A Feasible Sequential Convex Programming Method Applied to Free Material Optimization

Sonja Lehmann[1], **Klaus Schittkowski**[1], **Christian Zillober**[2]

[1] Department of Computer Science, University of Bayreuth, Germany
[2] Department of Mathematics, University of Würzburg, Germany

Abstract

We propose a modification of a sequential convex programming (SCP) method that ensures feasibility subject to a given set of convex inequality constraints. The resulting procedure is called feasible sequential convex programming method (FSCP). FSCP expands the standard analytical subproblem which is convex, separable and consists of inverse terms, by these nonlinear, convex feasibility constraints. It is guaranteed that objective function and the remaining constraints are evaluated only at feasible iterates. A line search based on an augmented Lagrangian merit function is performed to guarantee global convergence, i.e., the approximation of a KKT point. Our main motivation is to solve free material optimization (FMO) problems, i.e., generalized topology optimization problems. Design variables are the material properties represented by elasticity tensors or elementary material matrices, respectively, based on a given finite element discretization. Material properties are as general as possible, i.e., anisotropic, with the only restriction that the elasticity tensors are positive definite throughout the algorithm, to guarantee a positive definite global stiffness matrix for computing design constraints. The tensors may be arbitrarily small in case of vanishing material. Numerical results are presented with up to 120.000 variables and 60.000 constraints.

Supported by FP-6 STREP 30717 PLATO-N (Aeronautics and Space),
PLATO-N - A PLAtform for Topology Optimisation incorporating Novel, Large-Scale, Free-Material Optimisation and Mixed Integer Programming Methods

1 Introduction

A strictly *feasible sequential convex programming* algorithm (FSCP) is presented. The goal is to generate an iteration sequence which is strictly feasible subject to a given subset of constraints, called the feasibility constraints, while the remaining constraints may be violated. Typical applications are square roots or logarithmic functions of analytical expressions needed to evaluate objective function or constraints, where the arguments of roots or logarithmic functions are nonlinear terms. We proceed from the following formulation,

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j(x) \leq 0, & j = 1, \ldots, m \\ & e_j(x) \leq 0, & j = 1, \ldots, m_f \end{array} \tag{1}$$

All functions must be smooth, i.e., twice continuously differentiable on the whole $\mathbb{R}^n$. Feasibility constraints are denoted by $e_j(x)\,,\ j = 1, \ldots, m_f$. It is assumed that the remaining constraints $c_j(x)\,,\ j = 1, \ldots, m_c$, and the objective function $f(x)$ can only be evaluated on the feasible set

$$F \ := \ \{x \in \mathbb{R}^n \mid e_j(x) \leq 0,\ j = 1, \ldots, m_f\}\,. \tag{2}$$

To simplify the subsequent analysis and the notation, we suppress equality and box constraints. As usual, nonlinear equality constraints are linearized and box constraints are handled as nonlinear inequality constraints throughout the theoretical investigations. Our implementation takes them into account.

Our main motivation is to solve free material optimization (FMO) problems, see Bendsøe et al. [7], which is an extension of topology optimization, see Bendsøe and Sigmund [8]. Within a given design space, topology optimization finds the optimal material layout for a given set of loads and given material. An underlying finite element discretization is used to decide in each element whether to use material or not. The stiffness of the structure is defined by the so-called compliance function, which measures the displacement of the structure under loads. The smaller the compliance the stiffer the resulting structure. In addition, the total amount of material is bounded. To prevent numerical instabilities, i.e., checkerboard phenomena or grey zones, a filter can be used, see Ni, Zillober and Schittkowski [36]. Topology optimization problems are large-scale nonlinear programs, that can be solved efficiently by appropriate algorithms, e.g., the method of moving asymptotes of Svanberg [48] or Sigmund [45]. The resulting structure consists of void and material.

Free material optimization (FMO) is introduced in a series of papers, e.g., Bendsøe et al. [7], Bendsøe and Díaz [6], Bendsøe [5] and Zowe, Kočvara and Bendsøe [62]. FMO tries to find the best mechanical structure with respect to one or more given load cases in the sense that a design criterion, e.g., minimal weight or maximal stiffness, is obtained. The material properties as well as the material distribution in the available space are included in the optimization process. Therefore, FMO is a generalization of topology optimization. As shown, e.g., by Kočvara and Stingl [30], the FMO problem can be formulated for a given set of loads by a nonlinear semidefinite programming (NSDP) problem based on a finite element discretization. The common FMO formulation is to minimize the maximal compliance $f_j^T K^{-1}(E) f_j$ for load f_j, $j = 1, \ldots, l$, where l is the number of load cases and $K(E)$ the global stiffness matrix. A more detailed description is found in Hörnlein, Kočvara and Werner [24] and Kočvara and Zowe [31]. As a measure of the material stiffness, we use the traces of the elasticity matrices E_i, which are the design or optimization variables. The elasticity matrices E_i, $i = 1, \ldots, m$, fulfill the basic requirements of linear elasticity, i.e., they are symmetric and positive semidefinite. Moreover, volume constraints and box constraints preventing singularities are introduced.

The strictly *feasible sequential convex programming* (FSCP) method is an extension of the *sequential convex programming* (SCP) method, which is frequently used in mechanical engineering. The algorithm approximates the optimal solution by solving a sequence of convex and separable subproblems, where a line search procedure with respect to the augmented Lagrangian merit function is used for guaranteeing global convergence. SCP was originally designed for solving structural mechanical optimization problems and it is often applied in the field of topology optimization. Due to the fact that in some special cases, typical structural constraints become linear in the inverse variables, a suitable substitution is applied, which is expected to linearize these functions in some sense, see Zillober, Schittkowski and Moritzen [60].

SCP methods are derived from the optimization method CONLIN, see Fleury and Braibant [20] and Fleury [19]. The algorithm formulates convex and separable subproblems by linearizing the problem functions with respect to reciprocal variables, if the partial derivative is negative in the current iterate. Otherwise, it is linearized in the original sense. Since the starting point must be close to a solution and since the method can oscillate, Svanberg [48] extended the algorithm proposing the method of moving asymptotes (MMA). Two flexible asymptotes, a lower and an upper one, are introduced truncating the feasible region. The functions are linearized with respect to one of the asymptotes, depending on the sign of the partial derivative. The resulting convex and separable subproblems can be solved efficiently due to their special structure. The asymptotes are adapted in each iteration, to control the curvature of the Lagrangian function and thus influence the convergence.

SCP is an extension of MMA including a line search procedure to stabilize the algorithm and too achieve a global convergence result. Iterates are excepted subject to a merit function, which combines the descent of the objective function and the feasibility in a suitable way. The stepsize is reduced until a descent in the merit function, e.g., the augmented Lagrangian function, is obtained. An active set strategy can be applied to reduce the size of the subproblem, saving computational effort. The program SCPIP30 is an efficient implementation of SCP, where the sparse structure of the gradients and the Hessian is taken into account. Some comparative numerical tests of SCP, sequential quadratic programming (SQP) and some other nonlinear programming codes are available for test problems from mechanical structural optimization, see Schittkowski, Zillober and Zotemantel [44]. For the resulting SCP method global convergence is shown, see Zillober [55, 57].

Svanberg [49, 50, 51] presented extensions of the MMA method which are also globally convergent due to additional inner iterations ensuring descent of the given functions over the approximated ones. Ni [35] introduced a new version of MMA, where the convex subproblems are in addition restricted by a trust region. In contrast to MMA and SCP, his approach is only applicable for box constraints, while equality and inequality constraints cannot be handled. Ertel [15] combined the method of moving asymptotes with the filter approach proposed by Fletcher and Leyffer [17]. An iterate is accepted, if a descent in the objective function or a reduction of the constraint violation is obtained. Otherwise, the point is rejected and a new subproblem is generated by reducing the distance between the asymptotes. Filter methods induce a non-monotone iteration sequence. A convergence proof for a SQP-filter method is given by Fletcher, Toint and Leyffer [18].

Stingl, Kočvara and Leugering [47] proposed a generalization of SCP for semidefinite programs of the form

$$\begin{array}{lll} \min\limits_{Z} & f(Z) & Z \in \mathbb{S}^n \\ \text{s.t.} & c_j(Z) \le 0, & j = 1, \ldots, m_c \\ & Z - \underline{Z} \succeq 0 & \\ & \overline{Z} - Z \succeq 0 & \end{array} \tag{3}$$

where $\mathbb{S}^n$ denotes the space of symmetric matrices of size n. The algorithm creates a sequence of first order block-separable convex approximations. In contrast to MMA and SCP, the method uses constant asymptotes. Moreover, a line search procedure is applied to ensure a sufficient descent of objective function values. The resulting semidefinite subproblem can be solved efficiently due to its specific structure by appropriate solvers, see Kočvara and Stingl [28]. Global convergence of the resulting algorithm can be shown, see Stingl, Kočvara and Leugering [47].

Since SCP methods are often applied to solve practical topology optimization problems, they are also investigated for solving free material optimization problems. To guarantee positive definit elasticity tensors, feasibility constraints are introduced to guarantee a positive definite total stiffness matrix. The SCP method of Zillober [55] is extended to take feasibility constraints into account. They are supposed to be convex and passed to the convex, analytical and separable subproblem which must be solved in each step of an MMA algorithm. A line search is performed to ensure global convergence. The corresponding convergence proof of the resulting feasible sequential convex programming method is given for convex feasibility constraints.

In the literature, several feasible optimization methods can be found. The most important existing ones are feasible interior point methods, projection methods and feasible direction methods.

Feasible interior point methods start from the interior of the feasible region and compute an iteration sequence that approaches the boundary. A subclass are barrier methods, where a barrier parameter combines the constraints and the objective function. Typically the barrier function is only defined on the feasible region and tends to infinity at the boundary, e.g.,

$$f(x) + \mu \sum_{i=1}^{m_f} \ln\left(-e_j(x)\right), \tag{4}$$

where $\mu \in \mathbb{R}^+$ is the barrier parameter. Starting with a large μ, it is reduced iteratively such that solutions near the boundary can be obtained. These methods are especially successful for convex optimization problems, see Jarre and Stoer [25].

Another class of feasible optimization methods are projection methods. In each iterate $x^{(k)}$, the algorithms compute a search direction $d^{(k)} \in \mathbb{R}^n$ and project the resulting point $x^{(k)} + d^{(k)}$ on the boundary of the feasible region, if necessary. The projected point on the boundary is denoted by $x_P^{(k)} \in \mathbb{R}^n$. The projected search direction $d_P^{(k)} \in \mathbb{R}^n$ consists of two components. Inside the interior of the feasible region, the projected search direction is given by $d^{(k)}$. The second part is described by the segment of the boundary between the intersection point of $d^{(k)}$ with the boundary and the projection point $x_P^{(k)}$. A line search is performed along the projected search direction $d_P^{(k)}$. To ensure feasibility, the problems have to be convex. The effort to compute the projection depends on the algorithm and on the constraints of the optimization problem. Some popular projection methods are presented by Rosen [41, 42] and by Polak [40]. Projection methods are often combined with other efficient nonlinear optimization methods to compute the descent direction $d^{(k)}$. Jian, Zhang and Xue [26] developed a feasible SQP method in combination with a projection method.

Starting from a feasible point, feasible direction methods compute a feasible direction $d^{(k)}$ which ensures the existence of a descent step inside the feasible domain. If necessary, a line search is applied. The first algorithms go back to Zoutendijk [61]. A convergence proof can be given for convex constraints, see Bertsekas [10]. To improve the performance and to get a higher convergence order, a quadratic subproblem similar to SQP methods. The resulting search direction may not be feasible, since active constraints can lead to a search direction tangential to the feasible region, see Panier and Tits [38]. Thus, a correction is determined by tilting the original direction towards the feasible region. To ensure fast convergence near a solution an additional search direction is computed by bending. An extended line search is performed along the search arc consisting of all three directions, such that feasibility and a sufficient descent in the objective function is guaranteed. The computational complexity per iteration of the feasible SQP methods is significantly higher compared to usual SQP methods. In state-of-the-art methods the computational complexity has been reduced.

Moreover, feasible direction interior point algorithms (FDIP) are developed. In general, interior point methods (IPM) compute in each iteration a Newton descent direction by solving a linear system of equations. The resulting search direction might not be a feasible direction. Therefore, a second linear system is formulated where the right hand side is perturbed ensuring a feasible direction. Some of the FDIP methods solve a third linear system to ensure superlinear convergence near a stationary point. Analogue to feasible SQP methods, a line search along the search arc is performed to ensure both feasibility and a descent in the objective function. Several feasible direction interior point methods are given in the literature, e.g., Panier, Tits and Herskovits [39], Herskovits [21, 22], Bakhtiari and Tits [3] and Zhu [54]. The globally and locally superlinear convergent algorithm FAIPA belongs to the latest algorithms proposed, see Herskovits, Aroztegui, Goulart and Dubeux [23]. In each iteration, a feasible descent arc is computed such that a new interior point with a lower objective function value can be found. Three linear systems have to be solved in each iteration, where the matrices remain unchanged.

Most known feasible optimization methods guarantee evaluation of objective function values at feasible arguments and try to remain as close to the feasible domain as possible, but constraint functions may be computed at infeasible iterates, e.g., to check whether they are violated or not. Our own approach differs from all other feasible or feasible direction methods found in the literature, since we distinguish explicitly between feasibility constraints, which may be violated and which must be satisfied before requesting an evaluation of objective function and all other constraints, also before calling corresponding partial derivatives. In addition, many existing implementations are based on sequential quadratic programming (SQP) methods and not suitable for solving large optimization problems.

Our main design goal is to develop an algorithm for solving practical optimization problems in mechanical engineering. The new method is based on the method of moving asymptotes introduced by Svanberg [48] and the sequential convex programming (SCP) algorithm of Zillober [56]. Both algorithms a widely used in industry and academia to solve mechanical structural optimization problems, especially in topology optimization. The feasible sequential convex programming method (FSCP) is introduced in Section 2. The SCP algorithm of Zillober [56] is extended and adapted such that feasibility with respect to a subset of constraints is guaranteed in each iteration. A global convergence proof based on the usage of an augmented Lagrangian merit function also used by Schittkowski [43] in the frame of an SQP algorithm, is given in Section 3. The FMO model is outlined in Section 4, where elasticity tensors must be positive definite in order to evaluate a valid global stiffness matrix. Two formulations to replace semidefinite constraints by nonlinear ones are proposed and some numerical results are presented.

2 A Feasible Sequential Convex Programming Method

There exist optimization applications where the model functions are only defined on a certain domain specified by other constraints. Since standard nonlinear optimization methods cannot ensure feasibility during the solution process, these problems cannot be solved appropriately. Typical examples are logarithmic or square root functions, e.g.,

$$\begin{aligned} c_1(x) &:= \log(e_1(x)), \\ c_2(x) &:= \sqrt{e_2(x)}, \end{aligned}$$

where $e_1(x)$ and $e_2(x)$ are any nonlinear functions. To evaluate $c_1(x)$ and $c_2(x)$, the constraints $e_1(x) > 0$ and $e_2(x) > 0$ must be satisfied.

