Problem Sheet 1

Introduction. Templates and `.p` files regarding the MATLAB problems can be found in the `.zip` file, which is on the website.

Problem 1.1 Collapsing Initial Value Problem

We will look at the following initial value problem from the lecture

$$\dot{y} = \sin(1/y) - 2, \quad y(0) = 1.$$

Show, using the integral's monotonicity, that the solution $y \in \mathcal{C}^1([0, t_+[, \mathbb{R})$ satisfies the inequality

$$y(t) \leq 1 - t \text{ for } 0 \leq t < t_+.$$

Explain why the solution necessarily experiences a collapse.

Problem 1.2 Cannon

Situated at the coordinate origin of the (x_1, x_2)-plane is a cannon which fires projectiles with initial velocity v_0 at an angle α with respect to the x_1-axis. The goal is to hit a target placed at the point $(250, 0)$. To model the trajectory of the projectile of mass m we assume that it is affected by the gravitational force mg, acting in negative x_2-direction, as well as the air resistance of magnitude ρv^2, which is directed opposite to the instantaneous flight direction at all times. Newton's second law yields the autonomous [NODE, Def. 1.1.5] ordinary differential equation of second order.

$$\ddot{\mathbf{x}} = g \begin{bmatrix} 0 \\ -1 \end{bmatrix} - \frac{\rho}{m} \|\dot{\mathbf{x}}\| \dot{\mathbf{x}} \tag{1.2.1}$$

describing the projectile's trajectory $\mathbf{x}(t)$. In addition, we obtain the initial values

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \dot{\mathbf{x}}(0) = v_0 \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}.$$

In the following, we will assume that $v_0 = 100$, $g = 10$, $m = 20$ and $\rho = 0.1$.

(1.2a) Convert (1.2.1) into an equivalent autonomous first order system $\dot{\mathbf{y}} = f(\mathbf{y})$, see [NODE, Rem. 1.1.6]. Determine the corresponding initial values for the first order system.

(1.2b) Implement a MATLAB function

```
function cannon_trajectory
```

which calculates the trajectories of the projectile

- for a firing angle $\alpha = 20^\circ$ in the time interval $t \in [0, 5]$,
- for a firing angle $\alpha = 40^\circ$ in the time interval $t \in [0, 8.5]$,

using the standard integrator `ode45` in MATLAB and which plots both solutions in a single figure.

(1.2c) Develop a MATLAB function

```
function x1 = cannon_hit(alpha)
```

which determines the x_1-coordinate of the point of impact of the projectile as a function of the firing angle α, using the MATLAB functions `ode45` and `odeset('Events',...)`.

(1.2d) We wish to determine the firing angle $20^\circ < \alpha^* < 40^\circ$ for which the projectile will hit the above mentioned target. Formulate this problem in terms of a root-finding problem and solve it numerically by implementing a secant method in the MATLAB-function

```
function alpha = cannon_angle.
```

(1.2e) Show that the energy of the projectile is a conserved quantity if the air resistance is negligible.

Problem 1.3 Cat and Mouse

A cat chases after a mouse in the (x, y)-plane. In doing so, the cat invariably runs directly toward the mouse at a constant speed $v_C = 2$. The mouse on its part tries to escape to its mouse hole at velocity $v_M = 1$. The mouse hole is situated at the point $(0, 1)$. Let the mouse be at the point $(0, 0)$ at time $t = 0$ and let the cat be at the point $(1, 0)$.

(1.3a) Determine the IVP (see [NODE, Def. 1.1.2]) describing the trajectory of the mouse.

(1.3b) Determine the IVP describing the trajectory of the cat. What is the extended phase space Ω in this case (see [NODE, Def. 1.1.2])?

(1.3c) Write a MATLAB function

```
function trajectories
```

which calculates the evolution of the trajectories of both animals up to time $T = 0.5$ using the MATLAB function `ode45`. Plot the trajectories in the same figure.

(1.3d) Determine the IVP which describes the trajectory of the displacement vector between the mouse and the cat. Write a MATLAB function

```
function te = rendezvous
```

which calculates whether, and when, the cat and the mouse have approached each other up to 10^{-5} using the MATLAB function `ode45` and `odeset('Events', ...)`.

Published on 24 February 2016.

To be submitted by 1 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 February 22, 2016