# 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 3

---

**Problem P3.1** (**Debugging ODE solvers**)
A buggy Runge-Kutta solver implementation is provided alongside this exercise. You can choose to work with either the python or the octave version.
The code should be a realization of the Cash Karp Runge-Kutta method to solve a particular IVP. Cash Karp is a method of order 5.

For this exercise, you should debug this code. Please insert comments to track which changes you have made.

**Problem P3.2** (**Adaptive Refinement**)
In this exercise we want to experience the difference between solving an IVP on an equidistant grid compared to the adaptive refinement algorithm. We will test both implementations on the Lotka-Volterra equations of Exercise P2.1 with the paramters of P2.1c).

a) Implement the Dormand-Prince 45 method

$$y_1 = y_0 + h \sum_i b_i k_i$$
$$\hat{y}_1 = y_0 + h \sum_i \hat{b}_i k_i.$$

The Butcher tableau can be found here:
`https://en.wikipedia.org/wiki/Dormand-Prince_method`

b) Calculate the solution with $h = 10^{-3}$ and plot the solution on the time interval $I$.

c) Implement the adaptive stepsize algorithm (Algorithm 2.4.2). Since Dormand-Prince 45 is an embedded Runge-Kutta method it will provide both $y_1$ and $\hat{y}_1$ for the estimate of the local error: $\|y_1 - \hat{y}_1\|$.

d) Use the tolerance $\varepsilon = 10^{-4}$ and start with $h = 1/10$ to calculate the solution. Plot the solution on $I$.

e) Compare the effort of the adaptive solver to the equidistant solver:
How many Dormand-Prince evaluations are needed?
How many steps needed to be recalculated?
Is the effort twice as large, since you calculate $y$ and $\hat{y}$?
...

Plot the adaptive step size on the intervall $I$. Discuss the results.