We present an extended version of the SCP-algorithm of Zillober [55, 57] guaranteeing feasibility of a given subset of constraints in each iteration, i.e., of $e_j(x) \le 0,\ j = 1, \ldots, m_f$, which called the feasibility constraints. The resulting method is denoted as feasible sequential convex programming method (FSCP). It is assumed that the constraints $c_j(x)$, $j = 1, \ldots, m_c$, as well as the objective function $f(x)$ can only be evaluated at an $x \in \mathbb{R}^n$ with $e_j(x) \le 0$, $j = 1, \ldots, m_f$, i.e., at a parameter vector satisfying the feasibility constraints, see (1). Objective function $f(x)$ and constraints $c_j(x)$, $j = 1, \ldots, m_c$, are supposed to be continuously differentiable on the feasible set

$$F := \{x \in \mathbb{R}^n \mid e_j(x) \le 0,\ j = 1, \ldots, m_f\}. \tag{5}$$

The feasibility functions $e_j(x)$, $j = 1, \ldots, m_f$, are supposed to be convex and at least twice continuously differentiable on $\mathbb{R}^n$. Thus, F is convex which is important to guarantee feasibility, if the stepsize is reduced during a line search procedure.

Proceeding from a feasible starting point with respect to feasibility constraints, i.e., $x^{(0)} \in F$, we generate a sequence of convex subproblems, which are easy to solve due to their special structure even if the number of variables and constraints becomes very large. Moreover, the nonlinear constraints $e_j(x)$, $j = 1, \ldots, m_f$, are added to ensure feasibility of all iterates. Proceeding from index sets $I_+^{(k)}$ and $I_-^{(k)}$ defined by

$$I_+^{(k)} := \left\{ i = 1, \ldots, n \;\middle|\; \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \ge 0 \right\} \tag{6}$$

and

$$I_-^{(k)} := \left\{ i = 1, \ldots, n \;\middle|\; \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} < 0 \right\}, \tag{7}$$

we approximate objective function $f(x)$ and constraints $c_j(x)$, $j = 1, \ldots, m_c$, by convex and separable functions

$$
\begin{aligned}
f^{(k)}(x) \;:=\;& f\left(x^{(k)}\right) \\
&+ \sum_{I_+^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}}\right) \\
&- \sum_{I_-^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}}\right) \qquad (8)\\
&+ \sum_{I_+^{(k)}} \tau \frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} + \sum_{I_-^{(k)}} \tau \frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}}
\end{aligned}
$$

with $L_i^{(k)} < x_i < U_i^{(k)}$ and $\tau > 0$, see Svanberg [48] and Zillober [59]. In each iteration k, objective function and the inequality constraints are linearized with respect to the inverse variables $1/(U_i^{(k)} - x_i)$ or $1/(x_i - L_i^{(k)})$ depending on the sign of the corresponding partial derivative. To ensure strict convexity of $f^{(k)}(x)$ and to get an unique solution of the subproblem, the additional regularization parameter $\tau > 0$ is introduced, see Zillober [59].

The nonlinear inequality constraints $c_j(x)$, $j = 1, \ldots, m_c$, are approximated then by

$$
\begin{aligned}
c_j^{(k)}(x) \;:=\;& c_j\left(x^{(k)}\right) \\
&+ \sum_{I_+^{(j,k)}} \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}}\right) \qquad (9)\\
&- \sum_{I_-^{(j,k)}} \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}}\right)
\end{aligned}
$$

with $L_i^{(k)} < x_i < U_i^{(k)}$ and

$$
I_+^{(j,k)} := \left\{ i = 1, \ldots, n \;\middle|\; \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \geq 0 \right\}, \qquad (10)
$$

$$
I_-^{(j,k)} := \left\{ i = 1, \ldots, n \;\middle|\; \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} < 0 \right\}. \qquad (11)
$$

Note that $f^{(k)}(x)$ and $c_j^{(k)}(x)$ are first-order approximations of $f(x)$ and $g_j * x)$, respectively. We implicitly assume that the nonlinear functions $e_1(x)$, ..., $e_{m_f}(x)$ and their derivatives are much easier to evaluate than the functions and gradients of $f(x)$ and $c_1(x)$,

..., $c_{m_c}(x)$. At each iteration k, we have to solve the subproblem

$$\begin{array}{lll} \min\limits_{x} & f^{(k)}(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j^{(k)}(x) \leq 0, & j = m_e+1, \ldots, m_c \\ & e_j(x) \leq 0, & j = 1, \ldots, m_f \\ & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i = 1, \ldots, n \end{array} \tag{12}$$

The box constraints are defined by

$$\underline{x}_i^{(k)} := x_i^{(k)} - \omega \left(x_i^{(k)} - L_i^{(k)} \right), \; i = 1, \ldots, n \tag{13}$$

and

$$\overline{x}_i^{(k)} := x_i^{(k)} + \omega \left(U_i^{(k)} - x_i^{(k)} \right), \; i = 1, \ldots, n \tag{14}$$

where $\omega \in]0,1[$ is a given constant. The asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$ must be updated carefully to remain feasible, see also Zillober [56].

Definition 2.1. *A sequence of asymptotes $\left\{L^{(k)}\right\}$, $\left\{U^{(k)}\right\}$ is called feasible subject to a bounded sequence $\left\{x^{(k)}\right\}$ with $L_i^{(k)} < x_i^{(k)} < U_i^{(k)}$, $i = 1, \ldots, n$, if there exists a $\xi > 0$ and $L_{min}, U_{max} \in \mathbb{R}$, $L_{min} < U_{max}$ such that*

1. *$L_i^{(k)} \leq x_i^{(k)} - \xi$, $U_i^{(k)} \geq x_i^{(k)} + \xi$, for all $i = 1, ..., n$, and $k \geq 0$.*
2. *$L_i^{(k)} \geq L_{min}$, $U_i^{(k)} \leq U_{max}$, $\forall\, k \geq 0$, $i = 1, \ldots, n$.*

The first part of this definition prevents that the slope of the approximations from becoming too steep. Under no circumstances, the iterates $x^{(k)}$ are allowed to approach the asymptotes too close. Moreover, $x^{(k)}$ remains in the set $F_X^{(k)} \subset F$, where

$$F_X^{(k)} := F \cap X^{(k)} \tag{15}$$

$$X^{(k)} := \left\{ x \in \mathbb{R}^n \;\middle|\; \underline{x}^{(k)} \leq x \leq \overline{x}^{(k)} \right\}. \tag{16}$$

To assure global convergence of the algorithm, we apply a line search procedure subject to a differentiable augmented Lagrangian merit function, which is also used in Schittkowski [43] to prove convergence of an SQP algorithm,

$$\begin{aligned} \Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} := \; & f(x) + \sum_{j=m_e+1}^{m_c} \begin{cases} (y_c)_j\, c_j(x) \; + \; \dfrac{(\rho_c)_j}{2} c_j^2(x), & \text{if } -\dfrac{(y_c)_j}{(\rho_c)_j} \leq c_j(x) \\ -\dfrac{(y_c)_j^2}{2\,(\rho_c)_j}, & \text{otherwise} \end{cases} \\ & + \sum_{j=1}^{m_f} \begin{cases} (y_e)_j\, e_j(x) \; + \; \dfrac{(\rho_e)_j}{2} e_j^2(x), & \text{if } -\dfrac{(y_e)_j}{(\rho_e)_j} \leq e_j(x) \\ -\dfrac{(y_e)_j^2}{2\,(\rho_e)_j}, & \text{otherwise} \end{cases} \end{aligned} \tag{17}$$

for a given set of penalty parameters

$$\rho := \begin{pmatrix} \rho_c \\ \rho_e \end{pmatrix} \tag{18}$$

with $(\rho_c)_j > 0,\ j = 1, \ldots, m_c$, and $(\rho_e)_j > 0$, $j = 1, \ldots, m_f$. We denote the Lagrangian multipliers of the constraints $c_j(x)$, $j = 1, \ldots, m_c$, and of the feasibility constraints $e_j(x)$, $j = 1, \ldots, m_f$, by

$$y := \begin{pmatrix} y_c \\ y_e \end{pmatrix} \tag{19}$$

with $y_c = \left((y_c)_1, \ldots, (y_c)_{m_c}\right)^T \in \mathbb{R}^{m_c}$, and $y_e = \left((y_e)_1, \ldots, (y_e)_{m_f}\right)^T \in \mathbb{R}^{m_f}$.

The sufficient descent property of the merit function depends on parameters $\eta_i^{(k)}$, $i = 1, \ldots, n$, by which the curvature of the approximated objective function $f^{(k)}(x)$ is estimated,

$$\eta_i^{(k)} := \begin{cases} \left(\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau\right) \dfrac{2U_i^{(k)} - z_i^{(k)} - x_i^{(k)}}{\left(U_i^{(k)} - z_i^{(k)}\right)^2}, & \text{if } i \in I_+^{(k)} \\ -\left(\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau\right) \dfrac{-2L_i^{(k)} + z_i^{(k)} + x_i^{(k)}}{\left(z_i^{(k)} - L_i^{(k)}\right)^2}, & \text{otherwise} \end{cases} \tag{20}$$

and we define

$$\eta^{(k)} := \min_{i=1,\ldots,n} \eta_i^{(k)}. \tag{21}$$

The penalty parameters are updated in the following manner.

Algorithm 1. Update of penalty parameters
Let k be the iteration index and $x^{(k)} \in \mathbb{R}^n$ the current primal and $y^{(k)} \in \mathbb{R}^{m_c}$ the current dual variable. Moreover, let $\left(z^{(k)}, v^{(k)}\right)$ be the solution of subproblem (12) defined in $x^{(k)}$ and $\rho_j^{(k-1)}$ the previous penalty parameter.
If $j \in J_c^{(k)}$, $c_j(x^{(k)}) < 0$, and $v_j^{(k)} < y_j^{(k)}$ then

$$\rho_j^{(k)} := \max\left(\rho_j^{(k-1)}, -\frac{2(y_j^{(k)} - v_j^{(k)})}{c_j(x^{(k)})}\right) \tag{22}$$

else

$$\rho_j^{(k)} := \max\left(\rho_j^{(k-1)}, \frac{10(m_c + m_f)\left|y_j^{(k)}\left(v_j^{(k)} - y_j^{(k)}\right)\right|}{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2}\right) \tag{23}$$

Starting from $\rho_j^{(0)} > 0$ for $j = 1, \ldots, m_c + m_f$, the series of penalty parameters is non-decreasing. They are chosen to guarantee that the property

$$\nabla \Phi_{\rho^{(k)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \leq -\frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{2} \tag{24}$$

subject to the augmented Lagrangian merit function (17). Note that the choice of $\rho_j^{(k)}$ is a bit overestimated. The implementation must be carefully adapted to prevent a too large increase of the penalty parameters especially in the beginning of the algorithm, see Zillober [57], and to allow also a decrease temporarily.

If we denote the primal solution of subproblem (12) by $z^{(k)}$ and the dual solution by $v^{(k)}$, we start an iterative subprocess at $\sigma^{(k,0)} := 1$ and reduce it until

$$\Phi_{\rho^{(k)}} \left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)} \right) \leq \Phi_{\rho^{(k)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \mu \sigma^{(k,i)} \nabla \Phi_{\rho^{(k)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}, \tag{25}$$

where $\mu \in (0,1)$ is constant and where the search direction $d^{(k)} \in \mathbb{R}^{n+m_c}$ is given by

$$d^{(k)} := \begin{pmatrix} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{pmatrix}. \tag{26}$$

for the fist time, $i = 0, 1, 2, \ldots$, see Armijo [2] or Ortega and Rheinboldt [37].

Now we are able to summarize the feasible SCP algorithm.

Algorithm 2. Feasible Sequential Convex Programming

Step 0: *Choose feasible starting point $x^{(0)} \in F$. Set parameters $\xi > 0$, $L_{min} < U_{max}$, $\rho_j^{(0)} > 0$ for $j = 1, \ldots, m_c + m_f$, $\omega \in]0;1[$, $\mu \in (0,1)$, $\beta \in (0,1)$, $\tau > 0$, and $y^{(0)} \geq 0$. Compute $f\left(x^{(0)}\right), \nabla f\left(x^{(0)}\right), c_j\left(x^{(0)}\right), \nabla c_j\left(x^{(0)}\right)$, $j = 1, \ldots, m_c$, and $e_j\left(x^{(0)}\right), \nabla e_j\left(x^{(0)}\right)$, $j = 1, \ldots, m_f$. Set penalty parameters $\left(\rho_c^{(-1)}\right)_j > 0$, $j = 1, \ldots, m_c$, and $\left(\rho_e^{(-1)}\right)_j > 0$, $j = 1, \ldots, m_f$. Set $k := 0$.*

Step 1: *Determine feasible asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$. Let $f^{(k)}(x)$, $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, be defined by (8) and (9). Define $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, by (13) and (14). Formulate (12) for the corresponding iteration k.*

Step 2: *Solve (12). Let $z^{(k)}$ be the optimal solution of subproblem (12) and $v^{(k)}$ the vector of corresponding Lagrangian multipliers.*

Step 3: *If $z^{(k)} = x^{(k)}$, then STOP. $\left(x^{(k)}, v^{(k)}\right)$ is a KKT point of (12).*

Step 4: *Let $d^{(k)} := \begin{pmatrix} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{pmatrix}$, and $\eta^{(k)}$ as defined in (21).*

Let $i = 0$ *and* $\rho^{(k,0)} := \rho^{(k-1)}$.

Step 5: *Compute* $f\left(x^{(k)} + \sigma^{(k,i)}\left(z^{(k)} - x^{(k)}\right)\right), c_j\left(x^{(k)} + \sigma^{(k,i)}\left(z^{(k)} - x^{(k)}\right)\right)$, $j = 1, \ldots, m_c$, *and*$e_j\left(x^{(k)} + \sigma^{(k,i)}\left(z^{(k)} - x^{(k)}\right)\right)$, $j = 1, \ldots, m_f$,.
If (25) is not satisfied, let $\sigma^{(k,i+1)} := \beta\sigma^{(k,i)}$, $i = i + 1$ *and repeat (Armijo).*
Otherwise, $\sigma^{(k)} := \sigma^{(k,i)}$.

