Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**Отчёт**

к лабораторной работе

на тему:

Решение систем линейных алгебраических уравнений (СЛАУ)

методом Гаусса и с помощью его модификаций

                               Выполнил студент группы 253505

Шпаковский Антон Владимирович

Проверил: Анисимов Владимир Яковлевич

Минск 2023

# **ЦЕЛИ**

* Изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ
* Составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ
* Составить программу решения СЛАУ по разработанному алгоритму
* Выполнить тестовые примеры и проверить правильность работы программы

**Введение**

СЛАУ являются важным элементом в различных областях науки и техники. Они возникают при решении задач в физике, инженерии, экономике и других дисциплинах. Метод Гаусса — один из основных методов для нахождения решений СЛАУ. Он заключается в преобразовании исходной системы уравнений к эквивалентной системе с более простой структурой, что позволяет найти решение.

В данном отчете рассмотрены основы метода Гаусса и его применение для решения СЛАУ. Были представлены основные шаги метода и примеры его применения на практике. Также были рассмотрены модификации метода Гаусса, включая метод Гаусса с выбором главного элемента по столбцу и метод Гаусса с выбором главного элемента по всей матрице.

# **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Система линейных алгебраических уравнений (СЛАУ) - это набор уравнений, в которых все переменные входят в первой степени и коэффициенты при них являются константами. Общий вид СЛАУ можно записать следующим образом:



Здесь ***А*** и ***b*** заданы, требуется найти ***х***.

Методы решения систем линейных алгебраических уравнений (СЛАУ) можно разделить на две основные категории: прямые (или методы прямого подстановочного типа) и итерационные методы.

1. Прямые методы: Прямые методы применяются для точного решения СЛАУ и обеспечивают точное решение системы. Они основаны на последовательных математических операциях, таких как умножение, вычитание и деление, чтобы найти значения неизвестных.

2. Итерационные методы: Итерационные методы используются для приближенного решения СЛАУ. Они начинают с начального приближения и последовательно уточняют решение на каждой итерации.

Прямые методы обычно требуют более высокой вычислительной мощности и памяти, но они гарантируют точное решение. Наибольшее распространение среди прямых методов получили метод Гаусса и его модификации.

Метод Гаусса обладает рядом преимуществ по сравнению с другими методами:

1. Точное решение: Метод Гаусса гарантирует точное решение СЛАУ при условии, что система уравнений имеет решение и матрица коэффициентов невырожденная.

2. Универсальность: Метод Гаусса применим к разнообразным видам СЛАУ с различными коэффициентами и правыми частями. Он не зависит от специфических характеристик системы и применим к системам с произвольным числом переменных и уравнений.

3. Простота реализации: Алгоритм метода Гаусса относительно прост в понимании и реализации. Он состоит из двух основных этапов: прямого и обратного хода, что делает его доступным для программирования и вычислений вручную.

4. Стабильность: В большинстве случаев метод Гаусса стабилен и надежен. Он не подвержен численным неустойчивостям, которые могут возникнуть при использовании некоторых итерационных методов.

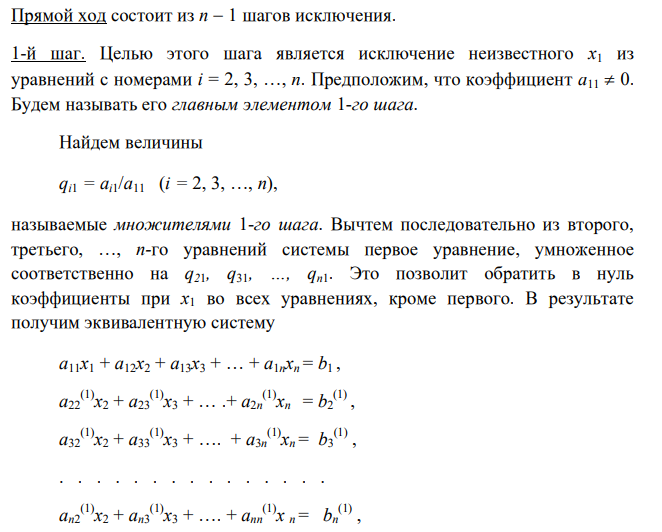
Метод Гаусса, используемый для решения систем линейных алгебраических уравнений (СЛАУ), включает два основных этапа: прямой и обратный ходы.

