

# Worst-case topology optimization of self-weight loaded structures using semi-definite programming

Erik Holmberg<sup>1,2</sup> · Carl-Johan Thore<sup>1</sup> · Anders Klarbring<sup>1</sup>

Received: 25 March 2015 / Revised: 2 June 2015 / Accepted: 7 June 2015 / Published online: 30 June 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** The paper concerns worst-case compliance optimization by finding the structural topology with minimum compliance for the loading due to the worst possible acceleration of the structure and attached non-structural masses. A main novelty of the paper is that it is shown how this min-max problem can be formulated as a non-linear semi-definite programming (SDP) problem involving a small-size constraint matrix and how this problem is solved numerically. Our SDP formulation is an extension of an eigenvalue problem seen previously in the literature; however, multiple eigenvalues naturally arise which makes the eigenvalue problem non-smooth, whereas the SDP problem presented in this paper provides a computationally tractable problem. Optimized designs, where the uncertain loading is due to acceleration of applied masses and the weight of the structure itself, are shown in two and three dimensions and we show that these designs satisfy optimality conditions that are also presented.

**Keywords** Topology optimization · Semi-definite programming · Worst-case compliance · Self-weight · Robust optimization

## 1 Introduction

Inclusion of acceleration loads due to self-weight are essential when performing topology optimization for parts to be used in the aircraft industry. An aircraft is subjected to accelerations in all directions; ranging from quite moderate in a passenger aircraft to more extreme in a fighter due to advanced maneuvers. We may for example think of applications such as a wing pylon with an external store, e.g. an engine on a passenger aircraft or a weapon or external fuel tank on a fighter aircraft. If we neglect aerodynamic loads, the loads will be a function of the mass of the external store, the self-weight of the pylon, and the accelerations that it is subjected to. Other examples include structural parts inside the fuselage and attachment brackets for electronic devices. Common for these type of applications is that the direction of the loading may vary in space.

### 1.1 Robust optimization

This paper deals with the problem of how to account for such loadings, which may act in any direction, in a topology optimization problem and particularly the difficulty that optimized designs tend to be very sensitive to small perturbations of the applied loads. Traditionally, this issue is dealt with by optimizing under multiple load cases. The selection of appropriate load cases, however, relies on the intuition of engineers, and may result in large-scale problems. Most importantly, this approach cannot guarantee that there does

---

✉ Erik Holmberg  
erik.holmberg@liu.se; erik.holmberg@saabgroup.com

Carl-Johan Thore  
carl-johan.thore@liu.se

Anders Klarbring  
anders.klarbring@liu.se

<sup>1</sup> Division of Solid Mechanics, Department of Management and Engineering, Institute of Technology, Linköping University, SE 581 83, Linköping, Sweden

<sup>2</sup> Present address: Saab AB, SE 581 88, Linköping, Sweden

not exist a load case causing a structural collapse. Thus, it is evident that there is a need for more systematic approaches to obtain optimized structures that are *robust* with respect to variations in the load. Development of such methods is the topic of the field *robust optimization* (Ben-Tal et al. 2009).

There are essentially two different approaches to robust optimization found in the literature: (i) the stochastic and (ii) the deterministic, or worst-case. In the first approach, data, such as applied loads, is assumed to be random but obeying some known or partially known probability distribution. The most common type of stochastic methods in structural optimization is reliability-based design optimization, where the probability of structural failure is quantified using so-called reliability indices (Kharmanda et al. 2004) which determines the formulation of a design optimization problem and is updated in an iterative process. In another class of methods the idea is to convert the stochastic problem into a "standard" multiple load-case problem, the salient point being that the loads and their relative importance are determined by sampling from a continuous probability distribution, (Christiansen et al. 2001; Dunning and Kim 2013).

The deterministic approach to robust optimization deals with worst-case scenarios: structures are optimized to withstand the worst possible loads or manufacturing errors, regardless of the probability of this happening. Worst-case approaches are typically more conservative than stochastic methods, as more restrictions are put on the optimal designs. Further, deterministic methods are not dependent on good statistical data and, most importantly, provide non-probabilistic guarantees, ensuring with certainty that an optimized structure will function as intended, rather than simply that the probability of failure is low. This is clearly desirable in situations where structural failure might cause considerable economical damage or even loss of life. For this reason, deterministic methods are most appropriate for aircraft design.

Ben-Tal and Nemirovski (1997) considered deterministic robust topology optimization of trusses with loads varying in an uncertainty set and showed that the corresponding optimization problem could be formulated as a linear semi-definite program (SDP). This is very nice from a theoretical point of view, since linear SDPs are convex and can be solved with algorithms having proven polynomial-time complexity. However, SDPs with large matrix inequalities are nevertheless beyond reach of current solvers; consequently, numerical examples are generally of small size (Ben-Tal and Nemirovski 1997; Achtziger and Kocvara 2007; Ohsaki et al. 1999).

Other researchers (Brittain et al. 2012; Cherkaev and Cherkaev 2008; de Gournay et al. 2008; Takezawa et al. 2011) have studied formulations for robust topology optimization involving generalized eigenvalues as objectives

or constraints. This way the complexity of SDPs with large matrices can be avoided, but the price to pay is the introduction of non-differentiability in the presence of multiple eigenvalues. Unfortunately, multiple eigenvalues are likely to occur in worst-case compliance problems, so simply ignoring non-differentiability, as is often done, is not satisfying. Therefore, de Gournay et al. (2008) used an SDP approach involving matrices of small size to calculate descent directions, rather than using gradients differentiated directly from the generalized eigenvalue problem.

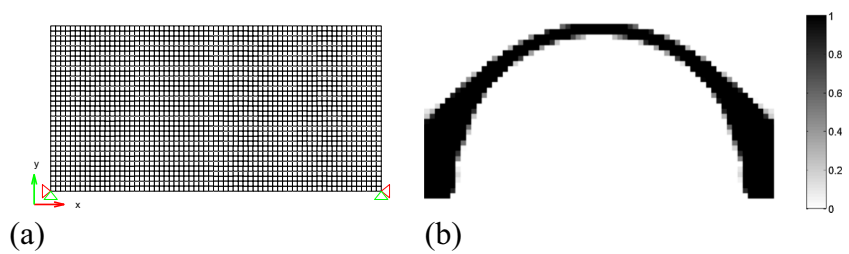
Another approach is to formulate the robust optimization problem as a mathematical program with equilibrium constraints (MPEC) (Kanno 2011). MPECs are, however, difficult to solve numerically, and MPEC-formulations of structural optimization problems will have a large number of non-linear constraints and/or possibly require the evaluation of second derivatives.

This paper focuses on deterministic methods and a main novelty of the paper is that we show how a worst-case problem can be formulated and solved as an SDP with, in contrast to Ben-Tal and Nemirovski (1997), a *small* matrix inequality constraint so that it can be solved efficiently, without introducing further approximations.

## 1.2 Optimization including self-weight

Topology optimization including self-weight was studied by Bruyneel and Duysinx (2005), who noticed that for design variables approaching the lower bound, the ratio between the load and the (SIMP-based) stiffness becomes infinite, leading to unbounded displacements and thus also unbounded compliance. This causes the optimization to converge to a design with intermediate design variable values. Bruyneel and Duysinx (2005) suggested a modified SIMP-model, which was linear below a threshold value, and the same modification was also used in Lee et al. (2012). Whereas they report that no problems were experienced due to the introduced non-differentiability, we prefer a differentiable penalization function. Therefore, we suggest a linear scaling, see (5) in Section 2.2, of the mass matrix when the force vector is created, such that no loads are applied in voids, allowing the optimization to converge with values equal to the lower bound. To the authors' knowledge, this is a new contribution to topology optimization of self-weight loaded structures and the result of this linear scaling is shown numerically in Fig. 1, where the problem from Bruyneel and Duysinx (2005) and Lee et al. (2012) is repeated using our (differentiable) scaling of the mass matrix. As in Lee et al. (2012) the design space, Fig. 1a, is meshed with 2048 elements (we use 4-node quadrilaterals in this example) and the final structure shall occupy 20 % of the design space and support its own

**Fig. 1** Re-calculation of the arch structure example from Bruyneel and Duysinx (2005). **a:** Mesh and boundary conditions, **b:** Black-and-white optimized design obtained for standard gravitational load



weight, subjected to gravitational acceleration. The minimum compliance design, here obtained using MMA (Svanberg 1987), is shown in Fig. 1b; as can be seen we obtain a black-and-white design without the "erratic intermediate density patterns" reported in Bruyneel and Duysinx (2005) and Lee et al. (2012).

Following this introduction, Section 2 presents the structural model, describes the loading and derives the semi-definite optimization problem. In Section 3 we derive the optimality conditions and the numerical treatment of the SDP-problem is discussed in Section 4, where we also show how the displacements can be obtained and present a method to obtain black-and-white designs. Gradients are derived in Section 5, examples are shown in Section 6 and conclusions are finally drawn in Section 7.

