Out: January 29, 2014
In: February 12–21, 2014

NTNU – Trondheim
Norwegian University of
Science and Technology

Exercise 6
TTK4130 Modeling and Simulation

Problem 1 (Lotka-Volterra predator-prey)

An example of the Volterra-Lotka predator-prey model is

$$\dot{u} = u\,(v-2) \tag{1a}$$
$$\dot{v} = v(1-u) \tag{1b}$$

The equations can be used to describe the dynamics of biological systems in which two species interact, one a predator and one its prey. Here, u may represent the number of predators (in thousands), for instance foxes, and v number of preys in thousands, for instance rabbits.

Consider an island where foxes are the only predator, and rabbits the only prey. When the number of rabbits gets large ($v > 2$), the number of foxes will grow. But when the number of foxes becomes large ($u > 1$), the number of rabbits will decrease, which in turn make the number of foxes decrease. This leads to large, periodic variations in the two populations, with a "phase shift" between the number of foxes and rabits.

Consider the "energy-like" function[1]

$$V = u - \ln u + v - 2\ln v \tag{2}$$

(a) Calculate

$$\dot{V} = \frac{\partial V}{\partial u}\dot{u} + \frac{\partial V}{\partial v}\dot{v} \tag{3}$$

and show that V is constant for solutions of the Lotka-Volterra system (1). Make an attempt at an interpretation.

Solution: First, note that

$$\frac{\partial V}{\partial u} = \frac{u-1}{u}$$
$$\frac{\partial V}{\partial v} = \frac{v-2}{v}$$

which gives

$$\dot{V} = \left(\frac{u-1}{u}\right)\dot{u} + \left(\frac{v-2}{v}\right)\dot{v}.$$

Insert the system equations to get

$$\begin{aligned}\dot{V} &= \left(\frac{u-1}{u}\right)u\,(v-2) + \left(\frac{v-2}{v}\right)v(1-u)\\ &= (u-1)\,(v-2) + (v-2)\,(1-u)\\ &= 0\end{aligned}$$

which imply that V is constant along solutions of the Lotka-Volterra system (1).

That $\dot{V} = 0$ (or V is constant) along with the definiteness of V implies that the trajectories of the system (1) are stable, that is, that they will not grow unbounded. (Technically, we would resort to LaSalle's theorem here, but this is not part of this course.)

[1] Strictly speaking this function is not positive definite. However, we can show that the states of the Lotka-Volterra system will remain positive if initialized positive, and for positive u and v, V will be positive (in general, after addition of a constant). Therefore, think of V as measure of "energy" (total population) in the system.

(b) Implement the system in Dymola, and simulate over 20 time units. Use $u = 1$ and $v = 4$ as initial conditions. Comment on the trajectories of u and v, and verify that V is constant. (That is, in addition to implementation of the two Lotka-Volterra equations, implement the equation for V.)

Solution: A proposal:

```
model LotkaVolterra
  Real u(start=1);
  Real v(start=4);
  Real V;
equation
  der(u) = u*(v - 2);
  der(v) = v*(1 - u);
  V = u - ln(u) + v - 2*ln(v);
end LotkaVolterra;
```

Simulation reveals periodic trajectories of u and v (number of foxes and rabbits), and that V is constant. See Figure 1.

