Exercises for Lecture Course on Numerical Optimization (NUMOPT)
Albert-Ludwigs-Universität Freiburg – Winter Term 2019-2020

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

Prof. Dr. Moritz Diehl, Dimitris Kouzoupis and Florian Messerer

Part I: General Information

This course's aim is to give an introduction into numerical methods for the solution of optimization problems in science and engineering. It is intended for students from two faculties, mathematics and physics on the one hand, and engineering and computer science on the other hand. The focus is on continuous nonlinear optimization in finite dimensions, covering both convex and nonconvex problems.

Organization of the course

The course during is based on two pillars, lectures and exercises, accompanied by written material for self-study. As the course is semi-online there will be no lecture held. Instead you can refer to the lectures recorded during the winter term 2015/16. Nonetheless we will meet every Tuesday, 14:00 to 16:00, in HS II (02 033), Albertstraße 23b. Usually every second Tuesday is dedicated to Q&A regarding the lecture. Normally both professor and teaching assistant will attend the Q&A session. Every other Wednesday there will be exercise sessions with the teaching assistant. There is a detailed calendar on the course homepage. Course language is English and all communication is made via the course homepage, where you will also find a link to the lecture recordings:

`https://www.syscop.de/teaching/ws2019/numerical-optimization-online`

This course gives 6 ECTS. It is possible to do a project to get an additional 3 ECTS, i.e., a total of 9 ECTS for course+project.

Exercises: The exercises are partially paper based and partially on the computer. Individual laptops with MATLAB installed are required. Please note that the reserved room is *not* a computer pool. The exercises will be distributed beforehand. You can then prepare yourselves for the exercise session, where you can work on the exercises and get help and feedback from the teaching assistants. We may also discuss solutions of previous sheets if there is demand. Solutions to the exercise sheets have to be handed in via e-mail to *florian.messerer@imtek.de* until the start of the next Q&A session. You will also have to indicate which of the exercises you successfully finished. We will not examine every solution of every student. Note however that we will do extensive random probing. Indicating a task as solved when this is not true will result in 0 points for the whole sheet. Also note the guidelines for handing in which you can find below. You will need at least 40% of the total points in order to pass.

Final evaluation: For engineering students the final grade of the course (6 ECTS) is based solely on a final written exam at the end of the semester. Students from the master in mathematics need to pass the written exam (ungraded) in order take a graded oral exam. The final exam is a closed book exam. Only pencil, paper, a calculator and four single A4 pages of self-chosen formulas are allowed.

Projects: The optional project (3 ECTS) consists in the formulation and implementation of a self-chosen optimization problem or numerical solution method, resulting in documented computer code, a project report, and a public presentation. Project work starts in the last third of the semester. Participants can work either individually or in groups of two people.

Guidelines for handing in exercises

For handing in the exercises via e-mail, please adhere to the following guidelines:

- One (!) file which is your main document (preferably pdf). At the top should be your names and an overview of which tasks you solved. If you have solved a task only partially, you can indicate so. This is then followed by your solutions to the pen-and-paper exercises, and for computer exercises the name(s) of the corresponding file(s). *Claiming tasks as solved when this is not true will result in 0 points for the whole sheet.*
- The main document can be a scan of your handwritten solutions or created with a text editor of your choice (with proper support for mathematical notation, e.g. Latex, MS Word, Open Office...)
- Hand in all of the relevant code files. It should be possible to run them to see all results. It should not be necessary to (un)comment lines for proper functioning. If there are several similar, but conflicting versions (e.g. different constraints), please hand them in as separate files. If you received helper functions as part of the exercise, please also hand them in. This makes it easier to run your files since everything is contained in one folder already. *Do not copy each other's code. This will result in 0 points for the sheet for all participating parties!*

Part II: Introduction to CasADi

The aim of this part is to learn how to use MATLAB and 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. **MATLAB:** The exercises of this course are exclusively done in 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/services-en/beschaffung-em/software-en/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`

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 only a symbolic framework. 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:

   (a) Download the current version (3.5.0) for MATLAB from `https://web.casadi.org/get/` and unzip.

(b) Move the folder called 'casadi-windows-matlabR2016a-v3.5.0' (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 2.

