Computer Assignment 3: Constrained Nonlinear Optimization

4DM20 Engineering Optimization
Eindhoven University of Technology
Department of Mechanical Engineering
Mauro Salazar and Dinesh Krishnamoorthy
Finn Vehlhaber and Leonardo Pedroso

Objectives

- Practice with formulating a design problem as a mathematical optimization problem,
- Practice with Matlab's `fmincon` or Python's `scypy.optimize` for a three variable constrained minimization problem, in particular regarding: contraint activity, Lagrange multiplier values, conditions for optimality, solution accuracy, user-supplied gradients, and the use of options setting.

Course material

- Chapters 1,3, 5 and 7 of Papalambros and Wilde [1]
- Matlab optimization toolbox [2]
- Scipy documentation: https://docs.scipy.org/doc/scipy/tutorial/optimize.html

Design Problem

Consider the design of a two-bar truss as shown in Figure 1 below, see problem 5.18 from [3]. The load P applied at node B, causes bar 1 (member AB) to be in compression and bar 2 (member CB) to be in tension. The design problem is to minimize the mass of the truss structure subject to two yield stress constraints for bars 1 and 2, and one buckling constraints for bar 1. For design variables we choose $\mathbf{x} = [d_1, d_2, H]^\top$ $(d_1, d_2, H > 0)$, where d_1 is the cross sectional diameter of bar 1, d_2 is the cross sectional diameter of bar 2, and H is the distance between nodes A and C. Length L is kept fixed.

The mass of the structure can be expressed as:

$$m = \rho A_1 L + \rho A_2 \sqrt{L^2 + H^2}, \tag{1}$$

with ρ the material density, and A_i the cross sectional area of bar i ($A_i = \frac{1}{4}\pi d_i^2$). The stress constraints in the two bars can be stated as:

$$\sigma_i = \frac{F_i}{A_i} \leq \frac{\sigma_y}{s}, \qquad i = 1,2 \tag{2}$$

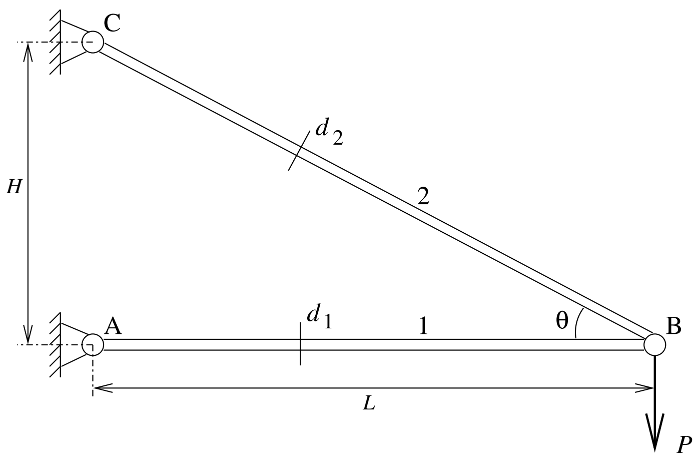

Figure 1: Two-bar truss

with F_i the absolute (tension/compression) force in bar i, σ_y the maximum yield stress, and s a safety factor. The buckling constraint in bar 1 is:

$$F_1 \leq \pi^2 \frac{EI_1}{L^2} \frac{1}{s} \tag{3}$$

where F_1 is the absolute value of the compression force in bar 1, E the elasticity modulus, and I_1 the moment of inertia. For a circular cross-section, we have $I_1 = \frac{1}{64}\pi d_1^4$.

From Figure 1 it can be derived that tension force F_2 (absolute value) is given by:

$$F_2 = \frac{P}{\sin\theta} = \frac{P\sqrt{L^2+H^2}}{H} \tag{4}$$

and compression force F_1 (absolute value) is given by:

$$F_1 = F_2 \cos\theta = \frac{PL}{H} \tag{5}$$

Questions

(a) Show that the optimization problem in negative null form can be stated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = x_1^2 L + x_2^2 \sqrt{L^2 + x_3^2} \\ \text{s.t.} \quad & g_1(\mathbf{x}) = \frac{A}{x_1^2 x_3} - 1 \leq 0 \\ & g_2(\mathbf{x}) = \frac{B\sqrt{L^2+x_3^2}}{x_2^2 x_3} - 1 \leq 0 \\ & g_3(\mathbf{x}) = \frac{C}{x_1^4 x_3} - 1 \leq 0 \\ & \chi : x_1, x_2, x_3 > 0 \end{aligned}$$

and give expressions for A, B, and C in terms of constant parameters σ_y, s, L, ρ, E, and P.

Given our bar material, bar length $L = 12.5$ $[cm]$, external force, and safety factor, we get: $A = 100$ $[cm^3]$, $B = 8$ $[cm^2]$, and $C = 3000$ $[cm^5]$.

(b) Use the `fmincon` algorithm from the Matlab optimization toolbox or `scipy.optimize` from the SciPy toolbox for Python to solve the problem. Set the algorithm options such that the iteration history is printed on screen.

Which minimizer do you obtain? Which constraints are active and what are the corresponding Lagrange multiplier values? (refer to lecture Conditions for Optimality)

(c) Which value for the exit flag is returned? What does this mean? (consult the documentation of `fmincon` or `scipy.optimize`).

(d) Edit the implementation of item c) such that gradients of the objective function and constraint functions are supplied to `fmincon` or `scipy.optimize`.

Follow the instructions in the code files provided.

Run the program, and compare the outcome with item c).

(e) Verify whether the calculated optimum solution satisfies the KKT conditions using the computed Lagrange multiplier value(s) (refer to lecture Conditions for Optimality).

NOTE: you do *not* need to re-calculate the Langrange multiplier values yourself.

(f) Comment on the solution you have obtained; do you trust it and why?

(g) Increase the solution accuracy to 10^{-8} and increase the maximum allowed number of function evaluations.

References

[1] Papalambros, P.Y. and Wilde, D.J., *Principles of optimal design: modeling and computation*, 3rd edition, Cambridge University Press, 2017.

[2] Matlab Optimization Toolbox, The Mathworks.

[3] Belegundu, A.D. and Chandrupatla, T.R., *Optimization concepts and applications in engineering*, Prentice Hall, 1999.