SCPY405: Computational Fluid Dynamics
Homework 1

Dulyawat Boonvut, 6205110

September 8, 2022

1. For a differential equation

$$\dot{x} = -\ln(x) \tag{1}$$

with initial condition $x(0) = 10$
Apply Euler method to equation (1) and we get

$$\begin{cases} x_{i+1} &= x_i - h\ln(x_i) \\ t_{i+1} &= t_i + h \end{cases} \tag{2}$$

Apply Heun method to equation (1) and we get

$$\begin{cases} x_{i+1} &= x_i + \frac{h}{2}(f_i + f_p) \\ t_{i+1} &= t_i + h \end{cases} \tag{3}$$

where f_i is a slope at x_i
and f_p is the slope at a predictor $x_p = x_i - h\ln(x_i)$
that is
$f_i = -\ln(x_i)$
and $f_p = -\ln(x_p) = \ln(x_i - h\ln(x_i))$

From (2) and (3), $h = 1$ and $h = 0.1$ are chosen from the problem.
The numerical solutions and the exact solutions (from ode45 MATLAB function) are shown below.

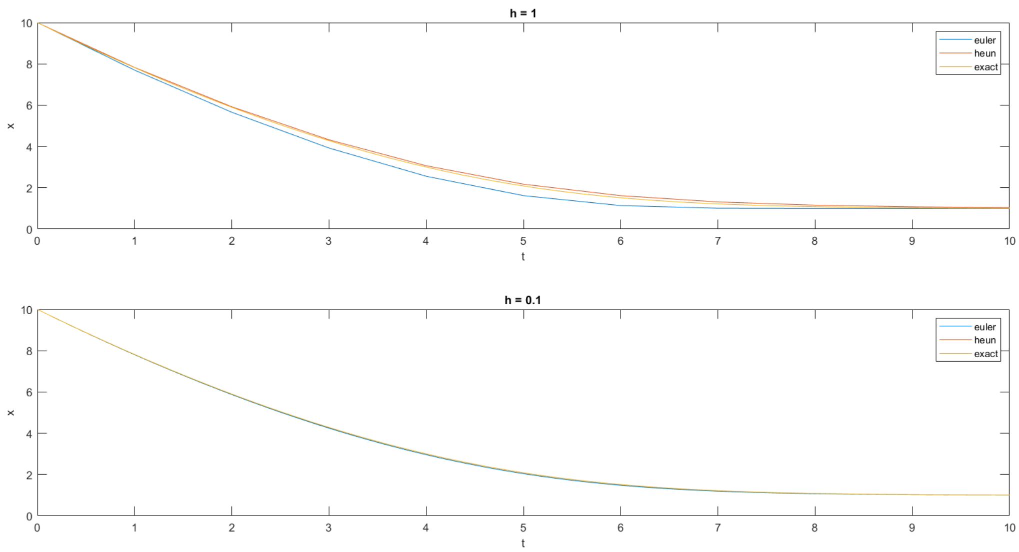

Figure 1: Numerical Solution and solution from ode45 of (1) for $h = 0.1$

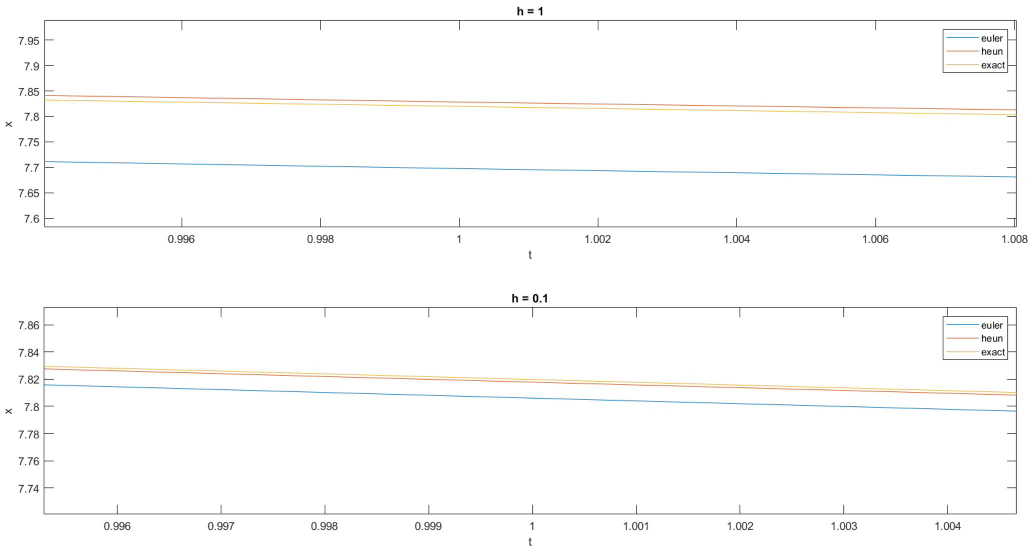

Figure 2: Numerical Solution and solution from ode45 of (1) for $h = 0.1$, at $x = 1$ and its neighbour

MATLAB Source Code:
Main: prb1hw1.m

```
clc;clear
% solve x' = -ln(x) with x(0)=10

x0 = 10;
ti = 0; tf = 10;

h1 = 1;
[t1e,x1e] = solve1_euler(x0,h1,ti,tf);
[t1h,x1h] = solve1_heun(x0,h1,ti,tf);

h2 = 0.1;
[t2e,x2e] = solve1_euler(x0,h2,ti,tf);
[t2h,x2h] = solve1_heun(x0,h2,ti,tf);

[t,x] = ode45(@(t,x) -log(x),[ti tf],x0);

subplot(211);
plot(t1e, x1e);hold on;
plot(t1h, x1h);
