

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»

Кафедра телекоммуникационных систем и вычислительных средств  
(ТС и ВС)

Отчет по лабораторной работе №1  
по дисциплине  
*Теория массового обслуживания*

по теме:  
ВВЕДЕНИЕ В MATLAB. ПЕРЕМЕННЫЕ, ФУНКЦИИ, ГРАФИКА

Студент:  
*Группа ИА-331*

*Я.А Гмыря*

Предподаватель:  
*Преподаватель*

*А.В Андреев*

Новосибирск 2025 г.

## СОДЕРЖАНИЕ

ЦЕЛЬ И ЗАДАЧИ .....	3
1 ЧАСТЬ 1. ПЕРЕМЕННЫЕ, ФУНКЦИИ, ГРАФИКА.....	5
2 ЧАСТЬ 2. РЕШЕНИЕ УРАВНЕНИЯ .....	7
3 ЧАСТЬ 3. ФУНКЦИЯ ДВУХ ПЕРЕМЕННЫХ И 3D-ГРАФИК.....	9
4 ЧАСТЬ 4. МАТРИЧНЫЕ ОПЕРАЦИИ, ПРОГРАММИРОВАНИЕ ФУНКЦИЙ.....	11
5 ЧАСТЬ 5. МАРКОВСКИЕ ЦЕПИ. ОПРЕДЕЛЕНИЕ И ПОСТРОЕНИЕ .....	12

## ЦЕЛЬ И ЗАДАЧИ

**Цель:** Изучение системы MATLAB для инженерных, финансовых и математических расчетов. Выполнение учебного математического расчета

**Задачи:**

### Часть 1. Переменные, функции, графика

- Определите функцию  $f(x)$ , заданную по варианту.
- Постройте графики функции  $f(x)$ , её производной  $f'(x)$  и интеграла

$$F(x) = \int_0^x f(y) dy.$$

- Для вычисления производной используйте `diff`, для интеграла — `integral`.

### Часть 2. Решение уравнения

**Условие**

- Решите уравнение вида  $a \cdot x + b = f(x)$  графическим и аналитическим методами.
- Коэффициенты  $a$  и  $b$  задайте самостоятельно.
- Для численного решения используйте `fsolve`.

### Часть 3. Функция двух переменных и 3D-график

- Определите функцию двух переменных  $F(x,y)$ , заданную по варианту.
- Постройте трёхмерный график  $F(x,y)$  с использованием `meshgrid` и `surf`.

## Часть 4. Матричные операции, программирование функций

1. Создать новый скрипт MATLAB.
2. Разработать функцию, выполняющую следующие действия:
  - Сгенерировать вектор-столбец, элементы которого равномерно распределены в заданном диапазоне.
  - Сгенерировать вектор-строку, элементы которого равномерно распределены в заданном диапазоне.
  - Перемножить два вектора.
  - Вычислить среднее значение и дисперсию матрицы, полученной в результате перемножения векторов.
  - В качестве аргумента функции передавать размерности векторов.
3. Произвести расчеты с помощью функции и вывести результаты.
4. Сохранить результаты работы в виде отчета.

## Часть 5. Марковские цепи. Определение и построение

1. Построить сеть из  $L=15$  узлов в виде ориентированного графа:
  - Каждый узел должен иметь минимум 3 исходящих маршрута.
  - Каждый узел должен иметь хотя бы один входящий маршрут.
2. Построить матрицу переходов  $T=[p_{ij}]$ , задав:
  - Размерность  $L=15$
  - Ненулевые элементы вероятностей вручную, чтобы строки удовлетворяли условиям стохастичности.
3. Написать функции в MATLAB:
  - `stochastic(matrix)`: проверяет матрицу на стохастичность.
  - `ergodic(matrix, epsilon)`: проверяет цепь Маркова на эргодичность.
4. Проверить:
  - Стохастичность построенной матрицы.
  - Эргодичность цепи Маркова.

## ЧАСТЬ 1. ПЕРЕМЕННЫЕ, ФУНКЦИИ, ГРАФИКА

Реализация:

```
IA_331_lab1_1.m
1 %plots, diff and integral
2
3 %define function
4 f = @(x) exp(sin(x) + cos(x));
5
6 %define arguments
7 x = -10 : 0.1 : 10;
8
9 %define f(x)
10
11 y = f(x);
12
13 %f(x) plot
14 figure;
15 plot(x, y);
16 title("Function plot")
17 legend('exp(sin(x) + cos(x))');
18 xlabel("f");
19 ylabel("f(x)");
20 grid on;
21
22 %compute f'(x)
23 y_diff = diff(y) ./ diff(x); %delta y / delta x = diff
24
25 %f'(x) plot
26 figure;
27 plot(x(1:end-1), y_diff);
28 title("Function diff plot");
29 legend("exp(sin(x) + cos(x))'");
30 ylabel("f'(x)");
31 grid on;
32
33 %compute integral
34 f_integral = zeros(1, length(x));
35
36 for k = 1:length(x)
37     f_integral(k) = integral(f, 0, x(k));
38 end
39
40 %f_integral plot
41 figure;
42 plot(x, f_integral);
43 title("Function integral plot");
44 legend("exp(sin(x) + cos(x))dx");
45 xlabel("x");
46 ylabel("F(x)");
47 grid on;
48
49
```

