**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»

«Вычисление собственных значений и собственных векторов симметричной матрицы методом А.М. Данилевского»

Студент группы ИУ9-72Б Виленский С. Д.

Преподаватель Посевин Д. П.

*Москва 2024*

# 1 Задание

Реализовать метод поиска собственных значений действительной симметричной матрицы A размером 4x4. Проверить корректность вычисления собственных значений по теореме Виета. Проверить выполнение условий теоремы Гершгорина о принадлежности собственных значений соответствующим объединениям кругов Гершгорина. Вычислить собственные вектора и проверить выполнение условия ортогональности собственных векторов. Проверить решение на матрице приведенной в презентации. Продемонстрировать работу приложения для произвольных симметричных матриц размером n x n с учетом выполнения пунктов приведенных выше.

# 2 Результаты

Исходный код программы представлен в листингах 1- 3.

Листинг 1 — Код

```
using LinearAlgebra
using PolynomialRoots

function I_matrix(n::Int)::Matrix{Float64}
    I = Matrix{Float64}(zeros(n, n))
    for i in 1:n
        I[i, i] = 1.0
    end
    return I
end

function danilevsky_method(A::Matrix{Float64})
    n = size(A, 1)

    B_i = Vector{Matrix{Float64}}(undef, n - 1)
    D = copy(A)

    for k in n:-1:2
        B_inv = I_matrix(n)
        B_inv[k - 1, :] = D[k, :]
        B_i[n - k + 1] = inv(B_inv)

        D = B_inv * D * B_i[n - k + 1]
    end
    P = D[1, :]
    eigen_vals = real.(roots(push!(-reverse(P), 1)))

    B = I_matrix(n)
    for B_ in B_i
```

Листинг 2 — Код

```
        B *= B_
    end

    y_i = [[eigen_val ^ i for i in (n-1):-1:0] for eigen_val in eigen_vals]

    x_i = [B * y for y in y_i]

    return eigen_vals, normalize.(x_i)
end

function check_by_Viet(A::Matrix{Float64}, eigen_vals::Vector{Float64})::Float64
    return abs(tr(A) - sum(eigen_vals))
end

function check_by_Gershgorin(A::Matrix{Float64}, eigen_vals::Vector{Float64})::Bool
    n = size(A, 1)

    start_union = undef
    stop_union = undef
    for i in 1:n
        diag_elem = A[i, i]
        line_sum = sum(abs.(A[i, :])) - abs(diag_elem)

        start = diag_elem - line_sum
        if start_union == undef || start < start_union
            start_union = start
        end

        stop = diag_elem + line_sum
        if stop_union == undef || stop > start_union
            stop_union = stop
        end
    end

    return all(start_union <= eigen_val <= stop_union for eigen_val in eigen_vals)
end

function check_ortogonal(eigen_vectors::Vector{Vector{Float64}})::Bool
    n = size(eigen_vectors, 1)
    for i in 1:(n-1)
        for j in (i+1):n
            if abs(eigen_vectors[i]'eigen_vectors[j]) > 1e-5
                return false
            end
        end
    end
    return true
end

A = [
    2.2 1.0 0.5 2.0;
```

Листинг 3 — Код

```
    1.0  1.3  2.0  1.0;
    0.5  2.0  0.5  1.6;
    2.0  1.0  1.6  2.0
]

eigen_vals, eigen_vectors = danilevsky_method(A)
println("Eigenvalues: ", eigen_vals)
error_viet = check_by_Viet(A, eigen_vals)
println("Error in calculating eigenvalues using Vieta's theorem: ", error_viet)
check = check_by_Gershgorin(A, eigen_vals)
println("Verification of Gershgorin's theorem: ", check)
println("Eigenvectors:")
for eigen_vector in eigen_vectors
    println(eigen_vector)
end
check_vects = check_ortogonal(eigen_vectors)
println("Orthogonality of eigenvectors: ", check_vects)

n = 5
A = Matrix{Float64}(Symmetric(rand(-10.0:0.1:10.0,n,n)))
eigen_vals, eigen_vectors = danilevsky_method(A)
println("Eigenvalues: ", eigen_vals)
error_viet = check_by_Viet(A, eigen_vals)
println("Error in calculating eigenvalues using Vieta's theorem: ", error_viet)
check = check_by_Gershgorin(A, eigen_vals)
println("Verification of Gershgorin's theorem: ", check)
println("Eigenvectors:")
for eigen_vector in eigen_vectors
    println(eigen_vector)
end
check_vects = check_ortogonal(eigen_vectors)
println("Orthogonality of eigenvectors: ", check_vects)
```

Результат работы программы представлен в листингах 4- 5.

Листинг 4 — Результат работы программы

```
Eigenvalues: [5.652032331764589, -1.420086593950619, 1.5454183350534156, 0.22263592713261507]
Error in calculating eigenvalues using Vieta's theorem: 1.7763568394002505e-15
Verification of Gershgorin's theorem: true
Eigenvectors:
[0.5317360693095499, 0.44619412190869223, 0.40881553418500616, 0.5924841071103837]
[-0.2220428365454722, 0.5159103236551117, -0.7572742312071333, 0.3332705439047439]
[0.62892976467108, -0.5725742255591189, -0.48565379676310105, 0.2018576157239048]
[-0.5219205710113896, -0.45486932161400195, 0.1534470183752563, 0.705086399217363]
Orthogonality of eigenvectors: true
```

Листинг 5 — Результат работы программы

```
Eigenvalues: [18.176061276778213, -8.924899361398401, 5.39378895734575, 1.874589613387166, -0.5195404861127171]
Error in calculating eigenvalues using Vieta's theorem: 1.2434497875801753e-14
Verification of Gershgorin's theorem: false
Eigenvectors:
[0.3996021709709015, -0.665653553721814, 0.5558056083971021, -0.17477433349077834, 0.2403279205586481]
[-0.5492894758466208, -0.10766023132451748, 0.24684288967053, 0.6180553989016193, 0.4937271088597009]
[-0.5156957615883632, -0.37662692283972776, -0.392079500533507, -0.5966956605656002, 0.28712017761845604]
[-0.2019522237982415, 0.5759766479137343, 0.5948706624020939, -0.48096669550969695, 0.20558728848053778]
[0.48153126322413037, 0.2678015740412564, -0.350073240577105, 0.009315724023380937, 0.7574773283713271]
Orthogonality of eigenvectors: true
```

# 3 Выводы

Проанализировав графики зависимостей ошибок разных методов от диагонального доминирования матриц можно сделать вывод о том, что самым оптимальным является модификация метода Гаусса перестановкой по строкам и столбцам, модификации метода Гаусса перестановкой по строкам или по столбцам относительно имеет близкую погрешность и классический метод Гаусса среди прочих имеет наибольшую относительную ошибку.