

1.

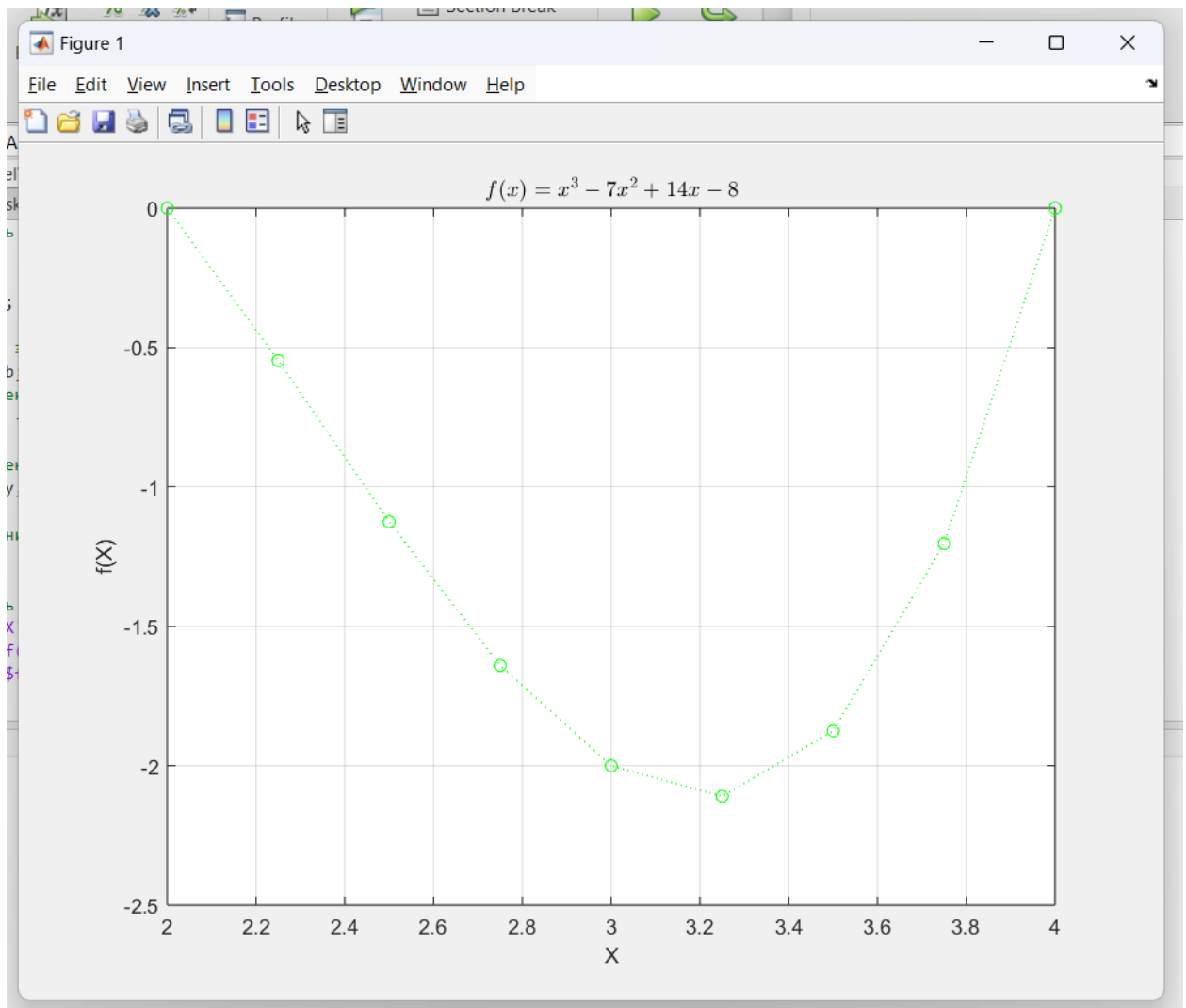
1. Построение графиков функций одной переменной в среде MatLab

1.1. Постройте график функции $f(x)$ в области определения $x \in [a; b]$ с шагом h с помощью функции `plot(x,y, LineSpec)`, при этом, задайте отображение линий графика цветом и стилем, указанными в вашем варианте (0.5 балла). Значения `LineSpec` см. в Приложении 1.

Подключите отображение сетки (команда `grid on`), подпишите оси X и Y (методы `xlabel()` и `ylabel()`). В заголовке области построения графика (`title()`) напишите выражение функции $f(x)$ по правилам TEX (0.5 балла).

MatLab Code:

```
% область определения и шаг
a = 2;
b = 4;
h = 0.25;
% Вектор значений x в заданном диапазоне с шагом h (массив иксов)
x = a:h:b;
% Вычисление значений функции f(x) для каждого x
y = x.^3 - 7*x.^2 + 14*x - 8;
% Построение графика с зелёной пунктирной линией и круглыми маркерами
plot(x, y, 'g:o')
% Включение сетки
grid on
% Подпись осей и заголовок графика
xlabel('X')
ylabel('f(X)')
title('$f(x) = x^3 - 7x^2 + 14x - 8$', 'Interpreter', 'latex')
```



1.2. В плоскости, где отображается график из задания 1.1 (команда `hold on`), постройте график этой же функции $f(x)$ с адаптивным выбором шага, обеспечивающим точность $\varepsilon = 10^{-5}$, с помощью функции `fplot(@fun, limits, tol, n, LineSpec)`, при этом, задайте отображение линий графика цветом и стилем, указанными в вашем варианте (0.8 балла).

Задайте отображение легенды с пояснением метода построения графика, например, `legend('Метод plot', 'Метод fplot', 'Location', ' ', 'Orientation', ' ')`, здесь в четвёртом и шестом параметрах укажите способ локации легенды и её расположение из вашего варианта (0.2 балла).

