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

**Отчёт по лабораторной работе**

**«Метод Гаусса»**

**Выполнил**:

студент группы 3821Б1ПМ2

Борисов С.А.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

**Оглавление**

**[Постановка задачи 3](#_Toc30018)**

**[Методы решения 4](#_Toc28017)**

**[Руководство пользователя 6](#_Toc30145)**

**[Описание программной реализации 8](#_Toc6569)**

**[Подтверждение корректности 9](#_Toc25952)**

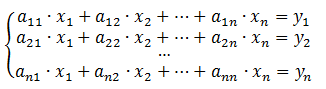
**[Заключение 1](#_Toc12541)0**

# Постановка задачи

Цель работы: 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.

# Методы решения

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



Изображение 1: система линейных алгебраических уравнений

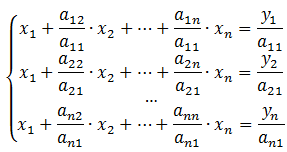
Данный метод состоит из двух этапов:

• Последовательное исключение

• Обратная подстановка

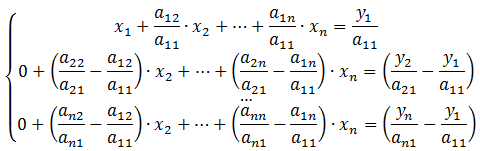
**Последовательное исключение:**

Исключения Гаусса основаны на идее последовательного исключения переменных по одной до тех пор, пока не останется только одно уравнение с одной переменной в левой части. Затем 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.



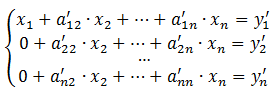
**Изображение 2: нормирование уравнений**

Затем вычитают получившуюся после перестановки первую строку из остальных строк:



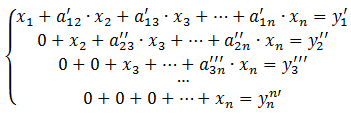
**Изображение 3: вычитание первой строки из остальных строк**

Получают новую систему уравнений, в которой заменены соответствующие коэффициенты



Изображение 4: новая система уравнений

После того, как указанные преобразования были совершены, первую строку и первый столбец мысленно вычёркивают и продолжают указанный процесс для всех последующих уравнений, пока не останется уравнение с одной неизвестной:



Изображение 5: верхний треугольный вид матрицы

**Обратная подстановка:**

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

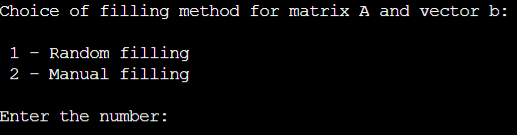
# Руководство пользователя

В начале программа предлагает пользователю ввести размерность квадратной матрицы:



Изображение 6: задание пользователем размерности квадратной матрицы

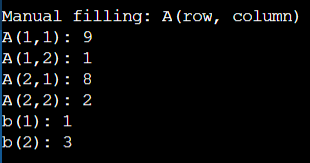
Далее программа предлагает пользователю выбрать метод заполнения квадратной матрицы А и вектора b для расширенной матрицы Ax = b



Изображение 7: выбор метода заполнения

Если пользователь ввёл 1, программа начнёт заполнение расширенной матрицы случайными элементами таким образом, чтобы диагональные элементы квадратной матрицы А были больше суммы всех других элементов этой матрицы.

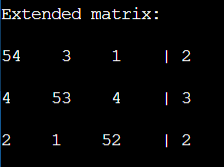
Если пользователь ввёл 2, то пользователь получит возможность ручного заполнения расширенной матрицы:



Изображение 8: пример ручного заполнения расширенной матрицы

Если же пользователь введёт значение, не удовлетворяющее описанным выше условиям, то программа выведет фразу «Invalid number of filling method» и сразу завершит свою работу.

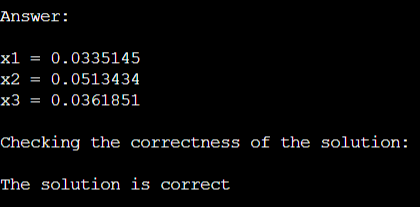
Если пользователь ввёл корректные данные, программа выводит случайно сгенерированную или заданную пользователем расширенную матрицу:



Изображение 9: пример вывода расширенной матрицы

Далее возможны несколько случаев:

1. Если программа обнаружила на главной диагонали нули, то выведется фраза «The system has no solution or has more than one solutions»
2. Если по окончанию решения системы линейных уравнений программа обнаружит противоречивую или нулевую строчку, то программа выведет соответственно фразу «No solutions» или «Infinitely many solutions»
3. Если же программа не обнаружила нулей на главной диагонали, а также нулевых и противоречивых строк, то программа выведет вектор значений, а также проверит решение на корректность, выведя фразу «The solution is correct» в случае корректного решения системы и «The solution is incorrect» в противном случае



Изображение 10: пример вывода решения и информации о его корректности

# Описание программной реализации

Программа содержит 3 шаблонных класса:

**Vector:**

При создании объекта данного класса для него создаётся динамический массив. Содержит в себе методы resize (изменяет размера вектора) и random\_filling (заполняет вектор случайными значениями). Также в данном классе перегружены операторы «=» и «[]»

**Matrix:**

Является наследником класса Vector. Содержит перегруженный оператор «[]»

**SLAU:**

Является наследником класса Matrix. Содержит 2 приватных метода create\_tmp\_copy (создаёт копию квадратной матрицы для обратного хода) и check (проверяет решение на корректность), а также один публичный метод solve (сама реализация метода Гаусса)

# Подтверждение корректности

Для подтверждения корректности программной реализации использовалась функция check (метод класса SLAU). 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 «The solution is incorrect». Если же погрешность ни разу не превысила максимального значения, то программа выведет фразу «The solution is correct».

**Проверка решения систем линейных уравнений при достаточно больших значениях размерности подтвердила корректность программной реализации.**

# Заключение

В ходе данной лабораторной работы был реализован на языке программирования C++ метод Гаусса с выбором ведущего элемента с использованием шаблонного класса вектор и унаследованного от него собственного шаблонного класса квадратная матрица, был описан алгоритм и программная реализация, а также была проведена проверка корректности вычислений.