plot(t,x);
legend('euler','heun','exact');
title('h = 1');

subplot(212);
plot(t2e, x2e);hold on;
plot(t2h, x2h);
plot(t,x);
legend('euler','heun','exact');
title('h = 0.1');
```

function: solve1_euler.m

```
function [t,x] = solve1_euler(x0,h,t1,t2)
%solve DE for problem1 (x'=-ln(x) with x(0)=10)
%using Euler method
    n = (t2-t1)/h;
    x=zeros(n,1); x(1)=x0;
    t=zeros(n,1); t(1)=t1;
    for i=1:n
        x(i+1) = x(i) - h*log(x(i));
        t(i+1) = t(i) + h;
    end
```

```
end
```

function: solve1_heun.m

```
function [t,x] = solve1_heun(x0,h,t1,t2)
% solve DE for problem1 (x'=-ln(x) with x(0)=10)
% using Heun method
    n = (t2-t1)/h;
    x=zeros(n,1); x(1)=x0;
    t=zeros(n,1); t(1)=t1;
    for i=1:n
        f = -log(x(i));
        x_ = x(i) - h*log(x(i)); f_ = -log(x_); %predictor
        x(i+1) = x(i) + h/2*(f + f_); %iteration

        t(i+1) = t(i) + h;
    end
end
```

2. For a differential equation

$$\ddot{\theta} + 0.1\dot{\theta} + \sin(\theta) = 0 \tag{4}$$

with initial conditions $\theta(0) = 1.5$ and $\dot{\theta}(0) = 0$
(4) can convert to system of first-order differential equations.

$$\begin{cases} \dot{\theta} & = \omega \\ \dot{\omega} & = -0.1\omega - \sin(\theta) \end{cases} \tag{5}$$

Apply Euler method to (5) and we get

$$\begin{cases} \omega_{i+1} & = \omega_i + h(-0.1\omega_i - \sin(\theta_i)) \\ \theta_{i+1} & = \theta_i + h\omega_i \\ t_{i+1} & = t_i + h \end{cases} \tag{6}$$

