Лабораторная работа №4.

Евдокимов Иван Андреевич. НФИбд-01-20

25 ноября, 2023, Москва, Россия

Российский Университет Дружбы Народов

Цель лабораторной работы

Цель лабораторной работы

Основной целью работы является изучение возможностей специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.

Процесс выполнения лабораторной работы

Часть 1: Произведение векторов

1. Задайте вектор v. Умножьте вектор v скалярно сам на себя и сохраните результат в dot_v.

```
using LinearAlgebra
v = [1 2 3 4 5 6 7 8 9]
dot_v = dot(v, v)

println("Вектор v:")
display(v)

println("Вектор dot_v:")
display(dot_v)
```

```
Вектор v:
1×9 Matrix{Int64}:
 1  2  3  4  5  6  7  8  9
Вектор dot_v:
285
```

Рис. 1: Пункт 1.1

2. Умножьте v матрично на себя (внешнее произведение), присвоив результат переменной $outer\_v$.

```
using LinearAlgebra
v = [1 2 3 4 5 6 7 8 9]
outer_v = kron(v, v)

println("Вектор v:")
display(v)

println("Матрица outer_v:")
reshape(outer_v, (9, 9))
```

```
Вектор v:
1×9 Matrix{Int64}:
 1  2  3  4  5  6  7  8  9
Матрица outer_v:

9×9 Matrix{Int64}:
 1   2   3   4   5   6   7   8   9
 2   4   6   8  10  12  14  16  18
 3   6   9  12  15  18  21  24  27
 4   8  12  16  20  24  28  32  36
 5  10  15  20  25  30  35  40  45
 6  12  18  24  30  36  42  48  54
 7  14  21  28  35  42  49  56  63
 8  16  24  32  40  48  56  64  72
 9  18  27  36  45  54  63  72  81
```

Рис. 2: Пункт 1.2

Часть 2: Системы линейных уравнений

3. Решить СЛАУ с двумя неизвестными.

$$\begin{cases} x + y = 2 \\ x - y = 3 \end{cases}, (3.1) \quad \begin{cases} x + y = 2 \\ 2x + 2y = 4 \end{cases}, (3.2) \quad \begin{cases} x + y = 2 \\ 2x + 2y = 5 \end{cases}, (3.3) \quad \begin{cases} x + y = 1 \\ 2x + 2y = 2 \\ 3x + 3y = 3 \end{cases}, (3.4) \quad \begin{cases} x + y = 2 \\ 2x + y = 1 \\ x - y = 3 \end{cases}, (3.5) \quad \begin{cases} x + y = 2 \\ 2x + y = 1 \\ 3x + 2y = 3 \end{cases}, (3.6)$$

Рис. 3: Уравнения

3.1.

```
# 3.1
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1; 1 -1]
b = [2, 3]

function proverca(A, b) # Решение СЛАУ
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
[2.5, -0.5]
```

Рис. 4: 3.1.

3.2.

```
# 3.2
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1; 2 2]
b = [2, 4]

function proverca(A, b) # Решение СЛАУ
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
Нет решения или их число бесконечное.
```

Рис. 5: 3.2.

3.3.

```
# 3.3
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1; 2 2]
b = [2, 5]

function proverca(A, b) # Решение СЛАУ
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
Нет решения или их число бесконечное.
```

Рис. 6: 3.3.

```
# 3.4
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1; 2 2; 3 3]
b = [1, 2, 3]

function proverca(A, b) # Решение СЛАУ
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
[0.4999999999999999, 0.5]
```

Рис. 7: 3.4.

3.5.

```
# 3.5
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1; 2 1; 1 -1]
b = [2, 1, 3]

function proverca(A, b) # Решение СЛАУ
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
[1.5000000000000004, -0.9999999999999997]
```

Рис. 8: 3.5.

```
# 3.6
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1; 2 1; 3 2]
b = [2, 1, 3]

function proverca(A, b) # Решение СЛАУ
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
[-0.9999999999999989, 2.9999999999999982]
```

Рис. 9: 3.6.

4.0.

4. Решить СЛАУ с тремя неизвестными.

$$\begin{cases} x + y + z = 2 \\ x - y - 2z = 3 \end{cases}, (4.1) \qquad \begin{cases} x + y + z = 2 \\ 2x + 2y - 3z = 4 \\ 3x + y + z = 1 \end{cases}, (4.2) \qquad \begin{cases} x + y + z = 1 \\ x + y + 2z = 0 \\ 2x + 2y + 3z = 1 \end{cases}, (4.3) \qquad \begin{cases} x + y + z = 1 \\ x + y + 2z = 0 \\ 2x + 2y + 3z = 0 \end{cases}, (4.4)$$

Рис. 10: Уравнения

4.1.

```
# 4.1
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1 1; 1 -1 -2]
b = [2, 3]

function proverca(A, b)
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
[2.2142857142857144, 0.35714285714285704, -0.5714285714285712]
```

Рис. 11: 4.1.

