Exercise 3 - TTK4130 Modeling and Simulation

Camilla Sterud

1 Problem 1

Volterra-Lotka predator-prey model:

$$\begin{aligned} \dot{u} &= u(v-3) \\ \dot{v} &= v(2-u) \end{aligned} \tag{1}$$

”Energy” function

$$V = u - 2\ln(u) + v - 3\ln(v)$$

1.1 a

$$\dot{V} = \frac{\partial V}{\partial u}\dot{u} + \frac{\partial V}{\partial v}\dot{v}$$

Calculate the partial derivatives of V and inserting the equations 1 gives us

$$\dot{V} = (1 - \frac{2}{u})u(v-3) + (1 - \frac{3}{v})v(2-u) = 0,$$

which means that V is constant for solutions of the system 1. This means that the system is stable and will not grow to $\pm\infty$.

1.2 b

As seen in Figure 1 the number of foxes and rabbits is periodic and V is constant.

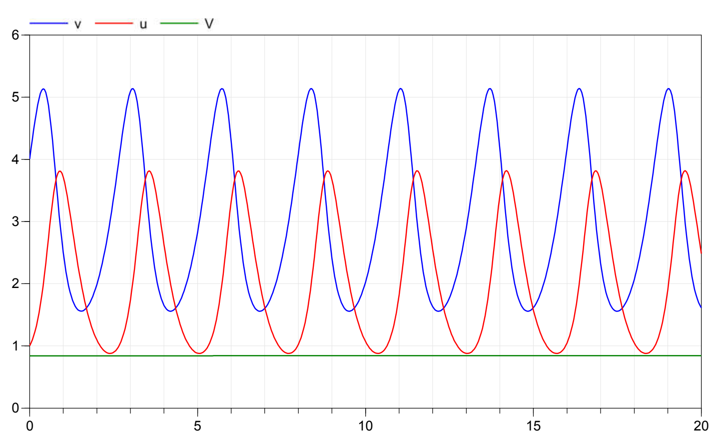

Figure 1: The system from 1 simulated in Dymola with $(u_0, v_0) = (1, 4)$.

1.3 c

Want to linearize the system around $(u^*, v^*) = (2, 3)$, so $\Delta u = u - u^*$, $\Delta v = v - v^*$, $\dot{u} = f_1$, $\dot{v} = f_2$.

$$\Delta\dot{u} = \frac{\partial f_1}{\partial u}\bigg|_{\substack{u\,=\,u^*\\ v\,=\,v^*}}\Delta u + \frac{\partial f_1}{\partial v}\bigg|_{\substack{u\,=\,u^*\\ v\,=\,v^*}} = 2(v-3)$$
$$\Delta\dot{v} = \frac{\partial f_2}{\partial u}\bigg|_{\substack{u\,=\,u^*\\ v\,=\,v^*}}\Delta u + \frac{\partial f_2}{\partial v}\bigg|_{\substack{u\,=\,u^*\\ v\,=\,v^*}} = 3(2-u)$$

This yields the linearized system

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \begin{bmatrix}0 & 2\\ -3 & 0\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix} + \begin{bmatrix}-6\\ 6\end{bmatrix}. \tag{2}$$

This system has the eigevalues $\lambda = \pm i\sqrt{6}$, so the linearized system is marginally stable.

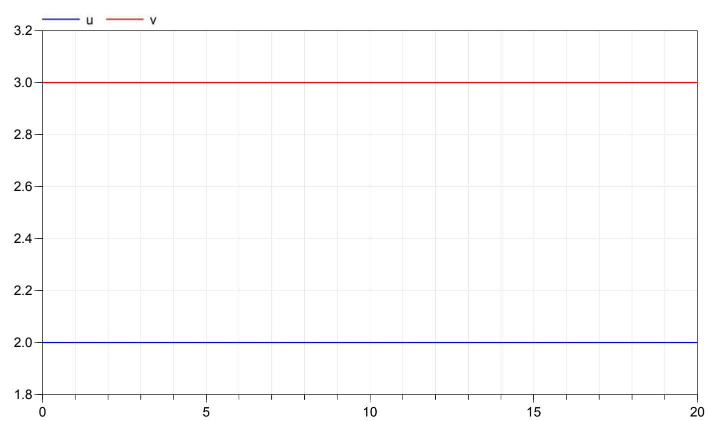

Figure 2: The linearized system from 2 simulated in Dymola with $(u_0, v_0) = (2, 3)$.

