A Strictly Feasible Sequential Convex Programming Method

Von der Universität Bayreuth
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

vorgelegt von

Sonja Lehmann

geboren in Erlangen

1. Gutachter: Prof. Dr. Klaus Schittkowski
2. Gutachter: Prof. Dr. Michael Stingl

Tag der Einreichung: 9. Juni 2011
Tag des Kolloquiums: 18. November 2011

Fakultät für Mathematik, Physik und Informatik
Angewandte Informatik VII

Abstract

In free material optimization (FMO), one tries to find the best mechanical structure by minimizing the weight or by maximizing the stiffness with respect to given load cases. 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, leading to positive definite elasticity tensors, which may be arbitrarily small in case of vanishing material. To guarantee a positive definite global stiffness matrix for computing design constraints, it is required that all iterates of an optimization algorithm retain positive definite tensors. Otherwise, some constraints, e.g., the compliance, cannot be evaluated and the algorithm fails.

FMO problems are generalizations of topology optimization problems. The goal of topology optimization is to find the stiffest structure subject to given loads and a limited amount of material. In contrast to FMO the material is explicitly given and cannot vary. Based on a finite element discretization, in each element it is decided whether to use material or not. The regions with vanishing material are interpreted as void. The resulting optimization problem can be solved by numerous efficient nonlinear optimization methods, for example sequential convex programming methods.

Sequential convex programming (SCP) formulates separable and strictly convex nonlinear subproblems iteratively by approximating the objective and the constraints. Lower and upper asymptotes are introduced to truncate the feasible region. Due to the special structure, the resulting subproblems can be solved efficiently by appropriate methods, e.g., interior point methods. To ensure global convergence, a line search procedure is introduced. Moreover, an active set strategy is applied to reduce computation time.

The iterates of SCP are not guaranteed to be inside the corresponding feasible region described by the constraints. As a consequence it is not able to solve free material optimization problems as the compliance function is only well-defined on the feasible region of some of the constraints.
We propose a modification of a SCP method that ensures feasibility with respect to a given set of inequality constraints. The resulting procedure is called feasible sequential convex programming method (SCPF). SCPF expands the resulting subproblems by additional nonlinear constraints, that are passed to the subproblem directly to ensure their feasibility in each iteration step. They are referred as feasibility constraints. In addition, other constraints may be violated within the optimization process. As globalization technique a line search procedure is used to ensure convergence. The

The research was 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

resulting subproblems can be solved efficiently taking the sparse structure into account. Moreover, semidefinite constraints have to be replaced by nonlinear ones, such that SCPF is applicable. SCPF successfully solved FMO problems with up to 120.000 variables and 60.000 constraints. Within a theoretical analysis global convergence of SCPF is shown for convex feasibility constraints.

The research was 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

Zusammenfassung

Ziel dieser Dissertation ist die Entwicklung eines effizienten Lösungsverfahrens für komplexe Optimierungsprobleme aus der Freien Materialoptimierung. Dabei handelt es sich um eine spezielle Problemstellung aus dem Bereich der mechanischen Strukturoptimierung. Aus einer vorgegebenen Menge an Material soll die stabilste Struktur eines Objekts, z.B. eines Bauteils, berechnet werden. Zu den Anwendungen zählen unter anderem der Fahrzeug- und Flugzeugbau. Die Variablen sind Elastizitätstensoren, die die Materialeigenschaften des zu optimierenden Objekts in jedem Element einer vorgegebenen Finite Elemente Approximation widerspiegeln. Diese können durch eine symmetrische 3×3 Matrix bzw. 6×6 Matrix dargestellt werden. Um den physikalischen Gesetzmäßigkeiten zu genügen, müssen diese Matrizen bestimmte mathematische Bedingungen erfüllen. Im Gegensatz zu anderen Problemklassen in der Strukturoptimierung sind die Materialeigenschaften nicht vorgegeben. Stattdessen ist die Wahl des Materials Teil der Optimierung, so dass in jedem Element unterschiedliches Material gewählt werden kann. Die Freie Materialoptimierung ist eine Verallgemeinerung der Topologieoptimierung. Bei der Topologieoptimierung ist das Material zur Bestimmung der optimalen Struktur für ein beliebiges Objekt unter Einfluss von verschiedenen Kräften vorgegeben. Im Gegensatz zur Freien Materialoptimierung existieren für die Topologieoptimierung geeignete effiziente Lösungsverfahren, die Problemstellungen mit einer großen Anzahl von Variablen und Nebenbedingungen lösen können. Da die Optimierungsvariablen für Probleme der Freien Materialoptimierung aus Elastizitätstensoren bestehen, können die bekannten effizienten Verfahren der Topologieoptimierung nicht in der Freien Materialoptimierung eingesetzt werden. Daher wird eine Weiterentwicklung des Optimierungsverfahrens 'Sequential Convex Programming' (SCP) vorgestellt.

