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 382003-1

Измайлов И.Р.

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021г.

**Содержание**

[Постановка задачи 3](#_Toc71150796)

[Алгоритм решения СЛУ методом Гаусса 4](#_Toc71150797)

[Руководство пользователя 5](#_Toc71150798)

[Описание программной реализации 7](#_Toc71150799)

[Приложение 8](#_Toc71150800)

# Постановка задачи

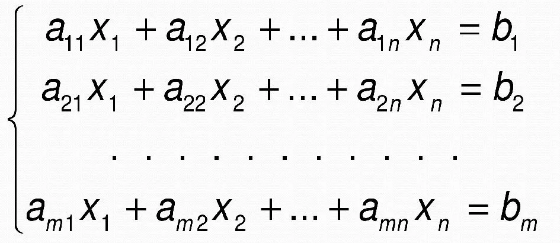
Передо мной была поставлена задача – повторить весь материал по языку программирования С++, который мы успели пройти во втором семестре. На основе этих знаний написать программу, которая бы реализовывала решение системы линейных уравнений методом Гаусса.

Линейное уравнение с n неизвестными X1, X2, …, Xn может быть определено при помощи выражения



где величины a1, a2, …, an и b представляют собой постоянные значения.

Множество n линейных уравнений



носит название Система линейных уравнений(или сокр. СЛУ). Обычно еще записывать в матричном виде

Ax = b,

где A – матрица размера n x n, а вектор b и х состоят из n элементов.

# Алгоритм решения СЛУ методом Гаусса

**Прямой ход**

Прямой ход метода Гаусса состоит из нескольких шагов.

Последовательно, то есть столбец за столбцом в матрице, находится ведущий элемент. Ведущий элемент выбирается по принципу, как самое большое по модулю значение. После этого ведущий элемент ставится на диагональную позицию. Согласно эквивалентным линейным преобразованиям, нам необходимо поменять местами строки. После этого, зануляем все элементы в столбце, находящиеся ниже главного элемента, согласно правилам эквивалентных линейных преобразований.

Используются следующие формулы:

(1) Умножение k-ой строки на число

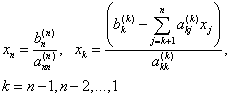
(2) Вычитание k-ой строки из j-о1 строки

(3)

**Обратный ход**

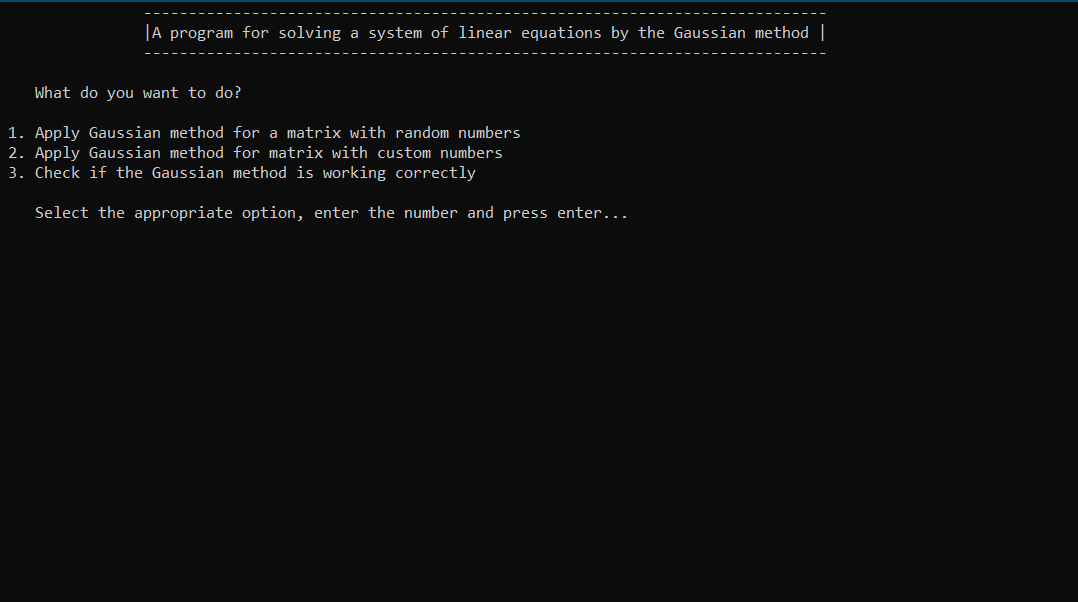
Вторая часть метода Гаусса заключается в определении значений неизвестных. Из последнего уравнения преобразованной системы можно вычислить значение переменной Xn-1, зачет из предпоследнего уравнения становится возможным определение переменной Xn-2 и т.д.