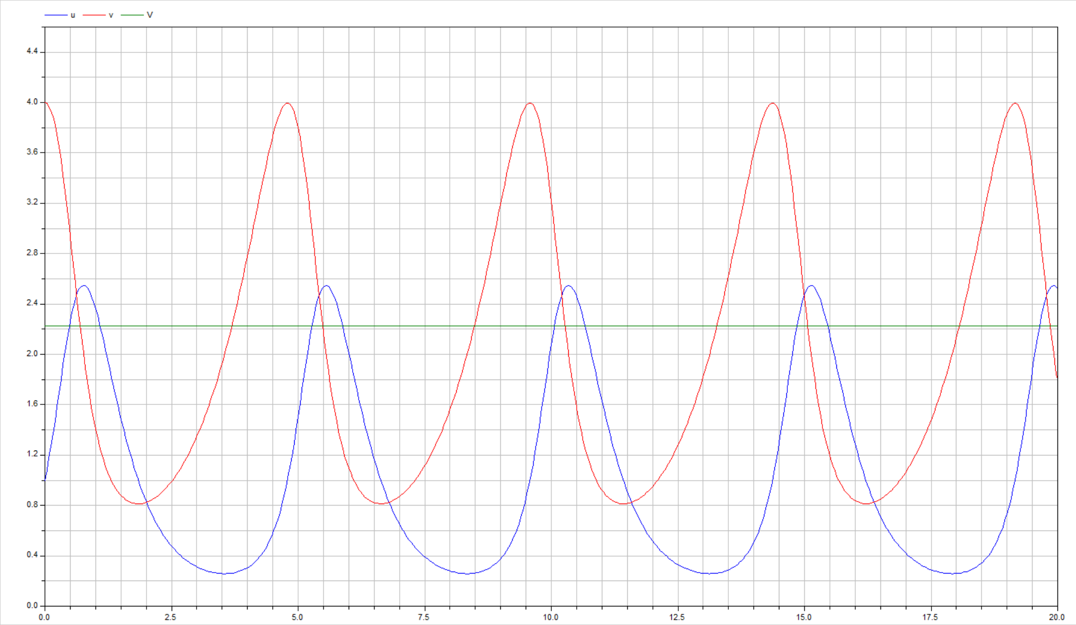


Figure 1: Dymola simulation of the Lotka-Volterra system

(c) Linearize the system about the equilibrium $\{u = u^* = 1, v = v^* = 2\}$, and calculate eigenvalues. Simulate the system at this equilibrium, and with small deviations from it, and comment on connection between eigenvalues and periods/frequencies.

Solution: Linearizing, we get

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} \\ \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} \end{bmatrix}_{(v^*, u^*)} = \begin{bmatrix} (v^* - 2) & u^* \\ -v^* & (1 - u^*) \end{bmatrix}, \tag{4}$$

which has the characteristic equation

$$\lambda^2 + \lambda\,(u^* - v^* + 1) + (2u^* + v^* - 2) = 0.$$

Inserting $u^* = 1$ and $v^* = 2$, we find

$$\lambda_{1,2} = \pm j\omega = \pm j\sqrt{2} \approx \pm j1.41.$$

That is, the (linearized) system is marginally stable, and close to the equilibrium the system behaves like an oscillator, with period $T = \frac{2\pi}{\sqrt{2}} \approx 4.44$ s. This is confirmed by simulations: By starting in the equilibrium, everything is constant, while starting slightly off leads to oscillations with period T. See Figure 2.