Сравните графики функции $f(x)$, сделайте вывод о необходимости коррекции шага h при построении графика с помощью функции `plot()`.

MatLab Code:

```
% область определения и шаг
```

```
a = 2;
```

```
b = 4;
```

```
h = 0.25;
```

```
% Вектор значений x в заданном диапазоне с шагом h (массив иксов)
```

```
x = a:h:b;
```

```

% Вычисление значений функции f(x) для каждого x
y = x.^3 - 7*x.^2 + 14*x - 8;

% Построение графика с зелёной пунктирной линией и круглыми маркерами
plot(x, y, 'g:o')

% hold on - для нескольких графиков на одном холсте
hold on

% Построение графика с использованием fplot
fplot(@(x) x.^3 - 7*x.^2 + 14*x - 8, [a, b], 'b-', 'MeshDensity', 200) % fplot
автоматически выбирает и адаптирует шаг на основании MeshDensity

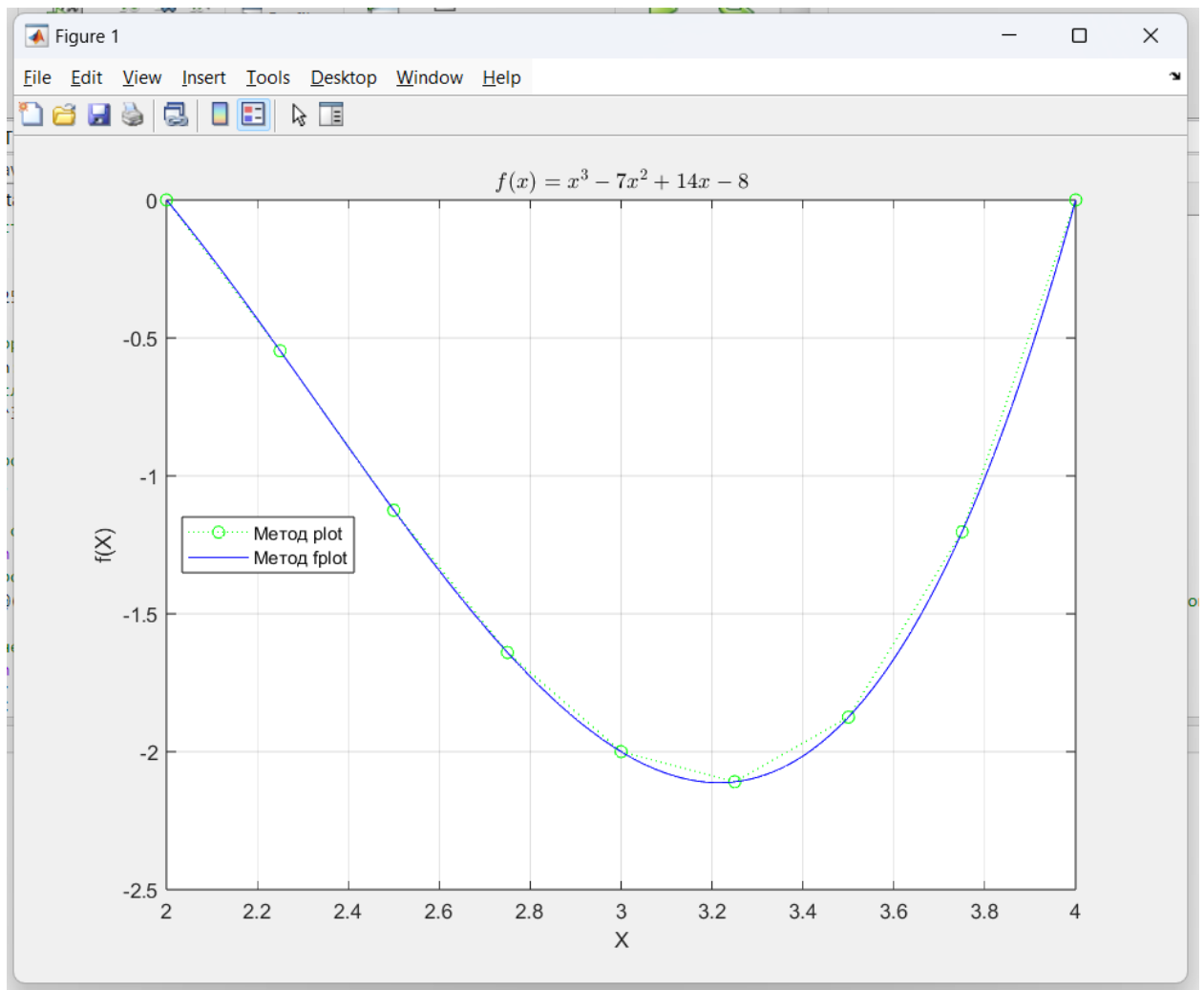
% Включение сетки и подпись осей
grid on
xlabel('X')
ylabel('f(X)')

% Заголовок с использованием TeX
title('$$$f(x) = x^3 - 7x^2 + 14x - 8$$$','Interpreter','latex')

% Легенда с пояснением метода построения графика
legend('Метод plot', 'Метод fplot', 'Location', 'west', 'Orientation', 'vertical')

hold off

```



2.

2. Построение графиков функций двух переменных в среде MatLab

С помощью функции `meshgrid()` сгенерируйте сетку для построения графика (поверхности) функции $f(x_1, x_2)$ в области определения $x_1 \in [a; b]$, $x_2 \in [c; d]$ с шагом изменения переменных h .

