

A Sequential Convex Semidefinite Programming Algorithm for Multiple-Load Free Material Optimization

M. Stingl*, M. Kočvara[†] and G. Leugering[‡]

August 11, 2007

Keywords. structural optimization, material optimization, semidefinite programming, sequential convex programming

AMS subject classifications. 74B05, 74P05, 90C90, 90C25, 90C22

Abstract

A new method for the efficient solution of free material optimization problems is introduced. The method extends the sequential convex programming (SCP) concept to a class of optimization problems with matrix variables. The basic idea of the new method is to approximate the original optimization problem by a sequence of sub-problems, in which nonlinear functions (defined in matrix variables) are approximated by block-separable, convex functions. The subproblems are semidefinite programs with a favorable structure, which can be efficiently solved by existing SDP software. The new method is shown to be globally convergent. The article is concluded by a series of numerical experiments demonstrating the effectiveness of the generalized SCP approach.

1 Introduction

Free material optimization (FMO) is a branch of structural optimization that gains more and more interest in the recent years. The underlying FMO model was introduced in [3] and has been studied in several further articles as, for example, [1, 24]. The optimization variable is the material tensor, which is allowed to vary from point to point. The method is supported by powerful optimization and numerical techniques, which allow for scenarios with complex bodies and fine finite-element meshes. Rather than solving the (primal) FMO problem directly, the most successful method for the solution of multiple load FMO problems is based on dualization of the original problem and leads to large scale semidefinite programming problems [1]. The dual method has been implemented in a software package MOPED, which has been recently applied to

*Institute of Applied Mathematics, University of Erlangen, Martensstr. 3, 91058 Erlangen, Germany (stingl@am.uni-erlangen.de)

[†]School of Mathematics, University of Birmingham, Edgbaston Birmingham B15 2TT, UK (kocvara@maths.bham.ac.uk)

[‡]Institute of Applied Mathematics, University of Erlangen, Martensstr. 3, 91058 Erlangen, Germany (leugering@am.uni-erlangen.de)

real-world applications. Nevertheless, the dual semidefinite approach has two major disadvantages. First of all, the computational complexity of the method depends cubically on the number of load cases [12]. This makes the approach unpractical for 3D problems with more than a few (typically 3-5) load cases. Moreover, it is almost impossible to apply the dual approach to extended (multi-disciplinary) FMO problems. This is a serious drawback, as additional constraints as, for instance, displacement-based constraints play an important role in many real-world applications (compare [9, 13]). A direct treatment of the primal problem seems to avoid both of these difficulties, but unfortunately no successful algorithmic concept has been found for the solution of this problem so far.

On the other hand during the last two decades powerful optimization methods have been developed for the solution of topology optimization problems, based on the so-called SIMP approach (see [2]), a related field of structural optimization. The most successful methods, CONLIN [6], the method of moving asymptotes [19, 20] and the sequential convex programming method [22, 23] are all based on separable convex first order approximations of nonlinear functions. The mathematical structure of SIMP-based optimization problems is closely related to the structure of the primal FMO problem. The only significant difference is that the design variables in FMO (material matrices/tensors) are defined in matrix spaces, while the variables in SIMP-based problems (density, thickness) are typically of real type. Motivated by this fact, we propose a new optimization method, which generalizes the sequential convex approximation concept to functions defined on matrix spaces. In the scope of this article we investigate theoretical as well as numerical aspects of the new method. Moreover, we demonstrate by numerical experiments that the new method offers a viable alternative and supplement to existing methods in the field of material optimization.

This article is structured as follows: In the second section we briefly repeat the free material optimization (FMO) model and define the basic problem statement. Then, in the third section, we introduce so-called separable hyperbolic approximations of functions defined on matrix spaces. In Section 4, these approximations are used to construct a globally convergent algorithm for the solution of certain convex semidefinite programs. In Section 5, we describe an algorithm used for the efficient solution of separable convex semidefinite programs, which appeared as sub-problems in Section 4. In Section 6, algorithmic details along with extensive numerical studies by means of FMO problems are presented. The article is concluded by an outlook on future research related to the subject.

Throughout this article we use the following notation: We denote by \mathbb{S}^d the space of symmetric $d \times d$ -matrices equipped with the standard inner product $\langle \cdot, \cdot \rangle_{\mathbb{S}^d}$ defined by $\langle A, B \rangle_{\mathbb{S}^d} := \text{Tr}(AB)$ for any pair of matrices $A, B \in \mathbb{S}^d$. We denote by \mathbb{S}_+^d the cone of all positive semidefinite matrices in \mathbb{S}^d and use the abbreviation $A \succ_{\mathbb{S}^d} 0$ for matrices $A \in \mathbb{S}_+^d$. Moreover, for $A, B \in \mathbb{S}^d$, we say that $A \succ_{\mathbb{S}^d} B$ if and only if $A - B \succ_{\mathbb{S}^d} 0$, and similarly for $A \preccurlyeq_{\mathbb{S}^d} B$. Further we make use of the operator $\text{smat} : \mathbb{S}^d \rightarrow \mathbb{R}^{\hat{d}}$ with $\hat{d} := d(d+1)/2$, which maps a matrix $A \in \mathbb{S}^d$ with entries $(a_{i,j})_{i,j=1}^d$ to the vector $(a_{1,1}, a_{2,1}, a_{2,2}, a_{3,1}, a_{3,2}, a_{3,3}, \dots, a_{d,1}, a_{d,2}, \dots, a_{d,d})$. Along with this operator we define $\text{svec} : \mathbb{R}^{\hat{d}} \rightarrow \mathbb{S}^d$ as the inverse operator of smat .

2 Free material optimization

We briefly introduce the free material optimization problem:

Let $\Omega \subset \mathbb{R}^2$ be a two-dimensional bounded domain¹ with a Lipschitz boundary. By $u(x) = (u_1(x), u_2(x))$ we denote the displacement vector at a point x of the body under an external load, and by

$$e_{ij}(u(x)) = \frac{1}{2} \left(\frac{\partial u_i(x)}{\partial x_j} + \frac{\partial u_j(x)}{\partial x_i} \right) \quad \text{for } i, j = 1, 2$$

the associated (small-)strain tensor. We assume that our system is governed by linear Hooke's law, i.e., the stress is a linear function of the strain

$$\sigma_{ij}(x) = E_{ijk\ell}(x) e_{k\ell}(u(x)) \quad (\text{in tensor notation}),$$

where E is the elastic stiffness tensor. The symmetries of E allow us to write the 2nd order tensors e and σ as vectors

$$e = (e_{11}, e_{22}, \sqrt{2}e_{12})^T \in \mathbb{R}^3, \quad \sigma = (\sigma_{11}, \sigma_{22}, \sqrt{2}\sigma_{12})^T \in \mathbb{R}^3.$$

Correspondingly, the 4th order tensor E can be written as a symmetric 3×3 matrix

$$E = \begin{pmatrix} E_{1111} & E_{1122} & \sqrt{2}E_{1112} \\ & E_{2222} & \sqrt{2}E_{2212} \\ \text{sym.} & & 2E_{1212} \end{pmatrix}. \quad (1)$$

In this notation, Hooke's law reads as $\sigma(x) = E(x)e(u(x))$.

Given a set of external load functions $f_k \in [L_2(\Gamma)]^2$, $k \in \mathcal{K} = \{1, 2, \dots, K\}$, where Γ is a part of $\partial\Omega$ that is not fixed by Dirichlet boundary conditions we are able to state for each load case $k \in \mathcal{K}$ a basic boundary value problem of the type:

$$\begin{aligned} \text{Find } u_k \in [H^1(\Omega)]^2, \text{ such that} \quad & (2) \\ \begin{aligned} \operatorname{div}(\sigma) &= 0 & \text{in } \Omega \\ \sigma \cdot n &= f_k & \text{on } \Gamma \\ u_k &= 0 & \text{on } \Gamma_0 \\ \sigma &= E \cdot e(u_k) & \text{in } \Omega \end{aligned} \end{aligned}$$

Here Γ and Γ_0 are open disjunctive subsets of $\partial\Omega$. Applying Green's formula we obtain corresponding weak forms, so called weak equilibrium equations, as follows:

$$\text{Find } u_k \in \mathcal{V}, \text{ such that} \quad (3)$$

$$\int_{\Omega} \langle E(x)e(u_k(x)), e(v(x)) \rangle dx = \int_{\Gamma} f_k(x) \cdot v(x) dx, \quad \forall v \in \mathcal{V},$$

where $\mathcal{V} = \{u \in [H^1(\Omega)]^2 \mid u = 0 \text{ on } \Gamma_0\} \supset [H_0^1(\Omega)]^2$ reflects the Dirichlet boundary conditions.

In *free material optimization* (FMO), the design variable is the elastic stiffness tensor E which is a function of the space variable x (see [3]). The only constraints on E are that it is physically reasonable, i.e., that E is symmetric and positive semidefinite. This gives rise to the following definition

$$\mathcal{E}_0 := \{E \in L^\infty(\Omega)^{3 \times 3} \mid E = E^\top, E \succeq \underline{\rho}I \text{ a. e. in } \Omega\},$$

¹The entire presentation is given for two-dimensional bodies, to keep the notation simple. Analogously, all this can be done for three-dimensional solids.

where $\rho \in \mathbb{R}^+$ is a suitable non-negative number and I denotes the identity matrix. The choice of L^∞ is due to the fact that we want to allow for maximal-material/minimal-material situations. A frequently used measure for the stiffness of the material tensor is its trace. In order to avoid arbitrarily stiff material, we add pointwise stiffness restrictions of the form $\text{Tr}(E) \leq \bar{\rho}$, where $\bar{\rho}$ is a finite real number. Accordingly, we define the *set of admissible materials* as

$$\mathcal{E} := \{E \in L^\infty(\Omega)^{3 \times 3} \mid E = E^\top, E \succeq \underline{\rho}I, \text{Tr}(E) \leq \bar{\rho} \text{ a. e. in } \Omega\}.$$

The following result is an immediate consequence of the definition of \mathcal{E} (see [14]):

Lemma 2.1. *If $\underline{\rho} > 0$, the bilinear form*

$$a_E : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}, (w, v) \mapsto \int_{\Omega} \langle E(x)e(w(x)), e(v(x)) \rangle dx,$$

is \mathcal{V} -elliptic and bounded for all $E \in \mathcal{E}$.