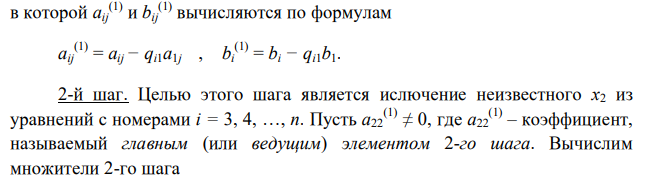
Прямой ход метода Гаусса заключается в преобразовании расширенной матрицы системы так, чтобы получить верхнюю треугольную матрицу с нулями под главной диагональю. Это достигается путем последовательного исключения переменных из уравнений.

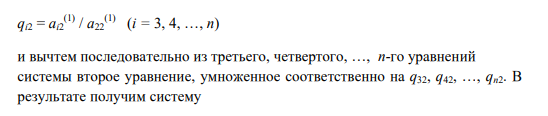
Обратный ход, который иногда называется методом Гаусса-Жордана, представляет собой этап, в котором матрица приводится к диагональному виду, и значения переменных вычисляются от последней к первой. Это также включает в себя последовательное исключение переменных, но на этот раз вверх от главной диагонали.

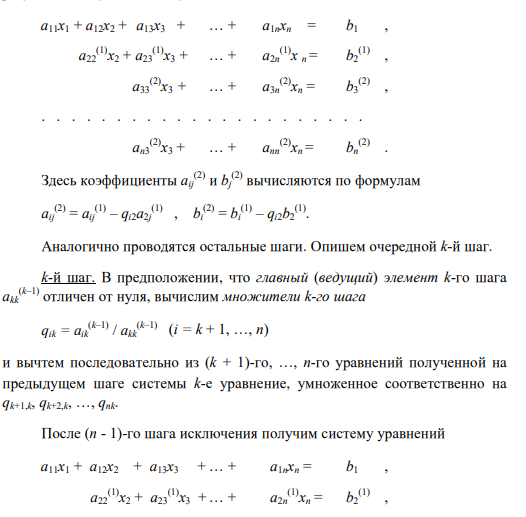
Таким образом, метод Гаусса и метод Гаусса-Жордана отличаются только последовательностью шагов исключения переменных. Оба метода являются прямыми методами решения СЛАУ, и оба обеспечивают точное решение системы, при условии, что система имеет единственное решение и матрица коэффициентов невырожденная.

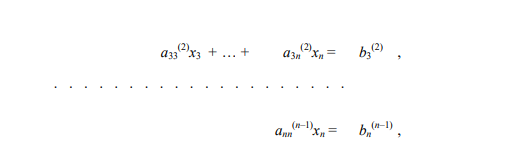
Метод Гаусса идеально подходит для решения систем содержащих больше трех линейных уравнений, для решения систем уравнений, которые не являются квадратными (чего не скажешь про метод Крамера и матричный метод). То есть метод Гаусса - наиболее универсальный метод для нахождения решения любой системы линейных уравнений, он работает в случае, когда система имеет бесконечно много решений или несовместна.



