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

Отчёт по практике

Решение системы линейных алгебраических уравнений методом Гаусса

2 курс, группа 2ИВТ

Выполнил:
___________________ Д. Э. Чич
«____» _____________ 2025 г.

Руководитель:
___________________ С. В. Теплоухов
«____» _____________ 2025 г.

Майкоп, 2025 г.

Оглавление

1 Теория **2**

2 Код программы **3**

3 Пример работы программы **6**

1 Теория

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений (СЛАУ). Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида, из которой последовательно, начиная с последних (по номеру), находятся все переменные системы.

Алгоритм решения СЛАУ методом Гаусса подразделяется на два этапа.

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. А именно, среди элементов первого столбца матрицы выбирают ненулевой, перемещают его на крайнее верхнее положение перестановкой строк и вычитают получившуюся после перестановки первую строку из остальных строк, домножив её на величину, равную отношению первого элемента каждой из этих строк к первому элементу первой строки, обнуляя тем самым столбец под ним. 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.

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.

В простейшем случае алгоритм выглядит так:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1n} \cdot x_n & = b_1 \quad (1) \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \ldots + a_{2n} \cdot x_n & = b_2 \quad (2) \\ \ldots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \ldots + a_{mn} \cdot x_n & = b_m \quad (m) \end{cases}$$

Рис. 1: Прямой ход метода Гаусса

Прямой ход:

$$
\begin{array}{rcl}
(2) \to (2) - (1) \cdot (\frac{a_{21}}{a_{11}}) & : & a'_{22} \cdot x_2 + a'_{23} \cdot x_3 + \ldots + a'_{2n} \cdot x_n = b'_2 \\
(3) \to (3) - (1) \cdot (\frac{a_{31}}{a_{11}}) & : & a'_{32} \cdot x_2 + a'_{33} \cdot x_3 + \ldots + a'_{3n} \cdot x_n = b'_3 \\
\ldots & & \\
(m) \to (m) - (1) \cdot (\frac{a_{m1}}{a_{11}}) & : & a'_{m2} \cdot x_2 + a'_{m3} \cdot x_3 + \ldots + a'_{mn} \cdot x_n = b'_n \\
(3) \to (3) - (2) \cdot (\frac{a'_{32}}{a'_{22}}) & : & a''_{33} \cdot x_3 + \ldots + a''_{3n} \cdot x_n = b''_3 \\
\ldots & & \\
(m) \to (m) - (m-1) \cdot (\frac{a^{(m-2)}_{m,n-1}}{a^{(m-2)}_{m-1,n-1}}) & : & a^{(m-1)}_{mm} \cdot x_m + \ldots + a^{(m-1)}_{mn} \cdot x_n = b^{(m-1)}_m
\end{array}
$$

Рис. 2: Детализация прямого хода

Обратный ход. Из последнего ненулевого уравнения выражаем базисную переменную через небазисные и подставляем в предыдущие уравнения. Повторяя эту процедуру для всех базисных переменных, получаем фундаментальное решение.

2 Код программы

```
#include <iostream>
#include <cmath>
using namespace std;

const double EPS = 1e-9;
const int MAX_SIZE = 10;

void printSystem(double matrix[MAX_SIZE][MAX_SIZE + 1], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << "| " << matrix[i][n] << endl;
    }
    cout << endl;
}

int main() {
    setlocale(0, "ru");
    int n;
    double matrix[MAX_SIZE][MAX_SIZE + 1] = { 0 };

    while (true) {
        cout << "Введите количество уравнений (1-" << MAX_SIZE << "): ";
        cin >> n;
        if (cin.good() && n > 0 && n <= MAX_SIZE) break;
        cout << "Ошибка! Введите число от 1 до " << MAX_SIZE << endl;
        cin.clear();
        cin.ignore(10000, '\n');
    }

    cout << "Введите коэффициенты системы (по строкам):\n";
```

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j <= n; j++) {
        while (true) {
            cin >> matrix[i][j];
            if (cin.good()) break;
            cout << "Ошибка! Введите число: ";
            cin.clear();
            cin.ignore(10000, '\n');
        }
    }
}

cout << "\nИсходная система:\n";
printSystem(matrix, n);

for (int col = 0; col < n; col++) {
    int max_row = col;
    for (int i = col + 1; i < n; i++) {
        if (abs(matrix[i][col]) > abs(matrix[max_row][col])) {
            max_row = i;
        }
    }

    if (max_row != col) {
        swap(matrix[col], matrix[max_row]);
        cout << "Меняем строки " << col + 1 << " и " << max_row + 1 << ":\n";
        printSystem(matrix, n);
    }

    if (abs(matrix[col][col]) < EPS) {
        cout << "Система вырождена!\n";
    }

    for (int j = n; j >= col; j--) {
        matrix[col][j] /= matrix[col][col];
    }

    cout << "Нормализуем строку " << col + 1 << ":\n";
    printSystem(matrix, n);

    for (int i = col + 1; i < n; i++) {
        double factor = matrix[i][col];
        for (int j = col; j <= n; j++) {
            matrix[i][j] -= factor * matrix[col][j];
        }
        cout << "Обнуляем строку " << i + 1 << ":\n";
        printSystem(matrix, n);
    }
}
```

```
    for (int i = 0; i < n; i++) {
        bool all_zero = true;
        for (int j = 0; j < n; j++) {
            if (abs(matrix[i][j]) > EPS) {
                all_zero = false;
                break;
            }
        }
        if (all_zero && abs(matrix[i][n]) > EPS) {
            cout << "Нет решений!\n";
        }
    }

    double solution[MAX_SIZE] = { 0 };
    for (int i = n - 1; i >= 0; i--) {
        solution[i] = matrix[i][n];
        for (int j = i + 1; j < n; j++) {
            solution[i] -= matrix[i][j] * solution[j];
        }
    }

    cout << "Решение системы:\n";
    for (int i = 0; i < n; i++) {
        cout << "x" << i + 1 << " = " << solution[i] << endl;
    }
}
```

3 Пример работы программы

```
Введите количество уравнений (1-10): 3
Введите коэффициенты системы (по строкам):
1 4 2 5
4 3 2 1
3 2 1 4

Исходная система:
1 4 2 | 5
4 3 2 | 1
3 2 1 | 4

Меняем строки 1 и 2:
4 3 2 | 1
1 4 2 | 5
3 2 1 | 4

Нормализуем строку 1:
1 0.75 0.5 | 0.25
1 4 2 | 5
3 2 1 | 4

Обнуляем строку 2:
1 0.75 0.5 | 0.25
0 3.25 1.5 | 4.75
3 2 1 | 4

Обнуляем строку 3:
1 0.75 0.5 | 0.25
0 3.25 1.5 | 4.75
0 -0.25 -0.5 | 3.25

Нормализуем строку 2:
1 0.75 0.5 | 0.25
0 1 0.461538 | 1.46154
0 -0.25 -0.5 | 3.25

Обнуляем строку 3:
1 0.75 0.5 | 0.25
0 1 0.461538 | 1.46154
0 0 -0.384615 | 3.61538

Нормализуем строку 3:
1 0.75 0.5 | 0.25
0 1 0.461538 | 1.46154
0 0 1 | -9.4

Решение системы:
x1 = 0.6
x2 = 5.8
x3 = -9.4
```

Рис. 3: Пример работы программы