Exercise 3 - TTK4130 Modeling and Simulation

Camilla Sterud

1 Problem 1

$$k_1 = f(y_n, t_n) \tag{1}$$
$$k_2 = f(y_n + ha_{21}k_1, t_n + hc_2) \tag{2}$$
$$y_{n+1} = y_n + h(b_1k_1 + b_2k_2) \tag{3}$$

Taylor expansion of a funciton of two variables:

$$f(y+\Delta, t+\delta) = f(y,t) + \Delta\frac{\partial f(y,t)}{\partial y} + \delta\frac{\partial f(y,t)}{\partial t} + O(\Delta^2) + O(\delta\Delta) + O(\delta^2) \tag{4}$$

1.1 a

$\frac{df(y_n,t_n)}{dt}) = \frac{\partial f(y_n,t_n)}{\partial y}\frac{dy}{dt} + \frac{\partial f(y_n,t_n)}{\partial t} = \frac{\partial f(y_n,t_n)}{\partial y}f(y_n,t_n) + \frac{\partial f(y_n,t_n)}{\partial t}$.

$a_{21} = c_1 = C$. Taylor expansion of Equation 2 using Equation 4:

$$\begin{aligned}k_2 &= f(y_n,t_n) + ha_{21}k_1\frac{\partial f(y,t)}{\partial y} + hc_2\frac{\partial f(y,t)}{\partial t} + O((ha_{21}k_1)^2) + O(h^2c_2a_{21}k_1) + O(h^2c_2^2)\\ &= k_1 + hC(k_1\frac{\partial f(y,t)}{\partial y} + \frac{\partial f(y,t)}{\partial t}) + O(h^2C^2)\end{aligned}$$

$$\underline{\underline{k_2 = f(y_n,t_n) + hC\frac{df(y_n,t_n)}{dt} + O(h^2)}} \tag{5}$$

1.2 b

From p. 518, Egeland & Gravdal: A method is of order p if p is the smallest number that satifies

$$y_{n+1} = y_n + hf(y_n,t_n) + ... + \frac{h^p}{p!}\frac{d^{p-1}f(y_n,t_n)}{dt^{p-1}} + O(h^{p+1}). \tag{6}$$

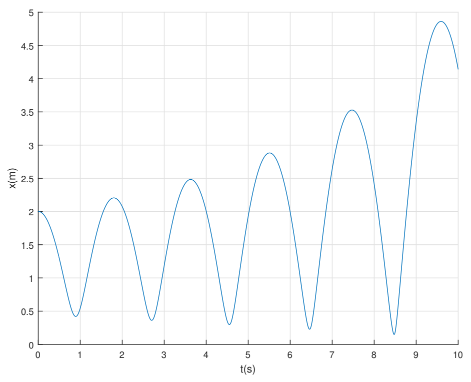
Putting the taylor expansion of k_2 from Equation 5 and Equation 1 into Equation 3 yields

$$\begin{aligned}y_{n+1} &= y_n + hb_1f(y_n,t_n) + hb_2(f(y_n,t_n) + hC\frac{df(y_n,t_n)}{dt} + O(h^2))\\ y_{n+1} &= y_n + h(b_1+b_2)f(y_n,t_n) + h^2b_2C\frac{df(y_n,t_n)}{dt} + O(h^3)\\ \Rightarrow b_1 + b_2 &= 1 \quad b_2C = \frac{1}{2!}\end{aligned}$$

$$\underline{\underline{c_2 = a_{12} = \frac{1}{2b_2}, \quad b_1 = 1 - b_2}}$$

2 Problem 2

2.1 a

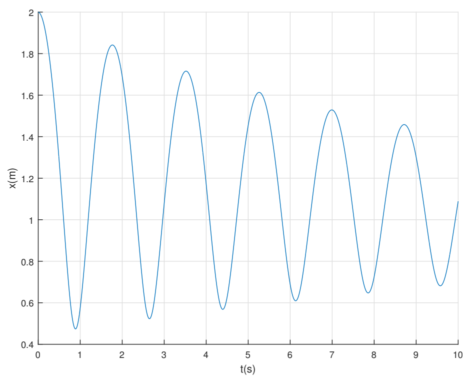


Figur 1: The pneumatic spring simulated with the explicit Euler method. The code for generating this plot is shown in Listing 1.

As seen in Figure 2.1, the explicit Euler method is unstable for this system. The position of the spring should be decreasing, but the error between the real position and the simulated position only grows ever larger. As seen in Figure 2.4, the energy in the spring increases as time goes on, and the system is clearly unstable.

2.2 b

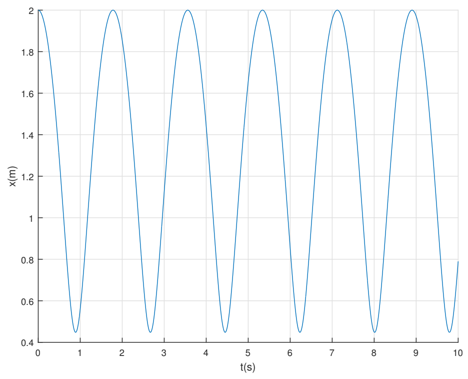
For the implicit Euler method, the system is stable. Both the energy and the position of the spring is decreasing (see Figures 2.2 and 2.4).



Figur 2: The pneumatic spring simulated with the implicit Euler method. The code for generating this plot is shown in Listing 1.