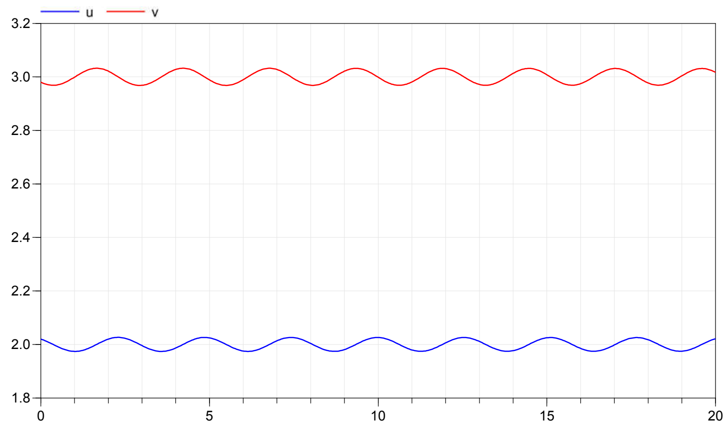

Figure 3: The linearized system from 2 simulated in Dymola with $(u_0, v_0) = (2.02, 2.98)$.

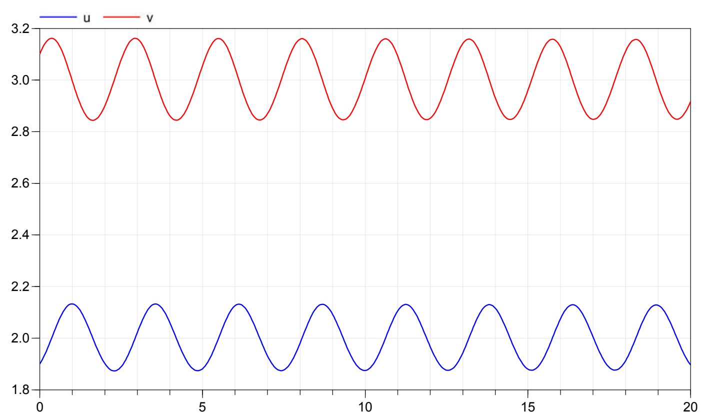

Figure 4: The linearized system from 2 simulated in Dymola with $(u_0, v_0) = (1.9, 3.1)$.

When simulating the linearized system with intital conditions $(u_0, v_0) = (2, 3)$, as seen in Figure 2, the solution is constant. When simulated with initial values with slight deviations from the equilibrium, as seen in figures 3 and 4 the linearized system is an oscillator with frequency close to $f = \frac{\sqrt{6}}{2\pi}$

1.4 d

The code for this excercise can be seen in Listing 1. From the phase plots in figures 5 to 7 and the plots of the "energy" in figures 8 and 9, we see that the explicit Euler method is unstable, while the implicit Euler method and the implicit midpoint rule are stable for this system with $h = 0.1$. For smaller h, the implicit Euler method has a constant V, but the explicit Euler method is unstable for all h when simulating this system.

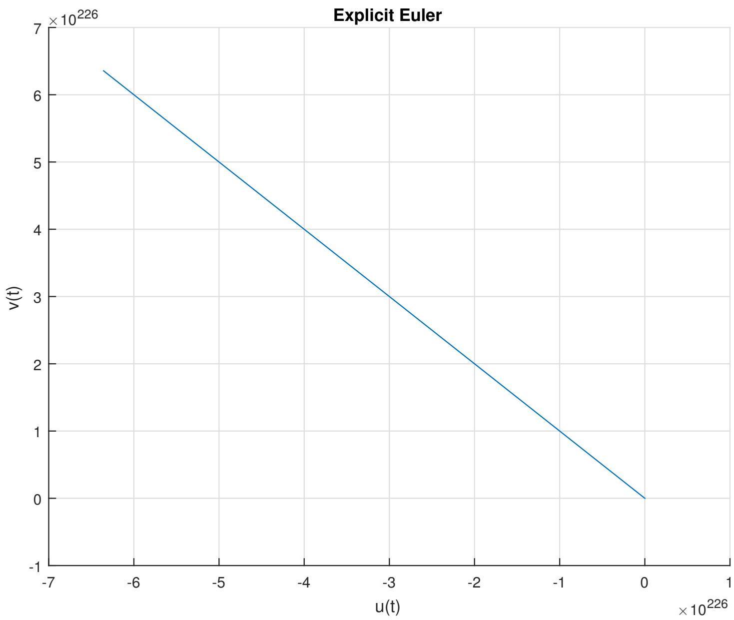

Figure 5: Phase plot of the system 1 simulated with the explicit Euler method.

