Optimization
Vol. 00, No. 00, Month-Month 200x, 1–21

Free Material Optimization: Recent Progress

Michal Kočvara, School of Mathematics, The University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K.
Michael Stingl, Institute of Applied Mathematics, University of Erlangen-Nürnberg Martensstr. 3, 91058 Erlangen, Germany
Jochem Zowe, Laimbacher Str. 73, 95447 Bayreuth, Germany

Dedicated to Bert Jongen on the occasion of his 60th birthday

(Received 00 Month 200x; In final form 00 Month 200x)

We present a compact overview of the recent development in free material optimization (FMO), a branch of structural optimization. The goal of FMO is to design the ultimately best material (its mechanical properties and distribution in space) for a given purpose. We show that the current FMO models naturally lead to linear and nonlinear semidefinite programming problems (SDP); their numerical tractability is then guaranteed by recently introduced SDP algorithms.

Subject Classification: 74P05,90C22,90C26

Keywords: optimization of elastic structures, material optimization, topology optimization, semidefinite programming, method of augmented Lagrangians

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 [4] and later developed in [21] and [2]. The design variable is the full elastic stiffness tensor that can vary from point to point; it should be physically available but is otherwise not restricted. This problem gives the best physically attainable material and can be considered the "ultimate" generalization of the structural optimization problem. The method is supported by powerful optimization and numerical techniques, which allow us to work with bodies of complex initial design and with very fine finite-element meshes, giving thus quite accurate solutions even for bodies with complex geometries. Recently, FMO has been used for conceptual design of aircraft components; the most prominent example is the design of ribs in the leading edge of the new Airbus A380. In this particular case, the use of the modern optimization techniques leads to a significant weight savings; see Fig. 1.
The goal of this article is to show recent developments in FMO formulations and models with the most important technological constraints, such as stress and displacement constraints. The article follows the overview paper [12] and shows the shift in the mathematical programming formulations:

This research was supported by the Academy of Sciences of the Czech Republic through grant No. A1075402. Most of the work was done while the first author was with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic.

the basis of the new models is linear and nonlinear large-scale semidefinite programming; see also [10] for FMO models with stability constraints that again lead to linear or nonlinear SDP problems. This shift in the modeling—from standard nonlinear to semidefinite programming—was enabled by the recent development of new algorithms for (nonlinear) SDP problems. All our FMO models are solved by code PENNON that has been originally developed for this purpose and later turned to a general-purpose NLP and SDP code. In the last section of this article we thus present a brief overview of the basic algorithm implemented in PENNON.

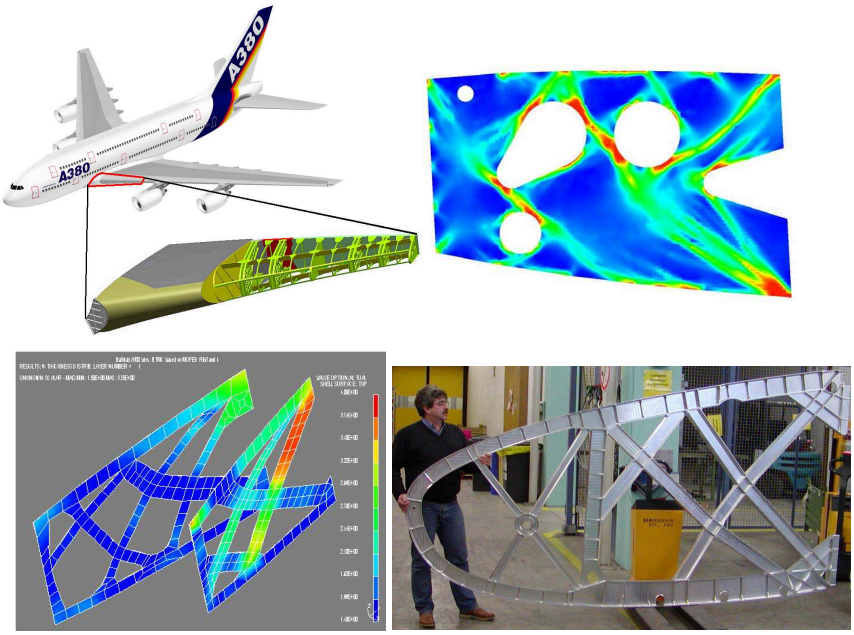

Figure 1. Desing of a leading edge rib of A380 using FMO: the wing leading edge; sample FMO result; post-optimization with technological constraints; final product.

2 Primal and dual FMO problem

We study the optimization of the design of a continuum structure that is loaded by multiple independent forces. In order to deal with the problem in a very general form, we consider the distribution of the material in space as well as the material properties at each point as design variables. The idea to treat the material itself as a function of the space variable goes back to the works [4, 15] and has also been studied in various other contexts; see [3].

2.1 *Problem formulation*

Let $\Omega \subset \mathbb{R}^{dim}$, $dim = 2, 3$, be a bounded domain (the elastic body) with a Lipschitz boundary Γ. We use the standard notation $[H^1(\Omega)]^{dim}$ and $[H^1_0(\Omega)]^{dim}$ for Sobolev spaces of functions $v : \Omega \to \mathbb{R}^{dim}$. By $u(x) = (u_1(x), \ldots, u_{dim}(x))$ with $u \in [H^1(\Omega)]^{dim}$ we denote the *displacement*

vector at point x of the body under load, by

$$e_{ij}(u(x)) = \frac{1}{2}\left(\frac{\partial u(x)_i}{\partial x_j} + \frac{\partial u(x)_j}{\partial x_i}\right) \quad \text{for } i,j = 1,\dots,3$$

the *(small-)strain tensor* and by $\sigma_{ij}(x)\,(i,j=1,\dots,3)$ the *stress tensor*. We assume that our system is governed by linear Hooke's law, i.e., the stress is a linear function of the strain

$$\sigma_{ij}(x) = E_{ijk\ell}(x)e_{k\ell}(u(x)) \qquad \text{(in tensor notation)}, \tag{1}$$

where E is the so-called (plain-stress) *elasticity* tensor of order 4. We will often skip the argument x in the following text. The strain and stress tensors are symmetric (e.g. $e_{ij} = e_{ji}$) and also E is symmetric in the following sense:

$$E_{ijk\ell} = E_{jik\ell} = E_{ij\ell k} = E_{k\ell ij} \qquad i,j,k,\ell = 1,\dots,dim.$$

These symmetries allow us to avoid the tensor notation and interpret the $2-$tensors e and σ as vectors

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

for $dim = 2$ and analogously as vectors in $\mathbb{R}^6$ for $dim = 3$. Correspondingly, the $4-$tensor E can be written as a symmetric 3×3 matrix

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

for $dim = 2$ and as an analogous symmetric 6×6 matrix for $dim = 3$.
Since E will be understood as a matrix in our paper, we will use double indices for the elements of E; the correspondence between E_{ij} and the tensor components $E_{ijk\ell}$ is clear from (2). Recall that in our approach not only e and σ but also E is a function of the space variable x. To include the case of material–no-material, it is natural to work with ($N = 3$ or 6)

$$E \in [L^\infty(\Omega)]^{N\times N} \quad \text{(in short: } E \in L^\infty(\Omega)).$$

