Solution to Homework 1

1. Solve the initial-value problem $\dot{x}(t) = -\ln x, \quad x(0) = 10$ using the Euler method and the Heun method with h = 1 and 0.1 s in the time interval 0 s - 10 s. Compare the numerical solutions with those computed using the MATLAB function `ode45`. Show the numerical schemes of the Euler and Heun methods for this problem, the graphs of solutions and your source codes. (6 points)

   <u>Answer:</u>

   The iterative scheme of the Euler method for this problem is

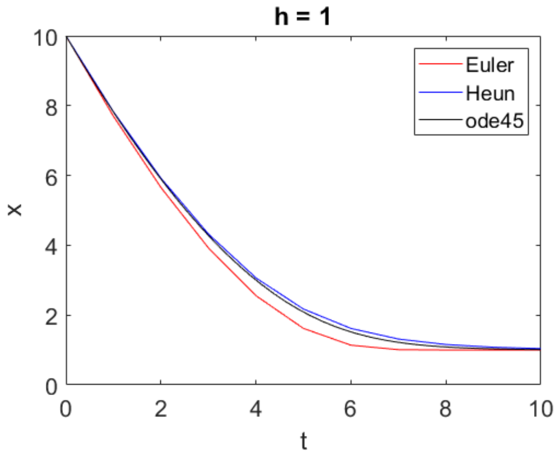
$$x_{n+1} = x_n - h \ln x_n$$

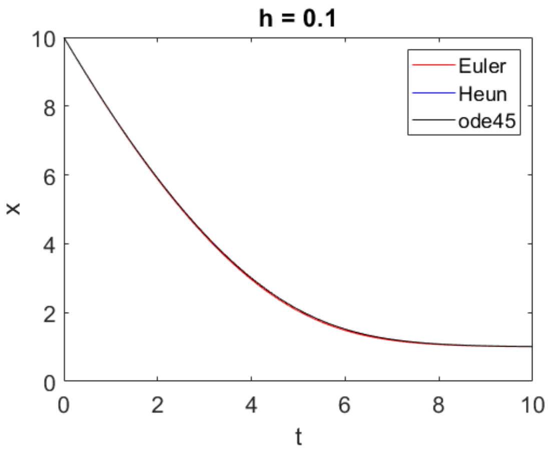
   where $x_n = x(t_n) = x(nh)$, and $n = 0, 1, 2, \ldots$.

   The iterative scheme of the Heun method for this problem is

$$\tilde{x}_{n+1} = x_n - h \ln x_n$$
$$x_{n+1} = x_n - h\left(\frac{\ln x_n + \ln \tilde{x}_{n+1}}{2}\right)$$

   The numerical results when h = 1 and h = 0.1 are shown below. When h = 1, it is obvious that the Heun method provided a more accurate result than that of the Euler method. Here, the result from function `ode45` is used as the reference which is more accurate than both the Euler and Heun methods. This is because function `ode45`a is an implementation of the 4th and 5th-order Runge-Kutta method. When h = 0.1, the results from the Euler and Heun methods almost coincide with the result of function `ode45`.





   The source code of the implementations is as follows.