Рисунок 1 — Реализация заданий из части 1

Результат:

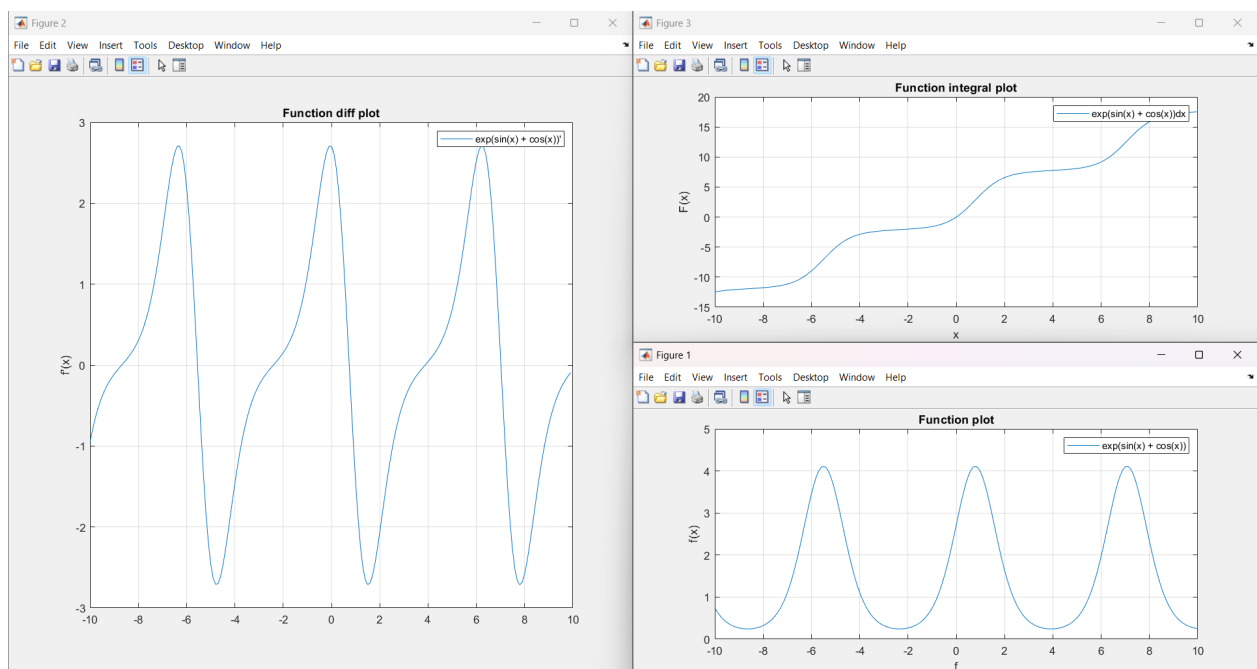
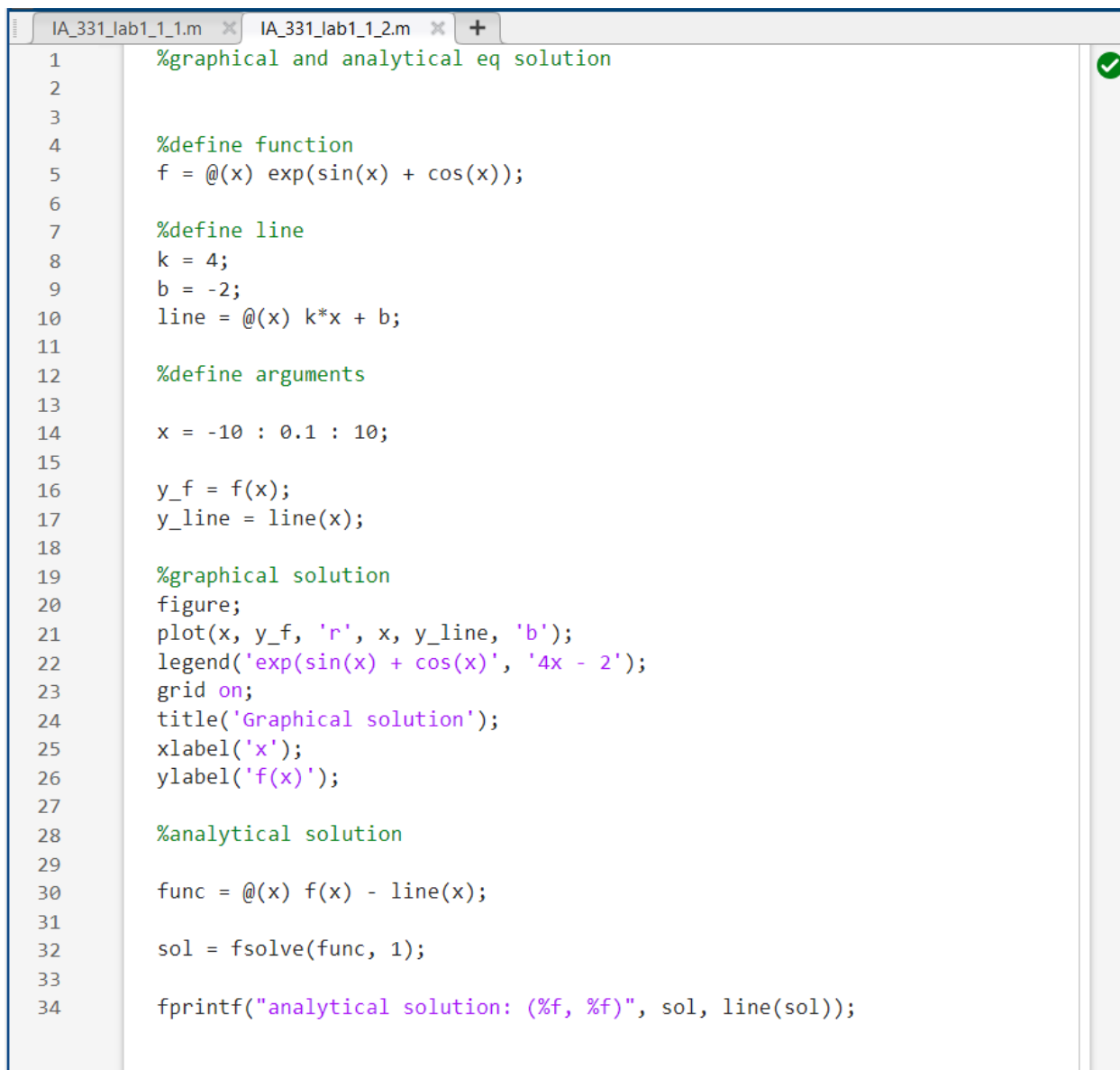