We consider a partitioning of the boundary Γ into two parts: $\Gamma = \overline{\Gamma}_1 \cup \overline{\Gamma}_2$, where Γ_1 and Γ_2 are open in Γ and $\Gamma_1 \cap \Gamma_2 = \emptyset$. Further we put
$\mathcal{H} = \{u \in [H^1(\Omega)]^{dim} \mid u_i = 0 \text{ on } \Gamma_1 \text{ for } i = 1 \text{ or } 2 \text{ or } 3 \text{ or any combination}\}$, i.e.,
$[H_0^1(\Omega)]^{dim} \subset \mathcal{H} \subset [H^1(\Omega)]^{dim}$. The system is in equilibrium (outer forces and inner reaction forces balance each other) for any u which solves the weak equilibrium equation

$$\int_\Omega \langle E(x)e(u(x)), e(v(x))\rangle \mathrm{d}x - \int_{\Gamma_2} f(x)\cdot v(x)\mathrm{d}x = 0 \quad \text{for all } v \in \mathcal{H}\,. \tag{3}$$

We now come to the crucial issue. Whereas (3) is the job of the nature, the designer will try to find a material (i.e., a matrix function E) for which the structure is as effective as possible. Physics tells us that the elasticity matrix E has to be *symmetric* and *positive semidefinite* on all of Ω, which we write as

$$E(x) \succcurlyeq 0 \quad \text{for all } x \in \Omega \quad \text{(in short: } E \succcurlyeq 0).$$

As "cost" of the material E we use the trace of E. We further introduce upper and lower bounds $\underline{\rho} \geq 0$ and $\overline{\rho} > 0$

$$\underline{\rho} \leq \mathrm{Tr}(E(x)) \leq \overline{\rho} \qquad \text{for all } x \in \Omega.$$

We look for a structure which can withstand a whole set of loads f^ℓ, $\ell = 1, \ldots, L$, in the worst-case sense. The stiffness of the structure for a particular load is measured by the *compliance* $\frac{1}{2}\int_{\Gamma_2} f^\ell(x) \cdot u^\ell(x) \mathrm{d}x$. We require with a given $\gamma > 0$:

$$\int_{\Gamma_2} f^\ell(x) \cdot u^\ell(x) \mathrm{d}x \leq \gamma, \quad \ell = 1, \ldots, L.$$

The *minimum weight FMO problem* reads as

$$\min_{\substack{(u^1,\ldots,u^L)\in\mathcal{H}\times\ldots\times\mathcal{H} \\ E\in L^\infty(\Omega)}} \int_\Omega \mathrm{Tr}(E)\mathrm{d}x \tag{4}$$

subject to

$$E \succeq 0$$

$$\underline{\rho} \leq \mathrm{Tr}(E) \leq \overline{\rho}$$

$$u^\ell \text{ solves (3) with } f = f^\ell \text{ for all } \ell = 1, \ldots, L$$

$$\int_{\Gamma_2} f^\ell(x) \cdot u^\ell(x) \mathrm{d}x \leq \gamma, \quad \ell = 1, \ldots, L\,.$$

Equivalently (up to a scaling), we seek the design function E which yields the smallest possible worst-case compliance

$$\min_{\substack{(u^1,\ldots,u^L)\in\mathcal{H}\times\ldots\times\mathcal{H} \\ E\in L^\infty(\Omega)}} \max_{\ell=1,\ldots,L} \int_{\Gamma_2} f^\ell(x) \cdot u^\ell(x) \mathrm{d}x \tag{5}$$

subject to

$$E \succeq 0$$

$$\underline{\rho} \leq \mathrm{Tr}(E) \leq \overline{\rho}$$

$$u^\ell \text{ solves (3) with } f = f^\ell \text{ for all } \ell = 1, \ldots, L$$

$$\int_\Omega \mathrm{Tr}(E) \leq V\,.$$

2.2 *Discretization*

In order to solve our (infinite-dimensional) problems numerically, we have to discretize them. For the discretization we use the finite element method. Details of the corresponding convergence analysis can be found in [20].
To simplify the notation, we use the same symbols for the discrete objects (vectors) as for the "continuum" ones (functions). Assume that Ω can be partitioned into m squares ($dim = 2$) or bricks ($dim = 3$) called Ω_i which are all of the same size (otherwise we use the standard isoparametric concept, see [6]). Let us denote by n the number of nodes (vertices of the elements). We approximate the matrix function $E(x)$ by a function that is constant on each element, i.e.,

characterized by a vector of matrices $E = (E_1, \ldots, E_m)$ of its element values. Further assume that the displacement vector $u(x)$ is approximated by a continuous function that is bi- or tri-linear (linear in each coordinate) on every element. Such a function can be written as $u(x) = \sum_{i=1}^{n} u_i \vartheta_i(x)$ where u_i is the value of u at i^{th} node and ϑ_i is the basis function associated with i^{th} node (for details, see [6]). Recall that, at each node, the displacement has dim components, so $u \in \mathbb{R}^{dim \cdot n}$.
With the basis functions $\vartheta_j,\ j = 1, \ldots, n$, we define (3×2) and (6×3) matrices

$$\widehat{B}_j = \begin{pmatrix} \frac{\partial \vartheta_j}{\partial x_1} & 0 \\ 0 & \frac{\partial \vartheta_j}{\partial x_2} \\ \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_2} & \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_1} \end{pmatrix} \qquad \widehat{B}_j = \begin{pmatrix} \frac{\partial \vartheta_j}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \vartheta_j}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \vartheta_j}{\partial x_3} \\ \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_2} & \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_1} & 0 \\ 0 & \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_3} & \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_2} \\ \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_3} & 0 & \frac{1}{2}\frac{\partial \vartheta_j}{\partial x_1} \end{pmatrix}$$

for $dim = 2$ and $dim = 3$, respectively. Now, for an element Ω_i, let $\mathcal{D}_i$ be an index set of nodes belonging to this element. Let nig denotes the number of Gauss integration points in each element. By $B_{i,k}$ we denote the block matrix composed of (3×2) blocks $\widehat{B}_j$ at the j-th position, $j \in \mathcal{D}_i$, (evaluated at the k-th integration point) and zeros otherwise. Hence the full dimension of $B_{i,k}$ is $(3 \times 2n)$.
The (global) stiffness matrix A is a sum of element stiffness matrices A_i:

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

After the discretization, problem (4) becomes

$$\min_{u,\ldots,u^L,\, E} \sum_{i=1}^{m} \operatorname{Tr}(E_i) \tag{6}$$

subject to

$$\begin{aligned} & E_i \succeq 0, \quad i = 1, \ldots, m \\ & \underline{\rho} \le \operatorname{Tr}(E_i) \le \overline{\rho} \quad i = 1, \ldots, m \\ & (f^\ell)^T u^\ell \le \gamma, \quad \ell = 1, \ldots, L \\ & A(E)u^\ell = f^\ell, \quad \ell = 1, \ldots, L\,. \end{aligned}$$

Problem (6) is a mathematical programming problem with linear matrix inequality constraints and standard nonlinear constraints; this is the so-called nonlinear semidefinite programming (NSDP) problem. In Section 5 we will present two modifications of an augmented Lagrangian algorithm used in our software package Pennon that can be used to the solution of NSDP problems of type (6). Using a simple trick, we can rewrite the nonconvex problem (4) as a convex one. Assume that $\underline{\rho} > 0$. Then $A(E)$ is positive definite (i.e., non-singular) and we can eliminate the displacement

variable using the equilibrium equation $u^\ell = A(E)^{-1} f^\ell$ to get a *reduced primal problem*

$$\min_E \sum_{i=1}^m \mathrm{Tr}(E_i) \tag{7}$$

subject to

$$E_i \succeq 0, \quad i = 1, \ldots, m$$
$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho}, \quad i = 1, \ldots, m$$
$$(f^\ell)^T A(E)^{-1} f^\ell \leq \gamma, \quad \ell = 1, \ldots, L\,.$$

which is now convex. This problem still includes a mixture of NLP and SDP constraints. However, using the Schur complement theorem, it can be written as a linear SDP problem:

$$\min_E \sum_{i=1}^m \mathrm{Tr}(E_i) \tag{8}$$

subject to

$$E_i \succeq 0, \quad i = 1, \ldots, m$$
$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho}, \quad i = 1, \ldots, m$$
$$\begin{pmatrix} \gamma & (f^\ell)^T \\ f^\ell & A(E) \end{pmatrix} \succeq 0, \quad \ell = 1, \ldots, L\,.$$

