The classroom exercises will be discussed on **September 25**, 2023.

Classroom Exercise 4.1 [Runge method]

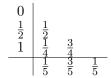
Consider the IVP

$$x'(t) = y(t), \quad y'(t) = 4x(t) - 2ty(t), \quad x(0) = \frac{1}{5}, \ y(0) = \frac{1}{2}.$$

Use one step of the Runge (modified Euler) method to obtain approximations for x(0.1) and y(0.1).

Classroom Exercise 4.2 [Runge-Kutta]

Consider the following Butcher array of an explicit Runge-Kutta scheme:



a) Consider the IVP

$$x'(t) = \lambda x(t), \qquad x(0) = 1$$

and derive an explicit formula for $\Psi^{\tau,0}\mathbf{x}$ using one step of the Runge-Kutta scheme given above.

b) Consider the error $\Psi^{\tau,0}\mathbf{x} - x(\tau)$. How does this error behaves for $\tau \to 0$? What does this mean for the consistency order of the Runge-Kutta scheme?

Classroom Exercise 4.3 [Error estimation]

Consider the convergence theorem (Theorem 2.11) from the lecture for the IVP $x'(t) = -\lambda x(t)$, $x(0) = x_0 \in \mathbb{R}$ on [0, T]. Given a truncation error tolerance $\epsilon > 0$ how should τ_{Δ} be chosen? Determine the (smallest) constants p, C, L for the Euler scheme of the above ODE with T = 1, $\epsilon = 0.1$, $x_0 = 1$, $\lambda = 1$ and an equidistant grid.

Programming Exercise 4.4 [Runge-Kutta methods](4 Points)

Implement an explicit Runge-Kutta solver in MATLAB. Try to implement a function that has an input/output structure similar to the one used by MATLAB's ODE solvers and which contains the Butcher array as an additional input, i.e.,

Your code should also work for systems of ODEs. Apply your function to the following IVPs

1.
$$y' = -5y$$
; $y(0) = 1$; $t \in [0, 3]$,

2.
$$y'' = -y - y'$$
; $y(0) = 1$, $y'(0) = 0$, $t \in [0, 10]$,

and use it with the Butcher arrays for the Euler, Runge and standard Runge-Kutta method.

Compare for each method the different approximate solutions for the equidistant step sizes $\tau \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$ and the exact solutions in one plot.

To be handed in: Executable Matlab code and figures of the programming exercises via ILIAS.