Problem Sheet 2

Problem 2.1 Hamiltonian of the Lotka-Volterra Differential Equation

In [NODE, Eq. 1.4], we got to know the Lotka-Volterra differential equation,

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

as an example for a differential equation with a non-trivial first integral $I(u, v) = \ln u - u + 2\ln v - v$. A further more in-depth structural quality of this differential equation is the subject of this exercise and will hopefully shine a new light on the invariant.

(2.1a) What differential equation are satisfied by the functions $p = \ln u$ and $q = \ln v$, where u and v are solutions of (2.1)?

Solution:
By the chain rule, we obtain

$$\begin{aligned} \dot{p} &= \tfrac{1}{u}\dot{u} = v - 2 = \mathrm{e}^q - 2, \\ \dot{q} &= \tfrac{1}{v}\dot{v} = 1 - u = 1 - \mathrm{e}^p. \end{aligned}$$

From this, we see that p and q satisfy the following differential equation

$$\begin{aligned} \dot{p} &= \mathrm{e}^q - 2, \\ \dot{q} &= 1 - \mathrm{e}^p. \end{aligned} \tag{2.1.1}$$

(2.1b) Show that the differential equation found in subproblem (2.1a) is Hamiltonian (c.f. [NODE, Def. 1.2.3]) and give the corresponding Hamiltonian function $H(p, q)$.

HINT: Apply the transformation $p = \ln u$ and $q = \ln v$ to the invariant $I(u, v)$
Solution:
We have

$$\begin{aligned} I(u, v) &= \ln u - u + 2\ln v - v \\ &= p - \mathrm{e}^p + 2q - \mathrm{e}^q \\ &= I(p, q). \end{aligned}$$

$I(p, q)$ is now a Hamiltonian of the system (2.1.1). In fact, it holds that

$$\frac{\partial H}{\partial p} = 1 - \mathrm{e}^p = \dot{q}, \qquad \frac{\partial H}{\partial q} = 2 - \mathrm{e}^q = -\dot{p}.$$

Problem 2.2 Initial Value Problem With Cross Product

We will observe the initial value problem

$$\dot{\mathbf{y}} = f(\mathbf{y}) := \mathbf{a} \times \mathbf{y} + c\mathbf{y} \times (\mathbf{a} \times \mathbf{y}), \quad \mathbf{y}(0) = \tfrac{\sqrt{3}}{3}(1,1,1)^\top, \tag{2.2.1}$$

where $c > 0$ and $\mathbf{a} \in \mathbb{R}^3 \setminus \{0\}$.

Note: $\mathbf{x} \times \mathbf{y}$ denotes the cross product between the vectors $\mathbf{x}$ and $\mathbf{y}$. It satisfies $\mathbf{x} \times \mathbf{y} \perp \mathbf{x}$. In MATLAB, it is available as the function `cross(x,y)`.

(2.2a) Show that the solution of (2.2.1) exists for all points in time and that $\|\mathbf{y}(t)\| = 1$, $\forall t$ holds.

Solution: We have

$$\begin{aligned}
&(\mathbf{a} \times \mathbf{y} + c(\mathbf{y} \times (\mathbf{a} \times \mathbf{y}))) \cdot \mathbf{y} = 0 \\
&\Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{y}(t)\|_2^2 = \dot{\mathbf{y}}(t) \cdot \mathbf{y}(t) = 0 \\
&\Rightarrow \|\mathbf{y}(t)\|_2^2 = \text{const. } \forall t,
\end{aligned}$$

meaning, that the exact evolution of the differential equation is length-preserving and ([NODE, Def. 1.3.1]) the solution must exist for all points in time: There can be no “collapse” nor “blow–up”.

(2.2b) Is $\mathbf{a}$ an attractive fixed point of (2.2.1)? Explain your answer.

HINT: Use the result from subproblem (2.2a)

Solution: No, as no length-preserving evolution can have an attractive fixed-point.

(2.2c) For $\mathbf{a} = (1,0,0)^\top$, (2.2.1) was solved with the standard MATLAB integrators `ode45` and `ode23s` up to the point $T = 10$ (default Tolerances). Explain the different dependence of the total number of steps from the parameter c observed in Figure 2.1.

Solution: In the plot, we see that, for the solver `ode45`, the number of steps rises with c. On the other hand, `ode23s` uses roughly the same amount of steps, irregardless of the value of c chosen.