2.3 *Dual formulation*

Problem (8) is a linear SDP, thus a problem of a seemingly simple structure and (still seemingly) simple to solve, given the existence of high-quality primal-dual linear SDP codes. However, it turns out that the problem is just too large for the current software. Recall that the main result of FMO is a picture; the finite elements serve as pixels in this picture and it is thus important to have a reasonably fine finite-element mesh in order to get a useful information. The minimum number of elements is considered 5000 in the two-dimensional situation and it can well go to 100000 in complex three-dimensional cases. The size of the matrix inequality in (8) is, roughly, *dim* times the number of elements.
The way out offers dualization of the original FMO formulation. Consider the discretized version of problem (5):

$$\min_{u,\ldots,u^L,\,E} \max_{\ell=1,\ldots,L} (f^\ell)^T u^\ell \tag{9}$$

subject to

$$E_i \succeq 0, \quad i = 1, \ldots, m$$
$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho} \quad i = 1, \ldots, m$$
$$A(E) u^\ell = f^\ell, \quad \ell = 1, \ldots, L$$
$$\sum_{i=1}^m \mathrm{Tr}(E_i) \leq V\,.$$

It is easy to see that this is equivalent to (6), up to scaling. This is further equivalent to the

following saddle-point problem

$$\min_{E} \max_{\ell=1,\ldots,L} \max_{u} -\frac{1}{2} u^T A(E) u - (f^\ell)^T u \tag{10}$$

subject to

$$E_i \succeq 0, \quad i = 1, \ldots, m$$
$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho} \quad i = 1, \ldots, m$$
$$\sum_{i=1}^{m} \mathrm{Tr}(E_i) \leq V\,.$$

Problem (10) can now be shown to be Lagrangian dual to the following problem

$$\min_{\substack{u^1,\ldots,u^L,\alpha \geq 0 \\ \lambda_\ell \geq 0, \sum \lambda_\ell \leq 1, \underline{\beta} \geq 0, \overline{\beta} \geq 0}} \alpha V - 2 \sum_{\ell=1}^{L} \lambda_\ell (f^\ell)^T u^\ell - \underline{\rho} \sum_{i=1}^{m} \underline{\beta}_i + \overline{\rho} \sum_{i=1}^{m} \overline{\beta}_i \tag{11}$$

subject to

$$\sum_{\ell=1}^{L} \lambda_\ell B_i(u^\ell) B_i(u^\ell)^T \preceq (\alpha - \underline{\beta}_i + \overline{\beta}_i) \cdot I_N, \quad i = 1, \ldots, m$$

with

$$B_i(v) = \sum_{k=1}^{nig} B_{i,k} v\,.$$

This, again, is a nonconvex SDP problem. However, using a simple change of variables, it can be transformed into a convex one. Put $\tilde{u}_\ell = \lambda_k u_\ell$. Then (11) becomes

$$\min_{\substack{\tilde{u}^1,\ldots,\tilde{u}^L,\alpha \geq 0 \\ \lambda_\ell \geq 0, \sum \lambda_\ell \leq 1, \underline{\beta} \geq 0, \overline{\beta} \geq 0}} \alpha V - 2 \sum_{\ell=1}^{L} (f^\ell)^T \tilde{u}^\ell - \underline{\rho} \sum_{i=1}^{m} \underline{\beta}_i + \overline{\rho} \sum_{i=1}^{m} \overline{\beta}_i \tag{12}$$

subject to

$$\sum_{\ell=1}^{L} \frac{1}{\lambda_\ell} B_i(\tilde{u}^\ell) B_i(\tilde{u}^\ell)^T \preceq (\alpha - \underline{\beta}_i + \overline{\beta}_i) \cdot I_N, \quad i = 1, \ldots, m\,,$$

which is now a problem with a linear objective and convex, though nonlinear, semidefinite constraints.

Remark 1 Consider a *single-load* problem with $L = 1$. Then (11) reduces to

$$\min_{\substack{u,\alpha \geq 0 \\ \underline{\beta} \geq 0, \overline{\beta} \geq 0}} \alpha V - 2 f^T u - \underline{\rho} \sum_{i=1}^{m} \underline{\beta}_i + \overline{\rho} \sum_{i=1}^{m} \overline{\beta}_i \tag{13}$$

subject to

$$B_i(u) B_i(u)^T \preceq (\alpha - \underline{\beta}_i + \overline{\beta}_i) \cdot I_N, \quad i = 1, \ldots, m$$

which simplifies to a problem with standard convex quadratic inequality constraints

$$\min_{\substack{u,\alpha\geq 0\\ \underline{\beta}\geq 0,\overline{\beta}\geq 0}} \alpha V - 2f^T u - \underline{\rho}\sum_{i=1}^{m}\underline{\beta}_i + \overline{\rho}\sum_{i=1}^{m}\overline{\beta}_i \tag{14}$$

subject to

$$u^T\left(\sum_{k=1}^{nig} B_{i,k}^T B_{i,k}\right) u \leq \alpha - \underline{\beta}_i + \overline{\beta}_i, \quad i=1,\ldots,m\,.$$

Problem (14) is the formulation derived directly in the original FMO paper [4] (up to the lower and upper bounds). This formulation (convex QCQP) can be solved very efficiently by recent large-scale NLP solvers.

Let us return the the convex SDP multiple-load formulation (12). It is readily seen that, using the Schur complement theorem, it is equivalent to the following linear SDP problem

$$\min_{\substack{\tilde{u}^1,\ldots,\tilde{u}^L,\alpha\geq 0\\ \lambda_\ell\geq 0,\sum\lambda_\ell\leq 1,\underline{\beta}\geq 0,\overline{\beta}\geq 0}} \alpha V - 2\sum_{\ell=1}^{L}(f^\ell)^T\tilde{u}^\ell - \underline{\rho}\sum_{i=1}^{m}\underline{\beta}_i + \overline{\rho}\sum_{i=1}^{m}\overline{\beta}_i \tag{15}$$

subject to

$$\begin{pmatrix} (\alpha-\underline{\beta}_i+\overline{\beta}_i)\cdot I_N & B_{i,1}\tilde{u}_1\ldots B_{i,nig}\tilde{u}_1 & \ldots & B_{i,1}\tilde{u}_L\ldots B_{i,nig}\tilde{u}_L \\ & \lambda_1 I_G & & \\ & & \ddots & \\ \text{sym.} & & & \lambda_L I_G \end{pmatrix} \succcurlyeq 0,$$

$$i=1,\ldots,m\,,$$

with $G = L\cdot nig$. But this is exactly the formulation derived directly in [2].