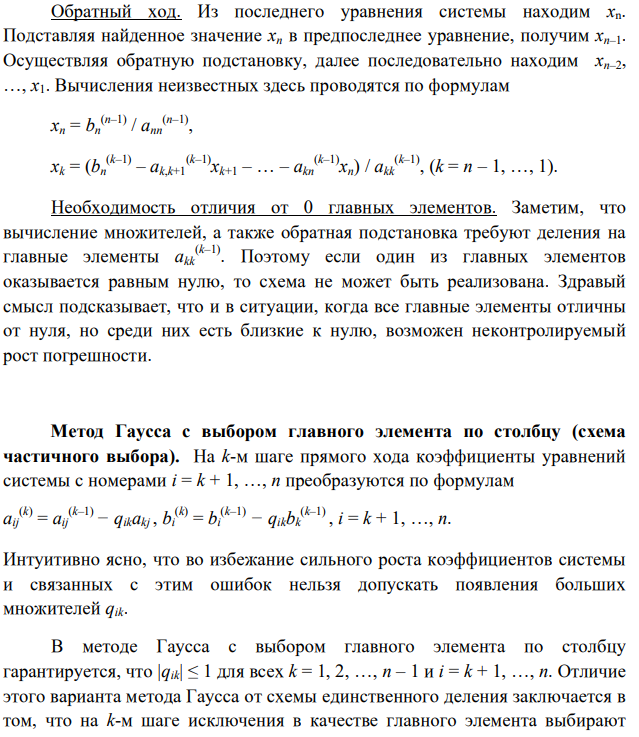


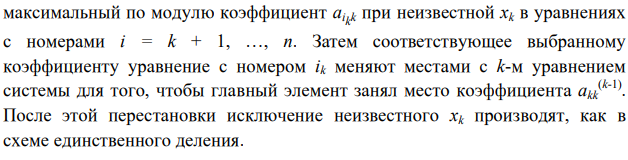


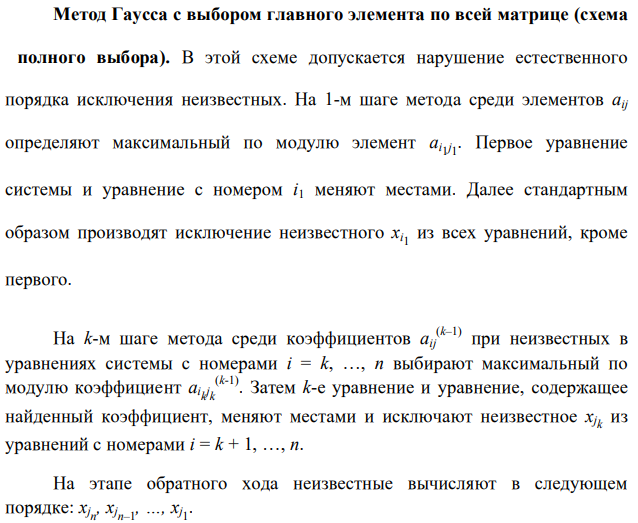




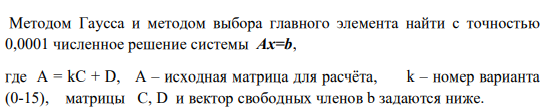




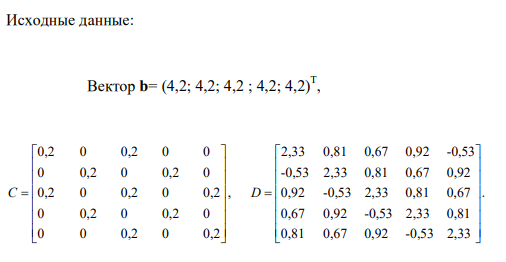




# **ЗАДАНИЕ**



Вариант 29



# **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

# #include <iostream>

# #include <iomanip>

# using namespace std;

# 

# const int n = 5;

# const int acc = 4;

# 

# double\* gauss1(double A[n][n], double\* b, int n)

# {

# double\* x = new double[n];

# for (int i = 0; i < n; i++)

# {

# if (A[i][i] != 0)

# {

# for (int j = i + 1; j < n; j++)

# {

# double q = A[j][i] / A[i][i];

# for (int k = i; k < n; k++)

# {

# if (k == i)

# A[j][k] = 0;

# else

# A[j][k] -= q \* A[i][k];

# }

# b[j] -= q \* b[i];

# }

# }

# else

# {

# cout << "Данным методом невозможно решить систему: ведущий элемент стал равен 0\n";

# return x;

# }

# }

# 

# for (int i = n - 1; i >= 0; i--)

# {

# for (int j = n - 1; j > i; j--)

# {

# b[i] -= x[j] \* A[i][j];

# }

# x[i] = b[i] / A[i][i];

# }

# 

# return x;

# }

# 

# 

# 

# double\* gauss2(double A[n][n], double\* b, int n)

# {

# double\* x = new double[n];

# 

# for (int i = 0; i < n; i++)

# {

# int max = abs(A[i][i]);

# int maxIndex = i;

# for (int j = i + 1; j < n; j++)

# {

# if (abs(A[j][i]) > max)

# {

# maxIndex = j;

# max = abs(A[j][i]);

# }

# }

# for (int j = 0; j < n; j++)

# {

# swap(A[i][j], A[maxIndex][j]);

# }

# swap(b[i], b[maxIndex]);

