ISTANBUL TECHNICAL UNIVERSITY

UCK 348E

Computer Applic. in Engineering

CRN : 15016

HomeWork 2

NAME : NESLİHAN GÜLSOY

STUDENT ID : 110160111

DEPARTMENT : AEROSPACE ENGINEERING

INSTRUCTOR : Bülent Tutkun

Deadline : 19.11.2021

Contents

1 Analysis of 2^{nd} Order Ordinary Differential Equation

1.1 Givens :

$$y'' = \frac{10000}{100 - 5t} - 9.8 \qquad y(0) = 0 \qquad y' = 0 \tag{1}$$

2^{nd} Order Ordinary Differential Equation (ODE) is given above.

initial values of height and velocity are $y(0) = 0$, $y' = 0$ respectively.

interval of time is $0 \leq t \leq 10$

step size is $h = 1$

Numerical computations of the above ODE will made by Fourth-Order Runge-Kutta Method.

1.2 Steps :

Firstly given 2^{nd} order ODE is rewritten as 2 separated 1^{st} order ODEs. It is noted that first derivative of height (y) is equal to velocity (v) and first derivative of velocity is equal to second derivative of height y''.

$$y' = v \tag{2}$$

$$y'' = v' = \frac{10000}{100 - 5t} - 9.8 \tag{3}$$

t values are calculated with

$$t_{i+1} = h + t_i \tag{4}$$

Velocity and height are calculated step by step from $t = 0$ *sec* to $t = 10$ *sec* with step size of $h = 1$ *sec*. Eq. (3) and Eq. (2) are used to compute Eq (5). It is noted that the ODE of velocity depends only on the time variable, while the ODE for height depends only on the velocity variable. Percent relative errors $(\%|\varepsilon_{rel}|)$ of height and velocity are calculated. Results are given as table and graphics at Section (2) and implemented code is given at Section (3.1).

1.2.1 4th Order Runge-Kutta Method

Assume that $v = f(t, v, y)$ and $y = g(t, v, y)$ then below formula can be written

$$
\begin{aligned}
v_{i+1} &= v_i + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\
y_{i+1} &= y_i + \frac{1}{6}\left(l_1 + 2l_2 + 2l_3 + l_4\right) \\
k_1 &= hf(t_i, v_i, y_i) \\
l_1 &= hg(t_i, v_i, y_i) \\
k_2 &= hf(t_i + h/2, v_i + k_1/2, y_i + l_1/2) \\
l_2 &= hg(t_i + h/2, v_i + k_1/2, y_i + l_1/2) \\
k_3 &= hf(t_i + h/2, v_i + k_2/2, y_i + l_2/2) \\
l_3 &= hg(t_i + h/2, v_i + k_2/2, y_i + l_2/2) \\
k_4 &= hf(t_i + h, v_i + k_3, y_i + l_3) \\
l_4 &= hg(t_i + h, v_i + k_3, y_i + l_3)
\end{aligned} \tag{5}
$$

2 Results and Comments :

$h = 1$				
$Time\ (sec)$	$Velocity_{(numeric)}\ (m/sec)$	$Height_{(numeric)}\ (m)$	$\%\varepsilon_{rel}$ (v)	$\%\varepsilon_{rel}$ (y)
0	0.000000	0.000000	0.00	0.00
1	92.786595	45.954701	100.00	100.00
2	191.121045	187.421191	51.45	75.48
3	295.637883	430.255993	35.35	56.44
4	407.087140	781.005759	27.38	44.91
5	526.364201	1247.036985	22.66	37.37
6	654.549970	1836.700420	19.58	32.10
7	792.965952	2559.542630	17.46	28.24
8	943.251422	3426.582940	15.93	25.30
9	1107.474259	4450.683154	14.83	23.01
10	1288.294749	5647.052507	14.04	21.19

Table 1: Numerical Calculations for Vertical Flight of a Rocket with 4^{th} Order Runge-Kutta Method with Step Size of 1

When Tables (1), (2) and Figures (1) and (2) are examined, curves of variable are as expected. The main point to be mentioned is error. According to calculated absolute percentage relative error ($\%|\varepsilon_{rel}|$), using method converges relatively faster for velocity variable than height.This is because the calculated velocity values are reused in the method to calculate the height values and the error accumulates. As expected relative error decreases with increased step number, to obtain much improved solution interval of time can be increased and/or step size, h, can be decreased.

