Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Решение СЛУ методом Гаусса»**

**Выполнил**:

студент группы 3824Б1ПМ1-2

Теселкин К.И.

**Проверила**:

Бусько П.В.

Нижний Новгород

2024

**Содержание**

[Введение 3](#_Toc182419046)

[Постановка задачи 4](#_Toc182419047)

[Руководство пользователя 5](#_Toc182419048)

[Описание программной реализации 6](#_Toc182419049)

[Результаты экспериментов 7](#_Toc182419050)

[Заключение 8](#_Toc182419051)

[Литература 9](#_Toc182419052)

[Приложение 10](#_Toc182419053)

# Введение

Решение систем линейных алгебраических уравнений — это фундаментальная задача в линейной алгебре, имеющая широкое применение в различных научных и технических областях. Метод Гаусса с выбором главного элемента является эффективным подходом для решения таких систем. В рамках данной работы реализован метод Гаусса с выбором ведущего элемента для работы с действительными квадратными матрицами. Программа разработана с использованием объектно-ориентированного подхода, включая шаблонные классы для вектора и матрицы.

# Постановка задачи

Основной целью работы является реализация метода Гаусса для решения СЛУ с применением объектно-ориентированного подхода и шаблонов. Для этого необходимо выполнить следующие условия:

Разработать шаблонный класс Vector<T>.

Создать класс Matrix<T>, который будет шаблоном Vector<Vector<T>> и представлять квадратную матрицу.

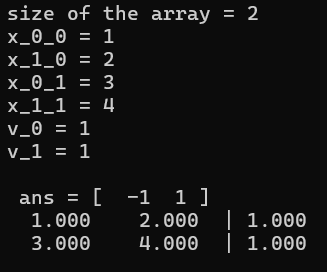
Реализовать класс CLY<T>, наследующийся от Matrix<T>, с функцией Gaussian\_method, которая реализует метод Гаусса.

Метод Gaussian\_method должен принимать вектор правой части и возвращать вектор решений.

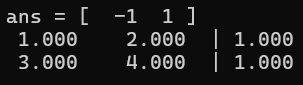
# Руководство пользователя

Запуск программы:

Для работы программы пользователь должен ввести матрицу коэффициентов и вектор правой части в консоль.



*Рис. 1. Пример входных данных*



*Рис. 2. Пример вывода решений*

# Описание программной реализации

Программа разработана на языке C++ с использованием шаблонов и классов. Основные компоненты реализации:

Класс Vector<T> : Шаблонный вектор фиксированной длины, поддерживающий инициализация, копирование и доступ по индексу.

Класс Matrix<T> : Представляет квадратную матрицу, реализованную как вектор векторов (Vector<Vector<T>>).

Класс СLY<T> : Наследник Matrix<T>, реализует метод Гаусса с выбором главного элемента.

Функция Gaussian\_method(Vector<T> v) : Возвращает вектор решений.

# Результаты экспериментов

Тестовая система уравнений:

Полученные результаты: , ,

Также были проведены испытания на других системах уравнений с заранее известными решениями. Метод успешно находил решения, что подтверждает корректность его реализации.

# Заключение

В рамках данной лабораторной работы была успешно реализована программа, использующая метод Гаусса для решения систем линейных уравнений с квадратной матрицей. Применение шаблонов и объектно-ориентированного подхода обеспечило универсальность, гибкость и масштабируемость программы. Проведенные тесты подтвердили корректность реализации.

# Литература

1. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования СИ //М.: Финансы и статистика. – 1992.
2. Кнут Д. Э. Искусство программирования: Сортировка и поиск. – Издательский дом Вильямс, 2000. – Т. 3.

# Приложение

void eliminate\_row(int Y, int X, vector<double>\*& arr, double\* V) {

double k;

for (int i = Y + 1; i < this->len; i++) {

if (\*arr[Y][X] != 0) {

k = (double)(\*arr[i][X]) / \*arr[Y][X];

for (int j = 0; j < this->len; j++) {

\*arr[i][j] -= \*arr[Y][j] \* k;

}

V[i] -= V[Y] \* k;

}

}

}

vector<double> Gaussian\_method(vector<T>& v\_) {

double\* w = new double[this->len];

vector<double> ans(this->len, w);

double\* V = new double[this->len];

for (int i = 0; i < this->len; i++) { V[i] = \*v\_[i]; }

vector<double>\* arr = new vector<double>[this->len];

for (int i = 0; i < this->len; i++) {

double\* q = new double[this->len];

for (int j = 0; j < this->len; j++) { q[j] = \*this->arr[i][j]; }

arr[i](this->len, q);

}

for (int i = 0; i < this->len; i++) {

for (int j = 0; j < this->len; j++) {

if (\*arr[i][j] != 0) {

eliminate\_row(i, j, arr, V);

break;

}

}

}

for (int i = this->len - 1; i >= 0; i--) {

for (int j = 0; j < this->len; j++) {

if (\*arr[i][j] != 0) {

ans.arr[j] = (V[i]) / \*(arr[i])[j];

for (int h = i; h >= 0; h--) {

V[h] -= (\*arr[h][j]) \* ans.arr[j];

\*arr[h][j] = 0;

}

break;

}

}

}

delete[] arr;

return ans;

}