Table of Contents

Assignemnt

Newton's Interpolation

```
% Name         : Mohamed Mafaz
% Roll Number  : AM25M009
% Department   : Applied Mechanics

clc
clear
close all
```

Part 1 (Preprocessing)

Lorenz system parameters

```
sigma = 10;
rho = 28;
beta = 8/3;

% Lorenz equations
fx_t = @(x, y) sigma * (y - x);
fy_t = @(x, y, z) x * (rho - z) - y;
fz_t = @(x, y, z) (x * y) - (beta * z);

% Initial conditions
x_ini = 3;
y_ini = 3;
z_ini = 20;

% Time setup
t_final = 100;
h = 0.01;
t = 0:h:t_final;
N = length(t);

% Allocate arrays
x = zeros(1, N);
y = zeros(1, N);
z = zeros(1, N);
x_rk = zeros(1, N);
y_rk = zeros(1, N);
z_rk = zeros(1, N);
```

```
% Set initial conditions
x(1) = x_ini;
y(1) = y_ini;
z(1) = z_ini;
x_rk(1) = x_ini;
y_rk(1) = y_ini;
z_rk(1) = z_ini;
```

Part 2 (Processing / Using the Algorithms)

```
% Euler Method
fprintf('Running Euler method...\n');
tic;
for i = 1:N-1
    x(i+1) = x(i) + h * fx_t(x(i), y(i));
    y(i+1) = y(i) + h * fy_t(x(i), y(i), z(i));
    z(i+1) = z(i) + h * fz_t(x(i), y(i), z(i));
end
time_euler = toc;
fprintf('Euler time: %f seconds\n', time_euler);

% RK4 Method
fprintf('Running RK4 method...\n');
tic;
for i = 1:N-1
    % k1
    k1x = fx_t(x_rk(i), y_rk(i));
    k1y = fy_t(x_rk(i), y_rk(i), z_rk(i));
    k1z = fz_t(x_rk(i), y_rk(i), z_rk(i));

    % k2
    k2x = fx_t(x_rk(i) + 0.5*h*k1x, y_rk(i) + 0.5*h*k1y);
    k2y = fy_t(x_rk(i) + 0.5*h*k1x, y_rk(i) + 0.5*h*k1y, z_rk(i) +
0.5*h*k1z);
    k2z = fz_t(x_rk(i) + 0.5*h*k1x, y_rk(i) + 0.5*h*k1y, z_rk(i) +
0.5*h*k1z);

    % k3
    k3x = fx_t(x_rk(i) + 0.5*h*k2x, y_rk(i) + 0.5*h*k2y);
    k3y = fy_t(x_rk(i) + 0.5*h*k2x, y_rk(i) + 0.5*h*k2y, z_rk(i) +
0.5*h*k2z);
    k3z = fz_t(x_rk(i) + 0.5*h*k2x, y_rk(i) + 0.5*h*k2y, z_rk(i) +
0.5*h*k2z);

    % k4
    k4x = fx_t(x_rk(i) + h*k3x, y_rk(i) + h*k3y);
    k4y = fy_t(x_rk(i) + h*k3x, y_rk(i) + h*k3y, z_rk(i) + h*k3z);
    k4z = fz_t(x_rk(i) + h*k3x, y_rk(i) + h*k3y, z_rk(i) + h*k3z);

    % Update
    x_rk(i+1) = x_rk(i) + (h/6)*(k1x + 2*k2x + 2*k3x + k4x);
    y_rk(i+1) = y_rk(i) + (h/6)*(k1y + 2*k2y + 2*k3y + k4y);
    z_rk(i+1) = z_rk(i) + (h/6)*(k1z + 2*k2z + 2*k3z + k4z);
end
time_rk4 = toc;
fprintf('RK4 time: %f seconds\n', time_rk4);
```

```
Running Euler method...
Euler time: 0.025191 seconds
Running RK4 method...
RK4 time: 0.129673 seconds
```