## 2 Problem statement

### 2.1 Model

We use the standard framework for structural optimization of linear elastic finite element discretized structures; i.e., there is a state problem of the type

$$\mathbf{K}(\mathbf{x}) \mathbf{u} = \mathbf{F}(\mathbf{x}, \mathbf{r}), \tag{1}$$

where  $\mathbf{K}(\mathbf{x}) \in \mathbb{S}^n$  (the space of symmetric  $n \times n$  matrices with real entries),  $n$  being the number of degrees of freedom, is the stiffness matrix;  $\mathbf{u} \in \mathbb{R}^n$  is the nodal displacement vector; and  $\mathbf{F}(\mathbf{x}, \mathbf{r}) \in \mathbb{R}^n$  is the nodal force vector. Both  $\mathbf{K}(\mathbf{x})$  and  $\mathbf{F}(\mathbf{x}, \mathbf{r})$  are taken to depend on a design variable vector  $\mathbf{x} = (x_1, \dots, x_m)^\top$ . For notational simplicity, we write the theory as if all elements belong to the design domain; thus, there is one design variable for each finite element and the structure consists of  $m$  such elements. The nodal force vector also depends on an uncertainty vector  $\mathbf{r}$ , to be described in Section 2.2.

To avoid mesh dependency and checkerboard patterns (Sigmund and Petersson 1998) we use a filter to define a set of filtered variables  $\boldsymbol{\rho} = (\rho_1, \dots, \rho_m)^\top$ , which are used to calculate the stiffness and mass of the structure. The filtered variables are therefore denoted *physical variables*. There are different types of filters (Sigmund 2007) but a linear local

averaging *design variable filter*, suggested by Bruns and Tortorelli (2001), can be written

$$\rho_i(\mathbf{x}) = \sum_{j=1}^m \Psi_{ij} x_j, \quad i = 1, \dots, m. \tag{2}$$

Here

$$\Psi_{ij} = \frac{\psi_{ij} v_j}{\sum_{k=1}^m \psi_{ik} v_k}, \quad \psi_{ij} = \max\left(0, 1 - \frac{\|\mathbf{e}_i - \mathbf{e}_j\|}{R}\right),$$

where  $v_j$  is the volume of element  $j$ ,  $\mathbf{e}_i$  denotes the position vector of the geometric center of element  $i$ ,  $R$  is the filter radius and  $\|\cdot\|$  is the Euclidean norm.

The design variables are restricted to the compact and convex set

$$\mathcal{H} = \left\{ \mathbf{x} \in \mathbb{R}^m \mid \varepsilon \leq x_i \leq 1, \sum_{i=1}^m d v_i \rho_i(\mathbf{x}) = \bar{M} \right\},$$

in which  $d$  is the constant density and  $\bar{M}$  is the available mass. The lower bound  $\varepsilon > 0$ , is introduced to maintain positive definiteness of the stiffness matrix (Appendix B provides a formulation allowing for a singular stiffness matrix). The stiffness matrix is in a SIMP formulation (Bendsøe 1989) written as follows:

$$\mathbf{K}(\mathbf{x}) = \sum_{i=1}^m (\rho_i(\mathbf{x}))^p \mathbf{K}_i, \tag{3}$$

where  $\mathbf{K}_i$  is an expanded element stiffness matrix and  $p > 1$  is a penalty exponent.

### 2.2 Uncertain loading

Due to the intended application to aircraft design, we include mass acceleration forces in the force vector  $\mathbf{F}(\mathbf{x}, \mathbf{r})$ . In the aircraft industry, the acceleration is usually expressed as multiples of the magnitude of the acceleration of gravity  $g$ , and the maximum acceleration factors,  $n_x$ ,  $n_y$  (and  $n_z$ ), in the  $x$ -,  $y$ - (and  $z$ -)directions, respectively, are generally of different magnitude. Thus, for example, in 2D we may have an elliptical loading region expressed by different accelerations  $g n_x$  and  $g n_y$ , as visualized in Fig. 2.

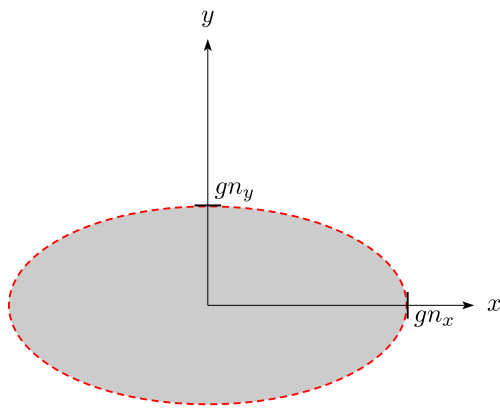


Fig. 2 Visualization of admissible accelerations in 2D

The maximum acceleration factors are placed on the diagonal of a diagonal matrix  $N \in \mathbb{S}^s$ , where  $s$  denotes the number of spatial dimensions. The mass acceleration forces can then be written as

$$F(x, r) = B(x)^T r = \mathbf{A} \sum_{i=1}^m g[\rho_i(x)] M_i + M_i^{\text{ext}}] G_i N r, \tag{4}$$

where  $\mathbf{A}$  is an assembly operator;  $M_i \in \mathbb{S}^{sh_i}$ , where  $h_i$  is the number of element nodes, is the mass matrix for element  $i$ ;  $M_i^{\text{ext}} \in \mathbb{S}^{sh_i}$  is a matrix representing external, non-structural, masses; and  $G_i = (I \dots I)^T \in \mathbb{R}^{sh_i \times s}$ , where  $I \in \mathbb{S}^s$  is the identity matrix. The linear scaling

$$q(\rho_i(x)) = \frac{\rho_i(x) - \varepsilon}{1 - \varepsilon}, \tag{5}$$

of the mass matrix is, as discussed in the introduction, intended to bound the ratio between stiffness and force; without introducing non-differentiable functions, such as that in Bruyneel and Duysinx (2005). By subtracting the lower bound  $\varepsilon$  in the nominator, we make sure that there are no applied loads, due to self-weight, in voids, and  $q(\rho_i)$  becomes 1 when  $\rho_i = 1$ .

Finally, uncertainty is accounted for by letting  $r$  vary in the unit ball  $T = \{r \in \mathbb{R}^s \mid \|r\| \leq 1\}$ . This implies that the (nodal) loads vary synchronously, which is the case under acceleration loads.

The extension to multiple load cases, with one  $r$  for each load case, is straightforward and an extension to handle both fixed and uncertain loads simultaneously is possible, Thore et al. (2015) (de Gournay et al. (2008) also treats this type of loading). Further, the methods in this paper also apply to variations of external nodal loads if  $B$ , instead of being formulated as in (4), is a constant matrix with the maximum nodal loads in the  $x$ -,  $y$ - and  $z$ -directions.

### 2.3 Calculation of the worst-case compliance

The compliance can be written

$$C(x, r) = \frac{1}{2} F(x, r)^T u,$$

where  $F(x, r) = B(x)^T r$  and, from (1),  $u(x, r) = K(x)^{-1} F(x, r)$ , assuming  $K(x)$  is non-singular (see Appendix B for a derivation where  $K(x)$  may be singular). The worst-case compliance is then

$$\begin{aligned} P(x) &= \max_{r \in T} C(x, r) = \max_{r \in T} \frac{1}{2} F(x, r)^T u(x, r) \\ &= \max_{r \in T} \frac{1}{2} r^T H(x) r, \end{aligned}$$

where the notation

$$H(x) = B(x) K(x)^{-1} B(x)^T \tag{6}$$

was used in the last equality. Since  $H(x) \in \mathbb{S}^s$  is positive semi-definite, the last problem is one of maximizing an upper semi-continuous, convex function over a convex, compact set. This implies that the maximum value is taken at extreme points of the feasible set  $T$  (Rockafeller 1972, Corollary 32.3.1). Since the extreme points of  $T$  are  $\{r \in \mathbb{R}^s \mid \|r\| = 1\}$  (Hiriart-Urruty and Lemarchal 1993, p. 110) we have

$$P(x) = \max_{\|r\|=1} \frac{1}{2} r^T H(x) r. \tag{7}$$

Using the Rayleigh-Ritz theorem (Horn and Johnson 1985, Theorem 4.2.2) we see that the optimal value equals the maximum eigenvalue of  $\frac{1}{2} H(x)$ . Thus, (7) may be written

$$P(x) = \frac{1}{2} \lambda_1(H(x)), \tag{8}$$

where  $\lambda_1(H(x))$  is the largest eigenvalue of  $H(x)$ .

### 2.4 Optimization problem

We are now interested in the problem

$$\min_{x \in \mathcal{H}} P(x). \tag{9}$$

A main drawback of this formulation is that if  $\lambda_1(H(x))$  is a multiple eigenvalue, only directional derivatives exist, resulting in a non-smooth problem. Directional derivatives may be obtained following Seyranian et al. (1994), Pedersen and Nielsen (2003) or Overton (1992, Theorem 4). In Pedersen and Nielsen (2003), the directional derivative is used in place of partial derivatives and in Overton (1992) and Seyranian et al. (1994) specialized algorithms are used. A further elevation of this difficulty is that for many problems the optima is likely to occur at a point of multiple eigenvalues. The analysis of Pataki (1998) indicates that

given a sufficiently large feasible set  $\mathcal{H}$  this is the case. This is also shown numerically in Fig. 6.

In order to find a computationally tractable problem that is equivalent to (9), we use a reformulation. First, the min-max problem (9) is rephrased in a so-called bound formulation. Recalling (8), this leads to the optimization problem

$$\begin{aligned} \min_{z \in \mathbb{R}, \mathbf{x} \in \mathcal{H}} \quad & z \\ \text{subject to} \quad & \lambda_1(\mathbf{H}(\mathbf{x})) \leq z. \end{aligned}$$

Then, it can be shown, Appendix A, that this problem is equivalent to the following semi-definite program:

$$\begin{aligned} \min_{z \in \mathbb{R}, \mathbf{x} \in \mathcal{H}} \quad & z \\ \text{subject to} \quad & z\mathbf{I} - \mathbf{H}(\mathbf{x}) \succeq \mathbf{0}, \end{aligned} \tag{10}$$

where “ $\succeq \mathbf{0}$ ” means positive semi-definite. This problem has a small matrix inequality constraint of size  $s \times s$ , and is the non-linear SDP-problem that we treat numerically.

