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/ка группы 3821Б1ПМ2

Соколов И. Д.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Содержание**

[Постановка задачи 3](#__RefHeading___Toc405_2279700929)

[Метод решения 4](#__RefHeading___Toc407_2279700929)

[Руководство пользователя 5](#__RefHeading___Toc409_2279700929)

[Описание программной реализации 6](#__RefHeading___Toc411_2279700929)

[Подтверждение корректности 7](#__RefHeading___Toc413_2279700929)

[Заключение 8](#__RefHeading___Toc415_2279700929)

# Постановка задачи

Цель лабораторной работы:

1) Реализовать Метод Гаусса для элементов с преобладающей диагональю

2) Описать программную реализацию

3) Подтвердить корректность данной программы

4) Сделать вывод

# Метод решения

Метод Гаусса – классический метод решения системы линейных алгебраических уравнений (СЛАУ). Назван в честь немецкого математика Карла Фридриха Гаусса. Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида, из которой последовательно, начиная с последних (по номеру), находятся все переменные системы.

Для начала реализуем три класса необходимые для метода Гаусса – класс вектор, в котором будут храниться переменные, класс матрица, которая будет хранить в себе все переменные из матрицы А (для уравнения A\*x = b, где x,b – векторы) и класс СЛАУ которая будет содержать в себе основную реализацию метода Гаусса.

# Руководство пользователя

При запуске программы, на экране появится следующее (рис.1).

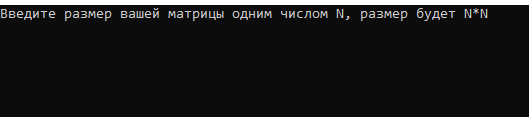


Рис.1

Рис.1

После этого программа будет ждать пока вы не введете(тип int) размерность вашей матрицы. Для примера введем число 3. После этого на экране появится следующее (рис.2).

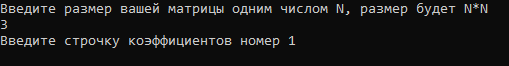


Рис.2

Программа будет печатать надпись „Введите строчку коэффициентов номер *i*“ каждый раз когда вы ввели числа n раз, где n – это размерность матрицы, пока *i* не дойдет до размерности матрицы, в нашем случае до 3.Таким образом мы заполняем коэффициенты матрицы А. Выглядеть это будет так (рис.3).

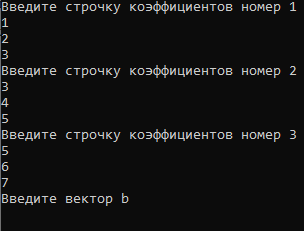


Рис.3

После этого (как видно на рис.3) программа будет ждать пока вы введете числа n раз, в нашем случае 3 раза. Таким образом мы заполним вектор b. После этого на экране появится следующее (рис.4).

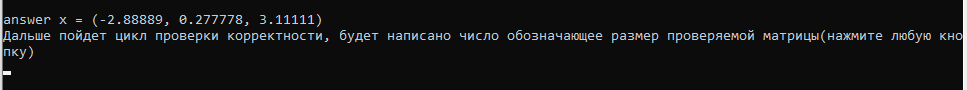


Рис.4

На экране появится ответ в результате решения метода Гаусса с введенными коэффициентами, то есть вектор x, в нашем случае длинной 3. После этого появится надпись предупреждающая, что дальше пойдет проверка корректности. Для продолжения нажмите на любую кнопку после чего появится следующее (Рис.5).

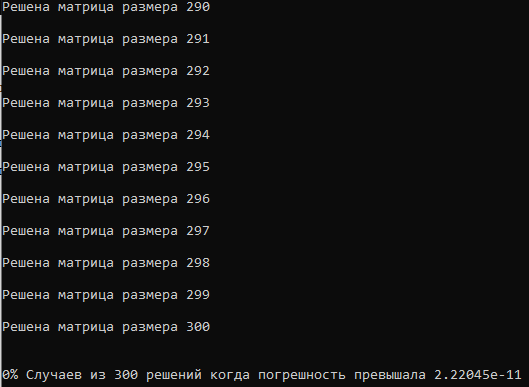


Рис.5

Программа напечатает 300 раз «Решена матрица размера i», что означает что в результате решения методом Гаусса СЛАУ размером i\*i, ответ оказался корректным с допустимой погрешностью(которая установлена по умолчанию на 10^(-11)). Если погрешность превышает допустимую то может быть написано следующее (Рис.6). В конце программы будет написан процент случаев из 300 решений когда погрешность превышала допустимую.



Рис.6

# Описание программной реализации

class Myexception — класс исключений который при вызове исключения в данной программе показывает один из трех вариантов ответов, что за исключение было вызвано.

Myexception::Myexception(int k):k(k){} - конструктор который присваивает полю k, аргумент k.

Myexception::const char\* what() const throw() - метод класса Myexception, который возвращает строку ответ, что за исключение было вызвано.

template<class T>class Vecktor — шаблонный класс вектор, который хранит в себе динамический массив чисел типа Т и размер массива.

Vecktor(){} – конструктор по умолчанию.

Vecktor(int n) — конструктор создающий динамический массив размера n.

void resize(int n) – меняет размер массива.

~Vecktor() - деструктор удаляющий динамический массив.

Vecktor& operator=(Vecktor& a) — оператор копирования векторов.

Vecktor& operator\*=(T s) – допускает умножение вектора на скаляр.

Vecktor& operator-=(Vecktor& a) – допускает вычитание векторов.

ostream& operator<<(ostream& out) – оператор позволяющий делать вывод вектора на экран.

T operator[](int i) – допускает обращение к вектору через квадратные скобки.

void set() – метод для задания значений вектора.

void setr(int s) – метод для случайного задания значений вектора.

template<class T>class Matrix : public Vecktor<Vecktor<T>> – шаблонный класс матрица, который хранит в себе вектор шаблонных векторов и их размеры.

Matrix(int n) :Vecktor<Vecktor<T>>(n) – конструктор матрицы, который «расширяет» матрицу до размера n\*n.

void set() – метод для задания значений матрицы.

ostream& operator<<(ostream& out) – оператор позволяющий делать вывод матрицы на экран.

template<class T>class SLAU : public Matrix<T> – шаблонный класс СЛАУ который хранит в себе шаблонные матрицу А и векторы x и b.

SLAU(int n):Matrix<T>(n) – создает СЛАУ и задает значения для ее для матрицы размера n\*n.

SLAU(int n,int s):Matrix<T>(n) – создает СЛАУ и задает случайные значения для ее для матрицы размера n\*n.

double Check( Vecktor<T>& B,Matrix<T>& M) – метод необходимый для проверки корректности метода Гаусса.

void Method\_Gaussa(Vecktor<T> &b1,Vecktor<T> &X) – метод Гаусса для которого вектор b = b1, заменяет вектор X на x.

int main() – cоздание СЛАУ с вызовом метода Гаусса, попытка поймать исключение на данном участке.

# Подтверждение корректности

Для подтверждения корректности метода Гаусса используется метод класса СЛАУ Check. В нем изначальная матрица умножается на полученный вектор x и результат сравнивается с изначальным вектором b. Если погрешность координаты xi превышает bi более чем на допустимую, то в программе выведем данную погрешность. Если нашлась координата у которой погрешность больше, то выведем ее.

# Заключение

В результате проведенной работы был реализован метод Гаусса и был проверен на корректность. Можно заметить что данная программа работает корректно, только потому что в цикле для проверки корректности матрица была с преобладающей диагональю. В ином случае погрешность может резко подскочить из-за невозможности компилятора правильно высчитывать несократимые дроби.