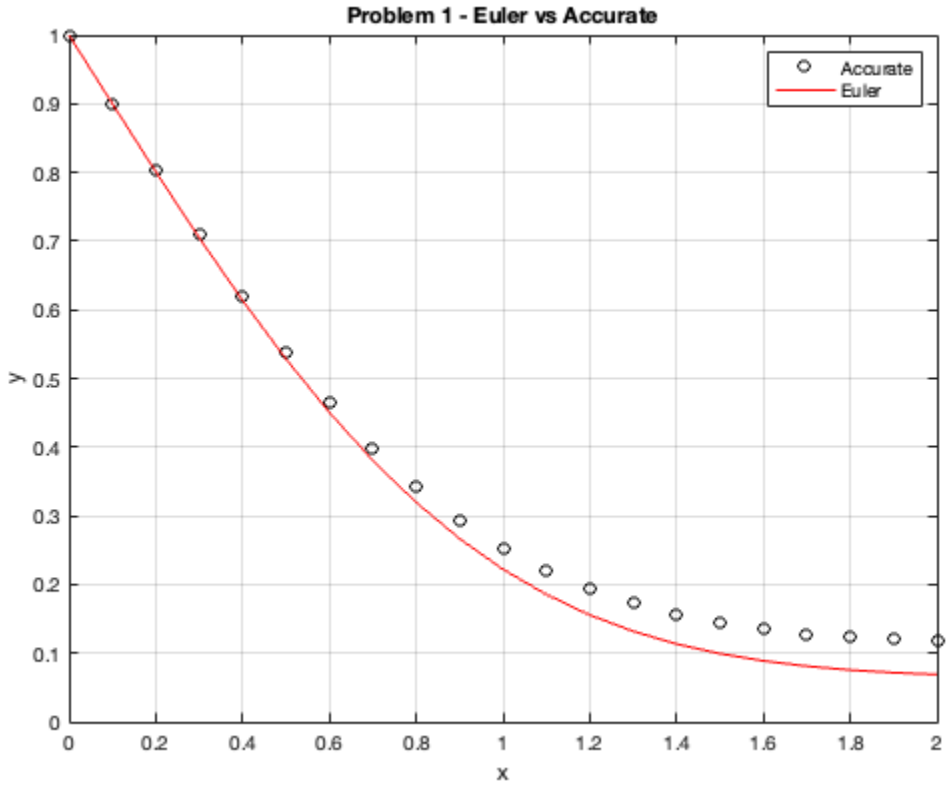
Table of Contents

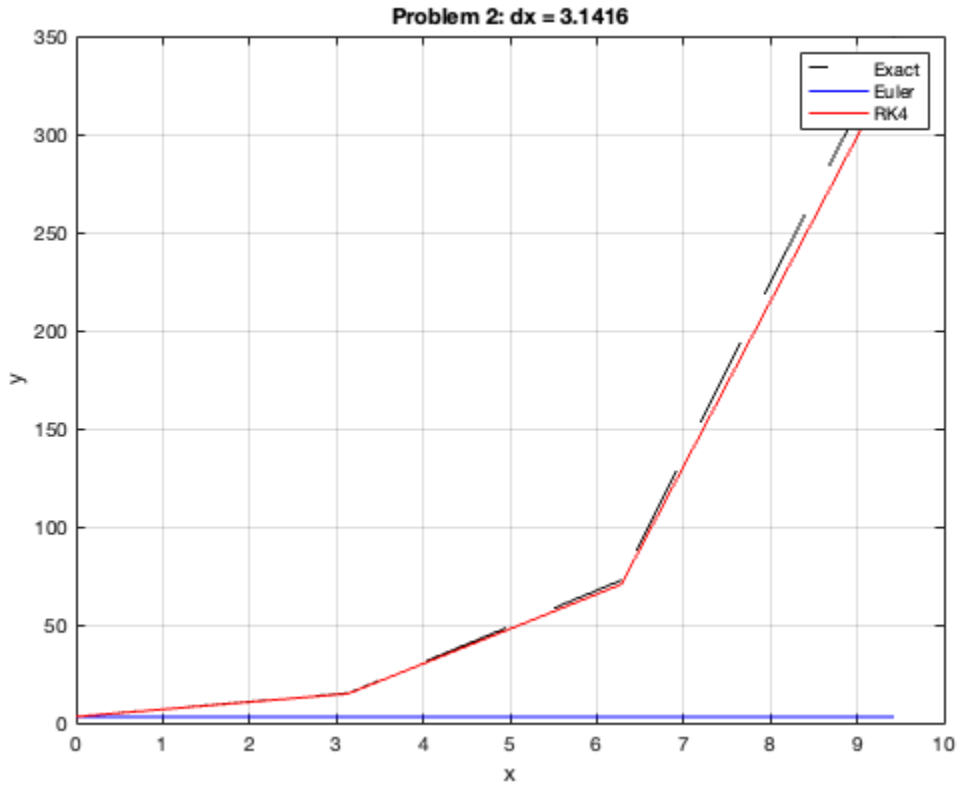

```
%{
Author: Ratnakaru Yalagathala
Assignment:ASEN 2012 - Coding Challenge 2
Creation Date: March 15 2025
Inputs: AccurateDataSP25.mat , Function Handles and IC
Outputs: Figures comparing Euler vs. Accurate Data and Euler vs. RK4 vs. Exact
solution
Purpose: The purpose of this assinment is to compare Euler and RK4
integration for some differential equations.
% Also calculates RMSEs and stores answers in a structure.
%}


clear;
clc;
close all;

% PROBLEM 1 - Euler Integration
% dy/dx = -exp(-x^2), y(0) = 1, x = 0 to 2, dx = 0.1

g1 = @(x,y) -exp(-x.^2);      % function handle
x0 = 0;
y0 = 1;
xf1 = 2;
dx1 = 0.1;

[x1, y1] = euler_integration(g1, [x0 y0], xf1, dx1);

% First load the accurate data from ode45
load('AccurateDataSP25.mat');

% Interpolate onto Euler grid
newy_accurate = interp1(x_accurate, y_accurate, x1, 'linear', 'extrap');

% RMSE calculation
RMSEp1 = sqrt(mean((y1 - newy_accurate).^2));

% Save answers
answers.yf = str2double(sprintf('%.4g', y1(end)));
answers.RMSEp1 = str2double(sprintf('%.4g', RMSEp1));

fprintf('Euler final y at x=2: %.4f\n', answers.yf);
fprintf('RMSE vs accurate data: %.4f\n', answers.RMSEp1);

% Plot it
figure;
plot(x1, newy_accurate, 'ko'); hold on;
```

