



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

**Лабораторная работа № 5.2**  
**по курсу «Численные методы линейной алгебры»**  
**«Изучение сходимости метода Зейделя»**

Студентка группы ИУ9-72Б Самохвалова П. С.

Преподаватель Посевин Д. П.

*Москва 2023*

# 1 Цель работы

Реализовать метод Зейделя.

## 2 Задание

- Реализовать метод Зейделя.
- Сравнить число итераций, необходимых для сходимости метода Якоби и метода Зейделя.

## 3 Практическая реализация

Исходный код программы представлен в листинге 1.

Листинг 1: Реализация метода Зейделя

```
1 from num_methods import *
2 import random
3
4
5 def uniform_norm_vec(vec):
6     n = len(vec)
7     norm = abs(vec[0])
8     for i in range(1, n):
9         v = abs(vec[i])
10        if v > norm:
11            norm = v
12    return norm
13
14
15 def increase_diagonal_elements_to_diagonal_predominance(a):
16     n = len(a)
17     for i in range(n):
18         s = 0
19         for j in range(n):
20             if i != j:
21                 s += abs(a[i][j])
22         a[i][i] = s + 10
23    return a
24
25
26 def generate_matrix(n, v1, v2):
```

```

27     return [[random.uniform(v1, v2) for i in range(n)] for j in range(n)
28 ]
29
30 def jacobi_method(a, f, epsilon):
31     n = len(a)
32     k = 0
33     x_old = [0] * n
34     while True:
35         x = [0] * n
36         for i in range(n):
37             s = 0
38             for j in range(n):
39                 if i != j:
40                     s += a[i][j] * x_old[j]
41             x[i] = (f[i] - s) / a[i][i]
42         k += 1
43         norm_sub = uniform_norm_vec(sub_vec(x, x_old))
44         if norm_sub < epsilon:
45             break
46         x_old = x
47     return x, k
48
49
50 def zeidel_method(a, f, epsilon):
51     n = len(a)
52     k = 0
53     x_old = [0] * n
54     while True:
55         x = [0] * n
56         for i in range(n):
57             s1 = 0
58             s2 = 0
59             for j in range(i):
60                 if i != j:
61                     s1 += a[i][j] * x[j]
62             for j in range(i + 1, n):
63                 if i != j:
64                     s2 += a[i][j] * x_old[j]
65             x[i] = (f[i] - s1 - s2) / a[i][i]
66         k += 1
67         norm_sub = uniform_norm_vec(sub_vec(x, x_old))
68         if norm_sub < epsilon:
69             break
70         x_old = x
71     return x, k

```

```

72
73
74 n = 5
75 a = generate_matrix(n, -10, 10)
76 x = [i for i in range(1, n + 1)]
77 a = increase_diagonal_elements_to_diagonal_predominance(a)
78 f = mult_matr_vec(a, x)
79 x_jacobi, k_jacobi = jacobi_method(a, f, 0.001)
80 x_zeidel, k_zeidel = zeidel_method(a, f, 0.001)
81
82 print("x by Jacobi method")
83 print(x_jacobi)
84 print("k =", k_jacobi)
85 print()
86
87 print("x by Zeidel method")
88 print(x_zeidel)
89 print("k =", k_zeidel)
90 print()

```

## 4 Результаты

Результаты работы программы представлен на рисунках 1 – 2.

```

x by Jacobi method
[0.9999004243329529, 2.000146506438567, 3.0000554726138526, 4.000098205186111, 5.00007279660995]
k = 11

```

Рис. 1 — Вектор x и число итераций метода Якоби

```

x by Zeidel method
[1.0001120597842992, 1.9999225787582926, 2.999947954305803, 4.000019123425153, 4.9999908519255145]
k = 7

```

Рис. 2 — Вектор x и число итераций метода Зейделя

## 5 Выводы

В результате выполнения лабораторной работы был реализован метод Зейделя.