4.2.

```
# 4.2
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1 1; 2 2 -3; 3 1 1]
b = [2, 4, 1]

function proverca(A, b)
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
[-0.5, 2.5, 0.0]
```

Рис. 12: 4.2.

4.3.

```
# 4.3
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1 1; 1 1 2; 2 2 3]
b = [1, 0, 1]

function proverca(A, b)
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
Нет решения или их число бесконечное.
```

Рис. 13: 4.3.

4.4.

```
# 4.4
using LinearAlgebra

# Коэффициенты уравнений
A = [1 1 1; 1 1 2; 2 2 3]
b = [1, 0, 0]

function proverca(A, b)
    try  A\b
    catch e
        return "Нет решения или их число бесконечное."
    else
        return A\b
    end
end
println(proverca(A, b))
```

```
Нет решения или их число бесконечное.
```

Рис. 14: 4.4.

Часть 3: Операции с матрицами

5. Приведите приведённые ниже матрицы к диагональному виду.

$$\begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}, (5.1) \qquad \begin{pmatrix} 1 & -2 \\ -2 & 3 \end{pmatrix}, (5.2) \qquad \begin{pmatrix} 1 & -2 & 0 \\ -2 & 1 & 2 \\ 0 & -2 & 0 \end{pmatrix}, (5.3)$$

Рис. 15: Матрицы

5.1.

```
# 5.1
function Diagonal_vid(матрица)
    Diag_M = Diagonal(матрица)
    return Diag_M
end

# Пример использования
A1 = [1 -2; -2 1]

Diag_M1 = Diagonal_vid(A1)

display(Diag_M1)
```

```
2×2 Diagonal{Int64, Vector{Int64}}:
 1  ⋅
 ⋅  1
```

Рис. 16: 5.1.

5.2.

```
# 5.2
function Diagonal_vid(Matrica)
    Diag_M = Diagonal(Matrica)
    return Diag_M
end

# Пример использования
A2 = [1 -2; -2 3]

Diag_M1 = Diagonal_vid(A2)

display(Diag_M1)
```

```
2×2 Diagonal{Int64, Vector{Int64}}:
 1  ⋅
 ⋅  3
```

Рис. 17: 5.2.

5.3.

```
# 5.3
function Diagonal_vid(Matrica)
    Diag_M = Diagonal(Matrica)
    return Diag_M
end

# Пример использования
A3 = [1 -2 0; -2 1 2; 0 -2 0]

Diag_M1 = Diagonal_vid(A3)

display(Diag_M1)
```

```
3×3 Diagonal{Int64, Vector{Int64}}:
 1  ⋅  ⋅
 ⋅  1  ⋅
 ⋅  ⋅  0
```

Рис. 18: 5.3.

6. Вычислите

$$\begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}^{10}, (6.1) \quad \sqrt{\begin{pmatrix} 5 & -2 \\ -2 & 5 \end{pmatrix}}, (6.2) \quad \sqrt[3]{\begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}}, (6.3) \quad \sqrt{\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}}, (6.4)$$

Рис. 19: Матрицы

6.1.

```
# 6.1
A = [1 -2; -2 1]
B = A^10
display(B)
```

```
2×2 Matrix{Int64}:
  29525  -29524
 -29524   29525
```

Рис. 20: 6.1.

6.2.

```
# 6.2
A = [5 -2; -2 5]
X = eigvecs(A)
el = eigvals(A)
el_diag = Diagonal(el)
el_end = [(z)^(1/2) for z in el_diag]
B = X*(X^(-1))*(el_end)
display(B)
```

```
2×2 Matrix{Float64}:
 1.73205  0.0
 0.0      2.64575
```

Рис. 21: 6.2.

```
# 6.3
A = [1.0 -2.0; -2.0 1.0]
X = eigvecs(A)
el = eigvals(A)
el_diag = Diagonal(el)
el_end = [(z + 0*im)^(1/3) for z in el_diag]
B = X*(X^(-1))*(el_end)
display(B)
```

```
2×2 Matrix{ComplexF64}:
 0.5+0.866025im      0.0+0.0im
 0.0+0.0im       1.44225+0.0im
```

Рис. 22: 6.3.

```
# 6.4
A = [1 2; 2 3]
X = eigvecs(A)
el = eigvals(A)
el_diag = Diagonal(el)
el_end = [(z + 0*im)^(1/2) for z in el_diag]
B = X*(X^(-1))*(el_end)
display(B)
```

```
2×2 Matrix{ComplexF64}:
 0.0+0.485868im     1.14251e-16+0.0im
 0.0-2.69711e-17im      2.05817+0.0im
```

Рис. 23: 6.4.

