

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
Инженерно-физический факультет  
Кафедра автоматизированных систем обработки информации и управления

Отчет по практике

Программная реализация численного метода  
*Решение системы линейных алгебраических  
уравнений методом Гаусса-Жордана.*

1 курс, группа УТС1

Выполнил:  
\_\_\_\_\_ А.А. Сокур  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.  
Руководитель:  
\_\_\_\_\_ С.В. Теплоухов  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

Майкоп, 2025 г.

# 1. Введение

## 1.1 Цель работы

Цель работы — реализовать программу, решающую систему линейных алгебраических уравнений методом Гаусса–Жордана. Метод заключается в полном исключении неизвестных, приводя матрицу к единичному виду.

## 2 Ход работы

### 2.1 Пример кода, решающего данную задачу

```
1  #include <iostream>
2  #include <vector>
3  #include <iomanip>
4  #include <cmath>
5  #include <limits>
6  #include <algorithm>      //          std::max
7  #define NOMINMAX          // <—      !
8  #include <windows.h>      //          SetConsoleCP / SetConsoleOutputCP
9
10 using namespace std;
11
12 //
13 bool gaussJordan(vector<vector<double>> matrix, vector<double>& solution) {
14     size_t n = matrix.size(); // <— size_t          int
15
16     for (size_t i = 0; i < n; ++i) {
17         size_t maxRow = i;
18         for (size_t k = i + 1; k < n; ++k)
19             if (std::fabs(matrix[k][i]) > std::fabs(matrix[maxRow][i]))
20                 maxRow = k;
21
22         if (std::fabs(matrix[maxRow][i]) < 1e-9)
23             return false;
24
25         swap(matrix[i], matrix[maxRow]);
26
27         double pivot = matrix[i][i];
28         for (size_t j = i; j <= n; ++j)
29             matrix[i][j] /= pivot;
30
31         for (size_t k = 0; k < n; ++k) {
32             if (k != i) {
33                 double factor = matrix[k][i];
34                 for (size_t j = i; j <= n; ++j)
35                     matrix[k][j] -= factor * matrix[i][j];
36             }
37         }
38     }
39
40     solution.resize(n);
41     for (size_t i = 0; i < n; ++i)
42         solution[i] = matrix[i][n];
43
44     return true;
45 }
46
```

```

47 int main() {
48     //
49     SetConsoleCP(1251);
50     SetConsoleOutputCP(1251);
51     setlocale(LC_ALL, "Russian");
52
53     int n;
54     cout.precision(6);
55     cout << "                                (n): ";
56     while (!(cin >> n) || n <= 0) {
57         cin.clear();
58         cin.ignore(numeric_limits<streamsize>::max(), '\n');
59         cout << "                                :
                                     n: ";
60     }
61
62     vector<vector<double>> matrix(n, vector<double>(n + 1));
63     cout << "                                (" << n << "
                                     " << n + 1 << "                                ):\n";
64
65     for (int i = 0; i < n; ++i) {
66         for (int j = 0; j <= n; ++j) {
67             while (true) {
68                 cout << "                                [" << i << "][" << j << "] = ";
69                 double val;
70                 if (cin >> val) {
71                     matrix[i][j] = val;
72                     break;
73                 }
74                 else {
75                     cin.clear();
76                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
77                     cout << "                                .
                                     .\n";
78                 }
79             }
80         }
81     }
82
83     vector<double> solution;
84
85     if (gaussJordan(matrix, solution)) {
86         cout << "\n                                :\n";
87         for (size_t i = 0; i < solution.size(); ++i)
88             cout << "x[" << i << "] = " << solution[i] << endl;
89     }
90     else {
91         cout << "\n
                                     .\n";
92     }
93
94     return 0;
95 }

```

## 2.2 Формулы и описание метода

Метод Гаусса–Жордана представляет собой модификацию классического метода Гаусса. Он заключается в том, чтобы привести расширенную матрицу к *единичной форме*, после чего решение находится в последнем столбце.

Для каждого шага  $i$  выполняется следующее:

1. Выбор ведущего элемента.
2. Перестановка строк, если нужно.
3. Нормализация строки по ведущему элементу.
4. Обнуление всех других элементов в текущем столбце.

### 3 Интерфейс программы

Реализация программы: решение системы линейных алгебраических уравнений методом Гаусса-Жордана.

```
Введите размерность системы (n): 3
Введите расширенную матрицу (3 строк по 4 чисел):
Матрица[0][0] = 1
Матрица[0][1] = 2
Матрица[0][2] = 3
Матрица[0][3] = 3
Матрица[1][0] = 2
Матрица[1][1] = 1
Матрица[1][2] = 4
Матрица[1][3] = 5
Матрица[2][0] = 6
Матрица[2][1] = 4
Матрица[2][2] = 3
Матрица[2][3] = 2

Решение системы:
x[0] = 0.0344828
x[1] = -0.586207
x[2] = 1.37931
```

Рис. 1: Скриншот программы в консоли

На рисунке 1 представлено окно командной строки с результатом выполнения программы.

### Заключение

В ходе выполнения лабораторной работы была реализована программа, решающая систему линейных алгебраических уравнений методом Гаусса-Жордана. Реализован контроль вводимых данных, а также оформлен отчёт.

## Список литературы

- [1] Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — Москва: Лаборатория Базовых Знаний, 2002 г.
- [2] Лафанов В.А. Программирование на языке C++. — Санкт-Петербург: Питер, 2018 г.