Step 6: *Let* $\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} := \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k)} d^{(k)},\ k := k + 1.$

Step 7: *Compute* $\nabla f\left(x^{(k)}\right)$, $\nabla c_j\left(x^{(k)}\right)$, $j = 1, \ldots, m_c$, $\nabla e_j\left(x^{(k)}\right)$, $j = 1, \ldots, m_f$ *and goto Step 1.*

It might happen that the constraints of subproblem (12) become inconsistent. In this case, the subproblem is extended by additional variables, see Zillober [58] for details. Also a procedure for computing feasible asymptotes is listed there. The line search procedure is a simple bisection method until sufficient descent is reached. It is easily extended by a more efficient one taking curvature information into account, e.g., by trying first the minimizer of a quadratic interpolation along the search direction, see Schittkowski **??**. The subset F described by the feasibility constraints is convex, an important assumption for the line search procedure. It is guaranteed that the feasibility constraints are satisfied whenever objective or constraint function values function are to be evaluated.

3 A Global Convergence Theorem

We proceed from the nonlinear program (1) with feasibility constraints and summarize a few technical preliminaries, which are outlined and proved in detail in Lehmann [32].

First, the box constraints of subproblem (12) are written in the form

$$
\begin{aligned}
b_i^{(k)}(x) &:= x_i - \overline{x}_i^{(k)} \leq 0, \quad i = 1, \ldots, n, &(27)\\
b_{n+i}^{(k)}(x) &:= \underline{x}_i^{(k)} - x_i \leq 0, \quad i = 1, \ldots, n. &(28)
\end{aligned}
$$

and the corresponding Lagrangian multipliers are $v_u^{(k)} \in \mathbb{R}^n$ for the upper bounds $b_i^{(k)}(x),\ i = 1, \ldots, n$, and $v_l^{(k)} \in \mathbb{R}^n$ for the lower bounds $b_{i+n}^{(k)}(x),\ i = 1, \ldots, n$. Moreover, we define the Jacobian matrices by

$$
\begin{aligned}
A_{u^{(k)}}(x) &:= \left(\nabla b_1^{(k)}(x), \ldots, \nabla b_n^{(k)}(x)\right) \in \mathbb{R}^{n \times n}, &(29)\\
A_{l^{(k)}}(x) &:= \left(\nabla b_{n+1}^{(k)}(x), \ldots, \nabla b_{2n}^{(k)}(x)\right) \in \mathbb{R}^{n \times n}. &(30)
\end{aligned}
$$

It is easy to see, that

$$
A_{u^{(k)}}(x) = -A_{l^{(k)}}(x) = I, \tag{31}
$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix. We denote the primal solution of subproblem (12) in iteration k by $z^{(k)} \in \mathbb{R}^n$, the dual solution by $v^{(k)} \in \mathbb{R}^{m_c+m_f}$ and define

$$\Delta x^{(k)} \quad := \quad z^{(k)} - x^{(k)}. \tag{32}$$

Moreover, we define index sets

$$J_c(x) \quad := \quad \left\{ j = 1, \ldots, m_c \,\middle|\, -\frac{(y_c)_j}{(\rho_c)_j} \leq c_j(x) \right\} \tag{33}$$

$$\overline{J}_c(x) \quad := \quad \left\{ j = 1, \ldots, m_c \,\middle|\, -\frac{(y_c)_j}{(\rho_c)_j} > c_j(x) \right\} \tag{34}$$

$$J_e(x) \quad := \quad \left\{ j = 1, \ldots, m_f \,\middle|\, -\frac{(y_e)_j}{(\rho_e)_j} \leq e_j(x) \right\} \tag{35}$$

$$\overline{J}_e(x) \quad := \quad \left\{ j = 1, \ldots, m_f \,\middle|\, -\frac{(y_e)_j}{(\rho_e)_j} > e_j(x) \right\} \tag{36}$$

Jacobian matrices are denoted by

$$A_c(x) \quad := \quad (\nabla c_1(x), \ldots, \nabla c_{m_c}(x)) \in \mathbb{R}^{n\times m_c} \tag{37}$$

$$A_e(x) \quad := \quad \left(\nabla e_1(x), \ldots, \nabla e_{m_f}(x)\right) \in \mathbb{R}^{n\times m_f} \tag{38}$$

$$A(x) \quad := \quad (A_c(x), A_e(x)) \in \mathbb{R}^{n\times m_c+m_f} \tag{39}$$

We then define

$$\overline{y}_c := \left((\overline{y}_c)_1, \ldots, (\overline{y}_c)_{m_c}\right)^T \quad \text{with} \quad (\overline{y}_c)_j := \begin{cases} (y_c)_j\,, & \text{if } j \in J_c(x) \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

$$\overline{y}_e := \left((\overline{y}_e)_1, \ldots, (\overline{y}_e)_{m_f}\right)^T \quad \text{with} \quad (\overline{y}_e)_j := \begin{cases} (y_e)_j\,, & \text{if } j \in J_e(x) \\ 0, & \text{otherwise} \end{cases} \tag{41}$$

$$\overline{v}_c := \left((\overline{v}_c)_1, \ldots, (\overline{v}_c)_{m_c}\right)^T \quad \text{with} \quad (\overline{v}_c)_j := \begin{cases} (v_c)_j\,, & \text{if } j \in J_c(x) \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

$$\overline{v}_e := \left((\overline{v}_e)_1, \ldots, (\overline{v}_e)_{m_f}\right)^T \quad \text{with} \quad (\overline{v}_e)_j := \begin{cases} (v_e)_j\,, & \text{if } j \in J_e(x) \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

$$\overline{c}(x) := \left(\overline{c}_1(x), \ldots, \overline{c}_{m_c}(x)\right)^T \quad \text{with} \quad \overline{c}_j(x) := \begin{cases} c_j(x), & \text{if } j \in J_c(x) \\ 0, & \text{otherwise} \end{cases} \tag{44}$$

$$\overline{e}(x) := \left(\overline{e}_1(x), \ldots, \overline{e}_{m_f}(x)\right)^T \quad \text{with} \quad \overline{e}_j(x) := \begin{cases} e_j(x), & \text{if } j \in J_e(x) \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

$$\widehat{c}(x) := \left(\widehat{c}_1(x), \ldots, \widehat{c}_{m_c}(x)\right)^T \quad \text{with} \quad \widehat{c}_j(x) := \begin{cases} c_j(x), & \text{if } j \in J_c(x) \\ -\frac{(y_c)_j}{(\rho_c)_j}, & \text{otherwise} \end{cases} \tag{46}$$

$$\widehat{e}(x) := \left(\widehat{e}_1(x), \ldots, \widehat{e}_{m_f}(x)\right)^T \quad \text{with} \quad \widehat{e}_j(x) := \begin{cases} e_j(x), & \text{if } j \in J_e(x) \\ -\frac{(y_e)_j}{(\rho_e)_j}, & \text{otherwise} \end{cases} \tag{47}$$

$$\widetilde{c}(x) := (\widetilde{c}_1(x), \ldots, \widetilde{c}_{m_c}(x))^T \quad \text{with} \quad \widetilde{c}_j(x) := \begin{cases} c_j(x), & \text{if } j \in J_c(x) \\ -\frac{(y_c)_j}{2(\rho_c)_j}, & \text{otherwise} \end{cases} \tag{48}$$

$$\widetilde{e}(x) := \left(\widetilde{e}_1(x), \ldots, \widetilde{e}_{m_f}(x)\right)^T \quad \text{with} \quad \widetilde{e}_j(x) := \begin{cases} e_j(x), & \text{if } j \in J_e(x) \\ -\frac{(y_e)_j}{2(\rho_e)_j}, & \text{otherwise} \end{cases} \tag{49}$$

Whenever necessary, we add the upper index $^{(k)}$ to denote the k-th iteration step and let

$$J_c^{(k)} := J_c(x^{(k)}) \tag{50}$$

$$\overline{J}_c^{(k)} := \overline{J}_c(x^{(k)}) \tag{51}$$

$$J_e^{(k)} := J_e(x^{(k)}) \tag{52}$$

$$\overline{J}_e^{(k)} := \overline{J}_e(x^{(k)}) \tag{53}$$

The Lagrangian function of the nonlinear program (1) with feasibility constraints and the corresponding gradient subject to the primal variable x is

$$L(x, y) = f(x) + y_c^T c(x) + y_e^T e(x) \tag{54}$$

$$\nabla_x L(x, y) = \nabla f(x) + A_c(x) y_c + A_e(x) y_e \tag{55}$$

and the augmented Lagrangian (17) is now written in the form

$$\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} = f(x) + y_c^T \widetilde{c}(x) + \frac{1}{2}\rho_c^T \overline{c}^2(x) + y_e^T \widetilde{e}(x) + \frac{1}{2}\rho_e^T \overline{e}^2(x) \tag{56}$$

with gradient

$$\nabla \Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \nabla f(x) + A_c(x)\left(\overline{y}_c + \Gamma_c \overline{c}(x)\right) + A_e(x)\left(\overline{y}_e + \Gamma_e \overline{e}(x)\right) \\ \widehat{c}(x) \\ \widehat{e}(x) \end{pmatrix} \tag{57}$$

Note that we are looking for a descent of the merit function subject to the primal and the dual variables.

We denote the Lagrangian function of subproblem (12) in iteration k by $L^{(k)}(z, v)$.

For the primal and dual solution $\left(z^{(k)}, v^{(k)}\right)$ we obtain the KKT conditions

$$\begin{aligned}
\nabla_x L^{(k)}\left(z^{(k)}, v^{(k)}\right) &= \nabla f^{(k)}\left(z^{(k)}\right) + A_{c^{(k)}}\left(z^{(k)}\right) v_c^{(k)} \\
&\quad + A_e\left(z^{(k)}\right) v_e^{(k)} + v_u^{(k)} - v_l^{(k)} = 0 && (58)\\
\left(v_c^{(k)}\right)_j c_j^{(k)}\left(z^{(k)}\right) &= 0, \qquad j = 1, \ldots, m_c && (59)\\
\left(v_e^{(k)}\right)_j e_j\left(z^{(k)}\right) &= 0, \qquad j = 1, \ldots, m_f && (60)\\
\left(v_b^{(k)}\right)_j b_j^{(k)}\left(z^{(k)}\right) &= 0, \qquad j = 1, \ldots, 2n && (61)\\
c_j^{(k)}\left(z^{(k)}\right) &\le 0, \qquad j = 1, \ldots, m_c && (62)\\
e_j\left(z^{(k)}\right) &\le 0, \qquad j = 1, \ldots, m_f && (63)\\
b_j^{(k)}\left(z^{(k)}\right) &\le 0, \qquad j = 1, \ldots, 2n && (64)\\
\left(v_c^{(k)}\right)_j &\ge 0, \qquad j = 1, \ldots, m_c && (65)\\
\left(v_e^{(k)}\right)_j &\ge 0, \qquad j = 1, \ldots, m_f && (66)\\
\left(v_b^{(k)}\right)_j &\ge 0, \qquad j = 1, \ldots, 2n && (67)
\end{aligned}$$

Applying now a Taylor approximation with residual terms $R_{f^{(k)}}(x)$, $R_{c_j^{(k)}}(x)$ and $R_{e_j}(x)$, we obtain

$$\begin{aligned}
f^{(k)}(x) &= f\left(x^{(k)}\right) + \underbrace{\nabla f^{(k)}\left(x^{(k)}\right)^T}_{=\nabla f\left(x^{(k)}\right)^T}\left(x - x^{(k)}\right) + R_{f^{(k)}}(x) && (68)\\
\nabla f^{(k)}(x) &= \nabla f\left(x^{(k)}\right) + \nabla R_{f^{(k)}}(x) && (69)
\end{aligned}$$

and

$$\begin{aligned}
c_j^{(k)}(x) &= c_j\left(x^{(k)}\right) + \underbrace{\nabla c_j^{(k)}\left(x^{(k)}\right)^T}_{=\nabla c_j\left(x^{(k)}\right)^T}\left(x - x^{(k)}\right) + R_{c_j^{(k)}}(x) && (70)\\
\nabla c_j^{(k)}(x) &= \nabla c_j\left(x^{(k)}\right) + \nabla R_{c_j^{(k)}}(x) && (71)\\
e_j(x) &= e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + R_{e_j}(x) && (72)\\
\nabla e_j(x) &= \nabla e_j\left(x^{(k)}\right) + \nabla R_{e_j}(x) && (73)
\end{aligned}$$

for $j = 1, \ldots, m_c$ and $j = 1, \ldots, m_f$. Moreover, the gradients of $c_j(x),\ j = 1, \ldots, m_c$, and $e_j(x),\ j = 1, \ldots, m_f$, at $z^{(k)}$ are

$$\begin{aligned}
\nabla c_j\left(x^{(k)}\right)^T \Delta x^{(k)} &= c_j^{(k)}\left(z^{(k)}\right) - c_j\left(x^{(k)}\right) - R_{c_j^{(k)}}\left(z^{(k)}\right) && (74)\\
\nabla e_j\left(x^{(k)}\right)^T \Delta x^{(k)} &= e_j\left(z^{(k)}\right) - e_j\left(x^{(k)}\right) - R_{e_j}\left(z^{(k)}\right). && (75)
\end{aligned}$$

Since all approximating functions are convex, the corresponding residuals are nonnegative. The same holds for the residuals of the feasibility constraints, i.e.,

$$e_j\left(x^{(k)}\right) - e_j\left(z^{(k)}\right) \quad \geq \quad -\nabla e_j\left(z^{(k)}\right)^T \Delta x^{(k)} \tag{76}$$

We denote

$$R_{c^{(k)}}(x) \quad := \quad \left(R_{c_1^{(k)}}(x), \ldots, R_{c_{m_c}^{(k)}}(x)\right)^T \in \mathbb{R}^{m_c} \tag{77}$$

$$\nabla R_{c^{(k)}}(x) \quad := \quad \left(\nabla R_{c_1^{(k)}}(x), \ldots, \nabla R_{c_{m_c}^{(k)}}(x)\right) \in \mathbb{R}^{n \times m_c} \tag{78}$$

and

$$R_e(x) \quad := \quad \left(R_{e_1}(x), \ldots, R_{e_{m_f}}(x)\right)^T \in \mathbb{R}^{m_f} \tag{79}$$

$$\nabla R_e(x) \quad := \quad \left(\nabla R_{e_1}(x), \ldots, \nabla R_{e_{m_f}}(x)\right) \in \mathbb{R}^{n \times m_f} \tag{80}$$

respectively.

