

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

Факультет Программной Инженерии и Компьютерной Техники
Дисциплина: «Вычислительная математика»

ОТЧЁТ

по лабораторной работе №1
Вариант №2

Выполнил

Студент группы Р3208

Грищенко Андрей Викторович

Поток 1.3

Преподаватель

Машина Екатерина Алексеевна

Санкт-Петербург

2025 г.

Цель

Изучить прямые и итерационные методы решения систем линейных алгебраических уравнений и реализовать один из них с помощью программирования

Описание метода

В методе Гаусса выделяют прямой и обратный ход для решения СЛАУ. Прямой ход заключается в приведении расширенной матрицы к треугольному виду, то есть в последовательном исключении неизвестных из уравнений, оставляя нули ниже диагонали. Затем, в обратном ходе это позволяет постепенно найти все неизвестные из системы уравнений.

Листинг программы

```
std::pair<double, double**> MatrixSolver::calc_triangular(int n, double** augmented_matrix) {
    double determinant_sign = 1.0;

    double** triangular = new double*[n];
    for (int i = 0; i < n; i++) {
        triangular[i] = new double[n+1];
        for (int j = 0; j <= n; j++) triangular[i][j] = augmented_matrix[i][j];
    }

    for (int j = 0; j < n; ++j) {
        int pivot_row = j;
        for (int i = j + 1; i < n; ++i) {
            if (fabs(triangular[i][j]) > fabs(triangular[pivot_row][j])) {
                pivot_row = i;
            }
        }

        if (triangular[pivot_row][j] == 0) {
            std::cout << "Матрица вырожденная, решение не может быть найдено однозначно." << std::endl;
            exit(1);
        }

        if (pivot_row != j) {
            std::swap(triangular[j], triangular[pivot_row]);
            determinant_sign *= -1.0;
        }

        for (int i = j + 1; i < n; ++i) {
            double factor = triangular[i][j] / triangular[j][j];
            for (int k = j; k <= n; ++k) {
                triangular[i][k] -= factor * triangular[j][k];
            }
        }
    }
    return std::pair<double, double**>(determinant_sign, triangular);
}

double* MatrixSolver::solve(int n, double** augmented_matrix) {
    auto triangular = calc_triangular(n, augmented_matrix).second;

    double* solution = new double[n];
    for (int i = n - 1; i >= 0; --i) {
        double sum = 0.0;
        for (int j = i + 1; j < n; ++j) {
            sum += triangular[i][j] * solution[j];
        }
        solution[i] = (triangular[i][n] - sum) / triangular[i][i];
    }
    return solution;
}
```

[Полный код](#)

Пример работы программы

```
Решение СЛАУ методом Гаусса
Выберите режим ввода матрицы:
1 - ввод из консоли
2 - ввод из файла
2
Выбран ввод из файла
Используется файл ввода "input.txt"
Введите n: 3
Введите построчно расширенную матрицу
2 1 1 9
1 3 2 13
1 2 4 16
```

```
СЛАУ:
2.0x1 + 1.0x2 + 1.0x3 = 9.0
1.0x1 + 3.0x2 + 2.0x3 = 13.0
1.0x1 + 2.0x2 + 4.0x3 = 16.0
```

```
Треугольная матрица:
2.00    1.00    1.00 | 9.00
0.00    2.50    1.50 | 8.50
0.00    0.00    2.60 | 6.40
```

```
Определитель: 13.00
Решение системы методом Гаусса
x1 = 2.31
x2 = 1.92
x3 = 2.46
```

```
Определитель через Eigen: 13.00
```

```
Решение системы через Eigen
x1 = 2.31
x2 = 1.92
x3 = 2.46
```

```
Невязка:
r1 = -0.00
r2 = 0.00
r3 = -0.00
```

Вывод

В результате проделанной лабораторной работы я ознакомился с прямыми численными и итерационными методами решения систем линейных алгебраических уравнений и реализовал метод Гаусса на языке C++. Также я проверил корректность его работы используя невязки путем сравнения с результатами полученными с помощью библиотеки Eigen.