### 3 Optimality conditions

Let  $\mathbf{y} = (\mathbf{x}, z)$  and introduce the Lagrangian

$$\begin{aligned} L(\mathbf{y}; \mu, \boldsymbol{\gamma}, \mathbf{Z}) = & z + \mu(\mathbf{c}^\top \mathbf{x} - \bar{M}) \\ & + \sum_{i=1}^m (\gamma_i(x_i - 1) + \gamma_{i+m}(\varepsilon - x_i)) \\ & + \text{tr}([\mathbf{H}(\mathbf{x}) - z\mathbf{I}]\mathbf{Z}), \end{aligned}$$

where  $\mathbf{Z} \in \mathbb{S}^s$ ,  $\mu$  and  $\boldsymbol{\gamma}$  are multipliers,  $\text{tr}(\mathbf{A})$  denotes the trace of  $\mathbf{A}$  and  $\mathbf{c}$  defines the left hand side of the mass constraint in  $\mathcal{H}$ . The first-order necessary optimality conditions for (10) are then obtained from Bonnans and Shapiro (2000, Theorem 3.9), and reads: If  $\mathbf{y}$  is an optimal solution to (10), then there exists  $\mu, \boldsymbol{\gamma}, \mathbf{Z}$  such that

$$\nabla L(\mathbf{y}; \mu, \boldsymbol{\gamma}, \mathbf{Z}) = \mathbf{0}, \tag{11a}$$

$$\text{tr}([\mathbf{H}(\mathbf{x}) - z\mathbf{I}]\mathbf{Z}) = 0, \tag{11b}$$

$$\gamma_i(x_i - 1) = 0, \quad i = 1, \dots, m, \tag{11c}$$

$$\gamma_{i+m}(\varepsilon - x_i) = 0, \quad i = 1, \dots, m, \tag{11d}$$

$$\gamma_i \geq 0, \quad i = 1, \dots, 2m, \tag{11e}$$

$$\mathbf{Z} \succeq \mathbf{0}. \tag{11f}$$

The necessity of the optimality conditions hinges on some constraint qualification being satisfied; here we use the Robinson’s constraint qualification (Bonnans and Shapiro 2000, p. 67). We now introduce the notations  $\nabla h(\mathbf{y})$  and  $\nabla g_i(\mathbf{y})$  for derivatives with respect to  $\mathbf{y}$  of, respectively, the mass constraint and the  $i$ :th box constraint in  $\mathcal{H}$ . Then, utilizing Corollary 2.101 in Bonnans and Shapiro (2000) we

find that Robinson’s constraint qualification holds for (10) if and only if there exists  $\mathbf{w} \in \mathbb{R}^{m+1}$  such that

$$\begin{aligned} \nabla h(\mathbf{x})^\top \mathbf{w} &= 0 \\ \sum_{k=1}^{m+1} w_k \frac{\partial [z\mathbf{I} - \mathbf{H}(\mathbf{x})]}{\partial y_k} &< \mathbf{0} \quad \text{if } z\mathbf{I} - \mathbf{H}(\mathbf{x}) = \mathbf{0} \\ \nabla g_i(\mathbf{x})^\top \mathbf{w} &< 0, \quad \forall i \in I(\mathbf{x}), \end{aligned} \tag{12}$$

where “ $< \mathbf{0}$ ” means negative definite,  $I(\mathbf{x}) \subset \{1, \dots, 2m\}$  denotes the index set of active inequalities at  $\mathbf{x}$  and the condition  $z\mathbf{I} - \mathbf{H}(\mathbf{x}) = \mathbf{0}$  means that the matrix inequality constraint is active.

In order to verify that a solution  $\mathbf{y}$  satisfies the first-order optimality conditions, we need to find a  $\mathbf{w}$  that satisfies (12), and if such a  $\mathbf{w}$  exist we proceed to find a  $\mathbf{Z}$  that satisfies (11a), (11b) and (11f). We may for example solve for  $\mathbf{Z}$  in the overdetermined linear system obtained from (11a) and (11b). Finally, we check that (11c) – (11f) are satisfied within some numerical tolerance.

### 4 Numerical treatment

#### 4.1 Implementation

The algorithms presented in this paper are implemented and solved in a Matlab based FE- and optimization program. The models are first pre-processed in a commercial program (Altair Engineering 2014) where geometry, mesh and boundary conditions etc. are defined and written to an indata file. The Matlab program then reads the indata file and solves the FE- and optimization problem and post-processes the result.

The semi-definite programming problem is treated using the open-source code `fminsdp` (Thore 2013) which reformulates (10) into a non-linear optimization problem (NLP) that is solved using the interior-point-solver IPOPT (Wächter and Biegler 2006).

#### 4.2 Evaluating the matrix constraint

To avoid explicitly forming  $\mathbf{K}(\mathbf{x})^{-1}$ , the matrix constraint function in (10) is evaluated by solving

$$\mathbf{K}(\mathbf{x}) \mathbf{R} = \mathbf{B}(\mathbf{x})^\top, \tag{13}$$

where  $\mathbf{R} = \mathbf{R}(\mathbf{x}) \in \mathbb{R}^{n \times s}$  is solved for at the same cost as solving (1) for  $s$  load cases. Given  $\mathbf{R}(\mathbf{x})$  then,  $\mathbf{H}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) \mathbf{R}(\mathbf{x})$ .

### 4.3 Methods to solve the semi-definite program

The matrix inequality constraint in (10) is of small size ( $s \times s$ ) but non-linear in  $\mathbf{x}$ . Therefore, specialized methods must be used. Examples include the following:

- The Cholesky factorization method, see Ertel et al. (2008) and Thore (2013). (A “nested” Cholesky method is suggested in Burer et al. (2002))
- The LDL-factorization method due to Fletcher (1985); see also Bogani et al. (2009), Vanderbei and Benson (2000), and Benson and Vanderbei (2003).
- PENNON or PENLAB, see Kočvara and Stingl (2003).
- The “feasible direction method” suggested by Aroztegui et al. (2014).

The Cholesky method has the advantage that it immediately leads to a standard, smooth, NLP formulation. The same is almost true for the LDL-factorization method, but extra care is required to make sure that certain nonlinear inequalities are not violated in the solution process, making it more difficult to use with standard NLP-solvers. PENNON (and its open-source version PENLAB), which implements an augmented Lagrangian-type method, has the drawback that it requires computation of the Hessian of the Lagrangian, something which is prohibitively costly for large-scale nested problems. The benchmarks used to demonstrate the feasible direction method in Aroztegui et al. (2014) are too small to be able to tell whether the algorithm will be efficient in practice, but the implementation described used a dense quasi-Newton approximation for the Hessian, and this is not feasible for large-scale problems.

In `fminsdp` the Cholesky factorization method is used to handle the matrix inequality constraint. The Cholesky factorization method is based on the fact that a symmetric matrix  $\mathbf{A} \succeq \mathbf{0}$  if and only if  $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ , where  $\mathbf{L}$  is a Cholesky factor of  $\mathbf{A}$ ; see Thore (2013) for further details. Therefore, to obtain solutions to (10) we solve the following “standard” NLP:

$$\begin{aligned} \min_{z \in \mathbb{R}, \mathbf{x} \in \mathcal{H}, \mathbf{L} \in \mathcal{L}_+^s} \quad & z \\ \text{subject to} \quad & z\mathbf{I} - \mathbf{H}(\mathbf{x}) = \mathbf{L}\mathbf{L}^\top, \end{aligned} \quad (14)$$

where  $\mathcal{L}_+^s$  denotes the set of lower triangular matrices with non-negative diagonal entries and, at a feasible point,  $\mathbf{L}$  is a Cholesky factor of  $z\mathbf{I} - \mathbf{H}(\mathbf{x})$ .

### 4.4 Calculating the displacements

The nodal displacements are not obtained explicitly when solving problem (10), but are often required or convenient to have for plotting or calculating compliance etc. Obviously, since the load (4) depends on  $\mathbf{r} \in T$ , there is no unique displacement associated to an optimal design  $\mathbf{x}$ . However, it

may be of interest to calculate the displacement  $\mathbf{u} \in \mathbb{R}^n$  that is associated to the worst-case compliance. To that end we note that (7) shows that the  $\mathbf{r} \in T$  that achieves the worst-case compliance is the unit length eigenvector of  $\mathbf{H}(\mathbf{x})$  that is associated with the maximum eigenvalue, if it is unique. Once this  $\mathbf{r} \in T$  is established the corresponding  $\mathbf{u}$  can be calculated by combining (1) and (4) into

$$\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{B}(\mathbf{x})^\top \mathbf{r} \Leftrightarrow \mathbf{u} = \mathbf{R}(\mathbf{x})\mathbf{r},$$

where the last expression follows from (13).

If the maximum eigenvalue of  $\mathbf{H}(\mathbf{x})$  is not unique we may use any linear combination of the eigenvectors corresponding to the multiple eigenvalues and obtain one (of several possible)  $\mathbf{r}$ , and corresponding  $\mathbf{u}$ , giving the worst-case compliance.