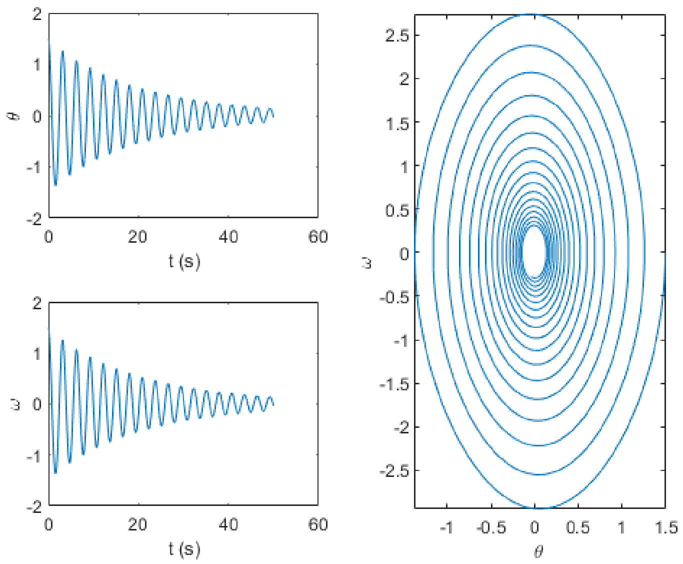
```
h = 1; t = 0:h:10; n = length(t);

% Implementation of Euler method
x = zeros(n,1); x(1) = 10;
for i=1:n-1
    x(i+1) = x(i)-h*log(x(i));
end

% Implementation of Heun method
x2 = zeros(n,1); x2(1) = 10;
for i=1:n-1
    xx = x2(i)-h*log(x2(i));
    x2(i+1) = x2(i)-0.5*h*(log(x2(i))+log(xx));
end
plot(t,x,'r',t,x2,'b');
```

```
% Implementation using function ode45
func = @(t,y) -log(y);
tspan = [0,10]; x0 = 10;
[t,x3] = ode45(func, tspan, x0);
hold on; plot(t,x3,'k'); hold off
set(gca,'FontSize',16)
legend('Euler','Heun','ode45')
xlabel('t'); ylabel('x'); title("h = "+h)
```

2. Solve the initial-value problem $\ddot{\theta}+0.1\dot{\theta}+\sin\theta=0,\ \ \theta(0)=1.5,\ \ \dot{\theta}(0)=0.$ using the Euler method and the Heun method with h = 0.1, 0.01, 0.001 s in the time interval 0 s - 10 s. Compare the numerical solutions with those computed using the MATLAB function `ode45`. Show the corresponding system of first-order ODEs, the numerical schemes of the Euler and Heun methods for this problem, the graphs of solutions and your source codes. An example of the solution graphs is shown on the right figure. (6 points)



Answer: The corresponding first-order ODEs are

$$\dot{\omega}=-0.1\omega-\sin\theta$$
$$\dot{\theta}=\omega$$

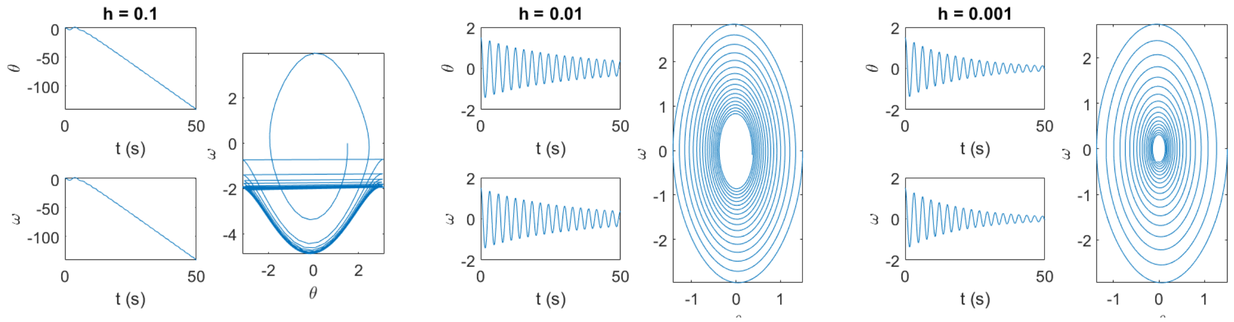
The iterative scheme of the Euler method for this problem is

$$\omega_{n+1}=\omega_n-h\left(0.1\omega_n+\sin\theta_n\right)$$
$$\theta_{n+1}=\theta_n+h\omega_n$$