MatLab Code:

```
% Задание областей определения и шага
```

```
a = -5;
```

```
b = 5;
```

```
c = -5;
```

```
d = 5;
```

```
h = 0.02;
```

```
% Создание сетки значений с использованием meshgrid
```

```
[x1, x2] = meshgrid(a:h:b, c:h:d);
```

```
% Определение функции Экли
```

```
f = 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) - exp(0.5 * (cos(2 * pi * x1) + cos(2 * pi * x2))));
```

2.1. Поверхность с каркасом и заливкой

1) С помощью функции `mesh()` постройте график функции $f(x_1, x_2)$ в виде каркасной поверхности, задайте прозрачность каркаса (команда `hidden off`) (0.2 балла).

2) Выведите график функции в виде поверхности с цветовой заливкой с помощью функции `surf()`. Скройте каркас (команда `shading flat`), подключите опцию плавной заливки поверхности цветом в зависимости от значения функции (команда `shading interp`), задайте цветовую палитру `cologmap()` на свой вкус (см. Приложение 2), выведите цветовую шкалу рядом с графиком (команда `cologbar`) (0.3 балла).

3) Представьте график функции с линиями уровня, спроецированными на плоскость xOy , с помощью функций `meshc()` или `surfc()` (0.2 балла).

4) Постройте поверхность с цветовой заливкой и адаптивным подбором плотности сетки с помощью функции:

```
fsurf(@fun, [xmin xmax ymin ymax], LineSpec, 'ShowContours', 'on').
```

Задайте значения параметра `LineSpec` на свой вкус. Подключите проекцию линий уровня на плоскость xOy (0.3 балла).

MatLab Code:

```
% Задание областей определения и шага
a = -5;
b = 5;
c = -5;
d = 5;
h = 0.02;

% Создание сетки значений с использованием meshgrid
[x1, x2] = meshgrid(a:h:b, c:h:d);

% Определение функции Экли
f = 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) - exp(0.5 * (cos(2 * pi * x1) + cos(2 * pi * x2))));

%% 1) Каркасная поверхность с прозрачностью
figure;
mesh(x1, x2, f)
hidden off; % Убирает закрашивание
xlabel('X1')
ylabel('X2')
zlabel('f(X1, X2)')
title('Каркасная поверхность функции Экли')

%% 2) Поверхность с цветовой заливкой
```

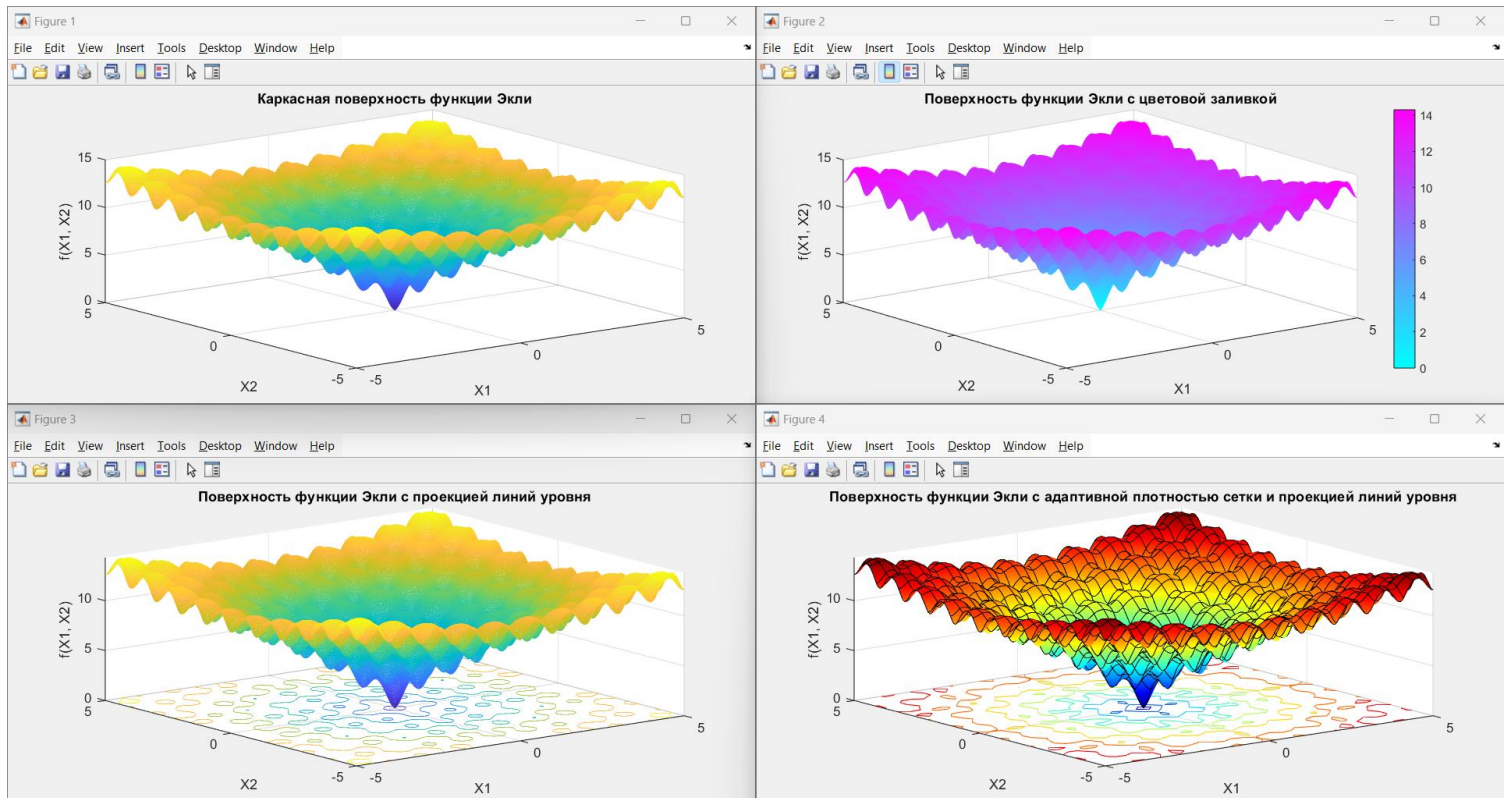
```
figure;
surf(x1, x2, f)
shading flat % Скроем каркас
shading interp % Плавная заливка
colormap('cool') % Выбор цветовой палитры
colorbar % Вывод цветовой шкалы
xlabel('X1')
ylabel('X2')
zlabel('f(X1, X2)')
title('Поверхность функции Экли с цветовой заливкой')
%% 3) График функции с проекцией линий уровня
```

```
figure;
meshc(x1, x2, f) % Построение каркасного графика с проекцией линий
уровня
```

```
xlabel('X1')
ylabel('X2')
zlabel('f(X1, X2)')
title('Поверхность функции Экли с проекцией линий уровня')
```

```
%% 4) Поверхность с адаптивным подбором плотности и проекцией линий
уровня
```

```
figure;
fsurf(@(x1, x2) 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) -
exp(0.5 * (cos(2 * pi * x1) + cos(2 * pi * x2))), ...
[a b c d], '-', 'ShowContours', 'on')
colormap('jet') % Задание цветовой палитры
xlabel('X1')
ylabel('X2')
zlabel('f(X1, X2)')
title('Поверхность функции Экли с адаптивной плотностью сетки и
проекцией линий уровня')
```