In der ursprünglichen Form zählt das SCP Verfahren zu den effizientesten Lösungsansätzen für Probleme der Topologieoptimierung. Der Algorithmus approximiert ein allgemeines nichtlineares Optimierungsproblem durch eine Folge streng konvexer, separabler Teilprobleme. Diese Teilprobleme lassen sich auf Grund ihrer Eigenschaften und ihrer Struktur effizient lösen. Iterativ wird aus der Lösung eines vorangegangenen Teilproblems ein neues formuliert. Unter bestimmten Voraussetzungen konvergiert die Folge der Lösungen der Teilprobleme gegen die optimale Lösung des Ausgangsproblems. Um globale Konvergenzaussagen zu erhalten, wird eine Schrittweitensteuerung angewendet, die eine Verbesserung der aktuellen Iterierten garantiert.

Das SCP Verfahren ist für die Freie Materialoptimierung nicht anwendbar, weshalb der Algorithmus umfassend weiterentwickelt werden muss. Da das ursprüngliche Verfahren semidefinite Nebenbedingungen nicht berücksichtigen kann, müssen diese Nebenbedingungen geeignet umformuliert werden. Von zentraler Bedeutung für Pro-

The research was 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

bleme aus der Freien Materialoptimierung ist es, dass bestimmte Nebenbedingungen in jedem Iterationsschritt erfüllt sind, da gewisse Funktionen und deren Gradienten nur dann berechnet werden können. Das in dieser Arbeit entwickelte, strikt zulässige SCP Verfahren (SCPF, für Feasible Sequential Convex Programming) garantiert die Zulässigkeit einer Menge von konvexen Nebenbedingungen in jeder Iteration. Diese Nebenbedingungen werden im Folgenden als strikt zulässige Nebenbedingungen bezeichnet. SCPF integriert die strikt zulässigen Nebenbedingungen direkt in das Teilproblem, während die übrigen Nebenbedingungen und die Zielfunktion durch konvexe, separable Funktionen approximiert werden. Dadurch wird sichergestellt, dass alle Iterationspunkte innerhalb der zulässigen Menge liegen, die von den strikt zulässigen Nebenbedingungen beschrieben wird. Durch die Einführung zweier flexibler Asymptoten wird der zulässige Bereich der Teilprobleme zusätzlich eingeschränkt. Das resultierende Teilproblem besitzt eine eindeutige Lösung und kann aufgrund seiner besonderen Struktur effizient mit Inneren Punkte Methoden gelöst werden. Das Verfahren SCPF wurde auf Probleme der Freien Materialoptimierung angewendet und hat Probleme mit bis zu 120.000 Variablen und 60.000 Nebenbedingungen erfolgreich gelöst. Außerdem können globale Konvergenzeigenschaften für convexe strikt zulässige Nebenbedingungen gezeigt werden.

The research was 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

CONTENTS

LIST OF SYMBOLS

SCPF	feasible sequential convex programming method, 1
FMO	free material optimization, 1
n	number of primal variables, 1
$x \in \mathbb{R}^n$	primal variable of dimension n, 1
$f(x)$	objective function, 1
m_e	number of equality constraints, 1
$c_j(x), \ j = 1, \ldots, m_e$	equality constraints, 1
m_c	number of equality and inequality constraints, 1
$c_j(x), \ j = m_e + 1, \ldots, m_c$	inequality constraints, 1
m_f	number of feasibility constraints, 1
$e_j(x), \ j = 1, \ldots, m_f$	feasibility constraints, 1
F	feasible set given by the feasibility constraints $e_j(x)$, 1
NSDP	nonlinear semidefinite programming, 2
l	number of load cases, 2
$f_j, \ j = 1, \ldots, l$	set of loads, 2
$K(E)$	global stiffness matrix, 2
m	number of finite elements, 2
$E_i, \ i = 1, \ldots, m$	elasticity matrices, 2
SCP	sequential convex programming method, 2
CONLIN	convex linearization method, 2
MMA	method of moving asymptotes, 2
SQP	sequential quadratic programming method, 3
GCMMA	globally convergent method of moving asymptotes, 3
$^{(k)}$	iteration index, 5
$d^{(k)} \in \mathbb{R}^{n+m_f+m_c}$	search direction with respect to primal and dual variables in iteration k (SQP and FSQP: with respect to primal variables only, i.e., $d^{(k)} \in \mathbb{R}^n$.), 5
$\theta > 0$	positive parameter to define feasible direction, 6
$\sigma \in (0, 1]$	stepsize to yield a descent in the merit function, 6

