4 Voluntary Matlab problems

Problem 7: Min-Max formulation of multiple load cases

Minimize the maximum compliance of a three load case problem. This problem can be solved by fiddling with the constants in the MMA optimizer.

An optimization problem looking like

$$\left.\begin{aligned} \min_{\boldsymbol{x}} \ &: \max_{k=1,\ldots,p} \{|h_k(\boldsymbol{x})|\} \\ \text{subject to}: &\ g_i(\boldsymbol{x}) \leq 0, \qquad i = 1, \ldots, q \\ : &\ x_j^{min} \leq x_j \leq x_j^{max}, \qquad j = 1, \ldots, n \end{aligned}\right\}, \tag{8}$$

can be re-written to

$$\left.\begin{aligned} \min_{\boldsymbol{x},\, z} \ &: z \\ \text{subject to}: &\ h_i - z \leq 0, \qquad i = 1, \ldots, p \\ : &\ -h_i - z \leq 0, \qquad i = 1, \ldots, p \\ : &\ g_i(\boldsymbol{x}) \leq 0, \qquad i = 1, \ldots, q \\ : &\ x_j^{min} \leq x_j \leq x_j^{max}, \qquad j = 1, \ldots, n \end{aligned}\right\}, \tag{9}$$

which may be solved by MMA using the constants

$$\begin{aligned} m &= 2p + q \\ f_0(\mathbf{x}) &= 0 \\ f_i(\mathbf{x}) &= h_i(\mathbf{x}), \qquad i = 1, \ldots, p \\ f_{p+i}(\mathbf{x}) &= -h_i(\mathbf{x}), \qquad i = 1, \ldots, p \\ f_{2p+i}(\mathbf{x}) &= g_i(\mathbf{x}), \qquad i = 1, \ldots, q \\ a_0 &= 1 \\ a_i &= 1, \qquad i = 1, \ldots, 2p \\ a_{2p+i} &= 0, \qquad i = 1, \ldots, q \\ d_i &= 0, \qquad i = 1, \ldots, m \\ c_i &= 1000, \qquad i = 1, \ldots, m \end{aligned} \tag{10}$$

Problem 8: Robust topology optimization

Implement the robust design formulation Sigmund (2009), Wang et al. (2011) and test it on the force inverter problem from course Problem 5. The 88-line code already has the Heaviside projection filter built-in so the main challenge consists in implementing the min-max formulation (like in course Problem 7) and solving for the three geometry cases in each iteration.

Problem 9: Interior point method

In Problem 4 you have written a program that calls an optimization routine (MMA in our case) to perform one design update. Most off-the-shelf optimization packages operate in a different way. The software user is required to provide a number of callback functions, which evaluate the value and the gradient of an objective function, values and gradients (Jacobian matrix) of the non-linear constraints, if any are present. Further, some solver may utilize second order derivatives