2.2. Поверхность, состоящая из линий уровня

С помощью функции `contour3()` постройте поверхность функции $f(x_1, x_2)$ в виде линий уровня следующими способами:

- 1) задайте число линий уровня в четвёртом параметре `contour3()` (0.2 балла);
- 2) задайте аппликаты для построения равноотстоящих линий уровня в виде диапазона с указанием шага (например, `levels=6.5:0.5:15`) (0.4 балла);
- 3) задайте вектор со значениями функции $f(x_1, x_2)$, в которых нужно отобразить линии уровня (0.4 балла).

MatLab Code:

```
% Задание областей определения и шага
```

```
a = -5;
```

```
b = 5;
```

```
c = -5;
```

```
d = 5;
```

```
h = 0.02;
```

```
% Создание сетки значений с использованием meshgrid
```

```
[x1, x2] = meshgrid(a:h:b, c:h:d);
```

```
% Определение функции Экли
```

```
f = 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) - exp(0.5 * (cos(2 * pi * x1) + cos(2 * pi * x2)));
```

```
%% 1) Построение линий уровня с заданием числа уровней
```

```
figure;
```

```
contour3(x1, x2, f, 15) % Задание числа уровней
```

```
xlabel('X1')
```

```
ylabel('X2')
```

```
zlabel('f(X1, X2)')
```

```
title('Линии уровня функции Экли (задано число уровней)')
```

```
%% 2) Построение линий уровня с равноотстоящими значениями
```

```
figure;
```

```
levels = 0:1.5:15; % Уровни от -1 до 20 с шагом 1
```

```
contour3(x1, x2, f, levels)
```

```
xlabel('X1')
```

```
ylabel('X2')
```

```
zlabel('f(X1, X2)')
```

```
title('Линии уровня функции Экли (равноотстоящие уровни)')
```

```
%% 3) Построение линий уровня по заданным значениям функции
```

```
figure;
```

```
specific_levels = [2, 4, 8, 12]; % Вектор заданных значений уровней
```

