TU Clausthal
Clausthal University of Technology
Institute of Chemical and Electrochemical Process Engineering

1. Exercise Optimization in Engineering Summer Term 2025

(Getting familiar with MATLAB/Python – Build your own Solver)

Please chose your preferred programming language and solve this exercise only in that language.

There are several built-in solvers available in MATLAB and Python. However, to enhance your programming skills, you will create your own numerical solvers for solving Algebraic Equations (AEs) and Ordinary Differential Equations (ODEs) in this exercise.

Task 1: NEWTON'S Method

Use a **self-programmed NEWTON'S** method to solve the following implicit AE system:

$$3x_1 - {x_2}^3 = e^{-x_3}$$
$${x_1}^2 + (x_1 x_3)^3 = e^{-x_2}$$
$$x_2 x_3 - {x_1}^2 = -e^{(x_1 x_2 x_3)}$$

First, implement a **function**, which **returns** a **column vector** with the **residuals** of the AE system. Afterwards, write a **second function** for the **NEWTON'S method**. This function should take **as arguments**, the **AEs**, an **initial guess** (e.g. x = [1,1,1]), a **stopping criteria** (e.g. 1e-6) and a **maximum number of iterations** to avoid an infinite loop. Proof that you found a suitable solution! How many iterations does your solver need? Hint: To calculate the Jacobian, finite differences could be applied.

Task 2: Explicit and Implicit Euler Method

The following initial value problem is given:

$$\dot{y} = -4 * (y - 2) \text{ with } y(t = 0) = 1$$

a) Solve this **ODE** using a self-programmed **explicit Euler** method. Please use a modular program structure as in task 1 (main script and functions). Use different step sizes (0.4, 0.2, 0.05) and plot the solutions in one figure. What are your observations from the plot? Please explain your observations.

b) Solve this **ODE** using a self-programmed **implicit Euler** method (modular program structure!). Again, use different step sizes (0.4, 0.2, 0.05) and plot the solutions in the same figure as in a). Discuss the suitability of these two methods (think about aspects such as computational effort, accuracy, …).

c) Solve this ODE using the MATLAB functions ODE15s and ODE45 (for Python use function "solve_ivp" with methods "RK45" and "BDF"). How many steps are necessary to solve the equation?

Lecturer
Lukas Gottheil, M.Sc.
gottheil@icvt.tu-clausthal.de

Submission per .zip file on Stud.IP
Till 27.04.2025 – 23:59