

# Лабораторная работа № 4. Линейная алгебра.

Логинов Сергей

НФИбд-01-18

# Выполнение примеров из пособия

В теоретической части лабораторной работы мы изучили следующие темы:

- ▶ Поэлементные операции над многомерными массивами
- ▶ Транспонирование, след, ранг, определитель и инверсия матрицы
- ▶ Вычисление нормы векторов и матриц, повороты, вращения
- ▶ Матричное умножение, единичная матрица, скалярное произведение
- ▶ Факторизация. Специальные матричные структуры
- ▶ Общая линейная алгебра

# Задания для самостоятельного выполнения

- ▶ Произведение векторов
- ▶ Системы линейных уравнений
- ▶ Операции с матрицами
- ▶ Линейные модели экономики

# 1 Произведение векторов

переменной outer\_v

```
v = [1, 2, 3]
```

3-element Vector{Int64}:

1  
2  
3

```
dot_v = dot(v, v)
```

14

```
outer_v = v * v'
```

3×3 Matrix{Int64}:

1 2 3  
2 4 6  
3 6 9

## 2 Системы линейных уравнений

$$\begin{cases} x + y = 2, \\ x - y = 3 \end{cases}$$

```
function slu(x, y)
    result = x \ y
    return result
end
```

slu (generic function with 1 method)

Уравнения с двумя неизвестными

```
# a)
slu([1 1; 1 -1], [2; 3])
```

2-element Vector{Float64}:  
 2.5  
 -0.5

### 3 Операции с матрицами

$$\begin{pmatrix} 1 & -2 & 0 \\ -2 & 1 & 2 \\ 0 & 2 & 0 \end{pmatrix}$$

```
# c)
spectr = eigen([1 -2 0; -2 1 2; 0 2 0])

Eigen{Float64, Float64, Matrix{Float64}, Vector{Float64}}
values:
3-element Vector{Float64}:
-2.1413361156553594
 0.5151380471280724
 3.6261980685272936
vectors:
3x3 Matrix{Float64}:
 0.421859  0.717093  0.554808
 0.6626   0.173846 -0.728518
-0.618866 0.674948 -0.401808
```

```
diagm(spectr.values)
```

```
3x3 Matrix{Float64}:
-2.14134  0.0  0.0
 0.0      0.515138  0.0
 0.0      0.0  3.6262
```

```
spectr.vectors * diagm(spectr.values) * inv(spectr.vectors)
```

```
3x3 Matrix{Float64}:
 1.0      -2.0 -1.11022e-16
-2.0      1.0  2.0
-2.22045e-16 2.0  5.88418e-15
```

## 4 Линейные модели экономики

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
# 1)
a = [1 2; 3 4]
```

```
2x2 Matrix{Int64}:
 1  2
 3  4
```

```
e = [1 0; 0 1]
```

```
2x2 Matrix{Int64}:
 1  0
 0  1
```

```
s = e - a
```

```
2x2 Matrix{Int64}:
 0  -2
-3  -3
```

```
y = [0.00001; 0.00002]
```

```
2-element Vector{Float64}:
 1.0e-5
 2.0e-5
```

```
x = y' * inv(s)
```

```
1x2 adjoint{::Vector{Float64}} with eltype Float64:
-5.0e-6 -3.33333e-6
```