**МАИ**

**Лабораторная работа №8**

Инструментальное средство EDITOR/DEBUGGER

Вариант №14

**Выполнил:**

Студент группы М7О-114БВ-24

Фельдман Лев Борисович

**Проверил:**Доцент Кафедры 702 Козлова Н.М.

Ассистент Кафедры 702 Милославский Я.Г.

Москва 2024

Цель: Знакомство с инструментальным средством Editor/Debugger, а также приобретение навыков работы со структурами выбора и повторения языка MATLAB.

Самостоятельная работа.

Самостоятельная работа состоит из двух заданий. Задания необходимо взять из предыдущего практикума согласно своему варианту. Решение каждого задания оформить в виде отдельного сценария.

Задание 1.

Создать сценарий для построения кривой линии из первого задания.

% Параметры спирали

a = 1; % Масштабный коэффициент

b = 0.1; % Коэффициент экспоненты

% Диапазон углов (в радианах)

phi = linspace(0, 4\*pi, 1000); % от 0 до 4\*pi с 1000 точками

% Вычисление радиус-вектора r

r = a .\* exp(b \* phi);

% Перевод в декартовы координаты

x = r .\* cos(phi);

y = r .\* sin(phi);

% Построение графика

figure;

plot(x, y, 'b', 'LineWidth', 1.5);

grid on;

axis equal;

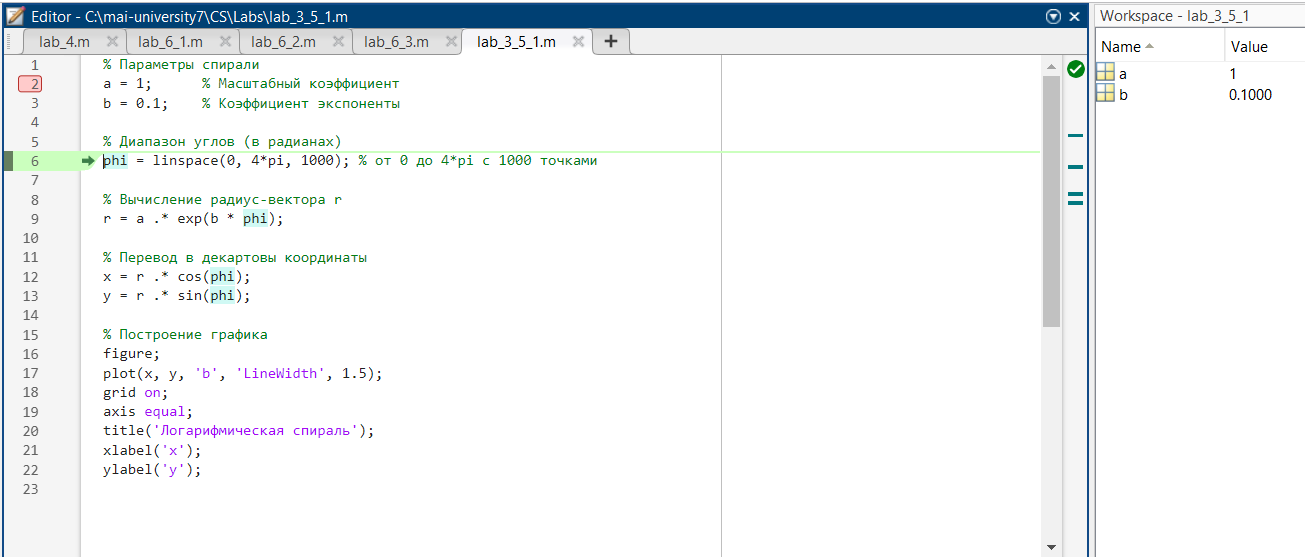
title('Логарифмическая спираль');

