



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»

Летучка № 1
по курсу «Численные методы линейной алгебры»
«Реализация метода прогонки»

Студент группы ИУ9-72Б Виленский С. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Реализовать и протестировать метод прогонки для решения СЛАУ с трех-диагональной матрицей коэффициентов.

2 Результаты

Исходный код программы представлен в листингах 1- 2.

Листинг 1 — Реализация метода прогонки

```
1 using LinearAlgebra
2 using Random
3
4 function generate_tridiagonal_matrix(n)
5     A = zeros(Float64, n, n)
6
7     for i in 1:n
8         if i > 1
9             A[i, i-1] = rand(-100.0:0.001:100.0)
10        end
11        if i < n
12            A[i, i+1] = rand(-100.0:0.001:100.0)
13        end
14        A[i, i] = abs(A[i, max(i-1, 1)]) + abs(A[i, min(i+1, n)]) + rand
15        (1.0:0.001:10.0)
16    end
17    return A
18 end
19
20 function run_algorithm_matrix(A, f)
21     n = size(A, 1)
22
23     a = zeros(Float64, n)
24     b = zeros(Float64, n)
25     c = zeros(Float64, n)
26
27     for i in 1:n
28         b[i] = A[i, i]
29         if i > 1
30             a[i] = A[i, i-1]
31         end
32         if i < n
33             c[i] = A[i, i+1]
34         end
35     end
36
37     p = zeros(Float64, n)
38     q = zeros(Float64, n)
39
40     p[1] = c[1] / b[1]
41     q[1] = d[1] / b[1]
```

Листинг 2 — Реализация метода прогонки

```
1   for i in 2:n-1
2       denominator = b[i] - a[i] * p[i-1]
3       p[i] = c[i] / denominator
4       q[i] = (f[i] - a[i] * q[i-1]) / denominator
5   end
6
7   q[n] = (f[n] - a[n] * q[n-1]) / (b[n] - a[n] * p[n-1])
8
9   x = zeros(Float64, n)
10  x[n] = q[n]
11
12  for i in n-1:-1:1
13      x[i] = q[i] - p[i] * x[i+1]
14  end
15
16  return x
17 end
18
19 function compute_error(A, x, f)
20     return norm(f - A * x, 2)
21 end
22
23 N = 100
24
25 A = generate_tridiagonal_matrix(N)
26 x = rand(-100.0:0.001:100.0, N)
27 f = A * x
28 println(compute_error(A, x, f))
```

Результат запуска представлен в листинге 3.

Листинг 3 — Вывод программы

```
1 0.000152
```

3 Выводы

Метод прогонки для нахождения решения СЛАУ с трехдиагональной матрицей коэффициентов является достаточно точным и имеет линейную сложность времени исполнения.