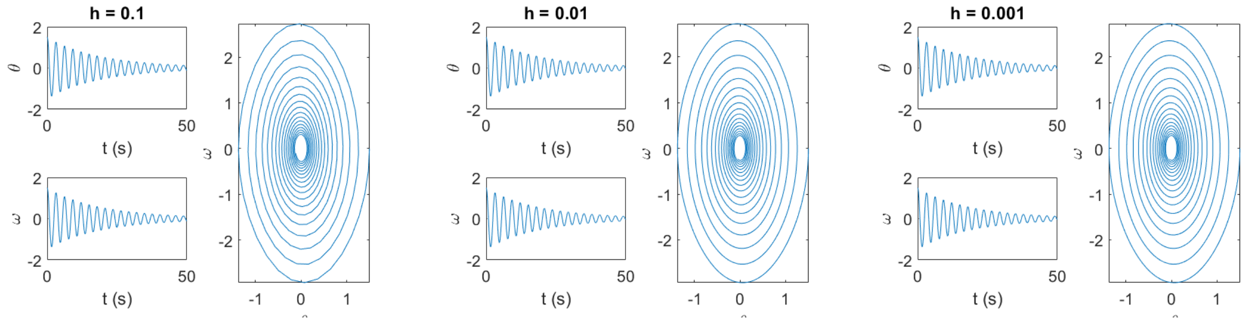
The iterative scheme of the Heun method for this problem is

$$\tilde{\omega}_{n+1}=\omega_n-h\left(0.1\omega_n+\sin\theta_n\right)$$
$$\tilde{\theta}_{n+1}=\theta_n+h\omega_n$$
$$\omega_{n+1}=\omega_n-h\frac{\left(0.1\omega_n+\sin\theta_n\right)+\left(0.1\tilde{\omega}_{n+1}+\sin\tilde{\theta}_{n+1}\right)}{2}$$
$$\theta_{n+1}=\theta_n+h\frac{\omega_n+\tilde{\omega}_{n+1}}{2}$$

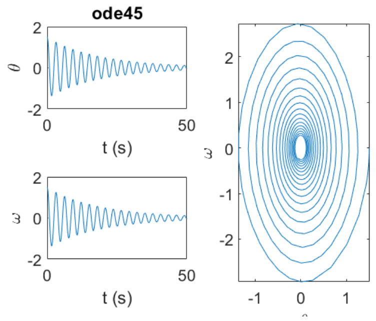
The numerical results of the Euler method when h = 0.1, 0.01, and 0.001 are shown below.



The numerical results of the Heun method when h = 0.1, 0.01, and 0.001 are shown below.



The numerical results from function `ode45` are shown below. This result is used as the reference.



The numerical results show that the Euler method provides an accurate result compared to that of function `ode45` when h = 0.001 while the Heun method provides an accurate result even when h = 0.1.

The implementation of the Euler method is as follows.

```
h = 0.001;
t = 0:h:50;
n = length(t);
theta = zeros(n,1);
omega = zeros(n,1);
theta(1) = 1.5;
for i=1:n-1
    w = omega(i);
    omega(i+1) = w-h*(0.1*w+5*sin(theta(i)));
    theta(i+1) = theta(i)+h*omega(i);
end
subplot(221);plot(t,theta);
set(gca,'FontSize',16)
xlabel('t (s)');ylabel('\theta');title("h = "+h)
subplot(223);plot(t,theta);
set(gca,'FontSize',16)
xlabel('t (s)');ylabel('\omega')
subplot(122);plot(wrapToPi(theta),omega);axis image
set(gca,'FontSize',16)
xlabel('\theta');ylabel('\omega')
```

The implementation of the Heun method is as follows.

```
h = 0.001;
t = 0:h:50;
n = length(t);
theta = zeros(n,1);
omega = zeros(n,1);
theta(1) = 1.5;
for i=1:n-1
    wi = omega(i);
    ti = theta(i);
    wi1 = wi-h*(0.1*wi+5*sin(ti));
    ti1 = ti+h*wi;
    omega(i+1) = wi-0.5*h*((0.1*wi+5*sin(ti))+(0.1*wi1+5*sin(ti1)));
    theta(i+1) = ti+0.5*h*(wi+wi1);
end
subplot(221);plot(t,theta);
set(gca,'FontSize',16)
xlabel('t (s)');ylabel('\theta');title("h = "+h)
subplot(223);plot(t,theta);
set(gca,'FontSize',16)
xlabel('t (s)');ylabel('\omega')
subplot(122);plot(wrapToPi(theta),omega);axis image
set(gca,'FontSize',16)
xlabel('\theta');ylabel('\omega')
```

The implementation using function `ode45` is as follows.