2.4 *Numerical solution: linear or nonlinear?*

In the previous section we have first derived a nonlinear SDP and later an equivalent linear SDP formulation of the multiple-load problem. Let us ask which of these formulations is more suitable for the numerical solution of large-scale problems, which is our primal goal.
Let us first introduce another change of variables in problem (12)

$$\lambda = \mu^2, \qquad u^\ell = v^\ell/\mu$$

to arrive at the problem

$$\min_{\substack{v^1,\dots,v^L,\alpha\geq 0\\ \mu,\underline{\beta}\geq 0,\overline{\beta}\geq 0}} \alpha V - 2\sum_{\ell=1}^{L}\mu^\ell (f^\ell)^T v^\ell - \underline{\rho}\sum_{i=1}^{m}\underline{\beta}_i + \overline{\rho}\sum_{i=1}^{m}\overline{\beta}_i \tag{16}$$

subject to

$$\sum_{\ell=1}^{L} B_i(v^\ell)B_i(v^\ell)^T \preccurlyeq (\alpha - \underline{\beta}_i + \overline{\beta}_i)I_N, \quad i=1,\dots,m$$

$$\sum_{\ell=1}^{L}(\mu^\ell)^2 \leq 1, \qquad \mu^\ell \geq 0, \quad \ell = 1,\dots,L\,.$$

The resulting problem is a semidefinite program with bilinear objective function and convex quadratic matrix inequality constraints. This kind of problems can be efficiently solved by the code PENBMI [8], a special version of PENNON; see Section 5. Note that the above formulation was first proposed in the thesis of R. Werner [20] who, however, did not consider it numerically important due to the lack of nonlinear SDP algorithm at that time.
The main (and only) advantage of (15) over (16) is the fact that it is a linear problem that can be solved by available linear SDP software. Benchmark examples, however, show that standard software packages have difficulties to solve large-scale problems; in fact, only PENNON and SeDuMi [17] can solve examples of practical dimensions [13]. The problem has several other disadvantages:

- the constraint matrices in (15) are sparse but relatively large and, compared to the single-load problem, the solution time is considerably high for large problems;
- the size of the constraint matrices is linearly proportional to the number of load cases; that means, the solution time for problems with a higher number of load cases is rather high;
- the dimension of the constraint matrices in (15) increases considerably when switching from 2d to the 3d problems (from $8\cdot 4\cdot L$ to $24\cdot 8\cdot L$);
- the optimal elasticity matrices $E_1^*,\dots,E_M^*$ are not readily available and their recovering requires solution of an auxiliary problem; see [2].

On the other hand, the advantages of (16) are significant; in particular

- the convex quadratic matrix constraints are of much smaller dimension than the linear constraints in (15); further, the dimension (and the number of the constraints) does not increase with the number of load cases and only mildly increases when going from 2d to 3d problems (from 3×3 to 6×6);
- the material matrices are the multipliers to these constraints and are readily available in the PENBMI code; this is not the case of (15) for which a postprocessing has to be performed which is of the same complexity as the problem solution itself.

2.5 *Complexity estimates and example*

Recall first that the reduced primal problem (7) is known to be quite inefficient, compared to the problem (15), as it is a nonlinear SDP problem with a difficult equilibrium constraint. It can be solved by the recent version of the code PENNON; however, because explicite calculation of second order derivatives is impossible due to computational complexity and storage requirements, and because PENNON is based on a second-order algorithm, we solve the Newton system approximately by a conjugate gradient method, which is based on Hessian-vector products. As a result, the whole approach is not that robust as the other two approaches, where the analytic second derivatives are available. In order to be able to work with the inverse of the stiffness matrix, we also have to relax the original constraint $0\leq \mathrm{Tr}(E_m)$ to $0<\underline{\rho}\leq \mathrm{Tr}(E_m)$.
It is now easy to see the dependence of the computational complexity of the three problems on the number of load-cases. While in the current formulation (15) this dependence is cubic, in the new

formulation (16) it is only quadratic. Furthermore, the dependence is only linear in the primal formulation (7). These complexity estimates will be clearly demonstrated in the following numerical example.

Example 2.1 We consider an academic two-dimensional problem with the number of load-cases increasing from two to ten. All problems are discretized using 5000 elements. In the next two figures we see the data (left) and optimal solution (right) for the problem with two and six load-cases, respectively.

Figure 2. Example 1: initial design and optimal result for two load-cases

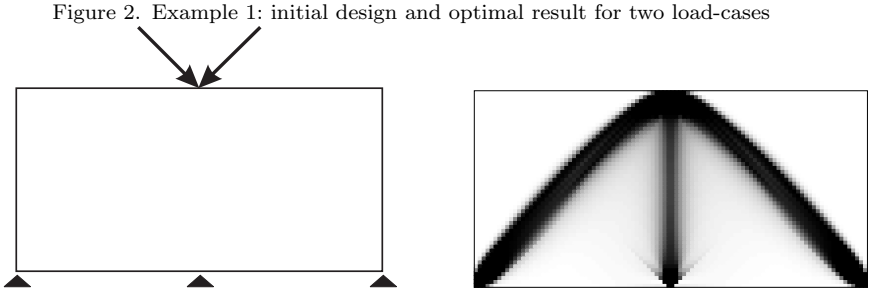

Figure 3. Example 1: initial design and optimal result for six load-cases

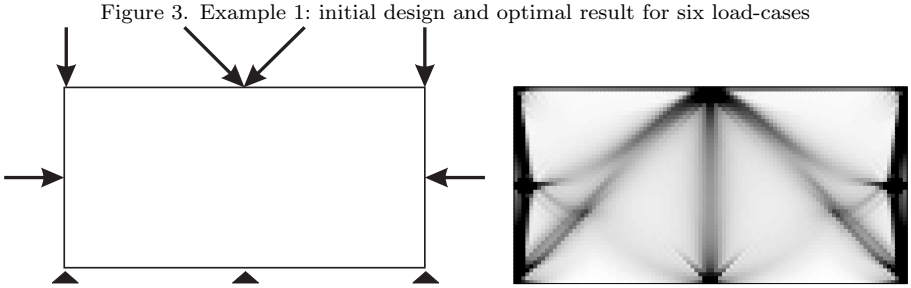

Table 1 gives the dimensions of the problems: while the number of the primal variables (given just by the number of finite elements) stays the same, the number of dual variables (given by the displacement vectors) increases linearly with the number of load-cases.

Table 1. Example 1: problem dimensions

elements	loads	dual variables	primal variables
5000	2	20598	30001
5000	4	41205	30001
5000	6	61812	30001
5000	10	103122	30001

Table 2 then presents the results for the three approaches: the linear SDP formulation (15), the dual nonlinear SDP formulation (16) and also the primal formulation of the problem (7). We can clearly see the effect predicted by the complexity estimates. For small number of load-cases, the dual nonlinear SDP formulation is clearly the best one. With increasing number of load-cases, it is better and better than the linear SDP formulation. The primal formulation, as expected, is by far the worst one for small number of load-cases. However, its complexity grows only linearly with this

number, and we can see that for large number of load-cases it actually becomes the most efficient formulation; a rather unexpected effect. But recall that the (first-order) primal formulation is much less robust than the other two and, for large real-world problems, this effect may not be confirmed.

