



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»  
(ДВФУ)**

---

## **ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информатики, математического и компьютерного  
моделирования**

### **ОТЧЕТ**

к лабораторной работе №6 по дисциплине  
«Вычислительная математика»

Направление подготовки  
01.03.02 «Прикладная математика и информатика»

Выполнил студент  
гр. Б9119-01.03.02 систпро  
Нагорнов С.С.

\_\_\_\_\_  
(ФИО)

\_\_\_\_\_  
(Подпись)

« 15 » \_\_\_\_\_ января 2021 г.

г. Владивосток  
2021

# Содержание

<b>Метод монотонной прогонки</b>	<b>2</b>
Постановка задачи . . . . .	2
Решение . . . . .	2
<b>Приложения</b>	<b>3</b>
<b>Вывод</b>	<b>4</b>

# Метод монотонной прогонки

## Постановка задачи

Требуется найти решение системы линейных алгебраических уравнений (СЛАУ). При этом СЛАУ имеет специальный вид:

$$A_k x_{k-1} + B_k x_k + C_k x_{k+1} = F_k, \quad k = 1, \dots, N \quad A_1 = C_N = 0$$

## Решение

Будем решать данную СЛАУ:

$$A = \begin{pmatrix} 2 & 2 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \\ 0 & 2 & 1 \end{pmatrix} \quad \vec{b} = \begin{pmatrix} 2 \\ 4 \\ 4 \\ 3 \end{pmatrix}$$

Для начала найдем все  $\alpha$  и  $\beta$ , исходя из формул:

$$\alpha_2 = -\frac{C_1}{B_1} \quad \beta = \frac{F_1}{B_1}$$
$$\alpha_k = -\frac{C_k}{B_k + A_k \alpha_k} \quad \beta = \frac{F_k - A_k \beta_k}{B_k + A_k \alpha_k}$$

Предположим, что:

$$x_{k-1} = \alpha_k x_k + \beta_k$$

Тогда решения будем искать по следующим формулам:

$$x_k = \alpha_k x_{k-1} + \beta_k$$

Чтобы найти переменную в  $N$  позиции будем использовать данную формулу:

$$x_N = \frac{F_N - A_N \beta_N}{A_N \alpha_N + B_N}$$

Следуя из всех формул выше, мы можем найти ответ СЛАУ:

$$\vec{x} = \begin{pmatrix} 2 \\ -1 \\ 4 \\ -3 \end{pmatrix}$$

## Приложения

```
1  low = [0, 1, 1, 2]
2  main = [2, 2, 2, 2]
3  high = [1, 1, 1, 0]
4  result = [3, 4, 4, 2]
5
6  alpha = [0]
7  betta = [0]
8  alpha.append(-high[0] / main[0])
9  betta.append(result[0] / main[0])
10
11 n = 4
12 i = 1
13
14 while i < n:
15     alpha.append(-high[i] / (main[i] + low[i] * alpha[i]))
16     betta.append((-low[i] * betta[i] + result[i]) / (main[i] + low
[i] * alpha[i]))
17     i += 1
18
19 print(alpha)
20 print(betta)
21
22 x = [(-low[n - 1] * betta[n - 1] + result[n - 1]) / (main[n - 1]
+ low[n - 1] * alpha[n - 1])]
23 i = n - 1
24 x_count = 0
25
26 while i != 0:
27     x.append(alpha[i] * x[x_count] + betta[i])
28     i -= 1
29     x_count += 1
30
31 x.reverse()
32 print(x)
33
```

---

### Листинг 1: Реализация метода прогонки

## Вывод

В данной лабораторной работе было произведено вычисление решения СЛАУ с помощью метода прогонки.