**Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»**

Лабораторная работа №1

по вычислительной математике

Вариант: 21

Выполнил: Носов Михаил Александрович

Группа: Р3212

Принял(а)

Преподаватель: Малышева Татьяна Алексеевна

САНКТ-ПЕТЕРБУРГ

2021

Цель работы

Цель работы: создать программу, реализующую метод решения систем линейных алгебраических уравнений

1. № варианта определяется как номер в списке группы согласно ИСУ.

2. В программе численный метод должен быть реализован в виде отдельной

подпрограммы или класса, в который исходные данные передаются в качестве

параметров, выходные - тоже (либо возвращаемое значение).

3. Размерность матрицы n<=20 (задается из файла или с клавиатуры - по выбору

конечного пользователя).

4. Должна быть реализована возможность ввода коэффициентов матрицы, как с

клавиатуры, так и из файла (по выбору конечного пользователя).

Обязательно: Тестовые данные на матрице большого размера (5\*5 / 6\*6...) + в

отчёт с решением.

Для прямых методов должно быть реализовано:

· Вычисление определителя

· Вывод треугольной матрицы (включая преобразованный столбец В)

· Вывод вектора неизвестных: 𝑥1,𝑥2,…,𝑥𝑛

· Вывод вектора невязок: 𝑟1,𝑟,…,𝑟𝑛

Описание метода

В соответствии вариантом (Вариант 21), задание сформулировано следующим

образом: написать программу, решающую систему линейных алгебраических

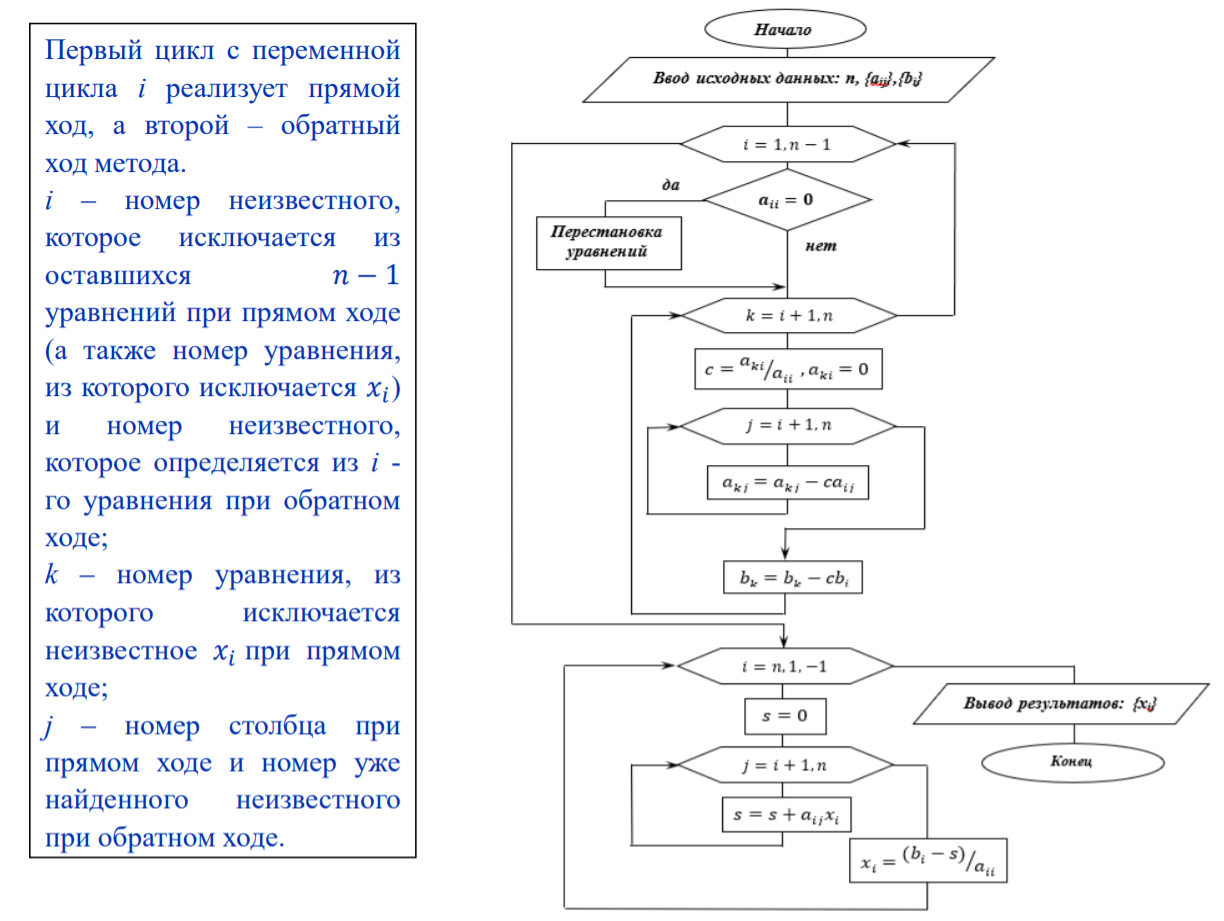
уравнений методом Гаусса.

Он основан на приведении матрицы системы к треугольному виду так, чтобы ниже ее главной диагонали находились только нулевые элементы.

Прямой ход метода Гаусса состоит в последовательном исключении неизвестных из уравнений системы. Сначала с помощью первого уравнения исключается 𝑥1 из всех последующих уравнений системы. Затем с помощью второго уравнения исключается 𝑥2 из третьего и всех последующих уравнений и т.д. Этот процесс продолжается до тех пор, пока в левой части последнего (n-го) уравнения не останется лишь один член с неизвестным 𝑥𝑛 , т. е. матрица системы будет приведена к треугольному виду.