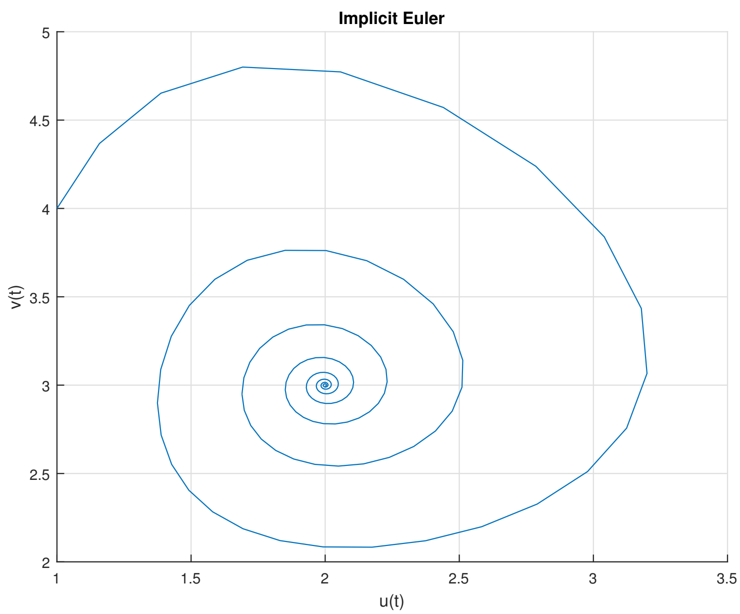

Figure 6: Phase plot of the system 1 simulated with the implicit Euler method.

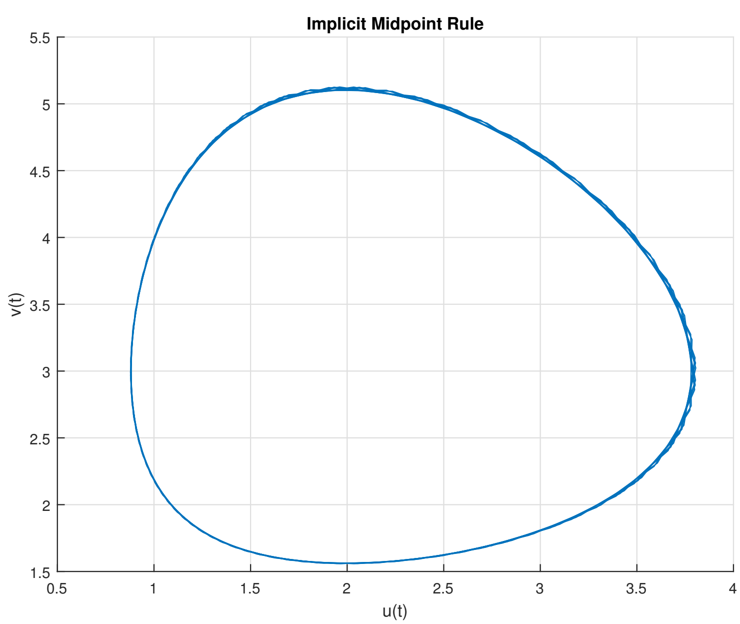

Figure 7: Phase plot of the system 1 simulated with the implicit midpoint rule.

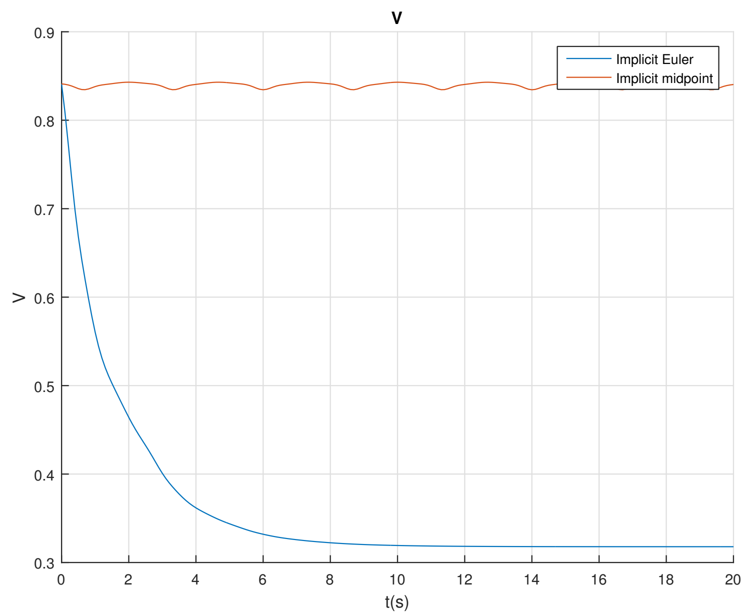

Figure 8: The ”energy” of the system 1 when simulated with the explicit Euler method.

