UNIVERSITÄT BERN

# Applied Optimization
# Exercise 3 - Convex Optimization Problems

October 14, 2021

## Hand-in instructions:

Please hand-in **only one** compressed file named after the following convention: `Exercise`$n$`-GroupMemberNames.zip`, where $n$ is the number of the current exercise sheet. This file should contain:

- The complete code folder except for the `build` subfolder. Make sure that your code submission compiles on your machine. Zip the code folder.
- A `readme.txt` file containing a description on how you solved each exercise (use the same numbers and titles) and the encountered problems.
- Other files that are required by your `readme.txt` file. For example, if you mention some screenshot images in `readme.txt`, these images need to be submitted too.
- Submit your solutions to ILIAS before the submission deadline.

## QCQP to SOCP (2 pts)

Convert the following QCQP into SOCP:

$$\begin{aligned} \text{minimize} \quad & x_1^2 + 4x_1x_2 + 4x_2^2 \\ \text{subject to} \quad & 9x_1^2 + 16x_2^2 \leq 25 \\ & x_1 - x_2 = 1 \end{aligned}$$

## Linear Programming

### Transform (2 pts)

For the following optimization problem

$$\begin{aligned} \text{minimize} \quad & ||(2x_1 + 3x_2, -3x_1)^T||_\infty \\ \text{subject to} \quad & |x_1 - 2x_2| \leq 3 \end{aligned}$$

(1) Express the problem as a a linear program. (2) Convert the LP so that all variables are in $\mathbb{R}_+$ and there is no other inequality constraints than ... $\geq 0$.

### Transform general LP to standard form (Bonus 2 pts)

A general linear program has the form

$$\begin{array}{ll} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

where $G \in \mathbb{R}^{m \times n} and A \in \mathbb{R}^{p \times n}$. Transform the general LP to its standard form:

$$\begin{array}{ll} \text{minimize} & p^T x' \\ \text{subject to} & Bx' = e \\ & x' \succeq 0 \end{array}$$

Explain in detail the relation between the feasible sets, the optimal solutions, and the optimal values of the standard form LP and the original LP.

## Programming Exercise: Mass Spring System (6 pts)

Consider a system of $m$ by $n$ springs connecting nodes as illustrated in Figure **??**. The coordinate **x** of a node indexed $a$ at grid point (i, j), where $a \in (0, ..., (m+1) * (n+1) - 1)$, $i \in (0, ..., m)$ and $j \in (0, ..., n)$, is not fixed in $\mathbb{R}^2$. Each edge of the grid has an elastic constant $k$ and each node is subject to an elastic force proportional to the distance between the nodes. Let $\mathbf{x_a}$ and $\mathbf{x_b}$ denote the positions of nodes $a$ and $b$ respectively connected by an edge with elastic constant $k_{a,b}$. Compute the total potential energy stored in the system for a given set of positions under the following modeling assumptions: (1) ideal springs without length and (2) springs with length. Denote with $||\mathbf{x_a} - \mathbf{x_b}||$ the euclidean distance between $\mathbf{x_a}$ and $\mathbf{x_b}$.

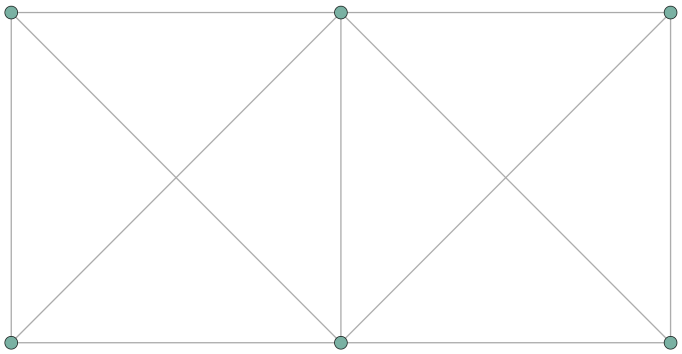

Figure 1: Example: $2 \times 1$ grid

**1. Springs without length** In this case magnitude of the force between this nodes will be given by:

$$F_{a,b} = k_{a,b}||\mathbf{x_a} - \mathbf{x_b}||,$$

and the potential energy between nodes

$$E_{a,b} = \frac{1}{2} k_{a,b}||\mathbf{x_a} - \mathbf{x_b}||^2,$$

Thus the total energy in the system is the sum of the energies of the individual edges $e(a,b)$

$$E = \sum_{e(a,b)} E_{a,b}$$

**2. Springs with length** Now let the rest length of the spring be $l_{i,j}$. The potential energy between nodes is

$$\hat{E}_{a,b} = \frac{1}{2} k_{a,b} (||\mathbf{x_a} - \mathbf{x_b}||^2 - \mathbf{l}_{\mathbf{a,b}}^2)^2,$$