### 4.5 Obtaining a black-and-white design

A drawback of using the design variable filter (2) is that a transition layer of elements with intermediate design variable values remain between elements representing the structure (black) and those representing holes (white). One way, suggested in the literature, to avoid this problem is to use a Heaviside projection filter (Guest et al. 2004) where a curvature parameter is increased successively until a black-and-white solution is obtained. The Heaviside filter has been proven to work well for the traditional minimum compliance problem, but we have experienced difficulties on other problem formulations where the discontinuity due to the parameter update has caused convergence problems. Therefore, we use instead a strategy where the problem is solved in two steps: first, the optimization problem is solved until convergence using the design variable filter (2); then all design variables that are at the upper or lower bound (or actually within the distance  $\varepsilon$  from these bounds) are fixed and only the design variables with intermediate values are allowed to vary in a second optimization, where no filter is used. The set of design variables which are allowed to change is thus smaller in the second step and the design is sufficiently constrained so that no new structural components will be created and no structural components will be removed.

The design obtained after solving (10) in the first step is denoted  $\boldsymbol{\rho}^{\text{opt}}$ , and the design variables in the second optimization step are denoted  $\hat{\mathbf{x}} \in \hat{\mathcal{H}}$ , where

$$\hat{\mathcal{H}} = \left\{ \hat{\mathbf{x}} \in \mathbb{R}^m \mid \varepsilon \leq \hat{x}_i \leq 1 \text{ if } 2\varepsilon < \rho_i^{\text{opt}} < 1 - \varepsilon, \right. \\ \left. \text{else } \hat{x}_i = \rho_i^{\text{opt}}, \sum_{i=1}^m dv_i \hat{\rho}_i(\mathbf{x}) = \bar{M} \right\}.$$

The initial design in the second optimization step is  $\hat{\mathbf{x}} = \boldsymbol{\rho}^{\text{opt}}$ , and since no filter is used  $\hat{\boldsymbol{\rho}} = \hat{\mathbf{x}}$ .

Obvious drawbacks of the proposed two-step strategy are that additional iterations are required and that it is not possible to use filters that are much larger than the average element size; if the transition layer consists of several elements we may obtain checkerboard patterns in the second step.

Interior-point methods such as IPOPT are known to be difficult to warm-start, i.e. to re-start from a given solution, which is the case in the second optimization step. However, by keeping the Lagrangian multipliers associated with the constraints and using these as starting values in the next optimization step together with a small initial value for the barrier parameter (see the documentation included in the IPOPT installation package (IPOPT 2014) for further details) we have not experienced any difficulties with convergence.

### 5 Gradient calculation

The solver IPOPT uses gradient data in order to iteratively change the design variables towards an optimum design. As we use fminsdp to solve (10), we do not need to treat the auxiliary variables related to the Cholesky factors that fminsdp uses to solve (14), but it is sufficient to provide gradients to the left hand side of the constraint in (10). The gradient with respect to  $z$  is straightforward and the gradient with respect to design variable  $x_i$  reads

$$\begin{aligned} \frac{\partial \mathbf{H}(\mathbf{x})}{\partial x_i} &= \sum_{j=1}^m \frac{\partial \mathbf{H}(\mathbf{x})}{\partial \rho_j} \frac{\partial \rho_j(\mathbf{x})}{\partial x_i} \\ &= \sum_{j=1}^m \frac{\partial \mathbf{H}(\mathbf{x})}{\partial \rho_j} \Psi_{ji}, \end{aligned} \tag{15}$$

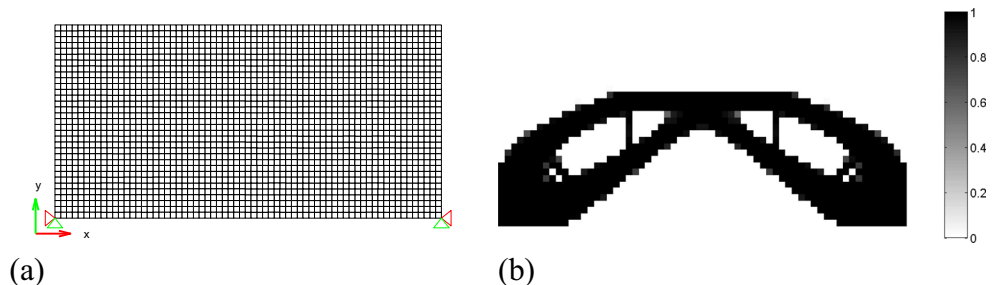
where  $\Psi_{ji}$  was defined in (2). The derivative with respect to the physical variable  $\rho_j$  is calculated as

$$\frac{\partial \mathbf{H}(\mathbf{x})}{\partial \rho_j} = \frac{\partial \mathbf{B}(\mathbf{x})}{\partial \rho_j} \mathbf{R}(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \frac{\partial \mathbf{R}(\mathbf{x})}{\partial \rho_j}, \tag{16}$$

where  $\partial \mathbf{R}(\mathbf{x}) / \partial \rho_j$  is obtained from (13) as

$$\frac{\partial \mathbf{R}(\mathbf{x})}{\partial \rho_j} = \mathbf{K}^{-1}(\mathbf{x}) \left( \frac{\partial \mathbf{B}^T(\mathbf{x})}{\partial \rho_j} - \frac{\partial \mathbf{K}(\mathbf{x})}{\partial \rho_j} \mathbf{R}(\mathbf{x}) \right). \tag{17}$$

**Fig. 3** Self weight arch. **a:** Mesh and boundary conditions, **b:** Optimized design for the maximum acceleration vector  $\mathbf{a} = \{1, 1, 0\}$



Thus, using (16) and (17), (15) can now be written

$$\begin{aligned} \frac{\partial \mathbf{H}(\mathbf{x})}{\partial x_i} &= \sum_{j=1}^m \left( \frac{\partial \mathbf{B}(\mathbf{x})}{\partial \rho_j} \mathbf{R}(\mathbf{x}) + \mathbf{R}^T(\mathbf{x}) \frac{\partial \mathbf{B}^T(\mathbf{x})}{\partial \rho_j} \right. \\ &\quad \left. - \mathbf{R}^T(\mathbf{x}) \frac{\partial \mathbf{K}(\mathbf{x})}{\partial \rho_j} \mathbf{R}(\mathbf{x}) \right) \Psi_{ji}, \end{aligned}$$

where, recalling (4),

$$\frac{\partial \mathbf{B}^T(\mathbf{x})}{\partial \rho_j} = \mathbf{A} g \frac{1}{1 - \varepsilon} \mathbf{M}_i \mathbf{G}_i N,$$

and  $\partial \mathbf{K}(\mathbf{x}) / \partial \rho_j = p \rho_j^{p-1} \mathbf{K}_j$  follows from (3).

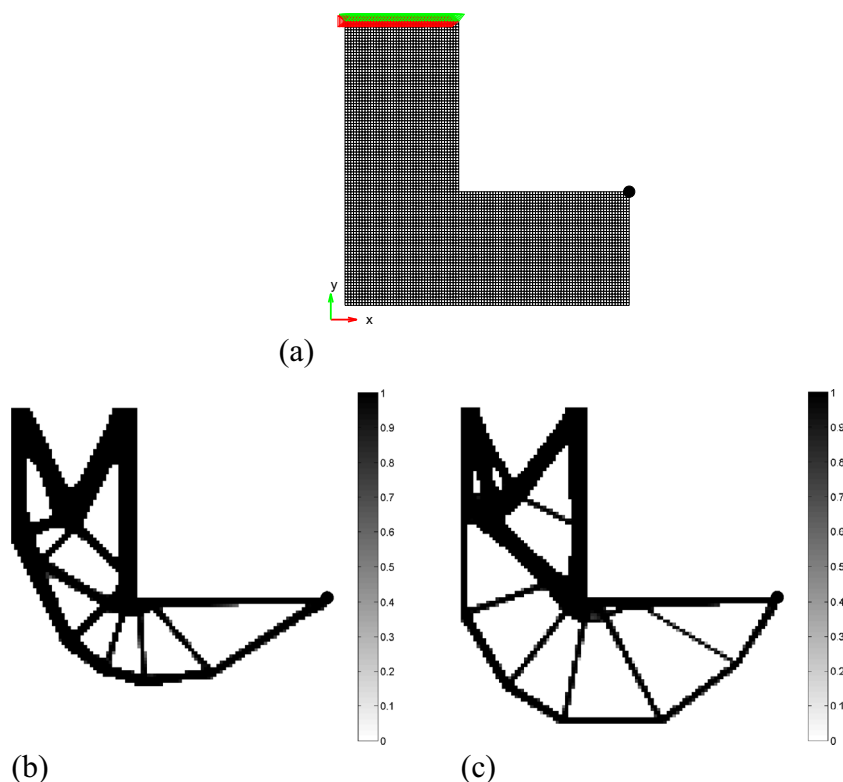
### 6 Examples

#### 6.1 Parameter settings and material

In this section we show four examples, all solved using the SDP-problem (10), and each showing different challenges. All designs shown satisfy the constraint qualification (12) and the optimality conditions (11), where we check numerically that the infinity norm of the equality conditions is below  $10^{-8}$ .