Table 2. Example 1: numerical results for 2-10 forces (process time in seconds)

loads	process time (linear SDP)	process time (dual nl. SDP)	process time (primal nl. SDP)
2	197	120	597
4	896	512	1171
6	2992	1473	1496
10	15356	5216	3012

3 Stress constraint

In engineering practise, it is not (only) the compliance but some measure of local strain that is of main interest and should be controlled. One of the most often causes of structural failure is high stress, so it is desirable to keep it within given limits during the optimization process.
This, however, is not an easy task. First, when designing general anisotropic material, it is unclear what kind of stress measure (failure criterion) to take. To a great extent, this depends on the realization of the optimal result: should the material be manufactured as fibrous composite, a laminate, by tape-laying procedure, should it be just approximated by isotropic material with reinforcement—all these technologies use different failure criteria which are sometimes not even well understood. We, however, do not want to limit ourselves to a particular manufacturing procedure in this phase, rather to keep the design process as general as possible. Hence we decided to evaluate the stress by a norm of the stress tensor, integrated over the finite element. Another reason why to take this particular measure of stress is to keep the problem computationally tractable. This is, in fact, the second reason why there are not many successful approaches to stress constraints reported in the literature. Stress constraints, added to topology or material design problem, lead to hard optimization problems with so-called vanishing constraints [1] and/or problems that do not satisfy standard constraint qualifications and are thus very difficult to solve by common software of mathematical programming.
In the continuous formulation, we would work with pointwise stresses, i.e., we would restrict the norm $\|\sigma(x)\|$ for all $x \in \Omega$. However, in the finite element approximation we use the primal formula (working with displacements) and it is a well-known fact that, generally, evaluation of stresses (from displacements) at points may be rather inexact. Hence we will consider the following integral form of stress and strain constraints

$$\int_{\Omega_i} \|\sigma\|^2 \leq s_\sigma |\Omega_i| \,; \tag{17}$$

here Ω_i is the i^{th} finite element and $|\Omega_i|$ its volume. The integrals will be further approximated by the Gaussian intergation formulas, as in the finite element interpolation. To simplify the notation, in the following, we will skip the multiplication by the volume $|\Omega_i|$ and consider it included in the upper bounds s_σ; in all numerical examples we will use elements of the same volume, so this should not lead to any misunderstanding.

3.1 *FMO with stress and strain constraints*

The stress in the FMO result is defined by $\sigma = Ee$, hence the FMO problem with stress constraints reads as

$$\min_{u,E} \sum_{i=1}^{m} \mathrm{Tr}(E_i) \tag{18}$$

subject to

$$E_i \succeq 0, \quad i = 1, \ldots, m$$

$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho} \quad i = 1, \ldots, m$$

$$f^T u \leq \gamma$$

$$A(E)u = f$$

$$\sum_{k=1}^{nig} \|EB_{i,k}u\|^2 \leq s_\sigma \overline{\rho}^2, \qquad i = 1, \ldots, m\,.$$

This is again a nonlinear semidefinite programming problem that can be solved by the variant of the code PENNON described in Section 5.
The classic example for testing the effect of stress constraints is the L-shaped domain; see Fig. 4. When made of homogeneous isotropic material, the structure has a stress concentration at the peak of the re-entrant corner. If we transform the problem to a local radial coordinate system, located at this corner, the radial stress components would go to infinity when approaching the origin. In the global Cartesian coordinate system, the norm of the stress tensor goes to infinity as we approach the re-entrant corner. When we solve the discretized problems with a homogeneous mesh, the stress would only go to infinity when the mesh size parameter went to zero. For fixed mesh size, however, the stress values still reach much higher values at the elements neighboring the corner than in the rest of the domain. For standard structural optimization techniques, like shape or topology optimization, the only way to remove the stress singularity is to change the geometry of the domain, in particular, to replace the sharp corner by a sort of smooth arc. For all the subsequent examples we will consider discretization of the domain by 7500 finite elements (squares) of the same size.

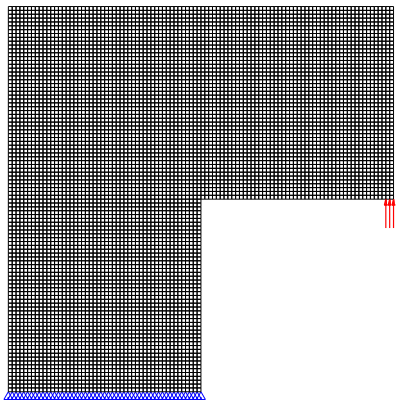

Figure 4. L-shaped domain: geometry, load and boundary conditions

Remark 1 Before presenting the results, let us mention two most important facts that we have observed. First, the FMO model offers much more freedom compared to topology optimization

problems with a given isotropic material. While in the topology optimization the stress concentration has to be removed by change of geometry of the optimal structure, in the FMO case it can be treated *solely by the material properties around the re-entrant corner*.
Second, we observed a rather surprising fact that the optimal material was always almost orthotropic, notwithstanding the stress/strain constraints. This has been tested and confirmed a-posteriori for all examples and all elements. We do not see any theoretical reason that would explain this, but it allows us to plot the material directions of the optimal material and to better demonstrate its properties.

Example 3.1 We first solve the FMO problem without additional stress or strain constraints, to get reference values. The optimal structure, shown in Fig. 5 has volume 232.04 and the maximal stress norm (17) is 0.202. We also evaluate the maximal values of the Tresca, von Mises, maximum stress and maximum strain failure criteria, computed at points of Gaussian integration.
We further solve the FMO problem (18) with stress constraints. Table 3 shows the values of optimal volume and maximal stress, strain and failure criteria for examples with three different upper bounds, namely $s_\sigma = 1.0$, $s_\sigma = 0.7$ and $s_\sigma = 0.5$. Obviously, the maximal stress values were on their upper bound for the stress constrained problem. The respective numbers are emphasized in the table.

Table 3. FMO problem with stress constraints

	no constr.	stress 1.0	stress 0.7	stress 0.5
volume	228.62	230.53	232.43	235.43
stress	0.201	*0.0625*	*0.04375*	*0.03125*
strain	3.431	20.15	15.3	32.34
Tresca	0.188	0.108	0.102	0.093
von Mises	0.0387	0.0142	0.0107	0.0092
$\sigma_{\max}$	0.321	0.181	0.149	0.123
$e_{\max}$	1.347	2.674	2.240	3.146

Figures 5–8 present the corresponding distributions of optimal material stiffness, and optimal stress and strain norms for a problem with no stress constraint and for the most restrictive stress constraint with $s_\sigma = 0.5$. For both examples, after presenting the optimal stiffness, we show detail of the optimal principal stress direction (and thus optimal material direction). We can see that in both cases the material around the re-entrant corner is composed of fibres making a smooth arc around the corner, thus preventing the stress singularity. The last two figures for each example show the optimal stress and strain distribution. We can see that the stress constraints are always active just around the stress singularity or in regions with a stiff material.

4 Displacement constraint

When solving practical problems, it is often important to consider additional constraints on the displacements of the optimal structure. There may be different motivations for this, for instance

- The deformed optimal structure should have prescribed shape. When designing, for instance, parabolic antenna, we want its boundary to be parabolic *after* the deformation of the structure by structural forces. Or we may require that a straight boundary of another structure remains straight even when we load the structure by external forces.
- Certain part of the boundary of the optimal structure should move in certain direction. This may be desirable when designing various mechanism; see, e.g., [3]. For instance, we may want to design such a structure that when we apply a force at one corner, the opposite corner will move in a prescribed direction.