All iterates satisfy the bound conditions and we get an important relation between the Lagrangian multipliers of the box constraints $v_u^{(k)}$ and $v_l^{(k)}$ and the search direction $\Delta x^{(k)}$, see Lehmann [32],

$$\left(v_u^{(k)}\right)^T \Delta x^{(k)} \quad \geq \quad 0 \tag{81}$$

$$\left(v_l^{(k)}\right)^T \Delta x^{(k)} \quad \leq \quad 0 \tag{82}$$

Moreover, we obtain an upper bound on the descent of the objective function in iteration $x^{(k)}$,

$$\begin{aligned} \nabla f\left(x^{(k)}\right)^T \Delta x^{(k)} \quad \leq \quad & -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} - \left(v_c^{(k)}\right)^T A_c(x^{(k)})^T \Delta x^{(k)} \\ & - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} \\ & + \left(v_e^{(k)}\right)^T \left(e\left(x^{(k)}\right) - e\left(z^{(k)}\right)\right) \end{aligned} \tag{83}$$

and, in addition,

$$\left(v_c^{(k)}\right)^T R_{c^{(k)}}(z^{(k)}) - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} \quad \leq \quad 0 \tag{84}$$

From the feasibility of the moving asymptotes and the definition of $\eta_i^{(k)}$, $i = 1, \ldots, n$, we easily get a positive lower bound

$$\min_{i=1,\ldots,n} \eta_i^{(k)} =: \eta^{(k)} > \eta > 0 \tag{85}$$

where

$$\eta \;:=\; \tau \frac{(2-\omega)\,\xi}{(U_{\max}-L_{\min})^2}. \tag{86}$$

An important part of the convergence proof is to show that the penalty parameters and the augmented Lagrangian function are uniformly bounded on $X^{(k)}$. First, we note that the gradients of $f^{(k)}(x)$ and $c_j^{(k)}(x)$, $j=1,\ldots,m_c$, are uniformly bounded, i.e., if F is nonempty and compact, then there exists $M_0>0$ and $M_j>0$, $j=1,\ldots,m_c$ such that

$$\left|\frac{\partial f^{(k)}(x)}{\partial x_i}\right| \;<\; M_0, \quad i=1,\ldots,n \tag{87}$$

$$\left|\frac{\partial c_j^{(k)}(x)}{\partial x_i}\right| \;\leq\; M_j, \quad i=1,\ldots,n, \;\; j=1,\ldots,m_c \tag{88}$$

holds for all $x \in X^{(k)}$ and $k=0,1,2,\ldots$. The proof, see Lehmann [32] for details, depends heavily on the feasibility of the asymptotes and the way the bounds of (12) are constructed. F may be assumed to be bounded without loss of generality, since we omitted upper and lower bounds in (1) only to simplify the notation, i.e., we may assume the the feasible domain of (1) is always bounded.

On the other hand, the boundedness of the multipliers $y^{(k)}$) or $v^{(k)}$), respectively, is by no means trivial and cannot be achieved without additional assumptions on the underlying structure of the given nonlinear program (1). One possibility is to require the linear independency constraint qualification (LICQ) for all iterates and all possible accumulation points. Since, however, the LICQ cannot be checked in advance, we assume for simplicity that all multipliers are bounded.

To prove a global convergence theorem, we will construct a contradiction to the boundedness of the augmented Lagrangian function (17) from below, see Zillober [55]. His proof is to be extended by adding feasibility constraints. Thus, the subsequent Lemma is essential for proving convergence.

Lemma 3.1. *Let F defined by (2) be nonempty and compact. Then there exists a $M_\Phi \in \mathbb{R}$ such that*

$$\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} \geq M_\Phi \tag{89}$$

holds for all $x \in F$, $y \in Y$, Y compact, $(\rho_c)_j \geq 1$, $j=1,\ldots,m_c$ and $(\rho_e)_j \geq 1$, $j=1,\ldots,m_f$.

Proof. Considering the augmented Lagrangian merit function (56), we obtain

$$
\begin{aligned}
\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} &= f(x) + y_c^T \widetilde{c}(x) + \frac{1}{2} \rho_c^T \overline{c}^2(x) + y_e^T \widetilde{e}(x) + \frac{1}{2} \rho_e^T \overline{e}^2(x) \\
&= f(x) + \sum_{j=1}^{m_c} \begin{cases} (y_c)_j \, c_j(x) + \underbrace{\frac{(\rho_c)_j}{2} c_j^2(x)}_{\geq 0}, & \text{if } j \in J_c(x) \\ -\frac{(y_c)_j^2}{2 \, (\rho_c)_j}, & \text{if } j \in \overline{J}_c(x) \end{cases} \\
&\quad + \sum_{j=1}^{m_f} \begin{cases} (y_e)_j \, e_j(x) + \underbrace{\frac{(\rho_e)_j}{2} e_j^2(x)}_{\geq 0}, & \text{if } j \in J_e(x) \\ -\frac{(y_e)_j^2}{2 \, (\rho_e)_j}, & \text{if } j \in \overline{J}_e(x) \end{cases} \\
&\geq f(x) + \sum_{j=1}^{m_c} \begin{cases} (y_c)_j \, c_j(x), & \text{if } j \in J_c(x) \\ -\frac{(y_c)_j^2}{2 \, (\rho_c)_j}, & \text{if } j \in \overline{J}_c(x) \end{cases} \\
&\quad + \sum_{j=1}^{m_f} \begin{cases} (y_e)_j \, e_j(x), & \text{if } j \in J_e(x) \\ -\frac{(y_e)_j^2}{2 \, (\rho_e)_j}, & \text{if } j \in \overline{J}_e(x) \end{cases}
\end{aligned}
$$

As F is nonempty and compact, there exists $\min_{x \in F} c_j(x) \leq 0$, $j = 1, \ldots, m_c$, $\min_{x \in F} f(x)$, and $\min_{x \in F} e_j(x) \leq 0$, $j = 1, \ldots, m_f$, respectively. Moreover, there exists by assumption a $y_{\max} \in \mathbb{R}$ such that $|y_i| \leq y_{\max}$, $i = 1, \ldots, m_c + m_f$. Thus,

$$
\begin{aligned}
\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} &\geq \min_{x \in F} f(x) + \sum_{j=1}^{m_c} \begin{cases} y_{\max} \min_{x \in F} c_j(x), & \text{if } j \in J_c(x) \\ -y_{\max}^2, & \text{if } j \in \overline{J}_c(x) \end{cases} \\
&\qquad + \sum_{j=1}^{m_f} \begin{cases} y_{\max} \min_{x \in F} e_j(x), & \text{if } j \in J_e(x) \\ -y_{\max}^2, & \text{if } j \in \overline{J}_e(x) \end{cases} \\
&\geq \min_{x \in F} f(x) + m_c \, y_{\max} \min_{j=1,\ldots,m_c} \left\{ \min_{x \in F} c_j(x), -y_{\max} \right\} \\
&\qquad + m_f \, y_{\max} \min_{j=1,\ldots,m_f} \left\{ \min_{x \in F} e_j(x), -y_{\max} \right\} \\
&=: M_\Phi
\end{aligned}
$$

$\square$

Summarizing all previous investigations, we are able now to prove the sufficient descent property of the augmented Lagrangian merit function (17) subject to search direction

$$d^{(k)} := \begin{pmatrix} \Delta x^{(k)} \\ \Delta y_c^{(k)} \\ \Delta y_e^{(k)} \end{pmatrix} \tag{90}$$

with

$$\begin{aligned} \Delta x^{(k)} &:= z^{(k)} - x^{(k)} && (91) \\ \Delta y_c^{(k)} &:= v_c^{(k)} - y_c^{(k)} && (92) \\ \Delta y_e^{(k)} &:= v_e^{(k)} - y_e^{(k)} && (93) \end{aligned}$$

Note that we consider descent of $\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix}$ subject to the primal and dual variables, see also Schittkowski [43], Zillober [55, 57], and Lemann [32].

Theorem 3.1. *Let the sequences $\{x^{(k)}, y^{(k)}\}$ and $\{z^{(k)}, v^{(k)}\}$ be computed by Algorithm 2, where the corresponding approximations $f^{(k)}(x)$ and $c_j^{(k)}(x)$ are defined by (8) and (9). Let the asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, be feasible according to Definition 2.1, and let F defined by (2) be nonempty and compact. Consider $d^{(k)} \in \mathbb{R}^{n+m_c+m_f}$ defined by (90) and $\eta^{(k)}$ defined by (20). If $z^{(k)} \neq x^{(k)}$ and if the penalty parameters $\rho^{(k)}$ are computed by (1), then $d^{(k)}$ is a descent direction for the augmented Lagrangian function, i.e.,*

$$\nabla \Phi_{\rho^{(k)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \leq -\frac{\eta^{(k)} \| z^{(k)} - x^{(k)} \|_2^2}{2}. \tag{94}$$

Proof.

$$\begin{aligned} &\nabla \Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\ &\overset{(57)}{=} \nabla f\left(x^{(k)}\right)^T \Delta x^{(k)} + \left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T \underbrace{A_c\left(x^{(k)}\right)^T \Delta x^{(k)}}_{=c^{(k)}\left(z^{(k)}\right) - c\left(x^{(k)}\right) - R_{c^{(k)}}\left(z^{(k)}\right),\ (74)} \\ &\quad + \widehat{c}\left(x^{(k)}\right)^T \Delta y_c^{(k)} + \left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T \underbrace{A_e\left(x^{(k)}\right)^T \Delta x^{(k)}}_{=e\left(z^{(k)}\right) - e\left(x^{(k)}\right) - R_e\left(z^{(k)}\right),\ (75)} \\ &\quad + \widehat{e}\left(x^{(k)}\right)^T \Delta y_e^{(k)} \end{aligned}$$

$$
\begin{aligned}
&\nabla \Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
\overset{(83)}{\leq}\;& -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \\
& - \left(v_c^{(k)}\right)^T \underbrace{A_c(x^{(k)})^T \Delta x^{(k)}}_{=c^{(k)}\left(z^{(k)}\right) - c\left(x^{(k)}\right) - R_{c^{(k)}}\left(z^{(k)}\right),\ (74)} - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} \\
& + \left(v_e^{(k)}\right)^T \left(e\left(x^{(k)}\right) - e\left(z^{(k)}\right)\right) \\
& + \underbrace{\left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T}_{\geq 0,\ \text{Definition (33)}} \left(c^{(k)}\left(z^{(k)}\right) - c\left(x^{(k)}\right) - R_{c^{(k)}}\left(z^{(k)}\right)\right) \\
& + \widehat{c}\left(x^{(k)}\right)^T \Delta y_c^{(k)} \\
& + \underbrace{\left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T}_{\geq 0,\ \text{Definition (35)}} \left(e\left(z^{(k)}\right) - e\left(x^{(k)}\right) - R_e\left(z^{(k)}\right)\right) \\
& + \widehat{e}\left(x^{(k)}\right)^T \Delta y_e^{(k)} \\
=\;& -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \\
& - \left(v_c^{(k)}\right)^T \left(c^{(k)}\left(z^{(k)}\right) - c\left(x^{(k)}\right) - R_{c^{(k)}}\left(z^{(k)}\right)\right) \\
& - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} + \left(v_e^{(k)}\right)^T \left(e\left(x^{(k)}\right) - e\left(z^{(k)}\right)\right) \\
& + \underbrace{\left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T c^{(k)}\left(z^{(k)}\right)}_{\leq 0} - \left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T c\left(x^{(k)}\right) \\
& - \underbrace{\left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T R_{c^{(k)}}\left(z^{(k)}\right)}_{\geq 0} + \widehat{c}\left(x^{(k)}\right)^T \Delta y_c^{(k)} \\
& + \underbrace{\left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T e\left(z^{(k)}\right)}_{\leq 0} - \left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T e\left(x^{(k)}\right) \\
& - \underbrace{\left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T R_e\left(z^{(k)}\right)}_{\geq 0} + \widehat{e}\left(x^{(k)}\right)^T \Delta y_e^{(k)} \\
\leq\;& -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} - \underbrace{\left(v_c^{(k)}\right)^T c^{(k)}\left(z^{(k)}\right)}_{=0,\ (59)} + \left(v_c^{(k)}\right)^T c\left(x^{(k)}\right) \\
& + \underbrace{\left(v_c^{(k)}\right)^T R_{c^{(k)}}\left(z^{(k)}\right) - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)}}_{\leq 0,\ (84)} \\
& + \left(v_e^{(k)}\right)^T e\left(x^{(k)}\right) - \underbrace{\left(v_e^{(k)}\right)^T e\left(z^{(k)}\right)}_{=0,\ (60)} \\
& - \left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T c\left(x^{(k)}\right) + \widehat{c}\left(x^{(k)}\right)^T \Delta y_c^{(k)} \\
& - \left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T e\left(x^{(k)}\right) + \widehat{e}\left(x^{(k)}\right)^T \Delta y_e^{(k)}
\end{aligned}
$$

Using the Definitions (40)-(49) and $\Delta y_c^{(k)} := v_c^{(k)} - y_c^{(k)}$ we get

$$
\begin{aligned}
&\nabla \Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
\leq\ & -\nabla R_{f^{(k)}} \left(z^{(k)}\right)^T \Delta x^{(k)} + \sum_{j \in J_c^{(k)}} \left(v_c^{(k)}\right)_j c_j \left(x^{(k)}\right) + \sum_{j \in \overline{J}_c^{(k)}} \underbrace{\left(v_c^{(k)}\right)_j c_j \left(x^{(k)}\right)}_{\leq 0} \\
& + \sum_{j \in J_e^{(k)}} \left(v_e^{(k)}\right)_j e_j \left(x^{(k)}\right) + \underbrace{\sum_{j \in \overline{J}_e^{(k)}} \left(v_e^{(k)}\right)_j e_j \left(x^{(k)}\right)}_{\leq 0} - \sum_{j \in J_c^{(k)}} \left(y_c^{(k)}\right)_j c_j \left(x^{(k)}\right) \\
& - \sum_{j \in J_c^{(k)}} (\rho_c)_j c_j^2 \left(x^{(k)}\right) + \sum_{j \in J_c^{(k)}} c_j \left(x^{(k)}\right) \left(\Delta y_c^{(k)}\right)_j - \sum_{j \in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j} \left(\Delta y_c^{(k)}\right)_j \\
& - \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j \left(x^{(k)}\right) - \sum_{j \in J_e^{(k)}} (\rho_e)_j e_j^2 \left(x^{(k)}\right) \\
& + \sum_{j \in J_e^{(k)}} e_j \left(x^{(k)}\right) \left(\Delta y_e^{(k)}\right)_j - \sum_{j \in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{(\rho_e)_j} \left(\Delta y_e^{(k)}\right)_j \\
\leq\ & -\nabla R_{f^{(k)}} \left(z^{(k)}\right)^T \Delta x^{(k)} + \underbrace{\sum_{j \in J_c^{(k)}} \left(v_c^{(k)}\right)_j c_j \left(x^{(k)}\right) - \sum_{j \in J_c^{(k)}} \left(y_c^{(k)}\right)_j c_j \left(x^{(k)}\right)}_{= \sum_{j \in J_c^{(k)}} c_j \left(x^{(k)}\right) \left(\Delta y_c^{(k)}\right)_j} \\
& - \sum_{j \in J_c^{(k)}} (\rho_c)_j c_j^2 \left(x^{(k)}\right) + \sum_{j \in J_c^{(k)}} c_j \left(x^{(k)}\right) \left(\Delta y_c^{(k)}\right)_j \qquad (95) \\
& - \sum_{j \in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j} \left(\Delta y_c^{(k)}\right)_j + \underbrace{\sum_{j \in J_e^{(k)}} \left(v_e^{(k)}\right)_j e_j \left(x^{(k)}\right)}_{\leq 0} - \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j \left(x^{(k)}\right) \\
& - \sum_{j \in J_e^{(k)}} \underbrace{(\rho_e)_j e_j^2 \left(x^{(k)}\right)}_{\geq 0} + \sum_{j \in J_e^{(k)}} e_j \left(x^{(k)}\right) \left(\Delta y_e^{(k)}\right)_j - \sum_{j \in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{(\rho_e)_j} \left(\Delta y_e^{(k)}\right)_j
\end{aligned}
$$

All together this leads to

$$\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \leq \underbrace{-\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)}}_{\text{(a)}} \\ \underbrace{+\sum_{j\in J_c^{(k)}} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \sum_{j\in J_c^{(k)}} \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)}_{\text{(b)}} \\ \underbrace{-\sum_{j\in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{\left(\rho_c\right)_j}\left(\Delta y_c^{(k)}\right)_j}_{\text{(c)}} \underbrace{-\sum_{j\in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)}_{\text{(d)}} \\ \underbrace{+\sum_{j\in J_e^{(k)}} e_j\left(x^{(k)}\right)\left(\Delta y_e^{(k)}\right)_j}_{\text{(e)}} \underbrace{-\sum_{j\in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j}\left(\Delta y_e^{(k)}\right)_j}_{\text{(f)}} \tag{96}$$

We now have to show that (96) is less than $-\frac{\eta^{(k)}\left\|z^{(k)}-x^{(k)}\right\|^2}{2}$. Therefore, we consider each part individually.