Now we are able to present the *worst-case multiple multiple-load FMO problem*:

$$\inf_{\substack{u \in \mathcal{V}, \\ E \in \mathcal{E}}} \max_{k \in \mathcal{K}} \int_{\Gamma} f_k(x) \cdot u_k(x) dx \quad (4)$$

subject to

u_1, u_2, \dots, u_K solve equilibrium equations of form (3),

$$v(E) \leq \bar{v}.$$

Here the volume $v(E)$ is defined as $\int_{\Omega} \text{Tr}(E) dx$ and $\bar{v} \in \mathbb{R}$ is an upper bound on overall resources. Moreover the objective, the so called worst-case compliance functional, measures how well the structure can carry the loads $f_k, k \in \mathcal{K}$. As alternative to problem (4), one can also consider the *weighted multiple-load FMO problem*

$$\inf_{\substack{u \in \mathcal{V}, \\ E \in \mathcal{E}}} \sum_{k \in \mathcal{K}} w_k \int_{\Gamma} f_k(x) \cdot u_k(x) dx \quad (5)$$

subject to

u_1, u_2, \dots, u_K solve equilibrium equations of form (3),

$$v(E) \leq \bar{v};$$

here the values $w_k \in \mathbb{R}_+$ ($k \in \mathcal{K}$) are given weights of the associated load cases. Note that for $k = 1$ (*single load FMO problem*), both problems coincide.

The major concern of this article is to find an efficient procedure for the numerical solution of the FMO problems above. Therefore, we continue directly with the presentation of the discrete counter-parts of problems (4) and (5). For a more detailed analysis of the infinite dimensional problems the interested reader is referred to [1, 21].

The most successful approach proposed for the numerical solution of problems (4) and (5) is based on dualization and subsequent discretization, see [1, 21]. Having a lot of advantages on the one hand, this strategy turns out to have two major disadvantages:

- The computational complexity depends cubically on the number of load cases (see [12]). This makes the approach unpractical for 3D problems with more than a few (typically 3-5) load cases.

- It is difficult to apply the dual approach for problem statements extended by additional constraints on the design variable E or the state variable u (see, for example, [9, 13]). Especially in the case of non-convexity the dual formulation may become useless due to the existence of a duality gap.

Motivated by this, we propose to solve (a discretized version) of problem (4) (or alternatively (5)) directly. We define the following finite element scheme, which is based on the discretization schemes used in [1, 21]:

The design space Ω is partitioned into m elements called $\Omega_i, i = 1, \dots, m$. For simplicity, we assume that all elements are of quadrilateral type of the same size $h \in \mathbb{R}$ (otherwise we use the standard isoparametric concept; see, for instance, [5]). We approximate the matrix function $E(x)$ by a function that is constant on each element, i.e., characterized by a vector of matrices $E = (E_1, \dots, E_m)$ of its element values. Hence the discrete counterpart of the set of admissible materials in algebraic form is

$$\tilde{\mathcal{E}} = \{E \in (\mathbb{S}^3)^m \mid E_i \succeq \underline{\rho}I, \text{Tr}(E_i) \leq \bar{\rho}, i = 1, \dots, m\}. \quad (6)$$

Moreover, the discrete resource constraint takes the form

$$\sum_{i=1}^m \text{Tr}(E_i) \leq V,$$

where $V = h\bar{v}$ and \bar{v} is the upper bound on resources introduced in (5). Further we assume that the displacement vectors $u_k(x)$ ($k \in \mathcal{K}$) are approximated by continuous functions that are bilinear in each coordinate on every element. Such functions can be written as $u_k(x) = \sum_{i=1}^n u_{k,i} \vartheta_i(x)$ for all $k \in \mathcal{K}$, where $u_{k,i}$ is the value of u_k at i^{th} node and ϑ_i is a basis function with nodal interpolation property associated with i^{th} node (for details, see [5]). Now each admissible displacement function can be identified with a vector in \mathbb{R}^n , where $n = 2N - \#(\text{components of } u_k \text{ fixed by Dirichlet b. c.})$ and N is the number of nodes (vertices of the elements Ω_i) in the discrete design space.

Next we derive the discrete counter part of $a_E(\cdot, \cdot)$. Along with the family of basis functions $\vartheta_l, l = 1, \dots, n$, we define a 3×2 matrix

$$\bar{B}_j^T = \begin{pmatrix} \frac{\partial \vartheta_j}{\partial x_1} & 0 & \frac{1}{2} \frac{\partial \vartheta_j}{\partial x_2} \\ 0 & \frac{\partial \vartheta_j}{\partial x_2} & \frac{1}{2} \frac{\partial \vartheta_j}{\partial x_1} \end{pmatrix}$$

and associate with each element Ω_i a set \mathcal{D}_i of nodes belonging to this element. We use a Gauss formula for the evaluation of the integral over each element Ω_i , assume that there are nig Gauss integration points on each element and denote by $x_{i,\ell}^G$ the ℓ -th integration point associated with the i -th element. Using this, we construct block matrices $B_{i,\ell} \in \mathbb{R}^{3 \times n}$ composed of (3×2) blocks $\bar{B}_j(x_{i,\ell}^G)$ at j -th position for every $j \in \mathcal{D}_i$ and zero blocks of the same size otherwise. Then the discrete counterpart of $a_E(\cdot, \cdot)$, the *stiffness matrix* A is

$$A(E) = \sum_{i=1}^m A_i(E), \quad A_i(E) = \sum_{\ell=1}^{nig} B_{i,\ell}^T E_i B_{i,\ell}. \quad (7)$$

Finally, assuming the load functions f_k ($k \in \mathcal{K}$) to be linear on each element and identifying each such function with a vector $f_k \in \mathbb{R}^n$, the discrete compliance functionals and equilibrium conditions read as

$$f_k^\top u, \quad A(E)u_k = f_k, \quad k \in \mathcal{K}, \quad (8)$$

respectively. Using the assumption $\rho > 0$ it follows from Lemma 2.1 that $A(E)$ is strictly positive definite and we are able to eliminate u_k , $k \in \mathcal{K}$ from the equations above and to rewrite the compliance functionals as

$$c_k(E) := f_k^\top A^{-1}(E) f_k \text{ for all } k \in \mathcal{K}.$$

Thus, after discretization, problems (4) and (5) become

- *discrete worst-case multiple-load FMO problem*

$$\min_{E \in \tilde{\mathcal{E}}} \max_{k \in \mathcal{K}} f_k^\top A^{-1}(E) f_k \quad (9)$$

subject to

$$\sum_{i=1}^m \text{Tr}(E_i) \leq V,$$

- *discrete weighted multiple-load FMO problem*

$$\min_{E \in \tilde{\mathcal{E}}} \sum_{k \in \mathcal{K}} w_k f_k^\top A^{-1}(E) f_k \quad (10)$$

subject to

$$\sum_{i=1}^m \text{Tr}(E_i) \leq V.$$

Definition 2.2. Let $\bar{x}_i \in \mathbb{R}^6$ for all $i = 1, \dots, m$ and define

$$\begin{aligned} \tilde{c}_k : \quad & (\mathbb{R}^6)^m \rightarrow \mathbb{R} \\ x = (\bar{x}_1^\top, \dots, \bar{x}_m^\top) & \mapsto c_k((\text{smat}(\bar{x}_1), \dots, \text{smat}(\bar{x}_m))). \end{aligned}$$

In the following Lemma we summarize some useful properties of the compliance functionals c_k ($k \in \mathcal{K}$):

Lemma 2.3. For all $k \in \mathcal{K}$:

- c_k is well-defined, infinitely often continuously differentiable and convex on $\tilde{\mathcal{E}}$. Moreover the formula

$$\frac{\partial}{\partial E_i} c_k(E) = - \sum_{\ell=1}^{nig} B_{i,\ell}^T u_k(E) u_k(E)^\top B_{i,\ell}^T, \quad u_k(E) := A^{-1}(E) f_k$$

holds true for all partial derivatives of c_k and $\frac{\partial}{\partial E_i} c_k(E)$ is negative semidefinite for all $i = 1, 2, \dots, m$ and all $E \in \tilde{\mathcal{E}}$.

- \tilde{c}_k is infinitely often continuously differentiable and convex on

$$\mathcal{X} := \left\{ (\bar{x}_1, \dots, \bar{x}_m) \mid (\text{smat}(\bar{x}_1), \dots, \text{smat}(\bar{x}_m)) \in \tilde{\mathcal{E}} \right\}.$$

- The Hessian of \tilde{c}_k is dense.

Proof. We start with the proof of assertion b): The global stiffness matrix $A(E)$ can be written as linear operator of the form

$$A(E_1, \dots, E_m) = \sum_{i=1}^m \sum_{j=1}^6 (\bar{x}_i)_j A_i^{p(j), q(j)},$$

where $\bar{x}_i := \text{svec}(E_i)$, $p(j)$ and $q(j)$ are the row and column indices of the element $(\bar{x}_i)_j$ in lower triangular part of the matrix E_i and the matrices $A_i^{p(j), q(j)}$ are defined as

$$A_i^{p(j), q(j)} = \begin{cases} \sum_{\ell=1}^{nig} (B_{i,\ell})_{p(j)} (B_{i,\ell})_{q(j)}^\top, & p(j) = q(j), \\ \sum_{\ell=1}^{nig} (B_{i,\ell})_{p(j)} (B_{i,\ell})_{q(j)}^\top + (B_{i,\ell})_{q(j)} (B_{i,\ell})_{p(j)}^\top, & p(j) \neq q(j), \end{cases} \quad (11)$$

with $(B_{i,\ell})_j$ denoting the j -th column of $B_{i,\ell}$. Thus the mapping

$$\tilde{A} : (\mathbb{R}^6)^m \rightarrow \mathbb{S}^n, \quad x \mapsto A(\text{smat}(\bar{x}_1), \dots, \text{smat}(\bar{x}_m))$$

is linear in $x := (\bar{x}_1, \dots, \bar{x}_m)$. Moreover it follows from Lemma 2.1 that $\tilde{A}(x)$ is positive definite for all $x \in \mathcal{X}$. Taking into account that the mapping $A \mapsto A^{-1}$ is convex (infinitely often continuously differentiable) on \mathbb{S}_+^n (see [8]) and writing \tilde{c}_k as

$$\tilde{c}_k(x) = \langle \tilde{A}^{-1}(x), f_k f_k^\top \rangle_{\mathbb{S}^n},$$

the assertion of part b) follows from the fact that \tilde{c}_k is the composition of a linear, a convex (infinitely often continuously differentiable) and a linear function.

