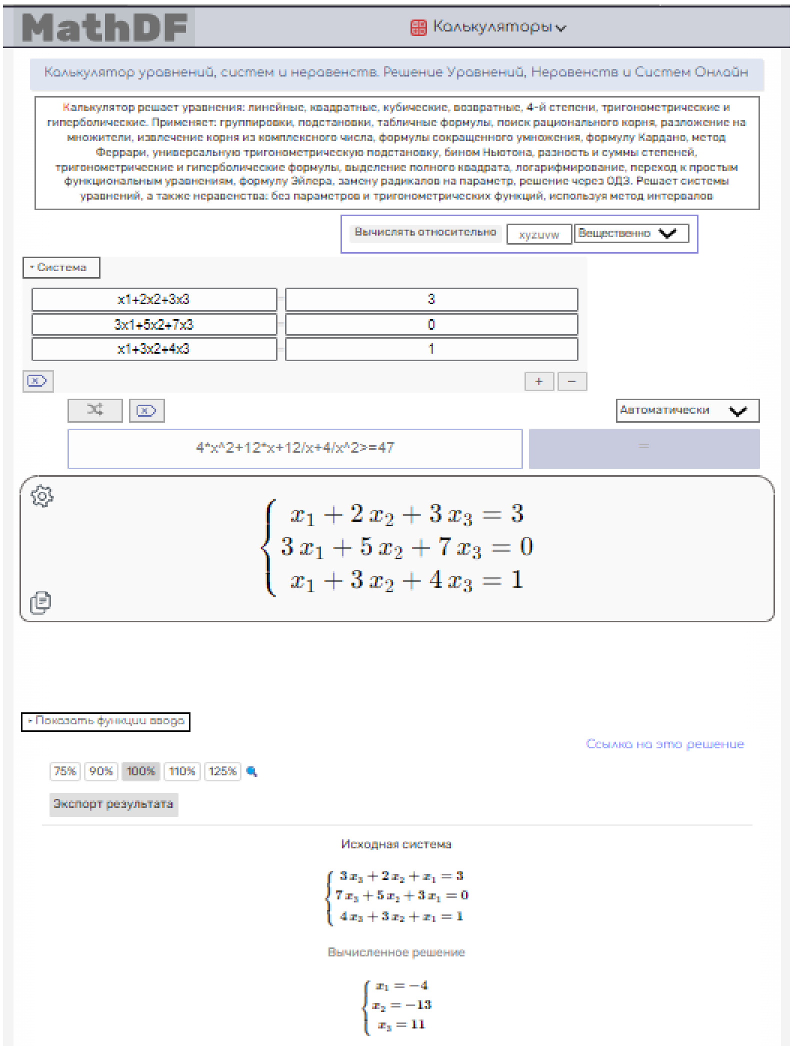

Рисунок 3 - Проверка

1.6 Оценка быстродействия алгоритма в асимптотической нотации

Асимптотическая сложность алгоритма — n^3. График асимптотической нотации представлен на рисунке 4.

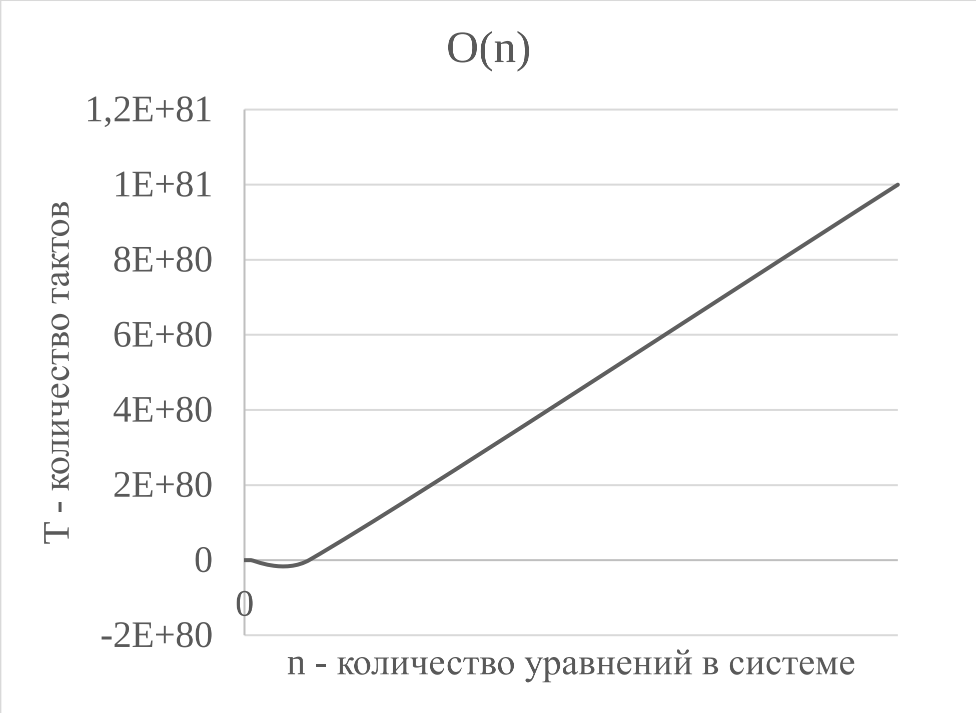

Рисунок 4 – Асимптотическая нотация

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы удалось найти оптимальный алгоритма, реализующий решение систем линейных алгебраических уравнений методом Гаусса.

Работа была выполнена успешно. На примере работы программы видно, что алгоритм работает правильно.

Данная лабораторная работа помогла лучше понять метод Гаусса, а также вспомнить основы программирования на C++ и анализ асимптотической сложности алгоритма.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ильин В. А., Позняк Э. Г. Линейная алгебра: Учебник для вузов. — 6-е изд., стер. — М.: ФИЗМАТЛИТ, 2004. — 280 с.
2. Mathdf.com Пошаговые калькуляторы MathDF: [Электронный ресурс]. URL: https://mathdf.com/ru/