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 3821Б1ПМ2

Ермолаев Д.А.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

Оглавление

**Постановка задачи3**

**Методы решения4**

**Руководство** **пользователя6**

**Описание программной реализации** **7**

**Подтверждение** **корректности9**

**Результаты экспериментов10**

**Заключение12**

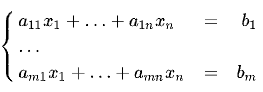
# Постановка задачи

Целью лабораторной работы является реализовать на языке программирования С++ свои шаблоны классов вектор, матрица, СЛАУ (системы линейных алгебраических уравнений). Также программа должна решить введённую СЛАУ. Нужно описать программную реализацию и алгоритмы работы данных вычислений. Необходимо подтвердить корректность реализации метода Гаусса путём вычисления вектора невязки СЛАУ.

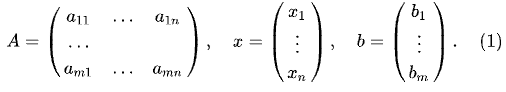
**Методы решения**

Для решения системы линейных алгебраических уравнений использовался модифицированный метод Гаусса.

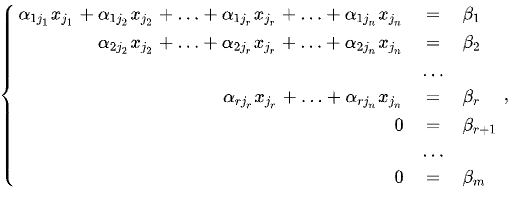
Пусть исходная система выглядит следующим образом:



Её можно записать в матричном виде: где



Тогда, согласно свойству элементарных преобразований над строками, основную матрицу этой системы можно привести к ступенчатому виду (эти же преобразования нужно применять к столбцу свободных членов):



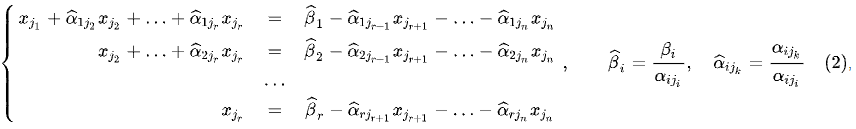
где

При этом будем считать, что базисный минор (ненулевой минор максимального порядка) основной матрицы находится в верхнем левом углу, то есть в него входят только коэффициенты при переменных .{\displaystyle x\_{j\_{1}},\ldots ,x\_{j\_{r}}}

Тогда переменные  {\displaystyle x\_{j\_{1}},\ldots ,x\_{j\_{r}}}  называются главными переменными. Все остальные называются свободными.

Если хотя бы одно число {\displaystyle \beta \_{i}\neq 0} где{\displaystyle i>r} то рассматриваемая система несовместна, т. е. у неё нет ни одного решения.

Пусть {\displaystyle \beta \_{i}=0}   для любых {\displaystyle i>r}

Перенесём свободные переменные за знаки равенств и поделим каждое из уравнений системы на свой коэффициент при самом левом {\displaystyle \alpha \_{ij\_{i}},\,i=1,\ldots ,r}где{\displaystyle i} —номер строки): где

Если свободным переменным системы (2) придавать все возможные значения и решать новую систему относительно главных неизвестных снизу вверх (то есть от нижнего уравнения к верхнему), то мы получим все решения этой СЛАУ. Так как эта система получена путём элементарных преобразований над исходной системой (1), то по теореме об эквивалентности при элементарных преобразованиях системы (1) и (2) эквивалентны, то есть множества их решений совпадают.

Руководство пользователя

Пользователю предлагается ввести размер СЛАУ и тип ввода коэффициентов: с клавиатуры или случайно сгенерированные. Если размер СЛАУ меньше 10, программа выводит систему, результат вычислений и корректность вычислений на экран. Если размер СЛАУ больше или равняется 10, программа выводит результат вычислений и корректность вычислений на экран.

Описание программной реализации

Библиотеки, которые использовались в работе: <iostream>, <cmath>, <ctime>.

