

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Российский государственный университет им. А.Н. Косыгина
(Технологии. Дизайн. Искусство)»

Кафедра автоматизированных систем обработки информации и управления

Отчет по лабораторной работе № 1
по дисциплине «Компьютерное моделирование»

Тема: «Основы MATLAB»

Выполнил: Ольховский Н.С., ИТА-123

Проверила: Самойлова Т.А.

Москва 2025

Вариант 13

Часть 1. Работа с матрицами

Задание:

- 1) Выполнить сложение и умножение матриц A и B.
- 2) Выполнить поэлементные операции: умножение и деление матриц A и B, а также возведение матриц в степень p.
- 3) Найти обратные матрицы для A и B. Выполнить проверку.
- 4) Транспонировать матрицы A и B.
- 5) Найти номера и значения элементов, удовлетворяющих заданному условию (в обеих матрицах).

Вариант 13.							
A				B			
-8	-8	12	-8	10	-2	16	-9
-5	-4	14	-4	7	16	-9	-5
9	6	-4	18	-19	-8	19	10
6	1	-11	3	3	4	-9	12
p=3							
Условие: неположительные элементы							

Рис. 1. Условие

```
A = [-8, -8, 12, -8;  
     -5, -4, 14, -4;  
     9, 6, -4, 18;  
     6, 1, -11, 3];  
  
B = [10, -2, 16, -9;  
     7, 16, -9, -5;  
     -19, -8, 19, 10;  
     3, 4, -9, 12];  
  
p = 2;  
  
%% 1. Сложение и умножение матриц  
sum_AB = A + B;  
prod_AB = A * B;  
  
%% 2. Поэлементные операции  
elementwise_mult = A .* B;  
elementwise_div = A ./ B;  
  
power_A = A .^ p;  
power_B = B .^ p;
```

Рис. 2. Программный код

```

%% 3. Обратные матрицы, проверка
inv_A = inv(A);
inv_B = inv(B);

check_A = A * inv_A;
check_B = B * inv_B;

%% 4. Транспонировать матрицы
transposed_A = A';
transposed_B = B';

%% 5. Номера и значения элементов, удовлетворяющих условию: неположительные элементы
[row_A, col_A, values_A] = find(A <= 0);
[row_B, col_B, values_B] = find(B <= 0);

% Вывод результатов
disp('Сложение матриц A и B:');
disp(sum_AB);

disp('Умножение матриц A и B:');
disp(prod_AB);

disp('Поэлементное умножение матриц A и B:');
disp(elementwise_mult);

disp('Поэлементное деление матриц A и B:');
disp(elementwise_div);

disp(['Матрица A в степени ' num2str(p) ':']);
disp(power_A);

disp(['Матрица B в степени ' num2str(p) ':']);
disp(power_B);

disp('Обратная матрица A:');
disp(inv_A);

disp('Обратная матрица B:');
disp(inv_B);

disp('Проверка A * inv(A):');
disp(check_A);

disp('Проверка B * inv(B):');
disp(check_B);

disp('Транспонированная матрица A:');
disp(transposed_A);

disp('Транспонированная матрица B:');
disp(transposed_B);

disp('Номера и значения элементов A <= 0:');
disp(table(row_A, col_A, values_A));

disp('Номера и значения элементов B <= 0:');
disp(table(row_B, col_B, values_B));