$y \in \mathbb{R}^{m_c+m_f}$	dual variable, 7
IPM	interior point method, 7
$\mathbb{F}$	feasible region, 11
$\mathbb{J}\left(x\right)$	set of active constraints, 12
$L\left(x,y\right)$	Lagrangian function, 12
$y_c \in \mathbb{R}^{m_c}$	dual variables with respect to constraints $c_j\left(x\right),$ $j = 1,\ldots,m_c$, 12
$y_e \in \mathbb{R}^{m_f}$	dual variables with respect to constraints $e_j\left(x\right),$ $j = 1,\ldots,m_f$, 12
$H^{(k)} = H\left(x^{(k)}, y^{(k)}\right)$	Hessian of Lagrangian with respect to x or adequate approximation in iteration k, 13
LICQ	linear independence constraint qualification, 13
KKT	Karush-Kuhn-Tucker first order optimality conditions, 13
$z^{(k)} \in \mathbb{R}^n$	primal solution of the subproblem in iteration k, 14
$v^{(k)} \in \mathbb{R}^{m_c+m_f}$	dual solution of the subproblem in iteration k , 14
$r \in (0,1)$	positive parameter used in Armijo condition, 14
FSQP	feasible sequential quadratic programming method, 17
QP	quadratic programming, 17
$d_0^{(k)} \in \mathbb{R}^n$	solution of QP in iteration k, 17
$\mathbb{J}^{(k)} = \mathbb{J}\left(x^{(k)}\right)$	set of active constraints in iteration k, 18
t	vector of weighting factors of appropriate size, 19
$A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{n\times\left\lvert\mathbb{J}^{(k)}\right\rvert}$	matrix of gradients of active constraints with respect to $\mathbb{J}^{(k)}$ in iteration k, 22
$e_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{\left\lvert\mathbb{J}^{(k)}\right\rvert}$	vector of active constraints with respect to $\mathbb{J}^{(k)}$ in iteration k, 22
$\mathbb{1}$	vector of ones of appropriate size, 23
$\mathbb{0}$	vector of zeros of appropriate size, 29
$U_i,\ i = 1,\ldots,n$	upper asymptote for primal variable $x_i,\ i = 1,\ldots,n$, 40
$L_i,\ i = 1,\ldots,n$	lower asymptote for primal variable $x_i,\ i = 1,\ldots,n$, 40
$I_+^{(k)}$	index set of nonnegative partial derivatives of objective in iteration k, 40
$I_-^{(k)}$	index set of negative partial derivatives of objective in iteration k, 40
$f^{(k)}\left(x\right)$	MMA / SCP / GCMMA / SCPF approximation of $f\left(x\right)$ in iteration k, 40
$I_+^{(j,k)}$	index set of nonnegative partial derivatives of inequality constraint $c_j\left(x\right),\ j = m_e+1,\ldots,m_c$, in iteration k, 41

$I_{-}^{(j,k)}$	index set of negative partial derivatives of inequality constraint $c_j\left(x\right),\ j = m_e+1,\dots,m_c$, in iteration k, 41
$c_j^{(k)}\left(x\right),\ j=1,\dots,m_c$	MMA / SCP / GCMMA / SCPF approximation of constraint $c_j\left(x\right),\ j=1,\dots,m_c$, in iteration k, 41
$\underline{x}_i^{(k)},\ i=1,\dots,n$	lower bound on primal variable $x_i,\ i=1,\dots,n$, for the subproblem generated in iteration k, 41
$\overline{x}_i^{(k)},\ i=1,\dots,n$	upper bound on primal variable $x_i,\ i=1,\dots,n$, for the subproblem generated in iteration k, 41
$\omega \in]0;1[$	constant to define minimal distance between box constraints of the subproblem and asymptotes, 41
$X^{(k)}$	domain of variables $x \in \mathbb{R}^n$ of subproblem formulated in iteration k, 41
T_1, T_2	positive parameters to define update of asymptotes, 43
$\Phi_\rho\left(x,y\right)$	augmented Lagrangian merit function, 44
$\rho \in \mathbb{R}^{m_c+m_f}$	penalty parameters, 44
$J^{(k)}$	set of active constraints with respect to augmented Lagrangian in iteration k, 44
$\overline{J}^{(k)}$	set of inactive constraints with respect to augmented Lagrangian in iteration k, 44
$\tau > 0$	positive parameter to ensure strict convexity of $f^{(k)}\left(x\right)$, 45
$\eta_i^{(k)} \geq \eta^{(k)} > 0, i=1,\dots,n$	estimation of curvature of $f^{(k)}\left(x\right)$ in variable x_i, $i=1,\dots,n$, in iteration k, 48
$\delta^{(k)}$	norm of primal search direction in iteration k, 48
$\kappa_1,\ \kappa_2$	positive parameters for penalty update, 49
ξ	positive constant value to prevent steep approximations, 49
$L_{\min}$	lower bound on asymptotes $L_i^{(k)},\ i=1,\dots,n$, in each iteration k, 49
$U_{\max}$	upper bound on asymptotes $U_i^{(k)},\ i=1,\dots,n$, in each iteration k, 49
T_3, T_4	positive parameters to define convex approximation of GCMMA, 52
F	feasible region with respect to feasibility constraints, 56
$F_X^{(k)}$	feasible region of subproblem with respect to feasibility constraints and box constraints in iteration k, 56
$\rho_c \in \mathbb{R}^{m_c}$	penalty parameters for equality and inequality constraints, 57
$\rho_e \in \mathbb{R}^{m_f}$	penalty parameters for feasibility constraints, 57