Unlike relative error, absolute percent true error,($\%|\varepsilon_{true}|$), increases with the number of steps for velocity (v). For height, while it converges relatively quickly with increasing step number in a certain interval,$0 - 7$ sec, it started to increase at some point. But true error for v is found many times lower than for y.

By looking at the calculated errors, it can be said that 4^{th} Order Runge-Kutta method seems suitable enough to solve given system. However, the numerical solution should be repeated with different methods and different step sizes in order to make a more detailed interpretation.

$h = 1$				
$Time\ (sec)$	$Velocity_{(analytical)}\ (m/sec)$	$Height_{(analytical)}\ (m)$	$\%\varepsilon_{true}$ (v)	$\%\varepsilon_{true}$ (y)
0	0.000000	0.000000	0.000000	0.000000
1	92.786589	45.954813	0.000006	0.000245
2	191.121031	187.421436	0.000007	0.000131
3	295.637859	430.256397	0.000008	0.000094
4	407.087103	781.006358	0.000009	0.000077
5	526.364145	1247.037826	0.000011	0.000067
6	654.549888	1836.701570	0.000013	0.000063
7	792.965832	2559.544182	0.000015	0.000061
8	943.251248	3426.585030	0.000018	0.000061
9	1107.474002	4450.685983	0.000023	0.000064
10	1288.294361	5647.056389	0.000030	0.000069

