



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 № 5.1
по курсу «Численные методы линейной алгебры»
«Изучение сходимости метода Якоби»

Студентка группы ИУ9-72Б Самохвалова П. С.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель работы

Реализовать метод Якоби.

2 Задание

- Реализовать метод Якоби.
- Ввести критерий остановки итерационного процесса используя равномерную норму.
- Проверить решение путем сравнения с решением любым методом Гаусса.
- Проверить выполнение условия диагонального преобладания.
- Используя согласованную векторную и матричную нормы проверить выполнение условия: $\|P\| \leq q < 1$.

3 Практическая реализация

Исходный код программы представлен в листинге 1.

Листинг 1: Реализация метода Якоби

```
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
48
49
50 def p_norm(a):
51     n = len(a)
52     ms = 0
53     for i in range(n):
54         ss = 0
55         for j in range(n):
56             if i != j:
57                 ss += abs(a[i][j] / a[i][i])
58         if ss > ms:
59             ms = ss
60     return ms
61
62
63 n = 5

```

```

64 a = generate_matrix(n, -10, 10)
65 x = [i for i in range(1, n + 1)]
66 a = increase_diagonal_elements_to_diagonal_predominance(a)
67 f = mult_matr_vec(a, x)
68 x_jacobi = jacobi_method(a, f, 0.001)
69 x_gauss = gauss_method(a, f)
70
71 print("x by Jacobi method")
72 print(x_jacobi)
73 print()
74
75 print("x by Gauss method")
76 print(x_gauss)
77 norm_p = p_norm(a)
78 print()
79
80 print("Check ||P|| < 1")
81 print(norm_p < 1)

```

4 Результаты

Результаты работы программы представлены на рисунках 1 – 3.

```

x by Jacobi method
[0.9997494520393246, 1.9999197031648264, 2.9995581173837076, 3.999712886266108, 4.999819596118547]

```

Рис. 1 — Вектор x , найденный методом Якоби

```

x by Gauss method
[1.0000000000000002, 2.0, 2.9999999999999996, 4.0, 4.999999999999999]

```

Рис. 2 — Вектор x , найденный методом Гаусса

```

Check ||P|| < 1
True

```

Рис. 3 — Проверка $\|P\| < 1$

5 Выводы

В результате выполнения лабораторной работы был реализован метод Якоби.