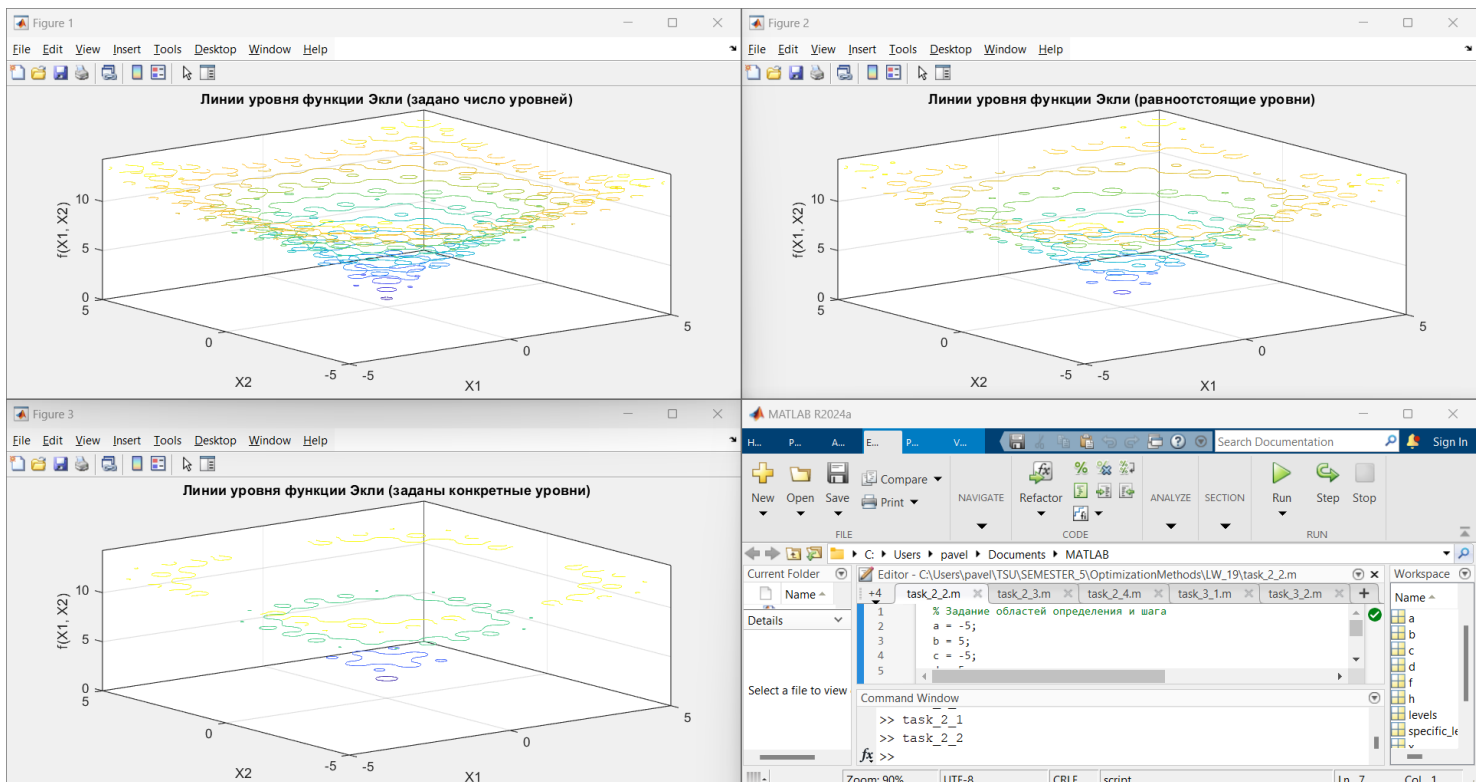
```
contour3(x1, x2, f, specific_levels)
```

```
xlabel('X1')
```

```
ylabel('X2')
```

```
zlabel('f(X1, X2)')
```

```
title('Линии уровня функции Экли (заданы конкретные уровни)')
```



2.3. *Изолинии на плоскости*

1) Постройте линии уровня функции $f(x_1, x_2)$ на плоскости с помощью функции `contour()`, отобразите координатную сетку (команда `grid on`) (0.2 балла).

2) Постройте изолинии $f(x_1, x_2)$ с цветовой заливкой на плоскости с помощью функции `contourf()`, отобразите координатную сетку (команда `grid on`), задайте цветовую палитру `colormap()` на свой вкус, выведите цветовую шкалу (команда `colorbar`) (0.3 балла).

Подпишите изолинии соответствующими значениями функции $f(x_1, x_2)$. Для этого используйте метод `clabel(CMatr, h)`. Наиболее подходящие значения параметров `CMatr` и `h` для этого метода можно получить, если предварительно вызвать функцию `contourf()` с двумя выходными параметрами, например, `[CMatr, h]=contourf(x,y,z)` (0.5 балла).

MatLab Code:

```
% Задание областей определения и шага
```

```
a = -5;
```

```
b = 5;
```

```