Table 2: Analytical Calculations for Vertical Flight of a Rocket

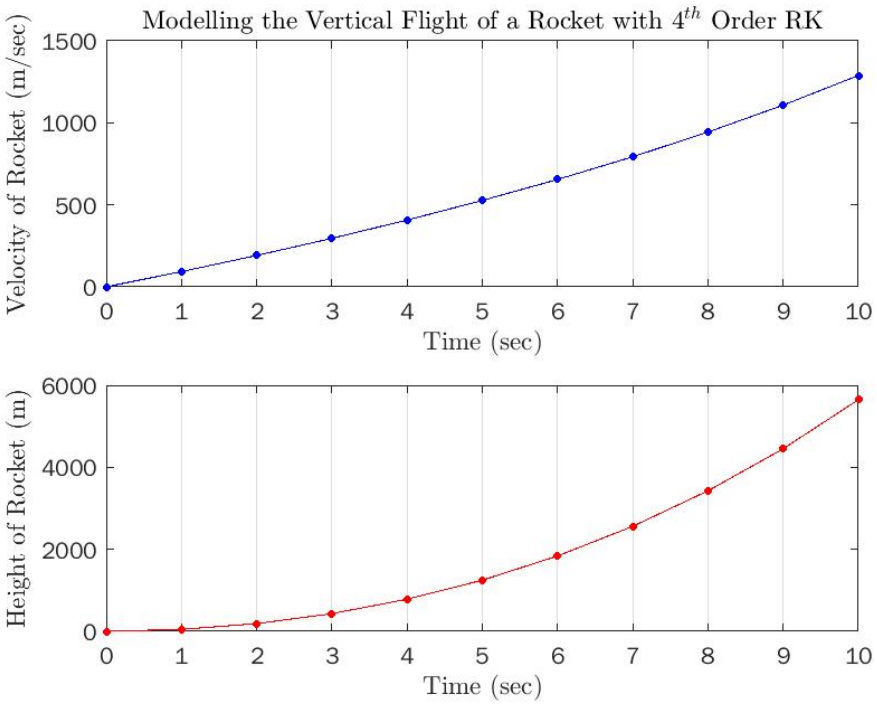

Figure 1: Numeric Calculation for Given ODEs with Step Size of $h = 1\ sec$

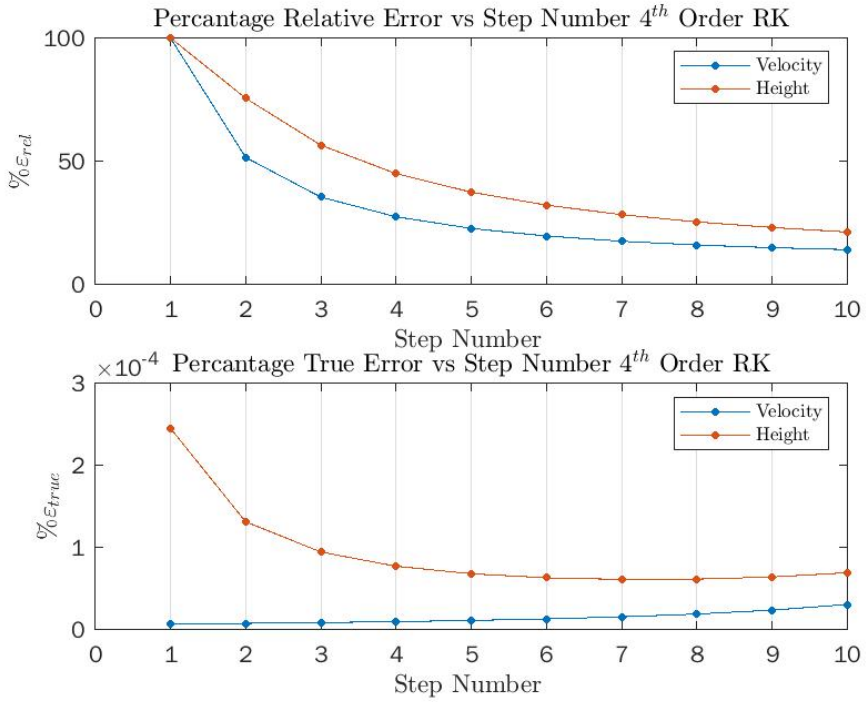

Figure 2: Absolute Error Graphics of Numeric Calculation for Each Step

3 APPENDIX :

3.1 Implemented MatLab Code :

```
%%%%%% HOMEWORK 1 %%%%%%
%%%% ANALYSIS : Differential of v (velocity) is already given (ODE2).
% And differential of y (vertical path) is known as v. This equation is
% given at ODE1. Both ODE contain 1 different variable. Also
% imaginary slopes of y (l1,l2,l3,l4) depend on imaginary slopes for
% v which are k1,k2,k3,k4.
% According to Mathematical background below code is written

%Written by Neslihan G lsoy

clear; clc; close all;

%%Givens

func_ODE1 = @(v) v;       %ODE of Vertical Path, y (m)
func_ODE2 = @(t) 10000 ./ (100 - 5.*t)-9.8;  %ODE of Velocity, v (m/sec)

v(1) = 0;        %Initial Value of Vertical Velocity (v)
y(1) = 0;        %Initial Value of Vertical Flight (y)

%To Calculate True Solutions
syms t1 C1 C2
int1 = int((10000 / (100 - 5*t1)-9.8),t1);     %Integration of y'' (y')
eq1 = subs(int1,0) + C1 == 0;        %To find Constant with Initial Value
C1 = vpasolve(eq1,C1);               %Constant for Integration of y'' (y')
func_true_v = int1 + C1;             %True Solution of Velocity (m/sec)

int2 = int(func_true_v);             %Integration of y' (y)
eq2 = subs(int2,0) + C2 == 0;        %To find Constant with Initial Value
C2 = vpasolve(eq2,C2);               %Constant for Integration of y' (y)
func_true_y = int2 + C2;             %True Solution of Height (m)

%Interval of t
t(1) = 0;        %Initial Value for time, t (sec)
t_end = 10;      %Stop Value for t
h = 1;           %Step Size = 1

%Create Time Points with Given Step Size
for i = 1 : t_end
    t(i+1,1) = h + t(i);
end

%To Calculate True Values

for i = 1 : length(t)
    true_v(i,1) = real(double(subs(func_true_v,t(i))));
    true_y(i,1) = real(double(subs(func_true_y,t(i))));
end
```

```
%To Make Numerical Computation

for i = 1 : length(t)-1
    k1(i,1) = h * func_ODE2(t(i));
    k2(i,1) = h * func_ODE2(t(i) + h/2);
    k3(i,1) = h * func_ODE2(t(i) + h/2);
    k4(i,1) = h * func_ODE2(t(i) + h);
    v(i+1,1) = v(i) + 1/6 * (k1(i) + 2*k2(i) + 2*k3(i) + k4(i));

    l1(i,1) = h * func_ODE1(v(i));
    l2(i,1) = h * func_ODE1(v(i) + k1(i)/2);
    l3(i,1) = h * func_ODE1(v(i) + k2(i)/2);
    l4(i,1) = h * func_ODE1(v(i) + k3(i));