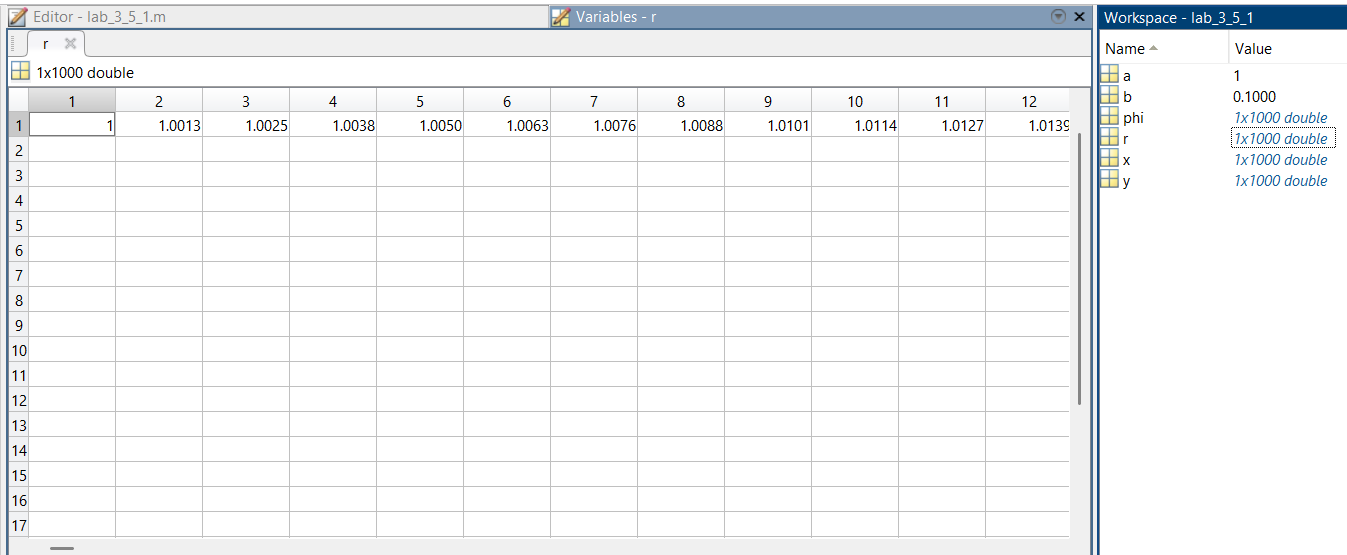
xlabel('x');

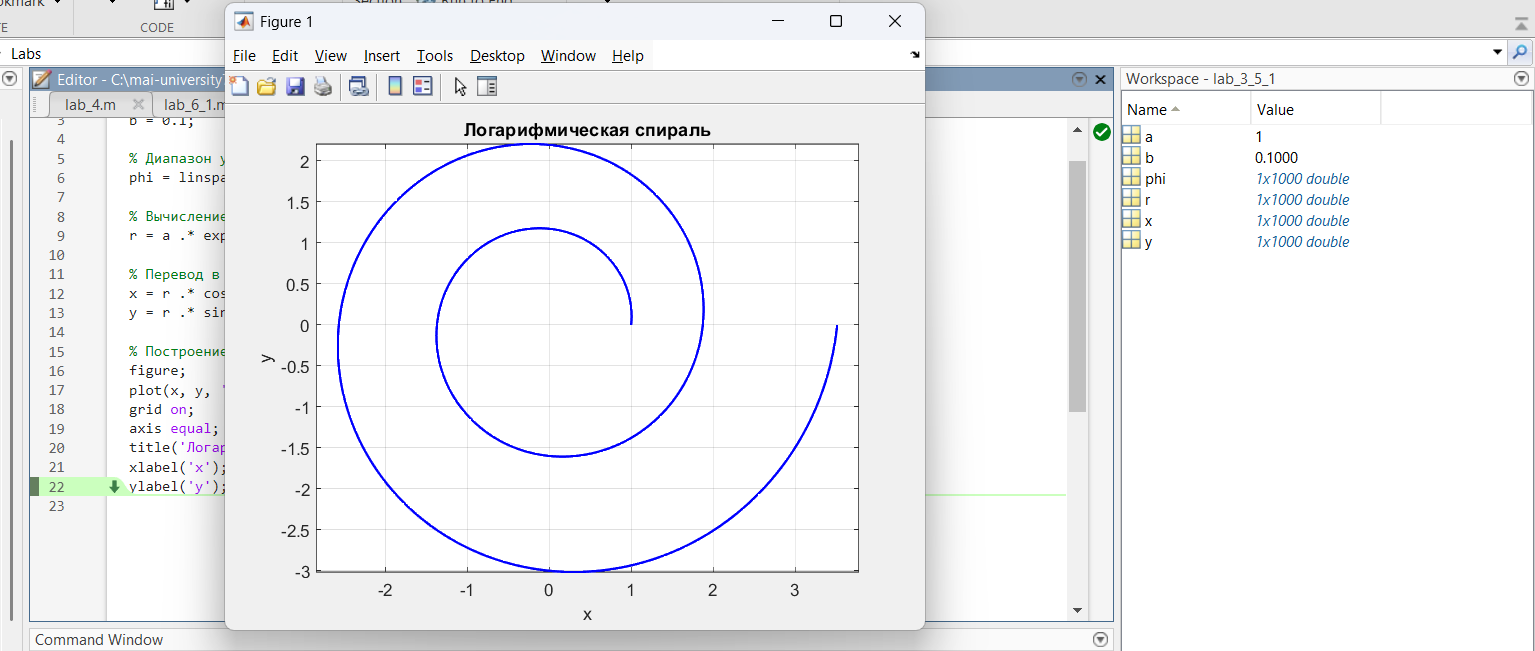
ylabel('y');