Apply Heun method to system (5) and we get

$$\begin{cases} \omega_{i+1} & = \omega_i + \frac{h}{2}(f_{\omega_i} + f_{\omega_p}) \\ \theta_{i+1} & = \theta_i + \frac{h}{2}(f_{\theta_i} + f_{\theta_p}) \\ t_{i+1} & = t_i + h \end{cases} \tag{7}$$

where f_{ω_i} is a slope at ω_i
f_{ω_p} is a slope at $\omega_p = \omega_i + h(-0.1\omega_i - \sin(\theta_i))$
f_{θ_i} is a slope at θ_i
and f_{θ_p} is a slope at $\theta_p = \theta_i + h\omega_i$

For predictor $\theta_p = \theta_i + h\omega_i$ and $\omega_p = \omega_i + h(-0.1\omega_i - \sin(\theta_i))$,
the slopes are
$f_{\omega_i} = -0.1\omega_i - \sin(\theta_i)$
$f_{\omega_p} = -0.1\omega_p - \sin(\theta_p)$
$f_{\theta_i} = \omega_i$
and $f_{\theta_p} = \omega_p$
The numerical solutions and the exact solutions (from ode45 MATLAB function) are shown below.

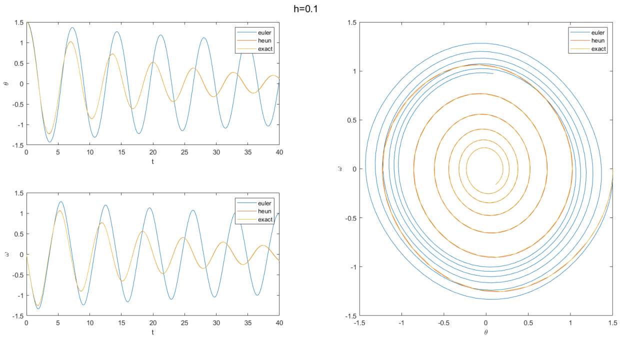

Figure 3: Numerical Solutions and exact solutions of (4) for $h = 0.1$

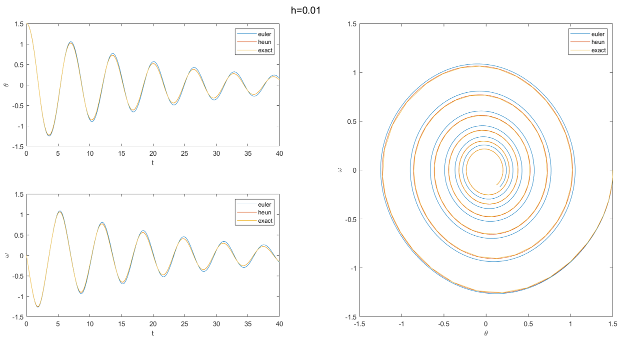

Figure 4: Numerical Solutions and exact solutions of (4) for $h = 0.01$

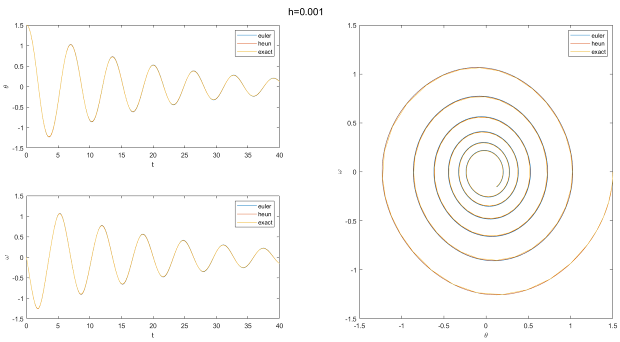

Figure 5: Numerical Solutions and exact solutions of (4) for $h = 0.001$

MATLAB Source Code: Main: prb2hw1.m

```
clc;clear
% problem 2: solve x'' + 0.1x' + sin(x) = 0 with x(0)=1.5 and x'(0)=0
% system:  x' = y, y' = -0.1y - sin(x)
% x = theta and y = omega: in this problem

x0 = 1.5;
y0 = 0;
ti = 0; tf = 40;

h1 = 0.1; t1 = ti:h1:tf;
[x1e, y1e] = solve2_euler(x0,y0,h1,ti,tf);
[x1h, y1h] = solve2_heun(x0,y0,h1,ti,tf);

h2 = 0.01; t2 = ti:h2:tf;
[x2e, y2e] = solve2_euler(x0,y0,h2,ti,tf);
[x2h, y2h] = solve2_heun(x0,y0,h2,ti,tf);

h3 = 0.001; t3 = ti:h3:tf;
```

```
[x3e, y3e] = solve2_euler(x0,y0,h3,ti,tf);
[x3h, y3h] = solve2_heun(x0,y0,h3,ti,tf);

[t_, v] = ode45(@f2, [ti tf], [x0 y0]);
x_ = v(:,1); y_ = v(:,2);

figure(1);sgtitle('h=0.1');plot2(t1,x1e,x1h,y1e,y1h,t_,x_,y_);
