Programming Exercises
Mathematical Modeling

Sheet 4

Due: Wednesday 02.07.2025, 14:00,
Per email at *eric.trebuchon@math.uni-freiburg.de*
Please write your programm in **Octave** or **Python**

Please hand in as pairs of students

Exercise 5: (4+6+6 Points)

The Python program shown on the next page implements the explicit Euler–Collatz method defined by the increment function

$$\Phi(t_k, y_k, \tau) = f\left(t_k + \frac{\tau}{2},\, y_k + \frac{\tau}{2} f(t_k, y_k)\right)$$

for the spring–pendulum equation

$$\ddot{y} + r\dot{y} + D(y - \bar{y}) = 0$$

with initial conditions $y(0) = y_0$ and $\dot{y}(0) = v_0$.

(i) Investigate experimentally how the approximate solutions depend on the parameters r and D.

(ii) Use the exact solution

$$y(t) = \frac{v_0}{\omega} e^{-rt/2} \sin(\omega t), \quad \text{with } \omega = \sqrt{D - \frac{r^2}{4}},$$

of the initial value problem for the special case $r = \frac{1}{10}$, $D = 1$, $y_0 = \bar{y} = 0$, $v_0 = 1$, and determine the approximation error

$$|y_K - y(t_K)|$$

for step sizes $\tau = 2^{-s}$, $\quad s = 1, 2, \ldots, 7$, at time $t_K = 100$.

(iii) Modify the program to implement the explicit and implicit Euler methods as well as Heun's method. Compare the qualitative behavior of the different approximate solutions over the time horizon $T = 1000$.

```python
import numpy as np
import matplotlib.pyplot as plt

def f(t, y):
    r = 1/10; D = 1; ell = 0
    return np.array([y[1], -r * y[1] - D * (y[0] - ell)])

def Phi(t, y, tau):
    return 0.5 * (f(t, y) + f(t + tau, y + tau * f(t, y)))

def federpendel(T=10, y0=0, v0=1, s=5):
    tau = 2 ** (-s)
    K = int(np.floor(T / tau))
    y = np.zeros((K + 1, 2))
    y[0, :] = [y0, v0]

    for k in range(K):
        t_k = k * tau
        y[k + 1, :] = y[k, :] + tau * Phi(t_k, y[k, :], tau)

    plt.plot(np.linspace(0, T, K + 1), y[:, 0], 'r')
    plt.axis([0, T, -5, 5])
    plt.xlabel('Time')
    plt.ylabel('Position')
    plt.title('Euler{Collatz Method for Damped Spring{Pendulum')
    plt.grid(True)
    plt.show()

federpendel()
```