Table of Contents

```
% Author(s): Andrew Patella
% Assignment title: Coding Challenge 4
% Purpose: Integration approximation functions
% Creation date: 11/8/23
% Revisions: N/A

clear; clc; close all;
```

Loading in accurate data

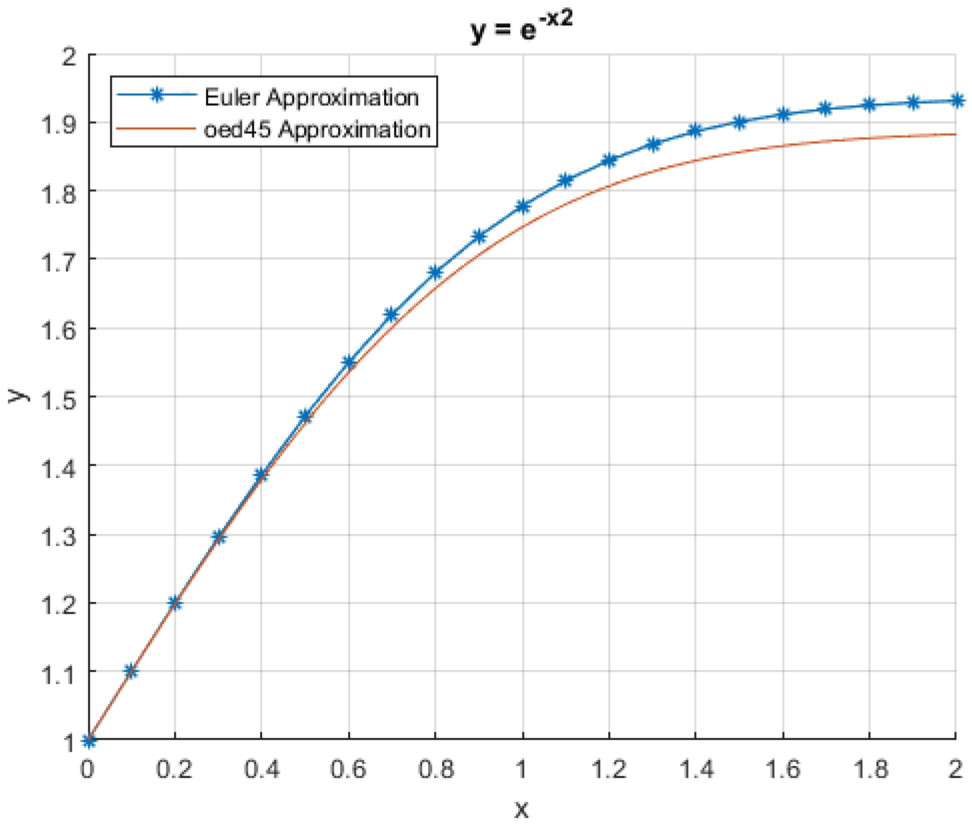
```
data = load('AccurateData.mat');
x_accurate = data.x_accurate;
y_accurate = data.y_accurate;
```

1) y=e^-x^2

```
%Initial Values
x0 = 0;
y0 = 1;
xf = 2;
delta = 0.1;

N = (xf-x0)/delta;

%Function handles
f1 = @(x,y) exp(-(x)^2);
ftest = @(x,y) (x);

%Calling Euler function to approximate the values of the solution
[x1,y1] = euler_explicit(f1,x0,xf,y0,delta);

%plotting results
figure(1);
hold on;
plot(x1,y1,'-*','LineWidth',1);
plot(x_accurate,y_accurate);
grid on;
xlabel('x');
ylabel('y');
title('y = e^-^x^2');
```

```
legend('Euler Approximation','oed45 Approximation','Location','Northwest')
grid on;

%interpolating the accurate data to find the correct values at time values
accurate_interp = interp1(x_accurate,y_accurate,x1,'linear','extrap');

%Calculating RMSE between the euler values and the accurate data,
%interpolated
rmse1 = rmse(transpose(y1),accurate_interp);

%Outputting values for the quiz
fprintf('Last y value (x=2):y = %f\n',y1(end));
fprintf('RMSE for Euler, part 1: %2.4f\n',rmse1);
```

```
Last y value (x=2):y = 1.931105
RMSE for Euler, part 1: 0.0326
```



