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 для действительной квадратной матрицы»**

**Выполнил**:

студент группы 382003-1

Лапин Д. А.

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Необходимо реализовать метод Гаусса, который принимает в качестве аргумента правую часть, для действительной квадратной матрицы с выбором ведущего элемента, шаблонный класс векторов и квадратных матриц как шаблон класса вектора от векторов.

# Метод решения

Решение СЛАУ используя метод Гаусса:

Метод Гаусса должен принимать на вход вектор свободных коэффициентов (далее вектор b), и с помощью элементарных преобразований над матрицей и вектором, помещает в соответствующее поле в классе SLAU и возвращает в ответе TRUE, если СЛАУ имеет единственное решение и FALSE в противном случае. Ведущий элемент на каждом шаге алгоритма выбирается как наибольший в столбце, элементы которого находятся не выше главной диагонали. После определения ведущего элемента, вся строка, содержащая его, перемещается так, чтобы элемент оказался на главной диагонали.

Прямой ход:

На каждом шаге алгоритм методом, описанным выше, выбирает ведущий элемент, после чего проверяет систему на линейную независимость (проверяет равен ли ведущий элемент нулю), если система линейно зависима, то алгоритм возвращает FALSE длины, если нет, то строка, содержащая ведущий элемент, и значение в векторе b, соответствующее ему, делятся на ведущий элемент. Следующим шагом алгоритм обнуляет все элементы ведущего столбца, находящиеся ниже диагонали, путём вычитания из соответствующих строк ведущей строки, умноженной на необходимый коэффициент. Так происходит до тех пор, пока алгоритм не обработает все столбцы. После этого матрица имеет верхне-треугольный вид и алгоритм приступает к следующему шагу.

Обратный ход:

На i-том шаге данного этапа вычисляется значение i-того икса, причём вычисление начинается с последнего и продолжается по порядку до первого икса. Вычисление происходит по формуле: , где xi – значение соответствующего икса, а aij  - элемент матрицы стоящий на пересечении i – строки и j – столбца.

Конструктор класса вектор по умолчанию обнуляет размер вектора и указатель на его первый элемент. Конструктор с параметрами сохраняет размер вектора как количество элементов в нем и выделяет динамическую память необходимого размера. Конструктор копирования позволяет создать вектор, идентичный принимаемому путем полного копирования данных второго. Деструктор освобождает выделенную память в куче, обнуляет указатель на первый элемент и обнуляет размер вектора. Оператор индексации переопределен для получения доступа к элементу вектора, операторы сложения и вычитания векторов осуществляют соответствующие операции с соответствующими элементами каждого вектора. Оператор умножения вектора на число производит умножение каждого элемента вектор на данный скаляр. Оператор присваивания позволяет одному вектору присвоить все характеристики второго. Оператор вывода соответственно перегружен для удобной работы пользователя в программе. Также с учетом всех возможных вариантов ввода пользователем размера определены функция, возвращающая размер вектора, и функция, его задающая. В каждом случае перед выходом указателя на первый элемент вектора из области видимости выделенная память возвращается в кучу.

Шаблонный класс матриц создан как вектор от векторов, поэтому некоторые его методы нет смысла прописывать в коде во избежание его дублирования. В данном классе аналогично классу векторов определены конструктор по умолчанию, конструктор с параметрами и конструктор копирования. В последнем в случае неравенства размеров матриц осуществляется расширение или же, наоборот, сужение изменяемой матрицы. В классе также переопределены операторы ввода и вывода матриц, реализована функция, задающая каждый элемент матрицы случайным числом.

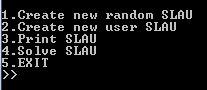
Метод Гаусса определен как функция, принимающая в качестве параметра правую часть (вектор значений). Путем последовательного выбора в каждом столбце максимального элемента не выше главной диагонали (ведущего) осуществляется обнуление всех элементов рассматриваемого столбца за исключением данного. Аналогичные действия одновременно проводятся с правой частью «уравнения», которая после полного прохода внешнего цикла по столбцам будет являться ответом задачи в случае единственного решения СЛУ.

Учтены также случаи, когда система имеет бесконечное множество решений и не имеет решений вовсе. Их программная реализация проведена с учетом « математики » метода Гаусса.

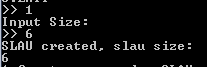
Удобство работы пользователя с программой обеспечивают инструкции на русском языке, выводимые на экран.