Рисунок 2 — Результат работы программы для части 1

## ЧАСТЬ 2. РЕШЕНИЕ УРАВНЕНИЯ

Реализация:



```
1 %graphical and analytical eq solution
2
3
4 %define function
5 f = @(x) exp(sin(x) + cos(x));
6
7 %define line
8 k = 4;
9 b = -2;
10 line = @(x) k*x + b;
11
12 %define arguments
13
14 x = -10 : 0.1 : 10;
15
16 y_f = f(x);
17 y_line = line(x);
18
19 %graphical solution
20 figure;
21 plot(x, y_f, 'r', x, y_line, 'b');
22 legend('exp(sin(x) + cos(x))', '4x - 2');
23 grid on;
24 title('Graphical solution');
25 xlabel('x');
26 ylabel('f(x)');
27
28 %analytical solution
29
30 func = @(x) f(x) - line(x);
31
32 sol = fsolve(func, 1);
33
34 fprintf("analytical solution: (%f, %f)", sol, line(sol));
```

Рисунок 3 — Реализация заданий из части 2

Результат:

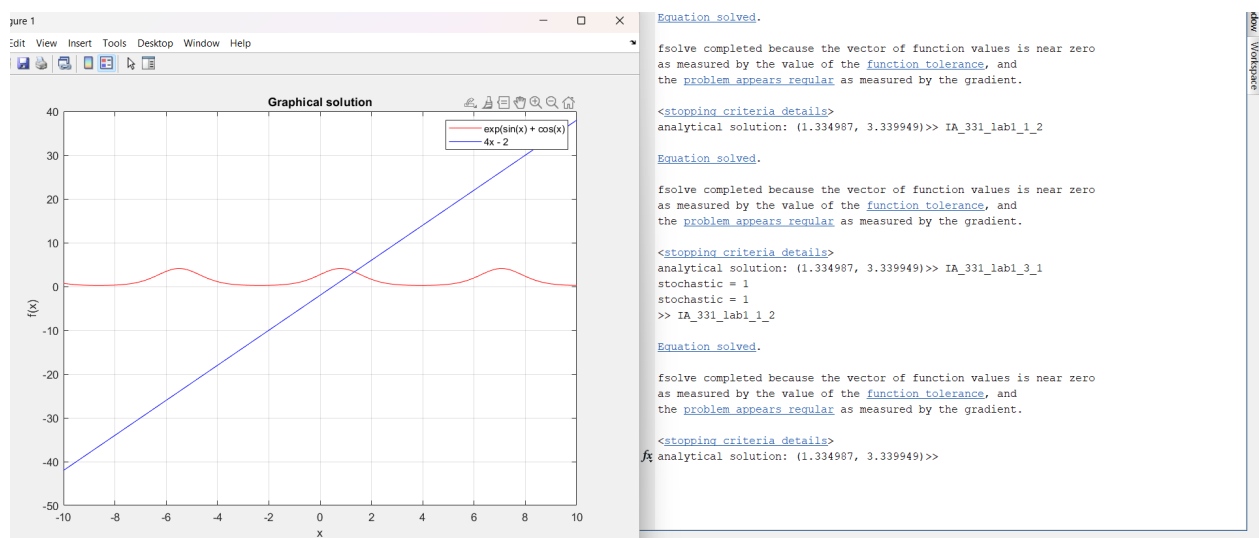
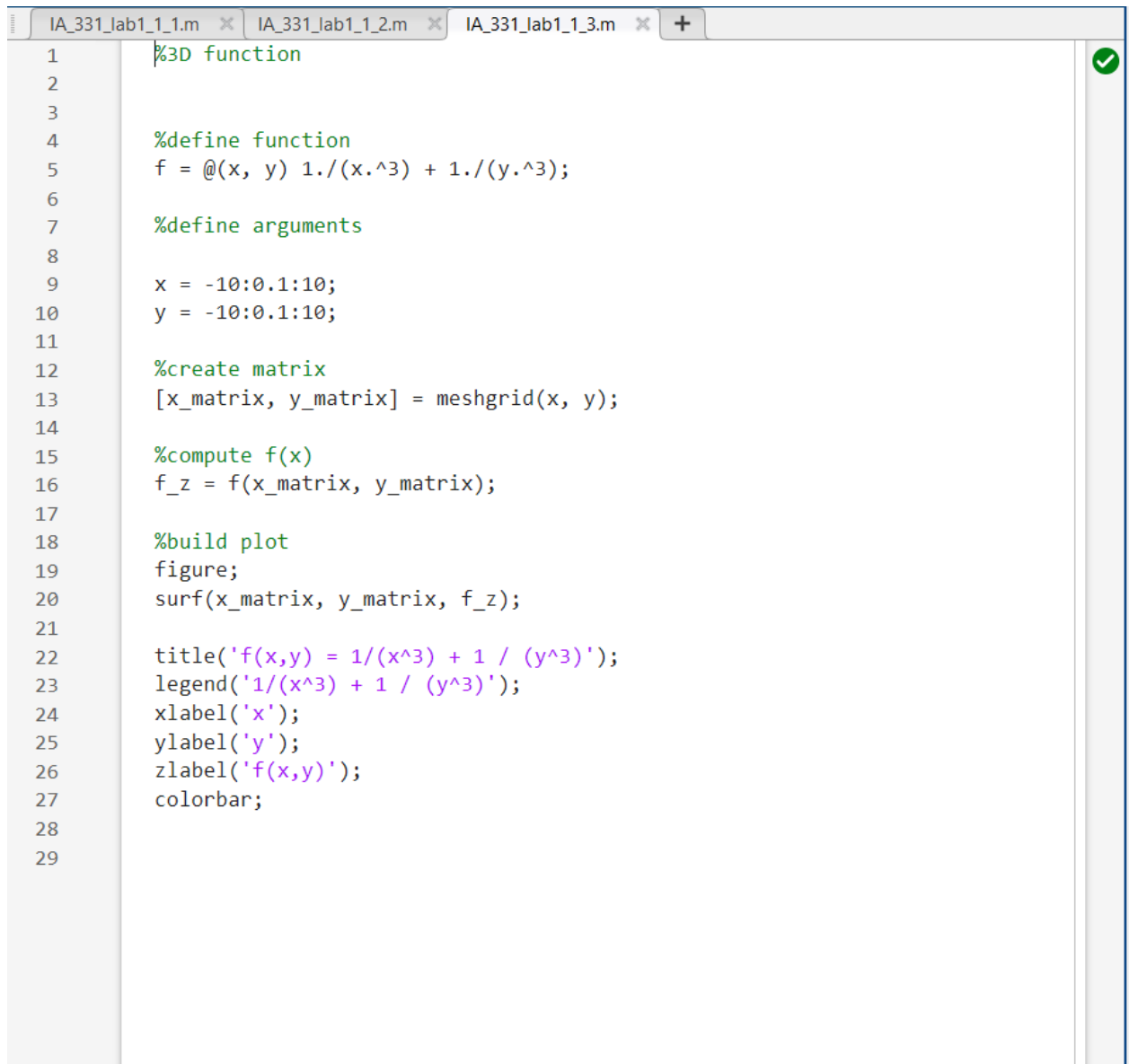


Рисунок 4 — Результат работы программы для части 2



### ЧАСТЬ 3. ФУНКЦИЯ ДВУХ ПЕРЕМЕННЫХ И 3D-ГРАФИК

Реализация:

A screenshot of a MATLAB script editor window. The window has three tabs at the top: 'IA\_331\_lab1\_1\_1.m', 'IA\_331\_lab1\_1\_2.m', and 'IA\_331\_lab1\_1\_3.m'. The third tab is active. The script contains 29 lines of MATLAB code for a 3D plot. The code defines a function  $f(x, y) = 1/(x^3) + 1/(y^3)$ , creates a meshgrid for  $x$  and  $y$  ranging from -10 to 10, computes the function values, and builds a 3D surface plot using the `surf` function. The plot is titled and labeled with axes and a colorbar. A green checkmark icon is visible in the top right corner of the editor window.

```
1 %3D function
2
3
4 %define function
5 f = @(x, y) 1./(x.^3) + 1./(y.^3);
6
7 %define arguments
8
9 x = -10:0.1:10;
10 y = -10:0.1:10;
11
12 %create matrix
13 [x_matrix, y_matrix] = meshgrid(x, y);
14
15 %compute f(x)
16 f_z = f(x_matrix, y_matrix);
17
18 %build plot
19 figure;
20 surf(x_matrix, y_matrix, f_z);
21
22 title('f(x,y) = 1/(x^3) + 1 / (y^3)');
23 legend('1/(x^3) + 1 / (y^3)');
24 xlabel('x');
25 ylabel('y');
26 zlabel('f(x,y)');
27 colorbar;
28
29
```