# 

# if (A[i][i] != 0)

# {

# for (int j = i + 1; j < n; j++)

# {

# double q = A[j][i] / A[i][i];

# for (int k = i; k < n; k++)

# {

# if (k == i)

# A[j][k] = 0;

# else

# A[j][k] -= q \* A[i][k];

# }

# b[j] -= q \* b[i];

# }

# }

# else

# {

# cout << "Данным методом невозможно решить систему: столбец возможных ведущих элементов стал равен 0\n";

# return x;

# }

# 

# }

# 

# for (int i = n - 1; i >= 0; i--)

# {

# for (int j = n - 1; j > i; j--)

# {

# b[i] -= x[j] \* A[i][j];

# }

# x[i] = b[i] / A[i][i];

# }

# return x;

# }

# 

# double\* gauss3(double A[n][n], double\* b, int n)

# {

# double\* x = new double[n];

# 

# for (int i = 0; i < n; i++)

# {

# double max = abs(A[i][0]);

# int maxIndexX = i;

# int maxIndexY = 0;

# for (int j = i; j < n; j++)

# {

# for (int k = 0; k < n; k++)

# {

# if (abs(A[j][k]) > max)

# {

# maxIndexX = j;

# maxIndexY = k;

# max = abs(A[j][k]);

# }

# }

# }

# for (int j = 0; j < n; j++)

# {

# swap(A[i][j], A[maxIndexX][j]);

# }

# swap(b[i], b[maxIndexX]);

# 

# if (A[i][maxIndexY] != 0)

# {

# for (int j = i + 1; j < n; j++)

# {

# double q = A[j][maxIndexY] / A[i][maxIndexY];

# for (int k = 0; k < n; k++)

# {

# if (k == maxIndexY)

# A[j][k] = 0;

# else

# A[j][k] -= q \* A[i][k];

# }

# b[j] -= q \* b[i];

# }

# }

# else

# {

# cout << "Данным методом невозможно решить систему: элементы в оставшихся строках стали нулевыми (система не имеет единственное решение)\n";

