# Numerical Analysis of Ordinary Differential Equations

# Programming Exercises

Summer Semester 2018

Universität Heidelberg - IWR
Prof. Dr. Guido Kanschat, Dr. Dörte Jando,
Bastian Boll, Pablo Lucero

Exercise Sheet 2

---

There will be no scores given and accumulated over the semester. Nevertheless, we strongly recommend solving the programming exercises. We will answer your questions concerning the programming exercises on Wednesdays in the CIP pool.

**Problem P2.1** (**Lotka-Volterra equations**)
The Lotka-Volterra equations are of the following form:

$$u_1'(t) = u_1(t)(a - bu_2(t))$$
$$u_2'(t) = u_2(t)(cu_1(t) - d)$$

with the initial value $u(0) = (u_{1,0}, u_{2,0})^\top$ and parameters $a, b, c, d \in \mathbb{R}$.

a) What do these equations model? Inform yourself with the help of Wikipedia.

b) Write a new function

```
f = rhs(t,u)
```

that represents the right-hand-side of these equations where $u$ is a vector. The input $u$ and the return value $f$ should be vector-valued. The parameters $a, b, c$ and $d$ should be stored within that function.

c) Use the parameters: $u(0) = (1, 1)^\top$, $(a, b, c, d) = (5, 2, 2, 1)$, $I = [0, 10]$, $n = 100$.

Solve the Lotka-Volterra equations with the explicit Euler method. Make sure that your program has the last timepoint $t_n$ at the interval end.

d) Plot the computed approximation $u_h$ of c).

**Problem P2.2** (**Runge-Kutta Solver**)
Now we want to solve the Lotka-Volterra equations with a different explicit method.

a) Implement the general Runge-Kutta method:

```
runge_kutta(f, t0, y0, h, nsteps, A, b, c)
```

This function gets the same arguments as `expl_euler`. Additionally the method needs to get a matrix `A` and two vectors `b`, `c` which define the Runge-Kutta method. (See the definition of the Butcher tableau in the lecture notes.)

**Remark:** The general form of an explicit Runge-Kutta method is given in the lecture notes.

b) Solve the Lotka-Volterra equations from Exercise P1.1c) with the classical Runge-Kutta method of fourth order and plot the solution.