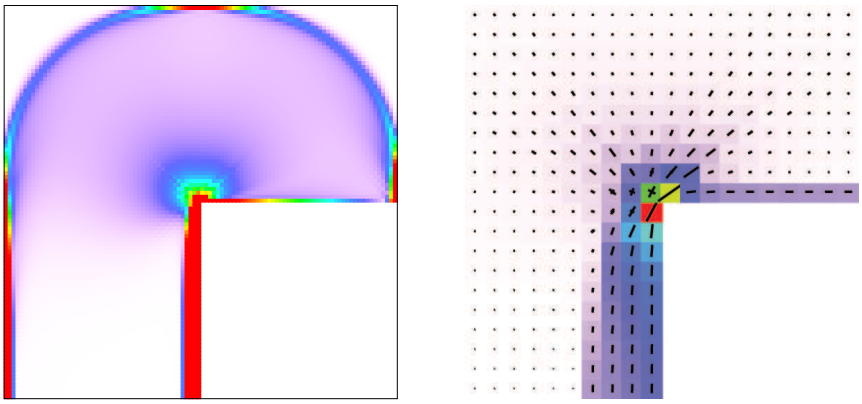

Figure 5. FMO formulation with no stress/strain constraints: optimal ρ (left) and optimal principal stress (right)

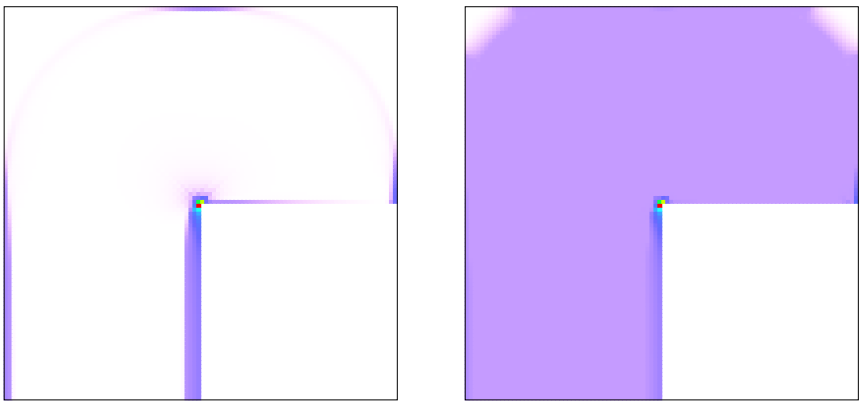

Figure 6. FMO formulation with no stress/strain constraints: optimal stress norms (left) and optimal strain norms (right)

In this paper, we consider, for simplicity, linear displacement constraints of the type

$$Cu \leq d$$

with a matrix $C \in \mathbb{R}^{r \times n}$ and a vector $d \in \mathbb{R}^r$.

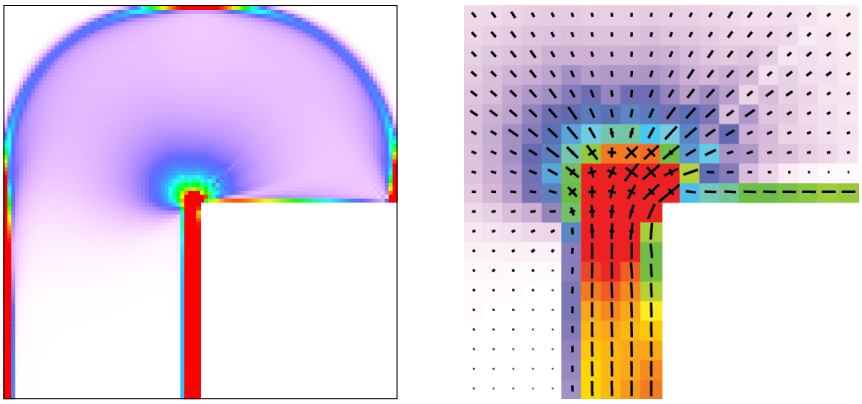

Figure 7. FMO formulation with stress constraints, $s_\sigma = 0.5$: optimal ρ (left) and optimal principal stress (right)

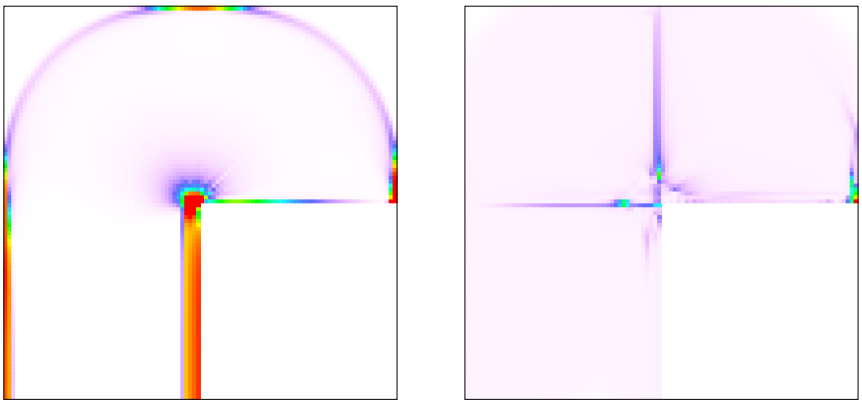

Figure 8. FMO formulation with stress constraints, $s_\sigma = 0.5$: optimal stress norms (left) and optimal strain norms (right)

The primal formulation of the free material optimization problem with displacement constraints then reads as follows:

$$\min_{u,E} \sum_{i=1}^{m} \mathrm{Tr}(E_i) \tag{19}$$

subject to

$$E_i \succeq 0$$

$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho}, \quad i = 1, \ldots, m$$

$$f^\top u \leq \gamma$$

$$A(E)u = f$$

$$Cu \leq d\,.$$

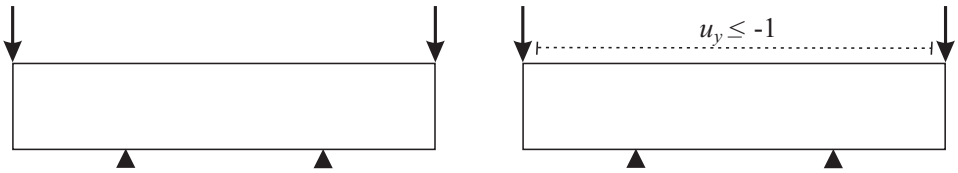

Figure 9. Beam without (left) and with (right) displacement constraints

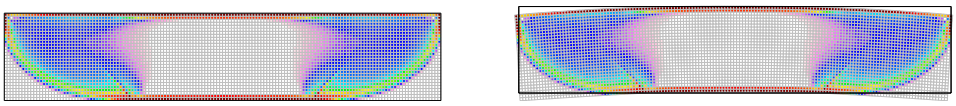

Figure 10. Optimal result for the beam without displacement constraints; density distribution and deformation

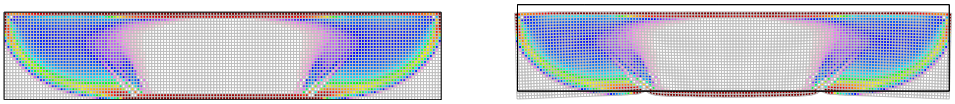

Figure 11. Optimal result for the beam with displacement constraints; density distribution and deformation

Just like with the stress constraint problem in the previous section, problem (19) is a nonlinear semidefinite programming problem that can be solved by the code PENNON.

4.1 *Examples*

Example 4.1 (Prescribed shape) We first consider a simply supported beam depicted in Fig. 9 left, together with the boundary conditions and the load. Figure 10 shows the density of the optimal material for a problem without any additional displacement constraints. The right-hand side of this figure shows the deformation of the optimal beam; as expected, the optimal structure bends. Let us now require that the top segment of the beam remains straight *after* the deformation. This leads to additional linear displacement constraints. The optimal result (density distribution and displacements) is presented in Fig. 11. We can see that, indeed, the displacement constraints are fulfilled and the deformation of the optimal structure differs significantly from the previous case without displacement constraints. Surprisingly, there is no significant change in the density distribution of the optimal material. That means that all changes needed to fulfill the displacement constraints are performed on the level of material properties of single elements.