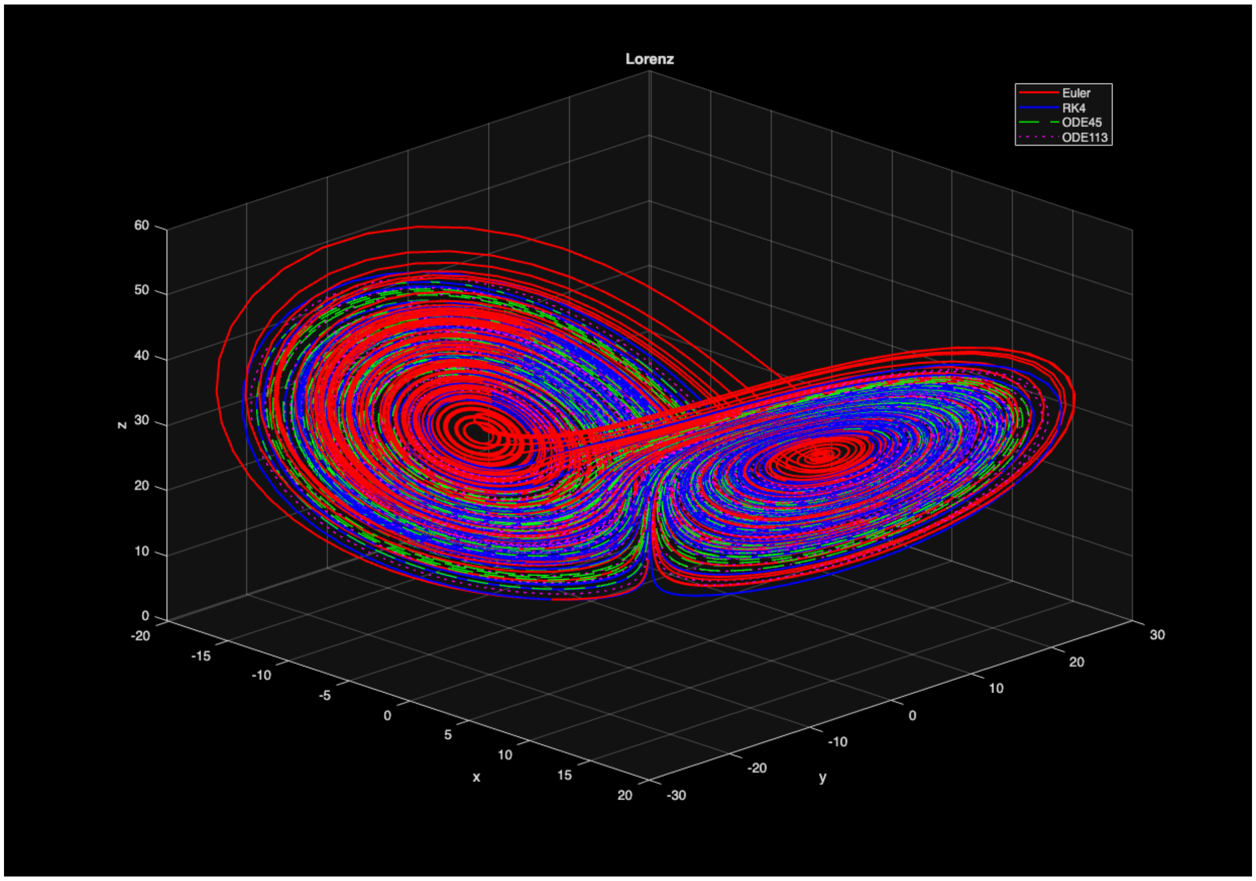
Part 3 (Post Processing / Plotting)