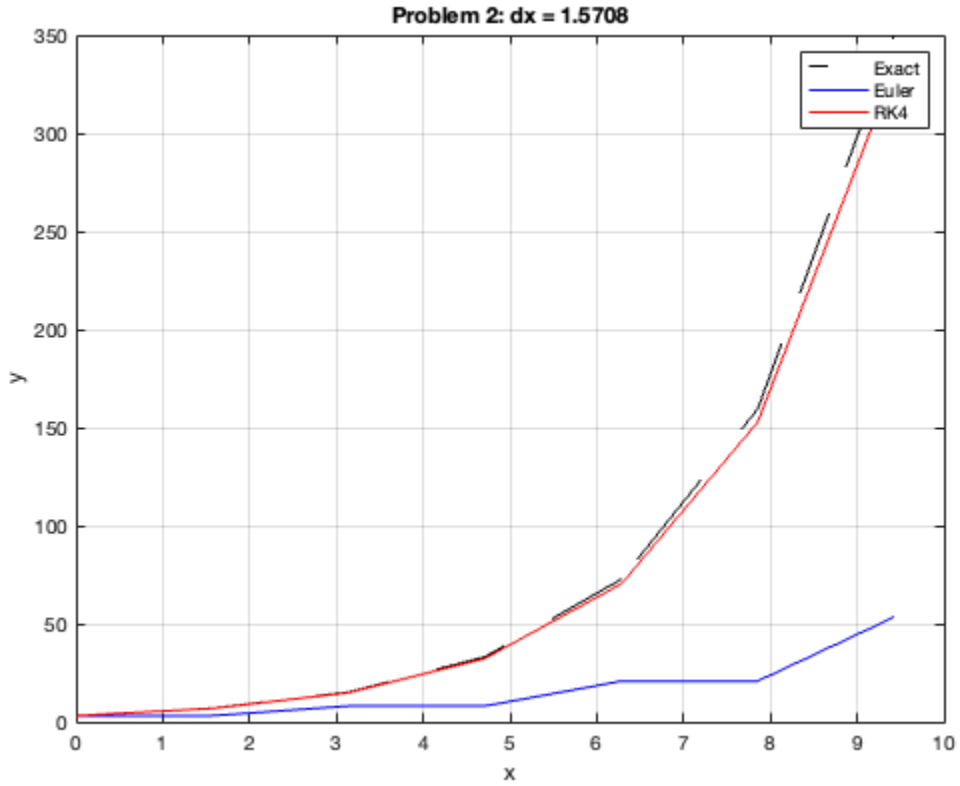
```
plot(x1, y1, 'r');
xlabel('x'); ylabel('y');
legend('Accurate','Euler');
title('Problem 1 - Euler vs Accurate');
grid on;

% PROBLEM 2 - Euler vs RK4
% dy/dx = y*sin^2(x), y(0)=pi, x = 0 to 3pi

g2 = @(x,y) y.*(sin(x).^2);
y_exact = @(x) pi * exp(x/2 - sin(2*x)/4);
x0 = 0;
y0 = pi;
xf2 = 3*pi;
dx_vals = [pi, pi/2, pi/4, pi/8, pi/16];

RMSEEuler = zeros(1,5);
RMSErk4 = zeros(1,5);

for i = 1:length(dx_vals)
    dx = dx_vals(i);

    [x_e, y_e] = euler_integration(g2, [x0 y0], xf2, dx);
    [x_rk, y_rk] = rk4_integration(g2, [x0 y0], xf2, dx);

    y_exact_e = y_exact(x_e);
    y_exact_rk = y_exact(x_rk);  % same x grid

    RMSEEuler(i) = sqrt(mean((y_e - y_exact_e).^2));
    RMSErk4(i)   = sqrt(mean((y_rk - y_exact_rk).^2));

    % Plot
    figure;
    plot(x_e, y_exact_e, 'k--'); hold on;
    plot(x_e, y_e, 'b');
    plot(x_rk, y_rk, 'r');
    title(['Problem 2: dx = ', num2str(dx)]);
    xlabel('x'); ylabel('y');
    legend('Exact','Euler','RK4');
    grid on;
end

% Save final results
answers.RMSEEuler = arrayfun(@(v) str2double(sprintf('%.4g', v)), RMSEEuler);
answers.RMSErk4 = arrayfun(@(v) str2double(sprintf('%.4g', v)), RMSErk4);

fprintf('\nEuler RMSE at dx = pi: %.4f\n', answers.RMSEEuler(1));
fprintf('Euler RMSE at dx = pi/2: %.4f\n', answers.RMSEEuler(2));
fprintf('RK4 RMSE at dx = pi: %.4f\n', answers.RMSErk4(1));


% REFLECTION
%{
1
```