The same parameter settings are used in all examples: the SIMP-penalization exponent is  $p = 3$ , the lower bound of the design variables, representing void, is chosen as  $\varepsilon = 0.001$  and the initial design is a uniform distribution satisfying the mass constraint, unless otherwise is specified in the text. The thickness of the 2D domains is 1 mm and the design material is an aluminum with Young’s modulus 71000 MPa, Poisson’s ratio 0.33 and density  $2.8 \times 10^{-9}$  tonne/mm<sup>3</sup>. The figures displaying optimized designs show the design variables in the final design, where black is material and white represents void. Recalling Section 4.5, no filter is used in the final iteration, and thus  $\mathbf{x} = \rho$ . The maximum acceleration factors  $n_x, n_y$  and  $n_z$  entering  $N$  in (4) are in this section presented as the vector  $\mathbf{a} = \{n_x, n_y, n_z\}$ .

**Fig. 4** L-shaped beam. **a:** Mesh, boundary conditions and point mass, **b:** Robust optimized design for the maximum acceleration vector  $\mathbf{a} = \{10, 5, 0\}$ , **c:** Non-robust design obtained for two load cases with maximum acceleration vectors  $\mathbf{a} = \{0, 5, 0\}$  and  $\mathbf{a} = \{10, 0, 0\}$



**6.2 Example 1 - Self weight arch**

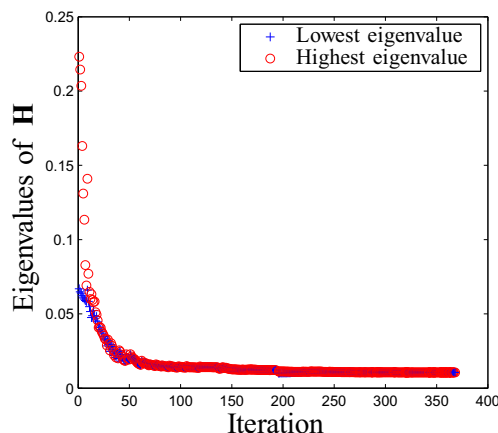
As a first example we look at the self weight arch, previously used in self-weight loaded problems by Bruyneel and Duysinx (2005) and Lee et al. (2012). The same example was also used in the introduction, Fig. 1, where we optimized for a standard gravitational acceleration,  $\mathbf{a} = \{0, -1, 0\}$ .

The design domain in Fig. 3a, now discretized with eight-node quadrilateral elements, is supported at the lower corners and has no other loading than its own weight. The dimensions are  $2000 \times 1000$  mm and the available mass is chosen to be 33 % of the mass of the entire design domain. The physical relevance of this problem may be questioned – a structure which is not supposed to carry any external load appears to be of no use – but it serves as an interesting academic example.

We now seek a design where the compliance is minimized for the worst possible acceleration, and which has lower or equal compliance for all other possible directions. Thus, we allow the acceleration to act with the same magnitude in any direction in the  $xy$ -plane, so  $\mathbf{a} = \{1, 1, 0\}$ , and we obtain the design shown in Fig. 3b. Compared to the single load case design in Fig. 1b, the worst-case compliance design has a higher compliance for that specific load case, but it is much stiffer for an acceleration in, for example, the  $x$ -direction.

**6.3 Example 2 - L-shaped beam**

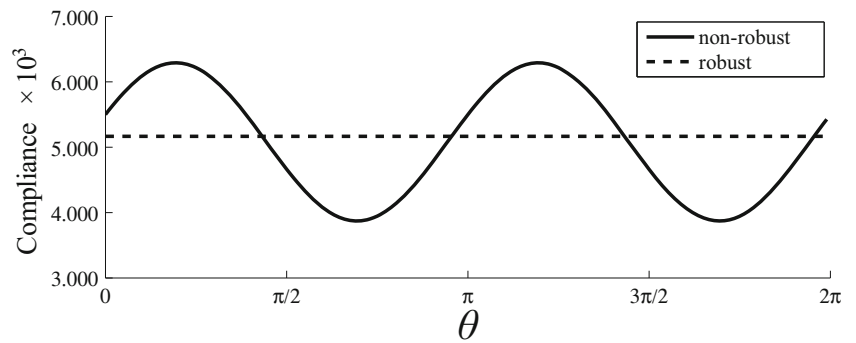
The second example is an L-shaped beam, meshed with 6400 eight-node quadrilateral elements as shown in Fig. 4a, where also boundary conditions and a point mass, illustrated as a black circle, are shown. The point mass has the same mass as the final structure, which is constrained to 33 % of the mass of the entire design domain, which has outer dimensions  $200 \times 200$  mm. In this example we use an elliptical loading region with a higher acceleration in the



**Fig. 5** Convergence history of the eigenvalues of  $\mathbf{H}$  for the design in Fig. 4b, the eigenvalues are multiple from iteration 20



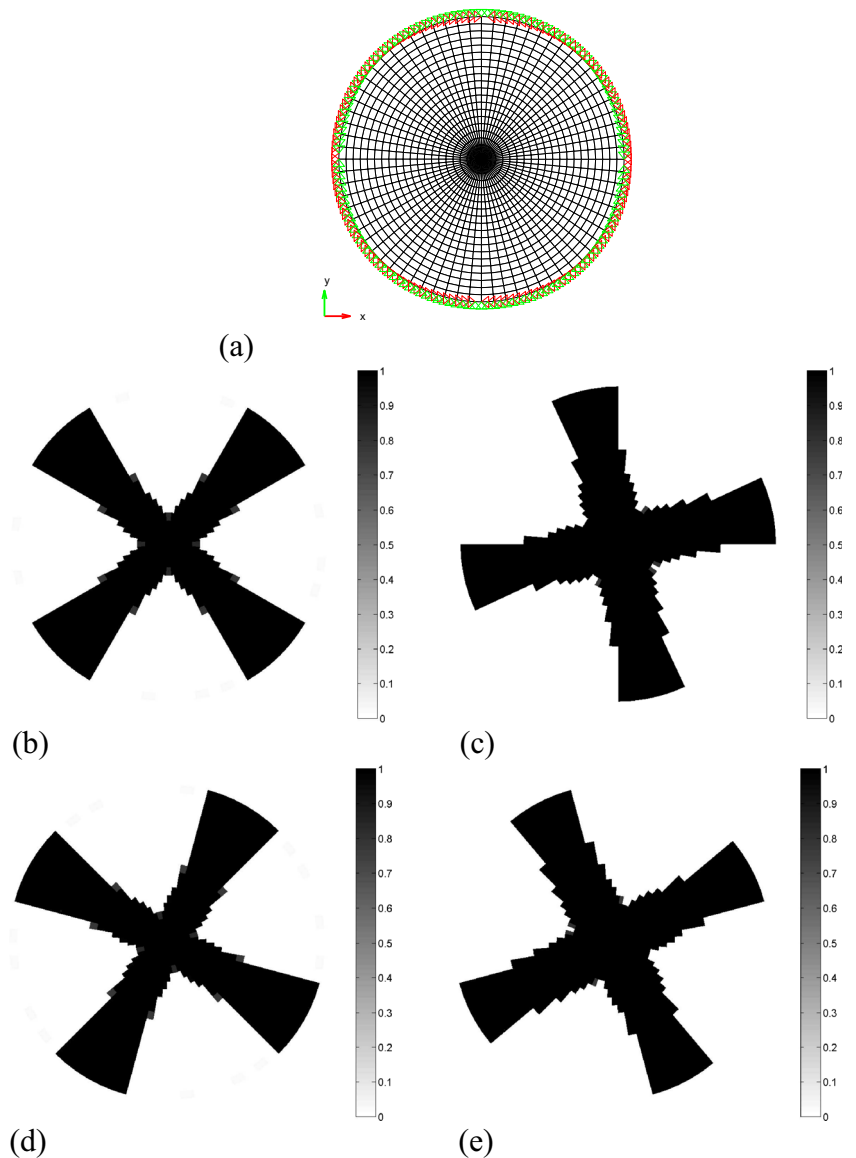
**Fig. 6** Compliance as a function of  $\theta$  in (18) for the optimized designs in Fig. 4b and c