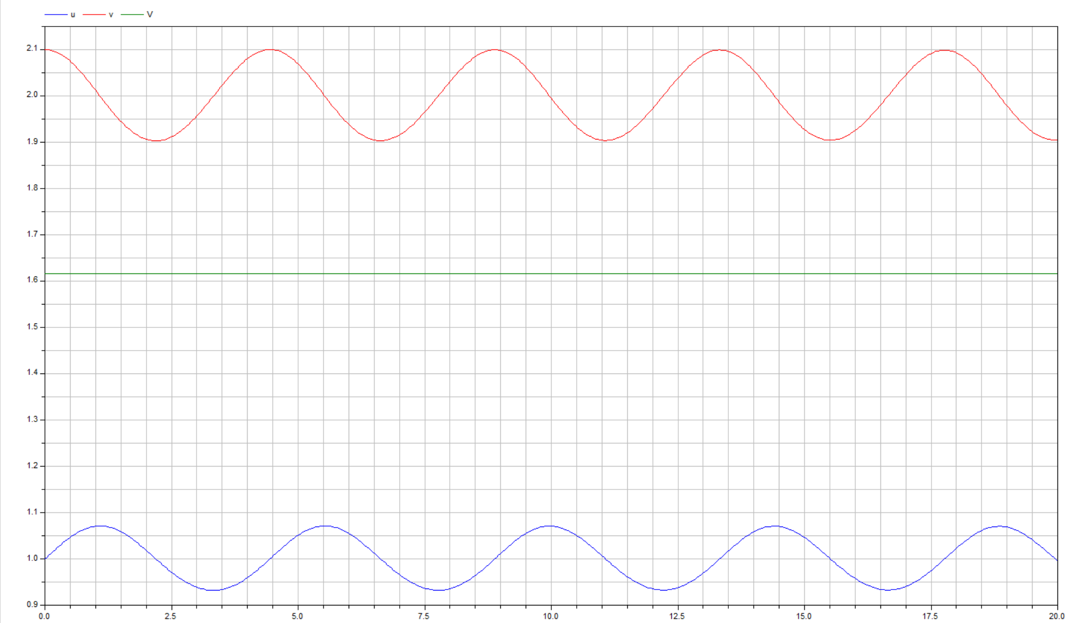


Figure 2: Dymola simulation of the Lotka-Volterra system, starting from $u = 1$, $v = 2.1$.

(d) Simulate this model (initial conditions and horizon as in (b)) in Matlab using Euler, implicit Euler, and the implicit midpoint rule. Re-use code from Exercise 4. Make plots of u and v (for instance in phase plots) and V for the different numerical solutions. Use $h = 0.1$ (for instance, but experiment). Comment!

Solution: The code below contains code for simulation and plotting:

```
% Parameters
u0 = 1; v0 = 4; y0 = [u0;v0]; % Initial values
h = 0.1; % Step size
t0 = 0; tstop = 100; % Time start and stop
time = t0:h:tstop; % Generate time vector
nstep = ((tstop-t0)/h)+1;
opt = optimset('Display','off','TolFun',1e-8); % Options for fsolve

% System
f = @(y,t) ( [ y(1)*(y(2) - 2); y(2)*(1 - y(1)) ] );

%% Create storage
y_EE = zeros(size(y0,1),size(time,2)); % Explicit Euler
y_IE = zeros(size(y0,1),size(time,2)); % Implicit Euler
y_IM = zeros(size(y0,1),size(time,2)); % Implicit Midpoint rule
```