figure(2);sgtitle('h=0.01');plot2(t2,x2e,x2h,y2e,y2h,t_,x_,y_);
figure(3);sgtitle('h=0.001');plot2(t3,x3e,x3h,y3e,y3h,t_,x_,y_);
```

function: solve2_euler.m

```
function [x, y] = solve2_euler(x0,y0,h,t1,t2)
% solve x'' + 0.1x' + sin(x) = 0 with x(0)=1.5 and x'(0)=0
% system:  x' = y, y' = -0.1y - sin(x)
% using Euler method

    n = (t2-t1)/h;
    x = zeros(n,1); x(1) = x0;
    y = zeros(n,1); y(1) = y0;
    for i=1:n
        y(i+1) = y(i) + h*(-0.1*y(i) - sin(x(i)));
        x(i+1) = x(i) + h*y(i);
    end
end
```

function: solve2_heun.m

```
function [x, y] = solve2_heun(x0,y0,h,t1,t2)
% solve x'' + 0.1x' + sin(x) = 0 with x(0)=1.5 and x'(0)=0
% system:  x' = y, y' = -0.1y - sin(x)
% using Heun method
    n = (t2-t1)/h;
    x = zeros(n,1); x(1) = x0;
    y = zeros(n,1); y(1) = y0;

    for i=1:n
        %predictor
        x_ = x(i) + h*y(i);
        y_ = y(i) + h*(-0.1*y(i) - sin(x(i)));

        fy = -0.1*y(i) - sin(x(i));
        fy_ = -0.1*y_ - sin(x_);
        y(i+1) = y(i) + h/2*(fy + fy_);

        fx = y(i);
        fx_ = y_;
        x(i+1) = x(i) + h/2*(fx + fx_);
    end

end
```

function: f2.m

```
function f = f2(¬,v)
%function to verify by using ode45 (problem 2)
    x = v(1); y = v(2);
    out_x = y;
    out_y = -0.1*y - sin(x);

    f = [out_x; out_y];
end
```

function: plot2.m

```
function [] = plot2(t, xe, xh, ye, yh, tex, xex, yex)
% plot solutions (problem 2)

subplot(2,2,1);
plot(t,xe); hold on;
```

```
plot(t,xh);
plot(tex,xex);
xlabel('t'); ylabel('\theta');
legend('euler','heun','exact');

subplot(2,2,[2 4]);
plot(xe,ye); hold on;
plot(xh,yh);
plot(xex,yex);
xlabel('\theta'); ylabel('\omega');
legend('euler','heun','exact');

subplot(2,2,3);
plot(t,ye); hold on;
plot(t,yh);
plot(tex,yex);
xlabel('t'); ylabel('\omega');
legend('euler','heun','exact');

end
```