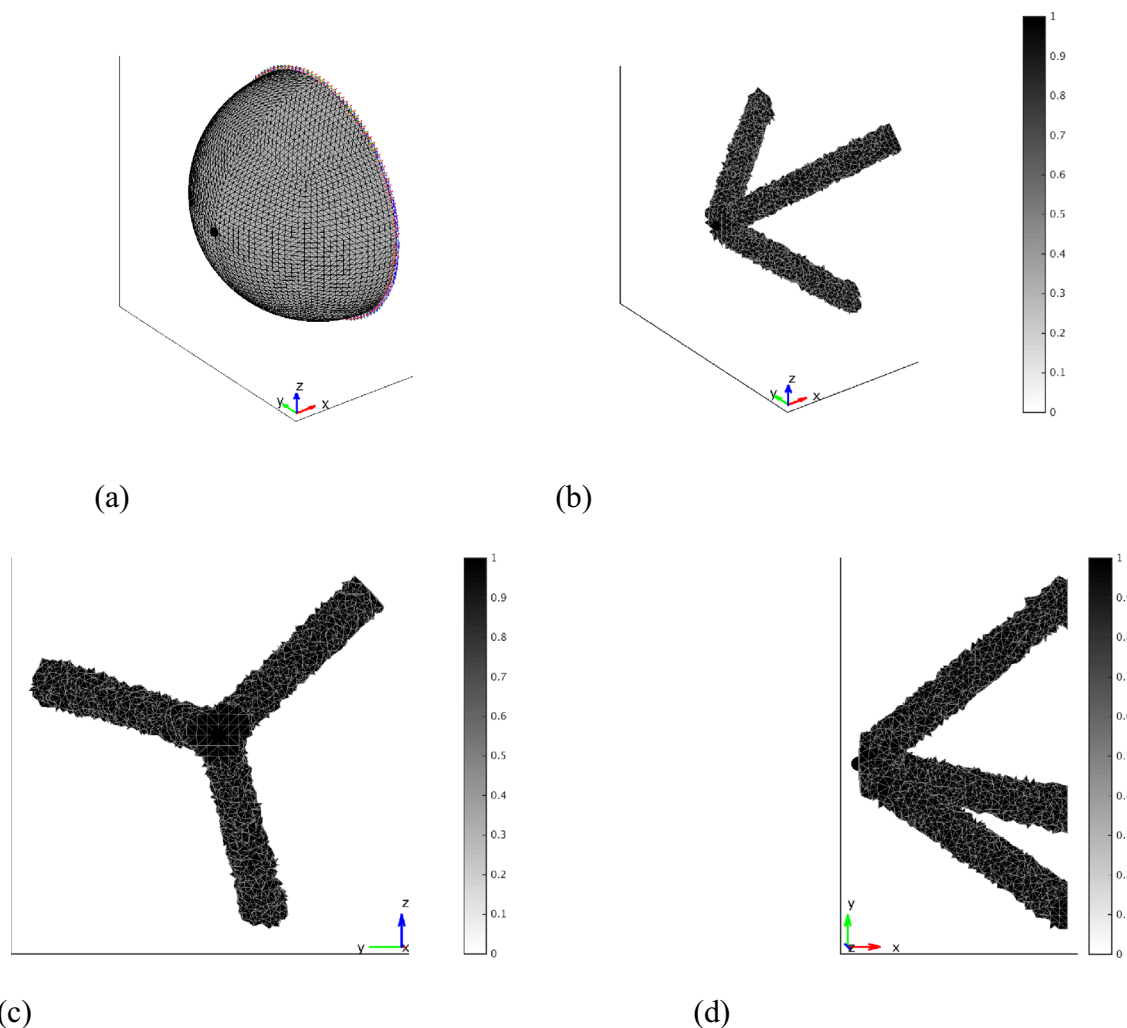


$x$ -direction,  $\mathbf{a} = \{10, 5, 0\}$ . The optimized design, Fig. 4b, does not use the entire height of the design domain as this is not beneficial for acceleration in the  $x$ -direction. Further,

we find that, because self-weight is considered, much material is placed close to the fixed boundary, and obviously, the structure connects the external mass to the fixed boundary.

**Fig. 7** Circular domain. **a:** Mesh, boundary conditions and point mass, **b:** Optimized design for a point mass with the same mass as the structure, **c:** Optimized design for a point mass much heavier than the structure, **d:** Same problem as (b) but with another starting point, **e:** Same problem as (c) but with another starting point





**Fig. 8** Hemisphere. **a:** Mesh, boundary conditions and point mass, **b, c, d:** Optimized design, isometric view, top view and side view

In order to compare the robust design with a conventional, non-robust, design we also solve a minimum weighted compliance problem, with the same mass constraint as for the robust design. The most probable choice of load cases from an engineering point-of-view would be to use two load cases,  $\mathbf{a} = \{0, 5, 0\}$  and  $\mathbf{a} = \{10, 0, 0\}$ , which is what we have used to obtain the design in Fig. 4c. The compliance for the robust and the non-robust multiple load case designs is compared in Section 6.3.1.

To highlight the fact that multiple eigenvalues are present, not only at the solution but also in most iteration steps towards the solution, we have calculated<sup>1</sup> the two

eigenvalues of  $\mathbf{H}$  and created the plot in Fig. 5. The eigenvalues are distinct in the first iterations, but from iteration 20 and onwards we have multiple eigenvalues.

### 6.3.1 Compliance for other accelerations

In order to see what the compliance for different acceleration loads is, we calculate and plot the compliance of the final designs due to the force in (4), with the vector  $\mathbf{r}$  varying as

$$\mathbf{r} = (\cos \theta, \sin \theta)^T, \tag{18}$$

for angles  $\theta \in [0, 2\pi]$ , where  $\theta = 0$  gives a vector parallel to the  $x$ -axis. The result is seen in Fig. 6. What is interesting to note is that the optimized robust design has approximately the same compliance for all possible loads

<sup>1</sup>Note that the eigenvalues are not calculated explicitly in our formulation, it is done here as an additional step in order to create the plot.

defined by (4). The difference between the maximum and the minimum compliance is on the order of  $10^{-8}$  %. This suggests that, unless there are geometric restrictions, the robust optimization will result in designs which have the same compliance in all directions. This strengthens the arguments in Section 2.4 and motivates the use of SDP compared to solving the eigenvalue problem (9) directly.

For the non-robust, multiple load case design in Fig. 4c we find that the maximum compliance is not attained for any of the two angles ( $\theta = 0$  and  $\theta = \pi/2$ ) corresponding to the applied loads, implying that this is not a robust design.

### 6.4 Example 3 - Circular domain

A circular design domain with radius 10 mm is discretized with a rotationally symmetric mesh consisting of second order triangular and quadrilateral elements. The circle is fixed at the outer boundary and has a point mass in the center as shown in Fig. 7a. The filter radius is chosen as  $R = 1$  mm and the allowable mass is 40 % of the mass of the entire design domain. Two different sizes of the point mass are used: Fig. 7b shows the optimized design when the point mass has the same mass as the final structure and Fig. 7c shows the optimized design when the point mass is 1000 times heavier than the structure. When the non-structural mass is much heavier than the structure, the design has a quite even thickness of the four structural members in order to minimize the displacement of the non-structural mass, whereas we find that more material is placed close to the fixed boundary when the structural- and non-structural masses are equal.

Because of the rotational symmetry, there exist several designs, differing only by a rotation, with the same performance. Thus, there exists several local optima, and if we start from the uniform design  $x_i = 0.35$ , instead of  $x_i = 0.40$ , we obtain the same, but rotated, designs seen in Fig. 7d and e.

### 6.5 Example 4 - Hemisphere

In this example we have modelled a hemisphere with radius 200 mm and filled it with a linear tetrahedral mesh, resulting in 327493 elements. All nodes on the flat surface are fixed in all directions and a point mass is applied as shown in Fig. 8a. The point mass is 10 times heavier than the final structure which is constrained to 5 % of the mass of the entire design domain. The maximum possible acceleration is the same in all directions, hence  $\mathbf{a} = \{1, 1, 1\}$ . The optimized design, Figs. 8b, c and d, is a tripod structure; i.e., a structure with three equally sized legs with 120 degree angle between them.

## 7 Conclusions

The proposed SDP-formulation provides a computationally tractable form of worst-case compliance topology optimization problems, previously solved with eigenvalue formulations which are non-smooth when multiple eigenvalues occur. The formulation of the SDP-problem, with a small matrix inequality constraint, makes the problem computationally efficient and, even though an infinite number of load cases are taken into account, the computational cost is similar to that of solving a standard compliance minimization problem with the same number of load cases as there are spatial dimensions. The presented scaling of the mass matrix (Section 2.2) allows us to optimize self-weight loaded structures with a SIMP-formulation without introducing non-differentiable functions. The method for obtaining a black-and-white design (Section 4.5), while still avoiding mesh dependency by limiting the smallest size of structural members, has proven to work very well for the presented examples, in which the final designs are almost completely black-and-white. Finally, necessary first-order optimality conditions are derived and checked numerically, providing confidence in that we have found locally optimal designs.

**Acknowledgments** This research was supported in part by NFFP Grant No. 2013-01221, which is funded by the Swedish Armed Forces, the Swedish Defence Materiel Administration and the Swedish Governmental Agency for Innovation Systems, and in part by the Swedish Foundation for Strategic Research, Grant No. AM13-0029.

## Appendix A: Equivalence between bounded eigenvalue and SDP

**Proposition 1** Let  $\mathbf{H} \in \mathbb{S}^s$  and  $\lambda_1(\mathbf{H}) = \max_{i=1, \dots, s} \lambda_i(\mathbf{H})$ .

Then

$$\lambda_1(\mathbf{H}) \leq z, \Leftrightarrow \mathbf{H} - z\mathbf{I} \preceq \mathbf{0}.$$

*Proof* “ $\mathbf{H} - z\mathbf{I} \preceq \mathbf{0}$ ” means that

$$\begin{aligned} \mathbf{y}^\top(\mathbf{H} - z\mathbf{I})\mathbf{y} &\leq 0 \quad \forall \mathbf{y} \in \mathbb{R}^s \Leftrightarrow \\ (\alpha\mathbf{v})^\top(\mathbf{H} - z\mathbf{I})(\alpha\mathbf{v}) &\leq 0 \quad \forall (\alpha, \mathbf{v}) \in \mathbb{R}^{s+1} : \|\mathbf{v}\| = 1 \Leftrightarrow \\ \alpha^2\mathbf{v}^\top(\mathbf{H} - z\mathbf{I})\mathbf{v} &\leq 0 \quad \forall (\alpha, \mathbf{v}) \in \mathbb{R}^{s+1} : \|\mathbf{v}\| = 1 \Leftrightarrow \\ \mathbf{v}^\top(\mathbf{H} - z\mathbf{I})\mathbf{v} &\leq 0 \quad \forall \mathbf{v} \in \mathbb{R}^s : \|\mathbf{v}\| = 1 \Leftrightarrow \\ \mathbf{v}^\top\mathbf{H}\mathbf{v} &\leq z \quad \forall \mathbf{v} \in \mathbb{R}^s : \|\mathbf{v}\| = 1 \Leftrightarrow \\ \lambda_1(\mathbf{H}) = \max_{\|\mathbf{v}\|=1} \mathbf{v}^\top\mathbf{H}\mathbf{v} &\leq z, \end{aligned}$$

where the equality to the left in the last line follows from the Rayleigh-Ritz theorem (Horn and Johnson 1985, Theorem 4.2.2).  $\square$

### Appendix B: Worst-case compliance and optimization problem for a singular stiffness matrix

For a singular stiffness matrix  $\mathbf{K}(\mathbf{x})$  the compliance is still a well-defined function and can be expressed as

$$C(\mathbf{x}, \mathbf{r}) = - \inf_{\mathbf{v} \in \mathbb{R}^n} \Pi(\mathbf{v}, \mathbf{x}, \mathbf{r}),$$

where

$$\Pi(\mathbf{v}, \mathbf{x}, \mathbf{r}) = \frac{1}{2} \mathbf{v}^\top \mathbf{K}(\mathbf{x}) \mathbf{v} - \mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v},$$

is the total potential energy for a general nodal displacement  $\mathbf{v}$ . (See Klarbring (2015) for a recent general discussion of compliance minimization.)