Задние 2.

Перейти в режим отладки и последовательно выполнить каждую команду с просмотром значений переменных.







Задание 3

Вывести в графическое окно пять вариантов кривой при различных значениях одного параметра (в случае наличия нескольких параметров выбрать один). Каждую кривую вывести своим цветом. Цвет кривой линии задать с помощью структуры выбора swtch case end.

% Параметры спирали

a\_values = [0.5, 1, 1.5, 2, 2.5]; % Различные значения a

b = 0.1; % Коэффициент экспоненты

% Диапазон углов (в радианах)

phi = linspace(0, 4\*pi, 1000); % от 0 до 4\*pi с 1000 точками

% Создание графика

figure;

hold on;

grid on;

axis equal;

title('Логарифмические спирали для разных значений a');

xlabel('x');

ylabel('y');

% Цвета для разных спиралей

for i = 1:length(a\_values)

a = a\_values(i);

r = a .\* exp(b \* phi);

x = r .\* cos(phi);

y = r .\* sin(phi);

% Определение цвета с помощью switch-case

switch i

case 1

color = 'r'; % Красный

case 2

color = 'g'; % Зелёный

case 3

color = 'b'; % Синий

case 4

color = 'm'; % Пурпурный

case 5

color = 'k'; % Чёрный

otherwise

color = 'c'; % Голубой (на случай непредвиденных ситуаций)

end

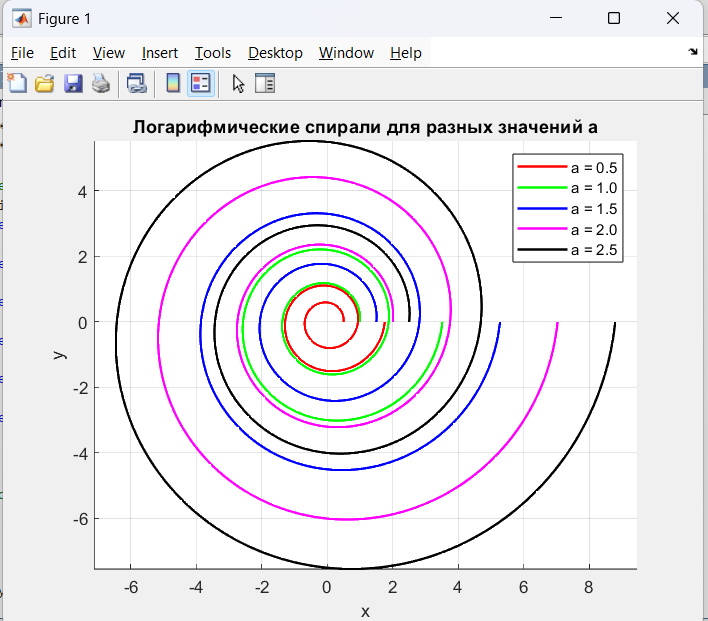
% Построение кривой

plot(x, y, 'Color', color, 'LineWidth', 1.5);