3. A system for problem 3 is

$$\begin{cases} \dot{v} = f(t,v,z) = -g + \dfrac{m_p v_e - 0.5 c_d \rho v|v| A}{m_c + m_p} \\ \dot{z} = g(t,v,z) = v \end{cases} \tag{8}$$

where $g = 9.8\text{m/s}^2$, $m_c = 50$ kg, $\rho = 1.23$ kg/m^3, $A = 0.1$ m^3, $v_e = 360$m/s, $c_d = 0.15$ and mass of the propellant at time t is

$$m_p(t) = \begin{cases} m_{po} t^2/8 & 0 \le t < 1 \\ m_{po}/8 + (t-1) m_{po}/4 & 1 \le t < 4 \\ m_{po} - m_{po}(5-t)^2/8 & 4 \le t \le 5 \\ m_{po} & t > 5 \end{cases} \tag{9}$$

where $m_{po} = 100$ kg

Apply Heun Method to system (8).

$$\begin{cases} v_{i+1} &= v_i + \frac{h}{2}(f(v_i) + f(v_p)) \\ z_{i+1} &= z_i + \frac{h}{2}(g(z_i) + g(z_p)) = z_i + \frac{h}{2}(v_i + v_p) \\ t_{i+1} &= t_i + h \end{cases} \tag{10}$$

where predictors $z_p = z_i + h v_i$ and $v_p = v_i + h f(v_i)$,

For this problem, a time step $h = 0.1$ is chosen. A numerical solution and an exact solution (from ode45 MATLAB function) for velocity and position are shown below.

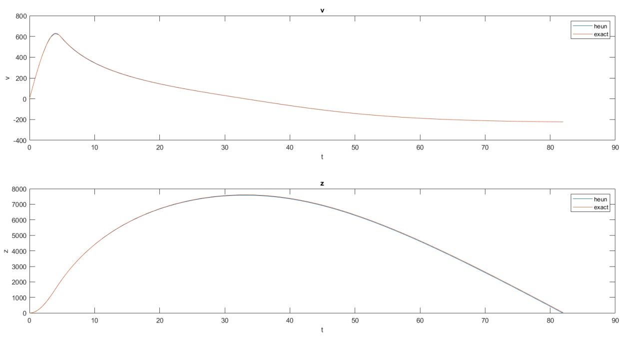

Figure 6: Numerical solution and solution from ode45 in (4)

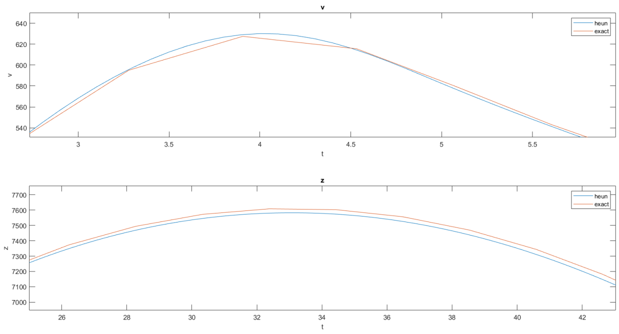

Figure 7: Numerical solution and solution from ode45 in (4), at maximum point

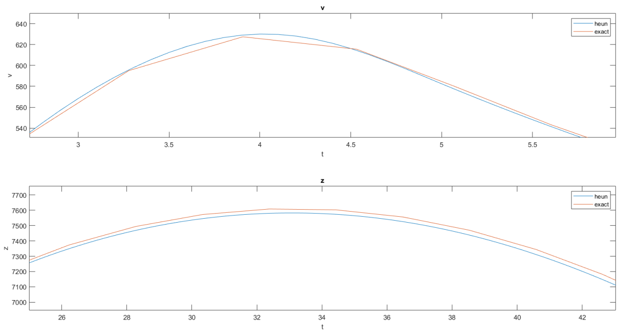

Figure 8: Numerical solution and solution from ode45 in (4), when the rocket hits the ground ($z = 0$)

