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**

**«Метод Гаусса»**

**Выполнил**:

Студент группы 3821Б1ПМ2

Логинов С. С.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

**Содержание**

[Постановка задачи 3](#_Toc105364384)

[Метод решения 4](#_Toc105364385)

[Руководство пользователя 5](#_Toc105364386)

[Описание программной реализации 6](#_Toc105364387)

[Подтверждение корректности 7](#_Toc105364388)

[Заключение 8](#_Toc105364389)

# Постановка задачи

Целью лабораторной работы являлось реализовать на языке программирования С++ метод Гаусса с выбором ведущего элемента, с использованием собственного шаблонного класса вектор и унаследованного от него собственного шаблонного класса квадратная матрица, описать алгоритм и программную реализацию, а также проверить корректность вычислений.

# Метод решения

Основным алгоритмом в программе является метод Гаусса с выбором ведущего элемента – метод решения систем линейных алгебраических уравнений (СЛАУ).

Метод состоит из двух этапов:

* Последовательное исключение;
* Обратная подстановка.

**Последовательное исключение:**

Исключения Гаусса основаны на последовательном исключении переменных по одной до тех пор, пока не останется только одно уравнение с одной переменной в левой части. После это уравнение решается относительно единственной переменной. Таким образом, систему уравнений приводят к треугольной форме. Для этого среди элементов первого столбца матрицы выбирают максимальный по модулю элемент и перемещают его на крайнее верхнее положение перестановкой строк. Затем нормируют все уравнения, разделив их на коэффициент ai1, где i - номер столбца.

**Обратная подстановка:**  
Обратная подстановка предполагает подстановку полученного на предыдущем шаге значения переменной xn в предыдущие уравнения и повторяется для всех оставшихся решений.

# Руководство пользователя

В начале работы программы пользователь видит строку “Введите размер матрицы”, после чего он должен ввести размер матрицы.

После этого программа предлагает выбрать способ заполнения матрицы. Если пользователь хочет ввести рандомные (случайные) данные, то он должен ввести 1. Если пользователь хочет заполнить матрицу сам, то он должен ввести 2. Если пользователь введет отличные от данных цифры, то программа выведет “Неверный номер способа заполнения” и завершит работу.

Если пользователь ввел корректные данные, то программа может вывести следующие варианты ответов:

1. Если программа обнаружила на главной диагонали нули или  
   по окончании решения СЛУ программа обнаружит противоречивую или нулевую строчку, то программа выведет фразу "ERROR";
2. Если программа не обнаружила на главной диагонали нули или по окончании решения СЛУ программа не обнаружит противоречивую или нулевую строчку, то программа выведет вектор значений, а также проверит решение на корректность, выведя фразу «Решение верное» в случае корректного решения системы и «Решение неверное» в противном случае.

# Описание программной реализации

Шаблонный класс MyVector (template <typename T> class MyVector) имеет следующие поля: int size – размер вектора, T\* arr– массив элементов T вектора, имеющий длину size.

В классе MyVector реализованы следующие методы:

Vector(int size = 5) – конструктор класса, выделяет память для массива arr и устанавливает размер size.

~MyVector () – деструктор класса, освобождает память, выделенную массиву arr.

void resize(int size) – изменяет размер вектора, копируя элементы (перевыделяет память для массива arr и устанавливает значение size).

T& operator [] (int i)– возвращает ссылку на arr[i].

MyVector<T>& operator = (const MyVector<T>& other) перегрузка оператора =

Шаблонный класс Matrix который наследуется от MyVector <MyVector<T»template<typename T>

class MyMatrix: public MyVector<MyVector<T».

MyMatrix (int size = 5): MyVector<MyVector<T»(size) – создаёт матрицу

MyVector<T>& operator [] (int i) - перегрузка оператора [], возвращает ссылку на this->arr[i].

Шаблонный класс slau template <typename T>

class slau - отвечает за решение системы линейных уравнений методом Гаусса и проверку void check(MyMatrix<T>& A, MyVector<T>& X\_arg, MyVector<T>& B)

# Подтверждение корректности

Функция check (метод класса slau) использовалась для подтверждения корректности программной. Данная функция вычисляет погрешность вычислений и сравнивает её с заданным максимальным

Значением погрешности, равным 10-10. Если погрешность хотя бы одного полученного значения переменной в системе больше, чем максимальная погрешность, то решение считается неверным и программа выводит фразу «Решение неверное». Если же погрешность ни разу не превысила максимального значения, то программа выведет фразу «Решение верное».

Проверка решения систем линейных уравнений (СЛУ) при достаточно больших значениях размерности подтвердила корректность программной реализации.

# Заключение

В ходе лабораторной работы были реализованы классы вектор, матрица и система линейных уравнений(СЛУ), содержащая реализацию метода Гаусса на языке программирования С++, а также описан алгоритм работы метода Гаусса.