

Exercise 1: General Information, Introduction to CasADi, Convex Optimization

Prof. Dr. Moritz Diehl, Dimitris Kouzoupis, Florian Messerer, Yizhen Wang

Part I: Introduction to CasADi

The aim of this part is to learn how to formulate and solve an optimization problem using CasADi, namely the minimization of the potential energy of a chain of masses connected by springs.

Prepare your laptop

1. **Coding Setup:** For the exercises of this course you can choose between Python or Matlab. Templates will be available for both, and we will also be able to support both. However, the solution videos show only the Matlab version.

MATLAB: Instructions on how to get a free student license from the online software shop can be found here:

```
https://www.rz.uni-freiburg.de/en/services/procurement/software/matlab-license?set\_language=en
```

If you are unfamiliar with MATLAB, here are some useful tutorials:

- <http://www.maths.dundee.ac.uk/ftp/na-reports/MatlabNotes.pdf>
- <http://www.math.mtu.edu/~msgocken/intro/intro.html>

Python: Python is free to use. You can download it from here:

- <https://www.python.org/downloads/>

Make sure that the installation adds Python to your PATH / Environment variables for convenient access from the terminal. After installation, you will also need a few packages for numerical calculation and plotting, these can be installed by running the following one-line command via terminal:

```
pip install numpy scipy matplotlib
```

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, there is also a tutorial available (in the form of Jupyter Notebook):

- <https://www.syscop.de/files/2022ws/msi/tutorials/PythonTutorial.zip>

2. **CasADi:** 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 see <https://web.casadi.org/docs/#document-opti>

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).

To download and install CasADi follow the instructions below:

MATLAB:

- (a) Download the current version for MATLAB from <https://web.casadi.org/get/> and unzip.
- (b) Move the folder called 'casadi-windows-matlabR2016a-v3.5.5' (or similar) to the default MATLAB directory or any directory of your choice.
- (c) Start MATLAB and go to the directory that you chose in Step (b).
- (d) Add the path of CasADi to the MATLAB path, by running:

```
>> addpath('casadi-windows-matlabR2016a-v3.5.5')
```

in the command line of MATLAB (adapt the folder name if necessary).

- (e) Test CasADi by running:

```
>> import casadi.*
>> x = MX.sym('x');
>> disp(jacobian(sin(x), x))
```

Your output should be `cos(x)`.

- (f) To save the path beyond your current session of MATLAB, run

```
>> savepath
```

Python:

- (a) This is again a simple one-line command: `pip install casadi`

Precaution for Conda (Anaconda / Miniconda) users: avoid `conda install casadi`. Regardless of which channel, you will miss necessary solver plugins. Instead, use "pip" inside a conda environment, which you are able to do by running `conda install pip` inside the environment first.

- (b) You can use CasADi in the same way as any other package. Here is a simple example.

```
import casadi as ca
x = ca.MX.sym("x")
print(ca.jacobian(ca.sin(x), x))
```

Your output should be `cos(x)`.

Exercise Tasks

3. **A tutorial example:** Let's first look at the following unconstrained optimization problem

$$\min_{x \in \mathbb{R}} x^2 - 2x \quad (1)$$

- (a) Derive first the optimal value for x on paper. Then, download the code provided for exercise 1 from the course homepage and run `ex1_toy_example.m` in MATLAB to solve the same problem with CasADi. Is the result the same?

$$x^* =$$

- (b) 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?

$$x^* =$$

- (c) Now modify the template to solve the two-dimensional problem:

$$\begin{aligned} \min_{x, y \in \mathbb{R}} \quad & x^2 - 2x + y^2 + y \\ \text{s.t.} \quad & x \geq 1.5, \\ & x + y \geq 0. \end{aligned} \quad (2)$$

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

$$x = \quad y =$$

4. **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) , $i = 1, \dots, N$. 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 $i = 1, \dots, N-1$, 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 $i = 1, \dots, N$, where g is the earth gravity constant, and $m = m_1 = \dots = m_N$. The total potential energy is thus given by

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

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

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

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

- What type of optimization problem is this?
- Formulate the problem using $N = 40$, $m = 4/N$ kg, $D = \frac{70}{40}N$ N/m, $g = 9.81$ m/s² with the outer masses fixed to $(\bar{y}_1, \bar{z}_1) = (-2, 1)$ and $(\bar{y}_N, \bar{z}_N) = (2, 1)$. You can start from the template code `ex1_hanging_chain.m`. Solve the problem using CasADi with IPOPT as solver and interpret the results.
- Introduce ground constraints: $z_i \geq 0.5$ and $z_i - 0.1y_i \geq 0.5$ for $i = 2, \dots, N-1$. What type of optimization problem is the resulting problem? Solve the problem and plot the result. Compare the result with the previous one.

Part II: Convex Optimization

In this part we learn how to recognize convex sets and functions. Moreover we revisit the hanging chain problem from the previous part and add different types of constraints.

Exercise Tasks

5. **Convex sets and functions:** Determine whether the following sets and functions are convex or not.

(a) A wedge, i.e., a set of the form (with $a_1, a_2 \in \mathbb{R}^n, b_1, b_2 \in \mathbb{R}$):

$$\{x \in \mathbb{R}^n \mid a_1^\top x \leq b_1, a_2^\top x \leq b_2\}$$

(b) The set of points closer to a given point than a given set (with $x_0 \in \mathbb{R}^n, \mathcal{S} \subseteq \mathbb{R}^n$):

$$\{x \in \mathbb{R}^n \mid \|x - x_0\|_2 \leq \|x - y\|_2 \forall y \in \mathcal{S}\}$$

(c) The set of points closer to one set than another ($\mathcal{S}, \mathcal{T} \subseteq \mathbb{R}^n$):

$$C := \{x \in \mathbb{R}^n \mid \text{dist}(x, \mathcal{S}) \leq \text{dist}(x, \mathcal{T})\} \text{ with } \text{dist}(x, \mathcal{S}) := \inf_{z \in \mathcal{S}} \|x - z\|_2$$

(d) The function $f(x_1, x_2) = 1/(x_1 x_2)$ on \mathbb{R}_{++}^2 .

(e) The function $f(x_1, x_2) = x_1/x_2$ on \mathbb{R}_{++}^2 .

6. **Minimum of coercive functions:** Prove that the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a continuous, coercive function, has a global minimum point.

Hint: Use the Weierstrass Theorem and the following definition.

Definition (Coercive functions). A continuous function $f(x)$ that is defined on \mathbb{R}^n is coercive if

$$\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$$

or equivalently, if $\forall M \exists R : \|x\| > R \Rightarrow f(x) > M$.

7. **Hanging chain, revisited:** Recall the hanging chain problem from the previous part.

(a) What would happen if you add, instead of the linear ground constraints, the nonlinear ground constraints $z_i \geq -0.2 + 0.1y_i^2$, for $i = 2, \dots, N - 1$ to your problem? Do not use MATLAB yet! What type of optimization problem is the resulting problem? Is it convex?

(b) What would happen if you add instead the nonlinear ground constraints $z_i \geq -y_i^2, i = 2, \dots, N - 1$? Do you expect this optimization problem to be convex?

(c) Now solve both variations using CasADi and plot the results (both chain and constraints). If any of these two optimization problems is non-convex, does it have multiple local minima? If yes, can you confirm that numerically by initializing the solver differently? Note that in CasADi you can provide an initial value x_0 for variable x via

```
opti.set_initial(x, x0)
```