Next we prove a): The well-definedness follows directly from Lemma 2.1. The differentiability and convexity follow from assertion b). Using the matrices $A_i^{j,p}$ defined in (11) to rewrite $A_i(E) = \sum_{j,p=1}^3 (E_i)_{j,p} A_i^{j,p}$ we obtain

$$\frac{\partial}{\partial (E_i)_{j,p}} c_k(E) = -u_k(E)^\top \left(\frac{\partial}{\partial (E)_{j,p}} A(E) \right) u_k(E) = -u_k(E)^\top A_i^{j,p} u_k(E),$$

where $u_k(E) = A^{-1}(E) f_k$ and $(E_i)_{j,p}$ denotes the matrix entry of E_i in row j and column p . Using the definition of $A_i^{j,p}$ we can write the latter expression as $\sum_{\ell=1}^{nig} (B_{i,\ell})_j u_k(E) u_k(E)^\top (B_{i,\ell})_p^\top$. Now the negative semidefiniteness follows immediately from the dyadic structure of $u_k(E) u_k(E)^\top$.

Finally we show assertion c): Using the same arguments as the proof of part a), we obtain

$$\frac{\partial^2}{\partial (x_i)_j \partial (x_p)_q} \tilde{c}_k(x) = 2u_k(E)^\top A_i^j A^{-1}(E) A_p^q u_k(E),$$

which is in general a non-zero value. \square

The following corollary is a direct consequence of Lemma 2.3.

Corollary 2.4. *Problems (9) and (10) are convex semidefinite programming problems.*

Remark 2.5. *As a consequence of Corollary 2.4, one could try to apply an existing non-linear (convex) SDP solver directly in order to solve (9) or (10). This is however not recommendable in practice with any SDP solver using explicit second order derivatives due to Lemma 2.3 c). Experiments with an SDP solver, avoiding the calculation and storage of explicit second order derivatives by Krylov-type methods (see [11]) led to moderate success, as the effort for the approximation of $\nabla^2 \tilde{c}_k$ turned out to be too expensive. For this reason we decided to develop an SDP solver, which is solely based on first order information.*

3 A block-separable convex approximation scheme

In this section we want to define block-separable convex approximations of continuously differentiable functions

$$f : \mathbb{S} \rightarrow \mathbb{R}, \text{ where } \mathbb{S} = \mathbb{S}^{d_1} \times \mathbb{S}^{d_2} \times \dots \times \mathbb{S}^{d_m} \text{ and } (d_1, d_2, \dots, d_m) \in \mathbb{N}^m. \quad (12)$$

Let $\mathcal{I} = \{1, 2, \dots, m\}$. On \mathbb{S} we define the scalar product $\langle \cdot, \cdot \rangle_{\mathbb{S}} := \sum_{i \in \mathcal{I}} \langle \cdot, \cdot \rangle_{\mathbb{S}^{d_i}}$, where $\langle \cdot, \cdot \rangle_{\mathbb{S}^{d_i}}$ is the standard scalar product in \mathbb{S}^{d_i} ($i \in \mathcal{I}$). Moreover, we denote by $\| \cdot \|_{\mathbb{S}}$ the norm induced by $\langle \cdot, \cdot \rangle_{\mathbb{S}}$. Finally, we denote directional derivatives of f of first and second order in directions $V, W \in \mathbb{S}$ by $\frac{\partial}{\partial V} f(Y; V)$ and $\frac{\partial^2}{\partial V \partial W} f(Y; V, W)$, respectively.

Definition 3.1. We call an approximation $g : \mathbb{S} \rightarrow \mathbb{R}$ of a function f of type (12) a convex first order approximation at $\bar{Y} = (\bar{Y}_1, \dots, \bar{Y}_m) \in \mathbb{S}$, if the following assumptions are satisfied:

$$(A1) \quad g(\bar{Y}) = f(\bar{Y}),$$

$$(A2) \quad \frac{\partial}{\partial Y_i} g(\bar{Y}) = \frac{\partial}{\partial Y_i} f(\bar{Y}) \text{ for all } i \in \mathcal{I},$$

$$(A3) \quad g \text{ is convex.}$$

In the following we want to construct a local block-separable *convex first order approximation* scheme for functions of type f . Our construction can be considered a generalization of the classic MMA-type approximations defined, for example, in [20, 23].

We start with the following definitions:

Definition 3.2. Let $f : \mathbb{S} \rightarrow \mathbb{R}$ be continuously differentiable on a subset $B \subset \mathbb{S}$. For all $i \in \mathcal{I}$ we define differential operators entry-wise by

$$(\nabla^i f)_{\ell, j} := \left(\frac{\partial f}{\partial Y_i} \right)_{\ell, j}$$

and denote by $\nabla_+^i f(\bar{Y})$ and $\nabla_-^i f(\bar{Y})$ the projections of $\nabla^i f(\bar{Y})$ onto $\mathbb{S}_+^{d_i}$ and $\mathbb{S}_-^{d_i}$, respectively.

Definition 3.3. Let $f : \mathbb{S} \rightarrow \mathbb{R}$ be continuously differentiable on a subset $B \subset \mathbb{S}$ and $\bar{Y} = (\bar{Y}_1, \bar{Y}_2, \dots, \bar{Y}_m) \in B$. Moreover let asymptotes $L = (L_1, L_2, \dots, L_m)^\top, U = (U_1, U_2, \dots, U_m)^\top$ be given such that

$$L_i \prec_{\mathbb{S}_+^{d_i}} \bar{Y}_i \prec_{\mathbb{S}_+^{d_i}} U_i \text{ for all } i \in \mathcal{I}$$

and $\tau := \{\tau_1, \tau_2, \dots, \tau_m\}$ be a set of non-negative real parameters. Then we define the hyperbolic approximation $f_{\bar{Y}}^{L, U, \tau}$ of f at \bar{Y} as

$$\begin{aligned} f_{\bar{Y}}^{L, U, \tau}(Y) &:= f(\bar{Y}) + \\ &\sum_{i=1}^m \langle \nabla_+^i f(\bar{Y}), (U_i - \bar{Y}_i)(U_i - Y_i)^{-1}(U_i - \bar{Y}_i) - (U_i - \bar{Y}_i) \rangle_{\mathbb{S}^{d_i}} - \\ &\sum_{i=1}^m \langle \nabla_-^i f(\bar{Y}), (\bar{Y}_i - L_i)(Y_i - L_i)^{-1}(\bar{Y}_i - L_i) - (\bar{Y}_i - L_i) \rangle_{\mathbb{S}^{d_i}} + \\ &\sum_{i=1}^m \tau_i \langle (Y_i - \bar{Y}_i)^2, (U_i - Y_i)^{-1} + (Y_i - L_i)^{-1} \rangle_{\mathbb{S}^{d_i}}. \end{aligned} \quad (13)$$

The following theorem says that (13) is a convex approximation in the sense of Definition 3.1.

Theorem 3.4.

- a) $f_Y^{L,U,\tau}$ satisfies assumptions (A1) to (A3).
- b) $f_Y^{L,U,\tau}$ is separable w.r.t. the matrix variables Y_1, Y_2, \dots, Y_m .
- c) Let B be a compact subset of \mathbb{S} , $\bar{\tau} \geq \tau_i \geq \underline{\tau} > 0$ for all $i \in \mathcal{I}$, and compact sets of asymptotes \mathcal{L} and \mathcal{U} satisfy the following condition:

$$(AS) \quad \forall \left\{ \begin{array}{l} L \in \mathcal{L} \\ U \in \mathcal{U} \end{array} \right\} \exists \mu > 0 : \forall i \in \mathcal{I} \forall Y \in B : \left\{ \begin{array}{l} Y_i - L_i \\ U_i - Y_i \end{array} \right\} \succeq \mu I_{\mathbb{S}^{d_i}}.$$

Then $f_Y^{L,U,\tau}$ is strongly convex on B and there is a common constant $\underline{\nu} > 0$ such that for all $L \in \mathcal{L}, U \in \mathcal{U}$ and $\bar{Y} \in B$

$$\frac{\partial}{\partial Y} f_Y^{L,U,\tau}(Y; X - Y) + \underline{\nu} \|X - Y\| \leq f_Y^{L,U,\tau}(X) - f_Y^{L,U,\tau}(Y),$$

for all $X, Y \in B$. Moreover the second order derivative of $f_Y^{L,U,\tau}$ is uniformly bounded for all $L \in \mathcal{L}, U \in \mathcal{U}$ and $\bar{Y} \in B$ in the sense that there is a constant $\bar{\nu} > 0$ such that

$$\frac{\partial^2}{\partial Y \partial Y} f_Y^{L,U,\tau}(Y; D, D) \leq \bar{\nu} \|D\|^2$$

for all $Y \in B$ and all $D \in \mathbb{S}$.

Proof. (A1) For $Y := \bar{Y}$ we have for all $i \in \mathcal{I}$

$$\begin{aligned} \langle \nabla_+^i f(\bar{Y}), (U_i - \bar{Y}_i)(U_i - \bar{Y}_i)^{-1}(U_i - \bar{Y}_i) - (U_i - \bar{Y}_i) \rangle_{\mathbb{S}^{d_i}} &= \\ \langle \nabla_+^i f(\bar{Y}), (U_i - \bar{Y}_i) - (U_i - \bar{Y}_i) \rangle_{\mathbb{S}^{d_i}} &= 0. \end{aligned}$$

Consequently, the first sum in (13) vanishes. Analogously we show that the second sum vanishes and with $\langle (\bar{Y}_i - \bar{Y}_i)^2, (U_i - Y_i)^{-1} + (Y_i - L_i)^{-1} \rangle_{\mathbb{S}^{d_i}} = 0$, we conclude $f_Y^{L,U,\tau}(\bar{Y}) = f(\bar{Y})$.