(d) Add the path of CasADi to the MATLAB path, by typing

```
>> addpath('casadi-windows-matlabR2016a-v3.5.0')
```

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
```

Exercise Tasks

3. **A tutorial example**: Lets first look at the following unconstrained optimization problem

$$\min_{x} x^2 - 2x$$

(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?

$x^* = 1$ (1 point)

(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?

$x^* = 1.5$ (1 point)

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

$$\min_{x,y} x^2 - 2x + y^2 + y \tag{1a}$$

$$\text{s.t. } x \geq 1.5 \tag{1b}$$

$$x + y \geq 0 \tag{1c}$$

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

$x = 1.5$ $y = 1.5$ (2 points)

4. **Equilibrium position of a hanging chain:** We want to model a chain attached to two supports and hanging in between. Let us discretize it with N mass points connected by $N-1$ springs. Each mass i has position $(y_i, z_i),\ i = 1, \ldots, N$. The equilibrium point of the system minimizes the potential energy. The potential energy of each spring is:

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

The gravitational potential energy of each mass is:

$$V_{\text{g}}^i = m \, g_0 \, z_i.$$

The total potential energy is thus given by:

$$V_{\text{chain}}(y, z) = \frac{1}{2} \sum_{i=1}^{N-1} D \left((y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 \right) + g_0 \sum_{i=1}^{N} m \, z_i,$$

where $y = [y_1, \cdots, y_N]^\top$ and $z = [z_1, \cdots, z_N]^\top$. We are interested in solving the optimization problem:

$$\begin{aligned} \underset{y,z}{\text{minimize}} \quad & V_{\text{chain}}(y, z) \\ \text{subject to} \quad & (y_1, z_1) \;=\; (-2, 1) \\ & (y_N, z_N) = \;(2, 1) \end{aligned}$$

(a) Formulate the problem using $N = 40,\ m = 4/N$ kg, $D = \frac{70}{40} N$ N/m, $g_0 = 9.81$ m/s^2 with the first and last mass point fixed to $(-2, 1)$ and $(2, 1)$, respectively (you can start from the template code `ex1_hanging_chain.m`). Solve the problem using CasADi with IPOPT as solver and interpret the results.

(4 points)

(b) Introduce ground constraints: $z_i \geq 0.5$ and $z_i - 0.1 \, y_i \geq 0.5$. Solve the resulting Quadratic Program (QP) and plot the result. Compare the result with the previous one.

(2 points)

Part III: 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 adding convex constraints, non-convex constraints and a more realistic chain model.

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:

   $\{x \in \mathbb{R}^n | a_1^\top x \leq b_1, a_2^\top x \leq b_2\}$ (1 point)

   $a_i^\top x \leq b_i$ each defines a halfspace, which is convex. The set is a conjunction of two convex sets (the halfspaces), therefore convex.

   Alternative: show set definition also holds for $z = (1-t)x + ty$ with x, y elements of the set.

   (b) The set of points closer to a given point than a given set:

   $\{x \in \mathbb{R}^n | \, ||x - x_0||_2 \leq ||x - y||_2 \forall y \in \mathcal{S}\}$ (1 point)

   The set can be equivalently written as the intersection

   $$\Omega = \cap_{y \in \mathcal{S}} \{x \in \mathbb{R}^n | \, ||x - x_0||_2 \leq ||x - y||_2\} = \cap_{y \in \mathcal{S}} \Omega_y.$$

   Now each Ω_y defines a halfspace, which is convex. Then Ω is an intersection of convex sets and therefore convex itself.

   To see each Ω_y defines a halfspace:

   $\|x - x_0\|_2^2 \leq \|x - y\|_2^2 \Leftrightarrow (x - x_0)^\top (x - x_0) \leq (x - y)^\top (x - y)$

   $\Leftrightarrow x^\top x - 2x^\top x_0 + x_0^\top x_0 \leq x^\top x - 2x^\top y + y^\top y$

   $\Leftrightarrow \underbrace{(y - x_0)^\top}_{a_y^\top} x \leq \underbrace{\frac{1}{2}\left(\|y\|_2^2 - \|x_0\|_2^2\right)}_{b_y} \Leftrightarrow a_y^\top x \leq b_y$