```

Рис. 3. Продолжение программного кода

```

>> Lab1_1
Сложение матриц A и B:
  2  -10  28  -17
  2   12   5   -9
 -10  -2  15  28
   9   5 -20  15

Умножение матриц A и B:
 -388 -240  244  136
 -356 -182  258  157
  262  182 -148   65
  285  104 -149 -133

Поэлементное умножение матриц A и B:
 -80  16  192  72
 -35 -64 -126  20
 -171 -48 -76  180
  18   4   99  36

Поэлементное деление матриц A и B:
 -0.8000  4.0000  0.7500  0.8889
 -0.7143 -0.2500 -1.5556  0.8000
 -0.4737 -0.7500 -0.2105  1.8000
  2.0000  0.2500  1.2222  0.2500

Матрица A в степени 2:
  64  64  144  64
  25  16  196  16
  81  36  16  324
  36   1  121   9

Матрица B в степени 2:
 100   4  256  81
  49  256  81  25
 361  64  361 100
   9  16  81  144

Обратная матрица A:
 -0.2013  0.4026 -0.0519  0.3117
 -0.2045  0.0341 -0.0568 -0.1591
 -0.0877  0.1753 -0.0065  0.0390
  0.1494 -0.1737  0.0990 -0.0942

Обратная матрица B:
 0.0485 -0.0197 -0.0264  0.0502
 0.0083  0.0785  0.0349  0.0099
 0.0429  0.0176  0.0312  0.0135
 0.0173 -0.0080  0.0184  0.0776

Проверка A * inv(A):
 1.0000 -0.0000 -0.0000  0.0000
 0.0000  1.0000 -0.0000  0.0000
 0  0.0000  1.0000 -0.0000
 -0.0000  0.0000  0.0000  1.0000

Проверка B * inv(B):
 1.0000  0.0000 -0.0000  0.0000
 0.0000  1.0000  0.0000  0
 -0.0000 -0.0000  1.0000  0.0000
 0  0.0000  0  1.0000

```

Транспонированная матрица A:

-8	-5	9	6
-8	-4	6	1
12	14	-4	-11
-8	-4	18	3

Транспонированная матрица B:

10	7	-19	3
-2	16	-8	4
16	-9	19	-9
-9	-5	10	12

Номера и значения элементов A <= 0:

row_A	col_A	values_A
1	1	true
2	1	true
1	2	true
2	2	true
3	3	true
4	3	true
1	4	true
2	4	true

Номера и значения элементов B <= 0:

row_B	col_B	values_B
3	1	true
1	2	true
3	2	true
2	3	true
4	3	true
1	4	true
2	4	true

Рис. 4. Результат выполнения программы

Часть 2. Построение графика функции и нахождение экстремума

Задание:

1. Построить график функции. Подписать оси на графике, настроить шрифты. Добавить легенду.
2. Найти экстремум функции

13	$y = \arctg x + \sqrt{x} + 2$	0	1	$\frac{\pi}{200}$	min
----	-------------------------------	---	---	-------------------	-----

Рис. 5. Условие

```
% Определение диапазона и шага
x = 0:pi/200:1;

% Вычисление функции
y = atan(x) + sqrt(x) + 2;

% Построение графика функции
figure;
plot(x, y, 'b-', 'LineWidth', 2);
xlabel('x', 'FontSize', 12);
ylabel('y', 'FontSize', 12);
title('График функции y = arctg(x) + sqrt(x) + 2', 'FontSize', 14);
legend('y = arctg(x) + sqrt(x) + 2', 'FontSize', 10);
grid on;

% Нахождение экстремума функции
min_value = inf; % Начальное значение для минимума
min_position = NaN; % Позиция минимума

for i = 1:length(x)
    if y(i) < min_value
        min_value = y(i);
        min_position = x(i);
    end
end

% Вывод результата
fprintf('Минимум функции достигается в точке x = %.4f и имеет значение y = %.4f\n', min_position, min_value);
```

Рис. 6. Программный код

```
>> Lab1_2
```

Минимум функции достигается в точке $x = 0.0000$ и имеет значение $y = 2.0000$

Рис. 7. Результат выполнения программы



Рис. 8. График функции

Часть 3. Решение уравнений

Задание:

1. Решить систему линейных алгебраических уравнений 3 способами:

- методом обратной матрицы
- методом Гаусса
- с помощью стандартных функций MATLAB

Выполнить проверку найденного решения.

2. Найти корни полинома. Проверить найденные корни с помощью графика.

Вариант 13.