(A2) Differentiating $f_Y^{L,U,\tau}$ w.r.t. Y_i we obtain

$$\begin{aligned} \frac{\partial}{\partial Y_i} f_Y^{L,U,\tau}(Y) &= (U_i - Y_i)^{-1}(U_i - \bar{Y}_i) \nabla_+^i f(\bar{Y})(U_i - \bar{Y}_i)(U_i - Y_i)^{-1} + \\ &\quad (Y_i - L_i)^{-1}(\bar{Y}_i - L_i) \nabla_-^i f(\bar{Y})(\bar{Y}_i - L_i)(Y_i - L_i)^{-1} + \\ &\quad \tau_i (I_{\mathbb{S}^{d_i}} - (U_i - Y_i)^{-1}(U_i - \bar{Y}_i)^2(U_i - Y_i)^{-1}) + \\ &\quad \tau_i (I_{\mathbb{S}^{d_i}} - (Y_i - L_i)^{-1}(\bar{Y}_i - L_i)^2(Y_i - L_i)^{-1}). \end{aligned} \quad (14)$$

Substituting $Y := \bar{Y}$ in (14), we obtain

$$\frac{\partial}{\partial Y_i} f_Y^{L,U,\tau}(\bar{Y}) = \nabla_+^i f(\bar{Y}) + \nabla_-^i f(\bar{Y}) = \nabla^i f(\bar{Y}) = \frac{\partial}{\partial Y_i} f(\bar{Y}).$$

(A3) Before we show convexity, we want to proof separability (assertion b)), but this follows immediately from (14), as

$$\frac{\partial}{\partial Y_j} \left(\frac{\partial}{\partial Y_i} f_Y^{L,U,\tau}(Y) \right) = 0.$$

Now, in order to prove convexity, it is sufficient to show that for all $i \in \mathcal{I}$, for all $Y \in \mathbb{S}$ and an arbitrary direction $D \in \mathbb{S}^{d_i}$

$$\frac{\partial^2}{\partial Y_i \partial Y_i} f_Y^{L,U,\tau}(Y; D, D) = \left\langle \nabla^i \left\langle \nabla^i f_Y^{L,U,\tau}(Y), D \right\rangle_{\mathbb{S}^{d_i}}, D \right\rangle_{\mathbb{S}^{d_i}} \geq 0. \quad (15)$$

Introducing the abbreviations $B_i^+ := (U_i - \bar{Y}_i) \nabla_+^i f(\bar{Y})(U_i - \bar{Y}_i)$ and $B_i^- := (\bar{Y}_i - L_i) \nabla_-^i f(\bar{Y})(\bar{Y}_i - L_i)$ it follows from (14) that

$$\begin{aligned} \left\langle \nabla^i \left\langle \nabla^i f_Y^{L,U,\tau}(Y), D \right\rangle_{\mathbb{S}^{d_i}}, D \right\rangle_{\mathbb{S}^{d_i}} = & \\ & 2 \left\langle D(U_i - Y_i)^{-1} D, (U_i - Y_i)^{-1} B_i^+ (U_i - Y_i)^{-1} \right\rangle_{\mathbb{S}^{d_i}} + \\ & 2 \left\langle D(Y_i - L_i)^{-1} D, (Y_i - L_i)^{-1} (-B_i^-) (Y_i - L_i)^{-1} \right\rangle_{\mathbb{S}^{d_i}} + \\ & 2\tau_i \left\langle D(U_i - Y_i)^{-1} D, (U_i - Y_i)^{-1} (U_i - \bar{Y}_i)^2 (U_i - Y_i)^{-1} \right\rangle_{\mathbb{S}^{d_i}} + \\ & 2\tau_i \left\langle D(Y_i - L_i)^{-1} D, (Y_i - L_i)^{-1} (\bar{Y}_i - L_i)^2 (Y_i - L_i)^{-1} \right\rangle_{\mathbb{S}^{d_i}}. \quad (16) \end{aligned}$$

Given that the matrices $B_i^+, -B_i^-, U_i - Y_i, Y_i - L_i, (U_i - \bar{Y}_i)^2, (\bar{Y}_i - L_i)^2$ are all positive semidefinite, we observe that all terms in (16) are nonnegative. Consequently the estimate (15) holds true and $f_Y^{L,U,\tau}$ is convex. Finally, given that (A5) holds for the compact sets \mathcal{L} and \mathcal{U} we use (16) to show

$$\begin{aligned} & \left\langle \nabla^i \left\langle \nabla^i f_Y^{L,U,\tau}(Y), D \right\rangle_{\mathbb{S}^{d_i}}, D \right\rangle_{\mathbb{S}^{d_i}} \\ & \geq 2\tau_i \left\langle D(U_i - Y_i)^{-1} D, (U_i - Y_i)^{-1} (U_i - \bar{Y}_i)^2 (U_i - Y_i)^{-1} \right\rangle_{\mathbb{S}^{d_i}} + \\ & \quad 2\tau_i \left\langle D(Y_i - L_i)^{-1} D, (Y_i - L_i)^{-1} (\bar{Y}_i - L_i)^2 (Y_i - L_i)^{-1} \right\rangle_{\mathbb{S}^{d_i}} \\ & \geq 4\tau_i \gamma^3 \mu^2 \langle D, D \rangle_{\mathbb{S}^{d_i}}, \end{aligned}$$

where γ is an upper bound for the maximal possible difference of eigenvalues of arbitrary elements in the compact sets B and \mathcal{L} or B and \mathcal{U} , respectively. Now the assertion of part c) follows with $\underline{\nu} := m4\tau\gamma^3\mu^2$. The uniform boundedness of the second order derivatives follows in an analogous way from (16) with $\bar{\nu} := m\gamma^2\mu^3 \left(2 \max_{i \in \mathcal{I}, \bar{Y} \in B} \|\nabla^i f(\bar{Y})\| + 4\bar{\tau} \right)$. \square

4 A globally convergent algorithm based on hyperbolic approximations

In the framework of this section we want to use the local hyperbolic approximations defined Section 3 in order to establish a solution scheme for the following generic optimization problem:

$$\begin{aligned}
& \min_{Y \in \mathbb{S}} f(Y) & (\mathcal{P}) \\
& \text{subject to} \\
& g_k(Y) \leq 0, \quad k = 1, 2, \dots, K, \\
& \underline{Y}_i \preceq_{\mathbb{S}^{d_i}} Y_i \preceq_{\mathbb{S}^{d_i}} \overline{Y}_i, \quad i = 1, 2, \dots, m;
\end{aligned}$$

here \mathbb{S} is defined as in (12). In what follows \mathcal{F} denotes the feasible domain of problem (\mathcal{P}) . Throughout this section we make the following assumptions:

(A1) $f : \mathbb{S} \rightarrow \mathbb{R}$ is convex. Moreover f is the maximum over a finite set of twice continuously differentiable functions. Therefore we may write

$$f(Y) = \max_{\ell \in \mathcal{I}_{\max}} f_\ell(Y)$$

for some index set \mathcal{I}_{\max} .

(A2) the functions $g_k : \mathbb{S} \rightarrow \mathbb{R}$ ($k = 1, 2, \dots, K$) are continuously differentiable and convex, such that \mathcal{F} is convex.

(A3) The interior of \mathcal{F} is non-empty.

(A4) The compact sets \mathcal{L} and \mathcal{U} satisfy property (AS) for the compact set \mathcal{F} .

Given an iteration index j and an associated feasible point Y^j of problem (\mathcal{P}) , we define a local hyperbolic approximation of f as

$$f^j(Y) := \max_{\ell \in \mathcal{I}_{\max}} f_\ell^j(Y) := \max_{\ell \in \mathcal{I}_{\max}} (f_\ell)_{Y^j}^{L^j, U^j, \tau^j}(Y).$$

Using this function, we further define a local approximation of (\mathcal{P}) close to Y^j as follows:

$$\begin{aligned}
& \min_{Y \in \mathbb{S}} f^j(Y) & (\mathcal{P}^j) \\
& \text{subject to} \\
& g_k(Y) \leq 0, \quad k = 1, 2, \dots, K, \\
& \underline{Y}_i^j \preceq_{\mathbb{S}^{d_i}} Y_i \preceq_{\mathbb{S}^{d_i}} \overline{Y}_i^j, \quad i = 1, 2, \dots, m.
\end{aligned}$$

Here the bounds $\underline{Y}_i^j, \overline{Y}_i^j$ are chosen to be compatible with the following assumption:

(A5) $\underline{Y}_i \preceq_{\mathbb{S}^{d_i}} \underline{Y}_i^j \preceq_{\mathbb{S}^{d_i}} Y_i^j \preceq_{\mathbb{S}^{d_i}} \overline{Y}_i^j \preceq_{\mathbb{S}^{d_i}} \overline{Y}_i$ for all $i = 1, 2, \dots, m$.

We denote the feasible domain of problem (\mathcal{P}^j) by \mathcal{F}^j . By construction the following corollary is an immediate consequence of Theorem 3.4:

Corollary 4.1.

- a) $f^j(Y^j) = f(Y^j)$.
- b) The subdifferentials of f^j and f coincide at Y^j , i. e. $\partial f^j(Y^j) = \partial f(Y^j)$.
- c) f^j is convex.

d) f^j is separable w.r.t. Y_1, Y_2, \dots, Y_m .

e) Let $\bar{\tau} \geq \tau_1, \tau_2, \dots, \tau_m \geq \underline{\tau} > 0$. Let further compact sets of asymptotes \mathcal{L} and \mathcal{U} be given, such that property (AS) holds for \mathcal{F}^j . Then f^j is strongly convex on \mathcal{F}^j . Moreover there is a common constant $\underline{\nu} > 0$ such that for all j

$$\frac{\partial}{\partial Y} f^j(Y; X - Y) + \underline{\nu} \|X - Y\| \leq f^j(X) - f^j(Y),$$

for all $X, Y \in \mathcal{F}^j$.

Proof. All assertions but the last one follow directly from Theorem 3.4. It remains to show that \mathcal{F}^j is compact and convex. The convexity follows directly from assumption (A2). Moreover each domain \mathcal{F}^j is compact as a closed subset of the domain $\{Y \in \mathbb{S} \mid \underline{Y}_i \preceq_{\mathbb{S}^{d_i}} Y_i \preceq_{\mathbb{S}^{d_i}} \bar{Y}_i, i = 1, 2, \dots, m\}$. \square

The following Proposition states some basic properties of (\mathcal{P}^j) .

Proposition 4.2. *Each sub-problem (\mathcal{P}^j) has a unique solution Y^{j+1} . Associated with Y^{j+1} there exist Lagrangian multipliers (v^{j+1}, V^{j+1}) such that $(Y^{j+1}, v^{j+1}, V^{j+1})$ is a KKT-point of (\mathcal{P}^j) .*

Proof. The existence and the uniqueness of a solution follows from the strong convexity of the objective function f^j on the compact set \mathcal{F}^j . Furthermore, assumption (A3) implies that the Slater condition holds for (\mathcal{P}^j) . Consequently a KKT point exists. \square

Now we are able to present the basic algorithm for the solution of (\mathcal{P}) :

Algorithm 4.3. *Let initial points $Y^1 \in \mathbb{S}$ and initial multipliers $(v^1, V^1) \in \mathbb{R}_+^k \times \mathbb{S}_+$ be given.*

- (1) Put $j = 1$.
- (2) Choose asymptotes $L^j \in \mathcal{L}, U^j \in \mathcal{U}$ and $\bar{\tau} \geq \tau_1^j, \tau_2^j, \dots, \tau_m^j \geq \underline{\tau} > 0$.
- (3) Solve problem (\mathcal{P}^j) . Denote the solution by $Y^+ \in \mathbb{S}$ and the associated Lagrangian multipliers by $(v^+, V^+) \in \mathbb{R}_+^k \times \mathbb{S}_+$.
- (4) Choose $\alpha^j = \min\{1, \hat{\alpha}\}$, where $\hat{\alpha} = \min_{\alpha \in \mathbb{R}_+} f(Y^j + \alpha(Y^+ - Y^j))$.
- (5) $(Y^{j+1}, v^{j+1}, V^{j+1}) = (Y^j, v^j, V^j) + \alpha^j ((Y^+, v^+, V^+) - (Y^j, v^j, V^j))$.
- (6) If Y^{j+1} is stationary for problem (\mathcal{P}) , STOP; otherwise put $j = j + 1$ and GOTO (2).

