Out: February 24, 2022
Due: March 18, 2022

NTNU – Trondheim
Norwegian University of Science and Technology

TTK4130 Modeling and Simulation
Assignment 7

Introduction

The objective of this assignment are:

- To understand how explicit Runge-Kutta (ERK) methods work, and to be able to implement them.
- To understand important concepts of ERK methods, such as accuracy and stability, and the related concepts of local and global error, as well as stability functions.
- To be able to explain how an adaptive integrator works.

Moreover, this is the first assignment where we are going to implement our own differential equation solver. The work done here will be the cornerstone for the solvers developed in assignment 8 and 9. In order to use and implement these solvers, it will be very advantageous to understand how one can convert symbolic expression into function handles or files with `matlabFunction`, and how one can create new function handles using the `@` sign.

Problem 1 (Explicit Runge-Kutta 2 methods)

For a dynamic model:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right) , \tag{1}$$

where for the sake of simplicity we assume the input $\boldsymbol{u} = \boldsymbol{u}_k$ to be constant on the time interval $[t_k,\ t_{k+1}]$, ERK2 methods are based on 2 evaluations of the function $\boldsymbol{f}$, the first of which occurs at $[\boldsymbol{x}\left(t_k\right), \boldsymbol{u}_k]$. They can generally be written as

$$\boldsymbol{k}_1 = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}(t_k), t_k\right) \tag{2a}$$

$$\boldsymbol{k}_2 = \boldsymbol{f}\left(\boldsymbol{x}_k + a \cdot \Delta t \cdot \boldsymbol{k}_1, \boldsymbol{u}(t_k + c\Delta t), t_k + c\Delta t\right) \tag{2b}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t \sum_{i=1}^{2} b_i \cdot \boldsymbol{k}_i. \tag{2c}$$

and have the Butcher tableau

$$\begin{array}{c|cc} 0 & & \\ c & a & \\ \hline & b_1 & b_2 \end{array}$$

(Zeros in the tableau are traditionally omitted)

The ERK2 method will have a numerical local error $e_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}\left(t_{k+1}|k\right)$, where $\boldsymbol{x}(t|k)$ is the actual trajectory of the ODE with the initial condition $\boldsymbol{x}(t_k|k) = \boldsymbol{x}_k$.

(a) For the sake of simplicity, consider the case where $\boldsymbol{f}$ is time-independent (t does not enter as an argument) and $\boldsymbol{u}(t) = \boldsymbol{u}_k$ is constant over the time interval $[t_k,\ t_{k+1}]$.
Provide conditions on $a, b_{1,2}$ and c such that the error e_k of the method is of order 3.
Hint: Observe that (notice the dot on second f - time derivative)

$$\boldsymbol{x}\left(t_{k+1}\right) = \boldsymbol{x}\left(t_k\right) + \Delta t \cdot \boldsymbol{f}\left(\boldsymbol{x}(t_k), \boldsymbol{u}_k\right) + \frac{\Delta t^2}{2} \cdot \dot{\boldsymbol{f}}\left(\boldsymbol{x}(t_k), \boldsymbol{u}_k\right) + \mathcal{O}\left(\Delta t^3\right). \tag{3}$$

(b) ERK2 methods are at best of order 2. How does that relate to $e_k = \mathcal{O}\left(\Delta t^3\right)$?

(c) **(Optional)** Consider a system having a trajectory given by a polynomial of order n, i.e.:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t_k) + \sum_{i=1}^{n} \frac{\alpha_i}{i!} (t - t_k)^i , \quad t \in [t_k, t_{k+1}]. \tag{6}$$

For what values of n, $b_{1,2}$, and c is the ERK2 method exact on (6)?
What do you observe from your computations?

Problem 2 (Accuracy and stability)

We will start by investigating the accuracy and computational cost of some explicit integration schemes. We will consider the explicit Euler scheme and other 2 explicit Runge-Kutta scheme of order 2 and 4, respectively. The Butcher tables of these three schemes are:

Table 1: Explicit Euler (RK1)

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

Table 2: Runge-Kutta 2 (RK2)

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

Table 3: Runge-Kutta 4 (RK4)

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & & \frac{1}{2} & & \\ 1 & & & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

We will use these methods to simulate the classic test system:

$$\dot{x} = \lambda x , \tag{9}$$

where $\lambda < 0$.

(a) Implement the 3 ERK schemes for a generic vector field $\boldsymbol{f}(t, \boldsymbol{x})$ (i.e. $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$).
Test the implemented codes for $\lambda = -2$ in the time interval $[0, 2]$. Choose a time-step $\Delta t = 0.4$ and the initial condition $x(0) = 1$. Comment on the results.
Hint: It is recommended to code a Matlab function that implements an arbitrary ERK method based on its Butcher table. The file `ERKTemplate.m` provides a template for such a function. In addition, you can use the routine `TestERK.m` to test your implementation. This routine should simulate the classical mass-damper-spring system.

(b) Investigate the evolution of the accuracy of the integrators (against the true solution of (9)) as a function of the time-step Δt. Moreover, find the actual order of the methods as a function of the time-step.
Add the corresponding plots to your answer, and compare the obtained results to the theoretical order of accuracy of the various schemes.

(c) For what value of $\lambda < 0$ will the different schemes become unstable?

Problem 3 (Van der Pol oscillator)

Consider the nonlinear dynamics:

$$\dot{x} = y \tag{10a}$$
$$\dot{y} = u \left(1 - x^2\right) y - x , \tag{10b}$$

with $u = 5$ and initial conditions $x(0) = 2$ and $y(0) = 0$.

(a) Simulate the dynamics (10) using the adaptive integrator `ode45` for a final time $t_f = 25$, and with the default solver options. Plot the obtained results and the discrete times selected by `ode45`.
What do you observe? Explain.

(b) Simulate the dynamics (10) with your own RK4 scheme using $\Delta t = 0.1$, and compare the results. What time-step size is necessary so that the results from RK4 match the ones from `ode45`? Compare your discrete time grid to the one of `ode45`.