$\Delta x^{(k)} \in \mathbb{R}^n$	search direction of primal variables in iteration k, 61
$R_{f^{(k)}}(x), R_{c_j^{(k)}}(x), R_{e_j}(x)$	residual of Taylor series to the corresponding functions $f^{(k)}(x), c_j^{(k)}(x),\ j = 1, \ldots, m_c$ and $e_j(x),\ j = 1, \ldots, m_f$, 64
$\kappa^{(k)}$	smallest singular value of Jacobian of the active constraints in iteration k, 81
$y_{\max}$	maximal Lagrangian multiplier, 82
$\Delta y_c \in \mathbb{R}^{m_c}$	search direction of dual variable with respect to inequality constraints $c_j(x),\ j = 1, \ldots, m_c$, 82
$\Delta y_e \in \mathbb{R}^{m_f}$	search direction of dual variable with respect to feasibility constraints $e_j(x),\ j = 1, \ldots, m_f$, 82
$\mathbb{S}^p$	space of symmetric matrices of size p, 109
$\mathbb{S}^p_+$	space of symmetric positive semidefinite matrices of size p, 109
$\mathbb{S}^p_{++}$	space of symmetric positive definite matrices of size p, 109
Ω	domain space for FMO, 109
FE	finite element discretization, 109
q	number of nodes of finite element discretization, 109
E	block diagonal matrix consisting of matrices E_i, $i = 1, \ldots, m$, 110
n_g	number of Gauss integration points, 111
$u_j \in \mathbb{R}^{2q},\ j = 1, \ldots, l$	displacement vector for corresponding load f_j, $j = 1, \ldots, l$, 111
I	identity matrix of appropriate size, 112
$\underline{\nu}\in \mathbb{R}^+$	positive value to prevent numerical instabilities, 112
$\alpha \in \mathbb{R}$	additional variable for the multiple load case, 112
V	amount of given material, 112
$\overline{\nu}$	upper bound to prevent numerical instabilities, 112
$s_{i,j}(E)$	stress constraints for element $i \in \{1, \ldots, m\}$ and load case $j \in \{1, \ldots, l\}$, 114
$s_\sigma \in \mathbb{R}^+$	upper bound on stress constraints, 114
E'	$E - \underline{\nu} I$, 115
$L(E') \Lambda(E') L(E')^T$	eigenvalue decomposition of E', 115
$L(E') = L$	lower triangular matrix of the eigenvalue decomposition of E', 115
$\Lambda(E')$	diagonal matrix containing eigenvalues of E', 115
$\lambda_j(E'),\ j = 1, \ldots, 3m$	eigenvalues of E', 115
E'_{j-1}	submatrix of E' of size $j - 1$, 116

k_{jj}	j-th diagonal entry of E', 116
k^j	first $j-1$ elements of j-th column of E', 116
L_{j-1}	submatrix of L of size $j-1$, 116
l^j	first $j-1$ elements of j-th column of L, 116
$\mathbb{1}_p$	p-th unity vector, 117
$d_j\left(E'\right),\ j=1,\ldots,3m$	determinant of E'_j, 119
k_{pq}	element of E' in row p and column q, 119
$\left(E'_j\right)_{pq}$	submatrix of E'_j reduced by row p and column q, 119
$a \in \mathbb{R}^+$	positive parameter to define active constraints for the corresponding active set strategy, 123
$\mathbb{A}^{(k)}$	active set, 124
$\mathbb{L}$	set of linear inequality constraints, 124
$\overline{\mathbb{L}}$	set of nonlinear inequality constraints, 124
$\mathbb{M}_e^{(k)}$	set of violated equality constraints in iteration k, 125
$\lvert\mathbb{M}_e^{(k)}\rvert$	number of violated equality constraints in iteration k, 125
$\mathbb{M}_c^{(k)}$	set of violated inequality constraints in iteration k, 125
$\lvert\mathbb{M}_c^{(k)}\rvert$	number of violated inequality constraints in iteration k, 125
$\mathbb{M}^{(k)}$	set of violated equality and inequality constraints in iteration k, 125
$\lvert\mathbb{M}^{(k)}\rvert$	number of violated equality and inequality constraints in iteration k, 125
$\gamma \in \mathbb{R}^{\lvert\mathbb{M}^{(k)}\rvert}$	additional variable to ensure feasibility of infeasible subproblems, 125

