C6.2/B2. Continuous Optimization

Numerical Laboratory 2 (optional)

The second numerical lab will take place on Tuesday, March 10th (Week 8), 16.00–18.00 (may finish sooner) in L3; it is entirely optional and there is no hand-in or marks. Please bring your laptop to the lab, with Matlab on it if possible. The lab sessions will be held in Matlab (but you may use other programming languages if you prefer).

The tasks for the second lab are given below. We encourage you to make progress on them before the lab, if possible, so that you can make the most of the lab (and bring your workings with you to the lab); but help and guidance will be provided to start you on the tasks if you have not managed to do so.

The aim of this lab is to explore existing optimization software and apply it to a given constrained optimization problem (the hanging catenary) and time permitting to some standard test functions with box constraints.

(**Algorithms/codes)**

- (**Knitro - a state-of-the-art optimization solver**) Download and install the *student* version of **Knitro** interfaced with Matlab, from

  `https://www.artelys.com/en/optimization-tools/knitro`

  If you prefer other interfaces, do not hesitate to use them.

  Please consult the documentation and find out about the algorithms that it implements. Learn how to run Knitro (see documentation and try for instance, the nonlinear programming examples that come with the package).

- (**Matlab optimization toolbox**) Explore the Matlab optimization toolbox: find out the built-in solvers it offers and the class of optimization problems that they are designed for. Select an appropriate solver(s) from the toolbox for the problems below and experiment with it.

(**Test functions)**

1. (**The hanging catenary problem.**) There are continuous versions of this problem that you may have encountered in modelling courses. Here we formulate this problem as a discrete problem (namely with finitely many continuous variables) and solve it using the Knitro solver or some Matlab solver. I give below a (nonconvex) discrete formulation as it appears in the CUTEr collection (problem **catena**); you are welcome to use this or alternative /modified formulations.

   Let the catenary have $n+1$ beams of length L, with the first beam fixed at the origin and the final beam fixed at a fraction $\gamma \in (0,1)$ of the total length of all the beams. Let (x_i, y_i, z_i), $i = 1, \ldots, n+1$ be the coordinates of the end of the beams and $(x_0, y_0, z_0) = (0,0,0)$ the coordinates of the first beam (fixed at the origin). The resulting problem minimizes the total downward force as the sum of the gravitational force on each beam, assuming it acts at the middle/centre of each beam, subject to the length of beams being preserved; we obtain

   $$\begin{aligned} \min \quad & mg\left(\tfrac{1}{2}y_0 + y_1 + \cdots + y_n + \tfrac{1}{2}y_{n+1}\right) \\ \text{subject to} \quad & (x_0, y_0, z_0) = (0,0,0), \quad x_{n+1} = \gamma(n+1)L, \quad y_{n+1} = 0, \\ & (x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 = L^2, \ \forall i = 0, \ldots, n, \end{aligned}$$

   where the variables are (x_i, y_i, z_i) for all i.

2. For other low-dimensional standard optimization test functions, please see for example, `https://en.wikipedia.org/wiki/Test_functions_for_optimization`.

   You may want to apply your codes to some of these functions, using the (linear) bound constraints given.

Ensure the solver works for your test problem(s) by checking any output messages or error codes. Verify that the minimizer found by the solver is feasible, to within the desired tolerance. What is the optimal objective value/gradient and Lagrange multipliers for the constraints?

Explore different starting points and final accuracy tolerances (if possible; note that most solvers do not require a feasible starting point); changing other allowed algorithm parameters is also acceptable. For the catenary problem, you may want to experiment with increasing number of beams. You can also experiment with different algorithms (in Knitro or Matlab) and possibly different input (provide only function values and gradients, or provide both of those and Hessian values).