MATH 345L - Spring 2022

LAB Project- Solving ODEs using Euler and Runge-Kutta methods

Maxwell Griffith

Problem 1. Review Chapter 8, Sections 8.1 and 8.3 of the textbook. Summarize the Euler and Runge-Kutta methods.

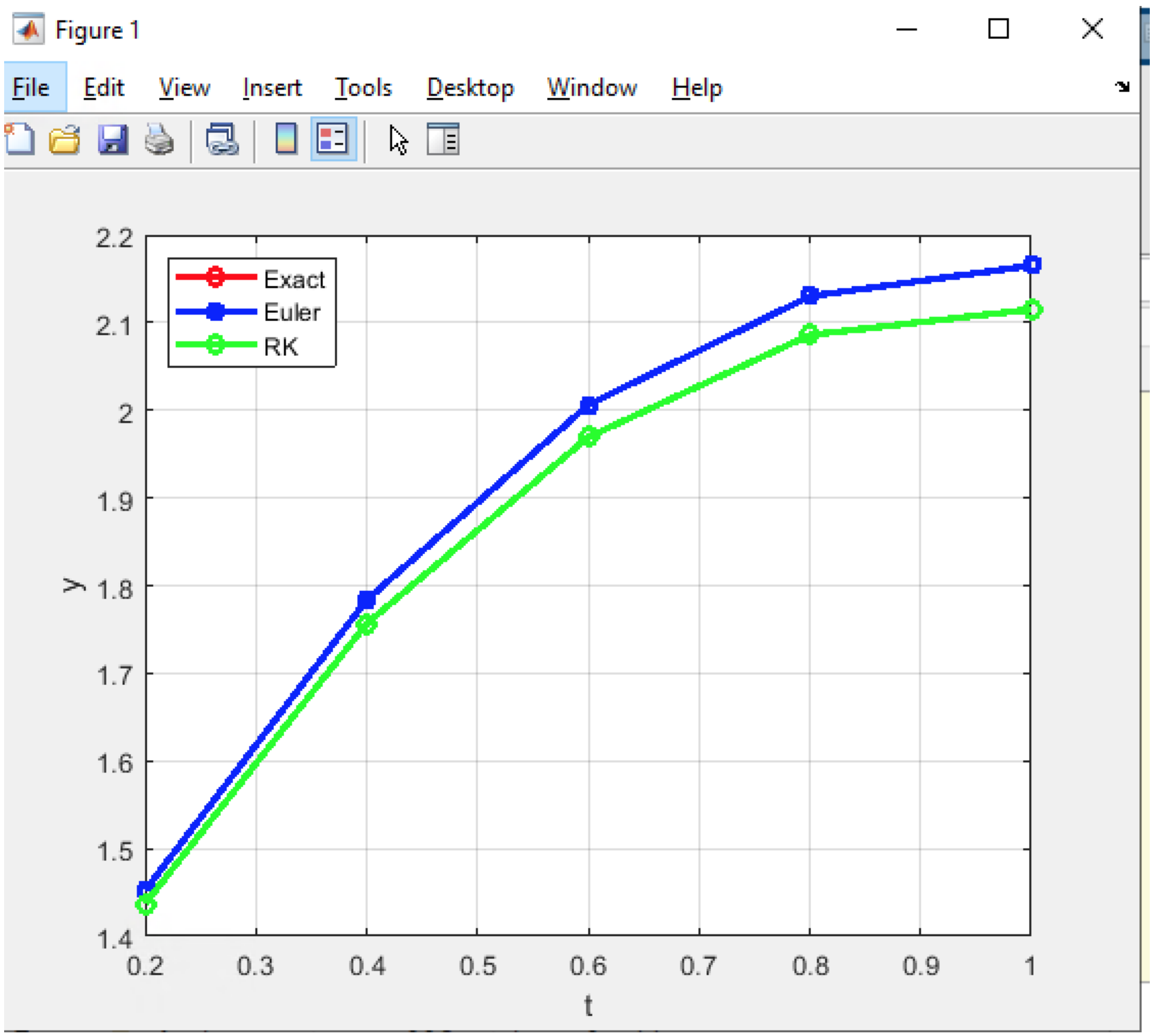
Euler method is using the tangent line method. It approximates values of the solution at specific t-values. We find the slope at a point as an approximation of the real solution curve on a short interval. Then for the next value of the estimation we use the slope found in the first step to approximate the value at the next t. So it is recursively using the estimation of the previous step to approximate the next y value.