7. Найдите собственные значения матрицы A, если

$$A = \begin{pmatrix} 140 & 97 & 74 & 168 & 131 \\ 97 & 106 & 89 & 131 & 36 \\ 74 & 89 & 152 & 144 & 71 \\ 168 & 131 & 144 & 54 & 142 \\ 131 & 36 & 71 & 142 & 36 \end{pmatrix}$$

Рис. 24: Матрица

Создайте диагональную матрицу из собственных значений матрицы A. Создайте нижнедиагональную матрицу из матрица A. Оцените эффективность выполняемых

7.1.

```julia
# 7.1
A = [140  97   74   168  131; 97   106  89   131  36; 74   89   152  144  71; 168  131  144  54   142; 131  36   71   142  36]

sobs_znach= eigvals(A)
println("Собственные значения матрицы A:\n")
display(sobs_znach)

diag_matrix = Diagonal(sobs_znach)
println("Диагональная матрица из собственных значений:\n")
display(diag_matrix)

low_diag_matrix = Bidiagonal(A, :L)
low_diag_matrix = low_diag_matrix .- Diagonal(A)
println("Нижнедиагональная матрица из матрицы A:\n")
display(low_diag_matrix)
```

```
Собственные значения матрицы A:

5-element Vector{Float64}:
 -128.49322764802145
  -55.887784553056875
   42.7521672793189
   87.16111477514521
  542.4677301466143
Диагональная матрица из собственных значений:

5×5 Diagonal{Float64, Vector{Float64}}:
 -128.493    ⋅        ⋅        ⋅        ⋅
     ⋅     -55.8878   ⋅        ⋅        ⋅
     ⋅        ⋅      42.7522   ⋅        ⋅
     ⋅        ⋅        ⋅      87.1611   ⋅
     ⋅        ⋅        ⋅        ⋅      542.468
Нижнедиагональная матрица из матрицы A:

5×5 Tridiagonal{Int64, Vector{Int64}}:
  0   0    ⋅    ⋅  ⋅
 97   0    0    ⋅  ⋅
  ⋅  89    0    0  ⋅
  ⋅   ⋅  144    0  0
  ⋅   ⋅    ⋅  142  0
```

Рис. 25: 7.1.

Часть 4: Линейные модели экономики

Линейная модель экономики может быть записана как СЛАУ $x - Ax = y$, где элементы матрицы A и столбца y — неотрицательные числа. По своему смыслу в экономике элементы матрицы A и столбцов x, y не могут быть отрицательными числами.

8. Матрица A называется продуктивной, если решение x системы при любой неотрицательной правой части y имеет только неотрицательные элементы x_i. Используя это определение, проверьте, являются ли матрицы продуктивными.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, (8.2) \qquad \frac{1}{2}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, (8.1) \qquad \frac{1}{10}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, (8.3)$$

Рис. 26: Матрица

8.1.

```
using LinearAlgebra

A1 = [1 2; 3 4]
E = ones(Int, 2, 2)

function proverca_prodyctiv(A1)
    prodyctiv = true
    n = size(A1, 1)
    y = [1.0; 2.0]
    x = (Diagonal(E) - A1)^(-1) * y  #  система уравнений Ax = y
    println(x)
    if any(x .< 0)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A1)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[-0.16666666666666663, -0.5]
Матрица не является продуктивной.
```

Рис. 27: 8.1.

```
using LinearAlgebra

A2 = [1 2; 3 4]*(1/2)
E = ones(Int, 2, 2)

# Функция для проверки продуктивности матрицы
function proverca_prodyctiv(A2)
    prodyctiv = true
    n = size(A2, 1)
    y = [3.0; 3.0]
    x = (Diagonal(E) - A2)^(-1) * y  #  система уравнений Ax = y
    println(x)
    if any(x .< 0)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A2)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[0.0, -3.0]
Матрица не является продуктивной.
```

Рис. 28: 8.2.

```
using LinearAlgebra

A3 = 0.1*[1 2; 3 4]
E = ones(Int, 2, 2)

function proverca_prodyctiv(A3)
    prodyctiv = true
    n = size(A3, 1)
    y = [4.0; 1.0]
    x = (Diagonal(E) - A3)^(-1) * y  #  система уравнений Ax = y
    println(x)
    if any(x .< 0)
        prodyctiv = false
    end
    return prodyctiv
end

result = proverca_prodyctiv(A3)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[5.416666666666667, 4.375]
Матрица является продуктивной.
```

Рис. 29: 8.3.