Основные классы:

1. vector<T>
2. matrix<T>
3. SLAU<T>

Поля класса vector<T>:

1. T\* mas – массив, в котором хранятся координаты вектора.
2. int size – размерность вектора.

Основные методы класса vector<T>:

1. Конструктор по умолчанию: vector() – присваивает полю size значение 0.
2. Конструктор с параметрами: vector(int size\_) – освобождает место в mas на size\_ элементов и присваивает size значение size\_.
3. Конструктор копирования: vector(vector& s) – создаёт вектор с данными из вектора s.
4. Деструктор: ~vector() – удаляет mas.
5. T& operator[] (int i) – возвращает i – й элемент mas.
6. void print\_vector() – выводит вектор на экран.
7. void division\_val(T val) – делит вектор на значение val.
8. void resize(int size\_) – меняет размер вектора на значение size\_.
9. void swap\_rows(int i1, int i2) – меняет i1 – ю и i2 – ю координаты вектора местами.

Класс matrix<T> наследуется от vector<vector<T>>.

Основные методы класса matrix<T>:

1. Конструктор по умолчанию: matrix() : vector<vector<T>>() – присваивает полю size значение 0.
2. Конструктор с параметрами: matrix(int size) : vector<vector<T>>(size) – освобождает место в mas на size\_ \* size\_ элементов и присваивает size значение size\_.
3. Конструктор копирования: matrix(matrix& s) – создаёт матрицу с данными из матрицы s.
4. void print\_matrix() - выводит матрицу на экран.
5. vector<T> multiply\_vec(vector<T>& s) – умножает матрицу на вектор s. Возвращает результат.
6. void swap\_rows(int i1, int i2) – меняет i1 – ю и i2 – ю строки матрицы местами.

Поля класса vector<T>:

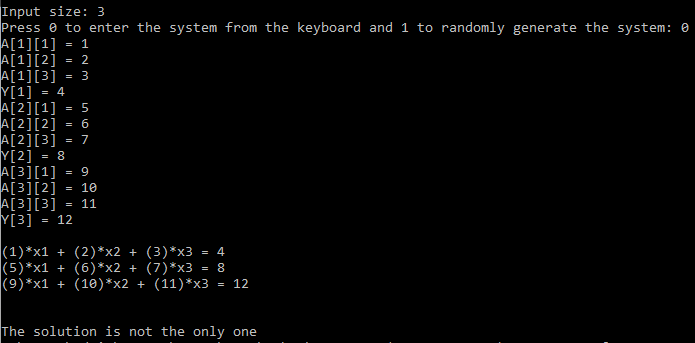
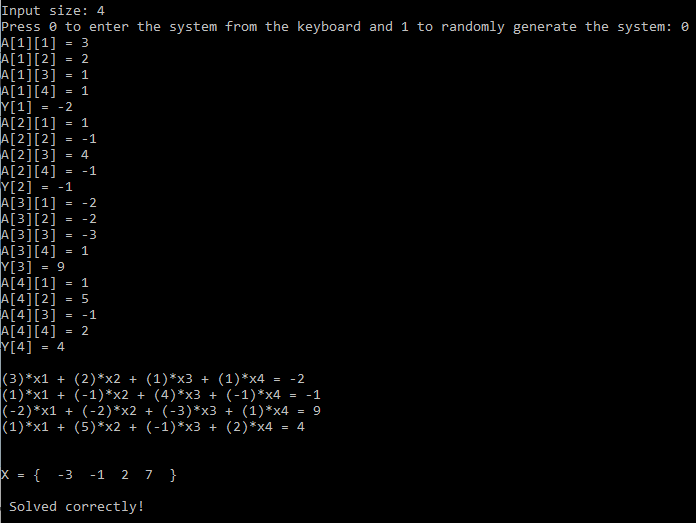
1. matrix<T> A – матрица A СЛАУ.
2. vector<T> Y – матрица b СЛАУ.
3. vector<T> X – матрица x СЛАУ.
4. int rows – количество неизвестных в СЛАУ.