СЛАУ:

$$\begin{cases} -x_1 + 16x_2 + 15x_3 - 5x_4 = -18 \\ 19x_1 + 19x_2 - 2x_3 + 11x_4 = 13 \\ -8x_1 - 17x_2 + 15x_3 + 3x_4 = -11 \\ 14x_1 + 12x_2 + 11x_3 + 17x_4 = -9 \end{cases}$$

Полином:

$$y = -8x^3 + 5x$$

Рис. 9. Условие

```
% Матрица коэффициентов и свободных членов
A = [-1 16 15 -5;
     19 19 -2 11;
     -8 -17 15 3;
     14 12 11 17];
b = [-18; 13; -11; -9];
% 1. Метод обратной матрицы
x_inverse = inv(A) * b;
% 2. Метод Гаусса
x_gauss = A\b;
% 3. Стандартные функции MATLAB
x_standard = linsolve(A, b);
% Проверка решений
check_inverse = A * x_inverse;
check_gauss = A * x_gauss;
check_standard = A * x_standard;

% Вывод результатов
fprintf('Решение методом обратной матрицы:\n');
disp(x_inverse);
fprintf('Проверка: A * x_inverse = \n');
disp(check_inverse);
fprintf('Решение методом Гаусса:\n');
disp(x_gauss);
fprintf('Проверка: A * x_gauss = \n');
disp(check_gauss);
fprintf('Решение стандартной функцией MATLAB:\n');
disp(x_standard);
fprintf('Проверка: A * x_standard = \n');
disp(check_standard);
% Определение коэффициентов полинома
coeffs = [-8, 0, 5, 0];
% Нахождение корней полинома
roots_poly = roots(coeffs);
% Вывод корней
fprintf('Корни полинома:\n');
disp(roots_poly);
% Проверка корней с помощью графика
x_range = linspace(-2, 2, 100);
y_values = polyval(coeffs, x_range);
```

Рис. 10. Программный код

```

% Построение графика полинома
figure;
plot(x_range, y_values, 'r-', 'LineWidth', 2);
hold on;
plot(roots_poly, zeros(size(roots_poly)), 'bo', 'MarkerSize', 10,
'MarkerFaceColor', 'b');
xlabel('x', 'FontSize', 12);
ylabel('y', 'FontSize', 12);
title('График полинома y = -8x^3 + 5x', 'FontSize', 14);
grid on;
legend('y = -8x^3 + 5x', 'Корни полинома', 'FontSize', 10);
hold off;

```

Рис. 11. Продолжение программного кода

```

>> Lab1_3
Решение методом обратной матрицы:
    3.4607
   -1.4904
   -0.1278
   -2.2447
Проверка: A * x_inverse =
   -18.0000
    13.0000
   -11.0000
    -9.0000
Решение методом Гаусса:
    3.4607
   -1.4904
   -0.1278
   -2.2447
Проверка: A * x_gauss =
   -18.0000
    13.0000
   -11.0000
    -9.0000

```

```

Решение стандартной функцией MATLAB:
    3.4607
   -1.4904
   -0.1278
   -2.2447
Проверка: A * x_standard =
   -18.0000
    13.0000
   -11.0000
    -9.0000
Корни полинома:
         0
    0.7906
   -0.7906

```

Рис. 12. Результат выполнения программы

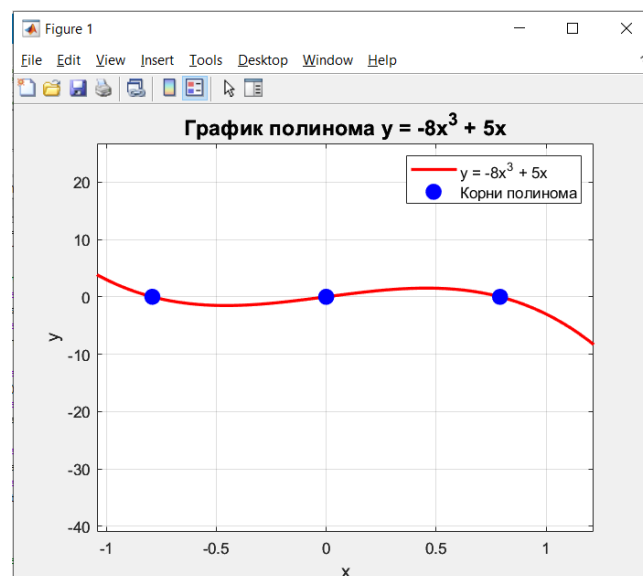


Рис. 13. График полинома