c = -5;
```

```
d = 5;
```

```
h = 0.02;
```

```
% Создание сетки значений с использованием meshgrid
```

```
[x1, x2] = meshgrid(a:h:b, c:h:d);
```

```
% Определение функции Экли
```

```
f = 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) - exp(0.5 * (cos(2 * pi * x1) + cos(2 * pi * x2))));
```

```
%% 1) Линии уровня на плоскости с использованием функции contour
```

```
figure;
```

```
contour(x1, x2, f, 20) % 20 линий уровня
```

```
grid on % Включаем сетку
```

```
xlabel('X1')
```

```
ylabel('X2')
```

```
title('Линии уровня функции Экли')
```

```
%% 2) Изолинии с цветовой заливкой на плоскости
```

```
figure;
```

```
[CMatr, h] = contourf(x1, x2, f, 10); % 20 изолиний с заливкой и выходными параметрами
```

```
grid on % Включаем сетку
```

```
colormap('pink') % Устанавливаем цветовую палитру
```

```
colorbar % Выводим цветовую шкалу
```

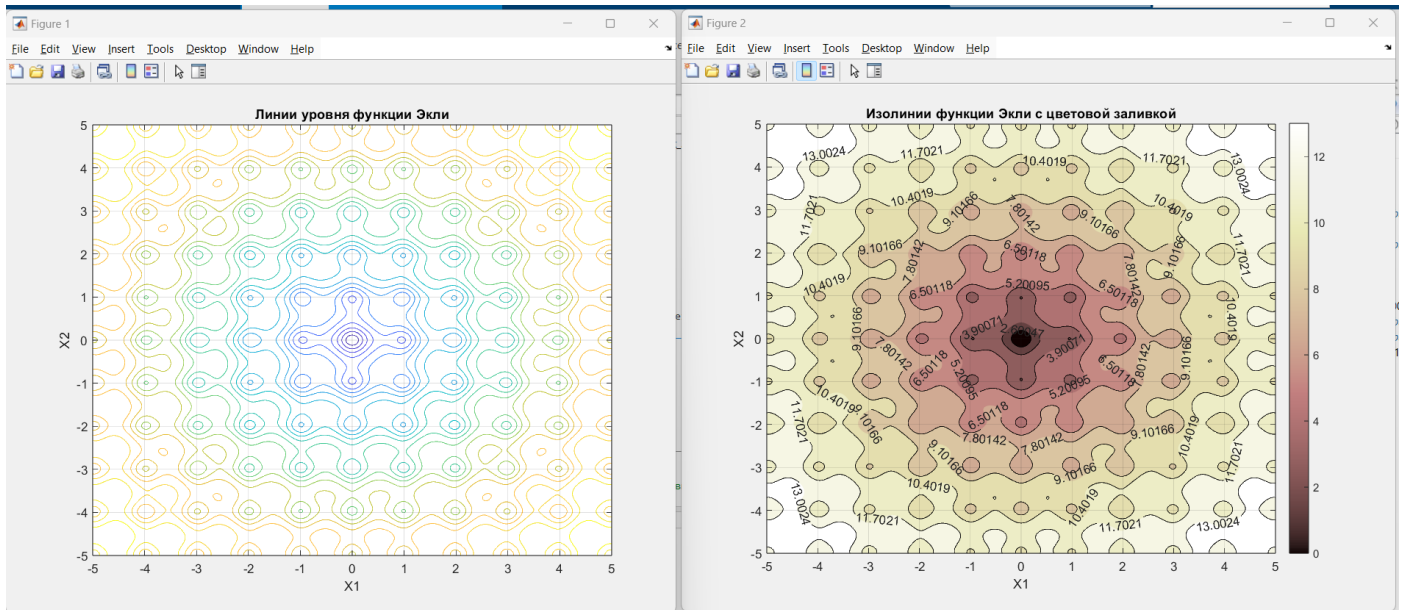
```
xlabel('X1')
```

```
ylabel('X2')
```

```
title('Изолинии функции Экли с цветовой заливкой')
```

```
% Подпись изолиний
```

```
clabel(CMatr, h) % Добавляем значения уровня на изолинии
```