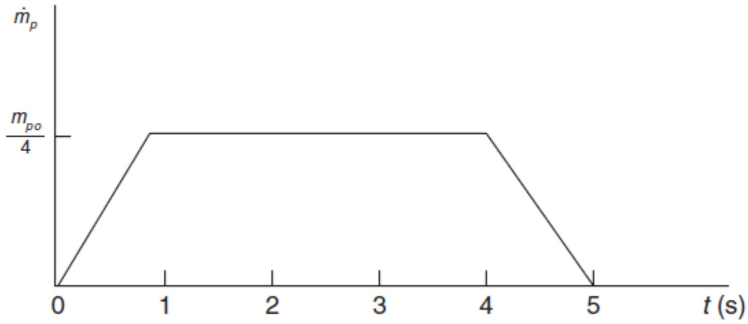
```
func = @(t,y) [-(0.1*y(1)+5*sin(y(2))); y(1)];
tspan = [0,50];
y0 = [0; 1.5];
[t,y] = ode45(func, tspan, y0);
omega = y(:,1);
theta = y(:,2);
subplot(221);plot(t,theta);
set(gca,'FontSize',16)
xlabel('t (s)');ylabel('\theta');title("ode45")
subplot(223);plot(t,theta);
set(gca,'FontSize',16)
xlabel('t (s)');ylabel('\omega')
subplot(122);plot(wrapToPi(theta),omega);axis image
set(gca,'FontSize',16)
xlabel('\theta');ylabel('\omega')
```

3. The equations of motion for a rocket are

$$\dot{v} = -g + \left(m_p v_e - \tfrac{1}{2} c_d \rho v |v| A\right) \Big/ \left(m_c + m_p\right)$$
$$\dot{z} = v$$

where the gravitational acceleration g = 9.8 m/s^2, the mass of rocket casing m_c = 50 kg, the air density ρ = 1.23 kg/m^3, the maximum cross-sectional area A = 0.1 m^2, the exhaust speed v_e = 360 m/s, the drag coefficient c_d = 0.15. The instantaneous mass of the propellant at time t, $m_p(t)$, is given by

$$m_p(t) = m_{po} - \int_0^t \dot{m}_p dt$$



where the initial weight of the propellant at time t = 0 is m_{po} = 100 kg, and the time-varying burn rate $\dot{m}_p$ is given in the right graph. Solve the equations of motion using the Heun method with h = 0.1 s. Determine the maximum speed of the rocket, the maximum height the rocket can reach, the time and velocity when the rocket hits the ground. Also use the MATLAB function `ode45` to determine these values. (8 points)

Answer:

The iterative scheme of the Heun method for this problem is

$$\tilde{v}_{n+1} = v_n + h\left[-g + \frac{\left(m_{p,n}v_e - \frac{1}{2}c_d\rho v_n|v_n|A\right)}{\left(m_c + m_{p,n}\right)}\right]$$

$$\tilde{z}_{n+1} = z_n + hv_n$$

$$v_{n+1} = v_n + \frac{h}{2}\left[-2g + \frac{\left(m_{p,n}v_e - \frac{1}{2}c_d\rho v_n|v_n|A\right)}{\left(m_c + m_{p,n}\right)} + \frac{\left(m_{p,n+1}v_e - \frac{1}{2}c_d\rho \tilde{v}_{n+1}|\tilde{v}_{n+1}|A\right)}{\left(m_c + m_{p,n+1}\right)}\right]$$

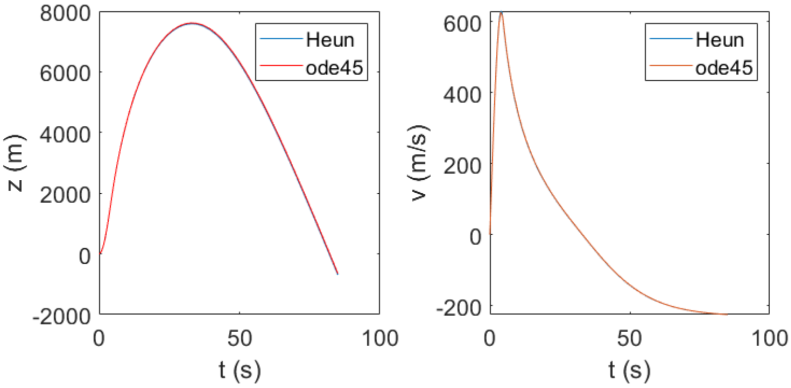
$$z_{n+1} = z_n + \frac{h}{2}\left(v_n + \tilde{v}_{n+1}\right)$$

where $m_{p,n} = m_p(t_n)$.