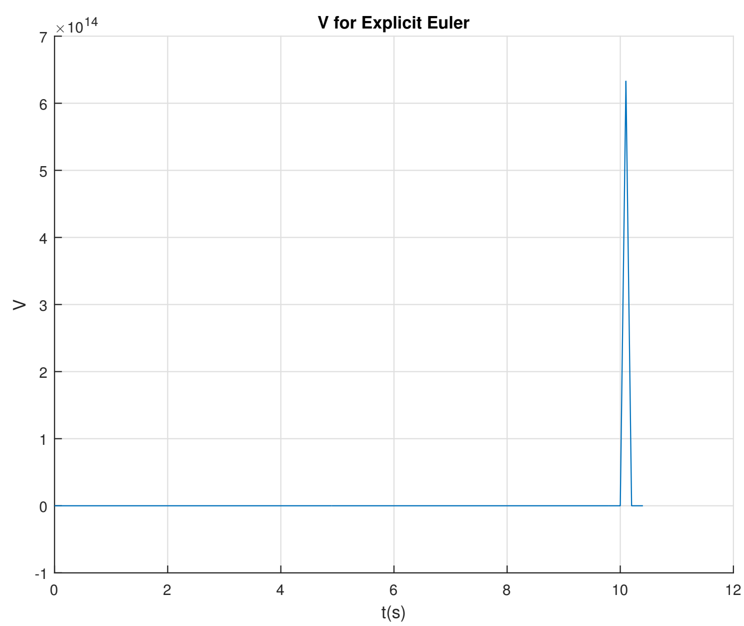

Figure 9: The ”energy” of the system 1 when simulated with the implicit Euler method adn midpoint rule.

Listing 1: Code for simulating the system from 1 in MATLAB with Euler and implicit Euler and implicit midpoint rule methods.

```
h = 0.01;
t = 0:h:20;
y0 = [1;4];

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(1).*(y(2) - 3);y(2).*(2 - y(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;

title('Explicit Euler');

plot(ya(1,:),ya(2,:));
xlabel('u(t)');
ylabel('v(t)');

print -depsc modsim_ex6_1d_ee.eps

figure;
hold on; grid on;

title('Implicit Euler');

plot(yb(1,:),yb(2,:));
xlabel('u(t)');
ylabel('v(t)');

print -depsc modsim_ex6_1d_ie.eps
```

```
figure;
hold on; grid on;

title('Implicit Midpoint Rule');

plot(yc(1,:),yc(2,:));
xlabel('u(t)');
ylabel('v(t)');
print -depsc modsim_ex6_1d_imr.eps

figure;
hold on; grid on;

title('V')

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

Eb = yb(1,:) - 2.*log(yb(1,:)) + yb(2,:) - 3.*log(yb(2,:));
Ec = yc(1,:) - 2.*log(yc(1,:)) + yc(2,:) - 3.*log(yc(2,:));

plot(t,Eb);
plot(t,Ec);
xlabel('t(s)');
ylabel('V');
legend('Implicit Euler','Implicit midpoint');

print -depsc modsim_ex6_1d_v.eps

figure;
hold on; grid on;

title('V for Explicit Euler')

Ea = ya(1,:) - 2.*log(ya(1,:)) + ya(2,:) - 3.*log(ya(2,:));

plot(t,Ea);
xlabel('t(s)');
ylabel('V');

print -depsc modsim_ex6_1d_vee.eps
```

2 Problem 2

2.1 a

Lobatto IIIA has stability function $R(s) = P_m^m(s)$. From the book we know that $|P_m^k(s)| \leq 1 \; \forall \; Re(s) \leq 0, \; k \leq m \leq k+2$. A method is A-stable if $|R(h\lambda)| \leq 1 \; \forall \; Re(h\lambda) \leq 0$, so by replacing s with $h\lambda$ and since $k = m$ for Lobatto IIIA, we see that it is clearly A-stable.