Possible choices of asymptotes (step 2) will be discussed in Section 6. In Section 5 we will propose an algorithm for the efficient solution of the sub-problem in step 3. Practical implementations of the line search in step 4, as well as the stopping criterion in step 6 will be given in section 6. Before we state the central convergence result for Algorithm 4.3, we make one more assumption:

(A6) The multiplier estimates generated by Algorithm 4.3 stay bounded.

Note that in [22] the assertion of assumption (A6) is proven to hold for standard inequality constrained nonlinear programs under the LICQ condition.

Theorem 4.4. Assume that (A1) to (A6) are satisfied. Then, either Algorithm 4.3 stops at a global minimizer of (\mathcal{P}) , or the sequence $\{Y^j\}_{j>0}$ generated by Algorithm 4.3 has at least one accumulation point and each accumulation point is a global minimizer of (\mathcal{P}) .

In order to be able to prove the convergence theorem, we make use of the following Lemmas:

Lemma 4.5.

Let $\mathcal{A}_{f^{(j)}}(Y) := \{\ell \mid f_\ell^{(j)}(Y) = f^{(j)}(Y)\} \subset \mathcal{I}_{\max}$. Then

$$\partial f^{(j)}(Y) = \text{conv}\{\nabla f_\ell^{(j)}(Y) \mid \ell \in \mathcal{A}(Y)\}, \quad (17)$$

where $\partial f^{(j)}(Y)$ denotes the subdifferential of $f^{(j)}$ at Y and conv is the convex hull.

Proof. Formula (17) is a direct consequence of Corollary 4.3.2 in [7]. \square

Lemma 4.6. If $Y^j \in \mathcal{F}^j$ is not stationary for (\mathcal{P}) , then the direction $D^j := Y^+ - Y^j$ is a descent direction for f at Y^j .

Proof. Using the fact that Y^+ is a unique minimizer of (\mathcal{P}^j) , we obtain from Corollary 4.1 e):

$$\frac{\partial}{\partial Y} f^j(Y^j; D^j) + \nu \|D^j\| \leq f(Y^+) - f(Y^j) \leq 0. \quad (18)$$

Consequently we obtain from the first order approximation properties of f^j :

$$\frac{\partial}{\partial Y} f(Y^j; D^j) < 0.$$

\square

Lemma 4.7. Algorithm 4.3 generates a sequence of feasible points Y^1, Y^2, \dots with

$$f(Y^{j+1}) \leq f(Y^j).$$

Proof. The convexity of \mathcal{F}^j and the fact that $\mathcal{F}^j \subset \mathcal{F}$ imply that all iterates remain feasible. Let us now consider a subproblem at an arbitrary iteration j . Then we have $f(Y^j) = f^j(Y^j)$. From assumption (A5) we know that Y^j is a feasible point of problem (\mathcal{P}^j) . Consequently it follows from Lemma 4.6 that $D^j := Y^+ - Y^j$ is a descent direction for f at Y^j . Now the assertion follows by construction of the line search defined in step 4 of Algorithm 4.3. \square

Lemma 4.8. Let $Y^* \in \mathcal{F}$ be an accumulation point of the sequence generated by Algorithm 4.3 applied to (\mathcal{P}) . Then Y^* is an unconstrained minimizer of f or the line search in step 4 of Algorithm 4.3 returns the result $\alpha^j = 1$ for almost all $j > 0$.

Proof. Given an arbitrary element $g^j \in \partial f(Y^j)$ with $\langle -g^j, D^j \rangle = \frac{\partial}{\partial Y} f(Y^j; D^j)$ it follows from (18) that

$$\frac{\langle -g^j, D^j \rangle_{\mathbb{S}}}{\|D^j\|_{\mathbb{S}}} \geq \nu > 0. \quad (19)$$

From (19) and the fact that $\|g^j\|_{\mathbb{S}}$ is bounded on the compact set \mathcal{F} , we obtain the existence of $\gamma > 0$ such that

$$\frac{\langle -g^j, D^j \rangle_{\mathbb{S}}}{\|D^j\|_{\mathbb{S}} \|g^j\|_{\mathbb{S}}} \geq \gamma.$$

Consequently the cosine of the angle between $\partial f(Y^j)$ and the descent direction D^j is bounded away from 0. Suppose now that Algorithm 4.3 generates infinitely many iterations j_s with $\min_{\alpha \in \mathbb{R}_+} f(Y^{j_s} + \alpha Y^+) \in [0; 1]$. Then it follows from the Theorem of Zoutendijk ([16, Thm.3.2]) that $\|g_j\|_{\mathbb{S}} \rightarrow 0$. Consequently we obtain $0 \in \partial f(Y^*)$ and conclude that $Y^* \in \mathcal{F}$ is a minimizer of the convex function f . \square

Now we are able to finish the proof of Theorem 4.4:

Proof. Suppose that Algorithm 4.3 does not stop at a stationary point. Then, according to Lemma 4.7 it generates an infinite sequence with $\{Y^k\}_{k>0}$ in \mathcal{F} such that $\{f(Y^k)\}_{k>0}$ is monotonically decreasing. As f is bounded from below on the compact set \mathcal{F} , the sequence $\{f(Y^k)\}_{k>0}$ converges. Consequently there is at least one accumulation point Y^* of the sequence $\{Y^k\}_{k>0}$.

Next we want to show that Y^* is a first-order critical point and thus a global optimizer for problem (\mathcal{P}) :

Proposition 4.2 guaranties that step 3 of Algorithm 4.3 is well-defined. Moreover, from Lemma 4.8 we have after finitely many iterations

$$(Y^+, v^+, V^+) = (Y^{j+1}, v^{j+1}, V^{j+1}).$$

Lemma 4.7 and the fact that (Y^+, v^+, V^+) is a KKT-point of problem (\mathcal{P}^j) imply the existence of an index $\bar{j} > 0$ and a subsequence $\{Y^{j_s}\}_{j_s > \bar{j}}$ such that:

- $Y^{j_s} \in \mathcal{F}$,
- $v^{j_s} \geq 0, V^{j_s} \succeq 0$,
- $g(Y^{j_s})^\top v^{j_s} = 0$,
- $\langle Y_i^{j_s} - \underline{Y}_i, V_i^{j_s} \rangle = 0, \langle \bar{Y}_i - Y_i^{j_s}, V_{i+m}^{j_s} \rangle = 0$ for all $i = 1, 2, \dots, m$.

Consequently Y^* is feasible. Moreover, making use of assumption (A6) we conclude that there exist non-negative multipliers (v^*, V^*) such that the triple (Y^*, v^*, V^*) satisfies the complementary slackness condition for (\mathcal{P}) . In order to complete the proof, we have to show that

$$\text{dist}(\{0\}, \partial_Y L(Y^{j_s}, v^{j_s}, V^{j_s})) \rightarrow 0 \quad (20)$$

for $j_s \rightarrow \infty$, where L denotes the Lagrangian function associated with problem (\mathcal{P}) and

$$\text{dist}(A, B) := \min_{X \in A, Y \in B} \|X - Y\|_{\mathbb{S}}$$

for two sets $A, B \subset \mathbb{S}$. Denoting by L^{j_s} the Lagrangian function associated with problem (\mathcal{P}^{j_s}) the following estimate holds true:

$$\begin{aligned} & \text{dist}(0, \partial_Y L(Y^{j_s}, v^{j_s}, V^{j_s})) \\ &= \text{dist}(0, \partial_Y f^{j_s}(Y^{j_s}) - \partial_Y f^{j_s}(Y^{j_s}) + \partial_Y L(Y^{j_s}, v^{j_s}, V^{j_s})) \\ &\leq \text{dist}(0, \partial_Y f(Y^{j_s}) - \partial_Y f^{j_s}(Y^{j_s})) + \text{dist}(0, \partial_Y L^{j_s}(Y^{j_s}, v^{j_s}, V^{j_s})) \\ &= \text{dist}(\partial_Y f^{j_s}(Y^{j_s}), \partial_Y f(Y^{j_s})) \\ &\leq \text{dist}(\partial_Y f^{j_s}(Y^{j_s-1}), \partial_Y f^{j_s}(Y^{j_s})) + \text{dist}(\partial_Y f(Y^{j_s-1}), \partial_Y f(Y^{j_s})). \end{aligned} \quad (21)$$

From the continuity of f we see (cmp., for example, [7])

$$\text{dist} \left(\partial_Y f(Y^{j_s-1}), \partial_Y f(Y^{j_s}) \right) \rightarrow 0$$

for $j_s \rightarrow \infty$. Moreover we conclude from the uniform boundedness of the second order directional derivatives stated in Theorem 3.4 c) for all $\ell \in \mathcal{I}_{\max}$:

$$\|\nabla f_\ell^{j_s}(Y^{j-1}) - \nabla f_\ell^{j_s}(Y^j)\| \rightarrow 0$$

for $j_s \rightarrow \infty$. Now, applying Lemma 4.5, we obtain

$$\text{dist} \left(\partial_Y f^{j_s}(Y^{j_s-1}), \partial_Y f^{j_s}(Y^{j_s}) \right) \rightarrow 0$$

for $j_s \rightarrow \infty$. Consequently (20) holds, Y^* is a first-order critical point and the proof of Theorem 4.4 is complete. \square

5 A modified barrier algorithm for the solution of sub-problems

In order to solve the sub-problems defined in Section 4 numerically, we rewrite them in terms of real variables $x = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m)^\top \in \mathbb{R}^{\bar{d}_1} \times \mathbb{R}^{\bar{d}_2} \times \dots \times \mathbb{R}^{\bar{d}_m} = \mathbb{R}^{\bar{d}}$. This yields the problem:

$$\min_{x \in \mathbb{R}^{\bar{d}}} \max_{\ell \in \mathcal{I}_{\max}} \tilde{f}_{\text{smat}(\bar{x}), \ell}^{L, U, \tau}(x) \quad (22)$$

subject to

$$g_k(\text{smat}(x)) \leq 0, \quad k \in \{1, 2, \dots, K\},$$

$$\underline{Y}_i \preceq_{\mathbb{S}^{d_i}} \text{smat}(\bar{x}_i) \preceq_{\mathbb{S}^{d_i}} \bar{Y}_i, \quad i \in \mathcal{I}.$$

In the scope of this section we work with the following additional assumption:

(A7) the functions $g_k : \mathbb{R}^{\bar{d}} \rightarrow \mathbb{R}$ ($k \in \{1, 2, \dots, K\}$) are separable w.r.t. to $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m$.