Thus the total energy in the system is the sum of the energies of the individual edges $e(i,j)$

$$\hat{E} = \sum_{e(a,b)} \hat{E}_{a,b}$$

In this exercise, you are requested to evaluate the function value, the gradient and the hessian of the mass spring system at a certain status. The system can be viewed as a combination of elements, with each spring being an element. In `SpringElement2D.hh` and `SpringElement2DWithLength.hh`, with the energy functions $E_{a,b}$ and $\hat{E_{a,b}}$ given above, the functions `eval_f(...)`, `eval_gradient(...)` and `eval_hessian(...)` should be implemented accordingly. Then the local data of each element is assembled into the MassSpringProblem. The total energy is simply the accumulation of every spring's function value. Then local gradient and hessian should be assembled in the global gradient and hessian according to the global indices of the end vertices of every spring. Figure **??** shows how the local gradient vector of the spring marked as red is assemble to the global gradient vector.

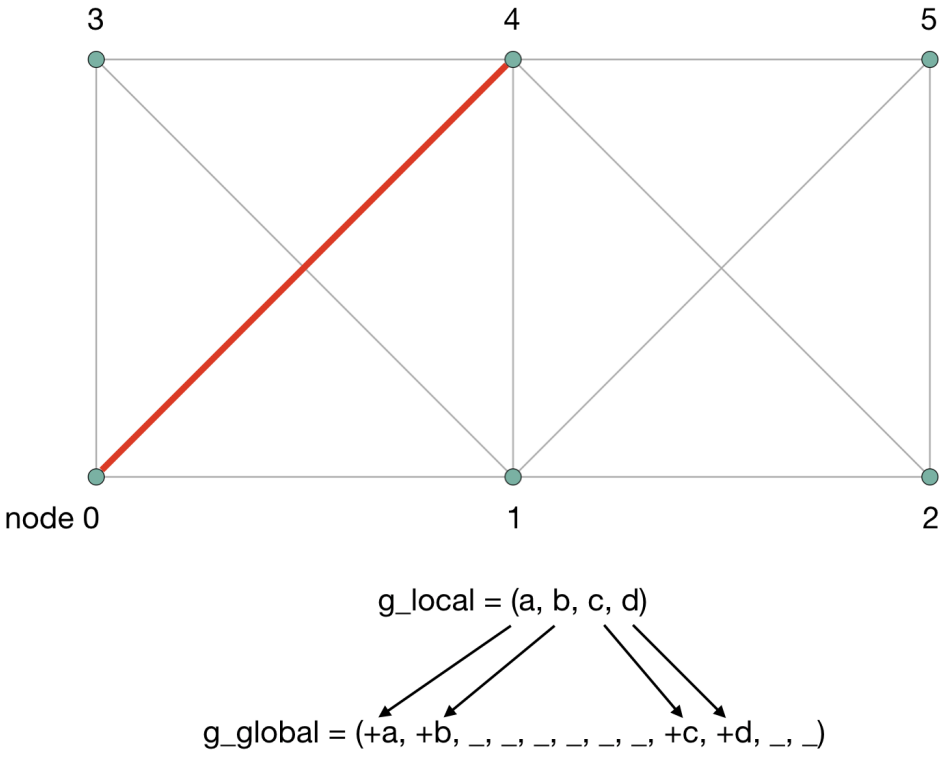


Figure 2: Example: assemble local gradient into the global gradient vector

There are two implementations of the MassSpringProblem, one with dense hessian matrix and the other with sparse hessian matrix. You should implement the corresponding functions in the two files `MassSpringProblem2DDense.hh` and `MassSpringProblem-2DSparse.hh`. Different from setting up the eigen dense matrix, we need to use the triplets to create the sparse matrix. By comparing the performance of these two versions, we will see the advantage of the sparsity.

For implementation, the first step is to implement the `setup_spring_graph()` function in `MassSpringSystemT_impl.hh` to construct the spring graph. You can make use of the data structure provided in `MassSpringSystem/SpringGraph.hh`. Make sure the spring graph is correctly setup. In the executable, you can choose to output the spring graph as two files (.csv) and visualize them on the website. Regarding the elastic constant $k$ and the length $l$, for simplicity all $k$ is set to 1 and the diagonal edge length $l$ is set to $\sqrt{2}$ and the rest is 1.

You can also run a series of unit tests by running the `MassSpringProblemEvaluation-test` executable that is built with the standard executable. Those tests allow you to incrementally verify that the smaller pieces of your implementation work as expected.

*NOTE:* passing all tests does not guarantee the validity of your solution.