Рисунок 5 — Реализация заданий из части 3

Результат:

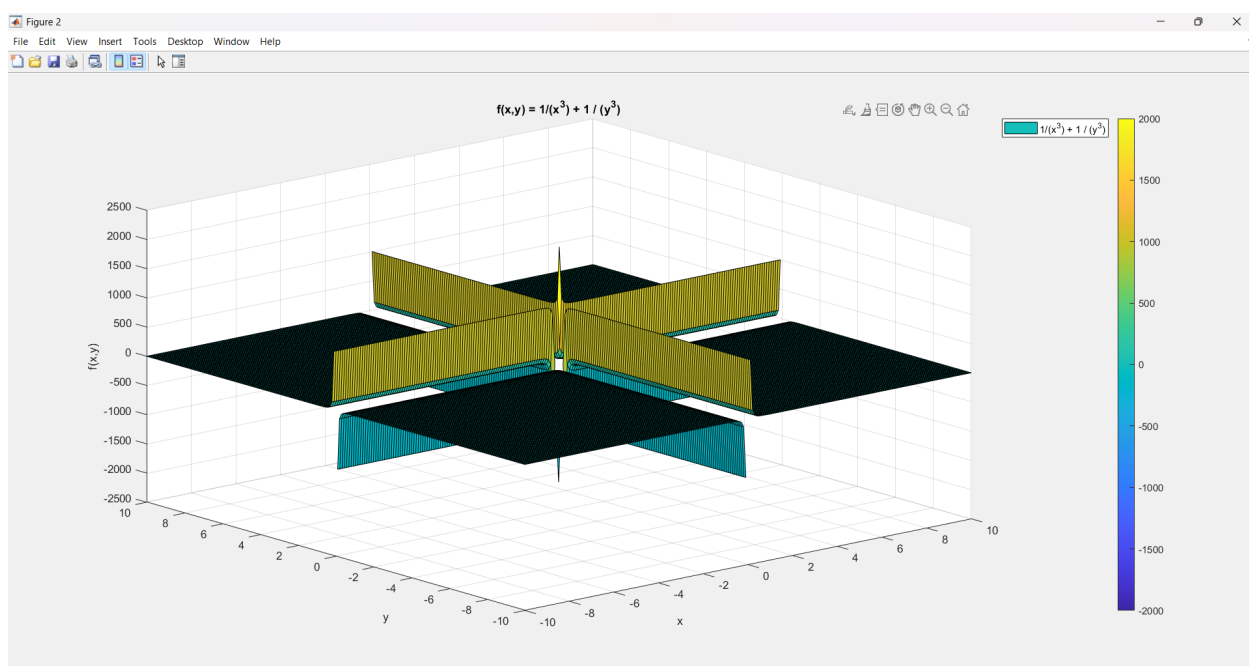
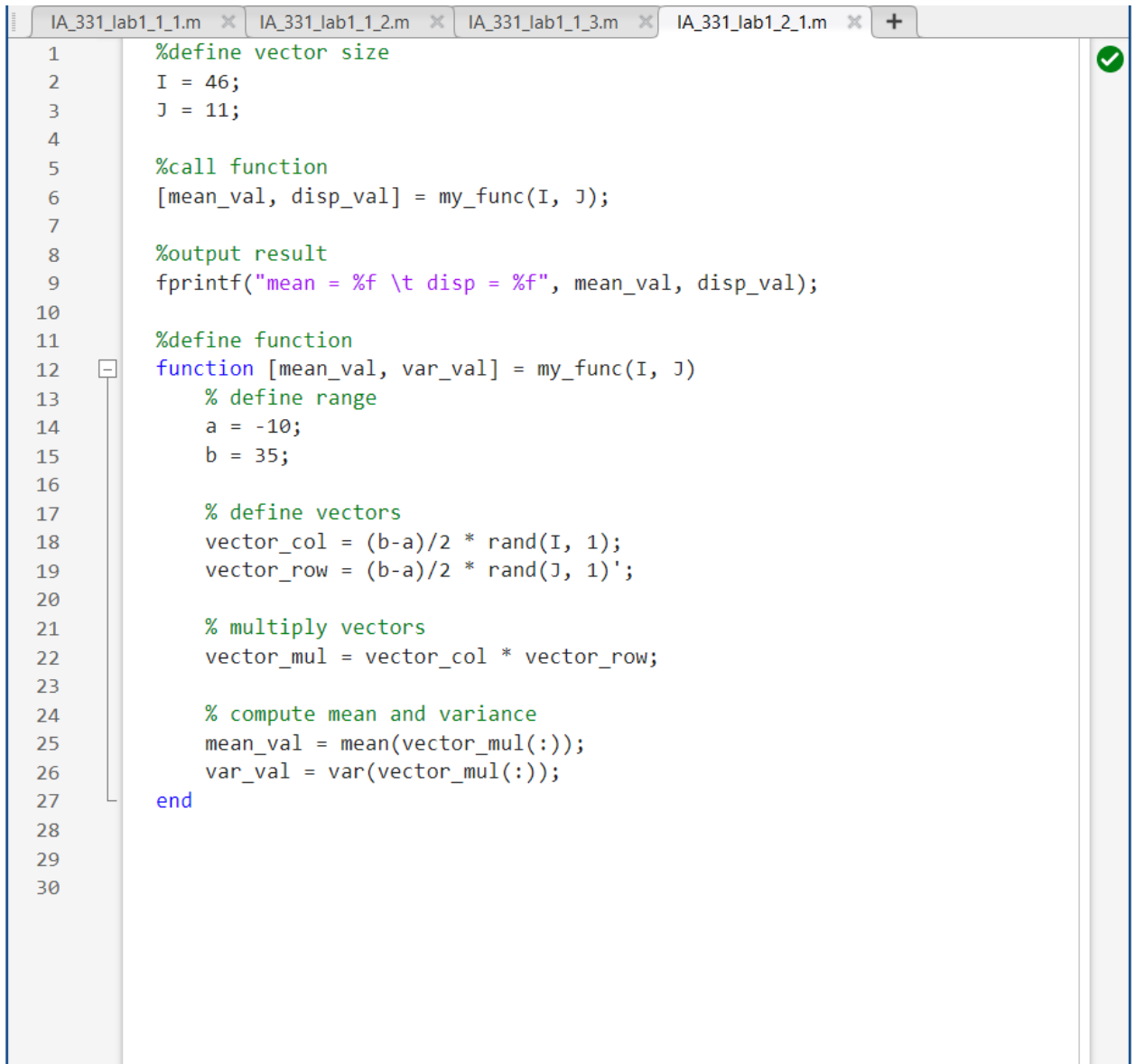