From the graphs, we can determine the maximum speed (v_{max}), the maximum height of the rocket (z_{max}), time (t_g) and velocity(v_g) that the rocket hits the ground, as shown on the table below.

physical quantities	values (numerical)	values (ode45)
v_{max} (m/s)	630.0231	627.4049
z_{max} (m)	7582.7	7608.4
t_g (s)	82	82
v_g (m/s)	-223.3890	-223.3514

Table 1: Table of values of v_{max}, z_{max}, t_g and v_g, obtained from both numerical solution and the solution from ode45

MATLAB Source Code: Main: hw1prb3.m

```
clc;clear
% problem 3: solve rocket equation
% v' = g + (mp*ve - 0.5*cd*rho*v*|v|*A)/(mc+mp)
% z' = v

global g ve cd rho A mc mpo;

g = 9.8;
mc = 50;
rho = 1.23;
A = 0.1;
ve = 360;
cd = 0.15;

mpo = 100;

v0 = 0;
z0 = 0;

h = 0.1;

```

```
v(1) = v0;
z(1) = z0;
t(1) = 0;

i = 1; % index
while 1
    t_ = t(i) + h;
    v_ = v(i) + h*fz(t(i),z(i),v(i));
    z_ = z(i) + h*v(i);

    v(i+1) = v(i) + h/2*(fz(t(i),z(i),v(i)) + fz(t_,z_,v_));
    z(i+1) = z(i) + h/2*(v(i) + v_);
    t(i+1) = t(i) + h;

    i=i+1;

    %check if the rocket hits the ground
    if(z(i)<0)
        break;
    end
end
%numerical
vmax = max(v); % maximum speed
zmax = max(z); % maximum height
tg = t(i); % time that the rocket hits the ground (z=0)
vg = v(i); % velocity that the rocket hits the ground (z=0)

[te, we] = ode45(@f3, [0 tg], [v0 z0]);
ve = we(:,1); ze = we(:,2);

%ode45
vmax_ = max(ve); % maximum speed
zmax_ = max(ze); % maximum height
tg_ = te(end); % time that the rocket hits the ground (z=0)
vg_ = ve(end); % velocity that the rocket hits the ground (z=0)

subplot(211);plot(t,v);hold on;plot(te,ve);
xlabel('t');ylabel('v');
legend('heun','exact'); title('v');
subplot(212);plot(t,z);hold on;plot(te,ze);
xlabel('t');ylabel('z');
legend('heun','exact'); title('z');

[vmax vmax_] %display vmax
[zmax zmax_] %display zmax
[tg tg_] %display tg
[vg vg_] %display vg
```

function: mp.m

```
function m = mp(t)
% mass of propellant at time t (problem 3)
    global mpo;
    if(t≥0)
        if(t≤1)
            dm = mpo*t^2/8;
        elseif(t>1 && t≤4)
            dm = mpo/8 + (t-1)*mpo/4;
        elseif(t≥4 && t≤5)
            dm = mpo - mpo/8*(5-t)^2;
        else
            dm = mpo;
        end
    end
    m = mpo - dm;
end
```

function: fz.m

```
function f = fz(t,¬,v)
%function to use in heun method (problem 3)
    global g ve cd rho A mc mpo;
    f = -g + (mp(t)*ve - 1/2*cd*rho*v*abs(v)*A)/(mc + mp(t));
end
```

function: f3.m

```
function f = f3(t,vector)
%function to verify by using ode45 (problem 3)
    global g ve cd rho A mc mpo;

    v = vector(1); z = vector(2);

    out_v = -g + (mp(t)*ve - 1/2*cd*rho*v*abs(v)*A)/(mc + mp(t));
    out_z = v;

    f = [out_v; out_z];
end
```