



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное 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 № 3**  
**по курсу «Численные методы линейной алгебры»**  
**«Разложение Холецкого»**

Студент группы ИУ9-72Б Шемякин В.А.

Преподаватель Посевин Д. П.

*Moskva 2025*

# 1 Задание

Реализовать разложение Холецкого.

# 2 Результаты

Исходный код программы представлен в листинге 1.

```
1 using LinearAlgebra
2 using Plots
3
4 function generate_symmetric_matrix(N)
5     A = rand(N, N)
6     return A * A'
7 end
8
9 function cholesky(A)
10    N = size(A, 1)
11    for I = 1:N
12        S = A[I, I]
13        for IP = 1:I-1
14            S -= A[I, IP] * A[I, IP]
15        end
16        A[I, I] = sqrt(S)
17        for J = I+1:N
18            S = A[J, I]
19            for IP = 1:I-1
20                S -= A[I, IP] * A[J, IP]
21            end
22            A[J, I] = S / A[I, I]
23        end
24    end
25    return A
26 end
27
28 function cholesky_solve(A::Matrix{Float64}, b::Vector{Float64})
29    N = length(b)
30    L = cholesky(copy(A))
31    y = zeros(N)
32    for i in 1:N
33        y[i] = b[i] - sum(L[i, 1:i-1] .* y[1:i-1])
34        y[i] /= L[i, i]
35    end
```

```

36     x = zeros(N)
37     for i in N:-1:1
38         x[i] = y[i] - sum(L[i+1:N, i] .* x[i+1:N])
39         x[i] /= L[i, i]
40     end
41     return x
42 end
43
44 function gauss_solve_vector(A::Matrix{Float64}, b::Vector{Float64})
45     n = length(b)
46     M = copy(A)
47     r = copy(b)
48     for i in 1:n
49         if M[i, i] == 0
50             throw(ErrorException(
51                         ."))
52         end
53         for k in (i+1):n
54             if M[k, i] != 0
55                 factor = M[k, i] / M[i, i]
56                 for j in i:n
57                     M[k, j] -= factor * M[i, j]
58                 end
59                 r[k] -= factor * r[i]
60             end
61         end
62         x = zeros(n)
63         for i in n:-1:1
64             x[i] = r[i]
65             for k in (i+1):n
66                 x[i] -= M[i, k] * x[k]
67             end
68             if M[i, i] == 0
69                 throw(ErrorException(
70                         ."))
71             end
72             x[i] /= M[i, i]
73         end
74     end
75
76 sizes = [3, 10, 100, 300]

```

```

77 cholesky_times = Float64 []
78 gauss_times = Float64 []
79
80 for n in sizes
81     A = generate_symmetric_matrix(n)
82     b = rand(n)
83     A_copy = copy(A)
84     push!(cholesky_times, @elapsed cholesky_solve(copy(A), b))
85     push!(gauss_times, @elapsed gauss_solve_vector(A, b))
86 end
87
88 plot(sizes, cholesky_times, label =
89       " ", xlabel = "n",
90       ylabel = " ", legend = :topright)
91 plot!(sizes, gauss_times, label =
92       " ", xlabel = "n",
93       ylabel = " ", legend = :
94       topright)

```

Результат представлен на рисунке 1.

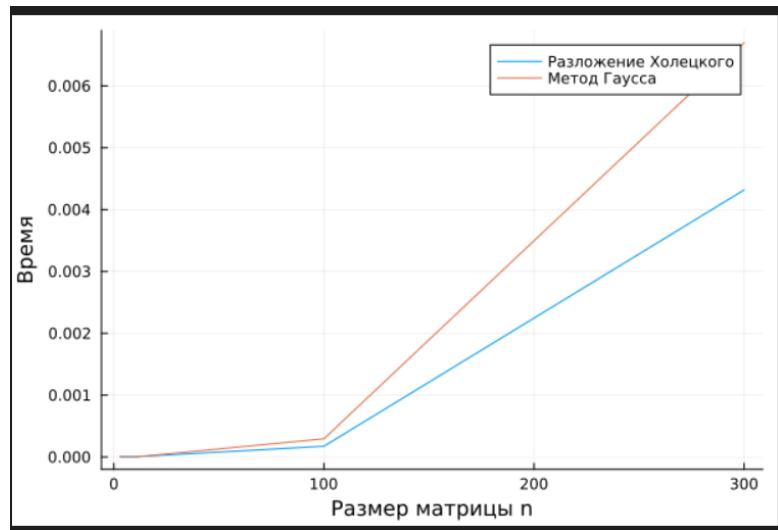


Рис. 1 — Результат