Exercise 3 - TTK4130 Modeling and Simulation

Camilla Sterud

1 Problem 1

$$k_1 = f(y_n, t_n) \tag{1}$$
$$k_2 = f(y_n + ha_{21}k_1, t_n + hc_2) \tag{2}$$
$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2) \tag{3}$$

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

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

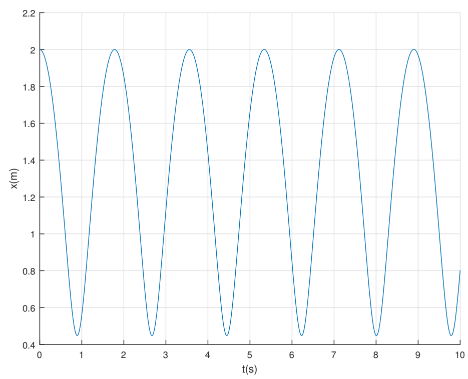
Putting the taylor expansion of k_2 from Equation 5 and Equation 1 into Equation 3 yields

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

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

2 Problem 2

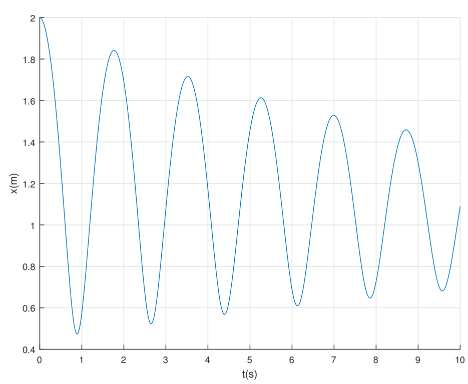
2.1 a



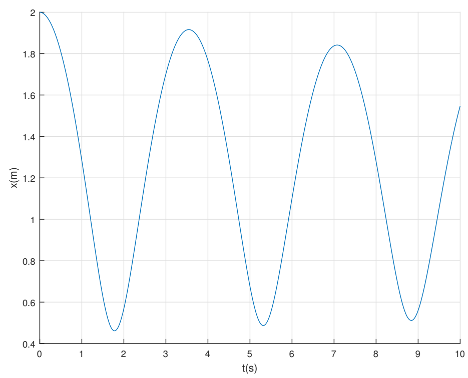
Figur 1: The pneumatic spring simulated with the explicit Euler method. The code for generating this plot is shown in Listing 1.

As seen in Figure 2.1, the explicit Euler method is on the verge of stability for this system. The position of the spring should be decreasing, but insted it oscillates around $\simeq 1.2$.

2.2 b



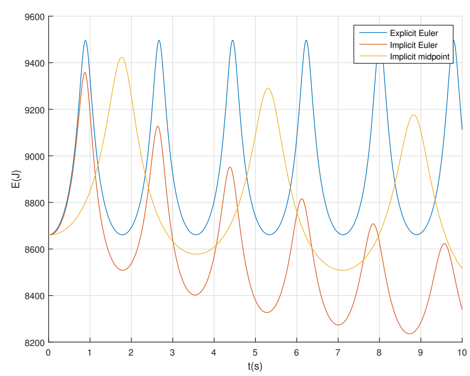
Figur 2: The pneumatic spring simulated with the implicit Euler method. The code for generating this plot is shown in Listing 2.



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

2.3 c

2.4 d



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

3 Listings

Listing 1: The explicit Euler method implemented i MATLAB

```
hold on; grid on;

kappa = 1.4;
g = 9.81;
```

```
h = 0.01;
t = 0:h:10;

ya = zeros (2,length(t));
ya(1,1) = 2;
ya(2,1) = 0;

f = @(ya) [ya(2);g*(ya(1)^(-kappa) - 1)];

for i = 1:(length(t) - 1)

    k_1 = f(ya(:,i));
    k_2 = f(ya(:,i) + 0.5*h.*k_1);

    ya(:,i+1) = ya(:,i) + h.*k_2;

end

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

print -depsc modsim_ex4_2a.eps
```

Listing 2: The implicit Euler method implemented i MATLAB

```
hold on; grid on;

kappa = 1.4;
g = 9.81;

h = 0.01;
t = 0:h:10;

yb = zeros (2,length(t));
yb(1,1) = 2;
yb(2,1) = 0;

f = @(yb) [yb(2);g*(yb(1)^(-kappa) - 1)];

opt = optimset('Display','off','TolFun',1e-8);

for i = 1:(length(t) - 1)
```

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

end

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

print -depsc modsim_ex4_2b.eps
```

Listing 3: The implicit midpoint rule implemented i MATLAB

```
hold on; grid on;

kappa = 1.4;
g = 9.81;

h = 0.01;
t = 0:h:10;

yc = zeros (2,length(t));
yc(1,1) = 2;
yc(2,1) = 0;

f = @(yc) [yc(2);g*(yc(1)^(-kappa) - 1)];

opt = optimset('Display','off','TolFun',1e-8);

for i = 1:(length(t) - 1)

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

end

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

print -depsc modsim_ex4_2c.eps
```

Listing 4: The implicit midpoint rule implemented i MATLAB

```
hold on; grid on;

h = 0.01;
t = 0:h:10;

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
```