1. INTRODUCTION

In this thesis a strictly *feasible sequential convex programming* algorithm (SCPF) is presented. The goal is to generate an iteration sequence which is strictly feasible for a special class of constraints, called feasibility constraints, while other constraints may be violated during the iteration process. The algorithm is motivated by applications in free material optimization (FMO), where some constraints and the objective function can only be evaluated, if certain feasibility constraints are satisfied. Other typical applications are square roots or logarithmic functions of analytical expressions. We proceed from the following problem formulation

$$\begin{array}{lll} \min\limits_{x} & f\left(x\right) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j\left(x\right) = 0, & j = 1, \ldots, m_e \\ & c_j\left(x\right) \leq 0, & j = m_e + 1, \ldots, m_c \\ & e_j\left(x\right) \leq 0, & j = 1, \ldots, m_f \end{array} \tag{1.1}$$

where the feasibility constraints $e_j\left(x\right),\ j = 1, \ldots, m_f$, are at least twice continuously differentiable on $\mathbb{R}^n$. We assume that some of the constraints $c_j\left(x\right),\ j = 1, \ldots, m_c$, and the objective function $f\left(x\right)$ can only be evaluated on the feasible set

$$F \;:=\; \left\{x \in \mathbb{R}^n \mid e_j\left(x\right) \leq 0,\ j = 1, \ldots, m_f\right\}. \tag{1.2}$$

In addition, the regular constraints $c_j(x),\ j = 1, \ldots, m_c$ and the objective function $f(x)$, are at least twice continuously differentiable on F. Moreover, box constraints can be added to optimization problem (1.1), which is typically the case in practical applications.

The development of SCPF is motivated by problems arising in free material optimization (FMO), 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 [61]. Topology optimization problems are large scale nonlinear programs, that can be solved efficiently by appropriate algorithms, e.g., the method of moving asymptotes, see Svanberg [80] and Sigmund [77]. 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 [107]. 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 [50], 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 [40] and Kočvara and Zowe [51]. 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* (SCPF) method is an extension of the *sequential convex programming* (SCP) method, which is frequently used in mechanical engineering. SCP does not ensure feasibility of its iterates, i.e., $m_f = 0$. 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 [105].

SCP methods are derived from the optimization method CONLIN (CONvex LINearization), see Fleury and Braibant [29] and Fleury [28]. 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. As the success of CONLIN is dependent on the starting point and the method might end in oscillation, Svanberg [80] 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, as no convergence proof can be given for MMA. The iterates are valuated with respect 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.f 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 [76]. For the resulting SCP method global convergence is shown, see Zillober [97, 102].

Although no convergence proof for the original version of MMA can be given, the algorithm yields good results in practice. In 1995, Svanberg [81] presented an extension which is globally convergent but in most cases not as efficient as the original MMA version. Later on, a new globally convergent method called GCMMA (globally convergent method of moving asymptotes) was developed, yielding good results in practice. It is only applicable for inequality constraints, i.e., $m_e = 0$. Proceeding from a feasible starting point $x^{(0)} \in F$, the algorithm creates a sequence of feasible iteration points, i.e., $m_c = 0$, $m_f \neq 0$. Svanberg [82, 83] proposed additional inner iterations ensuring

$$f\left(z^{(k,p)}\right) \leq f^{(k,p)}\left(z^{(k,p)}\right) \tag{1.3}$$

$$e_j\left(z^{(k,p)}\right) \leq e_j^{(k,p)}\left(z^{(k,p)}\right), \quad j = 1, \dots, m_f \tag{1.4}$$

where $f^{(k,p)}(x)$ is the strictly convex approximation of $f(x)$ and $e_j^{(k,p)}(x)$ is the convex approximation of $e_j(x)$, $j = 1, \dots, m_f$, in the outer iteration k and the inner iteration p. Moreover, $z^{(k,p)} \in \mathbb{R}^n$ is the optimal solution of the corresponding subproblem. If (1.3) or (1.4) is violated for at least one constraint or the objective function, a more conservative subproblem is formulated based on the MMA approximation. It can be shown that the inner iteration loop terminates within a finite number of iterations. Note that the functions have to be evaluated at infeasible points.

