Caroline Lasser, Ilja Klebanov
2nd Mai 2012

Exercises 1 and 2 are to be handed in on Wednesday, 09.05.2012, before the tutorial.

Exercise 1 (Autonomisation)

An ODE is called autonomous, if the right-hand side f does not depend on t explicitly, i.e. $y'(t) = f(y(t))$. Every non-autonomous ODE can be transformed into an autonomous one by the following trick called **autonomisation**:

$$y'(t) = f(t, y(t)),\ y(0) = y_0 \quad \longrightarrow \quad \tilde{y}'(t) = \begin{pmatrix} f(s(t), y(t)) \\ 1 \end{pmatrix} =: \tilde{f}(\tilde{y}(t)),\ \ \tilde{y}(0) = \begin{pmatrix} y_0 \\ t_0 \end{pmatrix}$$

(a) Explain, how these two ODEs correspond.

(b) A numerical method applied to both ODEs yields the approximations $y_n \approx y(t_n)$ and $\tilde{y}_n \approx \tilde{y}(t_n)$. One usually wants the method to be **invariant to autonomisation**, i.e.

$$\tilde{y}_n = \begin{pmatrix} y_n \\ t_n \end{pmatrix}.$$

Show: An explicit Runge-Kutta method (A, b, c), which is consistent (i.e. $\sum_{i=1}^{s} b_i = 1$), is invariant to autonomisation, if

$$\sum_{j=1}^{i-1} a_{ij} = c_i \qquad \text{for all } i = 1, \ldots, s.$$

Exercise 2 (Explicit Runge-Kutta Methods)

Suppose that an explicit s-stage Runge-Kutta method is applied to the ODE

$$y'(t) = \lambda y(t),\ y(0) = y_0.$$

(a) Show that it has the form

$$y_{n+1} = y_n p(h\lambda),$$

where $p \in \mathbb{P}_s$ is a polynomial of degree $\leq s$.

Hint: Using the principle of induction, show that $hk_i \in \mathbb{P}_i(h\lambda)$ for all $i = 1, \ldots, s$.

(b) Assume the method has order s (i.e. $y(t_n) = y_n \ \Rightarrow\ y(t_{n+1}) - y_{n+1} = O(h^{s+1})$). Prove:

$$y_n = y_0 \left(\sum_{k=0}^{s} \frac{(h\lambda)^k}{k!} \right)^n.$$

Hint: Use (a), the exact solution and the Taylor series $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$.

Exercise 3 (Arbitrary Runge-Kutta methods)
Write a MATLAB function

$$\texttt{function}[\texttt{y},\texttt{t}] = \texttt{RK}(\texttt{A},\texttt{b},\texttt{c},\texttt{f},\texttt{y0},\texttt{t0},\texttt{T},\texttt{h}),$$

which computes all approximations y_n for all time steps $t_n = t_0 + nh$ of an arbitrary Runge-Kutta method with Butcher tableau (A, b, c). f is a function-handle on the right-hand side $f(t, y)$, y_0 is the initial value, t_0 and T are the start and end times and h is the step size. Your function should be able to distinguish explicit and implicit Runge Kutta methods.Use the MATLAB method `fsolve` to solve the implicit equations.

Test your function on the pendulum problem from exercise sheet 2 for different Butcher tableaus.