Explanation: As `ode45` is an explicit solver, it suffers under stability-based step-size limits for large $c > 0$. The implicit solver `ode23s` must however not take smaller step-sizes to satisfy the tolerance for big c. In other words: The problem becomes stiffer, the greater the parameter c is.

(2.2d) Formulate the non-linear equation given by the implicit mid-point rule for the initial value problem (2.2.1).

Solution: With the formula for the implicit mid-point rule

$$\mathbf{y}_{k+1} = \mathbf{y}_k + hf\left(t_{k+\frac{1}{2}}, \frac{\mathbf{y}_k + \mathbf{y}_{k+1}}{2}\right)$$

the formulation of (2.2.1) is given as

$$\frac{\mathbf{y}_{k+1} - \mathbf{y}_k}{h} = \mathbf{a} \times \left(\frac{\mathbf{y}_k + \mathbf{y}_{k+1}}{2}\right) + c\left(\frac{\mathbf{y}_k + \mathbf{y}_{k+1}}{2}\right) \times \left(\mathbf{a} \times \left(\frac{\mathbf{y}_k + \mathbf{y}_{k+1}}{2}\right)\right)$$

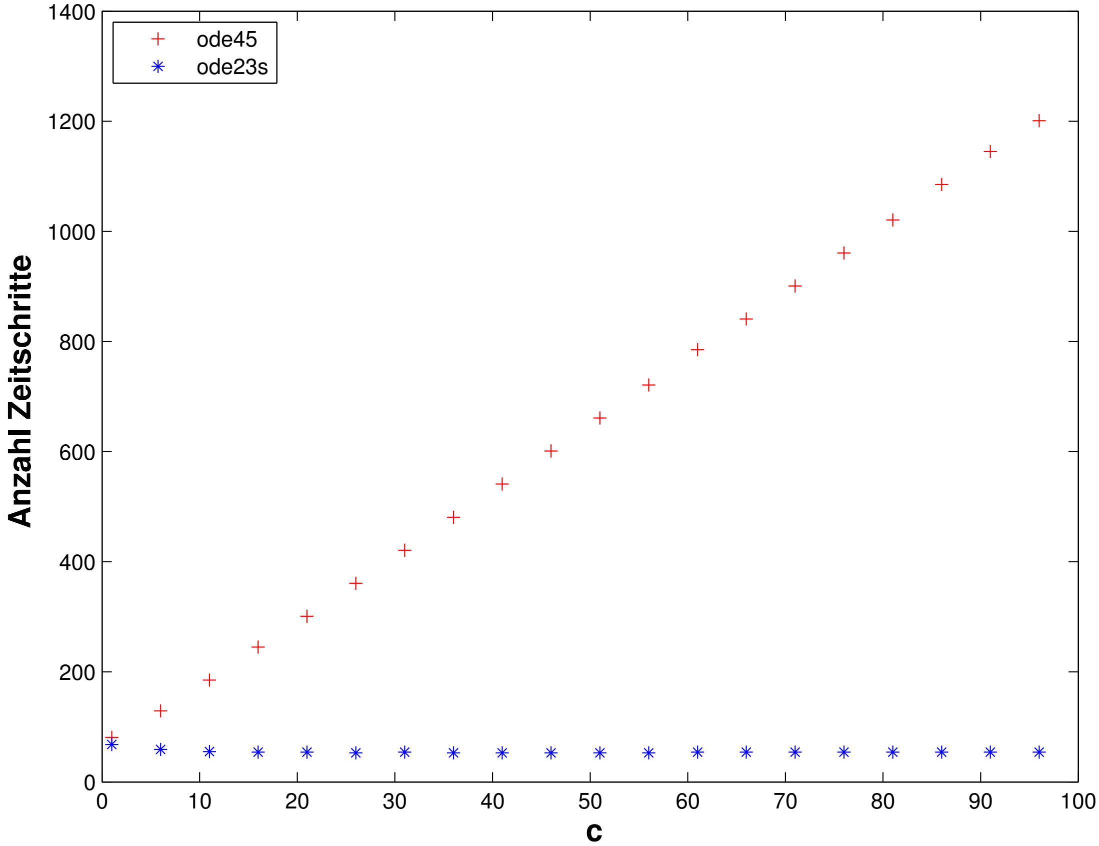

Figure 2.1: To subproblem (2.2c): number of steps used by standard MATLAB integrators in relation to the parameter c.