Using the provided graph of $\dot{m}_p$, we can compute $m_p(t)$ as

$$m_p(t) = \begin{cases} m_{po}\left(1 - t^2/8\right), & t < 1 \\ m_{po}\left(9/8 - t/4\right), & 1 \le t < 4 \\ m_{po}\left(t^2 - 10t + 25\right)/8, & 4 \le t \le 5 \\ 0, & 5 \le t \end{cases}$$

The profiles of vertical position z and velocity v obtained using the Heun method are shown in the figure below.



According to the numerical results, the maximum speed of the rocket ≈ 630 m/s, the maximum height ≈ 7582 m, the time when the rocket hits the ground ≈ 82 s, the velocity when the rocket hits the ground ≈ -223 m/s.

The source codes of the implementation are shown below.

```
function mp = propellant_mass(t,mpo)
if t < 1
    mp = mpo*(1-t*t/8);
elseif 1 <= t && t < 4
    mp = mpo*(9/8-t/4);
elseif t <= t && t < 5
    mp = mpo*(t*t-10*t+25)/8;
else
    mp = 0;
end
%---------------------------------------------
function out = rocket(t,y)
global g mc rho A ve cd mpo
mp = propellant_mass(t,mpo);
v = y(1);
fv = -g+(mp*ve-0.5*cd*rho*v*abs(v)*A)/(mc+mp);
out = [fv; v];
%---------------------------------------------
% Main code
global g mc rho A ve cd mpo
g = 9.8; mc = 50; rho = 1.23; A = 0.1; ve = 360; cd = 0.15; mpo = 100;

% implementation of the Heun method
h = 0.001; nt = 85000; t = (0:nt)*h;
z = zeros(nt,1); v = zeros(nt,1);
z(1) = 0; v(1) = 0; % initial conditions are satisfied automatically
for n=1:nt
    t = n*h; mp = propellant_mass(t,mpo);
    zn = z(n); vn = v(n);
    fv = -g+(mp*ve-0.5*cd*rho*vn*abs(vn)*A)/(mc+mp);
    vn1 = v(n)+h*fv;
    zn1 = z(n)+h*vn;
    mp1 = propellant_mass(t+h,mpo);
    fv1 = -g+(mp*ve-0.5*cd*rho*vn1*abs(vn1)*A)/(mc+mp1);
    v(n+1) = v(n)+0.5*h*(fv+fv1);
    z(n+1) = z(n)+0.5*h*(vn+vn1);
end
subplot(121);plot(t,z);set(gca,'FontSize',16);
xlabel('t (s)'); ylabel('z (m)')
subplot(122);plot(t,v);set(gca,'FontSize',16);
xlabel('t (s)'); ylabel('v (m/s)')

% determine the values of answers to the questions
i = find(z<0); i = i(1);
disp("Maximum speed is "+max(v));
disp("Maximum height is "+max(z));
disp("Time when the rocket hits the ground is " + t(i));
disp("Velocity when the rocket hits the ground is " + v(i));

% implementation using function ode45
tspan = [0,85]; y0 = [0; 0];
[t,y] = ode45(@rocket, tspan, y0);
v = y(:,1); z = y(:,2);
subplot(121);hold on;plot(t,z,'r');hold off;set(gca,'FontSize',16)
xlabel('t (s)');ylabel('z (m)');legend('Heun','ode45')
subplot(122);hold on;plot(t,v);hold off;set(gca,'FontSize',16)
xlabel('t (s)');ylabel('v (m/s)');legend('Heun','ode45')
```