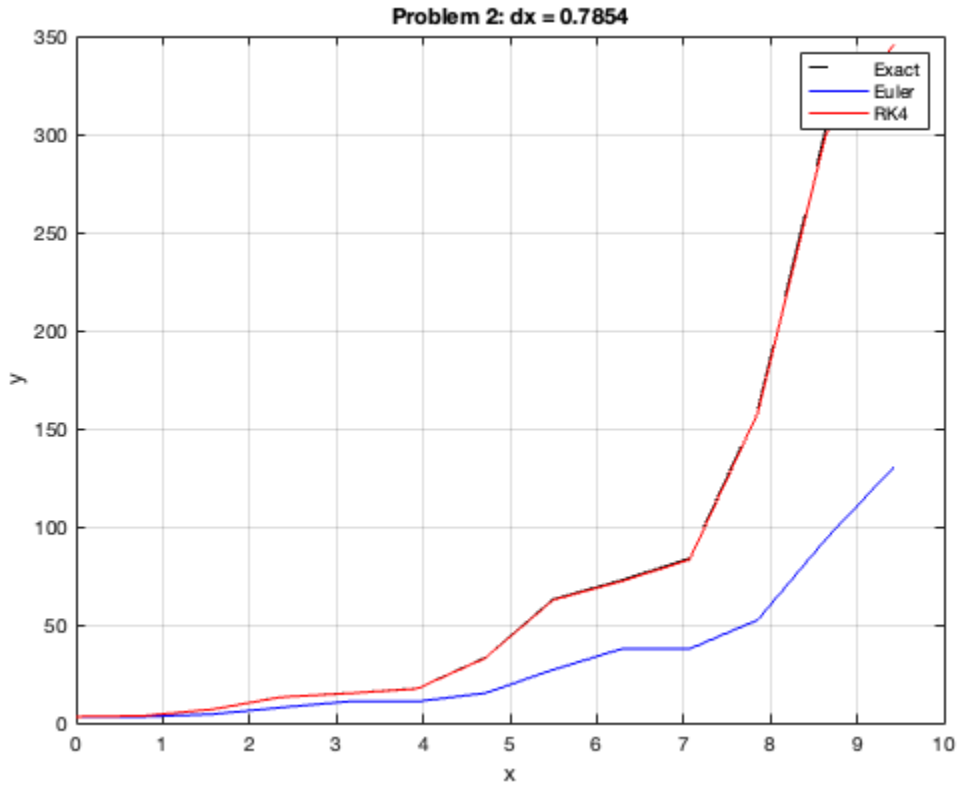
```
We downsampled ode45's data with interp1 so that it would match our Euler
x-points.
ode45 is taking adaptive steps so it's not fixed spacing.

2
Euler error reduces when step size is reduced. RK4 is more accurate with
bigger steps so that it is more cost-saving. To achieve up to RK4's accuracy,
Euler
has a very small step size which is time-consuming.

3
RK4 is more accurate if you can live with more calculations per step..
%}

% ---- Functions ----

function [x, y] = euler_integration(f, init, xf, dx)
    x0 = init(1);
    y0 = init(2);
    x = x0:dx:xf;
    y = zeros(1, length(x));
    y(1) = y0;
    for i = 1:length(x)-1