Runge-Kutta method uses a weighted average of slope values over a set interval to estimate the function. It is based on the euler formula to find the estimates that are then averaged.

h=.05					
t	exact	euler	Rk	euler abs error	rk abs error
0.2	1.4371	1.4521	1.4371	0.0149	0
0.4	1.7565	1.7835	1.7565	0.027	0
0.6	1.9694	2.006	1.9694	0.0367	0
0.8	2.0858	2.13	2.0858	0.0442	0
1	2.1151	2.1651	2.1151	0.05	0
h=.1					
t	exact	euler	Rk	euler abs error	rk abs error
0.2	1.4371	1.4675	1.4371	0.0304	0
0.4	1.7565	1.8114	1.7565	0.0549	0
0.6	1.9694	2.0438	1.9694	0.0744	0
0.8	2.0858	2.1755	2.0858	0.0897	0
1	2.1151	2.2164	2.1151	0.1013	0
h=.025					
t	exact	euler	Rk	euler abs error	rk abs error
0.2	1.4371	1.4445	1.4371	0.0074	0
0.4	1.7565	1.7699	1.7565	0.0134	0
0.6	1.9694	1.9876	1.9694	0.0182	0
0.8	2.0858	2.1078	2.0858	0.0219	0
1	2.1151	2.1399	2.1151	0.0248	0

Problem 4. Compare the results obtained from the Euler method with the RK method (i.e., using the absolute error).

The RK method is much more accurate because the absolute error values are almost zero.



Graph of solutions at h=.05

APPENDIX 1

```
%INPUT - ODE (string), y0=init cond (init), T=t_values (array), h= step size (int)
%OUTPT - euler_values = estimaties of y (array)
function [euler_values] = Euler_Method(ODE, y0, T, h)

%-------------------------------------------------------------
% Define the differential equation y'=f(t,y)
%-------------------------------------------------------------
f=inline(ODE,'t','y');

%-------------------------------------------------------------
% Set initial condition
%-------------------------------------------------------------
t0=0;

%-------------------------------------------------------------
% Loop to solve the IVP four times -- once for each entry in T
% Use Euler method.
%-------------------------------------------------------------
for j=1:length(T)

        % We need this many iterations to get to T with a stepsize of h.
        steps=round(T(j)/h);

        % Start at initial condition each time through
        y_n=y0;
        t_n=t0;
```

```
        % Implement the Euler method
        for i=1:steps
        y_np1=y_n+f(t_n,y_n)*h;
        t_n=t_n+h;
        y_n=y_np1;
        end

        euler_values{j}=y_n; %returns y values as array
end
```

APPENDIX 2

```
%INPUT - ODE (string), y0=init cond (init), T=t_values (array), h= step size (int), t0
%OUTPT - euler_values = estimaties of y (array)
function [rk_values] = RK_Method(ODE, y0, T, h, t0)

f=inline(ODE,'t','y');

for j=1:length(T)

  steps=round(T(j)/h);

  y=y0;
  t=t0;

        for i=1:steps
        k1=f(t,y);
        k2=f(t + h/2, y + h*k1/2);
        k3=f(t + h/2, y + h*k2/2);
        k4=f(t+h, y + h*k3);
        next_y=y + h*(k1 + 2*k2 + 2*k3 + k4)/6;
        t=t+h;
        y=next_y;
        end
  %add y value to rk_values
  rk_values{j}=y; %returns y values as array
end
```

APPENDIX 3

```
% INPUT - ODE (text), y0 (init) , t_values (array)
% Output - exact_values (array)

function [exact_values] = Exact_Method(ODE, y0, t_values)

eqn = strcat('Dy=',ODE)
inits = sprintf('y(0)=%0.2g',y0)

t = t_values;
y = dsolve(eqn,inits, 't');
exact_values= eval(vectorize(y));


%test values y' = 2y-3t; y(0)=1, at t=0.1, 0.2, 0.3
%exact_values = [1.2054,1.4230,1.6555,1.9064]
```

APPENDIX 4

```
clc; clear all; close all;

ODE = '3-2*t-0.5*y';
t0 = 0;
y0 = 1;
t_values = [0.2 0.4 0.6 0.8 1.0];

%------------------------------------------------------------------
% allow user to enter the stepsize h
%------------------------------------------------------------------
step_size=input('Enter the stepsize => ');

y_exact = Exact_Method(ODE, y0, t_values)
y_euler = Euler_Method(ODE, y0, t_values, step_size)
y_rk = RK_Method(ODE, y0, t_values, step_size, t0)

%% Find absolute error using |y_exact-y_n_tot|
EulerAbsError = abs(y_exact - cell2mat(y_euler))
RKAbsError = abs(y_exact - cell2mat(y_rk))

%%Create Table-how to export as csv?
%Euler Sol, RK Sol, Exact Sol, Euler Error, RK Error

%%Sketch Graph
figure;
plot(t_values,y_exact, '-ro','Linewidth',2.5); hold on;
plot(t_values,cell2mat(y_euler),'-bs', 'Linewidth',2.5);
plot(t_values,cell2mat(y_rk),'-go', 'Linewidth',2.5);
```

```
legend('Exact','Euler','RK', 'Location','NorthWest');
xlabel('t');
ylabel('y');
grid on;
```