2.4. Освещённая поверхность

1) С помощью функции `surf()` постройте освещённую поверхность функции $f(x_1, x_2)$ (0.25 балла).

2) Задайте команду `[Az, El]=view`, которая выдаст текущее положение наблюдателя, то есть азимут (Az) и угол его возвышения (El). Азимут отсчитывается от оси, противоположной y , а угол возвышения от плоскости xOy . По умолчанию $Az = -37.5^\circ$, $El = 30^\circ$. С помощью команды `view(Az, El)` поменяйте положение наблюдателя так, чтобы обзор поверхности был максимально удобным и информативным (например, чтобы было меньше скрытых частей поверхности). Например, `view(135, 45)` (0.25 балла).

3) Задайте одну из цветовых палитр (`correr`, `bone`, `gray`, `pink`), в которых интенсивность цвета изменяется линейно. Для получения плавно изменяющихся оттенков используйте команду `shading interp`. Подпишите оси (`xlabel()`, `ylabel()`, `zlabel()`) (0.25 балла).

4) В четвёртом аргументе `surf()` в виде вектора-строки задайте азимут и угол возвышения источника освещения. Например, измените текущий азимут на -90° (по умолчанию источник света имеет азимут, больший на 45° , чем наблюдатель, и тот же угол возвышения) (0.25 балла).

MatLab Code:

```
% Задание областей определения и шага
```

```
a = -5;
```

```
b = 5;
```

```
c = -5;
```

```
d = 5;
```

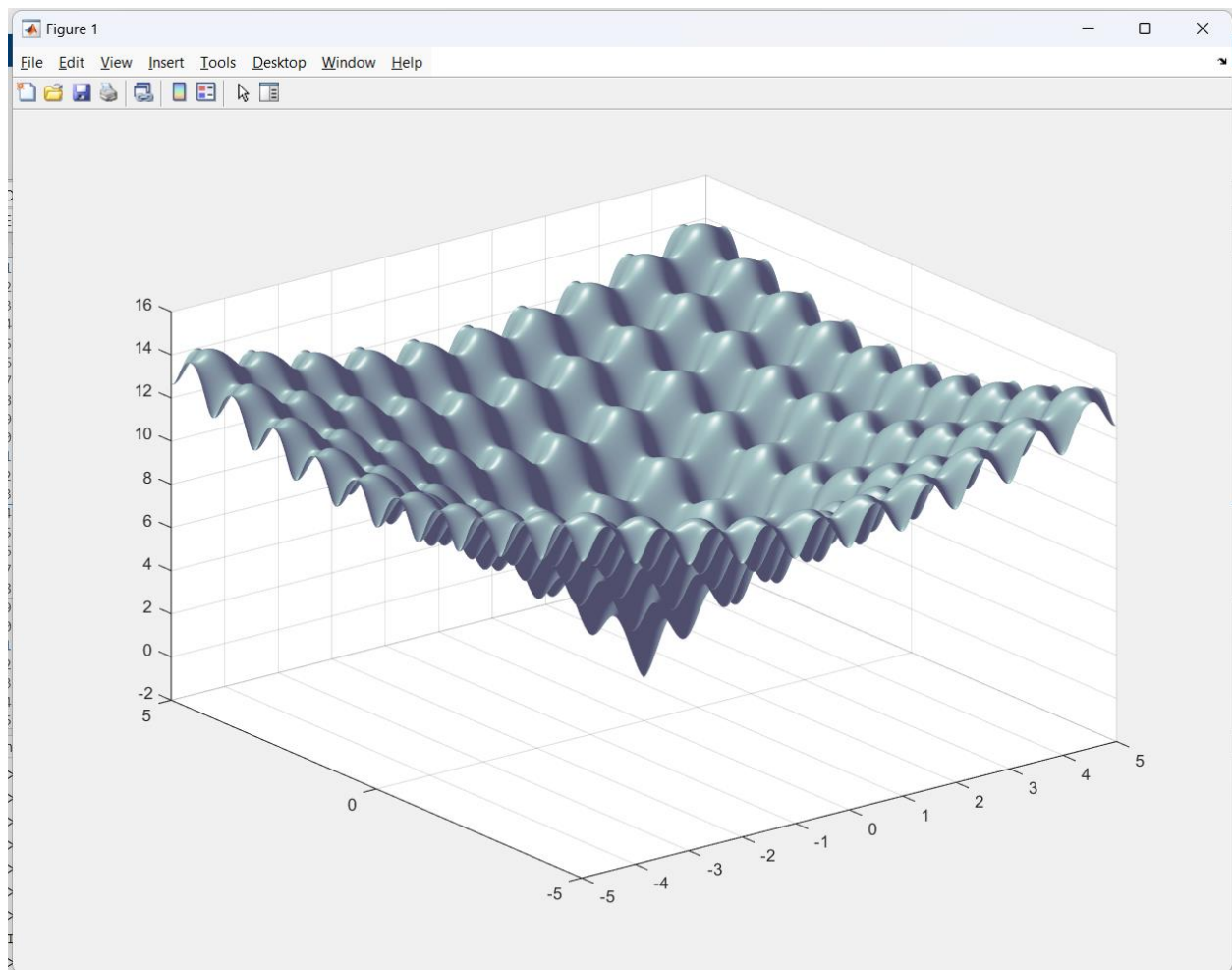
```
h = 0.02;
```

```
% Создание сетки значений с использованием meshgrid
```

```

[x1, x2] = meshgrid(a:h:b, c:h:d);
% Определение функции Экли
f = 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) - exp(0.5 * (cos(2 * pi * x1)
+ cos(2 * pi * x2)));
%% 1) Построение освещённой поверхности
figure;
surf(x1, x2, f)
shading interp % Плавное изменение цвета
%% 2) Определение текущего положения наблюдателя
[Az, El] = view; % Получение текущих значений азимута и угла возвышения
disp(['Текущий азимут: ', num2str(Az), '°, угол возвышения: ', num2str(El), '°'])
% Установка нового положения наблюдателя для лучшего обзора
view(135, 45) % Изменённые значения азимута и угла возвышения
%% 3) Настройка цветовой палитры и подпись осей
colormap bone % Установка цветовой палитры (можно заменить на bone, gray, pink)
xlabel('X1')
ylabel('X2')
zlabel('f(X1, X2)')
%% 4) Изменение параметров освещения
surf(x1, x2, f, [-90, 45]) % Задание азимута -90° и угла возвышения 45° для
источника света
shading interp % Плавное изменение оттенков цвета