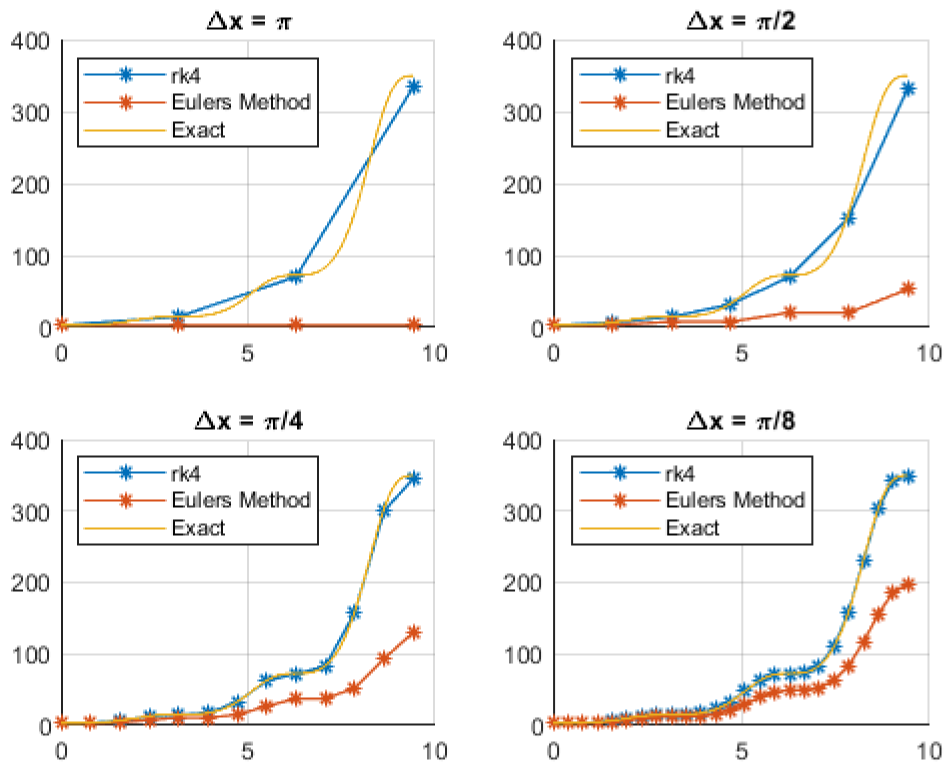
2) dydx=ysin^2(x)

```
%Delceration of constants
y02=pi;
x02=0;
xf2 = 3*pi;

%4 Different delta values
delta21 = pi;
```

```
delta22 = pi/2;
delta23 = pi/4;
delta24 = pi/8;

%Creating a matrix of x values for the exact solution
exactx = x02:.01:xf2;

%function handle of the exact solution
yexact = @(x)pi.*exp((2.*x-sin(2.*x))./4);

%Function handle for differential equation
dydx2 = @(x,y) y.*(sin(x)).^2;

%Calculating rk4 approximation for different delta values
[xout21,yout21] = runge4(dydx2,x02,xf2,y02,delta21);
[xout22,yout22] = runge4(dydx2,x02,xf2,y02,delta22);
[xout23,yout23] = runge4(dydx2,x02,xf2,y02,delta23);
[xout24,yout24] = runge4(dydx2,x02,xf2,y02,delta24);

%Calculating euler approximation for different delta values
[xout21e,yout21e] = euler_explicit(dydx2,x02,xf2,y02,delta21);
[xout22e,yout22e] = euler_explicit(dydx2,x02,xf2,y02,delta22);
[xout23e,yout23e] = euler_explicit(dydx2,x02,xf2,y02,delta23);
[xout24e,yout24e] = euler_explicit(dydx2,x02,xf2,y02,delta24);

%CAlculating exact solution with the correct number of points for RMSE
yvec1 = yexact(xout21);
yvec2 = yexact(xout22);
yvec3 = yexact(xout23);
yvec4 = yexact(xout24);

%Calculating RMSE for all sets of data
rmsevecrk4 =
 [rmse(transpose(yout21),yvec1),rmse(transpose(yout22),yvec2),rmse(transpose(yout23),yvec3
rmseveceuler =
 [rmse(transpose(yout21e),yvec1),rmse(transpose(yout22e),yvec2),rmse(transpose(yout23e),yv

%Plotting results
figure(2);
subplot(2,2,1);
hold on;
grid on;
plot(xout21,yout21,'-*','Linewidth',1);
plot(xout21e,yout21e,'-*','Linewidth',1);
plot(exactx,yexact(exactx),'Linewidth',1);
legend('rk4','Eulers Method','Exact','Location','northwest');
title('\Deltax = \pi');

subplot(2,2,2);
hold on;
grid on;
plot(xout22,yout22,'-*','Linewidth',1);
plot(xout22e,yout22e,'-*','Linewidth',1);
plot(exactx,yexact(exactx),'Linewidth',1);
```

```
legend('rk4','Eulers Method','Exact','Location','northwest');
title('\Deltax = \pi/2');

