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 3821Б1-ПМ2

Голубев Антон Михайлович

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

**Оглавление**

**Постановка задачи**

**Методы решения**

**Руководство пользователя**

**Описание программной реализации**

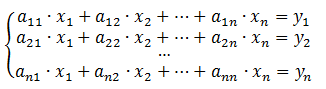
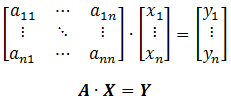
**Подтверждение корректности**

**Заключение**

# **Постановка задачи**

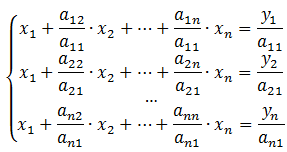
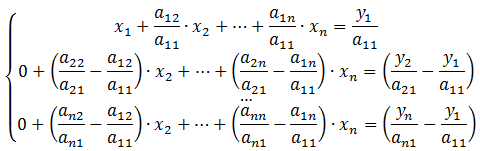
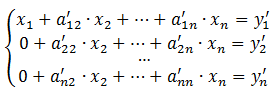
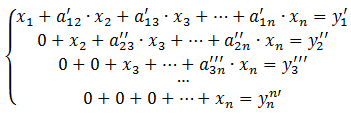
Используя язык программирования С++ реализовать метод Гаусса с выбором ведущего элемента с использованием собственного шаблонного класса вектор и унаследованного от него собственного шаблонного класса квадратная матрица, описать алгоритм и программную реализацию, а также проверить корректность вычислений.

# **Методы решения**

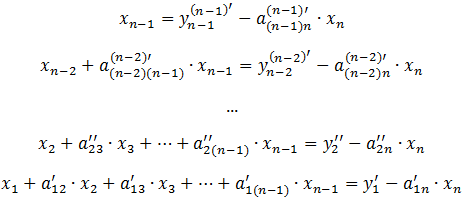
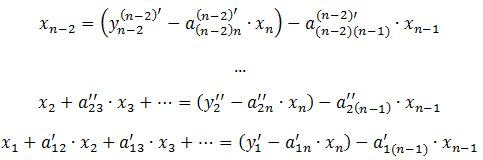
***Метод Гаусса*** — классический метод решения системы линейных алгебраических уравнений (СЛАУ).  
  
или в матричной форме  
  
Метод Гаусса решения системы линейных уравнений включает в себя 2 этапа:

* последовательное (прямое) 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 (ступенчатой) форме. Для этого среди элементов первого столбца матрицы выбирают ненулевой (а чаще максимальный) элемент и перемещают его на крайнее верхнее положение перестановкой строк. Затем нормируют все уравнения, разделив его на коэффициент **ai1**, где **i**– номер столбца.  
  
Затем вычитают получившуюся после перестановки первую строку из остальных строк:  
  
Получают новую систему уравнений, в которой заменены соответствующие коэффициенты.  
  
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:  


## **Обратная подстановка**

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


**Руководство пользователя**

В начале программа запрашивает размер квадратной матрицы, после нужно будет выбрать каким способом будет заполнена матрица (рандомно или вручную). Если вручную, то нужно будет заполнить каждый элемент матрицы. Потом нам выведет саму матрицу которая после будет решаться. Возможно несколько случаев:

* Если на главной диагонали будут нули, то программа выведет «Система не имеет решения или имеет более одного решения»
* Если по окончанию решения системы линейных уравнений программа обнаружит противоречивую или нулевую строчку, то программа выведет «Нет решений» или «Бесконечно много решений»
* Если же программа не обнаружила проблем, то выведет вектор значений, а также проверит решение на корректность. В случае некорректности решения выведет «Решение неверно»

# **Описание программной реализации**

Проект содержит классы: myVector<T>, myMatrix<T>(наследуется от класса myVector<T>), SLAU<T>(наследуется от myMatrix<T>). Определён макрос EPSILON – максимальная погрешность. В myVector перегружены операции [] и =, реализован метод resize для изменения размера вектора и arr\_random для заполнения вектора случайными элементами. В myMatrix Содержит перегруженный оператор []. В SLAU приватные методы create\_copy создаёт копию квадратной матрицы для обратного хода и check который проверяет решение на корректность, а также один публичный метод solve где сама реализация метода Гаусса

# **Подтверждение корректности**

Для подтверждения корректности в программе используется метод check (класс SLAU). В нём вычисляется погрешность и сравнивает его с EPSILON, если разность по модулю превышает фиксируемое значение, то в консоль выводится сообщение, что решение неверно.

# **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.