COMPUTER EXERCISE

To start JModelica.org, open a terminal and invoke `VERSION=1.13 jmodelica` This will start an interactive Python shell (which is similar to MATLAB's Command Window). Python scripts are written in plain text in your favorite text editor with the filename extension `.py`. To execute a Python script, invoke
`run <filename>` in the shell.

The focus of this exercise will be the Van der Pol oscillator:

$$\begin{aligned} \dot{x}_1(t) &= \left(1 - x_2^2(t)\right) x_1(t) - x_2(t) + u, \\ \dot{x}_2(t) &= x_1(t). \end{aligned} \tag{1}$$

Problem 1. Simulate (1) with $u \equiv 0$, $x(t_0) \neq 0$ and make a phase plot of the famous limit cycle. Note that if you simulate the system without specifying the input values, the input will be set to 0.

To view the documentation of a method, invoke
`<method name>?`
in the interactive Python shell. Press `q` to exit the documentation view. Take a quick look at the documentation of the simulation method, e.g.
`model.simulate?`

Problem 2.
a) Solve

$$\begin{aligned} \text{minimize} \quad & \int_0^{10} \left(x_1^2(t) + x_2^2(t) + u^2(t)\right) \mathrm{d}t, \\ \text{subject to} \quad & (1), \\ & x_1(0) = 0, \quad x_2(0) = 1. \end{aligned}$$

Make sure that the exit status of IPOPT is
`EXIT: Optimal Solution Found.`

b) The default discretization is 50 elements with 3 collocation points per element. See how coarse the discretization can be made before it significantly affects the accuracy of the solution.

Hint: The number of elements can be changed to for example 30 as follows:

```
op = transfer_optimization_problem(...)
opts = op.optimize_options()
opts["n_e"] = 30
res = op.optimize(options=opts)
```

Invoke `opts?` in the Python shell to see the documentation of all the options.

Remark: If time is scarce and you are more interested in minimum time problems, you can have a look at **Problem 4.** instead of **Problem 3**.

Problem 3.

a) A real controller will seldom realize a control signal that is continuous with respect to time. Solve **Problem 2.** with the additional constraint that u is piecewise constant and updates with a frequency of 1 Hz. How does this affect the optimal cost?

Hint: Piecewise constant controls can be enforced with the optimization option `blocking_factors`. Note that for this problem it is sufficient to supply a list of integers, instead of using the `BlockingFactors` class referenced in the documentation.

b) Use the solution to **Problem 2a)** as an initial and nominal trajectory to solve problem **Problem 3a)**. Confirm that the number of iterations is now fewer (ever so slightly, due to the simplicity of the problem).

Hint: See slides.

Problem 4. Solve

$$
\begin{aligned}
\text{minimize} \quad & t_f, \\
\text{subject to} \quad & (1), \\
& x_1(0) = 0, \quad x_2(0) = 0, \\
& \left|\left| x(t_f) - \begin{bmatrix} 1 & 1 \end{bmatrix}^T \right|\right|_2 \leq 0.1, \\
& -1 \leq u(t) \leq 0, \quad \forall t \in [0, t_f], \\
& x_1(t) \leq 1.2, \quad \forall t \in [0, t_f], \\
& t_f \geq 0.
\end{aligned}
$$

Remark: If time is abundant, either start on the home assignment or enjoy an early lunch.