Рисунок 6 — Результат работы программы для части 3

## ЧАСТЬ 4. МАТРИЧНЫЕ ОПЕРАЦИИ, ПРОГРАММИРОВАНИЕ ФУНКЦИЙ

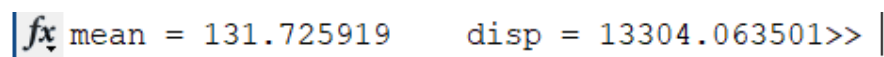
Реализация:



```
1 %define vector size
2 I = 46;
3 J = 11;
4
5 %call function
6 [mean_val, disp_val] = my_func(I, J);
7
8 %output result
9 fprintf("mean = %f \t disp = %f", mean_val, disp_val);
10
11 %define function
12 function [mean_val, var_val] = my_func(I, J)
13     % define range
14     a = -10;
15     b = 35;
16
17     % define vectors
18     vector_col = (b-a)/2 * rand(I, 1);
19     vector_row = (b-a)/2 * rand(J, 1)';
20
21     % multiply vectors
22     vector_mul = vector_col * vector_row;
23
24     % compute mean and variance
25     mean_val = mean(vector_mul(:));
26     var_val = var(vector_mul(:));
27 end
28
29
30
```

Рисунок 7 — Реализация заданий из части 4

Результат:

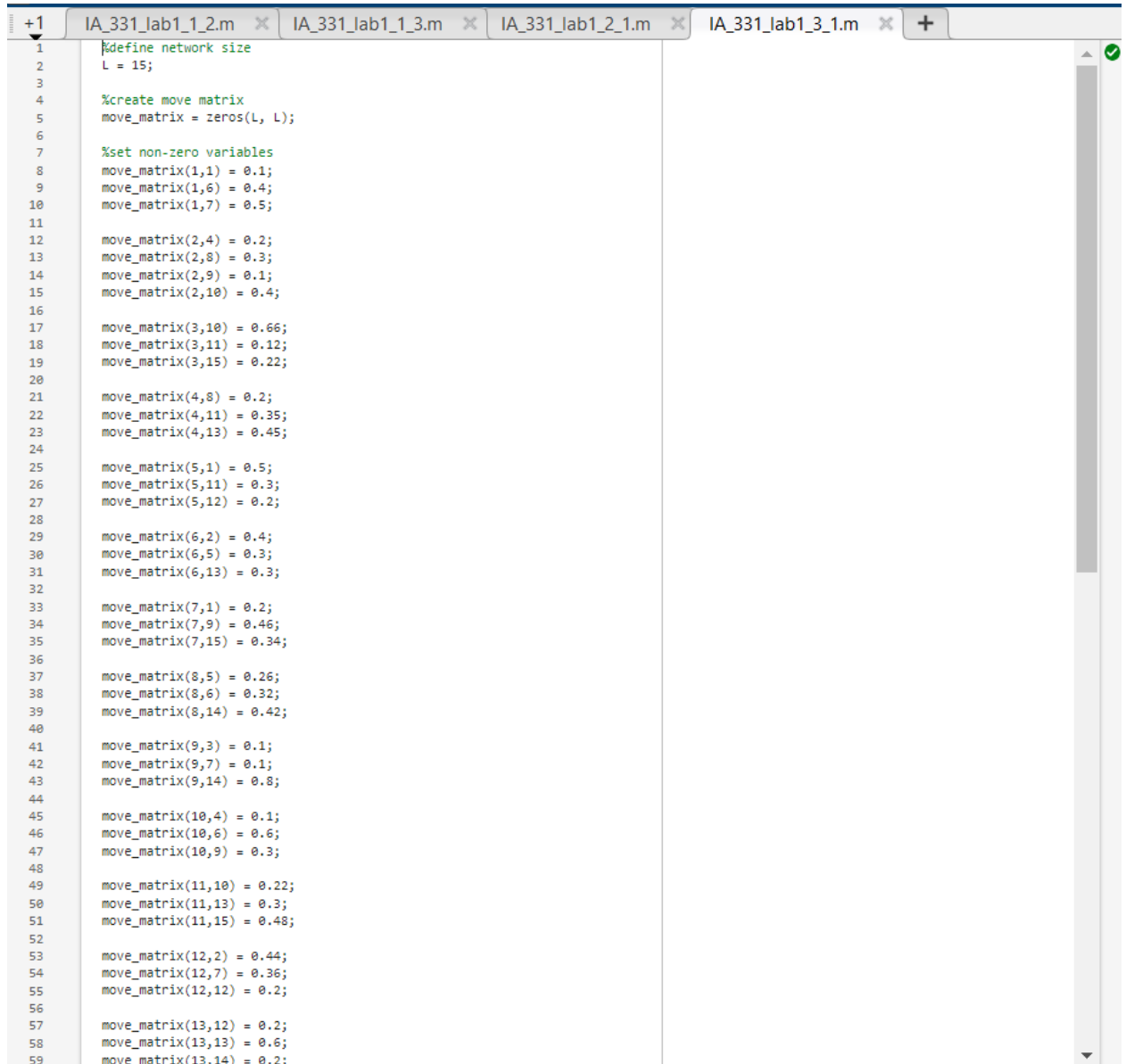


```
| fx mean = 131.725919    disp = 13304.063501>> |
```