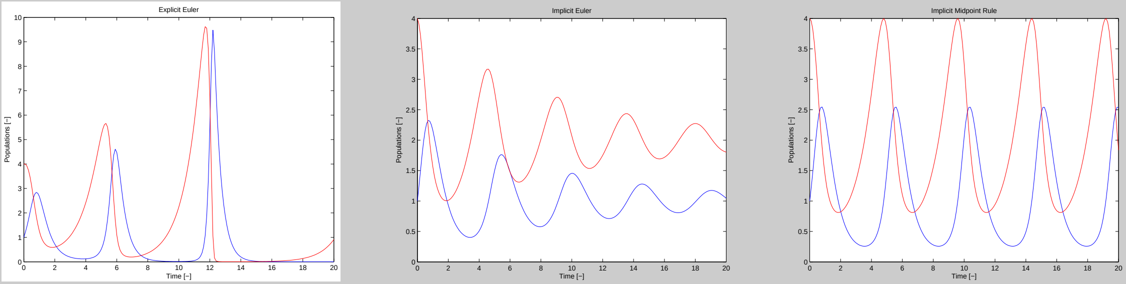
```
%% EXPLICIT EULER
y_EE(:,1) = y0; % Initial value
for i = 1:nstep-1,
    y_EE(:,i+1) = y_EE(:,i) + h*feval(f,y_EE(:,i),time(i));
end
% Plots the results for Explicit Euler
figure(1); plot(time,y_EE(1,:),time,y_EE(2,:),’r’);
xlabel(’Time [-]’); ylabel(’Populations [-]’); title(’Explicit Euler’)

%% IMPLICIT EULER
y_IE(:,1) = y0; % Initial value is set.
for i = 1:nstep-1,
    r = @(y) (y_IE(:,i) + h*feval(f,y,time(i+1)) - y); % Root function
    [y_IE(:,i+1),fval,exitflag,output] = fsolve(r,y_IE(:,i),opt);
end
% Plots the results for Implicit Euler
figure(2); plot(time,y_IE(1,:),time,y_IE(2,:),’r’);
xlabel(’Time [-]’); ylabel(’Populations [-]’); title(’Implicit Euler’)

%% IMPLICIT MIDPOINT RULE
y_IM(:,1) = y0; % Initial value is set.
for i = 1:nstep-1,
    r = @(y) (y_IM(:,i) + h*feval(f,(y_IM(:,i) + y)/2,time(i)+h/2) - y);
    [y_IM(:,i+1),fval,exitflag,output] = fsolve(r, y_IM(:,i), opt);
end
% Plots the results for Implicit Midpoint Rule
figure(3); plot(time,y_IM(1,:),time,y_IM(2,:),’r’);
xlabel(’Time [-]’); ylabel(’Populations [-]’); title(’Implicit Midpoint Rule’)

%% Plot energies
V_EE = y_EE(1,:) - log(y_EE(1,:)) + y_EE(2,:) - 2*log(y_EE(2,:));
V_IE = y_IE(1,:) - log(y_IE(1,:)) + y_IE(2,:) - 2*log(y_IE(2,:));
V_IM = y_IM(1,:) - log(y_IM(1,:)) + y_IM(2,:) - 2*log(y_IM(2,:));
figure(4); plot(time,V_EE); hold on;
plot(time,V_IE,’m--’); plot(time,V_IM,’c:’); hold off;
legend(’Explicit Euler’,’Implicit Euler’,...
    ’Implicit Midpoint Rule (Gauss 2)’,’Location’,’NorthWest’);
xlabel(’Time [-]’); ylabel(’V [-]’);
```

The trajectories for the three methods are shown in Figure 3, and V for the three methods are shown in Figure 4.



(a) Explicit Euler (b) Implicit Euler (c) Implicit midpoint rule

Figure 3: Simulation of Lotka-Volterra system using different Runge-Kutta methods

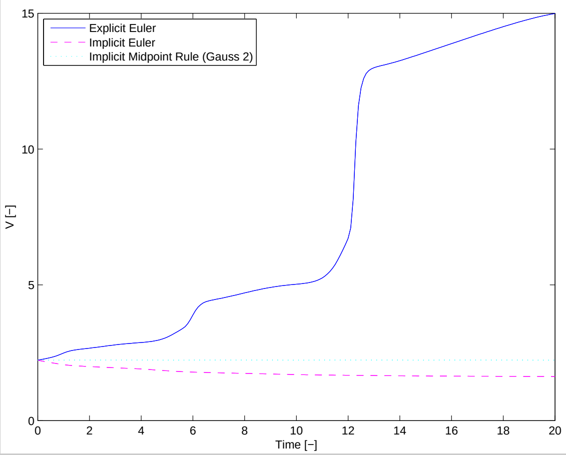


Figure 4: Plot of ”Energy” V for the three Runge-Kutta methods

Problem 2 (Lobatto IIIA)

(a) Show that Lobatto IIIA is A-stable, by showing that the conditions for A-stability is fulfilled.

