Problem Sheet 11

Problem 11.1 Implicit Euler Method for DAE of Order 1

We consider the following DAE

$$\begin{cases} \dot{x}_1 &= x_3\,, \\ \dot{x}_2 &= x_4\,, \\ \dot{x}_3 &= -2x_1x_5\,, \\ \dot{x}_4 &= -2x_2x_5 + 9\,, \\ 0 &= x_3^2 + x_4^2 - 2x_5 + 9x_2\,. \end{cases} \tag{11.1.1}$$

with initial conditions

$$x_1(0) = 1\,, \quad x_2(0) = x_3(0) = x_4(0) = x_5(0) = 0.$$

(11.1a) Check if the initial conditions are consistent with the algebraic equation.([NUMODE, Def. 3.8.7])

(11.1b) Show that, the DAE (11.1.1) has index 1 and state the equivalent ODE explicitly.

(11.1c) Apply the implicit Euler method to (11.1.1) and explicitly state the system of equations.

(11.1d) Complete the MATLAB-Template `firstDAE.m`, in which the system of equations from subproblem (11.1c) is solved using `nNewton` iterations of the Newton method. Apply the function for different numbers of time steps `N` and different values of `nNewton`. Plot the evolution of the algebraic equations. What do you observe?

Problem 11.2 Mechanical Arm

This problem is long and complex. It would be great if you could try your best and finish as much as you can.

A mechanical arm is built up from $n > 1$ arms of length $\ell = 1$ (see Figure 11.1). The movement of the mechanical arm is described by the change in the angle θ_i (with respect to the horizontal direction) of its arms.

(11.2a) Write a function $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^2$, which gives the position of the end of the arm as a function of the angles $\theta_1, \cdots, \theta_n$.

Now let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^\top$. A natural restriction is that the arm moves along a given path $\boldsymbol{\gamma} : [0, T] \to \mathbb{R}^2$. We want to rewrite the problem as a DAE.

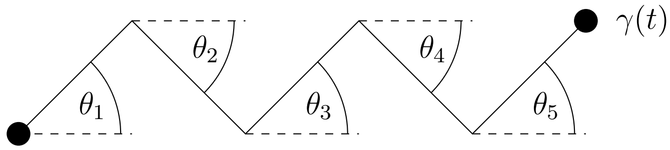

Figure 11.1: Mechanical Arm

(11.2b) The algebraic constraint for this problem is $\mathbf{g}(\boldsymbol{\theta}) = \gamma(t)$. Prove that when $n > 2$, i.e. there are more than 2 arms, the solution $\boldsymbol{\theta}$ to constraint equation cannot be unique. We will take the one with the smallest Euclidean norm $\|\dot{\boldsymbol{\theta}}(t)\|$. Prove that in this case $\dot{\boldsymbol{\theta}} \in \operatorname{Im} \mathrm{D}_{\boldsymbol{\theta}}\mathbf{g}(\boldsymbol{\theta})^T$, so there exists $\lambda \in \mathbb{R}^2$ such that $\dot{\boldsymbol{\theta}} = \mathrm{D}_{\boldsymbol{\theta}}\mathbf{g}(\boldsymbol{\theta})^T\lambda$.

As a result of previous problem, we have the following DAE equivalent to the equation we were discussing

$$\dot{\boldsymbol{\theta}} = \mathbf{d}(\boldsymbol{\theta}, \boldsymbol{\lambda}), \tag{11.2.1}$$

$$0 = c(t, \boldsymbol{\theta}, \boldsymbol{\lambda}), \tag{11.2.2}$$

with

$$c(t, \boldsymbol{\theta}, \boldsymbol{\lambda}) := \mathbf{g}(\boldsymbol{\theta}) - \boldsymbol{\gamma}(t),$$

$$\mathbf{d}(\boldsymbol{\theta}, \boldsymbol{\lambda}) := \mathrm{D}_{\boldsymbol{\theta}}c(t, \boldsymbol{\theta})^\top \boldsymbol{\lambda},$$

and $\boldsymbol{\lambda} \in \mathbb{R}^2$. Since c is independent of $\boldsymbol{\lambda}$, we may just write $c(t, \boldsymbol{\theta})$ instead of $c(t, \boldsymbol{\theta}, \boldsymbol{\lambda})$.

(11.2c) Derive the DAE (11.2.1) for $n = 5$ and

$$\boldsymbol{\gamma}(t) := \begin{pmatrix} 4 + \frac{4}{5}\sin^3(t) \\ \frac{13}{16}\cos(t) - \frac{5}{16}\cos(2t) - \frac{1}{8}\cos(3t) - \frac{1}{16}\cos(4t) - \frac{5}{16} \end{pmatrix}.$$

(11.2d) Write a straightforward MATLAB function `robotsolve1.m` that approximates the solution of the DAE (11.2c) using the MATLAB integrator `ode23t` (with default tolerance) on the interval $[0, 2\pi]$. Use the initial condition

$$\big(\boldsymbol{\theta}(0), \boldsymbol{\lambda}(0)\big)^\top = (0, 0, 0, \frac{\pi}{3}, -\frac{\pi}{3}, 0, 0)^\top.$$

Why (and where) does this approach fail?