Many optimization methods, for example SQP, apply trust region techniques to show global convergence. Ni [60] introduced a new version of MMA, where the convex subproblems are additionally restricted by a trust region. In contrast to MMA and SCP, it is only applicable for box constraints, $\underline{x} \leq x \leq \overline{x}$ while equality and inequality constraints cannot be handled, i.e., $m_e = m_c = m_f = 0$.

Ertel [19] combined the method of moving asymptotes with the filter approach proposed by Fletcher and Leyffer [26]. 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 [27].

Stingl, Kočvara and Leugering [79] proposed a generalization of SCP for semidefinite programs called PENSCP. They consider the following problem formulation

$$\begin{aligned} \min_{Z} \quad & f(Z) && Z \in \mathbb{S}^n \\ \text{s.t.} \quad & c_j(Z) \leq 0, && j = 1, \ldots, m_c \\ & Z - \underline{Z} \succeq 0 && \\ & \overline{Z} - Z \succeq 0 && \end{aligned} \tag{1.5}$$

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 in the objective function. The resulting semidefinite subproblem can be solved efficiently due to its specific structure by appropriate solvers, e.g., PENNON, see Kočvara and Stingl [48]. Global convergence of the resulting algorithm can be shown, see Stingl, Kočvara and Leugering [79].

As SCP achieves good results for topology optimization problems, it is to be applied to free material optimization. Some of the problem specific functions of FMO are only defined within the feasible region given by feasibility constraints $e_j(x)$, $j = 1, \ldots, m_f$, which are nonlinear reformulations of $E_i \succeq 0$, $i = 1, \ldots, m$. The SCP method proposed by Zillober [97] is not ensuring feasibility of the iterates during the solution process. Therefore, it is extended such that strict feasibility subject to a special set of constraints is guaranteed in each iteration step. The convex feasibility constraints are passed to the subproblem directly while the objective function as well as the remaining constraints are approximated based on the MMA approximation scheme. An active set strategy is applied for the remaining constraints only, to ensure feasibility whenever functions or gradients are to be evaluated. In addition, constraints that are expected to be active in the optimal solution are always included in the active set. The subproblems possess an unique solution. They can be solved efficiently exploiting the sparse structure of the gradients and Hessian. 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.

Feasible optimization methods compute a sequence of feasible iterates, i.e., only feasibility constraints $e_j(x)$, $j = 1, \ldots, m_f$, are considered, i.e., $m_e = m_c = 0$. In the literature, several feasible optimization methods can be found. In many real world applications, the optimization problems are of high dimension and the function and gradient evaluations might be time consuming. Using feasible optimization techniques, the optimization process can be aborted at each iterate yielding a feasible, although not optimal, solution. The most important feasible optimization methods 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. This yields to the so-called barrier function which is to be minimized, e.g., by Newton's method. Typically the barrier function is only defined on the feasible region and tends to infinity at the boundary. A popular barrier function is the logarithmic barrier function

$$f\left(x\right)+\mu\sum_{i=1}^{m_f}\ln\left(-e_j\left(x\right)\right), \tag{1.6}$$

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 [43].

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. Figure 1.1 shows the projection of an infeasible point on the boundary of the feasible set. The resulting projected search direction is given by the red line.

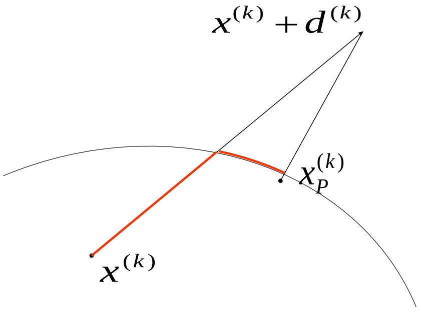


Fig. 1.1: Projection method

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 [71, 72] and by Polak [67]. Projection methods are often combined with other efficient nonlinear optimization methods to compute the descent direction $d^{(k)}$. Jian, Zhang and Xue [46] developed a feasible SQP method in combination with projection methods. The quadratic subproblem is solved to obtain a descent direction. Moreover, the iterate is projected on the boundary and a line search is performed.