Example 4.2 (Displacement inverter) In this example we want to design displacement inverter. The goal is to find a structure that converts an input force on the left-hand edge to a displacement in the opposite direction on the right-hand edge. The initial structure is shown in Fig. 12, together with the boundary conditions, the force and the displacement constraint. Figure 13 shows the FMO results *without* the displacement constraint; the structure is as stiff as possible in order to support the given force. The FMO results *with* the displacement constraints is presented in Fig. 14: we see the density distribution of the optimal material (left) and the deformation of the optimal structure (right). We can see sort of a compliant mechanism that indeed fulfills the displacement constraint and that, unlike in the beam example, differs significantly from the result without the displacement constraint.

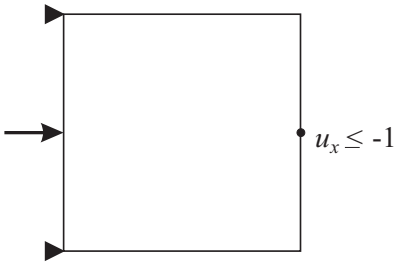

Figure 12. Displacement inverter: initial design, boundary conditions, force and displacement constraint

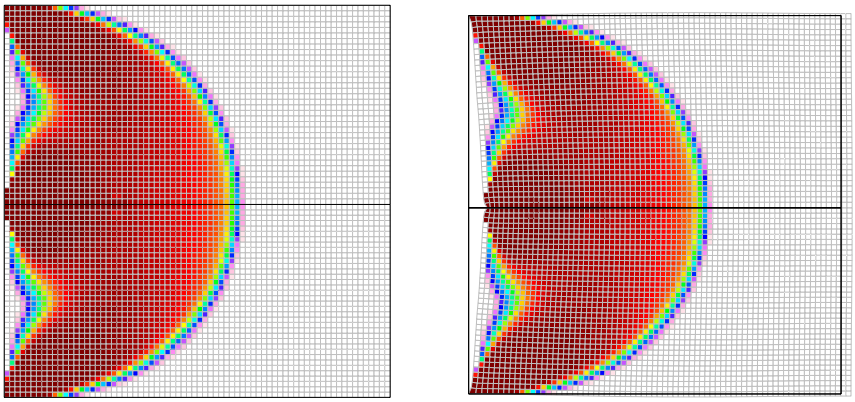

Figure 13. Optimal result for the displacement converter without displacement constraints; density distribution and deformation

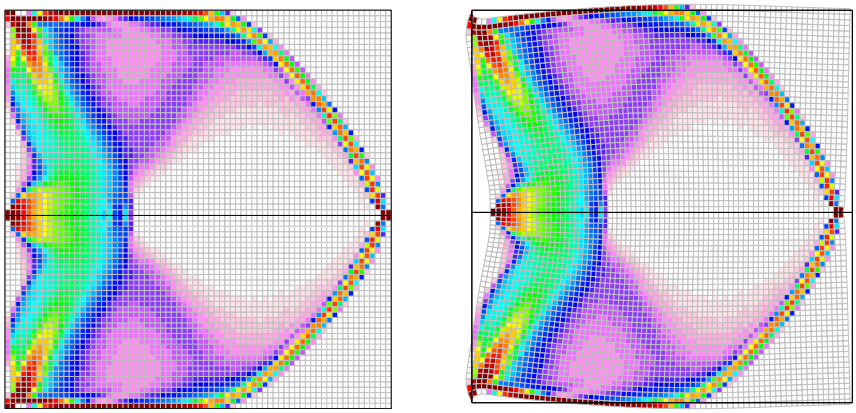

Figure 14. Optimal result for the displacement converter with displacement constraints; density distribution and deformation

5 The algorithm

The algorithm used to solve the problems of this article is based on a generalized augmented Lagrangian method for the solution of nonlinear (semidefinite) programs described in [9, 16]. Here we briefly recall it and show how it can be extended for the solution of the optimization problems introduced in Sections 2–4.
The goal of the algorithm is to solve general nonlinear semidefinite optimization problems of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &\text{subject to} \\ &\quad \mathcal{G}(x) \preccurlyeq 0\,; \end{aligned} \tag{20}$$

here $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{G}(x) : \mathbb{R}^n \to \mathbb{S}^m$ are twice continuously differentiable mappings and $\mathbb{S}^m$ is the space of symmetric $(m\times m)$-matrices.
The algorithm is based on a choice of a smooth modified barrier function $\Phi_p : \mathbb{S}^m \to \mathbb{S}^m$, depending on a parameter $p > 0$, that satisfies a number of assumptions (see [9]) guaranteeing, in particular, that

$$\mathcal{G}(x) \preccurlyeq 0 \Leftrightarrow \Phi_p(\mathcal{G}(x)) \preccurlyeq 0\,.$$

Thus for any $p > 0$, problem (20) has the same solution as the following "augmented" problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &\text{subject to} \\ &\quad \Phi_p(\mathcal{G}(x)) \preccurlyeq 0\,. \end{aligned} \tag{21}$$

A typical choice of Φ_p is

$$\Phi_p(\mathcal{G}(x)) = -p^2(\mathcal{G}(x) - pI)^{-1} - pI\,. \tag{22}$$

The Lagrangian of (21) can be viewed as a (generalized) augmented Lagrangian of (20):

$$F(x, U, p) = f(x) + \langle U, \Phi_p\left(\mathcal{G}(x)\right)\rangle_{\mathbb{S}_m}\,; \tag{23}$$

here $U \in \mathbb{S}^m$ is a Lagrangian multiplier associated with the inequality constraint. The algorithm is defined as follows:

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

$$\begin{aligned} &(i) && \text{Find } x^{k+1} \text{ satisfying } \|\nabla_x F(x^{k+1}, U^k, p^k)\| \le \alpha^k \\ &(ii) && U^{k+1} = D_{\mathcal{G}}\Phi_p(\mathcal{G}(x^{k+1}); U^k) \\ &(iii) && p^{k+1} \le p^k\,, \quad \alpha^{k+1} < \alpha^k\,. \end{aligned}$$

The unconstrained minimization problem in step (i) is approximately solved by modified Newton's method. Multiplier and penalty update strategies, as well as local and global convergence properties under standard assumptions are studied extensively in [16]. Let us only mention that, imposing standard assumptions, one can prove that any cluster point of the sequence

$\{(x^k, U^k)\}_{k>0}$ generated by the algorithm is a KKT point of problem (20). The proof given in [16] is an extension of results by Polyak [14] and Breitfeld and Shanno [5].
The algorithm, as implemented in the code PENNON [9], proved to be very efficient for the solution of the dual problems of type (15) and (16); see [13] for a comparison of various linear SDP solvers when solving (16). In the following we discuss how to solve the primal optimization problems of type (18) and (19) by variants of Algorithm *1*. We offer two alternatives: a reduced and a direct approach.

Remark 1 Note that problem (21) covers problems with several matrix inequalities as well as problems subject to scalar inequality constraints of the form $g_i(x) \leq 0,\ i = 1, \ldots, k$. Writing the augmented Lagrangian explicitly for this case, we obtain:

$$F(x, U, w, p) = f(x) + \sum_{i=1}^{l} \langle U_i, \Phi_p\left(\mathcal{G}_i(x)\right)\rangle_{\mathbb{S}_{m_i}} + \sum_{i=1}^{k} w_i \varphi_p\left(g_i(x)\right)$$

where φ is the scalar version of Φ and $w \in \mathbb{R}^k$ the associated multiplier.