(11.2e) Write a MATLAB function `robotimpe.m` that approximates the solution of the DAE (11.2c) using `N`=100 steps of the implicit Euler method on the interval $[0, 2\pi]$. Use the initial conditions

$$\big(\boldsymbol{\theta}(0), \boldsymbol{\lambda}(0)\big)^\top = (0, 0, 0, \frac{\pi}{3}, -\frac{\pi}{3}, 0, 0)^\top.$$

Your MATLAB function should also plot the exact path or the arm $\boldsymbol{\gamma}(t)$ and the approximated path of the arm $\mathbf{g}\big(\theta(t)\big)$ as well as the distance between the two. What do you observe?

HINT: Implicit means that $(\boldsymbol{\theta}_{k+1}, \boldsymbol{\lambda}_{k+1})^\top$ is calculated using one Newton iteration with initial value $(\boldsymbol{\theta}_k, \boldsymbol{\lambda}_k)^\top$.

(11.2f) Transform (11.2.1) into an equivalent DAE of index 1 by differentiating c and $\mathbf{d}$ from (11.2c) . Prove that the transformed DAE has index 1.

(11.2g) Write a MATLAB function `robotsolve2.m` that approximates the solution of the DAE subproblem (11.2f) using the MATLAB-integrator `ode23t` (default tolerance) on the interval $[0, 2\pi]$. Use the initial conditions

$$\left(\boldsymbol{\theta}(0), \boldsymbol{\lambda}(0)\right)^\top = \left(0, 0, 0, \frac{\pi}{3}, -\frac{\pi}{3}, 0, 0\right)^\top .$$

Your MATLAB-script should also plot the exact path or the arm $\boldsymbol{\gamma}(t)$ and the approximated path of the arm $\mathrm{g}\big(\theta(t)\big)$ as well as the distance between the two. What do you observe?

(11.2h) Show that, the DAE (11.2.1) is equivalent to the following ODE

$$\dot{\boldsymbol{\theta}} = \mathrm{D}_{\boldsymbol{\theta}} c(t, \boldsymbol{\theta})^\top \left(\mathrm{D}_{\boldsymbol{\theta}} c(t, \boldsymbol{\theta}) \mathrm{D}_{\boldsymbol{\theta}} c(t, \boldsymbol{\theta})^\top\right)^{-1} \dot{\boldsymbol{\gamma}}(t). \tag{11.2.3}$$

(11.2i) Write a MATLAB function `robotodesolve.m` that approximates the solution of the ODE (11.2.3) using an appropriate MATLAB integrator on the interval $[0, 2\pi]$. Use the initial conditions

$$\left(\boldsymbol{\theta}(0), \boldsymbol{\lambda}(0)\right)^\top = \left(0, 0, 0, \frac{\pi}{3}, -\frac{\pi}{3}, 0, 0\right)^\top .$$

Your MATLAB function should also plot the exact path or the arm $\boldsymbol{\gamma}(t)$ and the approximated path of the arm $\mathrm{g}\big(\theta(t)\big)$ as well as the distance between the two. What do you observe?

Problem 11.3 Robustness of L-Stable Method

In this exercise we investigate the *robustness* of L-stable 1-step methods for the scalar linear model problem [NUMODE, Eq. (3.1.1)]. Robustness denotes the special property of a numerical integrator, namely that the integration error can be reasonably estimated independently of a parameter in the differential equation.

We apply the two step Radau-2-method ($\rightarrow$ [NUMODE, Eq. (3.6.4)]) to the model problem

$$\dot{y} = \lambda y, \qquad y(0) = 1. \tag{11.3.1}$$

Hence we construct an equidistant mesh with step size $h = \frac{1}{N}$ on the time interval $[0, 1]$.

(11.3a) Determine an analytic expression for

$$e_N(\lambda) := |y_N - y(1)|, \tag{11.3.2}$$

by expressing the numerical solution using the stability function ($\rightarrow$ [NUMODE, Thm. 3.1.6]) of the method.

(11.3b) Write a MATLAB function

```
e = sup_e(N),
```

which determines a lower bound for

$$\sup_{0 \le \lambda \le N} e_N(\lambda)$$

by sampling on the interval $[0, N]$.

(11.3c) Try to give a constant C, independent of h and λ, such that

$$|y_N - y(1)| \leq Ch \ . \tag{11.3.3}$$

Does this C exist? Prove it.

(11.3d) State a 1-step method which when applied to (11.3.1) with uniform time step size h has the property (11.3.3) with C independent of h and λ.

Published on 9 May 2016.

To be submitted by 17 May 2016.

References

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

[NUMODE] Lecture Slides for the course "Numerical Methods for Ordinary Differential Equations", SVN revision # 52913.

Last modified on May 9, 2016