   (c) The set of points closer to one set than another:

   $C := \{x \in \mathbb{R}^n | \operatorname{dist}(x, \mathcal{S}) \leq \operatorname{dist}(x, \mathcal{T})\}$ with $\operatorname{dist}(x, \mathcal{S}) := \inf\{||x - z||_2 | z \in \mathcal{S}\}$ (1 point)

   Not convex. Counter example: $\mathcal{S} = \{-1, 1\}$, $\mathcal{T} = \{0\} \Rightarrow \mathcal{C} = \{x : \, x \leq -\frac{1}{2} \vee x \geq \frac{1}{2}\}$

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

   Convex if Hessian is positive semidefinite on the domain ($\mathbb{R}^2_{++}$)

   $\nabla^2 f(x_1, x_2) = \frac{1}{x_1 x_2} \begin{bmatrix} \frac{2}{x_1^2} & \frac{1}{x_1 x_2} \\ \frac{1}{x_1 x_2} & \frac{2}{x_2^2} \end{bmatrix}$

   A symmetric matrix A is positive definite iff its **leading** principal minors are > 0, i.e., the determinants of all upper-left quadratic submatrices. Check leading principal minors:

   $\det\left(\nabla^2 f(x_1, x_2)\right) = \frac{1}{x_1 x_2}\left(\frac{4}{x_1^2 x_2^2} - \frac{1}{x_1^2 x_2^2}\right) = \frac{3}{x_1^3 x_2^3} > 0 \; \forall x_1, x_2 \in \mathbb{R}^2_{++}$

   $\det\left(\frac{1}{x_1 x_2} \frac{2}{x_1^2}\right) = \frac{2}{x_1^3 x_2} > 0 \; \forall x_1, x_2 \in \mathbb{R}^2_{++}$

   $\Rightarrow$ Hessian is positive definite.

   Alternative: show $z^\top \nabla^2 f(x_1, x_2) z \geq 0 \; \forall z \in \mathbb{R}^2$, or compute eigenvalues.

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

   A is positive semidefinite $\Leftrightarrow$ **all** its principal minors are ≥ 0.

   $\nabla^2 f(x_1, x_2) = \begin{bmatrix} 0 & -\frac{1}{x_2^2} \\ -\frac{1}{x_2^2} & \frac{2x_1}{x_2^3} \end{bmatrix}$

   $\det\left(\nabla^2 f\right) = 0 - \frac{1}{x_2^4} < 0 \; \forall x_2 \in \mathbb{R}^2_{++} \Rightarrow$ Hessian is not PSD.

   Alternative: show $z^\top \nabla^2 f z \geq 0 \; \forall z \in \mathbb{R}^2$ does not hold, or compute eigenvalues.

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

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

with $f : \mathbb{R}^n \to \mathbb{R}$ 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

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

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

(2 points)

Choose $M = f(0)$. Then from coerciveness we know that $\exists r \geq 0 :\ \|x\| > r \Rightarrow f(x) > f(0)$. Now define set $\Omega := \{x : \|x\| \leq r\}$, which is compact. Then we know from the Weierstrass Theorem that $\exists x^* : f(x^*) \leq f(x)\ \forall x \in \Omega$, i.e., x^* is a minimizer of f on Ω.
More specifically also holds $f(x^*) \leq f(0)$, since $0 \in \Omega\ \forall r \geq 0$. We know that $f(0) < f(x) \forall x :$ $\|x\| > r$, so also $f(x^*) < f(x)\ \forall x \in \mathbb{R}^n \setminus \Omega$.
Therefore $f(x^*)$ is a global minimum of f on $\mathbb{R}^n$, and x^* a global minimizer.

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

   (a) What would happen if you add instead of the piecewise linear ground constraints, the nonlinear ground constraints $z_i \geq -0.2+0.1y_i^2$ to your problem? Do not use MATLAB yet! The resulting problem is no longer a QP, but do you think the problem is still convex?

   (1 point)

   The constraint describes a parabola opened in positive z-direction, where all points above the parabola are part of the set. This is a convex set, therefore the NLP is still convex.
   Alternative: bring into standard NLP form $h(x) \geq 0$. This is a concave function, therefore the NLP is still convex (see lecture notes theorem 3.4).

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

   (1 point)

   The constraint describes a parabola opened in negative z-direction, where all points above the parabola are part of the set. This is not a convex set, therefore the NLP is not convex.

