E a) Explain how initial value problems are solved numerically with a Runge-Kutta method, for example using the following Butcher-table

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline & 1/6 & 4/6 & 1/6 \end{array}$$

E b) The Butcher table for an explicit 2-stage Runge-Kutta method is

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

Write out the iterative method for this method. For which choice of parameters do you get the method of Heun and the method of Collatz?

E c) Derive the following multi-step methods using interpolation parabolas.

- 2-stage Adams-Bashforth-method:

$$y_{j+1} = y_j + h\left(\frac{23}{12}f(t_j, y_j) - \frac{16}{12}f(t_{j-1}, y_{j-1}) + \frac{5}{12}f(t_{j-2}, y_{j-2})\right).$$

- 2-stage Adams-Moulton-method:

$$y_{j+1} = y_j + h\left(\frac{5}{12}\,f(t_{j+1}, y_{j+1}) + \frac{8}{12}\,f(t_j, y_j) - \frac{1}{12}\,f(t_{j-1}, y_{j-1})\right).$$

- BDF(2)-method:

$$\frac{1}{h}\left(\frac{3}{2}y_{j+1} - 2y_j + \frac{1}{2}y_{j-1}\right) = f(t_{j+1}, y_{j+1}).$$

H a) Let $y : [0, 4) \to \mathbb{R}$ the solution to the initial value problem

$$y'(t) = 5(t + y(t)^2), \qquad y(3) = -1.$$

(a) **(2 Points)** Compute an approximation y_e to $y(3.1)$ by applying one explicit Euler step.

(b) **(2 Points)** Compute an approximation y_c to $y(3.2)$ (starting with $y(3)$) by applying one Collatz-step. Remember: The iteratiive method to solve a differential equation with right side f with the Collatz-method is

$$y_{j+1} = y_j + h\,f\left(t_j + \frac{h}{2},\; y_j + \frac{h}{2}\,f(t_j, y_j)\right), \qquad h = t_{j+1} - t_j.$$

H b) **(2 Points)** The 3/8-rule (with order 4) is defined by the following Butcher-Table.

$$
\begin{array}{c|cccc}
0 & & & & \\
1/3 & 1/3 & & & \\
2/3 & -1/3 & 1 & & \\
1 & 1 & -1 & 1 & \\
\hline
 & 1/8 & 3/8 & 3/8 & 1/8
\end{array}
$$

Write out the iterative method for this method.

Programming Assignment 1
Consider the initial value problem

$$y'(t) = \cos(t)\, y(t), \qquad y(0) = 1. \qquad (*)$$

Write a Matlab or Python function `solvercompare(h)`, which solves this problem in the interval $[0, 50]$ using (i) the explicit Euler method, (ii) the Collatz method, and (iii) the method of Heun. Plot the results for different step sizes, e. g. $h = 0.1$, $h = 0.2$ and $h = 0.5$. The exact solution to the initial value problem $(*)$ is the periodic function $y(t) = e^{\sin(t)}$.

Programming Assignment 2
Consider two masses m_1, m_2 which are at time t in the positions $\vec{x}_1(t), \vec{x}_2(t)$. According to Newton's law of gravitation the mass m_2 attracts the mass m_1 with the force

$$\vec{F}_1(\vec{x}_1(t), \vec{x}_2(t)) = \frac{\gamma\, m_1\, m_2}{\|\vec{x}_2(t) - \vec{x}_1(t)\|_2^3}(\vec{x}_2(t) - \vec{x}_1(t)),$$

where γ is the gravity constant. The attraction force of m_1 on m_2 is $\vec{F}_2 = -\vec{F}_1$. Let $\vec{v}_k(t) = \vec{x}_k'(t)$ the velocity of the mass k at time t. Then

$$m_k\, \vec{v}_k'(t) = \vec{F}_k(t), \quad k = 1, 2 \qquad \text{(force = mass times acceleration).}$$

The motion of the masses is described by the system of ordinary differential equations

$$\frac{d}{dt}\underbrace{\begin{bmatrix}\vec{x}_1(t)\\ \vec{x}_2(t)\\ \vec{v}_1(t)\\ \vec{v}_2(t)\end{bmatrix}}_{=:\vec{y}(t)} = \underbrace{\begin{bmatrix}\vec{v}_1(t)\\ \vec{v}_2(t)\\ \vec{F}_1(\vec{x}_1(t), \vec{x}_2(t))/m_1\\ \vec{F}_2(\vec{x}_1(t), \vec{x}_2(t))/m_2\end{bmatrix}}_{\vec{f}(\vec{y}(t))} \qquad (*)$$

Write a Matlab or Python function **twomasses**(m1,m2,x1,x2,p,h) which solves the system $(*)$ using a classical Runge-Kutta method and generate an animation showing the motion of the masses. Assume all vectors are 2-dimensional (plane motion). The parameter `h` is the time step size, and the parameter `p` determines the initial velocities:

$$\vec{v}_1(t_0) = \begin{bmatrix}0\\ p/m_1\end{bmatrix}, \quad \vec{v}_2(t_0) = \begin{bmatrix}0\\ -p/m_2\end{bmatrix}$$

Assume the gravitational constant is $\gamma = 1$. Hint: First write a function `y_new=ruku_step(f,y,h)` to compute one Runge-Kutta step with step size h for a differential equation with arbitrary right side f (not explicitly depending on time).