        y(i+1) = y(i) + f(x(i), y(i)) * dx;
    end
end

function [x, y] = rk4_integration(f, init, xf, dx)
    x0 = init(1);
    y0 = init(2);
    x = x0:dx:xf;
    y = zeros(1, length(x));
    y(1) = y0;
    for i = 1:length(x)-1
        k1 = f(x(i), y(i));
        k2 = f(x(i)+dx/2, y(i)+k1*dx/2);
        k3 = f(x(i)+dx/2, y(i)+k2*dx/2);
        k4 = f(x(i)+dx, y(i)+k3*dx);
        y(i+1) = y(i) + dx*(k1 + 2*k2 + 2*k3 + k4)/6;
    end
end
```

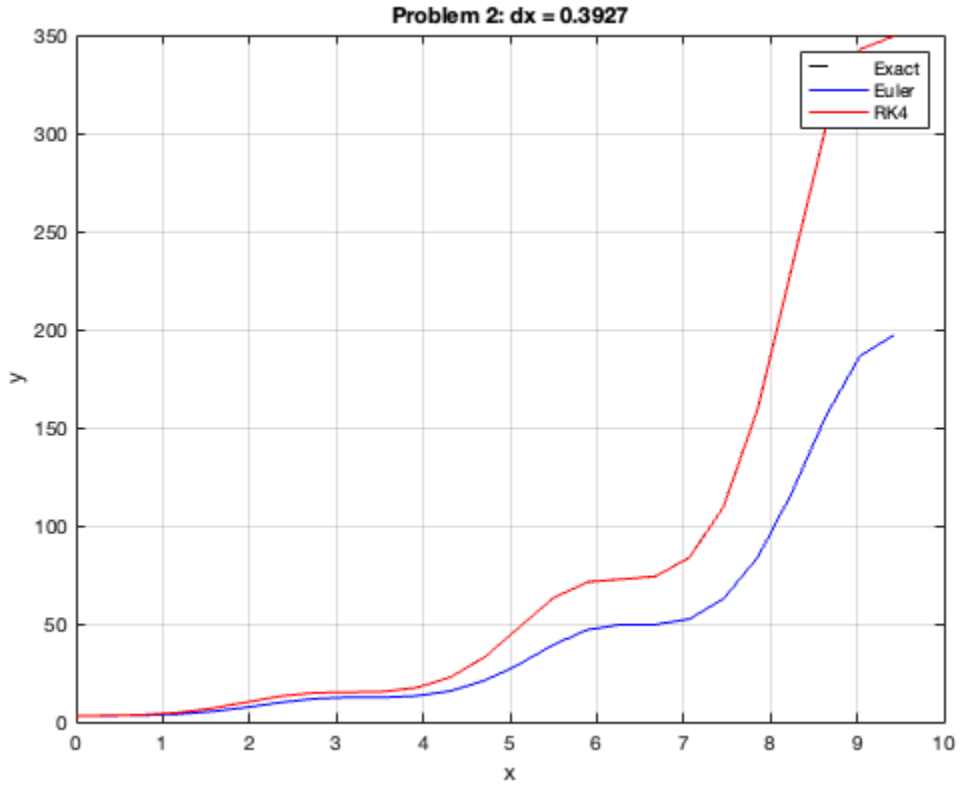
```
Euler final y at x=2: 0.0689
RMSE vs accurate data: 0.0326

Euler RMSE at dx = pi: 176.8000
Euler RMSE at dx = pi/2: 125.6000
RK4 RMSE at dx = pi: 7.7190
```