Рисунок 8 — Результат работы программы для части 4

## ЧАСТЬ 5. МАРКОВСКИЕ ЦЕПИ. ОПРЕДЕЛЕНИЕ И ПОСТРОЕНИЕ

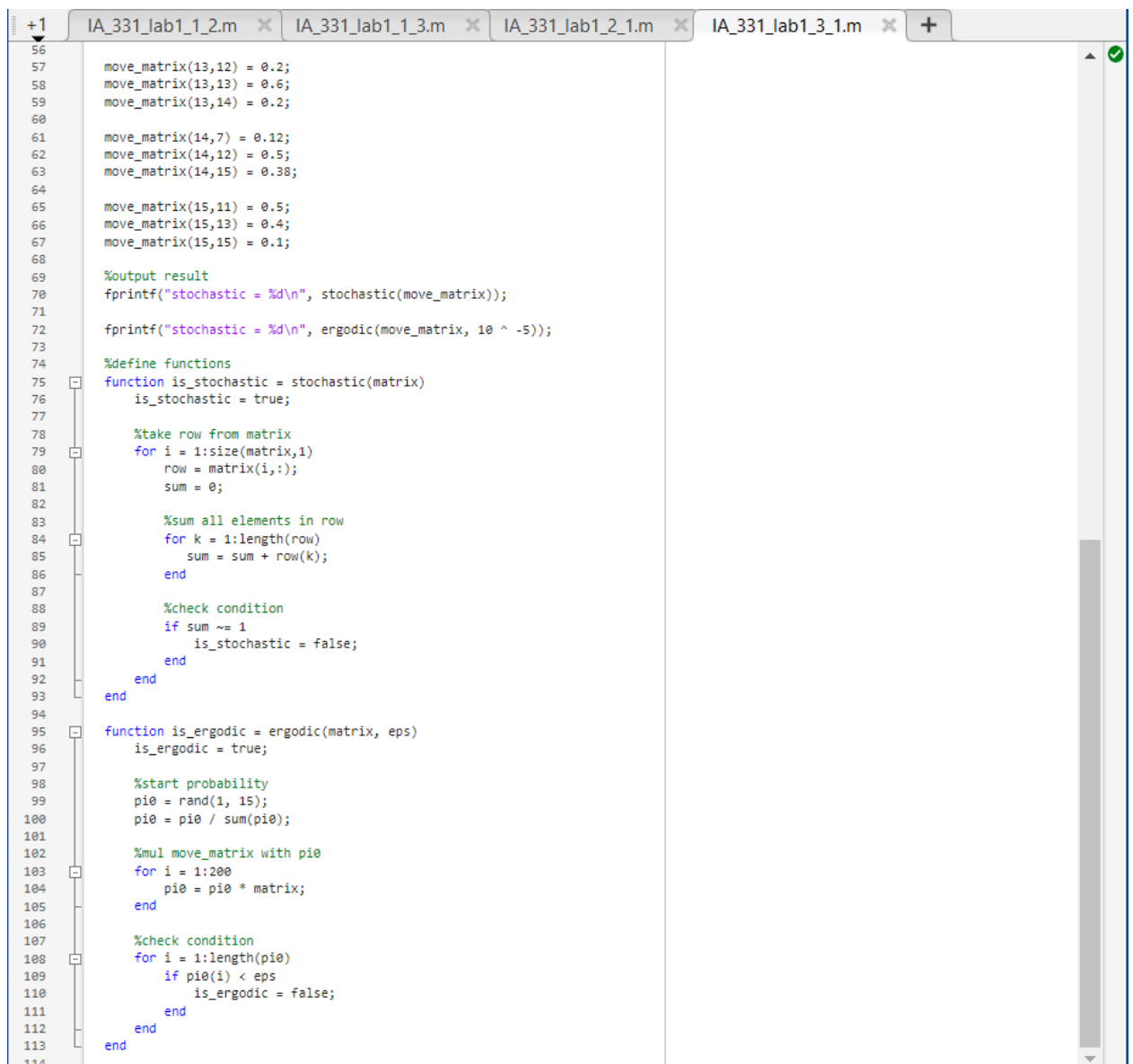
Реализация:

A screenshot of a MATLAB script editor window. The window has four tabs: 'IA\_331\_lab1\_1\_2.m', 'IA\_331\_lab1\_1\_3.m', 'IA\_331\_lab1\_2\_1.m', and 'IA\_331\_lab1\_3\_1.m'. The first tab is active. The script defines a network size L=15 and creates a 15x15 move matrix. It then sets various non-zero elements in the matrix with specific probabilities. The script is 59 lines long, with line numbers visible on the left. The code is as follows:

```
1 %define network size
2 L = 15;
3
4 %create move matrix
5 move_matrix = zeros(L, L);
6
7 %set non-zero variables
8 move_matrix(1,1) = 0.1;
9 move_matrix(1,6) = 0.4;
10 move_matrix(1,7) = 0.5;
11
12 move_matrix(2,4) = 0.2;
13 move_matrix(2,8) = 0.3;
14 move_matrix(2,9) = 0.1;
15 move_matrix(2,10) = 0.4;
16
17 move_matrix(3,10) = 0.66;
18 move_matrix(3,11) = 0.12;
19 move_matrix(3,15) = 0.22;
20
21 move_matrix(4,8) = 0.2;
22 move_matrix(4,11) = 0.35;
23 move_matrix(4,13) = 0.45;
24
25 move_matrix(5,1) = 0.5;
26 move_matrix(5,11) = 0.3;
27 move_matrix(5,12) = 0.2;
28
29 move_matrix(6,2) = 0.4;
30 move_matrix(6,5) = 0.3;
31 move_matrix(6,13) = 0.3;
32
33 move_matrix(7,1) = 0.2;
34 move_matrix(7,9) = 0.46;
35 move_matrix(7,15) = 0.34;
36
37 move_matrix(8,5) = 0.26;
38 move_matrix(8,6) = 0.32;
39 move_matrix(8,14) = 0.42;
40
41 move_matrix(9,3) = 0.1;
42 move_matrix(9,7) = 0.1;
43 move_matrix(9,14) = 0.8;
44
45 move_matrix(10,4) = 0.1;
46 move_matrix(10,6) = 0.6;
47 move_matrix(10,9) = 0.3;
48
49 move_matrix(11,10) = 0.22;
50 move_matrix(11,13) = 0.3;
51 move_matrix(11,15) = 0.48;
52
53 move_matrix(12,2) = 0.44;
54 move_matrix(12,7) = 0.36;
55 move_matrix(12,12) = 0.2;
56
57 move_matrix(13,12) = 0.2;
58 move_matrix(13,13) = 0.6;
59 move_matrix(13,14) = 0.2;
```

Рисунок 9 — Реализация заданий из части 5

Результат:



```
56
57     move_matrix(13,12) = 0.2;
58     move_matrix(13,13) = 0.6;
59     move_matrix(13,14) = 0.2;
60
61     move_matrix(14,7) = 0.12;
62     move_matrix(14,12) = 0.5;
63     move_matrix(14,15) = 0.38;
64
65     move_matrix(15,11) = 0.5;
66     move_matrix(15,13) = 0.4;
67     move_matrix(15,15) = 0.1;
68
69     %output result
70     fprintf("stochastic = %d\n", stochastic(move_matrix));
71
72     fprintf("stochastic = %d\n", ergodic(move_matrix, 10 ^ -5));
73
74     %define functions
75     function is_stochastic = stochastic(matrix)
76         is_stochastic = true;
77
78         %take row from matrix
79         for i = 1:size(matrix,1)
80             row = matrix(i,:);
81             sum = 0;
82
83             %sum all elements in row
84             for k = 1:length(row)
85                 sum = sum + row(k);
86             end
87
88             %check condition
89             if sum ~= 1
90                 is_stochastic = false;
91             end
92         end
93     end
94
95     function is_ergodic = ergodic(matrix, eps)
96         is_ergodic = true;
97
98         %start probability
99         pi0 = rand(1, 15);
100         pi0 = pi0 / sum(pi0);
101
102         %mul move_matrix with pi0
103         for i = 1:200
104             pi0 = pi0 * matrix;
105         end
106
107         %check condition
108         for i = 1:length(pi0)
109             if pi0(i) < eps
110                 is_ergodic = false;
111             end
112         end
113     end
114
```

Рисунок 10 — Реализация заданий из части 5

```
>> IA_331_lab1_3_1
stochastic = 1
ergodic = 1
```

Рисунок 11 — Результат работы программы для части 5