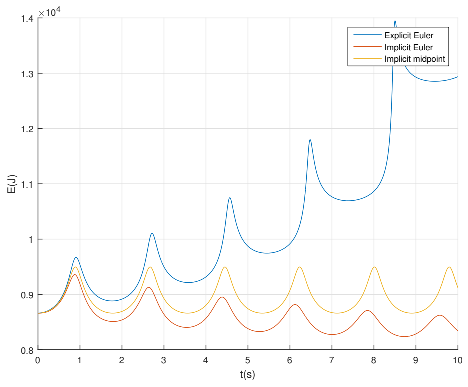
2.3 c

With the implicit midpoint rule, the system is marginally stable. Both the position and the energy are oscillating around a point, neither decaying to zero, nor rising to infinity (see Figures 2.3 and 2.4).



Figur 3: The pneumatic spring simulated with the implicit midpoint rule. The code for generating this plot is shown in Listing 1.

2.4 d



Figur 4: The energy in the system for the three solvers. The code for generating this plot is shown in Listing 1.

3 Problem 3

Voltage controlled DC motor:

$$
\begin{aligned}
L_1 \frac{di_a}{dt} &= -R_a i_a - K_E \omega_m + u_a \\
J_m \frac{d\omega_m}{dt} &= K_T i_a - T_L \\
K_E &= K_T.
\end{aligned}
$$

3.1 a

$T_L = u_a = 0$. Energy function for the system:

$$E = \frac{1}{2} L_a i_a^2 + \frac{1}{2} J_m \omega_m^2.$$

To prove stability, the energy in the system must be decreasing for all times.

$$
\begin{aligned}
\dot{E} &= L_a i_a \frac{di_a}{dt} + J_m \omega_m \frac{d\omega_m}{dt} \\
&= -R_a i_a^2 - K\omega_m i_a + i_a u_a + K\omega_m i_a - \omega_m T_L
\end{aligned}
$$

$$\underline{\underline{\dot{E} = -R_a i_a^2 \leq 0 \,\forall\, t > 0 \;\Rightarrow \text{The system is stable}}}$$

3.2 b

To prove passivity, the storage function, V, must fulfill the criteria $\dot{V} = \mathbf{u}^T \mathbf{y} - g(x), g(x) \geq 0$. $\mathbf{u} = \begin{bmatrix} u_a & -T_L \end{bmatrix}, \mathbf{y} = \begin{bmatrix} i_a & \omega_m \end{bmatrix}$.

$$
\begin{aligned}
V &= E \\
\dot{V} &= -R_a i_a^2 - K\omega_m i_a + i_a u_a + K\omega_m i_a - \omega_m T_L \\
&= -R_a i_a^2 + i_a u_a - \omega_m T_L
\end{aligned}
$$

$$\underline{\underline{\dot{V} = \mathbf{u}^T \mathbf{y} - R_a i_a^2,\; g(x) = R_a i_a^2 \geq 0 \Rightarrow \text{The system is passive.}}}$$

4 c

Since the system is passive, we can use two PID controllers to control the system to a given input.

Listing 1: All the methods implemented in MATLAB

```
kappa = 1.4;
g = 9.81;

h = 0.01;
```

```
t = 0:h:10;
y0 = [2;0];

ya = zeros (2,length(t));
yb = zeros (2,length(t));
yc = zeros (2,length(t));
ya(:,1) = y0;
yb(:,1) = y0;
yc(:,1) = y0;

f = @(y) [y(2);g*(y(1)^(-kappa) - 1)];
opt = optimset('Display','off','TolFun',1e-8);


for i = 1:(length(t) - 1)
    ya(:,i+1) = ya(:,i) + h*feval(f,ya(:,i));

    rb = @(ybnext) (yb(:,i) + ...
    h*feval(f, ybnext) - ybnext);
    yb(:,i+1) = fsolve(rb, yb(:,i), opt);

    rc = @(ycnext) (yc(:,i) + ...
    h*feval(f, (ycnext+yc(:,i))/2) - ycnext);
    yc(:,i+1) = fsolve(rc, yc(:,i), opt);
end


figure;
hold on; grid on;

plot(t,ya(1,:));
xlabel('t(s)');
ylabel('x(m)');

print -depsc modsim_ex4_2a.eps

figure;
hold on; grid on;

plot(t,yb(1,:));
xlabel('t(s)');
ylabel('x(m)');

print -depsc modsim_ex4_2b.eps

figure;
```

```
hold on; grid on;

plot(t,yc(1,:));
xlabel('t(s)');
ylabel('x(m)');

print -depsc modsim_ex4_2c.eps

figure;
hold on; grid on;

p0 = 2.5*10^5;
m = 200;
A = 0.01;

Ea = (1/(kappa-1))*p0*A.*ya(1,:).^(1-kappa) + ...
m*g.*ya(1,:) + 0.5*m.*ya(2,:).^2;
Eb = (1/(kappa-1))*p0*A.*yb(1,:).^(1-kappa) + ...
m*g.*yb(1,:) + 0.5*m.*yb(2,:).^2;
Ec = (1/(kappa-1))*p0*A.*yc(1,:).^(1-kappa) + ...
m*g.*yc(1,:) + 0.5*m.*yc(2,:).^2;


plot(t,Ea);
plot(t,Eb);
plot(t,Ec);
xlabel('t(s)');
ylabel('E(J)');
legend('Explicit Euler','Implicit Euler',...
'Implicit midpoint');

print -depsc modsim_ex4_2d.eps
```