2.2 b

A method is L-stable if it is A-stable and $\lim_{\omega\to\infty} |R(hj\omega)| = 0$.

$$\begin{aligned}\lim_{\omega\to\infty} |R(hj\omega)| &= \frac{|1 + \gamma_1 j\omega h + ... + \gamma_m (j\omega h)^m|}{|1 + \beta_1 j\omega h + ... + \beta_m (j\omega h)^m|} \\ &= \frac{|\gamma_m|}{|\beta_m|} \neq 0,\end{aligned}$$

so Lobatto IIIA is not L-stable.

3 Problem 3

3.1 a

The method is implicit. This can be seen from the butcher array, as it is not lower triangular.

3.2 b

Fromt the butcher array it can be seen that

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{5}{12} \end{bmatrix} \\ \mathbf{b} &= \begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \end{bmatrix}\end{aligned}$$

From the book we know that

$$R(h\lambda) = (1 + h\lambda \mathbf{b}^T (\mathbf{I} - h\lambda \mathbf{A})^{-1} \mathbf{1}).$$

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

Table 1: Butcher array for Runge-Kutta method.

Inserting and calculating yields

$$\underline{\underline{R(h\lambda) = \frac{1 + \frac{1}{3}h\lambda}{1 - \frac{2}{3}h\lambda + \frac{1}{6}(h\lambda)^2} = P_2^1(h\lambda)}}$$

3.3 c

This method is L-stable according to Theorem (14.6.5) in the book, since $m = k + 1$. Therefore it is also A-stable.

4 Problem 4

4.1 a

$$\begin{aligned}\dot{m} &= -\rho q \\ q &= C_v\sqrt{p - p_0} = C_v\sqrt{\rho g h} \\ \dot{m} &= \dot{h}A\rho\end{aligned}$$

$$\underline{\underline{\dot{h} = -\frac{C_v}{A}\sqrt{\rho g h}}} \tag{3}$$

4.2 b

$$\underline{\underline{\Delta\dot{h} = \frac{\partial(\dot{h})}{\partial h}\bigg|_{h\,=\,h^*}\Delta h = -\frac{C_v\rho g}{2A\sqrt{\rho g h^*}}\Delta h}}$$

This linearized system had eigenvalue $\lambda = -\frac{C_v\rho g}{2A\sqrt{\rho g h^*}}$ which approaches $-\infty$ as $h^* \to 0$.

4.3 c

ode45 implements a Runge-Kutta method with variable step length. A seen in Figure 10 the steplength is very small in towards the end of the simulation, where $h \to 0$. The code for simulating this system can be seen in Listing 2.

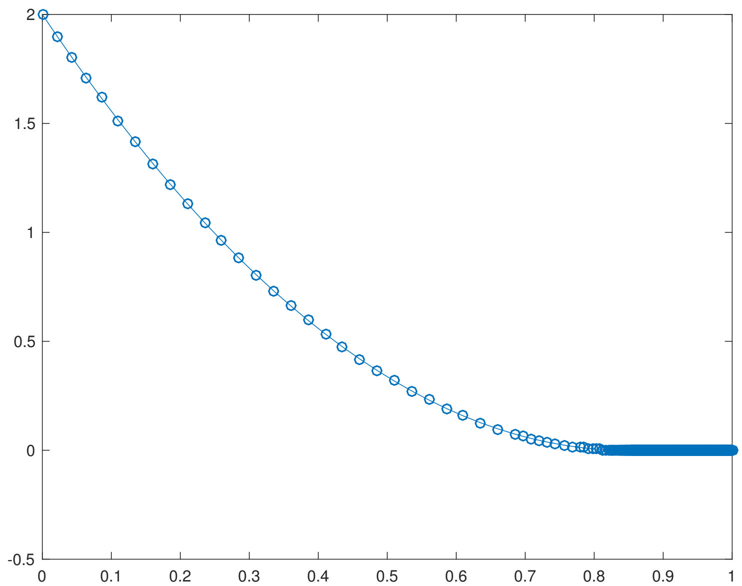

Figure 10: The system from Equation 3 simulated in MATLAB using ode45.

Listing 2: 4c

```
tspan = [0 1];
h0 = 2;
Cv = 0.15;
rho = 1000;
A = 4.5;
g = 10;


[t,h] = ode45(@(t,h) -(Cv/A)*sqrt(rho*g*h), tspan, h0);

plot(t,h,’-o’)
```