end

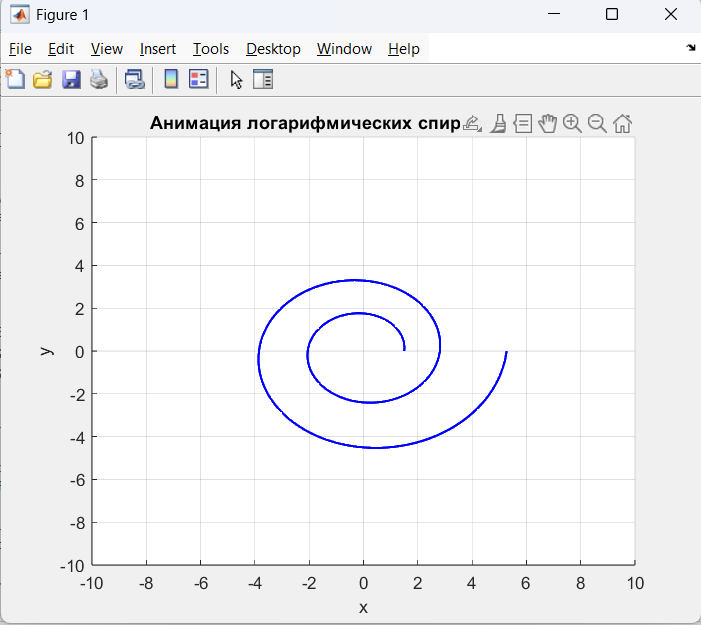
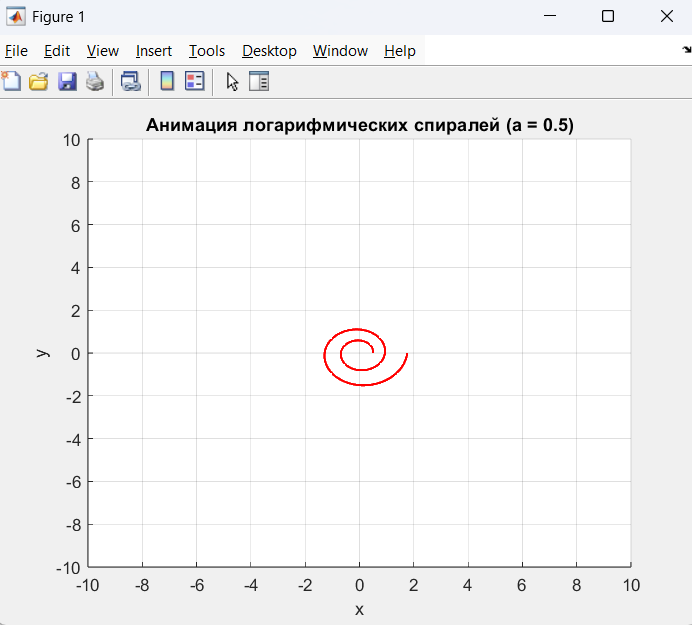
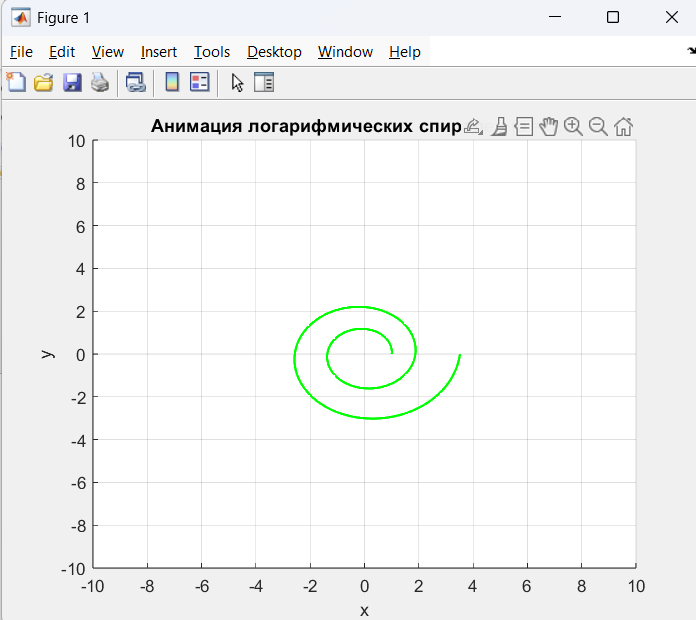
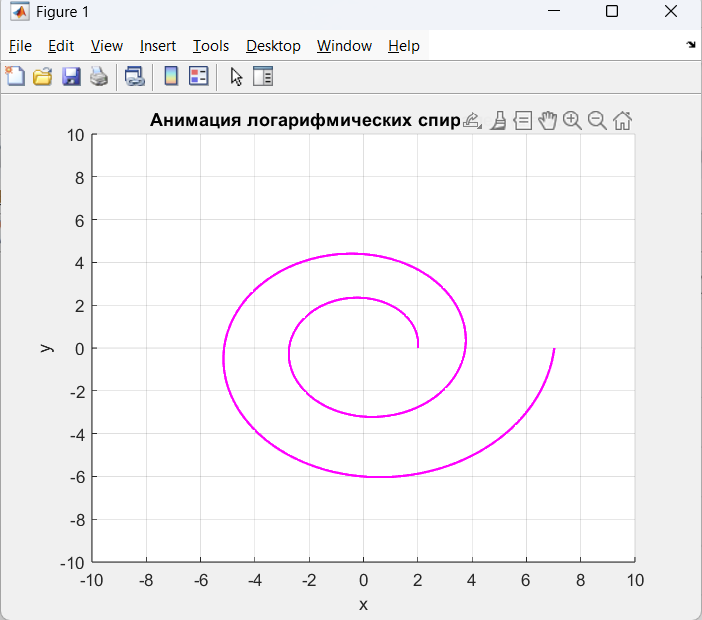
hold off;