```



3.

3. Визуализация точек экстремума

3.1. На трёхмерном графике функции $f(x_1, x_2)$ из задания 2 подпишите экстремальные точки. Для отыскания экстремумов воспользуйтесь тем, что значения функции в узлах сетки хранятся в матрице, и примените функции `min()` и `max()`. С помощью функции `find()` найдите строчные и столбцовые индексы экстремальных элементов матрицы. Например, `[r,c] = find(z == min(min(z)))`. Завершающий этап – вызов функции `text()`, где в качестве значений параметров укажите координаты минимальной или максимальной точки, а также надпись «Минимум» или «Максимум» (1 балл).

MatLab Code:

```
% Задание областей определения и шага
```

```
a = -5;
```

```
b = 5;
```

```
c = -5;
```

```
d = 5;
```

```

h = 0.02;

% Создание сетки значений с использованием meshgrid
[x1, x2] = meshgrid(a:h:b, c:h:d);

% Определение функции Экли
f = 20 + exp(1) - 20 * exp(-0.2 * sqrt(0.5 * (x1.^2 + x2.^2))) - exp(0.5 * (cos(2 * pi * x1)
+ cos(2 * pi * x2)));

% Построение 3D-графика функции
figure;
surf(x1, x2, f)
shading interp % Плавное изменение цвета
colormap bone % Установка цветовой палитры
colorbar % Отображение цветовой шкалы
xlabel('X1')
ylabel('X2')
zlabel('f(X1, X2)')
title('3D-график функции Экли с обозначением экстремумов')

% Нахождение минимального значения и его индексов
minValue = min(f(:)); % Минимальное значение функции
[rMin, cMin] = find(f == minValue); % Индексы всех минимальных значений

% Нахождение максимального значения и его индексов
maxValue = max(f(:)); % Максимальное значение функции
[rMax, cMax] = find(f == maxValue); % Индексы всех максимальных значений

% min(f(:)) и max(f(:)) вычисляют минимальное и максимальное значения во всей
матрице f.

% find(f == minValue) и find(f == maxValue) возвращают индексы всех позиций в
матрице f,

% где находятся минимальные и максимальные значения. Переменные rMin, cMin,
rMax, и
% cMax содержат строковые и столбцовые индексы этих позиций.

% заливка точки ('MarkerFaceColor', 'color').

% Подписи на графике
hold on

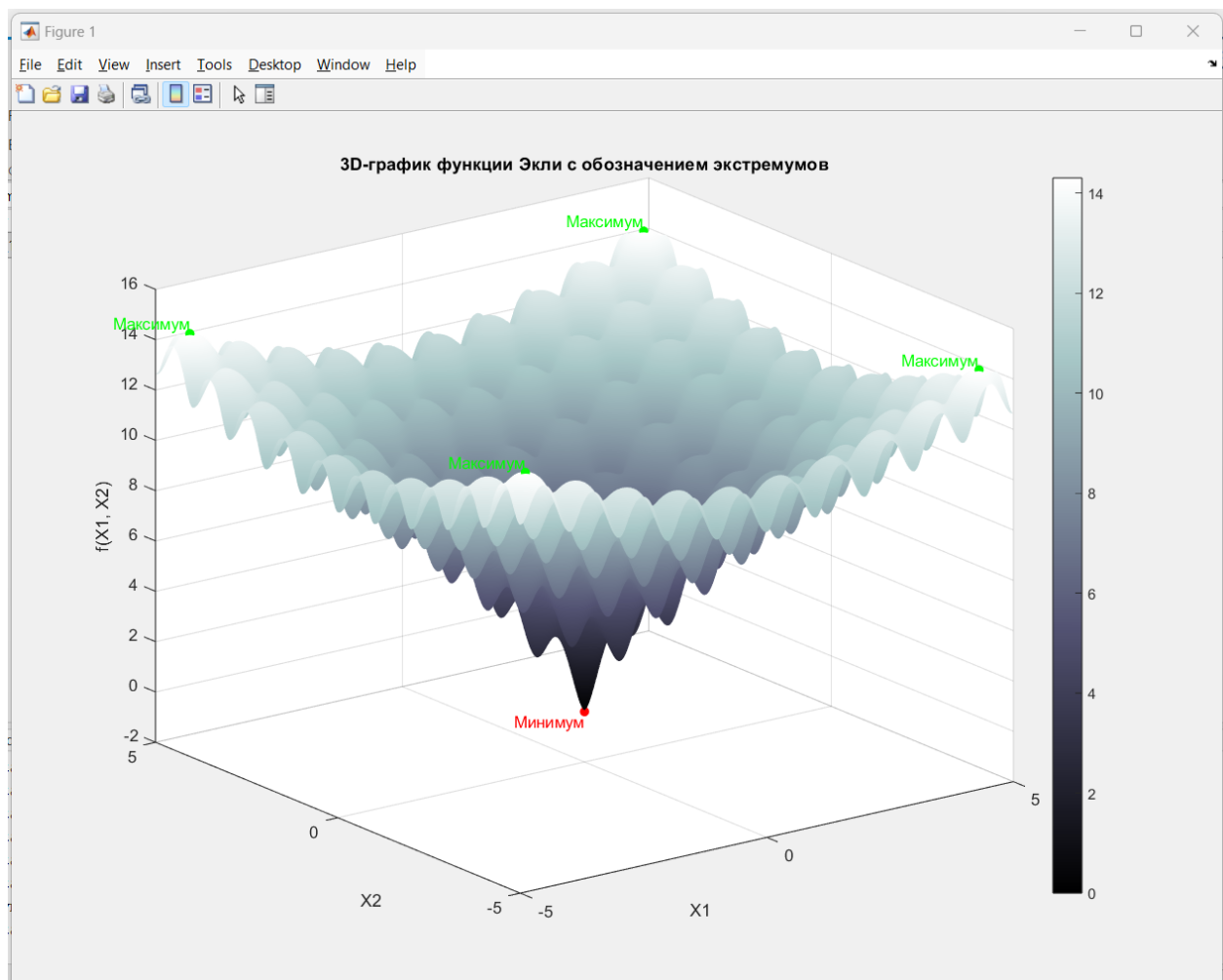
% Отображение всех минимумов
for i = 1:length(rMin)
    xMin = x1(rMin(i), cMin(i));

```

```

yMin = x2(rMin(i), cMin(i));
zMin = f(rMin(i), cMin(i));
plot3(xMin, yMin, zMin, 'r.', 'MarkerSize', 20) % Обозначение минимума
text(xMin, yMin, zMin, 'Минимум', 'VerticalAlignment', 'top', 'HorizontalAlignment',
'right', 'Color', 'red')
end
% Отображение всех максимумов
for i = 1:length(rMax)
xMax = x1(rMax(i), cMax(i));
yMax = x2(rMax(i), cMax(i));
zMax = f(rMax(i), cMax(i));
plot3(xMax, yMax, zMax, 'g.', 'MarkerSize', 20) % Обозначение максимума
text(xMax, yMax, zMax, 'Максимум', 'VerticalAlignment', 'bottom',
'HorizontalAlignment', 'right', 'Color', 'green')
end
hold off

```



3.2. Аналогичную задачу решите для функции $f(x)$ из задания 1 (1 бонус).

MatLab Code:

```
% область определения и шаг
a = 2;
b = 4;
h = 0.1;
% Вектор значений x в заданном диапазоне с шагом h (массив иксов)
x = a:h:b;
% Вычисление значений функции f(x) для каждого x
f = x.^3 - 7*x.^2 + 14*x - 8;
% Построение графика функции
figure;
plot(x, f, '-k.', 'LineWidth', 1.5, 'MarkerSize', 15)
grid on
xlabel('x')
ylabel('f(x)')
title('$f(x) = x^3 - 7x^2 + 14x - 8$', 'Interpreter', 'latex')
% Нахождение всех минимальных и максимальных значений
minValue = min(f); % Минимальное значение функции
maxValue = max(f); % Максимальное значение функции
% Индексы всех точек с минимальными и максимальными значениями
minIndices = find(f == minValue);
maxIndices = find(f == maxValue);
% Добавление маркеров и подписей для всех минимумов
hold on
for i = 1:length(minIndices)
    xMin = x(minIndices(i));
    plot(xMin, minValue, 'rv', 'MarkerSize', 10, 'MarkerFaceColor', 'red') % Точка
    минимума
    text(xMin, minValue, 'Минимум', 'VerticalAlignment', 'top', 'HorizontalAlignment',
'right', 'Color', 'red')
end
% Добавление маркеров и подписей для всех максимумов
for i = 1:length(maxIndices)
    xMax = x(maxIndices(i));
```

```

plot(xMax, maxValue, 'b^', 'MarkerSize', 10, 'MarkerFaceColor', 'blue') % Точка
максимума
text(xMax, maxValue, 'Максимум', 'VerticalAlignment', 'bottom',
'HorizontalAlignment', 'right', 'Color', 'blue')
end
hold off

```

