**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Лабораторная работа №2 по курсу “ВМА”

“Метод левой прогонки”

Вариант №3

Выполнил: Ёда Никита

3 курс, 6(а) группа

Преподаватель: Будник А.М.

2023

**Постановка задачи**

В данной лабораторной работе требуется решить систему линейных алгебраических уравнений Ах=b, используя метод левой прогонки.

План:

- Решить СЛАУ методом левой прогонки

- Найти вектор невязки

**Алгоритм решения**

Введём обозначения:

n = A.length

Вычисление неизвестных будем вести по следующим формулам:

=

=

= ,

=

Решение системы:

**Листинг программы**

public static double[] leftSweepMet(double[][] A, double[] b) {

double[] diagonalC = new double[A.length];

double[] diagonalA = new double[A.length];

double[] diagonalB = new double[A.length];

for (int i = 0; i < A.length; i++) {

diagonalC[i] = A[i][i];

}

for (int i = 0; i < A.length - 1; i++) {

diagonalA[i + 1] = -A[i + 1][i];

diagonalB[i] = -A[i][i + 1];

}

System.out.println("\nДиагональ A:\n\t" + Arrays.toString(diagonalA));

System.out.println("\nДиагональ B:\n\t" + Arrays.toString(diagonalB));

System.out.println("\nДиагональ C:\n\t" + Arrays.toString(diagonalC));

double[] alpha = new double[A.length];

double[] beta = new double[A.length];

alpha[A.length - 1] = diagonalA[A.length - 1] / diagonalC[A.length - 1];

beta[A.length - 1] = b[A.length - 1] / diagonalC[A.length - 1];

for (int i = A.length - 2; i >= 1 ; i--) {

alpha[i] = diagonalA[i] / (diagonalC[i] - diagonalB[i] \* alpha[i + 1]);

beta[i] = (beta[i + 1] \* diagonalB[i] + b[i]) / (diagonalC[i] - diagonalB[i] \* alpha[i + 1]);

}

System.out.println("\nКоэффициенты Alpha:\n\t" + Arrays.toString(alpha));

System.out.println("\nКоэффициенты Beta:\n\t" + Arrays.toString(beta));

double[] massX = new double[A.length];

massX[0] = (b[0] + beta[1] \* diagonalB[0]) / (diagonalC[0] - alpha[1] \* diagonalB[0]);

for (int i = 1; i < A.length; i++) {

massX[i] = alpha[i] \* massX[i - 1] + beta[i];

}

System.out.println("\nВектор решений X:\n\t" + Arrays.toString(massX));

return massX;

}

**Результаты и его анализ**

*Результат:*

Диагональ A:

[0.0, -0.0944, -0.0, -0.0, -0.0182]

Диагональ B:

[0.0908, -0.0, 0.2541, 0.0363, 0.0]

Диагональ C:

[0.6897, 1.0799, 0.8676, 0.8531, 1.0164]

Коэффициенты Alpha:

[0.0, -0.08741550143531808, -0.0, -0.0, -0.01790633608815427]

Коэффициенты Beta:

[0.0, -0.806463561440874, 3.1056186737517266, -2.0187533988705293, 2.9060409287682014]

Вектор решений X:

[0.9862905573831378, -0.8926806450754404, 3.1056186737517266, -2.0187533988705293, 2.942189405607481]

Вектор невязки:

[1.1102230246251565E-16, 0.0, 0.0, 0.0, -4.440892098500626E-16]

*Анализ:*

Метод прогонки является прямым методом, его точность сопоставима с точностью метода Гаусса (кубические нормы векторов невязки имеют порядок 10^(-16)). Количество операций -порядка 8n, но метод прогонки можно применять только на трехдиагональных матрицах или на матрицах с диагональным преобладанием.

*Сравнение методов:*  
Точность: Оба метода обеспечивают приближенное решение системы линейных уравнений. Однако, точность решения может зависеть от особенностей конкретной системы и ее численных свойств. В целом, метод Гаусса с выбором главного элемента по столбцам обеспечивает точность, сравнимую с методом левой прогонки. Однако, в редких случаях, когда система обладает особыми свойствами (например, близкими к вырожденности или плохо обусловленности), метод Гаусса с выбором главного элемента может быть более устойчивым и точным.

Экономичность: Под экономичностью в контексте решения СЛАУ понимается затраты на вычислительные ресурсы (время и память). В этом аспекте метод левой прогонки имеет преимущество. Он требует меньшего количества операций и памяти по сравнению с методом Гаусса с выбором главного элемента. Метод левой прогонки особенно полезен для разреженных матриц, когда большинство элементов матрицы равны нулю.