Considering (a):
Using the derivative of the residual term of $f^{(k)}$, we get:

$$\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \\ = \sum_{I_+^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau\right]\left[\frac{\left(z_i^{(k)} - x_i^{(k)}\right)^2\left(2U_i^{(k)} - z_i^{(k)} - x_i^{(k)}\right)}{\left(U_i^{(k)} - z_i^{(k)}\right)^2}\right] \tag{97}$$
$$-\sum_{I_-^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau\right]\left[\frac{\left(z_i^{(k)} - x_i^{(k)}\right)^2\left(-2L_i^{(k)} + z_i^{(k)} + x_i^{(k)}\right)}{\left(z_i^{(k)} - L_i^{(k)}\right)^2}\right] \\ \overset{(20)}{=} \sum_{I_+^{(k)}} \eta_i^{(k)}\left(z_i^{(k)} - x_i^{(k)}\right)^2 + \sum_{I_-^{(k)}} \eta_i^{(k)}\left(z_i^{(k)} - x_i^{(k)}\right)^2$$

Together with Definition $\eta^{(k)} := \min_{i=1,\ldots,n} \eta_i^{(k)}$ given in (21) we get

$$-\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \leq -\eta^{(k)}\left\|\Delta x^{(k)}\right\|_2^2 = -\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2 < 0 \tag{98}$$

Considering (b):
As $0 \leq \left(v_c^{(k)}\right)_j \leq y_{\max}$ and $0 \leq \left(y_c^{(k)}\right)_j \leq y_{\max}$,

$$\left|\Delta \left(y_c^{(k)}\right)_j\right| = \left|\left(v_c^{(k)}\right)_j - \left(y_c^{(k)}\right)_j\right| \leq y_{\max} \tag{99}$$

holds. This leads to

$$\begin{aligned}
&\sum_{j\in J_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right] \\
&\quad\leq \sum_{j\in J_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| \underbrace{\left|\left(\Delta y_c^{(k)}\right)_j\right|}_{\leq y_{\max},\ (99)} - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right] \\
&\quad\leq \sum_{j\in J_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right] \qquad (100)
\end{aligned}$$

In the case of $c_j\left(x^{(k)}\right) = 0$, the corresponding term of (b) is equal to zero. We define

$$Z_c^{(k)} \quad := \quad \left\{j \in J_c^{(k)} \;\middle|\; c_j\left(x^{(k)}\right) = 0\right\}. \tag{101}$$

To ensure property (94), we assume that the penalty parameters $\left(\rho_c\right)_j,\ j \in J_c^{(k)} \backslash Z_c^{(k)}$ are larger than $\left(\rho_1^{(k)}\right)_j,\ j \in J_c^{(k)}$ given by

$$\left(\rho_1^{(k)}\right)_j := \frac{2y_{\max}}{\left|c_j\left(x^{(k)}\right)\right|}, \text{ i.e., } \left(\rho_1^{(k)}\right)_j \leq \left(\rho_c\right)_j,\ \forall j \in J_c^{(k)} \backslash Z_c^{(k)}. \tag{102}$$

With (102) in (100) we get

$$\begin{aligned}
&\sum_{j\in J_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right] \\
&\quad= \sum_{j\in J_c^{(k)}\backslash Z_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right] \\
&\qquad + \underbrace{\sum_{j\in Z_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right]}_{=0} \\
&\quad\leq \sum_{j\in J_c^{(k)}\backslash Z_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&\sum_{j\in J_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - (\rho_c)_j\, c_j^2\left(x^{(k)}\right)\right] \\
&\leq \sum_{j\in J_c^{(k)}\setminus Z_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \left(\rho_1^{(k)}\right)_j c_j^2\left(x^{(k)}\right)\right] \\
&= \sum_{j\in J_c^{(k)}\setminus Z_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \frac{2y_{\max}}{\left|c_j\left(x^{(k)}\right)\right|} c_j^2\left(x^{(k)}\right)\right] \\
&= \sum_{j\in J_c^{(k)}\setminus Z_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - 2y_{\max}\left|c_j\left(x^{(k)}\right)\right|\right] \\
&= 0
\end{aligned}
$$

In total we get

$$
\sum_{j\in J_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - (\rho_c)_j\, c_j^2\left(x^{(k)}\right)\right] \leq 0 \tag{103}
$$

for each $(\rho_c)_j \geq \rho_1^{(k)} := \max\limits_{j\in J_c^{(k)}\setminus Z_c^{(k)}} \left\{\frac{2y_{\max}}{\left|c_j\left(x^{(k)}\right)\right|}\right\}$.

Considering (c):
The term contains the inactive inequality constraints with respect to the augmented Lagrangian function. We get

$$
\begin{aligned}
\left|\sum_{j\in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\left(\Delta y_c^{(k)}\right)_j\right| &\leq \sum_{j\in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j} \underbrace{\left|\left(v_c^{(k)}\right)_j - \left(y_c^{(k)}\right)_j\right|}_{\leq y_{\max},\ (99)} \\
&\leq y_{\max} \sum_{j\in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j} \\
&\leq \left(y_{\max}\right)^2 \sum_{j\in \overline{J}_c^{(k)}} \frac{1}{(\rho_c)_j}
\end{aligned}
$$

This can be summarized by

$$
\left|\sum_{j\in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\left(\Delta y_c^{(k)}\right)_j\right| \leq \left(y_{\max}\right)^2 \sum_{j\in \overline{J}_c^{(k)}} \frac{1}{(\rho_c)_j}. \tag{104}
$$

To ensure property (94), we assume that the penalty parameters $(\rho_c)_j\,,\ j \in \overline{J}_c^{(k)}$ are larger than $\rho_2^{(k)}$ given by:

$$\rho_2^{(k)} := m_c \frac{10\left(y_{\max}\right)^2}{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2},\ \text{i.e., } \rho_2^{(k)} \leq (\rho_c)_j\,,\ j \in \overline{J}_c^{(k)}. \tag{105}$$

Using (104) and the definition of $\rho_2^{(k)}$ in (105) we get

$$\begin{aligned}
\left|\sum_{j\in\overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\left(\Delta y_c^{(k)}\right)_j\right| &\leq \left(y_{\max}\right)^2 \sum_{j\in\overline{J}_c^{(k)}} \frac{1}{(\rho_c)_j} \\
&\leq \left(y_{\max}\right)^2 \sum_{j\in\overline{J}_c^{(k)}} \frac{1}{\rho_2^{(k)}} \\
&\leq \frac{m_c\left(y_{\max}\right)^2}{\rho_2^{(k)}} \\
&= \frac{m_c\left(y_{\max}\right)^2 \eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2}{10 m_c\left(y_{\max}\right)^2} \\
&= \frac{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2}{10}
\end{aligned}$$

As result we get

$$\left|\sum_{j\in\overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\left(\Delta y_c^{(k)}\right)_j\right| \leq \frac{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2}{10}, \tag{106}$$

for each $(\rho_c)_j \geq \rho_2^{(k)},\ j \in J_e^{(k)}$.

Considering the feasibility constraints we can exploit the fact, that each iterate is feasible, i.e., the following inequalities hold

$$e_j\left(x^{(k)}\right) \leq 0, \qquad j = 1, \ldots, m_f \tag{107}$$

$$e_j\left(z^{(k)}\right) \leq 0, \qquad j = 1, \ldots, m_f \tag{108}$$

Considering (d):

Due to (35) we get for each $j \in J_e^{(k)}$

$$\begin{aligned} 0 \;\geq\; e_j\left(x^{(k)}\right) &\geq \frac{-\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j} \\ \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j} &\geq -e_j\left(x^{(k)}\right) = \left|e_j\left(x^{(k)}\right)\right| \end{aligned}$$

This leads to

$$\begin{aligned} \left|\sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)\right| &\leq y_{\max} \sum_{j \in J_e^{(k)}} \left|e_j\left(x^{(k)}\right)\right| \\ &\leq y_{\max} \sum_{j \in J_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j} \\ &\leq \left(y_{\max}\right)^2 \sum_{j \in J_e^{(k)}} \frac{1}{\left(\rho_e\right)_j} \end{aligned}$$

To ensure property (94), we assume that the penalty parameters $\left(\rho_e\right)_j,\ j \in J_e^{(k)}$ are larger than $\rho_3^{(k)}$ given by:

$$\rho_3^{(k)} \;:=\; m_f \frac{10\left(y_{\max}\right)^2}{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2},\ \text{i.e., } \rho_3^{(k)} \leq \left(\rho_e\right)_j,\ j \in J_e^{(k)} \tag{109}$$

which leads to

$$\begin{aligned} \left|\sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)\right| &\leq \left(y_{\max}\right)^2 \sum_{j \in J_e^{(k)}} \frac{1}{\left(\rho_e\right)_j} \\ &\leq \left(y_{\max}\right)^2 \sum_{j \in J_e^{(k)}} \frac{1}{\rho_3^{(k)}} \\ &\leq \left(y_{\max}\right)^2 m_f \frac{1}{\rho_3^{(k)}} \\ &= \left(y_{\max}\right)^2 m_f \frac{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2}{10 m_f \left(y_{\max}\right)^2} \\ &= \frac{\eta^{(k)}\left\|z^{(k)} - x^{(k)}\right\|^2}{10} \end{aligned}$$

As result we get

$$\left| \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right) \right| \leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} \tag{110}$$

for each $(\rho_e)_j \geq \rho_3^{(k)},\ j \in J_e^{(k)}$.

Considering (e):
Analogue to (d) we can show that

$$\begin{aligned}
\left| \sum_{j \in J_e^{(k)}} e_j\left(x^{(k)}\right) \left(\Delta y_e^{(k)}\right)_j \right| &\leq \sum_{j \in J_e^{(k)}} \left| e_j\left(x^{(k)}\right) \right| \left| \left(\Delta y_e^{(k)}\right)_j \right| \\
&\leq y_{\max} \sum_{j \in J_e^{(k)}} \left| e_j\left(x^{(k)}\right) \right| \\
&\leq y_{\max} \sum_{j \in J_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{(\rho_e)_j} \\
&\leq (y_{\max})^2 \sum_{j \in J_e^{(k)}} \frac{1}{(\rho_e)_j}
\end{aligned}$$

To ensure property (94), we assume that the penalty parameters $(\rho_e)_j,\ j \in J_e^{(k)}$ are larger than $\rho_3^{(k)}$ given by (109) which leads to

$$\begin{aligned}
\left| \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right) \right| &\leq (y_{\max})^2 \sum_{j \in J_e^{(k)}} \frac{1}{(\rho_e)_j} \\
&\leq (y_{\max})^2 \sum_{j \in J_e^{(k)}} \frac{1}{\rho_3^{(k)}} \\
&\leq (y_{\max})^2 m_f \frac{1}{\rho_3^{(k)}} \\
&= (y_{\max})^2 m_f \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10 m_f (y_{\max})^2} \\
&= \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10}
\end{aligned}$$

As result we get

$$\left| \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j \left(x^{(k)}\right) \right| \leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} \tag{111}$$

for each $\left(\rho_e\right)_j \geq \rho_3^{(k)},\ j \in \overline{J}_c^{(k)}$.

Considering (f):
It can be shown analogously to (c) that

$$\left| \sum_{j \in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j} \left(\Delta y_e^{(k)}\right)_j \right| \leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} \tag{112}$$

for

$$\left(\rho_e\right)_j \geq \rho_3^{(k)} := m_f \frac{10 \left(y_{\max}\right)^2}{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2},\ j \in \overline{J}_e^{(k)}. \tag{113}$$

Proceeding from (96) we can summarize previous calculations as follows:

$$
\begin{aligned}
&\nabla \Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
\leq\ & \underbrace{-\nabla R_{f^{(k)}} \left(z^{(k)}\right)^T \Delta x^{(k)}}_{\leq -\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2,\ (98)} \\
& \underbrace{+ \sum_{j \in J_c^{(k)}} 2 c_j \left(x^{(k)}\right) \left(\Delta y_c^{(k)}\right)_j - \sum_{j \in J_c^{(k)}} (\rho_c)_j\, c_j^2 \left(x^{(k)}\right)}_{\leq 0,\ (103)} \\
& \underbrace{- \sum_{j \in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j} \left(\Delta y_c^{(k)}\right)_j}_{\leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10},\ (106)} \underbrace{- \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j \left(x^{(k)}\right)}_{\leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10},\ (110)} \\
& \underbrace{+ \sum_{j \in J_e^{(k)}} e_j \left(x^{(k)}\right) \left(\Delta y_e^{(k)}\right)_j}_{\leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10},\ (111)} \underbrace{- \sum_{j \in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{(\rho_e)_j} \left(\Delta y_e^{(k)}\right)_j}_{\leq \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10},\ (112)} \\
\leq\ & -\eta^{(k)} \left(\delta^{(k)}\right)^2 + \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} + \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} \\
& + \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} + \frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{10} \\
<\ & -\frac{\eta^{(k)} \left\| z^{(k)} - x^{(k)} \right\|^2}{2} \\
<\ & 0
\end{aligned}
$$

for $(\rho_c)_j \geq \overline{\rho}_c^{(k)}$ with $\overline{\rho}_c^{(k)} := \max\{\rho_1^{(k)}, \rho_2^{(k)}\}, \forall j = 1, \ldots, m_c$, and $(\rho_e)_j \geq \overline{\rho}_e^{(k)}$ with $\overline{\rho}_e^{(k)} := \rho_3^{(k)}, \forall j = 1, \ldots, m_f$. $\square$

To prove a convergence property, it is essential to show that there exist a subsequence of $\{x^{(k)}\}$ with $\left\|\Delta x^{(k)}\right\| \leq \varepsilon$, see Schittkowski [43].