Feasible direction methods compute a feasible direction $d^{(k)}$, which ensures the existence of $\theta^{(k)} \in \mathbb{R}^+$, such that $x^{(k)} + \sigma^{(k)} d^{(k)}$ is feasible for all $\sigma^{(k)} \leq \theta^{(k)}$, where $\sigma^{(k)} \in \mathbb{R}$ is the stepsize. Many different feasible direction methods can be found in the literature. The first feasible direction algorithm is the P1 algorithm developed by Zoutendijk in 1960, see [106]. In each iteration step, an improving feasible search direction is determined and an extended line search is performed, yielding a sufficient descent in the objective function and satisfying the constraints $e_j(x) \leq 0,\ j = 1, \ldots, m_f$. Proceeding from a feasible starting point $x^{(0)}$, in each iteration k a search direction $d^{(k)}$ is computed, which is a descent direction with respect to the objective function and the ε active constraints $\mathbb{J}_\varepsilon^{(k)} := \left\{ j = 1, \ldots, m_f \middle|\ e_j\left(x^{(k)}\right) \geq -\varepsilon \right\},\ \varepsilon \in \mathbb{R}^+$, i.e.,

$$\begin{array}{rcl} \nabla f\left(x^{(k)}\right)^T d^{(k)} & \leq & 0 \\ \nabla e_j\left(x^{(k)}\right)^T d^{(k)} & \leq & 0, \quad j \in \mathbb{J}_\varepsilon^{(k)}. \end{array} \tag{1.7}$$

Iteratively, a linear subproblem, is formulated, which maximizes the minimal descent, see Großmann and Kleinmichel [34] and Ishutkin and Großmann [42]. We denote the solution of

$$\begin{array}{lll} \min\limits_{\delta,d} & \delta & d \in \mathbb{R}^n,\ \delta \in \mathbb{R} \\ \text{s.t.} & \nabla f\left(x^{(k)}\right)^T d \leq \delta & \\ & \nabla e_j\left(x^{(k)}\right)^T d \leq \delta, & j \in \mathbb{J}_\varepsilon^{(k)} \\ & \|d\|_\infty \leq 1 & \end{array} \tag{1.8}$$

by $\left(d^{(k)}, \delta^{(k)}\right)$. If ε is adapted adequately, it can be shown that $\delta^{(k)} \leq 0$, for all $k = 0, 1, \ldots$. The size of ε is very important for the convergence of the algorithm. If ε becomes too small, a typical oscillating behavior can be observed. An enhancement of Zoutendijk's P1 algorithm is his P2 algorithm, developed in 1961, see Zoutendijk [106], which is more robust than the first method, as ε need not be adapted. We get $\left(d^{(k)}, \delta^{(k)}\right)$ by solving

$$\begin{array}{lll} \min\limits_{\delta,d} & \delta & d \in \mathbb{R}^n,\ \delta \in \mathbb{R} \\ \text{s.t.} & \nabla f\left(x^{(k)}\right)^T d \leq \delta & \\ & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq \delta, & j \in \mathbb{J}_\varepsilon^{(k)} \\ & \|d\|_\infty \leq 1 & \end{array} \tag{1.9}$$

with $\mathbb{J}_\varepsilon^{(k)} := \left\{ j = 1, \ldots, m_f \middle| \; e_j\left(x^{(k)}\right) \geq -\varepsilon \right\}$. A convergence proof for both methods can be given for convex constraints $e_j(x)$, $j = 1, \ldots, m_f$, see Bertsekas [10].

For Zoutendijk's P1 and P2 methods only linear convergence can be shown. Therefore, the subproblems are extended such that second order information is included. One possibility is to compute a descent direction $d_0^{(k)} \in \mathbb{R}^n$ by solving a quadratic subproblem (QP), i.e., a quadratic objective function and linear constraints, according to SQP methods, see Schittkowski and Yuan [75],

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2} d^T H\left(x^{(k)}, y^{(k)}\right) d + \nabla f\left(x^{(k)}\right)^T d & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq 0, & j = 1, \ldots, m_f \end{array} \tag{1.10}$$

where $H\left(x^{(k)}, y^{(k)}\right) \in \mathbb{R}^{n \times n}$ is the Hessian of the Lagrangian function with respect to x or an appropriate approximation. Moreover, $y^{(k)} \in \mathbb{R}^{m_f}$ is the dual variable. The resulting search direction $d_0^{(k)}$ may not be feasible, as for active constraints $\nabla e_j\left(x^{(k)}\right)^T d_0^{(k)} = 0$ is allowed, which yields to a search direction tangential to the feasible region, see Panier and Tits [65]. Therefore, 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, several 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 [66], Herskovits [35, 36], Bakhtiari and Tits [3] and Zhu [95]. The globally and locally superlinear convergent algorithm FAIPA belongs to the latest algorithms and is briefly presented, see Herskovits, Aroztegui, Goulart and Dubeux [37]. In each iteration k 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. We proceed from a feasible starting point $x^{(0)} \in \mathbb{R}^n$. The first linear system is derived from Newton's

method applied to the KKT conditions computing $\left(d_0^{(k)}, y_0^{(k)}\right)$