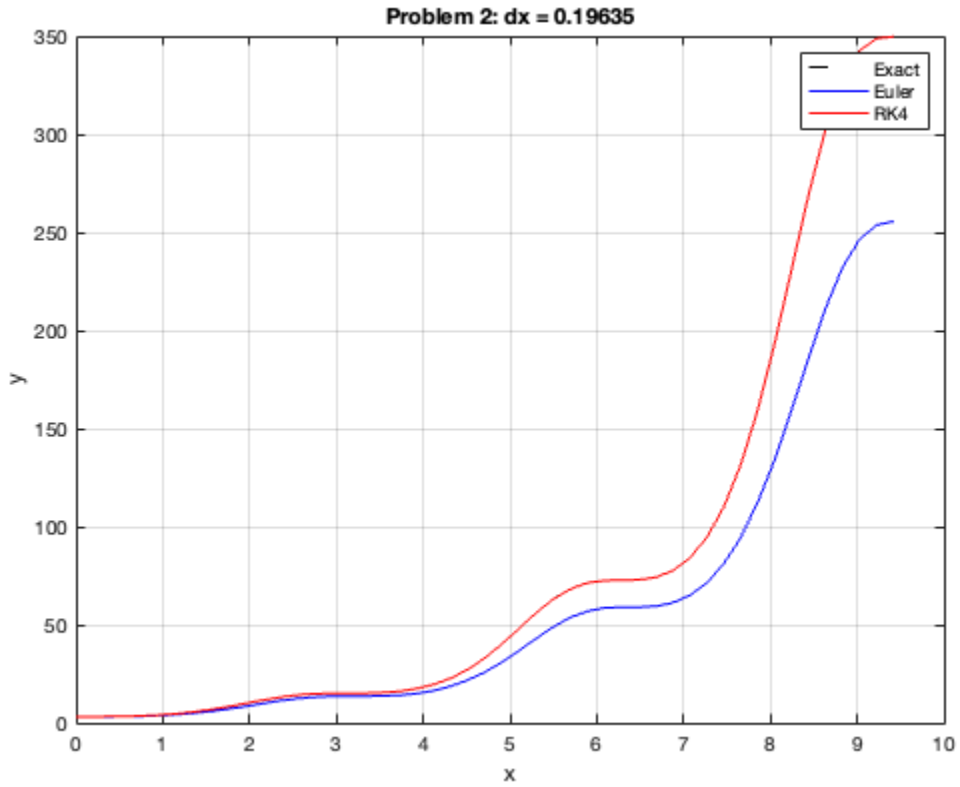

Problem 1 - Euler vs Accurate
Accurate
Euler
y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.4
0.6
0.8
1.2
1.4
1.6
1.8
2


Problem 2: dx = 3.1416
Exact
Euler
RK4
y
x
0
50
100
150
200
250
300
350
1
2
3
4
5
6
7
8
9
10


Problem 2: dx = 1.5708
Exact
Euler
RK4
y
x
0
50
100
150
200
250
300
350
0
1
2
3
4
5
6
7
8
9
10


Problem 2: dx = 0.7854
Exact
Euler
RK4
y
x
0
50
100
150
200
250
300
350
0
1
2
3
4
5
6
7
8
9
10


Problem 2: dx = 0.3927
Exact
Euler
RK4
y
x
0
50
100
150
200
250
300
350
0
1
2
3
4
5
6
7
8
9
10


Problem 2: dx = 0.19635
Exact
Euler
RK4
y
x
0
50
100
150
200
250
300
350
0
1
2
3
4
5
6
7
8
9
10

Published with MATLAB® R2024b