(2.2e) The linear-implicit mid-point rule for (2.2.1) can be received by a simple linearisation of the incremental equation of the implicit mid-point rule by the current solution value.

Give the defining equation of the linear-implicit mid-point rule for the general autonomous differential equation

$$\dot{\mathbf{y}} = f(\mathbf{y})$$

with smooth f.

Solution: The linear implicit mid-point rule is preserved, by developing the increment $\mathbf{k}{-}_1$ of the implicit mid point rule by its Taylor series

$$\begin{aligned}\mathbf{k}_1 &= f\left(\mathbf{y}_k + \frac{h}{2}\mathbf{k}_1\right) \\ &= f(\mathbf{y}_k) + \frac{h}{2}Df(\mathbf{y}_k)\mathbf{k}_1 + O(h^2)\end{aligned}$$

and only taking the linear terms. By reformulation we receive

$$\begin{aligned}\mathbf{k}_1^{\text{lin. IMP}} &:= \left(I - \frac{h}{2}Df(\mathbf{y}_k)\right)^{-1} f(\mathbf{y}_k) \\ \mathbf{y}_{k+1} &= \mathbf{y}_k + h\mathbf{k}_1^{\text{lin. IMP}}\end{aligned}$$

(2.2f) Complete the implementation of the linear–implicit mid-point rule in the MATLAB–template `LinImpMidPointSolve.m`.

Solve (2.2.1) in the MATLAB–template `linimpmprsol.m` for the values $\mathbf{a} = (1, 0, 0)^{\top}$ and $c = 1$ as well as with `LinImpMidPointSolve.m` with step size $h = 0.1$ as well as with the MATLAB solver `ode45` up to the point $T = 5$. Plot the first component y_1 for both solutions.

Solution:

Relevant code in `LinImpMidPointSolveSol.m`:

```
for j=1:N
  y(:,j+1) = LinImpMidPointStep(f,df,y(:,j),t(j),h);
end

function y = LinImpMidPointStep(f,df,y0,t0,h)
DF = df(t0,y0);
k1 = ( eye(size(DF)) - h/2*DF )\f(t0,y0);
% oder %
% k1 = inv(eye(size(DF))-h/2*DF)*f(t0,y0)
y = y0 + h*k1;
```

See file `linimpmprSol.m` (Listing 2.1)

Listing 2.1: code of `linimpmprSol.m`

```
function a3fSol
%
%  Aufgabe 3f.
%  Name:
%
% A3F solves the initial value problem from 3e with the linear implicit
% midpoint rule and the matlab solver ode45.

  y0 = [1;1;1]/sqrt(3);
  T = [0; 5];
  h = 0.1;

% Implement the right hand side and Jacobi matrix here:
% Note: t is a 'dummy' variable to make the functions ode45-compatible.
  a = [1;0;0]; c = 1;
  f = @(t,y) cross(a,y) + c*cross(y,cross(a,y));
% or
% f = @(t,y) [c*(y(2)^2 + y(3)^2);  -(y(3) + c*y(1)*y(2)); y(2) - c*y(1)*y(3);];
% Jacobi for a=(1,0,0)
  df = @(t,y) [0, 2*c*y(2), 2*c*y(3); -c*y(2), -c*y(1), -1; -c*y(3), 1, -c*y(1)];

```

```
% Solve the IVP with LinImpMidPointSolve and ode45 here:
[t,yLinIMP] = LinImpMidPointSolveSol(f,df,y0,T,h);
[tRef,yRef] =
   ode45(f,T,y0,odeset('Abstol',1e-8,'Reltol',1e-8));

% Plot the first component of the solutions here:
figure;
plot(t,yLinIMP(1,:),'+-',tRef,yRef(:,1));
title('{\bf Aufgabe 3: Solution with ode45 and
   LinImpMidPointSolveSol}')
legend('LinIMP','ode45');
```

