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**:

студентка группы 382003-1

Капустина П.Д.

**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

Реализовать метод Гаусса для действительной квадратной матрицы с выбором ведущего элемента, используя реализованный самостоятельно шаблонный класс вектор, а также класс квадратная матрица, являющаяся шаблоном класса вектор от вектора. Метод Гаусса должен принимать на вход систему линейных алгебраических уравнений, а выводить на экран вектор значений. Экспериментально подтвердить работу созданного алгоритма.

# Метод решения

*Прямой ход:*

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 максимальный элемент и перемещают его на крайнее верхнее положение перестановкой строк. Затем нормируют первое уравнение, разделив его на коэффициент ai1, где i– номер столбца. Затем из остальных строк вычитают получившуюся после перестановки первую строку, умноженную на элемент a1j, где j – номер строки. 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.

*Обратный ход:*

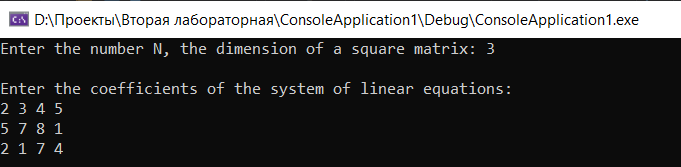
Обратная подстановка предполагает подстановку полученного на предыдущем шаге значения переменной xn в предыдущие уравнения. Эта процедура повторяется для всех оставшихся решений.

# Руководство пользователя

Пользователь должен ввести одно натуральное число, которое задает размеры матрицы N.

После ввода числа Пользователь должен ввести коэффициенты системы уравнений, где первые N чисел строки являются коэффициентами, а N+1 число, то чему равно значение выражения первой строки. Это правило действует для всех N строк.

Пример правильной работы с программой представлен ниже:



.

# Описание программной реализации

В программе содержатся следующие функции и структуры:

**Vector –** шаблонный класс вектор, хранит значения любого типа, при вызове создает динамический массив. Имеет в себе различные методы, функции и перегруженные бинарные операции «+», «-», «\*», «/» и «=».

**Matrix –** шаблонный класс матрица, является наследником класса вектор от вектора.

**Gauss –** класс, являющийся наследником класса матрица. Содержит функцию, вычисляющая наличие ответа.

**Otvet –** функция типа Vector, в ней реализован метод решения Гаусса. Возвращает вектор, являющийся ответом системы.

# Подтверждение корректности

Для подтверждения корректности в программе присутствует проверка, в конце реализации функции *Otvet.* Цикл находит определитель получившейся треугольной матрицы и если тот равен 0, то выводит сообщение об ошибке и возвращает вектор со значениями «-nan(ind)».

# Результаты экспериментов

В качестве эксперимента я ввела самую простую матрицу, значения которой можно легко посчитать вручную и почти сразу видно решаема она или нет. Я решила подставить различные значение переменных и проверить, что выводит моя программа на экран. Часть проверенных значений представлено ниже.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Матрица | | Решения | Переменные | Вектор | Проверка |
| 1 | 2 | 3 | X1 | -1 |  |
| 4 | 5 | 6 | X2 | 2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Матрица | | Решения | Переменные | Вектор | Проверка |
| 0 | 2 | 6 | X1 | -nan(ind) |  |
| 0 | 1 | 3 | X2 | inf |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Матрица | | Решения | Переменные | Вектор | Проверка |
| 0 | 0 | 0 | X1 | -nan(ind) |  |
| 0 | 0 | 0 | X2 | -nan(ind) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Матрица | | Решения | Переменные | Вектор | Проверка |
| 6666 | 0 | 1 | X1 | 0.000150015 |  |
| 0 | 6666 | 1 | X2 | 0.000150015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Матрица | | Решения | Переменные | Вектор | Проверка |
| 1018 | 0 | 1 | X1 | 10-18 |  |
| 0 | 10 19 | 1 | X2 | 10-19 |

# Заключение

Из-за моей реализации метода Гаусса при наличии нулевого столбца в матрице вектор решения не может быть представлен корректно и на данный момент выводит сообщение об ошибке и неправильный вектор. Так же при увеличении числа переменных и самих чисел матрицы понижается точность вычислений.

.

# Приложение

class Matrix : public Vector<Vector<T>> {

public:

Matrix(size\_t stolb, const Vector<T>& stroka) : Vector<Vector<T>>(stolb, stroka) {}

};

template <typename T>

class Gauss : public Matrix<T> {

public:

Gauss(size\_t stolb, const Vector<T>& stroka) : Matrix<T>(stolb, stroka) {};

Vector<T> Otvet() {

size\_t razmer = this->GetSize();

Vector<T> otvet(razmer);

size\_t max;

int zero = 0;

for (size\_t i = 0; i < razmer; i++) {

zero = 0;

for (size\_t j = i + 1; j < razmer; j++)

if ((\*this)[i][j] == 0) zero++;

if (zero == razmer) error(3);

}

for (size\_t i = 0; i < razmer; i++) {

max = i;

for (size\_t j = i + 1; j < razmer; j++) {

if ((\*this)[i][j] \* (\*this)[i][j] > (\*this)[i][max] \* (\*this)[i][max]) max = j;

}

swap<Vector<T>>((\*this)[i], (\*this)[max]);

(\*this)[i] = (\*this)[i] / (\*this)[i][i];

for (size\_t j = i + 1; j < razmer; j++) (\*this)[j] = (\*this)[j] - ((\*this)[i] \* (\*this)[j][i]);

}

for (int i = razmer - 1; i >= 0; i--) {

otvet[i] = (\*this)[i][razmer];

for (int j = razmer - 1; j >= i + 1; j--) otvet[i] = otvet[i] - (\*this)[i][j] \* otvet[j];

}

int check = 1;

for (int i = 0; i < razmer; i++)

check = check \* (\*this)[i][i];

if (check != 0)

return otvet;

else {

error(1);

return otvet;

}

}

};

Вывод: лаба написана, время потрачено, а с результатом всё ещё проблемы

Для проверки использовался онлайн-калькулятор: https://ru.onlinemschool.com/math/assistance/equation/gaus/