# Руководство пользователя

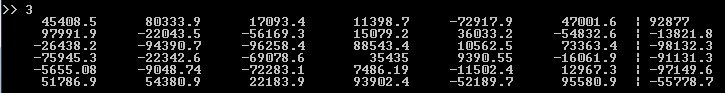
Cначала пользователю на выбор предлагается случайная генерация или собственный поэлементный ввод матрицы для решения задачи.



Пользователь вправе самостоятельно выбрать размер квадратной матрицы.

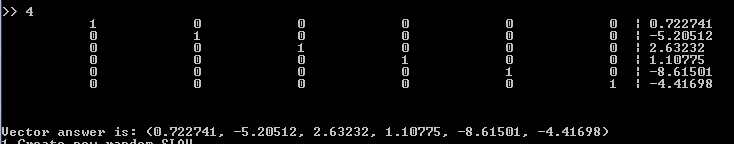


С помощью пункта меню (3) Print SLAU пользователь может вывести СЛАУ, которую или он ввел с помощью пункта меню (2), или рандомно сгенерировал с помощью пункта меню (2).



В итоге на экран выведется транспонированный столбец – ответ задачи. Вывод вектора осуществляется в строчку для улучшения читабельности ответа.

Далее пользователь может получить решение СЛАУ с помощью пункта (4) solve СЛАУ . В ответ выведется решение, или строчка “SLAU unsolavable”, если у СЛАУ нет совместных решений.



# Описание программной реализации

Для выполнения задачи требовалось создать шаблонный класс векторов, квадратных матриц как шаблон класса вектора от векторов и класс СЛАУ. Для класса векторов создана переменная, хранящая его размер, и указатель на первый элемент. С помощью указателей обеспечен ввод пользователем длины вектора. В классе созданы конструктор по умолчанию, конструктор с параметрами, конструктор копирования и деструктор. В классе переопределены операторы индексации ( [ ] ), (\*) как умножение вектора на скаляр, (+), (+=) и (-),(-=) как сложение и вычитание векторов соответственно, операция присваивания (=). В классе также определены функции задания и вывода значения размера вектора.

Класс квадратных матриц реализован как вектор векторов.

Для класса СЛАУ создана переменная, хранящая размерность матрицы, переменная matrix, определенная как вектор векторов, переменная input, определенная, как вектор чисел. Аналогично созданы конструктор от размера матрицы, конструктор от правой части СЛАУ, конструктор от правой и левой части СЛАУ. В классе реализованы функции вывода матриц, генерации случайных элементов матриц заданного размера, проверки чисел на равенство с учетом погрешности и алгоритма Гаусса.

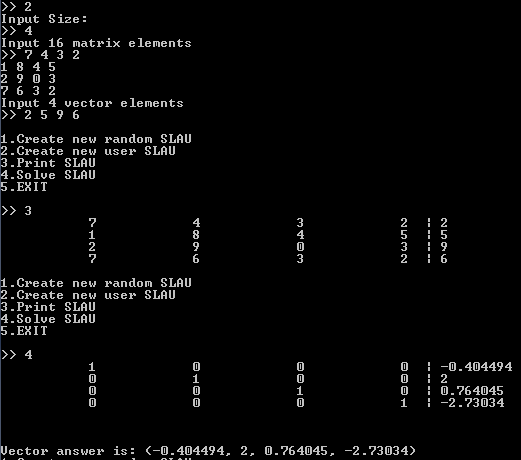
Реализация класса Vector и класса SLAU вынесена в Vector.h.

# Подтверждение корректности

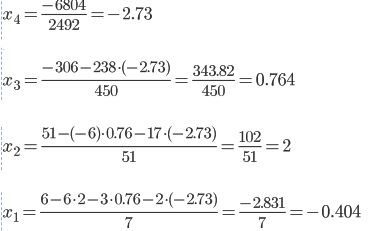
Для подтверждения корректности используются средства Интернета, позволяющие решать СЛУ онлайн. Проверка проводилась как для случайных, так и для введенных вручную матриц и столбцов.

* Пример проверки случайно сгенерированной матрицы:

Результаты, полученные программой:



Результаты, полученные в сети Интернет:



# Результаты экспериментов

По данным экспериментам видно, что программа получает верные значения. Кроме представленных экспериментов было проведено еще несколько с учетом всевозможных вариантов решения, справедливость результатов которых так же была подтверждена в Интернет-ресурсах.

# Заключение

Задача работы выполнена. Необходимые требования соблюдены, программа работает правильно.

# Приложение