5.1 *The reduced approach*

Many approaches for the solution of optimal design problems are based on reduced formulations, which are solely defined in the design variables. Often the reduced problems are solved by first order optimization algorithms, where the calculations of the first order derivatives are based on solutions of adjoint problems. In this section we want to describe a similar approach. However, rather than restricting ourselves to a first order algorithm, we will demonstrate how second order derivatives can be efficiently calculated and exploited by a variant of Algorithm *1*.
We start with the derivation of reduced formulations for the stress constrained problem (18); the corresponding formulation for the displacement constrained problem (19) would be similar. It is easily seen that for positive $\underline{\rho}$ the stiffness matrix $A(E)$ is positive definite. Thus we can eliminate the state variable u by substituting $u := A(E)^{-1} f$ and formulate the reduced problem as

$$\min_{E} \sum_{i=1}^{m} \mathrm{Tr}(E_i) \tag{24}$$

subject to

$$E_i \succeq 0, \quad i = 1, \ldots, m$$

$$\underline{\rho} \leq \mathrm{Tr}(E_i) \leq \overline{\rho}, \quad i = 1, \ldots, m$$

$$f^T (A(E))^{-1} f \leq \gamma$$

$$\sum_{k=1}^{nig} \|E B_{i,k} (A(E))^{-1} f\|^2 \leq s_\sigma \overline{\rho}^2, \qquad i = 1, \ldots, m\,.$$

Obviously, the problem is an instance of the general optimization problem (20). Nevertheless it is not recommendable to apply Algorithm *1* directly. The reason is twofold: First, the Hessian of the augmented Lagrangian associated with problem (24) is a large dense matrix and the algorithm may run out of memory. Second, Algorithm *1* does not maintain the feasibility of inequalities strictly throughout the optimization process. Consequently, the global stiffness matrix could become indefinite and the algorithm may fail.
The first issue can be resolved by the use of approximate Newton's method for the solution of step (*i*) in Algorithm *1*. Recently, the authors have successfully implemented and tested a version of Algorithm *1*, where the solution of the Newton system is based on Krylov type methods (see [11]). Instead of calculating the full Hessian of the augmented Lagrangian F, this algorithm requires just Hessian-vector products.

In order to get rid of the second difficulty mentioned above, we treat the inequalities that should be strictly feasible during the optimization process by a classic barrier function. For this reason we introduce an additional matrix inequality

$$\mathcal{S}(x) \preccurlyeq 0$$

in problem (21) and define the augmented Lagrangian

$$\widetilde{F}(x, U, p, s) = f(x) + \langle U, \Phi_p(\mathcal{G}(x))\rangle_{\mathbb{S}_m} + s\Phi_{\text{bar}}(\mathcal{S}(x))\,, \tag{25}$$

where s is a barrier parameter and Φ_{bar} can be defined, for example, by

$$\Phi_{\text{bar}}(\mathcal{S}(x)) = -\log\det(-\mathcal{S}(x)).$$

5.2 *The direct approach*

As an alternative to the reduced approach presented in the preceding section, one can try to solve problem (18) directly. We changed the concept of equality handling in our algorithm by treating them directly on the level of the subproblem; this concept is successfully used in modern primal-dual interior point algorithms (see, e.g., [18, 19]). Consider the optimization problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &\text{subject to} \\ &\qquad \mathcal{G}(x) \preccurlyeq 0 \\ &\qquad h(x) = 0\,, \end{aligned} \tag{26}$$

where f, $\mathcal{G}$ and $\mathcal{S}$ are defined as in the previous sections and $h : \mathbb{R}^n \to \mathbb{R}^d$ represents a set of equality constraints. Then we define the augmented Lagrangian

$$\overline{F}(x, U, v, p, s) = f(x) + \langle U, \Phi_p(\mathcal{G}(x))\rangle_{\mathbb{S}_m} + v^\top h(x)\,,$$

where U, Φ, p are defined as before and $v \in \mathbb{R}^d$ is the vector of Lagrangian multipliers associated with the equality constraints. Now, on the level of the subproblem, we attempt to find an approximate solution of the following system (in x and v):

$$\begin{aligned} \nabla_x \overline{F}(x, U, v, p) &= 0 \\ h(x) &= 0\,, \end{aligned} \tag{27}$$

where the penalty parameter p and the multiplier U are fixed. In order to solve systems of type (27), we apply the damped Newton method. Descent directions are calculated utilizing the factorization routine MA27 from the Harwell subroutine library [7] in combination with an inertia correction strategy as described in [19]. Moreover, the step length is derived using an augmented Lagrangian merit function defined as

$$\overline{F}(x, U, v, p) + \frac{1}{2\mu}\|h(x)\|_2^2$$

along with an Armijo rule.

References

[1] W. Achtziger and Ch. Kanzow. Mathematical programs with vanishing constraints: Optimality conditions and constraint qualifications. Technical Report No. 263, Institute of Applied Mathematics and Statistics, University of Würzburg, Germany, Nov. 2005.

[2] 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.

[3] M. Bendsøe and O. Sigmund. *Topology Optimization. Theory, Methods and Applications*. Springer-Verlag, Heidelberg, 2002.

[4] 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.

[5] 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.

[6] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, New York, Oxford, 1978.

[7] I. S. Duff and J. K. Reid. MA27—A set of Fortran subroutines for solving sparse symmetric sets of linear equations. Tech. Report R.10533, AERE, Harwell, Oxfordshire, UK, 1982.

[8] M. Kočvara, F. Leibfritz, M. Stingl, and D. Henrion. A nonlinear SDP algorithm for static output feedback problems in COMPlib. LAAS-CNRS research report no. 04508, LAAS, Toulouse, 2004.

[9] M. Kočvara and M. Stingl. PENNON—a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18(3):317–333, 2003.

[10] M. Kočvara and M. Stingl. Solving nonconvex sdp problems of structural optimization with stability control. *Optimization Methods and Software*, 19(5):595–609, 2004.

[11] M. Kočvara and M. Stingl. On the solution of large-scale SDP problems by the modified barrier method using iterative solvers. *Math. Program., Ser. B*, 2006. DOI: 10.1007/s10107-006-0029-9.

[12] M. Kočvara and J. Zowe. Free material optimization: An overview. In A.H. Siddiqi and M. Kočvara, editors, *Trends in Industrial and Applied Mathematics*, pages 181–215. Kluwer Academic Publishers, Dordrecht, 2002.

[13] H. Mittelmann. Benchmarks for optimization software; as of November 26, 2005. Available at http://plato.la.asu.edu/bench.html.

[14] R. Polyak. Modified barrier functions: Theory and methods. *Math. Program.*, 54:177–222, 1992.

[15] U. Ringertz. On finding the optimal distribution of material properties. *Structural Optimization*, 5:265–267, 1993.

[16] 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.

[17] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11 & 12:625–653, 1999. Available at `http://fewcal.kub.nl/sturm/`.

[18] R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.

[19] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math. Program.*, 106:25–57, 2006.

[20] R. Werner. *Free Material Optimization. Mathematical Analysis and Numerical Solution.* PhD thesis, Inst. of Applied Mathematics, University of Erlangen, 2000.

[21] J. Zowe, M. Kočvara, and M. Bendsøe. Free material optimization via mathematical programming. *Math. Program., Ser. B*, 79:445–466, 1997.