The worst-case compliance is defined as

$$P(\mathbf{x}) = \sup_{\mathbf{r} \in T} C(\mathbf{x}, \mathbf{r}). \tag{19}$$

By straightforward manipulations we get<sup>2</sup>

$$\begin{aligned} P(\mathbf{x}) &= - \inf_{\mathbf{r} \in T} \left( \inf_{\mathbf{v} \in \mathbb{R}^n} \Pi(\mathbf{v}, \mathbf{x}, \mathbf{r}) \right) \\ &= - \inf_{\mathbf{v} \in \mathbb{R}^n} \left( \inf_{\mathbf{r} \in T} \Pi(\mathbf{v}, \mathbf{x}, \mathbf{r}) \right) \\ &= - \inf_{\mathbf{v} \in \mathbb{R}^n} \left( \frac{1}{2} \mathbf{v}^\top \mathbf{K}(\mathbf{x}) \mathbf{v} + \inf_{\mathbf{r} \in T} (-\mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v}) \right) \\ &= - \inf_{\mathbf{v} \in \mathbb{R}^n} \left( \frac{1}{2} \mathbf{v}^\top \mathbf{K}(\mathbf{x}) \mathbf{v} - \sup_{\mathbf{r} \in T} \mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v} \right). \end{aligned}$$

The inner maximization problem is solved by noting that

$$\mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v} \leq |\mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v}| \leq \|\mathbf{r}\| \|\mathbf{B}(\mathbf{x}) \mathbf{v}\| \leq \|\mathbf{B}(\mathbf{x}) \mathbf{v}\|,$$

using the Cauchy-Schwarz inequality and the fact that  $\|\mathbf{r}\| \leq 1$  for  $\mathbf{r} \in T$ . Equality holds for  $\mathbf{r} = \mathbf{B}(\mathbf{x}) \mathbf{v} / \|\mathbf{B}(\mathbf{x}) \mathbf{v}\|$ , so we have

$$\sup_{\mathbf{r} \in T} \mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v} = \sup_{\|\mathbf{r}\|=1} \mathbf{r}^\top \mathbf{B}(\mathbf{x}) \mathbf{v} = \|\mathbf{B}(\mathbf{x}) \mathbf{v}\|, \tag{20}$$

where the first equality follows from Rockafeller (1972, Corollary 32.3.1). Substitution in (19) yields

$$\begin{aligned} P(\mathbf{x}) &= - \inf_{\mathbf{v} \in \mathbb{R}^n} \left( \frac{1}{2} \mathbf{v}^\top \mathbf{K}(\mathbf{x}) \mathbf{v} - \|\mathbf{B}(\mathbf{x}) \mathbf{v}\| \right) \\ &= \sup_{\mathbf{v} \in \mathbb{R}^n} \left( \|\mathbf{B}(\mathbf{x}) \mathbf{v}\| - \frac{1}{2} \mathbf{v}^\top \mathbf{K}(\mathbf{x}) \mathbf{v} \right). \end{aligned} \tag{21}$$

We are now interested in essentially the equivalent problem to (9), i.e.,

$$\min_{\mathbf{x} \in \mathcal{H}} P(\mathbf{x}),$$

but now without requiring a non-singular stiffness matrix. As for (9) we rephrase it into a bound formulation, which,

<sup>2</sup>Here we use  $\inf_x [\inf_u f(x, u)] = \inf_u [\inf_x f(x, u)]$  and  $\sup_x f(x) = - \inf_x [-f(x)]$ , for  $x$  and  $u$  varying over arbitrary sets.

using (21), leads to the following optimization problem:

$$\begin{aligned} &\min_{z \in \mathbb{R}, \mathbf{x} \in \mathcal{H}} z \\ &\text{subject to } \|\mathbf{B}(\mathbf{x}) \mathbf{v}\| - \frac{1}{2} \mathbf{v}^\top \mathbf{K}(\mathbf{x}) \mathbf{v} \leq \frac{1}{2} z, \quad \forall \mathbf{v} \in \mathbb{R}^n. \end{aligned} \tag{22}$$

This is a semi-infinite problem since it has a finite number of variables, but an infinite number of constraints. However, it can be shown by the following theorem that this problem is equivalent to a semi-definite program.

#### Theorem

$$\begin{pmatrix} z \mathbf{I} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{K} \end{pmatrix} \succeq \mathbf{0} \Leftrightarrow -\frac{1}{2} \mathbf{v}^\top \mathbf{K} \mathbf{v} + \|\mathbf{B} \mathbf{v}\| \leq \frac{1}{2} z, \quad \forall \mathbf{v} \in \mathbb{R}^n.$$

*Proof* The proof resembles that of Lemma 2.2 in Ben-Tal and Nemirovski (1997).

Since the matrix is positive semi-definite,

$$\begin{aligned} (-\boldsymbol{\tau} \ \mathbf{u}) \begin{pmatrix} z \mathbf{I} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{K} \end{pmatrix} \begin{pmatrix} -\boldsymbol{\tau} \\ \mathbf{u} \end{pmatrix} &\geq 0 \quad \forall (\boldsymbol{\tau}, \mathbf{u}) \in \mathbb{R}^{s+n} \Leftrightarrow \\ z \boldsymbol{\tau}^\top \boldsymbol{\tau} + \mathbf{u}^\top \mathbf{K} \mathbf{u} - 2 \mathbf{u}^\top \mathbf{B}^\top \boldsymbol{\tau} &\geq 0 \quad \forall (\boldsymbol{\tau}, \mathbf{u}) \in \mathbb{R}^{s+n} \Leftrightarrow \\ z \lambda^2 + \mathbf{u}^\top \mathbf{K} \mathbf{u} - 2 \mathbf{u}^\top \mathbf{B}^\top \boldsymbol{\tau} &\geq 0 \quad \forall (\boldsymbol{\tau}, \mathbf{u}, \lambda) \\ &\in \mathbb{R}^{s+n+1} : \|\boldsymbol{\tau}\|^2 = \lambda^2. \end{aligned}$$

Now let  $\mathbf{u} = \lambda \mathbf{y}$  and  $\boldsymbol{\tau} = \lambda \boldsymbol{\kappa}$ . Then the last inequality is equivalent to

$$\begin{aligned} z \lambda^2 + \lambda^2 \mathbf{y}^\top \mathbf{K} \mathbf{y} - 2 \lambda^2 \mathbf{y}^\top \mathbf{B}^\top \boldsymbol{\kappa} &\geq 0 \quad \forall (\boldsymbol{\kappa}, \mathbf{y}, \lambda) \\ &\in \mathbb{R}^{s+n+1} : \|\boldsymbol{\kappa}\|^2 = 1 \Leftrightarrow \\ z + \mathbf{y}^\top \mathbf{K} \mathbf{y} - 2 \mathbf{y}^\top \mathbf{B}^\top \boldsymbol{\kappa} &\geq 0 \quad \forall (\boldsymbol{\kappa}, \mathbf{y}) \\ &\in \mathbb{R}^{n+s} : \|\boldsymbol{\kappa}\|^2 = 1. \end{aligned}$$

The latter holds in particular for  $\boldsymbol{\kappa} = \arg \max_{\|\boldsymbol{\kappa}\|=1} \mathbf{y}^\top \mathbf{B}^\top \boldsymbol{\kappa}$ , for which  $\mathbf{y}^\top \mathbf{B}^\top \boldsymbol{\kappa} = \|\mathbf{B} \mathbf{y}\|$ , see (20). The last inequality is thus equivalent to

$$\begin{aligned} z + \mathbf{y}^\top \mathbf{K} \mathbf{y} - 2 \|\mathbf{B} \mathbf{y}\| &\geq 0 \quad \forall \mathbf{y} \in \mathbb{R}^n \Leftrightarrow \\ -\mathbf{y}^\top \mathbf{K} \mathbf{y} + 2 \|\mathbf{B} \mathbf{y}\| &\leq z \quad \forall \mathbf{y} \in \mathbb{R}^n \Leftrightarrow \\ -\frac{1}{2} \mathbf{y}^\top \mathbf{K} \mathbf{y} + \|\mathbf{B} \mathbf{y}\| &\leq \frac{1}{2} z \quad \forall \mathbf{y} \in \mathbb{R}^n. \end{aligned}$$

□

Using this theorem, the semi-infinite optimization problem (22) can be formulated as the semi-definite program

$$\begin{aligned} &\min_{z \in \mathbb{R}, \mathbf{x} \in \mathcal{H}} z \\ &\text{subject to } \begin{pmatrix} z \mathbf{I} & \mathbf{B}(\mathbf{x}) \\ \mathbf{B}(\mathbf{x})^\top & \mathbf{K}(\mathbf{x}) \end{pmatrix} \succeq \mathbf{0}. \end{aligned} \tag{23}$$

This problem is convex if  $\mathbf{B}$  and  $\mathbf{K}$  are linear in  $\mathbf{x}$  and does not require  $\mathbf{K}$  to be non-singular. Unfortunately, currently available solvers for semi-definite programs are not suitable for matrix inequalities with large matrices (see however Bogani et al. (2009) and Stingl et al. (2009)).