# return x;

# }

# }

# 

# bool\* xFound = new bool[n];

# for (int i = 0; i < n; i++)

# {

# xFound[i] = 0;

# }

# 

# for (int i = n - 1; i >= 0; i--)

# {

# int curX;

# for (int j = 0; j < n; j++)

# {

# if (A[i][j] != 0 && xFound[j] == 0)

# {

# curX = j;

# }

# if (A[i][j] != 0 && xFound[j] == 1)

# {

# b[i] -= x[j] \* A[i][j];

# }

# }

# x[curX] = b[i] / A[i][curX];

# xFound[curX] = 1;

# }

# return x;

# }

# 

# int main()

# {

# cout.precision(acc);

# cout.setf(ios::fixed, ios::floatfield);

# double b[n] = { 4.2, 4.2, 4.2, 4.2, 4.2 };

# double A[n][n];

# double C[n][n] = { {0.2,0,0.2,0,0},

# {0,0.2,0,0.2,0},

# {0.2,0,0.2,0,0.2},

# {0,0.2,0,0.2,0},

# {0,0,0.2,0,0.2} };

# 

# double D[n][n] = { {2.33,0.81,0.67,0.92,-0.53},

# {-0.53,2.33,0.81,0.67,0.92},

# {0.92,-0.53,2.33,0.81,0.67},

# {0.67,0.92,-0.53,2.33,0.81},

# {0.81,0.67,0.92,-0.53,2.33} };

# 

# for (int i = 0; i < n; i++)

# for (int j = 0; j < n; j++)

# A[i][j] = 29 \* C[i][j] + D[i][j];

# int choice;

# while (true)

# {

# cout << "1. Метод Гаусса (схема единственного деления)\n";

# cout << "2. Метод Гаусса (cчема частичного выбора)\n";

# cout << "3. Метод Гаусса (cхема полного выбора)\n";

# cout << "4. Выход\n";

# cin >> choice;

# switch (choice)

# {

# case 1:

# {

# double\* x = gauss1(A, b, n);

# for (int i = 0; i < n; i++)

# {

# cout << x[i] << "\n";

# }

# cout << "\n";

# return 0;

# break;

# }

# case 2:

# {

# double\* x = gauss2(A, b, n);

# for (int i = 0; i < n; i++)

# {

# cout << x[i] << "\n";

# }

# cout << "\n";

# return 0;

# break;

# }

# case 3: {

# double\* x = gauss3(A, b, n);

# for (int i = 0; i < n; i++)

# {

# cout << x[i] << "\n";

# }

# cout << "\n";

# return 0;

# break;

# }

# case 4:

# {

# return 0;

# }

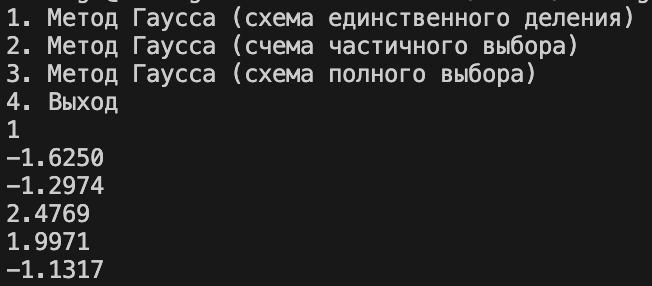
# }

# }

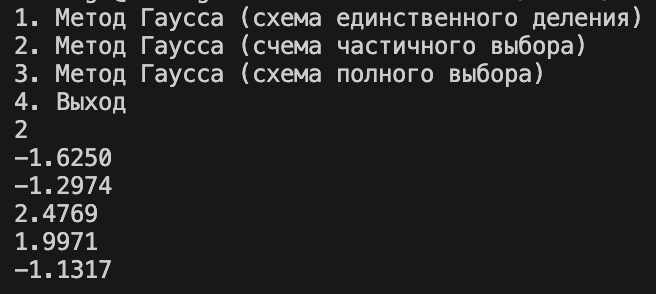
# }

# **ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ**

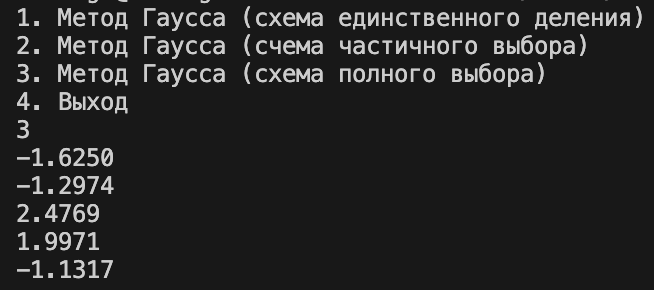
Метод Гаусса (схема единственного деления):