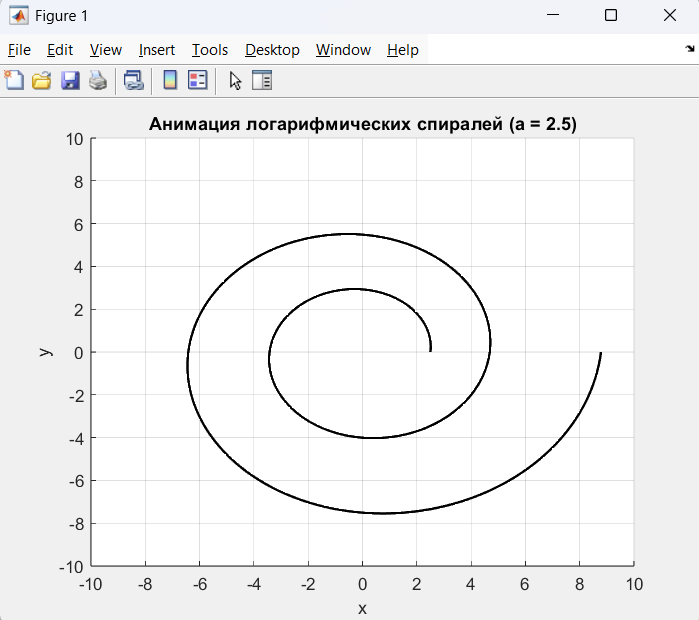
legend(arrayfun(@(a) sprintf('a = %.1f', a), a\_values, 'UniformOutput', false));



Задание 4

Используя структуры повторения for end иwhile end, создать анимацию. При выполнении анимации в графическом окне циклически появляются и исчезают первый вариант линии, второй, третий, четвертый, пятый, четвертый, третий, второй. При использовании первой структуры повторения организовать десять повторов, а в случае второй структуры реализовать бесконечное число повторов.



Задание 5

Написать сценарий для решения второго задания.

% Параметры

a = 1;

b = 2;

c = 1.5;

h = 5;

% Сетка координат

t = linspace(0, 2\*pi, 100); % Угловые значения

z = linspace(-h, h, 100); % Высота по оси z

[T, Z] = meshgrid(t, z);

% Уравнение конуса

X = a \* Z .\* cos(T); % x = a\*z\*cos(t)

Y = b \* Z .\* sin(T); % y = b\*z\*sin(t)

Z = c \* Z; % Масштабирование по оси z

% Построение поверхности

figure;

surf(X, Y, Z, Y); % Ориентация цветов по оси y

colormap(jet); % Цветовая палитра

shading interp; % Интерполяция цвета

axis equal; % Одинаковые масштабы

xlabel('X');

ylabel('Y');

zlabel('Z');

title('Конус второго порядка');

grid on;

view(45, 30); % Азимут 45°, высота 30°

Задание 6

Создать анимацию, в которой выполняется вращение изображения.