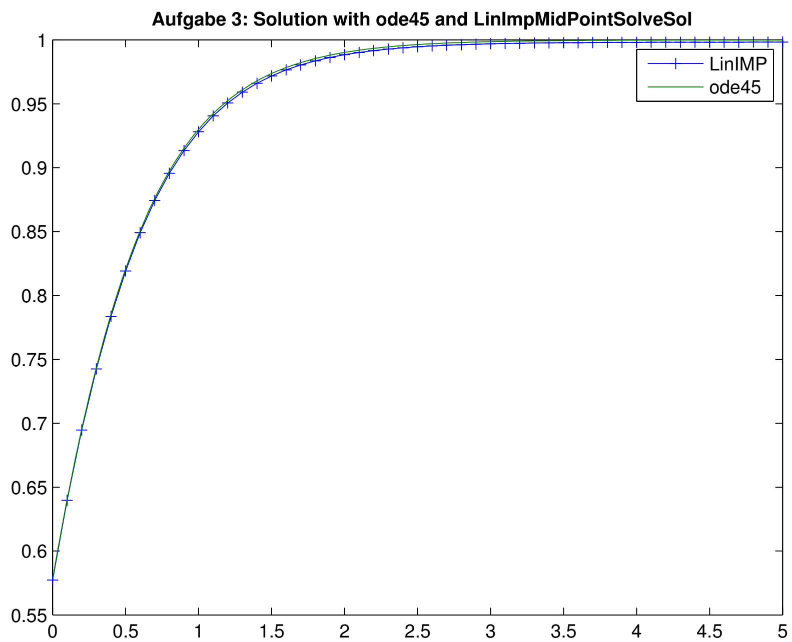

Figure 2.2: subproblem (2.2f)

Problem 2.3 Transforming the Thomas-Fermi Differential Equation

Let the *Thomas-Fermi* differential equation

$$\ddot{y}(t) = \frac{y^{\frac{3}{2}}(t)}{t^{\frac{1}{2}}}, \quad y(0) = y_0 \neq 0, \quad \dot{y}(0) = z_0 \tag{2.3.1}$$

be given. This form does not satisfy the local Lipschitz condition (cf.[NODE, Def. 1.3.4], [NODE, Ex. 1.3.5], [NODE, Thm. 1.3.6]) and is furthermore not suitable as an entry for numerical integration.

Transform (2.3.1) into a system of 1^{st} order that satisfies the Lipschitz condition.

HINT: Use the substitution

$$s = t^{\frac{1}{2}}, \quad y(t) = w(s), \quad u(s) = \frac{w'(s)}{s}.$$

Solution: We see, that the original system is not Lipschitz-continuous at $t = 0$.

Let $s = t^{\frac{1}{2}}$, $y(t) = w(s)$, $u(s) = \frac{w'(s)}{s}$. Using the chain rule, it follows that

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}t^2}(w(s(t))) &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}}{\mathrm{d}s}(w(s(t))) \cdot \frac{\mathrm{d}}{\mathrm{d}t}s(t)\right) \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\frac{w'(s(t))}{s}\right) \\
&= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}u(s(t)) \\
&= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}s}(u(s(t))) \cdot \frac{\mathrm{d}}{\mathrm{d}t}s(t) \\
&= \frac{1}{4}\frac{u'(s(t))}{s}
\end{aligned}$$

and so

$$u'(s) = 4s\frac{\mathrm{d}^2}{\mathrm{d}t^2}(w(s(t))) = 4s\ddot{y}(t) = 4w^{\frac{3}{2}}(s).$$

The transformed system is thus

$$w'(s) = su(s), \qquad w(0) = y_0 \tag{2.3.2}$$

$$u'(s) = 4w^{\frac{3}{2}}(s), \qquad u(0) = 2z_0, \tag{2.3.3}$$

where the initial value $u(0)$ follows from

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}w(s(t)) = \frac{u(s(t))}{2}.$$

The right hand side is locally Lipschitz-continuous by [NUMODE, Lem. 1.3.3], as

$$f(s, u(s), w(s)) = \begin{pmatrix} su(s) \\ 4w^{\frac{3}{2}}(s) \end{pmatrix}$$

and

$$D_{u,w}f(s, u(s), w(s)) = \begin{pmatrix} s & 0 \\ 0 & 6w^{\frac{1}{2}}(s) \end{pmatrix}$$

are continuous on $\Omega = \overline{(0, s) \times (y_0, u(s)) \times (2z_0, w(s))}$.

Published on 29 February 2016.

To be submitted by 8 March 2016.

References

[NUMODE] Lecture Slides for the course "Numerical Methods for Ordinary Differential Equations", SVN revision # 63606.

[NODE] Lecture Notes for the course "Numerical Methods for Ordinary Differential Equations".

Last modified on April 22, 2016