Theorem 3.2. *Let the assumptions of Theorem 3.1 hold. Let* $\left\{x^{(k)}, y^{(k)}\right\}$ *be computed by Algorithm 2. Then there exists for each* $\varepsilon > 0$ *at least one* k *such that*

$$\left\|\Delta x^{(k)}\right\| \leq \varepsilon. \tag{114}$$

Proof. We prove by contradiction, that (114) holds for at least one k. We assume that $\left\|\Delta x^{(k)}\right\| > \varepsilon$ for a fixed ε and each k. We consider the sequence $\left\{x^{(k)}, y^{(k)}\right\}$ starting in iteration $\overline{k}$ and define the corresponding constant vector of penalty parameters by $\overline{\rho}_r$. Moreover,

$$\nabla\Phi_{\overline{\rho}_r}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} < -\frac{\eta\varepsilon^2}{2} < 0 \tag{115}$$

holds with $d^{(k)} := \begin{pmatrix} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{pmatrix} \neq \mathbb{0}$, see Theorem 3.1. There exists a i_0 independent from k such that the Armijo condition (25) is satisfied for all $i \geq i_0$, see Schittkowski [43]. As a consequence $\sigma^{(k)} \geq \underline{\sigma} := \beta^{i_0}$ and we get

$$\begin{aligned} \Phi_{\overline{\rho}_r}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} - \Phi_{\overline{\rho}_r}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} - \sigma^{(k)} d^{(k)}\right) &\geq -\mu \underbrace{\sigma^{(k)}}_{\geq\underline{\sigma}} \underbrace{\nabla\Phi_{\overline{\rho}_r}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}}_{\leq -\frac{\eta\varepsilon^2}{2}} \\ &\geq \mu\underline{\sigma}\frac{\eta\varepsilon^2}{2} \end{aligned}$$

for all $k \geq \overline{k}$. This leads to

$$\lim_{i\to\infty} \Phi_{\overline{\rho}_r}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} = -\infty, \tag{116}$$

which is a contradiction to Lemma 3.1. Therefore, the assumption is wrong and (114) holds for at least one k.

□

With these results it is possible to formulate and prove the main convergence theorem, see Zillober [55] Theorem 5.12, and Schittkowski [43] respectively.

Theorem 3.3. *Let the assumptions of Theorem 3.1 hold. Let* $\left\{x^{(k)}\right\}$, $\left\{y^{(k)}\right\}$, $\left\{z^{(k)}\right\}$, $\left\{v^{(k)}\right\}$*, be computed by Algorithm 2. Then there exists an accumulation point* $(x^\star, v^\star)$ *of* $\left\{(x^{(k)}, v^{(k)})\right\}$ *satisfying the KKT conditions for problem (1).*

Proof. The primal and dual iterates $\left\{x^{(k)}, v^{(k)}\right\}$ are elements of the bounded set $(F \times Y)$ with $Y := \{y \in \mathbb{R}^{m_c+m_f} \mid y \in [\,\mathbb{0}, y_{\max}\mathbb{1}]\}$, where $\mathbb{1}$ is a vector of ones of appropriate size and F is defined by (2). The results of Theorem 3.2 and the boundedness of $\left\{x^{(k)}\right\}$ and $\left\{v^{(k)}\right\}$, guarantee the existence of at least one accumulation point $(x^\star, v^\star)$ and of an infinite subset $S \subseteq \mathbb{N}$ such that

$$\lim_{k\in S} \Delta x^{(k)} = 0, \tag{117}$$

$$\lim_{k\in S} x^{(k)} = x^\star, \tag{118}$$

$$\lim_{k\in S} v^{(k)} = v^\star. \tag{119}$$

The statement follows then directly from the KKT conditions of subproblem (12) and the transition to the limit $\Delta x^{(k(} \to 0$, see Lehmann [32] for details. $\square$

4 Free Material Optimization

The goal of free material optimization (FMO), see Bendsøe et. al. [7] and Zowe, Kočvara and Bendsøe [62], is to find the best mechanical structure in the sense of minimal weight or maximal stiffness with respect to a set of given loads based on a finite element discretization. Moreover, additional constraints have to be satisfied. The material itself, as well as its distribution in the available space is optimized. As shown, e.g., by Kočvara and Stingl [30], the FMO problem can be formulated as a nonlinear semidefinite programming (NSDP) problem. Other problem formulations are given by Kočvara, Beck, Ben-Tal and Stingl [27].

FMO was first introduced by Bendsøe et al. [7], Bendsøe and Díaz [6], Bendsøe [5] and Zowe, Kočvara and Bendsøe [62]. The continuous problem formulation leads to a saddle-point problem for which the existence of a solution can be shown, see Mach [34] and Werner [53]. Based on a finite element discretization, in each finite element it is determined which material is used. The goal is to find the distribution of material such that the resulting structure becomes as stiff as possible, i.e., the compliance becomes as small as possible.

We define the space of symmetric matrices of size p by $\mathbb{S}^p$. Moreover, symmetric positive semidefinite matrices of size p are defined by $\mathbb{S}^p_+$ and symmetric positive definite matrices by $\mathbb{S}^p_{++}$.

We proceed from a bounded domain Ω in the two or three dimensional space with a Lipschitz boundary and a corresponding underlying finite element (FE) discretization with m elements and q nodes of the design space. For a detailed description of the Lipschitz boundary, we refer the reader to, e.g., Werner [53] and for the FE theory to, e.g., Mach [34].

The design variable E is a block diagonal matrix consisting of symmetric matrices $E_i,\ i = 1, \ldots, m$, that represent material properties in each finite element. The matrices $E_i,\ i = 1, \ldots, m$, have to be symmetric and positive semidefinite, to satisfy the basic requirements of linear elasticity, see Bendsøe et. al. [7]. Moreover, the variables might become zero in some regions. This situation is known as vanishing material and interpreted as void.

The design variable E is a matrix, dependent on the dimension of the given design space and the number of finite elements m.

$$E := \begin{pmatrix} E_1 & & & 0 \\ & E_2 & & \\ & & \ddots & \\ 0 & & & E_m \end{pmatrix} \tag{120}$$

For the two dimensional space the matrices $E_i,\ i = 1, \ldots, m$, are 3×3 matrices

$$E_i := \begin{pmatrix} e_{i1} & e_{i2} & e_{i4} \\ e_{i2} & e_{i3} & e_{i5} \\ e_{i4} & e_{i5} & e_{i6} \end{pmatrix} \succeq 0, \quad i = 1, \ldots, m, \tag{121}$$

see, e.g., Werner [53] for a detailed description of the derivation. This yields 6 variables for each matrix $E_i,\ i = 1, \ldots, m$, since E_i is symmetric. In the three dimensional space we get 6×6 matrices, i.e., 21 variables for each matrix

$$E_i := \begin{pmatrix} e_{i1} & e_{i2} & e_{i4} & e_{i7} & e_{i11} & e_{i16} \\ e_{i2} & e_{i3} & e_{i5} & e_{i8} & e_{i12} & e_{i17} \\ e_{i4} & e_{i5} & e_{i6} & e_{i9} & e_{i13} & e_{i18} \\ e_{i7} & e_{i8} & e_{i9} & e_{i10} & e_{i14} & e_{i19} \\ e_{i11} & e_{i12} & e_{i13} & e_{i14} & e_{i15} & e_{i20} \\ e_{i16} & e_{i17} & e_{i18} & e_{i19} & e_{i20} & e_{i21} \end{pmatrix} \succeq 0, \quad i = 1, \ldots, m. \tag{122}$$

Therefore,

$$E \in \mathbb{S}_+^{3m}, \tag{123}$$
$$E_i \in \mathbb{S}_+^{3}, \quad i = 1, \ldots, m \tag{124}$$

holds for the two dimensional case and

$$E \in \mathbb{S}_+^{6m}, \tag{125}$$
$$E_i \in \mathbb{S}_+^{6}, \quad i = 1, \ldots, m \tag{126}$$

for the three dimensional case, respectively, where m is the number of finite elements. In the sequel, we focus on the two dimensional case.

The so-called compliance function is a measure of the stiffness of the resulting structure. The smaller the value of the compliance the more robust is the structure with respect to loads $f_j \in \mathbb{R}^{2q}$, $j = 1, \ldots, l$, where l denotes the number of load cases and q the number of nodes. The stiffness of the structure is dependent on the material properties of each element $E_i,\ i = 1, \ldots, m$, and is given by the global stiffness matrix $K(E) \in \mathbb{R}^{2q \times 2q}$, see Ciarlet [14],

$$K(E) := \sum_{i=1}^{m} K_i(E) \tag{127}$$

$$K_i(E) := \sum_{k=1}^{n_g} B_{i,k}^T E_i B_{i,k} \tag{128}$$

where $K_i(E) \in \mathbb{R}^{2q \times 2q}$, $B_{i,k} \in \mathbb{R}^{3 \times 2q}$ and $n_g \in \mathbb{R}$ defines the number of Gauss integration points. A detailed description how to compute the matrices $B_{i,k}$ is given in Hörnlein, Kočvara and Werner [24], Kočvara and Zowe [31], Zowe, Kočvara and Bendsøe [62]. The behavior of the structure with respect to loads $f_j,\ j = 1, \ldots, l$, is given by

$$f_j^T u_j(E), \quad j = 1, \ldots, l, \tag{129}$$

where $u_j(E) \in \mathbb{R}^{2q},\ j = 1, \ldots, l$, is the displacement vector.

The displacement vector is determined by the equilibrium condition

$$K(E)\, u_j(E) = f_j, \quad j = 1, \ldots, l \tag{130}$$

derived from linear Hooke's law, which describes the equilibrium of internal forces and the acting loads. This means

$$K(E)\, u_j(E) = f_j \iff u_j(E) = K(E)^{-1} f_j, \qquad j = 1, \ldots, l, \tag{131}$$

$$\implies f_j^T u_j(E) = f_j^T K(E)^{-1} f_j, \quad j = 1, \ldots, l, \tag{132}$$

where $K^{-1}(E)\, f_j$ can be computed by solving the linear system (130) to save computational effort.

To ensure that the linear system is solvable we request that $K(E)$ is positive definite, i.e., $K(E) \in \mathbb{S}^{2r}_{++}$. This leads to the requirement that each matrix $E_i,\ i = 1, \ldots, m$ is positive definite, since

$$K(E) \in \mathbb{S}^{2r}_{++} \iff E \in \mathbb{S}^{3m}_{++} \tag{133}$$

$$\iff E_i \in \mathbb{S}^{3}_{++}, \quad i = 1, \ldots, m \tag{134}$$

Therefore, we require $E_i - \underline{\nu} I \succeq 0,\ i = 1, \ldots, m$, where I is the identity matrix and $\underline{\nu} \in \mathbb{R}^+$ is a small positive value, see Kočvara and Stingl [29]. We have to ensure, that the semidefinite constraints $E_i - \underline{\nu} I \succeq 0$, $i = 1, \ldots, m$, are satisfied, whenever the linear system (130) is to be solved.

As FMO treats multiple load cases, i.e., different set of loads are acting independently, we introduce an additional variable $\alpha \in \mathbb{R}$, which is to be minimized, requiring for each load case

$$f_j^T K^{-1}(E)\, f_j \le \alpha, \quad j = 1, \ldots, l, \tag{135}$$

see Ben-Tal, Kočvara, Nemirovski and Zowe [4].

The sum of the diagonal elements of the matrices $E_i,\ i = 1, \ldots, m$, is a measure for stiffness of the material in coordinate directions. The trace of $E_i,\ i = 1, \ldots, m$, can be used as a cost function, see Bendsøe et al. [7], to represent the limited amount of material. We introduce the upper bound $V \in \mathbb{R}$ and require

$$\sum_{i=1}^{m} \mathrm{Trace}\,(E_i) \quad \leq \quad V, \tag{136}$$

$$\mathrm{Trace}\,(E_i) \quad := \quad e_{i1} + e_{i3} + e_{i6}, \quad i = 1, \ldots, m \tag{137}$$

The trace of each element is bounded by $\overline{\nu} \in \mathbb{R}^+$, since it is not possible to produce arbitrarily stiff material. This leads to

$$\mathrm{Trace}\,(E_i) \quad \leq \quad \overline{\nu}, \quad i = 1, \ldots, m, \tag{138}$$

see, e.g., Ben-Tal, Kočvara, Nemirovski and Zowe [4]. Moreover, from $E_i - \underline{\nu} I \succeq 0$, $i = 1, \ldots, m$, we derive additional lower bounds on the trace, i.e.,

$$3\underline{\nu} \leq \mathrm{Trace}\,(E_i)\,, \quad i = 1, \ldots, m. \tag{139}$$

They can be expressed as box constraints for the diagonal variables, i.e., e_{i1}, e_{i3} and e_{i6},

$$e_{i1} \quad \geq \quad \underline{\nu}, \tag{140}$$

$$e_{i3} \quad \geq \quad \underline{\nu}, \tag{141}$$

$$e_{i6} \quad \geq \quad \underline{\nu}. \tag{142}$$

In the three dimensional case, the variables $e_{i1}, e_{i3}, e_{i6}, e_{i10}, e_{i15}$ and e_{i21} are restricted. In general, there are two possibilities to formulate the free material optimization problem. One possibility is to minimize the volume function $\sum_{i=1}^{m} \mathrm{Trace}\,(E_i)$ with respect to a given stability of the resulting structure. Another approach maximizes the stiffness with respect to a limited volume. We focus on the second approach, which results in the nonlinear convex semidefinite problem

$$\begin{array}{lll}
\min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\
\text{s.t.} & \sum\limits_{i=1}^{m} \mathrm{Trace}\,(E_i) - V \leq 0 & \\
& \mathrm{Trace}\,(E_i) - \overline{\nu} \leq 0, & i = 1, \ldots, m \\
& E_i - \underline{\nu} I \succeq 0, & i = 1, \ldots, m \\
& f_j^T K\,(E)^{-1} f_j - \alpha \leq 0, & j = 1, \ldots, l \\
& e_{i1} \geq \underline{\nu}, & i = 1, \ldots, m \\
& e_{i3} \geq \underline{\nu}, & i = 1, \ldots, m \\
& e_{i6} \geq \underline{\nu}, & i = 1, \ldots, m
\end{array} \tag{143}$$