   (c) Check the above results numerically 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 `x0` for variable `x` via

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

   (1 point)

   As demonstrated in code, for the non-convex constraint from (b) different initializations find different solutions (local optima).

8. **Hanging chain, more realistic:** So far, our problem formulation uses the assumption that the springs have a rest length $L_i = 0$ which is not very realistic. A more realistic model includes the rest length L_i in the potential energy of the string in the following way:

$$d_i := \sqrt{(y_i - y_{i+1})^2 + (z_i - z_{i+1})^2} - L_i \tag{2a}$$

$$V_{\text{el}}^i = \frac{1}{2} D\, d_i^2, \quad i = 1, \dots, N-1. \tag{2b}$$

   where $L_i = L/(N-1)$ and L the length of the chain. Note that setting $L = 0$ we obtain the same expression as in Exercise sheet 1. Furthermore, some chain materials (e.g., a string) are characterized by an asymmetric force. They can exhibit tension but buckle under compression. The potential energy of each spring is given in that case by:

$$V_{\text{el}}^i = \frac{1}{2} D \max(0, d_i)^2. \tag{3}$$

   (a) Using Equation (2b) for the potential energy, is the problem still convex? What about Equation (3)? Assume only constraints on the two ends of the chain.

   (1 point)

   For easier analysis set $z_i = z_{i+1}$ and $\Delta y := y_{i+1} - y_i$. Consider this one dimensional version of d_i^2: $d_i^2(\Delta y) = \left(\sqrt{\Delta y^2} - L\right)^2 = (|\Delta y| - L)^2$.
   Substracting L from the absolute value causes negative parts in the center (for $L > 0$). When squared these negative values are responsible for non-convexity. This can be generalized to the original d_i^2. Therefore the objective function is a sum of non-convex functions and also not convex.
   The $\max$ removes the negative parts of d_i that caused the nonconvexity. Therefore the problem with eq. (3) is still convex.

(b) Use Equation (3) and solve the problem with CasADi and IPOPT. For the chain length take $L = 1\,\mathrm{m}$. Initialize y with `y0=linspace(-1,1,N)`.

Hint: Introduce new optimization variables s_i to substitute the terms $\max(0, d_i)$ in the objective and add suitable constraints on the problem. Keep in mind that we are minimizing over the optimization variables and equalities can often be relaxed to inequalities without changing the optimal solution.

Introduce slack variables $s \in \mathbb{R}^N - 1$. Instead of V_{el}^i use $\frac{1}{2} D s_i^2$ in the objective function. Replace the $\max$ with the constraints $s_i \geq 0$ and $s_i \geq d_i$. At a solution one of these constraints well be active since the s_i have to be as small as possible (objective) and are not constrained otherwise. Therefore at a solution we have $s_i = \max(0, d_i)$.

$$\min_{\substack{y, z \in \mathbb{R}^N \\ s \in \mathbb{R}^{N-1}}} \quad \frac{1}{2} \sum_{i=1}^{N-1} D s_i^2 + g_0 \sum_{i=1}^{N} m z_i \tag{4a}$$

$$\text{s.t.} \quad y_1 + 2 = 0, \tag{4b}$$

$$z_1 - 1 = 0, \tag{4c}$$

$$y_N - 2 = 0, \tag{4d}$$

$$z_N - 1 = 0, \tag{4e}$$

$$s_i \geq 0 \qquad i = 1, ..., N-1, \tag{4f}$$

$$s_i - \left(\sqrt{(y_i - y_{i+1})^2 + (z_i - z_{i+1})^2} - L_i \right) \geq 0 \qquad i = 1, ..., N-1 \tag{4g}$$

(2 points)

(c) **Extra:** For the NLP from (b): what happens if you don't initialize any of the variables explicitly? Why?

Hint: By default CasADi initializes all variables at 0.

(1 bonus point)

If all y_i and z_i are initialized at the same value, the d_i are initialized at $\sqrt{0}$ which is not differentiable. CasADi hands a Jacobian with NaNs to IPOPT and IPOPT complains about invalid number.

This sheet gives in total 25 points and 1 bonus point.