Обратный ход реализуется по следующим формулам, начиная с последнего уравнения системы:

(4)

# Руководство пользователя

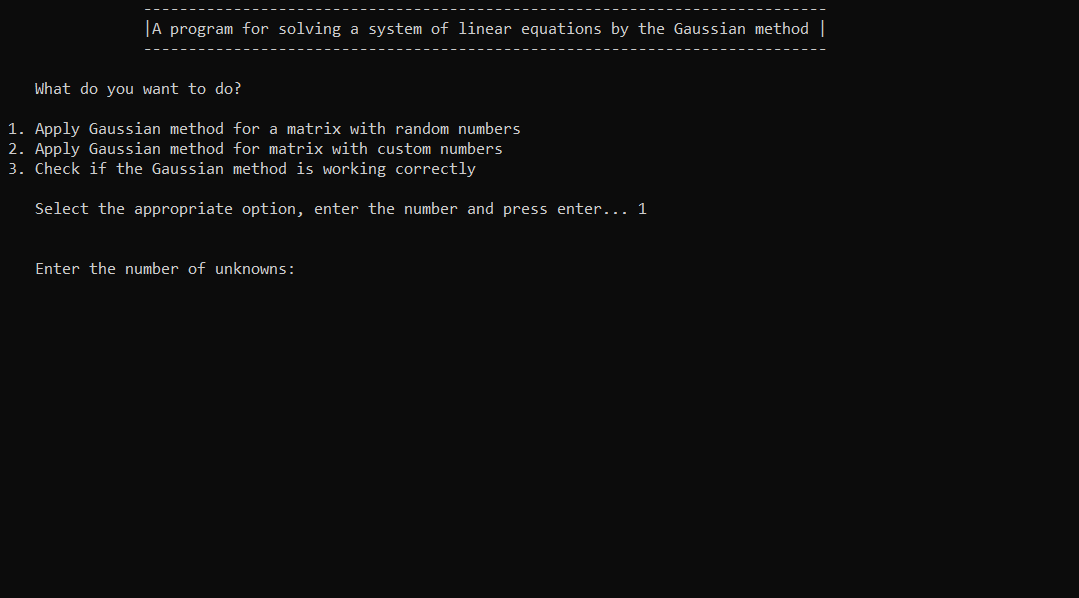
1. Запуск программы



Стартовая страница программы предлагает пользователю три функции:

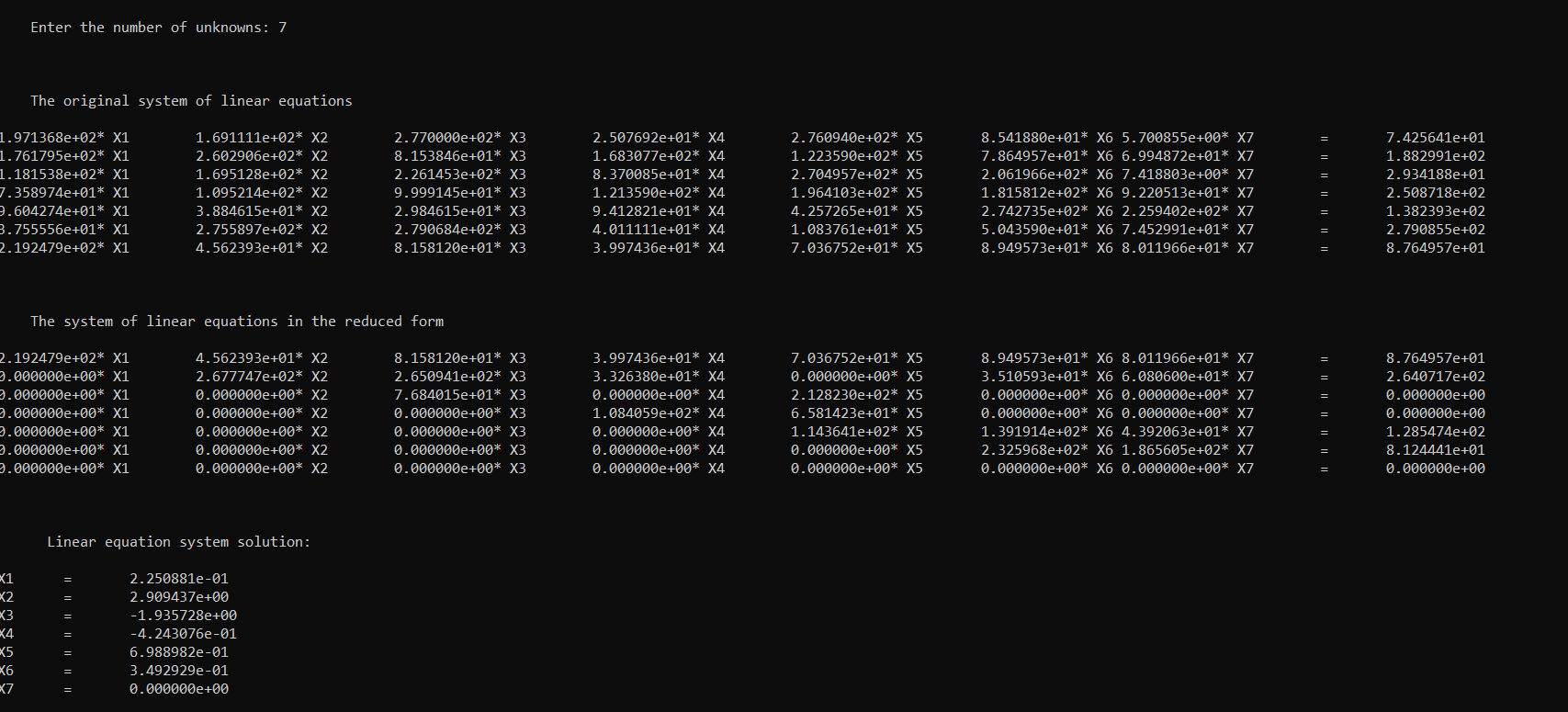
1. Применить метод Гаусса для решения системы линейных уравнений, коэффициенты неизвестных и свободных членов заданы случайно (переведено с англ.)
2. Применить метод Гаусса для решения системы линейных уравнений, коэффициенты которых и свободные члены введены пользователем (переведено с англ.)
3. Проверка правильности выполнения программы (переведено с англ.)