    y(i+1,1) = y(i) + 1/6 * (l1(i) + 2*l2(i) + 2*l3(i) + l4(i));

    %Relative Percent Error
    rer_y(i+1,1) = abs( (y(i+1)-y(i)) / (y(i+1)) ) *100;
    rer_v(i+1,1) = abs( (v(i+1)-v(i)) / (v(i+1)) ) *100;
end

%True Percent Error
ter_v(2:i+1,1) = abs( (true_v(2:end) - v(2:end)) ./ true_v(2:end) ) * 100;
ter_y(2:i+1,1) = abs( (true_y(2:end) - y(2:end)) ./ true_y(2:end) ) * 100;

% To Create Table for h = 1
Output.t = t;
Output.v_true = true_v;
Output.v =  v;
Output.y_true = true_y;
Output.y =  y;
Output.true_error_v = ter_v;
Output.rel_error_v = rer_v;
Output.true_error_y = ter_y;
Output.rel_error_y = rer_y;
%
disp('   <strong> h = 1</strong>');
T = struct2table( Output );
T.Properties.VariableNames = {'t (sec)','v_true','v_numeric','y_true', ...
    'y_numeric','%true. error (v)','%rel. error (v)', '%true. error (y)',...
    '%rel. error (y)'};
disp(T) %Display Outputs
%
%To Create an Excel File to Form Proper Table
writetable(T,'Table_HW2.xls')

%%%% PLOTTING
figure(1)
subplot(2,1,1)
plot(t,v,'b','MarkerSize',12,'Marker','.')
ylabel('Velocity of Rocket (m/sec)','Interpreter','latex')
xlabel('Time (sec)','Interpreter','latex');
ax = gca;
ax.XTick = (0:1:10);
ax.XGrid = 'on';
```

```
title('Modelling the Vertical Flight of a Rocket with $4^{th}$ Order RK'...
    ,'Interpreter','latex')

subplot(2,1,2)
plot(t,y,'r','MarkerSize',12,'Marker','.')
ylabel('Height of Rocket (m)','Interpreter','latex')
xlabel('Time (sec)','Interpreter','latex');
ax = gca;
ax.XTick = (0:1:10);
ax.XGrid = 'on';

saveas(gcf,'HW2fig1.jpg')
saveas(gcf,'HW2fig1.fig')

% ERROR PLOTTING
figure(2)
subplot(2,1,1)
plot( t(2:end),[rer_v(2:end),rer_y(2:end)],...
    'MarkerSize',12,'Marker','.');
title('Percantage Relative Error vs Step Number $4^{th}$ Order RK'...
    ,'Interpreter','latex')
% Create ylabel
ylabel('$\% \varepsilon_{rel}$','Interpreter','latex');
% Create xlabel
xlabel('Step Number ','Interpreter','latex');
legend('Velocity','Height','Interpreter','latex')
ax = gca;
ax.XTick = (0:1:10);
ax.XGrid = 'on';

subplot(2,1,2)
plot( t(2:end),[ter_v(2:end),ter_y(2:end)],...
    'MarkerSize',12,'Marker','.');
title('Percantage True Error vs Step Number $4^{th}$ Order RK'...
    ,'Interpreter','latex')
% Create ylabel
ylabel('$\% \varepsilon_{true}$','Interpreter','latex');
% Create xlabel
xlabel('Step Number ','Interpreter','latex');
legend('Velocity','Height','Interpreter','latex')
ax = gca;
ax.XTick = (0:1:10);
ax.XGrid = 'on';
saveas(gcf,'HW2fig2.jpg')
saveas(gcf,'HW2fig2.fig')
```