For the special case of a non-singular stiffness matrix, we note two interesting results obtained from (23):

1. We may use the Schur-complement theorem (Boyd and Vandenberghe (2004, p. 560-561), Horn and Johnson (1985, Theorem 7.7.6)) and (6) to rewrite (23) into (10) derived in Section 2.4.
2. We may also obtain the generalized eigenvalue formulation proposed by Brittain et al. (2012). To start with, it is straightforward to show that

$$\begin{pmatrix} z\mathbf{I} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{K} \end{pmatrix} \succeq \mathbf{0} \Leftrightarrow \begin{pmatrix} \mathbf{K} & \mathbf{B}^\top \\ \mathbf{B} & z\mathbf{I} \end{pmatrix} \succeq \mathbf{0}.$$

Then assuming  $z > 0$  and applying the Schur-complement theorem to the right side of the equivalence one finds that this inequality is equivalent, in turn, to

$$\begin{aligned} \mathbf{K} - z^{-1}\mathbf{B}^\top\mathbf{B} &\succeq \mathbf{0} && \Leftrightarrow \\ \mathbf{x}^\top(\mathbf{K} - z^{-1}\mathbf{B}^\top\mathbf{B})\mathbf{x} &\geq 0 \quad \forall \mathbf{x} \in \mathbb{R}^n && \Leftrightarrow \\ z &\geq \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{K}\mathbf{x}} \quad \forall \mathbf{x} \in \mathbb{R}^n && \Leftrightarrow \\ z &= \max_{\|\mathbf{x}\|=1} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{K}\mathbf{x}} = \mu_1(\mathbf{K}, \mathbf{B}^\top\mathbf{B}), \end{aligned}$$

where  $\mu_1(\cdot, \cdot)$  is the largest generalized eigenvalue of a given matrix pencil. The last step in the above chain of equivalences follows from Theorem 2.4 in Hestnes (1975, Chapter 2). Problem (23) can now be replaced by

$$\min_{\mathbf{x} \in \mathcal{H}} \mu_1(\mathbf{K}(\mathbf{x}), \mathbf{B}(\mathbf{x})^\top\mathbf{B}(\mathbf{x})),$$

which is similar to the generalized eigenvalue formulation found in Brittain et al. (2012).

## References

- Achtziger W, Kocvara M (2007) Structural topology optimization with eigenvalues. *SIAM J Optim* 18(4):1129–1164
- Altair Engineering Inc. (2014) HyperMesh 13.0, users manual
- Aroztegui M, Herskovits J, Roche JR, Bazan E (2014) A feasible direction interior point algorithm for nonlinear semidefinite programming. *Struct Multidiscip Optim*:1–17. doi:10.1007/s00158-014-1090-2. ISSN 1615-147X
- Ben-Tal A, El Ghaoui L, Nemirovski A (2009) *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press
- Ben-Tal A, Nemirovski A (1997) Robust truss topology design via semidefinite programming. *SIAM J Optim* 7(4):991–1016
- Bendsøe MP (1989) Optimal shape design as a material distribution problem. *Struct Optim* 1(4):193–202
- Benson HY, Vanderbei RJ (2003) Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming. *Math Program* 2:279–302
- Bogani C, Kocvara M, Stingl M (2009) A new approach to the solution of the VTS problem with vibration and buckling constraints. In: 8th World Congress on Structural and Multidisciplinary Optimization
- Bonnans JF, Shapiro A (2000) *Perturbation analysis of optimization problems*. Springer
- Boyd S, Vandenberghe L (2004) *Convex optimization*. Cambridge University Press
- Brittain K, Silva M, Tortorelli DA (2012) Minmax topology optimization. *Struct Multidiscip Optim* 45(5):657–668
- Bruns TE, Tortorelli DA (2001) Topology optimization of non-linear elastic structures and compliant mechanisms. *Comput Methods Appl Mech Eng* 190(26-27):3443–3459
- Bruyneel M, Duysinx P (2005) Note on topology optimization of continuum structures including self-weight. *Struct Multidiscip Optim* 29:245–256
- Burer S, Monteiro RDC, Zhang Y (2002) Solving a class of semidefinite programs via nonlinear programming. *Math Program* 93:9702123
- Cherkaev E, Cherkaev A (2008) Minimax optimization problem of structural design. *Comput Struct* 86(13):1426–1435
- Christiansen S, Patriksson M, Wynter L (2001) Stochastic bilevel programming in structural optimization. *Struct Multidiscip Optim* 21(5):361–371
- Dunning PD, Kim HA (2013) Robust topology optimization: minimization of expected and variance of compliance. *AIAA J* 51(11):2656–2664
- Ertel S, Schittkowski K, Zillober C (2008) Sequential convex programming for free material optimization with displacement and stress constraints. Technical report, Department of Computer Science. University of Bayreuth
- Fletcher R (1985) Semi-definite matrix constraints in optimization. *SIAM J Control Optim* 23:493–513
- de Gournay F, Allaire G, Jouve F (2008) Shape and topology optimization of the robust compliance via the level set method. *ESAIM Control Optim Calc Var* 14(01):43–70
- Guest JK, Prévost JH, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int J Numer Methods Eng* 61(2):238–254
- Hestnes M (1975) *Optimization theory*. John Wiley & Sons
- Hiriart-Urruty JB, Lemarchal C (1993) *Convex analysis and minimization algorithms I*. Springer
- Horn RA, Johnson CR (1985) *Matrix analysis*. Cambridge University Press
- IPOPT (2014) A tutorial for downloading, installing, and using IPOPT. <http://www.coin-or.org/Ipopt/documentation/>
- Kanno Y (2011) An implicit formulation of mathematical program with complementarity constraints for application to robust structural optimization. *J Oper Res Soc Jpn-Keiei Kagaku* 54(2):65
- Kharmanda G, Olhoff N, Mohamed A, Lemaire M (2004) Reliability-based topology optimization. *Struct Multidiscip Optim* 26(5):295–307
- Klarbring A (2015) Design optimization based on state problem functionals. *Struct Multidiscip Optim*. doi:10.1007/s00158-015-1240-1
- Kočvara M, Stingl M (2003) PENNON - A code for convex nonlinear and semidefinite programming. *Optim Methods Softw* 18:317–333
- Lee E, James KA, Martins JRR (2012) Stress-constrained topology optimization with design-dependent loading. *Struct Multidiscip Optim* 46(5):647–661
- Ohsaki M, Fujisawa K, Katoh N, Kanno Y (1999) Semi-definite programming for topology optimization of trusses under

- multiple eigenvalue constraints. *Comput Methods Appl Mech Eng* 180(1):203–217
- Overton ML (1992) Large-scale optimization of eigenvalues. *SIAM J Optim* 2(1):88–120
- Pataki G (1998) On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math Oper Res* 23(2):339–358
- Pedersen NL, Nielsen AK (2003) Optimization of practical trusses with constraints on eigenfrequencies, displacements, stresses, and buckling. *Struct Multidiscip Optim* 25(5):436–445
- Rockafeller RT (1972) *Convex analysis*. Princeton
- Seyranian AP, Lund E, Olhoff N (1994) Multiple eigenvalues in structural optimization problems. *Struct Multidiscip Optim* 8(4):207–227
- Sigmund O (2007) Morphology-based black and white filters for topology optimization. *Struct Multidiscip Optim* 33(4):401–424
- Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct Multidiscip Optim* 16(1):68–75
- Stingl M, Kocvara M, Leugering G (2009) Free material optimization with fundamental eigenfrequency constraints. *SIAM J Optim* 20(1):524–547
- Svanberg K (1987) The method of moving asymptotes - a new method for structural optimization. *Int J Numer Methods Eng* 24(2):359–373
- Takezawa A, Nii S, Kitamura M, Kogiso N (2011) Topology optimization for worst load conditions based on the eigenvalue analysis of an aggregated linear system. *Comput Methods Appl Mech Eng* 200(25):2268–2281
- Thore C-J (2013) Fminsdp – a code for solving optimization problems with matrix inequality constraints. <http://www.mathworks.com/matlabcentral/fileexchange/43643-fminsdp>
- Thore C-J, Holmberg E, Klarbring A (2015) Large-scale robust topology optimization under load-uncertainty. In: *Proceedings, 11th World Congress on Structural and Multidisciplinary Optimization*
- Vanderbei RJ, Benson HY (2000) On formulating semidefinite programming problems as smooth convex nonlinear optimization problems. Technical report. Center for Discrete Mathematics & Theoretical Computer Science
- Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program* 106(1):25–57