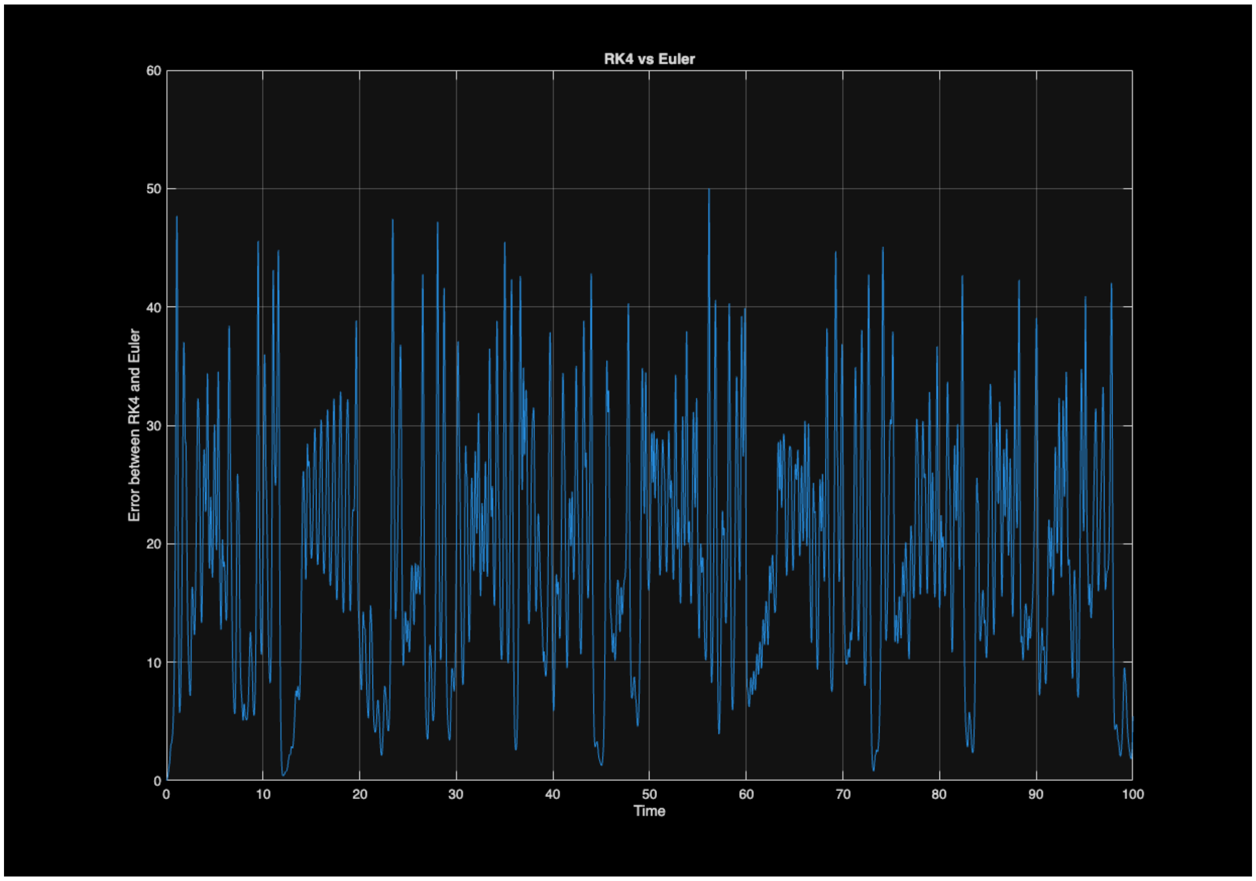
```
% Compare with ODE45
fprintf('Running ODE45...\n');
lorenz_system = @(t, Y) [sigma*(Y(2)-Y(1)); Y(1)*(rho-Y(3))-Y(2); Y(1)*Y(2)-
beta*Y(3)];
options = odeset('AbsTol', 1e-6, 'RelTol', 1e-6);
tic;
[t_ode45, Y_ode45] = ode45(lorenz_system, [0 t_final], [x_ini; y_ini;
z_ini], options);
time_ode45 = toc;
fprintf('ODE45 time: %f seconds\n', time_ode45);

% Compare with ODE113
fprintf('Running ODE113...\n');
tic;
[t_ode113, Y_ode113] = ode113(lorenz_system, [0 t_final], [x_ini; y_ini;
z_ini], options);
time_ode113 = toc;
fprintf('ODE113 time: %f seconds\n', time_ode113);

% Plots
figure(1);
plot3(x, y, z, 'r-', 'LineWidth', 1.5);
hold on;
plot3(x_rk, y_rk, z_rk, 'b-', 'LineWidth', 1.5);
plot3(Y_ode45(:,1), Y_ode45(:,2), Y_ode45(:,3), 'g--', 'LineWidth', 1);
plot3(Y_ode113(:,1), Y_ode113(:,2), Y_ode113(:,3), 'm:', 'LineWidth', 1);
xlabel('x');
ylabel('y');
zlabel('z');
title('Lorenz');
legend('Euler', 'RK4', 'ODE45', 'ODE113', 'Location', 'best');
grid on;
view(45, 30);

% Calculate errors between RK4 and Euler
errors = zeros(1, N);
for i = 1:N
    errors(i) = sqrt((x(i) - x_rk(i))^2 + (y(i) - y_rk(i))^2 + (z(i) -
z_rk(i))^2);
end
```