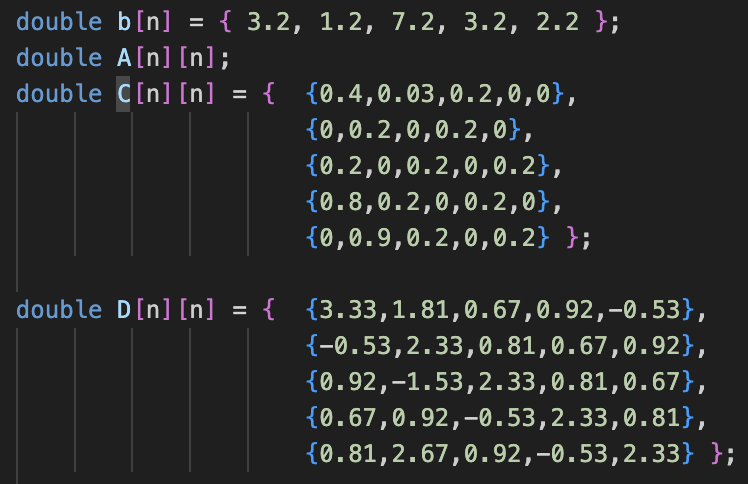
Метод Гаусса (схема частичного выбора):



Метод Гаусса (схема полного выбора):



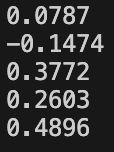
**Тестовый пример 1**

****

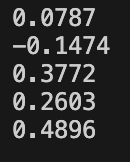
Должны были получится следующие корни: 0.0787, -0.1474, 0.3772, 0.2603, 0.4896

Согласно результатам программы:

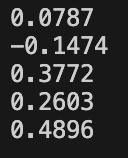
Метод Гаусса (схема единственного деления):



Метод Гаусса (схема частичного выбора):

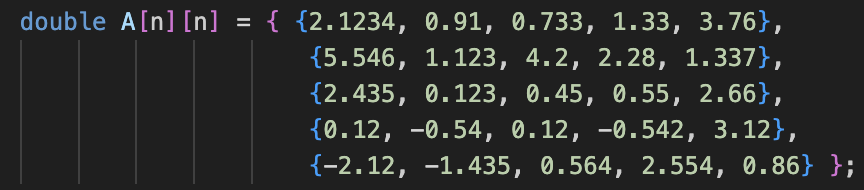


Метод Гаусса (схема полного выбора):



Что соответствует ожидаемым корням.

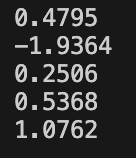
**Тестовый пример 2**

****

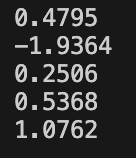
Должны были получится следующие корни: 0.4795, -1,9364,0.2506,0.5398,1.0762

Согласно результатам программы:

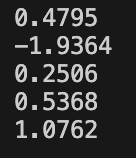
Метод Гаусса (схема единственного деления):



Метод Гаусса (схема частичного выбора):

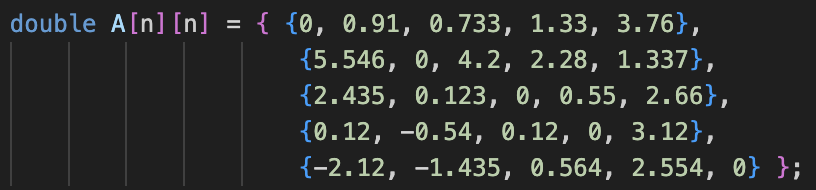


Метод Гаусса (схема полного выбора):



Что соответствует ожидаемым корням.

**Тестовый пример 3**

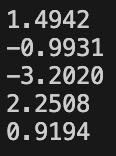


Результат выполнения программы:

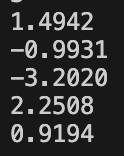
Метод Гаусса (схема единственного деления):



Метод Гаусса (схема частичного выбора):



Метод Гаусса (схема полного выбора):



# **Выводы**

В ходе выполнения данной лабораторной работы был применён

метод Гаусса по схеме единственного деления, метод Гаусса с выбором

главного элемента по столбцу (схема частичного выбора) и метод Гаусса с

выбором главного элемента по всей матрице (схема полного выбора) для

решения системы линейных уравнений, также были составлены

алгоритмы и созданы реализации соответствующих программ на языке

С++ для решения поставленной задачи.