Обратный ход метода Гаусса состоит в последовательном вычислении искомых неизвестных: решая последнее уравнение, находим единственное в этом уравнении неизвестное 𝑥𝑛 . Далее, используя это значение, из предыдущего уравнения вычисляем 𝑥𝑛−1 и т. д. Последним найдем 𝑥1 из первого уравнения.

Блок-схема численного метода

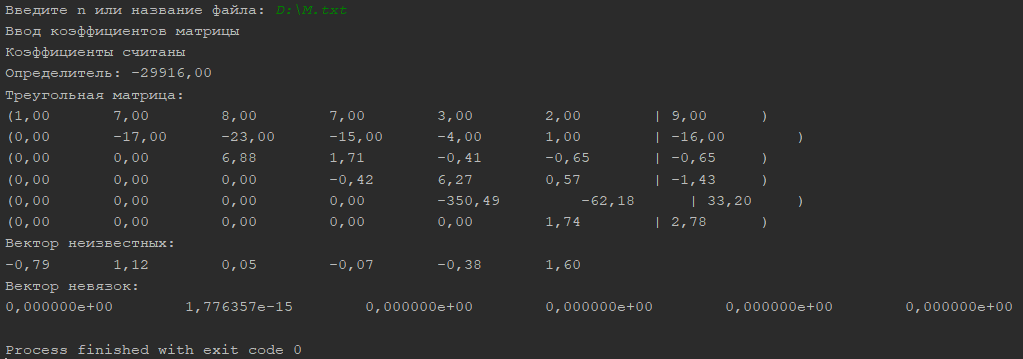


Листинг программы

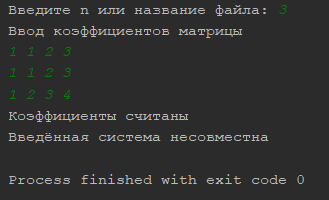
private static Answer calculate(int n, double[][] a, double[] b) throws badMatrixException {  
 double[][] aR = a; //Для вычисления невязкости  
 double[] bR = b;  
 double[] x = new double[n];  
 int swapCounter = 0;  
 for(int i = 0; i < n - 1; i++){  
 if(a[i][i] == 0) { //если "первый" коэффициент в строке равен нулю  
 boolean notZeroExists = false;  
 for (int j = i + 1; j < n; j++) { //ищем строку с ненулевым  
 if (a[j][i] != 0){  
 a = *swapA*(a, i, j); //меняем местами строки  
 b = *swapB*(b, i, j);  
 notZeroExists = true;  
 break;  
 }  
 }  
 if(!notZeroExists){  
 throw new badMatrixException(); //если не нашли, значит система несовмеснтна, кидаем ошибку ввода  
 }  
 swapCounter++; //увеличиваем счётчик замен  
 }  
 for(int k = i + 1; k < n; k++){  
 double c = a[k][i]/a[i][i]; //коэффициент для вычитания строк  
 a[k][i] = 0;  
 for(int j = i + 1; j < n; j++){  
 a[k][j] -= c\*a[i][j]; //вычитание строк  
 }  
 b[k] -= c\*b[i];  
 }  
 }  
 if(a[n-1][n-1] == 0){ // проверяем, что единственный коэффициент в последней строке не ноль  
 throw new badMatrixException();  
 }  
 x[n-1] = b[n-1]/a[n-1][n-1]; //вычисляем неизвестную в последней строке  
 for(int i = n - 2; i >= 0; i--){  
 double s = 0;  
 for(int j = i + 1; j < n; j++){  
 s += a[i][j]\*x[j]; //сумма очередной строки  
 }  
 x[i] = (b[i] - s)/a[i][i]; //единственная оставшаяся неизвестная в строке  
 }  
 double det = Math.*pow*(-1, swapCounter); //Вычисление детерминанта  
 for(int i = 0; i < n; i++){  
 det \*= a[i][i];  
 }  
 double[] r = new double[n]; // невязка  
 for(int i = 0; i < n; i++){  
 for(int j = 0; j < n; j++){  
 r[i] += aR[i][j]\*x[j];  
 }  
 r[i] -= bR[i];  
 }  
 Answer answer = new Answer(det, n, a, b, x, r);  
 return answer;  
}

Примеры работы программы

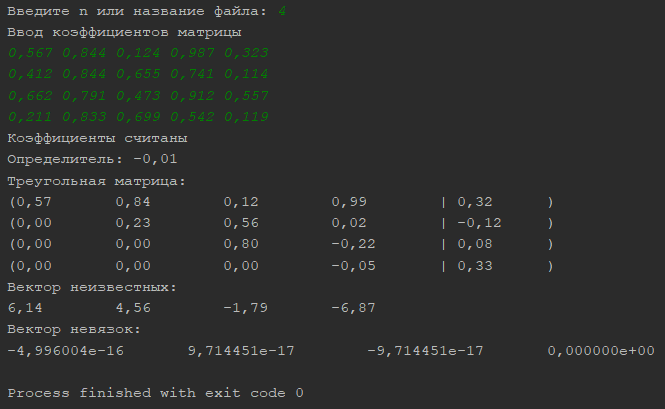
Пример №1



Пример №2



Пример №3



Вывод: решать системы линейных алгебраических уравнений приятнее программой, чем своими руками, а метод Гаусса хорош в обоих случаях.