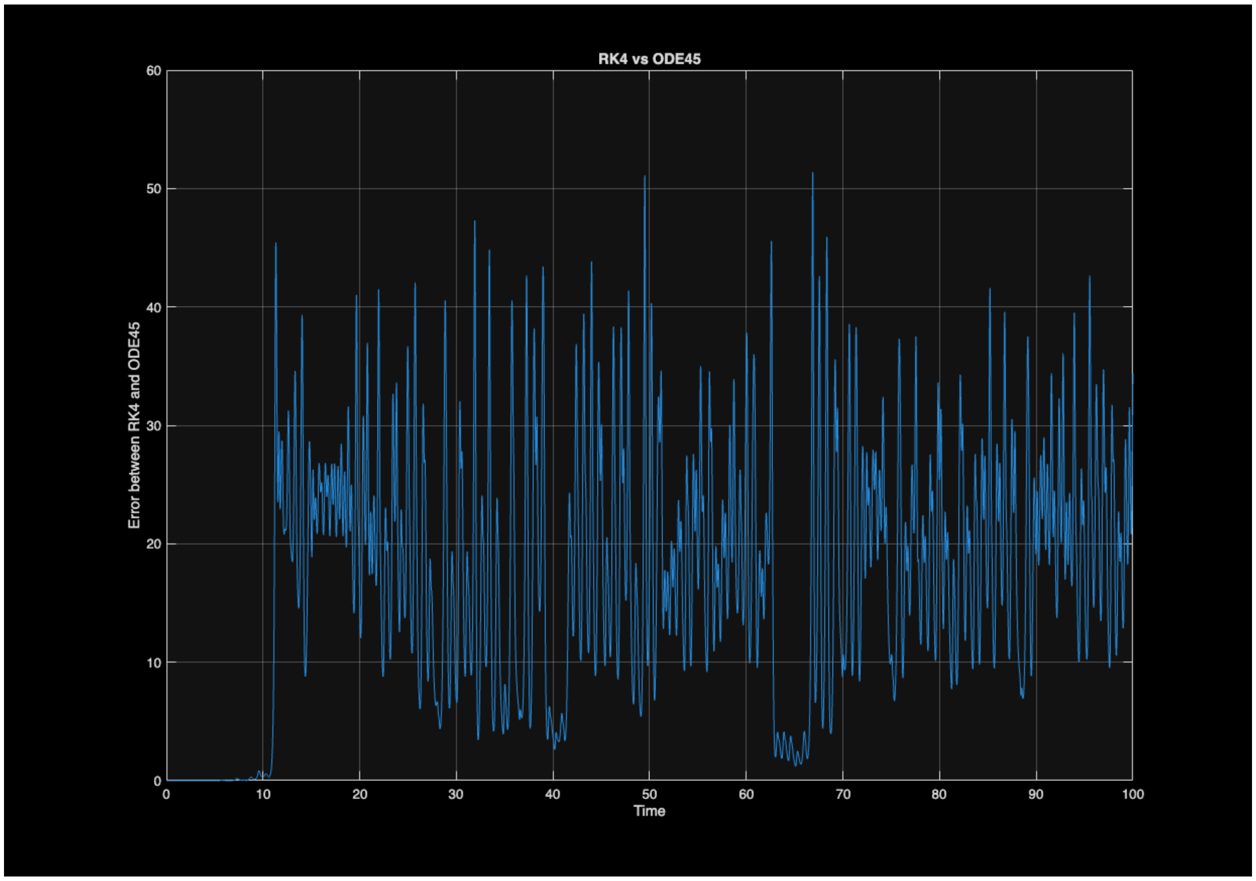
```
% Interpolate ODE45 results to match our time grid
Y_ode45_interp = interp1(t_ode45, Y_ode45, t, 'linear');
errors_rk4_vs_ODE45 = zeros(1, N);
for i = 1:N
    errors_rk4_vs_ODE45(i) = sqrt((Y_ode45_interp(i, 1) - x_rk(i))^2 +
(Y_ode45_interp(i, 2) - y_rk(i))^2 + (Y_ode45_interp(i, 3) - z_rk(i))^2);
end

% Interpolate ODE113 results to match our time grid
Y_ode113_interp = interp1(t_ode113, Y_ode113, t, 'linear');
errors_rk4_vs_ODE113 = zeros(1, N);
for i = 1:N
    errors_rk4_vs_ODE113(i) = sqrt((Y_ode113_interp(i, 1) - x_rk(i))^2 +
(Y_ode113_interp(i, 2) - y_rk(i))^2 + (Y_ode113_interp(i, 3) - z_rk(i))^2);
end

figure(2);
plot(t, errors);
xlabel('Time');
ylabel('Error between RK4 and Euler');
title('RK4 vs Euler');
grid on;

figure(3);
plot(t, errors_rk4_vs_ODE45);
xlabel('Time');
ylabel('Error between RK4 and ODE45');
title('RK4 vs ODE45');
grid on;

figure(4);
plot(t, errors_rk4_vs_ODE113);
xlabel('Time');
ylabel('Error between RK4 and ODE113');
title('RK4 vs ODE113');
grid on;


% Show timing results
fprintf('\n=== TIMING COMPARISON ===\n');
fprintf('Euler:  %f seconds\n', time_euler);
fprintf('RK4:    %f seconds\n', time_rk4);
fprintf('ODE45:  %f seconds\n', time_ode45);
fprintf('ODE113: %f seconds\n', time_ode113);


% Final values at t=100
fprintf('\n=== VALUES AT t=100 ===\n');
fprintf('Euler:  x=%f, y=%f, z=%f\n', x(end), y(end), z(end));
fprintf('RK4:    x=%f, y=%f, z=%f\n', x_rk(end), y_rk(end), z_rk(end));
fprintf('ODE45:  x=%f, y=%f, z=%f\n', Y_ode45(end,1), Y_ode45(end,2),
Y_ode45(end,3));
fprintf('ODE113: x=%f, y=%f, z=%f\n', Y_ode113(end,1), Y_ode113(end,2),
Y_ode113(end,3));
```

