AWESCO Winter School on Numerical Optimal Control with DAE - University of Freiburg

Exercise 3: Explicit and Implicit Integrators

Andrea Zanelli, Joel Andersson, Joris Gillis, Sebastien Gros, Moritz Diehl

In this exercise we will implement two different integration schemes and we compare their performance on a non-trivial example that will highlight the advantages and disadvantages of explicit and implicit integration schemes. Consider the following Ordinary Differential Equation (ODE) describing the dynamics of the so called Van der Pol oscillator:

$$\dot{x} = f(x, \mu) := \begin{bmatrix} \mu(x_1 - \frac{1}{3}x_1^3 - x_2) \\ \frac{1}{\mu}x_1 \end{bmatrix} \tag{1}$$

with $x \in \mathbb{R}^2$ being the differential states of the system and $\mu \in \mathbb{R}_{++}$ being a parameter that, as we will see later, plays a role in the stability of the integration scheme used to integrate the ODE.

3.1 Derive the linearization of the system:

$$\dot{x} = A(x_L, \mu)x,$$

where $A(x_L, \mu) := \frac{\partial f}{\partial x}$ and compute the eigenvalues of A. How do the eigenvalues change as a function of x_L and μ? For which values of x_{L} and μ would you expect an integration scheme to potentially encounter problems in integrating the dynamics?

3.2 Using the template provided with the exercise, write a function `van_der_pol.m` that returns an evaluation of the dynamics (1).

3.3 Implement the explicit Euler method $x^+ = x_0 + hf(x_0, u)$ and run a simulation from $t = 0$ to $t = 10$ starting from $\bar{x}_0 = [1,\ 2]^T$, with $h = 0.1$ and compare the obtained solution with the output of the MATLAB integrator for stiff equations `ode15s`.

3.4 Then, implement the explicit Runge-Kutta scheme of order 4 (RK4)

$$\begin{aligned} k_1 &= f(x_0) \\ k_2 &= f(x_0 + \frac{1}{2}hk_1) \\ k_3 &= f(x_0 + \frac{1}{2}hk_2) \\ k_4 &= f(x_0 + hk_3) \\ x^+ &= x_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \end{aligned} \tag{2}$$

where h is the discretization step-size. Run the same simulation as before, this time using your implementation of RK4 , with $\mu = 1$ and $h = 0.1$ and compare the obtained solution with the one obtained with the explicit Euler scheme.

3.4 Code now an implicit scheme instead. For simplicity we will implement the implicit Euler method:

$$x^+ = x_0 + hf(x^+)$$

Define a CasADi function that evaluates the implicit function

$$x^+ - x_0 - hf(x^+) = 0 \tag{3}$$

and its derivatives. You can use the implementation of the full-step Newton method provided with the exercise by calling the function `IEU` to solve equations (3). Run the same simulation as before and compare the accuracy of the three numerical approximations using the plot provided in the template. Which of the schemes is most accurate?

3.5 Compare now the accuracy of the three schemes running a simulation with $\mu = 10$ (keep the other settings unchanged). Which of the two schemes is performing best?

3.6 In order to have a better understanding of how stability and accuracy of the two schemes change as a function of the step-size h, run a series of simulations varying this parameter. Fix $\mu = 1$ and vary h from, store the results in a vector and plot the accuracy in logarithmic scale. Which considerations can you make?

3.7 **Extra:** Derive on paper the stability region for the explicit and implicit Euler scheme. *Hint:* we are interested in the stability of the numerical approximation of the solution to the scalar ODE $\dot{x} = \lambda x$, as a function of $\lambda \in \mathbb{C}$. Write the expression of the numerical approximation x^+ for a single step of the integration scheme and study the stability of the obtained discrete time system as a function of $z := h\lambda$.

3.8 **Extra:** Derive the stability region of the RK4 scheme. *Hint:* for this task you will need to use MATLAB.