Exercises for Lecture Course on Numerical Optimization (NUMOPT)
Albert-Ludwigs-Universität Freiburg – Winter Term 2018-2019

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

(to be completed during exercise sessions on Oct 26 and Nov 2, 2018 or sent by e-mail to messerer@tf.uni-freiburg.de before 2pm, Nov 9, 2018)

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 Friday, 14:00 to 16:00, in SR 226, Hermann-Herder-Str. 10 (Rechenzentrum). Usually every second Friday is dedicated to Q&A regarding the lecture. Normally both professor and teaching assistant will attend the Q&A session. Every other Friday 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/ws2018/numerical-optimization-online`

This course gives 6 ECTS. Students of mathematics can 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 complete the tasks in teams of 2 and show the results to the teaching assistant. Groups that require more time or cannot make it to the exercise session may send their solutions by e-mail (*messerer@tf.uni-freiburg.de*) until the start of the next Q&A session. Note that groups that complete the tasks during the exercise session *do not need to send a report by email*. You will need at least 40% of the total points in order to be eligible for the exam.

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 faculty of mathematics need to pass the written exam (ungraded) in order take an oral exam, which determines their grade of the course part (6 ECTS). If they decide to do a project, their final grade will be a weighted sum of course and project (6+3 ECTS). 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 project (3 ECTS), which is only an option for students of mathematics, consists in the formulation and implementation of a self-chosen optimization problem and 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 and participants can work either individually or in groups of two people.

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 and unzip the toolbox (version 3.4.5) for MATLAB from

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

   (b) Move the folder called 'casadi-windows-matlabR2016a-v3.4.5' (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 2.

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

   ```
   >> addpath('casadi-windows-matlabR2016a-v3.4.5')
   ```

   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
   ```

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 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 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?

   $x =$ $y =$ (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)

   (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)

   (c) The set of points closer to one set than another:
   $C := \{x \in \mathbb{R}^n | \, \text{dist}(x, \mathcal{S}) \leq \text{dist}(x, \mathcal{T})\}$ with $\text{dist}(x, \mathcal{S}) := \inf\{||x - z||_2 | z \in \mathcal{S}\}$

   (1 point)

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

   (2 points)

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

   (2 points)

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)

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)

   (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)

   (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)

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)

   (b) Use Equation (3) and solve the problem with CasADi and IPOPT. For the chain length take $L = 1\,\text{m}$. 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.

   (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)

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