The algorithm used to solve sub-problems of the form (22) is based on a generalized augmented Lagrangian method for the solution of nonlinear (semidefinite) programs and described in detail in [10, 18]. We briefly recall the basic here and show how this algorithm can be adapted to take care of the special structure of problem (22).

The algorithm described in [10, 18] is designed for the solution of general nonlinear semidefinite optimization problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) \quad (23)$$

subject to

$$\mathcal{G}_j(x) \preceq 0, \quad j \in \mathcal{J} = \{1, 2, \dots, J\};$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and $\mathcal{G}_j(x) : \mathbb{R}^n \rightarrow \mathbb{S}^{m_j}$ ($j \in \mathcal{J}$) are twice continuously differentiable mappings. The algorithm is based on a choice of smooth modified barrier functions $\Phi_p : \mathbb{S}^{m_j} \rightarrow \mathbb{S}^{m_j}$ ($j \in \mathcal{J}$), depending on a parameter $p > 0$, that satisfies a number of assumptions (see [10]) guaranteeing, in particular, that

$$\mathcal{G}_j(x) \preceq 0 \Leftrightarrow \Phi_p(\mathcal{G}_j(x)) \preceq 0 \quad \forall j \in \mathcal{J}.$$

Thus, for any $p > 0$, problem (23) has the same solution as the following “augmented” problem

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ & \text{subject to} \\ & \Phi_p(\mathcal{G}_j(x)) \preceq 0, \quad j \in \mathcal{J}. \end{aligned} \quad (24)$$

A typical choice of Φ_p is

$$\Phi_p(\mathcal{A}(x)) = -p^2(\mathcal{A}(x) - pI)^{-1} - pI. \quad (25)$$

The Lagrangian of (24) can be viewed as a (generalized) augmented Lagrangian of (23):

$$F(x, U, p) = f(x) + \sum_{j \in \mathcal{J}} \langle U_j, \Phi_p(\mathcal{G}_j(x)) \rangle_{\mathbb{S}_m}; \quad (26)$$

here $U = (U_1, U_2, \dots, U_J)^\top \in \mathbb{S}^{m_1} \times \mathbb{S}^{m_2} \times \dots \times \mathbb{S}^{m_J}$ are Lagrangian multipliers associated with the inequality constraints. Defining $\mathcal{G} := (\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_J)^\top$ and $\Phi_p \mathcal{G} := (\Phi_p(\mathcal{G}_1), \Phi_p(\mathcal{G}_2), \dots, \Phi_p(\mathcal{G}_J))^\top$, the augmented Lagrangian algorithm is defined as follows:

Algorithm 5.1. *Let x^1 and U^1 be given. Let $p^1 > 0, \alpha^1 > 0$. For $k = 1, 2, \dots$ repeat until a stopping criterion is reached:*

- (1) Find $x^{\ell+1}$ satisfying $\|\nabla_x F(x^{\ell+1}, U^\ell, p^\ell)\| \leq \alpha^\ell$
- (2) $U^{\ell+1} = D_{\mathcal{G}} \Phi_p(\mathcal{G}(x^{\ell+1}); U^\ell)$
- (3) $p^{\ell+1} \leq p^\ell, \quad \alpha^{\ell+1} < \alpha^\ell$.

The unconstrained minimization problem in step (1) is approximately solved by the damped Newton method. Multiplier and penalty update strategies, as well as local and global convergence properties under standard assumptions are studied extensively in [18]. Let us only mention here that, imposing standard assumptions, one can prove that any cluster point of the sequence $\{(x^\ell, U^\ell)\}_{\ell \geq 0}$ generated by Algorithm 5.1 is a KKT point of problem (23). The proof given in [18] is an extension of results by Polyak [17] and Breitfeld and Shanno [4].

Identifying \mathbb{S}_1 with \mathbb{R} and introducing a slack variable, it is easy to see that problem (22) can be written in the form (23). Consequently Algorithm (5.1) is directly applicable to problem (22). In the sequel we want to show, how the separable structure of (22) can be exploited by the damped Newton method applied in step (1) of Algorithm 5.1:

Each search direction at a point \hat{x} is computed as a solution of a linear system of the form

$$\nabla_x^2 F(\hat{x}, U^\ell, p^\ell) d = -\nabla_x F(\hat{x}, U^\ell, p^\ell). \quad (27)$$

In order to understand the structure of the system matrix $\nabla_x^2 F(\hat{x}, U^\ell, p^\ell)$, we state F explicitly for problem (22) (assuming $|\mathcal{I}_{\max}| = 1$ for simplicity):

$$\begin{aligned} F(x, u^\ell, U^\ell, p^\ell) &= \tilde{f}_{\text{smat}(\bar{x})}^{L, U, \tau}(x) + \sum_{k \in \mathcal{K}_1} u_j^\ell \varphi_{p^\ell}(g_j(x)) + \sum_{k \in \mathcal{K}_0} u_j^\ell \varphi_{p^\ell}(g_j(x)) \\ &\quad + \sum_{j=1, \dots, m} \langle U_j^\ell, \Phi_{p^\ell}(\text{smat}(x) - \underline{Y}) \rangle + \\ &\quad + \sum_{j=1, \dots, m} \langle U_{j+m}^\ell, \Phi_{p^\ell}(\bar{Y} - \text{smat}(x)) \rangle; \end{aligned}$$

here u^ℓ are Lagrangian multipliers associated with real valued constraints, φ_{p^ℓ} is the scalar version of Φ_{p^ℓ} ,

$$\mathcal{K}_1 := \{k \in \{1, 2, \dots, K\} \mid g_k \text{ depends on exactly one matrix variable}\}$$

and $\mathcal{K}_0 = \{1, 2, \dots, K\} \setminus \mathcal{K}_1$. Now we define

$$q(x) := \sum_{k \in \mathcal{K}_0} u_k^\ell \varphi_{p^\ell}(g_k(x)), \quad r(x) := F(x, u^\ell, U^\ell, p^\ell) - q(x).$$

Obviously $r(x)$ is separable w.r.t. $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m$. On the other hand we obtain for $q(x)$:

$$\nabla_x^2 q(x) = \sum_{k \in \mathcal{K}_0} u_k^\ell \left(\varphi_{p^\ell}''(g_k(x)) \nabla_x g_k(x) \nabla_x g_k(x)^\top + \varphi_{p^\ell}'(g_k(x)) \nabla_{xx}^2 g_k(x) \right)$$

Now, defining β as a vector with entries $\beta_k^\ell(x) := 1/(\varphi_{p^\ell}'(g_k(x)))$ ($k \in \mathcal{K}_0$),

$$H^\ell(x) := \nabla_{xx}^2 r(x) + \sum_{k \in \mathcal{K}_0} u_k^\ell \varphi_{p^\ell}'(g_k(x)) \nabla_{xx}^2 g_k(x)$$

and denoting by $A^\ell(x)$ the matrix with columns $\nabla_x g_k(x)$ ($k \in \mathcal{K}_0$) we are able to prove the following result:

Proposition 5.2. *Any vector $d \in \mathbb{R}^{\hat{n}}$ satisfying the equation*

$$\begin{pmatrix} H^\ell(x) & A^\ell(x) \\ A^{\ell,\top}(x) & -\text{diag}(\beta) \end{pmatrix} \begin{pmatrix} d \\ y \end{pmatrix} = \begin{pmatrix} -\nabla_x F(\hat{x}, U^\ell, p^\ell) \\ 0 \end{pmatrix} \quad (28)$$

is a solution of the linear system (27). Moreover the matrix $H^\ell(x)$ is separable w.r.t. $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m$.

Proof. From the second line in system (28) we see

$$y = \text{diag}(\beta)^{-1} A^{\ell,\top}(x) d. \quad (29)$$

Now, substituting y by the right hand side of (29), the first row of (28) becomes:

$$(H^\ell(x) + A^\ell(x) \text{diag}(\beta)^{-1} A^{\ell,\top}(x)) d = -\nabla_x F(\hat{x}, U^\ell, p^\ell),$$

but this is exactly system (27). \square

Depending on the cardinality of the set \mathcal{K}_0 the system (28) is solved directly or by the following strategy:

1. Compute $(H^\ell(x))^{-1} \nabla_x F(\hat{x}, U^\ell, p^\ell)$ and $(H^\ell(x))^{-1} A^\ell(x)$.
2. Solve the system

$$(A^{\ell,\top}(x) (H^\ell(x))^{-1} A^\ell(x) - \text{diag}(\beta)) y = (H^\ell(x))^{-1} \nabla_x F(\hat{x}, U^\ell, p^\ell).$$

3. Compute $d = -(H^\ell(x))^{-1} A^\ell(x) y - (H^\ell(x))^{-1} \nabla_x F(\hat{x}, U^\ell, p^\ell)$.

The second variant can be viewed as a generalization of the dual technique used in the original MMA paper [19].

6 Numerical experiments

The main goal of this section is

1. to provide algorithmic details of our implementation of Algorithm 4.3,
2. to present the results of numerical experiments with Algorithm 4.3 applied to free material optimization problems of the form (9).