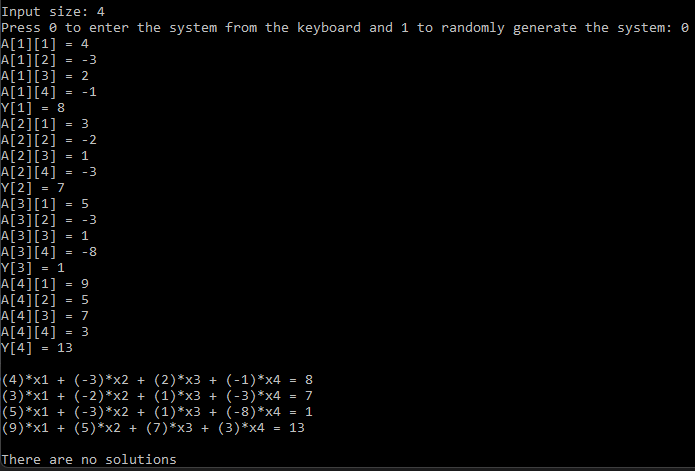
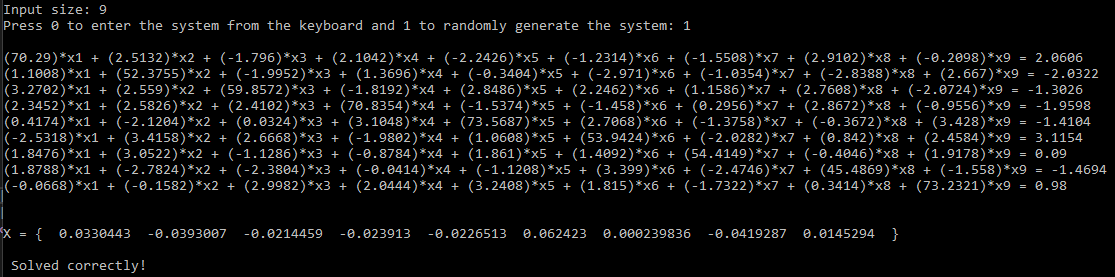
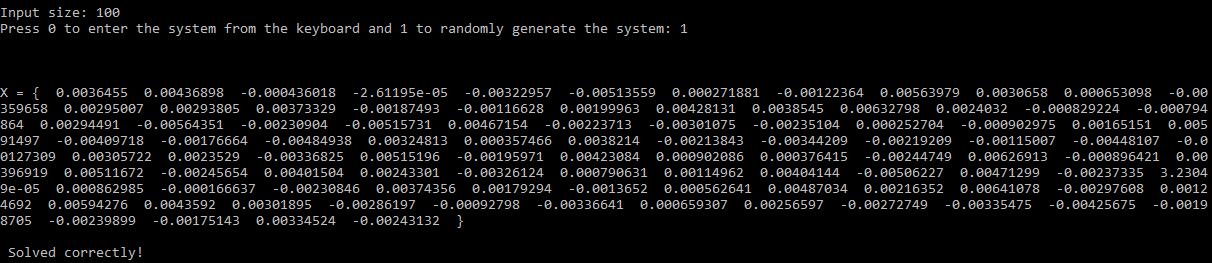
Основные методы класса SLAU<T>:

1. Конструктор с параметрами:
2. void set\_SLAU() – ввод коэффициентов СЛАУ.
3. void print\_SLAU()– вывод системы на экран.
4. vector<T> Gauss() – метод Гаусса для решения СЛАУ.
5. void fill\_rand() – случайная генерация диагонально преобладающей СЛАУ.
6. void correct\_check() – проверка корректности решения.

Подтверждение корректности

Для подтверждения корректности в программе высчитывался вектор невязки СЛАУ. Если для каждой координаты вектора невязки отклонение меньше 0,0000001, то система решена правильно. Если у системы нет решений, программа выводит на экран “There are no solutions”. Если у системы решение не единственное, программа выводит на экран “The solution is not the only one”.

**Результаты экспериментов**



**Заключение**

В ходе лабораторной работы на языке программирования С++ были реализованы шаблоны классов вектор, матрица и СЛАУ, а также модифицированный метод Гаусса, решающий введённую систему. Были описаны алгоритмы работы данных вычислений, их программная реализация и проведены эксперименты для подтверждения их корректности, методом вычисления отклонения каждой координаты вектора невязки СЛАУ. Проведенные эксперименты показали, что программа работает корректно.