NTNU – Trondheim
Norwegian University of
Science and Technology

Exercise 4
TTK4130 Modeling and Simulation

Problem 1 (Taylor expansions and order conditions)
The Butcher array for an explicit Runge-Kutta method with two stages is

$$\begin{array}{c|cc} 0 & & \\ c_2 & a_{21} & \\ \hline & b_1 & b_2 \end{array}$$

Assume (for simplicity) that the problem is scalar, that is, the dimension of $\mathbf{y}$ is 1. The stage computations are

$$\begin{aligned} k_1 &= f(y_n, t_n), \\ k_2 &= f(y_n + ha_{21}k_1, t_n + hc_2). \end{aligned}$$

Recall that the Taylor expansion of a function of two variables can be written

$$f(y + \Delta, t + \delta) = f(y, t) + \Delta \frac{\partial f(y, t)}{\partial y} + \delta \frac{\partial f(y, t)}{\partial t} + O(\Delta^2) + O(\delta\Delta) + O(\delta^2).$$

(a) Derive a first order taylor expansion of k_2, assuming that $a_{21} = c_2$, and using that

$$\frac{\mathrm{d}f(y_n, t_n)}{\mathrm{d}t} = \frac{\partial f(y_n, t_n)}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial f(y_n, t_n)}{\partial t} = \frac{\partial f(y_n, t_n)}{\partial y} f(y_n, t_n) + \frac{\partial f(y_n, t_n)}{\partial t}.$$

(b) Derive the conditions on b_1, b_2 and $c_2 = a_{21}$ for the method to be of order 2.

Problem 2 (Implementing solvers for the pneumatic spring)
This problem continues with the pneumatic spring, described by

$$\ddot{x} + g\left[1 - \left(\frac{1}{x}\right)^{\kappa}\right] = 0,$$

where $\kappa = 1.40$ and $g = 9.81$. Defining $y_1 = x$ and $y_2 = \dot{x}$, this is on state-space form

$$\begin{aligned} \dot{y}_1 &= y_2, \\ \dot{y}_2 &= -g\left[1 - \left(\frac{1}{y_1}\right)^{\kappa}\right]. \end{aligned}$$

Recall (from the book or previous exercises) that since there is no damping, the physical solution is that the position oscillates around the equilibrium position $y_1 = 1$.

(a) Implement the explicit Euler method (or an explicit two-stage Runge-Kutta method) in Matlab and simulate from $t = t_0 = 0$ to $t = 10$ s, with step length $h = 0.01$ s. Use initial condition $y_0 = (2, 0)^\top$. Plot the position, and comment.

(b) Implement the implicit Euler method for the same problem. Use fsolve from the optimization toolbox (or implement a Newton-type algorithm yourself) to solve the nonlinear equation. For example: Define the model as

```
f = @(y,t) [ y(2); -g*(1-(1/y(1))^K) ];
```

Then, in each iteration of the for-loop, define the function to be solved ($r(y_{n+1}) = y_n + hf(y_{n+1}, t_{n+1}) - y_{n+1} = 0$), and call fsolve.

```
r = @(ynext) (y(:,i) + h*feval(f, ynext, time(i+1)) - ynext);
y(:,i+1) = fsolve(r, y(:,i), opt);
```

To get this to work, it is important to set small tolerances for the solution:

```
opt = optimset('Display','off','TolFun',1e-8); % Options for fsolve
```

Remark: Using fsolve for this is not particularly efficient. If you want, you can provide the Jacobian of r to fsolve to speed up the solutions (the Jacobian of f was calculated in an earlier exercise). You may also experiment by using the procedure outlined in Chapter 14.8.1 in the book (not required).

(c) Implement the implicit midpoint rule (Gauss order 2) for this problem, $y_{n+1} = y_n + hf((y_n + y_{n+1})/2, t_n + h/2)$.

(d) The energy for the system is

$$E = \frac{1}{\kappa - 1} p_0 A x^{-(\kappa-1)} + mgx + \frac{1}{2} m v^2$$

Plot the energy for all three solutions above. Assume $A = 0.01$ m^2, $m = 200$ kg og $p_0 = 2 \cdot 10^5$ N/m^2.

Problem 3 (Tank and valve-model)

In this problem, we will model a tank that is being emptied through valve. See Figure 5.

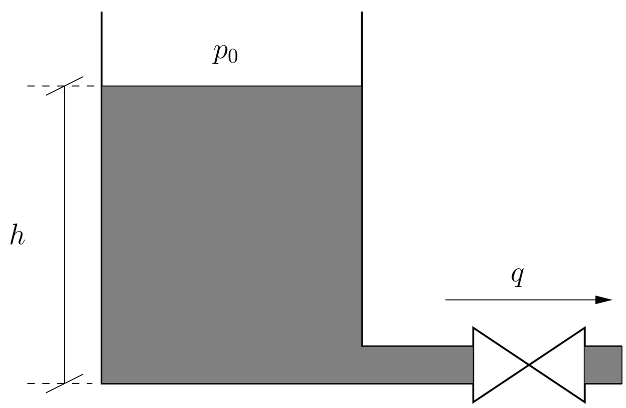

Figure 5: Tank being emptied through a valve

We model the valve with the valve equation

$$q = C_v \sqrt{p - p_0}.$$

We assume constant density $\rho = 1000\text{kg/m}^3$, tank area $A = 4.5\text{m}^2$, gravity of acceleration $g = 10\text{m/s}^2$, valve constant $C_v = 0.15\text{m}^3/(\text{s} \cdot \sqrt{\text{Pa}})$, and outside pressure is athmospheric $p_0 = 10^5\text{Pa}$.

(a) Using a mass balance for the tank, show that the level h is modeled by the differential equation

$$\frac{\mathrm{d}h}{\mathrm{d}t} = -\frac{C_v}{A}\sqrt{\rho g h}.$$

(b) Linearize the system. What happens to the eigenvalue as $h \to 0$?

(c) Implement and simulate the model in Matlab, using the ODE solver ode45. Use initial condition $h(t = 0) = 2$, and simulate for one second. To show the steplengths, use the plot-command `plot(t,h,'o-')`. Comment on the steplengths in view of the answer in (b).