6.1 Algorithmic details

The choice of the asymptotes As a consequence of Lemma (4.2) the formula for the hyperbolic approximation of c_k ($k \in \mathcal{K}$) in the j -th iteration reduces to

$$\begin{aligned} (c_k)_{E^j}^{L,\tau}(E) &:= c_k(E^j) + \\ &\sum_{i=1}^m \left\langle \nabla^i c_k(E^j), (E_i^j - L_i)(E_i - L_i)^{-1}(E_i^j - L_i) - (E_i^j - L_i) \right\rangle_{\mathbb{S}^3} + \\ &\sum_{i=1}^m \tau_i \left\langle (E_i - E_i^j)^2, (E_i - L_i)^{-1} \right\rangle_{\mathbb{S}^3}. \end{aligned} \quad (30)$$

For this reason we neglect the upper asymptotes U below. We have investigated two different types of schemes, moving schemes and constant schemes. For the moving schemes we used direct generalizations of the update rules recommended in [20, 23]. In the case of the constant scheme we simply used

$$L_i^j = L_0 \prec \rho I \text{ for all } i \in \mathcal{I} \text{ and all iterates } j = 1, 2, 3, \dots$$

The result of our experiments showed that (in sharp contrast to the original MMA/SCP-approach) the moving schemes bring almost no benefit compared to the constant one. Additionally, the constant scheme has an important advantage:

The feasible set of the sub-problems can be kept during all iterations. This allows for an extensive use of warm starts, when solving the inner convex semidefinite programs. As a consequence we observed a significant reduction of inner iterations. This is of particular importance, as the sub-problem in FMO is much more expensive to solve as, for example, in SIMP-based problems.

For this reason we report only on experiments with constant schemes of asymptotes. The most efficient constant choice we found was $L_0 = 0$. Note that the combination of Algorithm 4.3 with this simple choice applied to FMO-type problems can be interpreted as a direct generalization of CONLIN [6]. The latter choice leads to a further simplification of (30):

$$\begin{aligned} (c_k)_{E^j}^{\tau}(E) &:= c_k(E^j) + \sum_{i=1}^m \left\langle \nabla^i c_k(E^j), E_i^j E_i^{-1} E_i^j - E_i^j \right\rangle_{\mathbb{S}^3} + \\ &\sum_{i=1}^m \tau_i \left\langle (E_i - E_i^j)^2, E_i^{-1} \right\rangle_{\mathbb{S}^3}, \quad k \in \mathcal{K}. \end{aligned}$$

The subproblems Using the constant asymptote scheme described above the subproblems of (9) become

$$\begin{aligned} \min_{E \in \tilde{\mathcal{E}}} (c_k)_{E^j}^\tau(E) \quad & \text{subject to} \\ \sum_{i=1}^m \text{Tr}(E_i) & \leq V, \end{aligned} \quad (31)$$

During all iterations, we solve the subproblems approximately. We use the following strategy: we start with a moderate accuracy of $\varepsilon = 10^{-3}$ for the KKT error of (31). Whenever the calculated search direction fails to be a descent direction, we decrease the precision by a constant factor.

The line search Instead of performing an exact minimization in step (4) of Algorithm 4.3, we use a simple Armijo rule, guaranteeing sufficient descent. Our experience shows that already after few outer iterates of Algorithm 4.3, the step length $\alpha^j = 1$ is accepted in almost all iterates.

The choice of τ The parameters τ_i ($i \in \mathcal{I}$) are chosen such that the following condition remains valid throughout all iterations:

$$-\nabla^i c_k(E^j) + \tau_i I \succeq \delta I \quad (i \in \mathcal{I})$$

for all $i \in \mathcal{I}$ and all $k \in \mathcal{K}$. A typical choice for δ is 10^{-4} .

A practical stopping criterion? So far, we did not implement a KKT-based stopping criterion for Algorithm 4.3. We stop the algorithm as soon as a maximal number of iterations is exceeded (typically 200-500) or, when the difference of the objective falls below some given threshold (typically 10^{-4}).

The code We have implemented the new algorithm in the C programming language. In what follows we refer to the resulting code as PENSCHP.

6.2 Numerical studies with FMO problems

Before we start presenting the numerical results we want to have a brief look at the (theoretical) computational complexity of Algorithm 4.3, when applied to FMO problems of type (9).

Computational complexity We split the whole process in two sub-tasks, namely *model calculation* and *optimization*. The model calculation includes the evaluation of $c_k(E)$ ($k \in \mathcal{K}$), the computation of all partial derivatives and some pre-assembling steps for the hyperbolic approximations. The optimization covers the solution of a sub-problem of type (31). Clearly, the model calculation is dominated by the factorization of the global stiffness matrix $A(E)$. The factorization is performed by a sparse Cholesky method (see [15]) whose complexity depends linearly on the number of non-zero entries in $A(E)$. From this, formulas (7), (8) and Lemma 2.3 we conclude: The computational complexity of the *model calculation* phase depends

- *linearly* on number of elements in discretization,

- *linearly* on number of load cases.

In the *optimization* phase, the most time consuming steps are the calculation of gradients and Hessians of the hyperbolic approximations and the factorization of (28). Obviously both types of operations depend

- *linearly* on number of elements in discretization,
- *linearly* on number of load cases.

Goals of the numerical experiments The goals of the numerical experiments presented in the remainder of this section are

- the numerical verification of the linear dependence of the computational complexity of Algorithm 4.3 on the number of elements,
- numerical verification of the linear dependence of the computational complexity of Algorithm 4.3 on the number of load cases,
- a comparison with MOPED3, the most recent implementation of the dual method described in [1],
- the effect of a SIMP-like preprocessing step.

All experiments have been performed on a Sun Opteron machine with 8 Gbyte of memory and a processor speed of approximately 3 GHz.

2D examples The goal of our first experiment is to verify the linear dependence of the computational complexity of Algorithm 4.3 on the number of elements in the FEM discretization. In order to do that, we solve several instances of the test problem depicted in figure 1 with an increasing number of elements. Moreover we compare the calculation times of PENSCH and MOPED3 on this example. The results are summarized in the following table. The meaning of columns 1 to 5 is: The number of finite elements, the number of iterations performed by PENSCH, the relative precision reached by PENSCH (w.r.t. a high quality approximation of the accurate solution computed by MOPED3), the computation time required by PENSCH and the computation time required by MOPED3:

FEs	Iter	precision	time in sec. (opt/mod)	time in sec. MOPED3
1.250	500	2.1e-4	151 (77/74)	147
5.000	500	1.0e-4	1.019 (600/419)	988
20.000	500	3.0e-4	7.002 (4.312/2.690)	8.560

Experiment 1: 1.250-20.000 elements

For both codes we observe for a factor 4 in the number of elements a corresponding factor of 7-8 in the computational time. Thus the numerical experiment confirms the theoretically predicted behavior. Optimal density distributions obtained from Experiment 1 are visualized in Figures 2 and 3.

By means of our second experiment we want to verify the linear dependence of the computational complexity of Algorithm 4.3 on the number of load cases. Therefore, we solve the basic test with an increasing number of load cases. Again we compare the results of PENSCH to MOPED3. The results are summarized in the following table. Note that here and below # LC denotes the number of load cases.

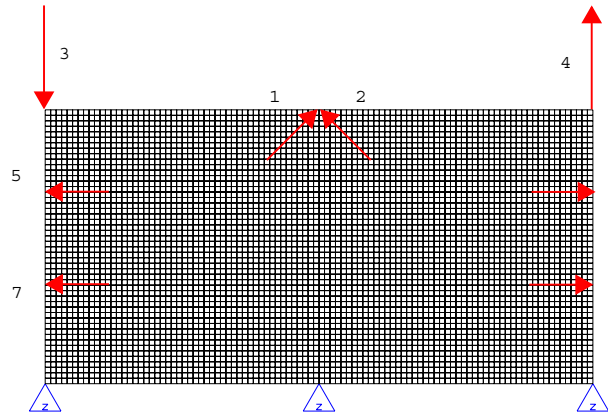


Figure 1: Basic test problem – mesh, boundary conditions and forces

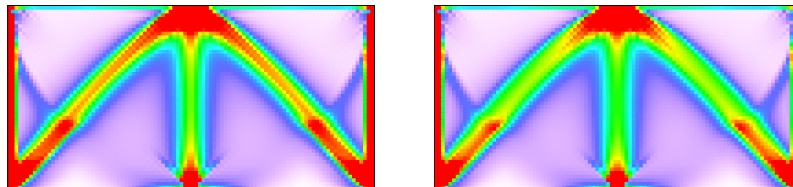


Figure 2: 5000 elements, 4 load cases: MOPED3 (left) / PENSCHP (right)

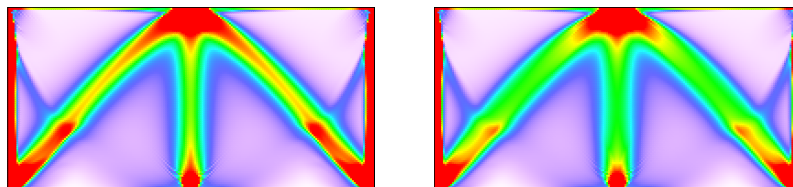


Figure 3: 20.000 elements, 4 load cases: MOPED3 (left) / PENSCHP (right)

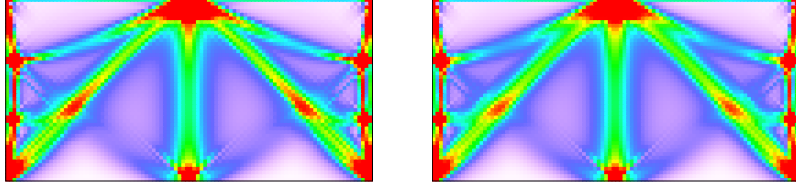


Figure 4: 5.000 elements, 8 load cases: MOPED3 (left) / PENSCLP (right)

# LC	Iter	precision	time in sec. (opt/mod)	time in sec. MOPED3
2	500	< 1.0e-4	630 (320/310)	175
4	500	1.0e-4	783 (420/363)	988
8	500	< 1.0e-4	1.139 (622/517)	7.319

Experiment 2: 2-8 load cases

Obviously PENSCLP shows a much better behavior as MOPED3 here. For a factor 2 in the number of load cases we observe only a factor 1.2-1-4 in the computational time. On the other hand MOPED3 shows the predicted quadratic to cubic behavior: For a factor 2 in the number of load cases we observe a factor 6-8 in the computational time.

By means of a third experiment we want to test the effect of a pre-processing strategy. The goal is here, to reduce the comparatively high number of outer iterations required by PENSCLP. We make use of the following *Preprocessing-Strategy*

- 1) Run a few (10-20) steps of Algorithm 4.3 for a SIMP model obtained from (31) by setting $E_i = \rho_i^2 I$ for all $i = 1, 2, \dots, m$ and letting $\rho = (\rho_1, \rho_2, \dots, \rho_m)^\top$ the design variable.
- 2) Approximate material tensors using the formula

$$E_i \approx \frac{\rho_i}{\#LC} \sum_{k \in \mathcal{K}} \bar{e}(u_k) \bar{e}(u_k)^T,$$

where u_k are displacements associated with the intermediate densities calculated in 1) and $\bar{e}(u_k)$ is a corresponding normed small strain tensor.