The optimization variables are the entries $e_{ip},\ p = 1,\dots,6$ of the elementary stiffness matrices $E_i,\ i = 1,\dots,m$. The derivatives are specified in Ertel, Schittkowski and Zillober [16] for all $i = 1,\dots,m$, $p = 1,\dots,6$ by

$$\frac{\partial}{\partial e_{ip}}\left(\sum_{i=1}^{m}\mathrm{Trace}\left(E_i\right) - V\right) = \begin{cases} 1, & \text{if } p = 1,3,6 \\ 0, & \text{otherwise} \end{cases} \tag{144}$$

$$\frac{\partial}{\partial e_{ip}}\left(\mathrm{Trace}\left(E_i\right) - \overline{\nu}\right) = \begin{cases} 1, & \text{if } p = 1,3,6 \\ 0, & \text{otherwise} \end{cases} \tag{145}$$

$$\frac{\partial}{\partial e_{ip}}\left(f_j^T K\left(E\right)^{-1} f_j - \alpha\right) = -u_j\left(E\right)^T\left(\frac{\partial K\left(E\right)}{\partial e_{ip}}\right)u_j\left(E\right) \tag{146}$$

$$\begin{aligned}\frac{\partial}{\partial e_{ip}}K\left(E\right) &= \frac{\partial}{\partial e_{ip}}\left(\sum_{i=1}^{m}\sum_{k=1}^{n_g}B_{i,k}^T E_i B_{i,k}\right) \\ &= \sum_{i=1}^{m}\sum_{k=1}^{n_g}B_{i,k}^T\frac{\partial E_i}{\partial e_{ip}}B_{i,k}\end{aligned} \tag{147}$$

with

$$\frac{\partial E_i}{\partial e_{ip}} = \left(\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix},\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix},\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix},\begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix},\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\end{bmatrix},\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}\right)^T \tag{148}$$

Moreover, additional constraints can be added to optimization problem (143). Especially stress constraints, such as 'von Mises stress conditions', see, e.g., Li, Steven and Xie [33], are very important from the engineering point of view. In the two dimensional space the von Mises stress in an element $i \in \{1,\dots,m\}$, and load case $j \in \{1,\dots,l\}$, can be formulated as

$$s_{i,j}\left(E\right) := \sum_{k=1}^{n_g}u_j\left(E\right)^T B_{i,k}^T E_i I E_i B_{i,k} u_j\left(E\right) \tag{149}$$

with

$$I = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \tag{150}$$

see Kočvara and Stingl [30]. The integration of stress constraints lead to numerical problems for the optimization method, as regularity conditions such as the linear independency constraint qualification, are not satisfied, see Achtziger and Kanzow [1] and Stingl [46].

To ensure stability $s_{i,j}(E)$ may not exceed a given threshold $s_\sigma \in \mathbb{R}^+$. For each load case $j \in \{1,\dots,l\}$, and each element $i \in \{1,\dots,m\}$, we get one additional constraint that can be added to (143):

$$s_{i,j}(E) \;\; \leq \;\; s_\sigma, \quad i=1,\dots,m,\; j=1,\dots,l. \tag{151}$$

The corresponding derivatives are given by

$$\begin{aligned} & \frac{\partial}{\partial e_{ip}} \left(\sum_{k=1}^{n_g} u_j(E)^T B_{i,k}^T E_i I E_i B_{i,k} u_j(E) - s_\sigma \right) \\ = \;\; & \sum_{k=1}^{n_g} -2u_j(E)^T K^{-1}(E) \frac{\partial K(E)}{\partial e_{ip}} B_{i,k}^T E_i I E_i B_{i,k} u_j(E) \\ & + \sum_{k=1}^{n_g} 2u_j(E)^T B_{i,k}^T \frac{\partial E_i}{\partial e_{ip}} I E_i B_{i,k} u_j(E), \end{aligned} \tag{152}$$

see Ertel, Schittkowski, and Zillober [16].

Our goal is to solve (143) by Algorithm 2. Since the feasible SCP method is not designed to handle semidefinite constraints, they are reformulated by nonlinear constraints. Benson and Vanderbei [9] compute the eigenvalues of $E_i - \nu I$ successively based on a recursion formula, which could be used to define the feasibility functions The disadvantage, however, is that one has to invert matrices, which are positive definite only within the feasible region. At the boundary of F, we would get singularities.

Thus, we modify their approach. The idea is to consider the submatrices of matrix E'. A matrix is positive semidefinite, if each submatrix is positive semidefinite, i.e.,

$$E' \succeq 0 \iff E'_j \succeq 0, \quad \forall j=1,\dots,3m \tag{153}$$

where $E'_j \in \mathbb{S}^j$ describes the j-th submatrix of E'. Moreover, a matrix E' is positive semidefinite, if the corresponding subdeterminants are greater or equal to zero. We get

$$E' \succeq 0 \iff d_j(E') \geq 0, \quad \forall j=1,\dots,3m \tag{154}$$

where $d_j(E')$, $j=1,\dots,3m$, is the determinant of E_j given by the Laplace formula

$$d_j(E') := \det\left(E'_j\right) = \sum_{q=1}^{j} (-1)^{p+q} k_{pq} \det\left(\left(E'_j\right)_{pq}\right), \tag{155}$$

and where k_{pq} is the element of E' in row p and column q. Moreover, $\left(E'_j\right)_{pq}$ is the submatrix of E'_j reduced by row p and column q, i.e.,

$$E'_{pq} := \begin{pmatrix} k_{11} & \dots & k_{1\ q-1} & k_{1\ q+1} & \dots & k_{1\ 3m} \\ \vdots & & \vdots & \vdots & & \vdots \\ k_{p-1\ 1} & \dots & k_{p-1\ q-1} & k_{p-1\ q+1} & \dots & k_{p-1\ 3m} \\ k_{p+1\ 1} & \dots & k_{p+1\ q-1} & k_{p+1\ q+1} & \dots & k_{p+1\ 3m} \\ \vdots & & \vdots & \vdots & & \vdots \\ k_{3m\ 1} & \dots & k_{3m\ q-1} & k_{3m\ q+1} & \dots & k_{3m\ 3m} \end{pmatrix} \tag{156}$$

It can be shown that the resulting functions $d_j(E'),\ j = 1, \dots, 3m$ are nonconvex and polynomial. As (154) holds, the feasible region defined by $d_j(E') \geq 0,\ j = 1, \dots, 3m$, is convex. The design variable E' is block diagonal, i.e., of the form

$$E' := \begin{pmatrix} \blacksquare & 0 & 0 & \cdots & 0 \\ 0 & \blacksquare & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \blacksquare & 0 \\ 0 & \cdots & \cdots & 0 & \blacksquare \end{pmatrix} \tag{157}$$

Therefore it is sufficient to show that each 3×3 block $\blacksquare$ of E' is positive semidefinite. The blocks are given by $E_i - \underline{\nu} I,\ i = 1, \dots, m$. We get the following three inequality constraints from (155) and (121)

$$d_{i1}(E_i - \underline{\nu} I) := e_{i1} - \underline{\nu} \geq 0 \tag{158}$$

$$d_{i2}(E_i - \underline{\nu} I) := (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \geq 0 \tag{159}$$

$$\begin{aligned} d_{i3}(E_i - \underline{\nu} I) := {} & (e_{i6} - \underline{\nu}) \big((e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \big) \\ & - e_{i4}^2 (e_{i3} - \underline{\nu}) - e_{i5}^2 (e_{i1} - \underline{\nu}) + 2 e_{i2} e_{i4} e_{i5} \geq 0 \end{aligned} \tag{160}$$

for each elementary stiffness matrix $E_i,\ i = 1, \dots, m$. As the first submatrix of a block $E_i - \underline{\nu} I,\ i = 1, \dots, m$, consists of only one element, it can be handled as box constraint. Therefore, $2m$ additional constraints have to be introduced.

The following problem formulation (161) is equivalent to (143) and can be solved efficiently by Algorithm 2.

$$
\begin{array}{llr}
\min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\
\text{s.t.} & \sum\limits_{i=1}^{m} \operatorname{Trace}(E_i) - V \leq 0 & \\
& \operatorname{Trace}(E_i) - \overline{\nu} \leq 0, & i = 1,\dots,m \\
& f_j^T K(E)^{-1} f_j - \alpha \leq 0, & j = 1,\dots,l \\
& (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \geq 0, & i = 1,\dots,m \\
& (e_{i6} - \underline{\nu})\left((e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2\right) & \\
& -e_{i4}^2(e_{i3} - \underline{\nu}) - e_{i5}^2(e_{i1} - \underline{\nu}) + 2e_{i2}e_{i4}e_{i5} \geq 0, & i = 1,\dots,m \\
& e_{i1} \geq \underline{\nu}, & i = 1,\dots,m \\
& e_{i3} \geq \underline{\nu}, & i = 1,\dots,m \\
& e_{i6} \geq \underline{\nu}, & i = 1,\dots,m
\end{array}
\tag{161}
$$

This problem is nonlinear and nonconvex but exhibits a convex feasible region, as the constraints $d_j(E') \geq 0,\ j = 1,\dots,3m$, describe a convex domain.

Applying Algorithm 2 to solve (161), we have to compute first and second order derivatives and evaluate the problem functions efficiently. The first subdeterminant is handled as a box constraint while the first and second order derivatives of $d_{i2}(E_i - \underline{\nu}I)$ and $d_{i3}(E_i - \underline{\nu}I),\ i = 1,\dots,m$, are given explicitly. We consider an arbitrary finite element $i \in \{1,\dots,m\}$ and the corresponding matrix E_i. The determinant of the (2×2) submatrix is dependent on three variables. We get

$$
d_{i2}(E_i - \underline{\nu}I) = (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \tag{162}
$$

$$
\frac{\partial d_{i2}(E_i - \underline{\nu}I)}{\partial e_{ip}} = \begin{pmatrix} e_{i3} - \underline{\nu} \\ -2e_{i2} \\ e_{i1} - \underline{\nu} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{163}
$$

$$
\frac{\partial d_{i2}(E_i - \underline{\nu}I)}{\partial e_{ip}\partial e_{iq}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{164}
$$

The determinant of the (3×3) matrix is dependent on all variables. Its first and second order derivatives are

$$\begin{aligned} d_{i3}\left(E_i - \underline{\nu} I\right) &:= \left(e_{i6} - \underline{\nu}\right)\left(\left(e_{i3} - \underline{\nu}\right)\left(e_{i1} - \underline{\nu}\right) - e_{i2}^2\right) \\ &\quad -e_{i4}^2\left(e_{i3} - \underline{\nu}\right) - e_{i5}^2\left(e_{i1} - \underline{\nu}\right) + 2e_{i2}e_{i4}e_{i5} \\ &= \left(e_{i6} - \underline{\nu}\right) d_{i2}\left(E_i - \underline{\nu} I\right) - e_{i4}^2\left(e_{i3} - \underline{\nu}\right) - e_{i5}^2\left(e_{i1} - \underline{\nu}\right) \\ &\quad +2e_{i2}e_{i4}e_{i5} \end{aligned} \tag{165}$$

$$\frac{\partial d_{i3}\left(E_i - \underline{\nu} I\right)}{\partial e_{ip}} = \begin{pmatrix} \left(e_{i3} - \underline{\nu}\right)\left(e_{i6} - \underline{\nu}\right) - e_{i5}^2 \\ 2e_{i4}e_{i5} - 2e_{i2}\left(e_{i6} - \underline{\nu}\right) \\ \left(e_{i1} - \underline{\nu}\right)\left(e_{i6} - \underline{\nu}\right) - e_{i4}^2 \\ 2e_{i2}e_{i5} - 2\left(e_{i3} - \underline{\nu}\right) e_{i4} \\ 2e_{i2}e_{i4} - 2e_{i5}\left(e_{i1} - \underline{\nu}\right) \\ d_{i2}\left(E_i - \underline{\nu} I\right) \end{pmatrix} \tag{166}$$

$$\frac{\partial d_{i3}\left(E_i - \underline{\nu} I\right)}{\partial e_{ip}\partial e_{iq}} =$$

$$\begin{pmatrix} 0 & 0 & e_{i6} - \underline{\nu} & 0 & -2e_{i5} & e_{i3} - \underline{\nu} \\ 0 & -2\left(e_{i6} - \underline{\nu}\right) & 0 & 2e_{i5} & 2e_{i4} & -2e_{i2} \\ e_{i6} - \underline{\nu} & 0 & 0 & -2e_{i4} & 0 & e_{i1} - \underline{\nu} \\ 0 & 2e_{i5} & -2e_{i4} & -2\left(e_{i3} - \underline{\nu}\right) & 2e_{i2} & 0 \\ -2e_{i5} & 2e_{i4} & 0 & 2e_{i2} & -2\left(e_{i1} - \underline{\nu}\right) & 0 \\ e_{i3} - \underline{\nu} & -2e_{i2} & e_{i1} - \underline{\nu} & 0 & 0 & 0 \end{pmatrix} \tag{167}$$

5 Numerical Implementation and Results

Algorithm 2 is implemented in Fortran and the subroutine is called SCPF10. The user has to provide function and gradient values for all functions, moreover the Hessian matrice of the feasibility constraints. The outer iteration sequence consists of the iterates computed by Algorithm 2. Note that the gradients are to be computed for constraints included in an estimate of the active set.

Subproblem (12) is solved iteratively. During this inner iterations, functions, gradients and second order derivatives are to be evaluated only for the feasibility constraints $e_j(x)$, $j = 1, \ldots, m_f$. The subproblems are solved by IPOPT, see Wächter and Biegler [52].