%% Исходные данные и построение конуса

% Параметры

a = 1;

b = 2;

c = 1.5;

h = 5;

% Сетка координат

t = linspace(0, 2\*pi, 100); % угловые значения

z = linspace(-h, h, 100); % высота по оси z

[T, Z\_grid] = meshgrid(t, z);

% Уравнение конуса

X0 = a \* Z\_grid .\* cos(T); % x = a\*z\*cos(t)

Y0 = b \* Z\_grid .\* sin(T); % y = b\*z\*sin(t)

Z0 = c \* Z\_grid; % масштабирование по оси z

%% Анимация 4: Последовательное вращение вокруг осей Z, Y, X

figure;

hSurface = surf(X0, Y0, Z0, Y0); % отображение поверхности (цвет по оси Y)

colormap(jet);

shading interp;

axis equal;

% Первоначальные подписи осей (будут заменяться в процессе анимации)

xlabel('X'); ylabel('Y'); zlabel('Z');

title('Конус. Последовательное вращение (Z -> Y -> X)');

grid on;

view(45,30);

hold on;

angleStep = 1; % шаг изменения угла (в градусах)

pauseTime = 0.005; % уменьшенная задержка для ускорения анимации

% Инициализация базовых координат для последовательного поворота

X\_base = X0;

Y\_base = Y0;

Z\_base = Z0;

%% Вращение вокруг оси Z (обновляем подпись оси Z)

hZLabel = get(gca, 'ZLabel'); % получаем объект подписи оси Z

for angle = 0:angleStep:360

theta = deg2rad(angle);

Rz = [cos(theta) -sin(theta) 0;

sin(theta) cos(theta) 0;

0 0 1];

coords = [X\_base(:)'; Y\_base(:)'; Z\_base(:)'];

newCoords = Rz \* coords;

X\_rot = reshape(newCoords(1,:), size(X0));

Y\_rot = reshape(newCoords(2,:), size(Y0));

Z\_rot = reshape(newCoords(3,:), size(Z0));

set(hSurface, 'XData', X\_rot, 'YData', Y\_rot, 'ZData', Z\_rot);

if mod(angle,45) == 0

set(hZLabel, 'Color', rand(1,3)); % изменение цвета подписи оси Z

end

set(hZLabel, 'String', sprintf('%d°', angle));

drawnow;

pause(pauseTime);

end

% Обновляем базу после вращения вокруг Z

X\_base = X\_rot;

Y\_base = Y\_rot;

Z\_base = Z\_rot;

%% Вращение вокруг оси Y (обновляем подпись оси Y)

hYLabel = get(gca, 'YLabel'); % получаем объект подписи оси Y

for angle = 0:angleStep:360

theta = deg2rad(angle);

Ry = [ cos(theta) 0 sin(theta);

0 1 0;

-sin(theta) 0 cos(theta)];

coords = [X\_base(:)'; Y\_base(:)'; Z\_base(:)'];

newCoords = Ry \* coords;

X\_rot = reshape(newCoords(1,:), size(X0));

Y\_rot = reshape(newCoords(2,:), size(Y0));

Z\_rot = reshape(newCoords(3,:), size(Z0));

set(hSurface, 'XData', X\_rot, 'YData', Y\_rot, 'ZData', Z\_rot);

if mod(angle,45) == 0

set(hYLabel, 'Color', rand(1,3));

end

set(hYLabel, 'String', sprintf('%d°', angle));

drawnow;

pause(pauseTime);

end

% Обновляем базу после вращения вокруг Y

X\_base = X\_rot;

Y\_base = Y\_rot;

Z\_base = Z\_rot;

%% Вращение вокруг оси X (обновляем подпись оси X)

hXLabel = get(gca, 'XLabel'); % получаем объект подписи оси X

for angle = 0:angleStep:360

theta = deg2rad(angle);

Rx = [1 0 0;

0 cos(theta) -sin(theta);

0 sin(theta) cos(theta)];

coords = [X\_base(:)'; Y\_base(:)'; Z\_base(:)'];

newCoords = Rx \* coords;

X\_rot = reshape(newCoords(1,:), size(X0));

Y\_rot = reshape(newCoords(2,:), size(Y0));

Z\_rot = reshape(newCoords(3,:), size(Z0));

set(hSurface, 'XData', X\_rot, 'YData', Y\_rot, 'ZData', Z\_rot);

if mod(angle,45) == 0

set(hXLabel, 'Color', rand(1,3));

end

set(hXLabel, 'String', sprintf('%d°', angle));

drawnow;

pause(pauseTime);

end