9.0.

9. Критерий продуктивности: матрица A является продуктивной тогда и только тогда, когда все элементы матрица $(E - A)^{-1}$ являются неотрицательными числами. Используя этот критерий, проверьте, являются ли матрицы продуктивными.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, (9.1) \qquad \frac{1}{2}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, (9.2) \qquad \frac{1}{10}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, (9.3)$$

Рис. 30: Матрица

```
using LinearAlgebra

A1 = [1 2; 3 4]
E = ones(Int, 2, 2)

function proverca_prodyctiv(A1)
    prodyctiv = true
    x = (Diagonal(E) - A1)^(-1)  #  система уравнений Ax = y
    display(x)
    if any(x .< 0)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A1)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
2×2 Matrix{Float64}:
  0.5  -0.333333
 -0.5   0.0
Матрица не является продуктивной.
```

Рис. 31: 9.1.

```
using LinearAlgebra

A1 = [1 2; 3 4]*(1/2)
E = ones(Int, 2, 2)

function proverca_prodyctiv(A1)
    prodyctiv = true
    x = (Diagonal(E) - A1)^(-1)  #  система уравнений Ax = y
    display(x)
    if any(x .< 0)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A1)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
2×2 Matrix{Float64}:
  0.5   -0.5
 -0.75  -0.25
Матрица не является продуктивной.
```

Рис. 32: 9.2.

9.3.

```
using LinearAlgebra

A1 = [1 2; 3 4]*(0.1)
E = ones(Int, 2, 2)

function proverca_prodyctiv(A1)
    prodyctiv = true
    x = (Diagonal(E) - A1)^(-1)  #  система уравнений Ax = y
    display(x)
    if any(x .< 0)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A1)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
2×2 Matrix{Float64}:
 1.25   0.416667
 0.625  1.875
Матрица является продуктивной.
```

Рис. 33: 9.3.

10.0.

10. Спектральный критерий продуктивности: матрица является продуктивной тогда и только тогда, когда все её собственные значения по модулю меньше 1. Используя этот критерий, проверьте, являются ли матрицы продуктивными.

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix},\ (10.1)\qquad \frac{1}{2}\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix},\ (10.2)\qquad \frac{1}{10}\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix},\ (10.3)\qquad \begin{pmatrix}0.1 & 0.2 & 0.3\\ 0 & 0.1 & 0.2\\ 0 & 0.1 & 0.3\end{pmatrix},\ (10.4)$$

Рис. 34: Матрица

10.1.

```
using LinearAlgebra

A1 = [1 2; 3 4]

function proverca_prodyctiv(A1)
    prodyctiv = true
    sobs_znach = eigvals(A1)
    x = abs.(sobs_znach)
    println(x)
    if any(x .> 1)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A1)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[0.3722813232690143, 5.372281323269014]
Матрица не является продуктивной.
```

Рис. 35: 10.1.

10.2.

```
using LinearAlgebra

A2 = [1 2; 3 4]*(1/2)

function proverca_prodyctiv(A2)
    prodyctiv = true
    sobs_znach = eigvals(A2)
    x = abs.(sobs_znach)
    println(x)
    if any(x .> 1)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A2)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[0.18614066163450715, 2.686140661634507]
Матрица не является продуктивной.
```

Рис. 36: 10.2.

10.3.

```
using LinearAlgebra

A3 = [1 2; 3 4]*(1/10)

function proverca_prodyctiv(A3)
    prodyctiv = true
    sobs_znach = eigvals(A3)
    x = abs.(sobs_znach)
    println(x)
    if any(x .> 1)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A3)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[0.037228132326901475, 0.5372281323269015]
Матрица является продуктивной.
```

Рис. 37: 10.3.

9.1.

```
using LinearAlgebra

A4 = [0.1 0.2 0.3; 0.0 0.1 0.2; 0.0 0.1 0.3]

function proverca_prodyctiv(A4)
    prodyctiv = true
    sobs_znach = eigvals(A4)
    x = abs.(sobs_znach)
    println(x)
    if any(x .> 1)
        prodyctiv = false
    end
    return prodyctiv
end

# Проверяем матрицу на продуктивность
result = proverca_prodyctiv(A4)
if result
    println("Матрица является продуктивной.")
else
    println("Матрица не является продуктивной.")
end
```

```
[0.02679491924311228, 0.1, 0.37320508075688774]
Матрица является продуктивной.
```

Рис. 38: 10.4.

Выводы

Выводы

Мною были изучены возможности специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.