We proceed from the FMO problem (161), now written in the form i.e.,

$$
\begin{array}{lll}
\min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\
\text{s.t.} & \sum\limits_{i=1}^{m} \operatorname{Trace}(E_i) - V \leq 0 & \\
& \operatorname{Trace}(E_i) - \overline{\nu} \leq 0, & i = 1, \ldots, m \\
& f_j^T K(E)^{-1} f_j - \alpha \leq 0, & j = 1, \ldots, l \\
& -d_{i2}(E_i - \underline{\nu} I) \leq -\varepsilon, & i = 1, \ldots, m \\
& -d_{i3}(E_i - \underline{\nu} I) \leq -\varepsilon, & i = 1, \ldots, m \\
& e_{ip} \leq \overline{e}, & i = 1, \ldots, m, \quad p = 1, \ldots, 6 \\
& e_{ip} \geq \underline{e}, & i = 1, \ldots, m, \quad p = 2, 4, 5 \\
& e_{ip} \geq \underline{e}_d, & i = 1, \ldots, m, \quad p = 1, 3, 4 \\
& \underline{\alpha} \leq \alpha \leq \overline{\alpha} &
\end{array}
\tag{168}
$$

with

$$
\begin{aligned}
d_{i2}(E_i - \underline{\nu} I) &:= (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 && (169) \\
d_{i3}(E_i - \underline{\nu} I) &:= (e_{i6} - \underline{\nu})\left((e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2\right) \\
&\quad - e_{i4}^2 (e_{i3} - \underline{\nu}) - e_{i5}^2 (e_{i1} - \underline{\nu}) + 2 e_{i2} e_{i4} e_{i5} && (170)
\end{aligned}
$$

The parameter $\varepsilon \in \mathbb{R}^+$ is introduced to prevent numerical instabilities in case of vanishing material and is a small tolerance. Otherwise, the linear independency constraint qualification might be violated. Moreover, we introduce box constraints for each variable to ensure a compact feasible set F, defined by

$$
\begin{aligned}
F \;:=\; & \{e_{ip} \in \mathbb{R},\ i = 1, \ldots, m,\ p = 1, \ldots, 6 \mid -d_{i2}(E_i - \underline{\nu} I) \leq -\varepsilon\} \\
& \cap \{e_{ip} \in \mathbb{R},\ i = 1, \ldots, m,\ p = 1, \ldots, 6 \mid -d_{i3}(E_i - \underline{\nu} I) \leq -\varepsilon\} \\
& \cap X
\end{aligned}
\tag{171}
$$

with

$$
\begin{aligned}
X \;:=\; & \{e_{ip} \in \mathbb{R},\ i = 1, \ldots, m,\quad p = 2, 4, 5 \mid \underline{e} \leq e_{ip} \leq \overline{e}\} \\
& \cap \{e_{ip} \in \mathbb{R},\ i = 1, \ldots, m,\quad p = 1, 3, 4 \mid \underline{e}_d \leq e_{ip} \leq \overline{e}\} \\
& \cap \{\alpha \in \mathbb{R} \mid \underline{\alpha} \leq \alpha \leq \overline{\alpha}\}
\end{aligned}
\tag{172}
$$

Note that the lower bound on diagonal entries ensures that $e_{i1} - \underline{\nu} \neq 0$, i.e., $\underline{e}_d > \underline{\nu}$.

Variables are stored in stacked form $x \in \mathbb{R}^{6m+1}$, i.e., six variables for each E_i, $i = 1, \ldots, m$, and the additional variable α,

$$
x \;:=\; (e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{21}, e_{22}, \ldots, e_{m6}, \alpha)^T . \tag{173}
$$

All parameters of (168) are summarized in Table 1.

Parameter	Value
$E^{(0)}$	$10I$
$\alpha^{(0)}$	1.2
$\overline{e}$	1.D5
$\underline{e}$	-1.D5
$\underline{e}_d$	0.33333
$\overline{\alpha}$	1.D5
$\underline{\alpha}$	0.D0
ε	1.D-1
$\overline{\nu}$	100
$\underline{\nu}$	0.3333
V	$0.3333\overline{\nu}$
s_σ	1.D1

Table 1: Parameters solving free material optimization problems

The active set parameter a is set to 1.1, while the stopping accuracy is 1.D-5.

The nonlinear FMO problem (168) is to be solved by Algorithm 2. The algorithm was integrated into the PLATO-N interface, see Boyd [13]. We consider given test cases of the PLATO-N academic test case library, see Bogomolny [12]. An overview of the test set is given in Figure 1. Fixed nodes are denoted by ▶ or ▲ , while allows to move in horizontal direction. Loads are specified by arrows pointing at the corresponding node. In case of several load cases the loads are enumerated. Note that the test case library does not always use the same discretization granularity for the test cases.

Figure 1: Test cases, see Bogomolny [12]

Some results are illustrated by plotting the traces of each finite element, see Bodnár [11], and some further data and numerical results are listed in Table 2.

Figure 2: Color Scheme

Figure 3: Single Load Problem 1 with 384 Elements

Figure 4: Single Load Problem 2 with 9.500 Elements

Test case	Variables	Regular Constraints	Feasibility Constraints	Total Constraints	Iterations	Time	Load cases
1.1	577	98	192	290	123	2min 19sec	1
1.2	2.305	386	768	1.154	274	25min 14sec	1
2.1	577	98	192	290	284	5min 38sec	1
2.2	2.305	386	768	1.154	148	12min 33sec	1
2.3	57.601	9.602	19.200	28.802	331	17h 36min	1
3.1	577	99	192	291	108	2min 5sec	2
3.2	2.305	387	768	1.155	419	39min 4sec	2
3.3	57.601	9.603	19.200	28.803	400	20h 48min	2
4.1	1.801	302	600	902	185	10min 42sec	1
4.1stress	1.801	602	600	1.202	214	15min 34sec	1
4.2	7.801	1.202	2.400	3.602	183	38min 23sec	1
4.3	45.001	7.502	15.000	22.502	210	8h 22min	1
5.1	4.801	802	1.600	2.402	111	19min 55sec	1
5.2	120.001	20.002	40.000	60.002	300	48h 41min	1

Table 2: Data of numerical results

6 Conclusions

We propose a method by which objective function and constraints are always evaluated at iterates, which are feasible subject to another set of convex inequality constraints, the so-called feasibility constraints. To our knowledge, this approach is new. Proceeding from an augmented Lagrangian merit function, sufficient decrease from one iterate to the next is shown, and we are able to conclude that a stationary point is approximated.

Our main motivation is to solve free material optimization problems, where the positive

Figure 5: Single Load Problem 3 with 9.500 Elements

Figure 6: Single Load Problem 4 with 7.500 elements

Figure 7: Single Load Problem 5 with 20.000 Elements

definite tensors are guaranteed by non-negative determinants of submatrices. Unfortunately, the feasibility functions are no longer convex as requested by our FSCP method, but, on the other hand, the feasible set F is at least convex. We did not observe irregularities due to this drawback.

Feasibility constraints are passed to the analytical, convex, and separable subproblem of a sequential convex programming algorithm, which then becomes a general convex program. However, the special structure can be exploited depending on the structure of the feasibility constraints, e.g., if they consist of simple analytical functions. In the same way, it is possible to replace the outer SCP method by another one, e.g., an SQP method.

From the theoretical as well as the practical point of view, the assumption that requirement feasibility constraints are convex, is quite restrictive. We need it due to the line search procedure used. By applying other globalization techniques, e.g., a filter method, we might overcome the convexity condition.

Although our implementation is efficient in case of 2D structures, additional efforts are needed to improve computational performance in order to solve also 3D FMO problems of reasonable size. A reduction of the calculation time is essential and could be achieved by exploiting the special subproblem structure more than we did so far.

References

[1] W. Achtziger and C. Kanzow. Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications. Technical report, Institute of Applied Mathematics and Statistics, University of Würzburg, Germany, 2005.

[2] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1), 1966.

[3] S. Bakhtiari and A. Tits. A simple primal-dual feasible interior-point method for nonlinear programming with monotone descent. *Computational Optimization and Applications*, 25:17–38, 2003.

[4] A. Ben-Tal, M. Kočvara, A. Nemirovski, and J. Zowe. Free material design via semidefinite programming. The multi-load case with contact conditions. *SIAM Journal Optimization*, 9(4):813–832, 1999.

[5] M.P. Bendsøe. *Optimization of Structural Topology, Shape and Material.* Springer, Heidelberg, 1995.

[6] M.P. Bendsøe and A.R. Díaz. Optimization of material properties for Mindlin plate design. *Structural Optimization*, 6:268–270, 1993.

[7] M.P. Bendsøe, J.M. Guedes, 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. Appl. Mech.*, 61:930–937, 1994.

[8] M.P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications.* Springer-Verlag Berlin, 2003.

[9] H.Y. Benson and R.J. Vanderbei. Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming. *Mathematical Programming*, Ser B 95:279–302, 2003.

[10] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

[11] G. Bodnár. D7 - efficient algorithms for visualising FMO results. Technical report, PLATO-N public report PU-R-5-2007, 2007.

[12] M. Bogomolny. D34 - test case library PLATOlib. Technical report, PLATO-N public report PU-R-6-2008, 2008.

[13] R. Boyd. Specification of aircraft topology optimisation system, PLATO-N. Technical report, PLATO-N public report PU-R-2-2007, 2007.

[14] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1979.

[15] S. Ertel. Anwendungen von Filtermethoden auf das Optimierungsverfahren SCP. Diplomarbeit, Mathematisches Institut, Universität Bayreuth, 2006.

[16] S. Ertel, K. Schittkowski, and Ch. Zillober. D14 - sequential convex programming for free material optimization with displacement and stress constraints. Technical report, PLATO-N public report PU-R-2-2008, 2008.

[17] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, 2002.

[18] R. Fletcher, P.L. Toint, and S. Leyffer. On the global convergence of a SQP-filter algorithm. *SIAM J. Optim.*, 13(1), 2002.

[19] C. Fleury. CONLIN: an efficient dual optimizer based on convex approximation concepts. *Structural Optimization*, 1:81–89, 1989.

[20] C. Fleury and V. Braibant. Structural Optimization – a new dual method using mixed variables. *Int. J. Num. Meth. Eng.*, 23:409–428, 1986.

[21] J. Herskovits. A two-stage feasible directions algorithm for nonlinear constrained optimization. *Mathematical Programming*, 36(1):19–38, 1986.

[22] J. Herskovits. Feasible direction interior-point technique for nonlinear optimization. *Journal of Optimization Theory and Applications*, 99(1):121–146, 1998.

[23] J. Herskovits, J.M. Aroztegui, E. Goulart, and V. Dubeux. Large scale structural optimization. *Variational Formulations in Mechanics: Theory and Applications*, 2006.

[24] H. Hörnlein, M. Kočvara, and R. Werner. Material optimization: bridging the gap between conceptual and preliminary design. *Aerospace Science and Technology*, 5, 2001.

[25] F. Jarre and J. Stoer. *Optimierung*. Springer-Verlag Berlin, 2004.

[26] J.-B. Jian, K. Zhang, and S. Xue. A superlinearly and quadratically convergent SQP type feasible method for constrained optimization. *Appl. Math. J. Chinese Univ. Ser. B*, 15(3):319–331, 2000.

[27] M. Kočvara, A. Beck, A. Ben-Tal, and M. Stingl. PLATO-N work package 4: FMO models, task 2.2: Software specification, selection of FMO problem formulations. Report, 2007.

[28] M. Kočvara and M. Stingl. PENNON - a generalized augmented Lagrangian method for semidefinite programming. Report, University Erlangen, 2001.

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

[30] M. Kočvara and M. Stingl. Free Material Optimization: towards the stress constraints. *Structural and Multidisciplinary Optimization*, 33(4-5):323–335, 2007.

[31] M. Kočvara and J. Zowe. *Free Material Optimization: an overview.* In A.H. Siddiqi and M. Kočvara, eds., Trends in Industrial and Applied Mathematics, Kluwer Academic Publishers, 2002.

[32] S. Lehmann. *A Strictly Feasible Sequential Convex Programming Method for Free Material Optimization.* Dissertation, University of Bayreuth, Fakulty of Mathematics, Physics, and Informatics, 2011.

[33] Q. Li, G.P. Steven, and Y.M. Xie. On equivalence between stress criterion and stiffness criterion in evolutionary structural optimization. *Structural Optimization*, 18, 1999.

[34] J. Mach. Finite element analysis of free material optimization problem. *Applications of Mathematics*, 49(4):285–307, 2004.

[35] Q. Ni. A globally convergent method of moving asymptotes with trust region technique. *Optimization Methods and Software*, 18:283–297, 2003.

[36] Q. Ni, Ch. Zillober, and K. Schittkowski. Sequential convex programming methods for solving large topology optimization problems: implementation and computational results. *Journal of Computational Mathematics*, 23(5):491–502, 2005.

[37] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in applied mathematics. Academic Press, New York, 1970.

[38] E.R. Panier and A.L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, 59:261–276, 1993.

[39] E.R. Panier, A.L. Tits, and J.N. Herskovits. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Control and Optimization*, 2(4), 1988.

[40] E. Polak. *Computational Methods in Optimization*. Academic Press, 1971.

[41] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[42] J. B. Rosen. The gradient projection method for nonlinear programming. part ii. nonlinear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961.

[43] K. Schittkowski. On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function. *Optimization*, 14, 1983.

[44] K. Schittkowski, Ch. Zillober, and R. Zotemantel. Numerical comparison of nonlinear programming algorithms for structural optimization. *Structural Optimization*, 7:1–19, 1994.

[45] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.

[46] M. Stingl. *On the Solution of Nonlinear Semidefinite Programs by Augmented Lagrangian Methods*. Dissertation, Shaker Verlag, 2006.

[47] M. Stingl, M. Kočvara, and G. Leugering. A sequential convex semidefinite programming algorithm with an application to multiple-load free material optimization. *SIAM Journal on Optimization*, 20(1):120–155, 2009.

[48] K. Svanberg. The Method of Moving Asymptotes – a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.

[49] K. Svanberg. A globally convergent version of MMA without linesearch. In N. Olhoff and G.I.N. Rozvany, editors, *Proceedings of the First World Congress of Structural and Multidisciplinary Optimization*, pages 9–16. Pergamon, 1995.

[50] K. Svanberg. A new globally convergent version of the method of moving asymptotes. Technical Report TRITA/MAT-99-OS2, Department of Mathematics, KTH, Stockholm, 1999.

[51] K. Svanberg. A class of globally convergent optimization methods based on conservate convex separable approximations. *SIAM J. Optimization*, 12(2):555–573, 2002.

[52] A. Wächter and L.T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[53] R. Werner. *Free Material Optimization - Mathematical Analysis and Numerical Solution*. PhD thesis, Institut für Angewandte Mathematik II, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2001.

[54] Z. Zhu. An efficient sequential quadratic programming algorithm for nonlinear programming. *Journal of Computational and Applied Mathematics*, 175:447–464, 2005.

[55] Ch. Zillober. *Eine global konvergente Methode zur Lösung von Problemen aus der Strukturoptimierung*. PhD thesis, Technische Universität München, 1992.

[56] Ch. Zillober. A globally convergent version of the method of moving asymptotes. *Structural Optimization*, 6(3):166–174, 1993.

[57] Ch. Zillober. *Numerical Solution of Nonlinear Programming Problems by Convex Approximation Methods*. Habilitation, University of Bayreuth, 2001.

[58] Ch. Zillober. SCPIP - an efficient software tool for the solution of structural optimization problems. *Structural and Multidisciplinary Optimization*, 24(5), 2002.

[59] Ch. Zillober. Software manual for SCPIP 3.0. Technical report, 2004.

[60] Ch. Zillober, K. Schittkowski, and K. Moritzen. Very large scale optimization by sequential convex programming. *Optimization Methods and Software*, 19(1):103–120, 2004.

[61] G. Zoutendijk. *Methods of Feasible Directions*. Elsevier, Amsterdam, 1970.

[62] J. Zowe, M. Kočvara, and M. Bendsøe. Free material optimization via mathematical programming. *Mathematical Programming, Series B*, 79:445–466, 1997.