Solution: The Butcher array for Lobatto IIIA is

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

To check for A-stability, we first have to calculate the stability function. We use the formula (book (14.142), page 539)

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

We get

$$\begin{aligned}
R(h\lambda) &= 1 + h\lambda \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2}h\lambda & 1 - \frac{1}{2}h\lambda \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= 1 + \frac{h\lambda}{1 - \frac{1}{2}h\lambda} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 - \frac{1}{2}h\lambda & 0 \\ \frac{1}{2}h\lambda & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= 1 + \frac{h\lambda}{1 - \frac{1}{2}h\lambda} \\
&= \frac{1 + \frac{1}{2}\lambda h}{1 - \frac{1}{2}\lambda h}.
\end{aligned}$$

For A-stability, we require

$$|R(h\lambda)| \leq 1, \quad \forall \mathrm{Re}(\lambda) \leq 0.$$

If $\mathrm{Re}(h\lambda) \leq 0$, the absolute value of the numerator of $R(h\lambda)$ will be larger or equal to the denominator. This implies A-stability (as we would expect, for example from the table on page 560 in the book).

(b) Is Lobatto IIIA L-stable? Calculations are required.

Solution: The method is L-stable if it is A-stable and $|R(j\omega h)| \to 0$ when $\omega \to \infty$:

$$\begin{aligned}
\lim_{\omega\to\infty} |R(j\omega h)| &= \lim_{\omega\to\infty} \left| \frac{1 + \frac{1}{2}j\omega h}{1 - \frac{1}{2}j\omega h} \right| \\
&= \lim_{\omega\to\infty} \left| \frac{\frac{1}{\omega} + \frac{1}{2}jh}{\frac{1}{\omega} - \frac{1}{2}jh} \right| \\
&= \left| \frac{\frac{1}{2}jh}{-\frac{1}{2}jh} \right| = 1.
\end{aligned}$$

Since this does not approach 0 when $\omega \to \infty$, the method is not L-stable.

Problem 3 (Linear stability of Runge-Kutta methods)

Given the Runge-Kutta method

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

(a) Is this an explicit or implicit method? Why?

Solution: Since the A-matrix has non-zero elements on and above the diagonal, the stage computations cannot be performed explicitly. Therefore, it is an implicit method.

(b) Find the stability function for the method. Is this a Padé-approximation?

Solution: Use

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

where

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

This gives

$$\begin{aligned} R(h\lambda) &= \frac{1 + \frac{1}{3}h\lambda}{1 - \frac{2}{3}h\lambda + \frac{1}{6}(h\lambda)^2} \\ &= P_2^1(h\lambda) \end{aligned}$$

where $P_m^k(s)$ is the Padé-approximation (to e^s) with numerator of degree k and denominator of degree m (see Table 14.2 in the book).

(c) Is this method A-stable? L-stable? Justify your answers, not necessarily with computations.

Solution: From Section 14.6.5 in the book, we know that methods with $R(s) = P_m^k(s)$ for $m = k+1$ are L-stable (and therefore also A-stable).

Problem 4 (Tank and valve-model)

In this problem, we will model a tank that is being emptied through valve. See Figure 5.

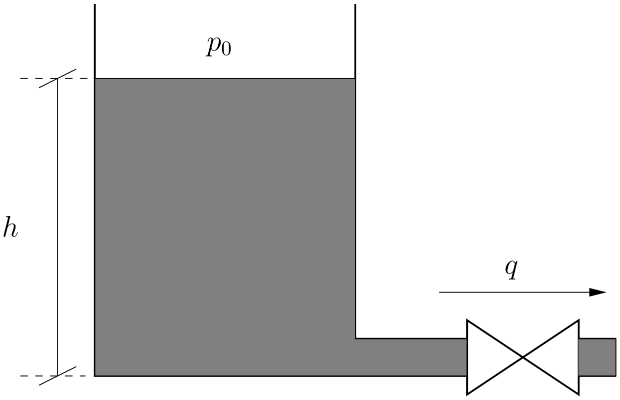


Figure 5: Tank being emptied through a valve

We model the valve with the valve equation

$$q = C_v\sqrt{p - p_0}.$$

We assume constant density $\rho = 1000\text{kg/m}^3$, tank area $A = 3.5\text{m}^2$, gravity of acceleration $g = 10\text{m/s}^2$, valve constant $C_v = 0.1\text{m}^3/(\text{s}\cdot\sqrt{\text{Pa}})$, and outside pressure is athmospheric $p_0 = 10^5\text{Pa}$.