Для наглядности работы программы мы введем цифру 1 и нажмем Enter.



Программа просит нас «ввести количество неизвестных величин (прим. В системы линейных уравнений)». Пользователю необходимо определиться для какого количество неизвестных необходимо применить метод Гаусса. Как можно понять исходя из идеи реализации, тем самым мы задаем размерность матрицы для коэффициентов.

В этом примере мы будем работать с 7 неизвестными величинами. Введем число 7 и нажмем Enter.



Результат выполнения программы:

Сначала программа выведет заданную систему линейных уравнений, в которой 7 неизвестных величин, коэффициенты которого вместе со свободным членом заполнены произвольными величинами.

После этого программа покажет нашу систему в приведенной форме ( то есть после выполнения прямого хода программы).

И наконец решение системы, которая показывает корни уравнений.

# Описание программной реализации

Мой проект имеет название Laboratory 2. В ней содержится 5 файлов: main.cpp, matrix.h, vector.h, interface.h, interface.cpp. В файле main.cpp находится функция int main(), с которой начинается выполнение программы. Она выводит название нашей программы. В файле interface.h находится функция, которая предоставляет пользователю варианты работы с программой и имеет название logic() . В зависимости от введенной пользователем цифры, будет выполняться та или иная функция.

Решение СЛУ, коэффициенты и свободные члены которой заданы случайными числами, реализует функция RandomNumberGauss(), находящаяся также в файле interface.h решение пользовательской СЛУ – userValues() , проверка правильности выполнения программы - Check() . В каждой из этих функций мы работаем с шаблонными классами Vector и Matrix. Class Matrix – дочерний class Vector. Класс Matrix находится в файле matrix.h, а класс Vector – в vector.h.

# Приложение

void GaussMethod(Vector<T>& x)

{

for (size\_t i = 0; i < this->size; i++)// Прямой ход

{

int max = maxIndexInCol(i);

swapRow(max, i);

for (int row = i + 1; row < this->size; row++)

{

if (row != i && this->data[row][i] != 0) {

T mnozhitel = (this->data[row][i]) / (this->data[i][i]); // формула (1)

for (int k = i; k < this->size+1; k++)

{

this->data[row][k] -= (this->data[i][k]) \* mnozhitel; //формула (3)

if (this->data[row][k] < MIN)

{

this->data[row][k] = 0;

}

}

}

}

}

for (int i = this->size - 1; i >= 0; i--) // Обратный ход

{

double sum = this->data[i][this->size];

for (size\_t j = this->size-1; j > i; j--)

{

sum -= this->data[i][j] \* x[j]; // формула (4)

}

if (this->data[i][i] == 0)

{

x[i] = 0;

}

else

{

x[i] = sum / this->data[i][i];формула (4)

}

}

}