subplot(2,2,3);
hold on;
grid on;
plot(xout23,yout23,'-*','Linewidth',1);
plot(xout23e,yout23e,'-*','Linewidth',1);
plot(exactx,yexact(exactx),'Linewidth',1);
legend('rk4','Eulers Method','Exact','Location','northwest');
title('\Deltax = \pi/4');

subplot(2,2,4);
hold on;
grid on;
plot(xout24,yout24,'-*','Linewidth',1);
plot(xout24e,yout24e,'-*','Linewidth',1);
plot(exactx,yexact(exactx),'Linewidth',1);
legend('rk4','Eulers Method','Exact','Location','northwest');
title('\Deltax = \pi/8');


%Printing values for the quiz
fprintf('RK4: RMSE for pi = %3.4f\n',rmsevecrk4(1));
fprintf('RK4: RMSE for pi/2 = %3.4f\n',rmsevecrk4(2));
fprintf('RK4: RMSE for pi/4 = %3.4f\n',rmsevecrk4(3));
fprintf('RK4: RMSE for pi/8 = %3.4f\n',rmsevecrk4(4));

disp(' ');
fprintf('Euler: RMSE for pi = %3.4f\n',rmseveceuler(1));
fprintf('Euler: RMSE for pi/2 = %3.4f\n',rmseveceuler(2));
fprintf('Euler: RMSE for pi/4 = %3.4f\n',rmseveceuler(3));
fprintf('Euler: RMSE for pi/8 = %3.4f\n',rmseveceuler(4));
```

```
RK4: RMSE for pi = 7.7193
RK4: RMSE for pi/2 = 7.1569
RK4: RMSE for pi/4 = 1.5324
RK4: RMSE for pi/8 = 0.0835

Euler: RMSE for pi = 176.8436
Euler: RMSE for pi/2 = 125.6068
Euler: RMSE for pi/4 = 91.3529
Euler: RMSE for pi/8 = 61.4778
```



Comments

```
% You have to down sample the accurate data because it is a more accurate
% strategy than up sampling the approximation. The step size for the
% accurate data was 0.008, and the step size for the euler's data was 0.1.
% It is more accurate to find a value between 2 values 0.008 units apart
% than to do it if they are 0.1 apart. Since the model is linear, the best
% way to interpolate is by down sampling. 'extrap' specifies the method of
% extrapolation which allows interp1 to return values outside of the domain
% of the input domain. If you wanted to interpolate at 0.3 but the data
% only had 0.2 and 0.4, extrap would be necessary otherwise it would return
% NaN.
```

Published with MATLAB® R2022a