(a) Using a mass balance for the tank, show that the level h is modeled by the differential equation

$$\frac{\mathrm{d}h}{\mathrm{d}t} = -\frac{C_v}{A}\sqrt{\rho g h}.$$

Solution: A mass balance is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\rho A h\right) = -\rho q.$$

Together with the valve equation and $p = p_0 + \rho g h$, this gives the result.

(b) Linearize the system. What happens to the eigenvalue as $h \to 0$?

Solution: The linearized model (around $h = h^*$) is

$$\Delta\dot{h} = -\frac{C_v \rho g}{2A\sqrt{\rho g h^*}}\Delta h.$$

The eigenvalue is

$$\lambda = -\frac{C_v \rho g}{2A\sqrt{\rho g h^*}},$$

which goes to minus infinity as $h^* \to 0$ (from above).

(c) Implement and simulate the model in Matlab, using the ODE solver `ode45`. Use initial condition $h(t = 0) = 2$, and simulate for one second. To show the steplengths, use the plot-command `plot(t,h,’o-’)`. Comment on the steplengths in view of the answer in (b).

Solution: A Matlab script simulating the model, is:

```
Cv = .1; A = 3.5; rho = 1000;
g = 10; Po = 1;

% ODE
f = @ (t,h) ( -(Cv/A) * sqrt(rho*g*h) );

% Simulate
[t,h] = ode45(f,[0 1],2,options);
```

A plot is shown in Figure 6.

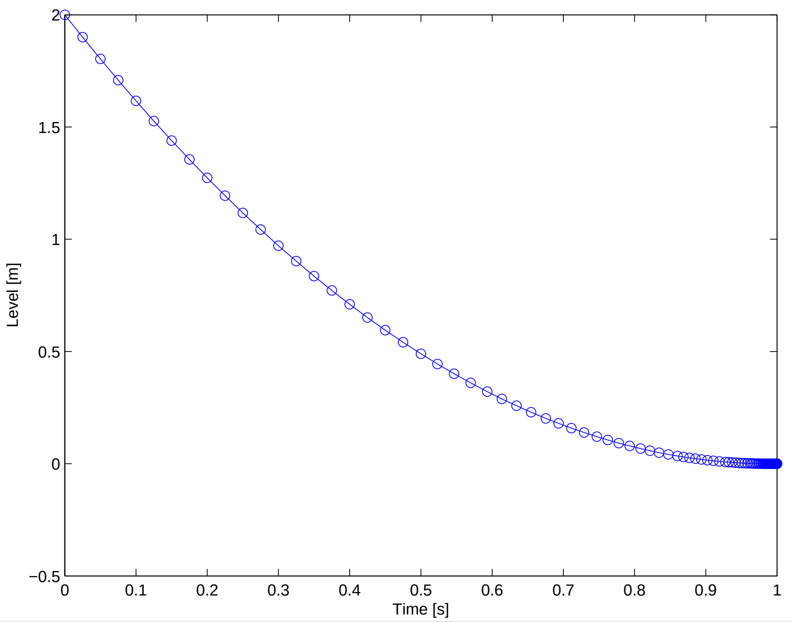


Figure 6: Simulating the level. The 'O' marks each step by the ODE solver.

As we can see, as $h \to 0$, the step sizes becomes smaller. This is in correspondence with the eigenvalues approaching infinity.

Remark: This valve model is not good for small flows, for instance, the solution will typically become complex as the model (square root) is evaluated for negative h. An explanation for this problem is that the valve equation is based on an assumption of turbulent flows, and for smaller flows the flow will at some point become laminar. The solution to this problem is to regularize the valve model, for instance as explained in Section 4.2.3 in the book.