Exercises for Lecture Course on Numerical Optimization (NUMOPT)
Albert-Ludwigs-Universität Freiburg – Summer Semester 2026

# Exercise 1: General Information and Introduction to CasADi

Léo Simpson, Prof. Dr. Moritz Diehl, with contributions from previous teaching assistants

*The solutions for these exercises will be given and discussed during the exercise session on April the 28th, 2026.*
*To receive feedback on your solutions, please hand it in during the exercise session on April the 28th, 2026, or by e-mail to leo.simpson@imtek.uni-freiburg.de before the same date.*

# I Guide for the coding setup

For the exercises of this course you have to code in Python. Python is free to use. You can follow the following installation guide to install python, jupyter notebook and required packages using conda:

`https://www.syscop.de/files/2025ss/NUMOPT/installation_guide.pdf`

Alternatively, you can directly install python here:

`https://www.python.org/downloads/`

and add the packages required with the following one-line command via terminal:

```
pip install numpy scipy matplotlib jupyter ipython casadi
```

If you don't have an IDE or code editor yet, VS Code with proper extensions can be a good one to start with:

`https://code.visualstudio.com/download`

If you are unfamiliar with Python or NumPy, we provide two tutorials (in the form of Jupyter Notebook):

- `https://www.syscop.de/files/2025ss/NUMOPT/exercise0.ipynb`
- `https://www.syscop.de/files/2022ws/msi/tutorials/PythonTutorial.zip`

For this and future exercises we need to install CasADi. CasADi is an open-source tool for nonlinear optimization and algorithmic differentiation. Further information can be found at:

`https://web.casadi.org`

We will use CasADi's Opti stack because it provides a syntax close to paper notation. For the documentation, look for Opti stack in the CasADi documentation:

`https://web.casadi.org/docs`

*Note*: CasADi is mainly a symbolic framework to formulate optimization problems and generate derivatives. To solve the problems it needs some underlying solvers installed, such as IPOPT, qpOASES, WORHP, KNITRO, ... (some of which are already included).

*Do not hesitate to ask question on the forum if you have some problems with the installation.*

# II Exercises

*For the exercises, you first have to download the template code from the homepage (download the template.zip file, which contains* `ex1_toy_example.py` *and* `ex1_hanging_chain.py`*). Unzip the file and open the code in your favorite IDE.*

## II.1 A tutorial example

In this exercise we will solve a simple unconstrained optimization problem with CasADi. Let's first look at the following unconstrained optimization problem

$$\underset{x \in \mathbb{R}}{\text{minimize}} \quad x^2 - 2x \tag{1}$$

1. Derive first the optimal value for $x$ on paper.
2. Complete the code in the template `ex1_toy_example.py` to solve the problem using CasADi. Is the result the same as the one you obtained on paper?
3. Have a closer look at the template and adapt it to include the inequality constraint $x \geq 1.5$. What is the new result? Is it what you would intuitively expect?
4. Now modify the template to solve the two-dimensional problem:

$$\begin{aligned} \underset{x,\, y \in \mathbb{R}}{\text{minimize}} \quad & x^2 - 2x + y^2 + y \\ \text{subject to} \quad & x \geq 1.5, \\ & x + y \geq 0. \end{aligned} \tag{2}$$

   What are the optimal values for $x$ and $y$ returned by CasADi?

## II.2 Equilibrium position of a hanging chain

We want to model a chain of springs attached to two supports and hanging freely in between. The chain consists of $N$ masses connected by $N-1$ massless springs. Each mass $m_i$ has position $(y_i, z_i)$, $i = 1, \ldots, N$. We would like to find the equilibrium position which minimizes the potential energy of the full system. The potential energy associated with each spring is given by

$$V_{\text{el}}(y_i, y_{i+1}, z_i, z_{i+1}) = \frac{1}{2} D \left( (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 \right),$$

for $i = 1, \ldots, N-1$, and with spring constant $D \in \mathbb{R}^+$. The gravitational potential energy of each mass is

$$V_{\text{g}}(z_i) = m\, g\, z_i,$$

for $i = 1, \ldots, N$, where $g$ is the earth gravity constant, and $m = m_1 = \ldots m_N$. The total potential energy is thus given by

$$V_{\text{chain}}(y, z) = \frac{1}{2} D \sum_{i=1}^{N-1} \left( (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 \right) + mg \sum_{i=1}^{N} z_i,$$

where $y = (y_1, \ldots, y_N)$ and $z = (z_1, \ldots, z_N)$. The minimizing chain position is therefore the solution of the optimization problem

$$\begin{aligned} \underset{y,\, z \in \mathbb{R}^N}{\text{minimize}} \quad & V_{\text{chain}}(y, z) \\ \text{subject to} \quad & y_1 = \bar{y}_1, \\ & y_N = \bar{y}_N, \\ & z_1 = \bar{z}_1, \\ & z_N = \bar{z}_N. \end{aligned} \tag{3}$$

where $(\bar{y}_1, \bar{z}_1)$ and $(\bar{y}_N, \bar{z}_N)$ are the fixed positions of the outer masses.

1. What type of optimization problem is this?

2. Complete the code in the template `ex1_hanging_chain.py` to solve the problem using CasADi, using the following numerical values:

   $N = 40$, $m = 4/N$ kg, $D = \frac{70}{40} N$ N/m, $g = 9.81$ m/s$^2$, the outer masses fixed to $(\bar{y}_1, \bar{z}_1) = (-2, 1)$ and $(\bar{y}_N, \bar{z}_N) = (2, 1)$.

   Solve the problem using CasADi with IPOPT as solver and interpret the results.

3. Introduce ground constraints: $z_i \geq 0.5$ and $z_i - 0.1 y_i \geq 0.5$ for $i = 2, \ldots, N-1$. What type of optimization problem is the resulting problem?

4. Solve the new problem and plot the result. Compare the result with the previous one.