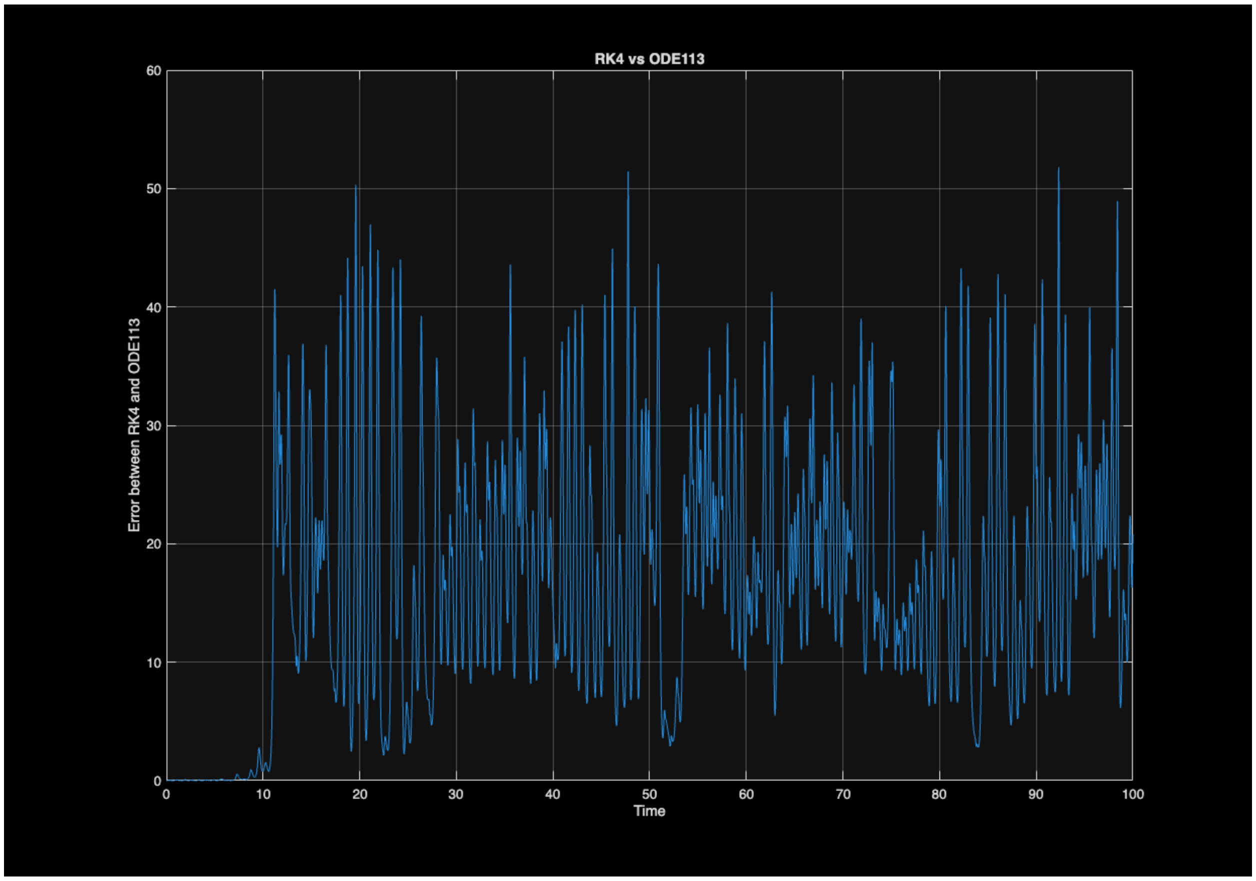
```
% Test with tighter tolerance
fprintf('\n=== TESTING WITH TIGHTER TOLERANCE ===\n');
options_tight = odeset('AbsTol', 1e-10, 'RelTol', 1e-10);
tic;
[t_ode45_tight, Y_ode45_tight] = ode45(lorenz_system, [0 t_final], [x_ini;
y_ini; z_ini], options_tight);
time_ode45_tight = toc;
fprintf('ODE45 (tight): %f seconds\n', time_ode45_tight);
fprintf('Final values: x=%.10f, y=%.10f, z=%.10f\n', Y_ode45_tight(end,1),
Y_ode45_tight(end,2), Y_ode45_tight(end,3));
```

```
Running ODE45...
ODE45 time: 0.046469 seconds
Running ODE113...
ODE113 time: 0.059556 seconds

=== TIMING COMPARISON ===
Euler:  0.025191 seconds
RK4:    0.129673 seconds
ODE45:  0.046469 seconds
ODE113: 0.059556 seconds

=== VALUES AT t=100 ===
Euler:  x=-17.561295, y=-17.509485, z=40.286522
RK4:    x=-15.703918, y=-17.437323, z=35.188276
ODE45:  x=3.439145, y=5.407364, z=17.939183
ODE113: x=-5.046758, y=-5.681900, z=21.786153

=== TESTING WITH TIGHTER TOLERANCE ===
ODE45 (tight): 0.130138 seconds
Final values: x=-8.8036125270, y=-12.8092875380, z=21.2678783936
```


Lorenz
Euler
RK4
ODE45
ODE113
x
y
z


RK4 vs Euler
Error between RK4 and Euler
Time
0
10
20
30
40
50
60
0
10
20
30
40
50
60
70
80
90
100


RK4 vs ODE45
Error between RK4 and ODE45
Time
0
10
20
30
40
50
60
0
10
20
30
40
50
60
70
80
90
100



Published with MATLAB® R2025a