For a motivation of the above strategy we refer the reader to [24]. Using preprocessing, we computed our basic example with

- 5.000 finite elements and 8 load cases,
- 20.000 finite elements and 4 load cases.

PENSCLP was stopped after 150 iterations in both cases. The resulting density plots are depicted in Figures 5 and 6.

It seems that the preprocessing strategy significantly improved the result after 150 iterations. In both cases we could save about 60 percent of the computation time.

3D experiments We performed experiments with PENSCLP on two 3D-examples. In our first experiment we used a solid block discretized by approximately 10.000 finite elements. Moreover we applied 2 load cases (see Figure 7, top). PENSCLP stopped after almost 500 iterations and approximately 1.5 hours computation time. MOPED3

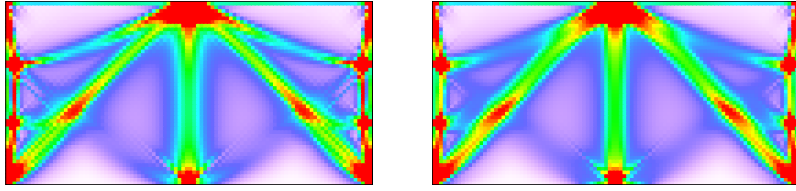


Figure 5: 5.000 elements, 8 LC, with Preprocessing: MOPED3 (left) / PENSCHP (right)

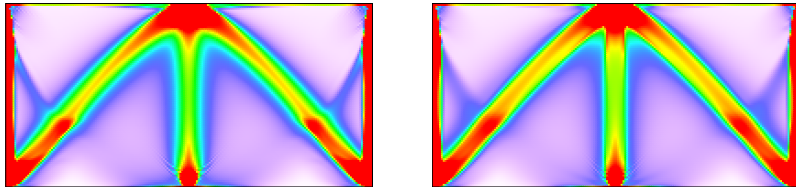


Figure 6: 20.000 elements, 4 LC, with Preprocessing: MOPED3 (left) / PENSCHP (right)

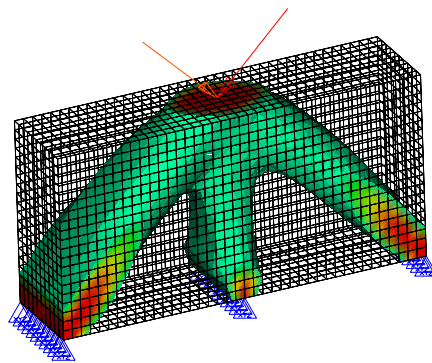
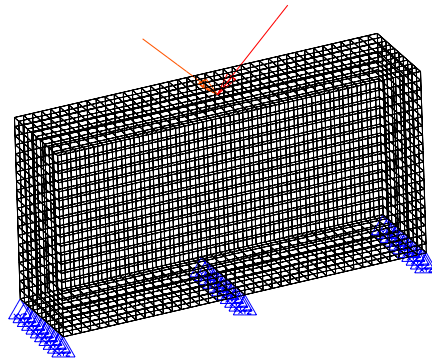


Figure 7: Problem setting: ≈ 10.000 elements, 2 LC (left), Density result, PENSCHP (right)

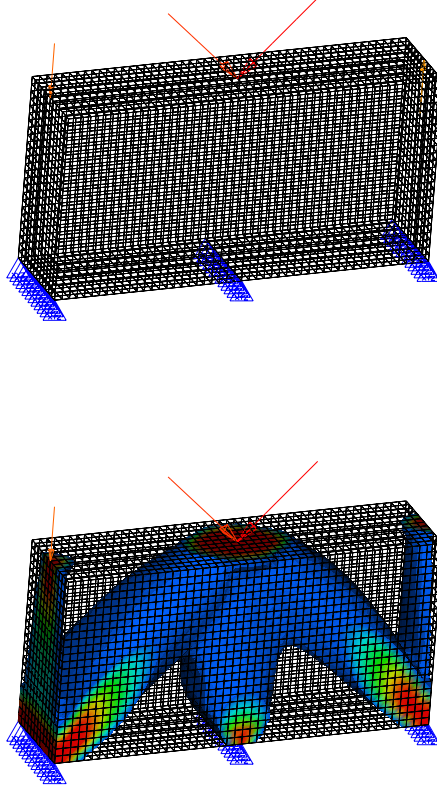


Figure 8: Problem setting: ≈ 20.000 elements, 4 LC (left), Density result, PENSCH (right)

required already 8 hours. The problem setting as well as the density result generated by PENSCH can be seen in Figure 7.

In our second 3D-experiment we used again a solid block, this time subjected to 4 load cases and discretized by approximately 20.000 finite elements. PENSCH was able to generate a solution in approximately 4 hours. MOPED3 failed for this example, because the memory of 8 Gbyte was exceeded. An estimate based on the results from the 3D-experiment described above yields a computation time of approximately two weeks for this example. The problem setting along with the density result computed by PENSCH is depicted in Figure 8.

7 Conclusion and Outlook

We have developed a globally convergent method for the minimization of convex non-linear functions defined on matrix spaces over convex sets described by (separable) convex constraints. The new method turned out to be particularly efficient when ap-

plied to free material optimization problems with multiple load cases. The key strategy of the new method is to replace the basic optimization problem by a sequence of convex semidefinite programs. The structure of these semidefinite programs has a strong influence on the efficiency of the overall method. For example, we have seen that an efficient solution is possible, if all constraints are separable or even linear. If this is not the case, or even worse, if one wants to deal with non-convex functions, the situation is more involved. The authors are currently investigating a generalized algorithmic concept for this case, in which not only the objective function, but also the constraints are replaced by sequences of hyperbolic approximations. As a consequence of the fact that the feasible domain is approximated, such a generalized algorithm may produce infeasible iterates. It turns out that in this case, the line search defined in Section 4 – which is purely based on evaluations of the objective function – can be replaced by a line search which approximately minimizes a merit function of the form

$$\begin{aligned} F(Y, v, W) \quad := \quad & f(Y) + \frac{1}{2p} \sum_{k=1}^K (v_k - pg_k(Y))_+^2 - (v^k)^2 + \\ & \frac{1}{2p} \sum_{i \in \mathcal{I}} \left\| (W^i - p(Y_i - \underline{Y}))_+ \right\|^2 - \|W^i\|^2 + \\ & \frac{1}{2p} \sum_{i \in \mathcal{I}} \left\| (W^{i+m} - p(\bar{Y} - Y_i))_+ \right\|^2 - \|W^{i+m}\|^2. \end{aligned}$$

Here p is a penalty parameter and v, W are Lagrangian multipliers. It is possible to show that using this function, Algorithm 4.3 can be generalized for the solution of arbitrary nonlinear semidefinite programming problems. A corresponding convergence proof is still under construction and will be subject of a forthcoming paper.

Acknowledgements

This work has been partially supported by the EU Commission in the Sixth Framework Program, Project No. 30717 PLATO-N.

References

- [1] A. Ben-Tal, M. Kočvara, A. Nemirovski, and J. Zowe. Free material design via semidefinite programming. The multi-load case with contact conditions. *SIAM J. Optimization*, 9:813–832, 1997.
- [2] M. Bendsøe and O. Sigmund. *Topology Optimization. Theory, Methods and Applications*. Springer-Verlag, Heidelberg, 2002.
- [3] M. P. Bendsøe, J. M. Guades, R.B. Haber, P. Pedersen, and J. E. Taylor. An analytical model to predict optimal material properties in the context of optimal structural design. *J. Applied Mechanics*, 61:930–937, 1994.
- [4] M. Breitfeld and D. Shanno. A globally convergent penalty-barrier algorithm for nonlinear programming and its computational performance. Technical report, Rutcor Research Report, Rutgers University, New Jersey., 1994.

- [5] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, New York, Oxford, 1978.
- [6] C. Fleury. Conlin: An efficient dual optimizer based on convex approximation concepts. *Struct. Optim.*, 1:81–89, 1989.
- [7] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer-Verlag Berlin-Heidelberg, 1993.
- [8] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [9] M. Kočvara J. Zowe and M. Stingl. Free material optimization: Recent progress. *Optimization*, 2007.
- [10] M. Kočvara and M. Stingl. PENNON—a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18(3):317–333, 2003.
- [11] M. Kočvara and M. Stingl. The worst-case multiple-load fmo problem revisited. In N. Olhoff M.P. Bendsoe and O. Sigmund, editors, *IUTAM-Symposium on Topological Design Optimization of Structures, Machines and Material: Status and Perspectives*, pages 403–411. Springer, 2006.
- [12] M. Kočvara and M. Stingl. Free material optimization: Towards the stress constraints. *Structural and Multidisciplinary Optimization*, 33(4-5):323–335, 2007.
- [13] M. Kočvara and M. Stingl. On the solution of large-scale sdp problems by the modified barrier method using iterative solvers. *Mathematical Programming (Series B)*, 109(2-3):413–444, 2007.
- [14] J. Nečas and I. Hlaváček. *Mathematical Theory of Elastic and Elasto-Plastic Bodies: An Introduction*. Elsevier Science, Amsterdam, 1981.
- [15] E. Ng and B. W. Peyton. Block sparse cholesky algorithms on advanced uniprocessor computers. *SIAM J. Scientific Computing*, 14:1034–1056, 1993.
- [16] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.
- [17] R. Polyak. Modified barrier functions: Theory and methods. *Math. Prog.*, 54:177–222, 1992.
- [18] M. Stingl. *On the Solution of Nonlinear Semidefinite Programs by Augmented Lagrangian Methods*. PhD thesis, Institute of Applied Mathematics II, Friedrich-Alexander University of Erlangen-Nuremberg, 2005.
- [19] K. Svanberg. The method of moving asymptotes – a new method for structural optimization. *International Journal for numerical methods in Engineering*, 24:359–373, 1987.
- [20] K. Svanberg. A class of globally convergent optimization methods based on conservative separable approximations. *SIOPT*, 12:555–573, 2002.
- [21] R. Werner. *Free Material Optimization*. PhD thesis, Institute of Applied Mathematics II, Friedrich-Alexander University of Erlangen-Nuremberg, 2000.

- [22] C. Zillober. *Eine global konvergente Methode zur Lösung von Problemen aus der Strukturoptimierung*. PhD thesis, Technische Universität München, 1992.
- [23] C. Zillober. Global convergence of a nonlinear programming method using convex approximations. *Numerical Algorithms*, 27(3):256–289, 2001.
- [24] J. Zowe, M. Kočvara, and M. Bendsøe. Free material optimization via mathematical programming. *Math. Prog., Series B*, 79:445–466, 1997.