$$\begin{bmatrix} H\left(x^{(k)}, y^{(k)}\right) & \nabla e\left(x^{(k)}\right) \\ Y^{(k)} \nabla e\left(x^{(k)}\right)^T & E\left(x^{(k)}\right) \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} = - \begin{bmatrix} \nabla f\left(x^{(k)}\right) \\ 0 \end{bmatrix} \tag{1.11}$$

where $E\left(x^{(k)}\right) := \operatorname{diag}\left(e_1\left(x^{(k)}\right), \ldots, e_{m_f}\left(x^{(k)}\right)\right)$, and $y_i^{(k)}$, $i = 1, \ldots, m_f$, are the Lagrangian multipliers forming $Y^{(k)} := \operatorname{diag}\left(y_1^{(k)}, \ldots, y_{m_f}^{(k)}\right)$. The Hessian of the Lagrangian function or an appropriate approximation is denoted by $H\left(x^{(k)}, y^{(k)}\right) \in \mathbb{R}^{n \times n}$ and $\nabla e\left(x^{(k)}\right) := \left[\nabla e_1\left(x^{(k)}\right), \ldots, \nabla e_{m_f}\left(x^{(k)}\right)\right]$.

The resulting search direction $d_0^{(k)} \in \mathbb{R}^n$ is not necessarily a feasible direction. The right hand side is to be perturbed, to ensure that $d_0^{(k)}$ does not become tangent to the feasible region. The solution of

$$\begin{bmatrix} H\left(x^{(k)}, y^{(k)}\right) & \nabla e\left(x^{(k)}\right) \\ Y^{(k)} \nabla e\left(x^{(k)}\right)^T & E\left(x^{(k)}\right) \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} = - \begin{bmatrix} 0 \\ y^{(k)} \end{bmatrix} \tag{1.12}$$

is denoted by $\left(d_1^{(k)}, y_1^{(k)}\right)$. The resulting direction $d_1^{(k)} \in \mathbb{R}^n$ improves feasibility. With help of $d_0^{(k)}$ and $d_1^{(k)}$ the feasible descent direction $d^{(k)} \in \mathbb{R}^n$ can be computed by $d^{(k)} := d_0^{(k)} + t^{(k)} d_1^{(k)}$, where $t^{(k)} \in \mathbb{R}^+$ ensures a sufficient descent in the objective function, see Herskovits and Santos [39]. To ensure superlinear convergence near a stationary point, an additional direction $\widetilde{d}^{(k)} \in \mathbb{R}^n$ and the corresponding multipliers $\widetilde{y}^{(k)} \in \mathbb{R}^{m_f}$ have to be computed by solving

$$\begin{bmatrix} H\left(x^{(k)}, y^{(k)}\right) & \nabla e\left(x^{(k)}\right) \\ Y^{(k)} \nabla e\left(x^{(k)}\right)^T & E\left(x^{(k)}\right) \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} = - \begin{bmatrix} 0 \\ Y^{(k)} \widetilde{w}^{(k)} \end{bmatrix} \tag{1.13}$$

where the feasibility factor $\widetilde{w}^{(k)} \in \mathbb{R}^{m_f}$ estimates the curvature of the constraints by approximating their second order derivatives analogously to Taylor. A line search is performed along the search arc given by $x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$, such that the following conditions hold

$$f\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) < f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)} \tag{1.14}$$

$$e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq 0, \quad j = 1, \ldots, m_f \tag{1.15}$$

with $r \in (0,1)$. However, all these methods cannot prevent function evaluations at infeasible points, as $x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$ might be infeasible for $\sigma^{(k)} = 1$, as $\theta^{(k)} < 1$ is possible.

In the following chapter, basic theory of nonlinear optimization is reviewed. Optimality criteria and convergence rates are presented.

In Chapter 3 a review of the state-of-the-art feasible direction sequential quadratic programming methods is given. The algorithms are formulated and convergence rates are given.

In Chapter 4 sequential convex programming methods are discussed. Based on the method of moving asymptotes as proposed by Svanberg [80] in Section 4.1, the approximation schemes are presented. In Section 4.2 the SCP algorithm developed by Zillober [98], which is based on a line search procedure and the corresponding augmented Lagrangian merit function, is summarized. Moreover, the globally convergent MMA algorithm is presented in Section 4.3, where inner and outer iteration cycles are combined to ensure global convergence.

The strictly feasible sequential convex programming method (SCPF) is introduced in Chapter 5. The SCP algorithm is extended and adapted such that feasibility with respect to a subset of constraints is guaranteed in each iteration. The resulting strictly feasible sequential convex programming method is presented and formulated in Section 5.1. A global convergence proof is given in Section 5.2.

The main applications arise in free material optimization where elasticity tensors must be positive definite in order to evaluate a valid global stiffness matrix. The problem formulation is derived in Chapter 6. Reformulations to replace semidefinite constraints by nonlinear ones are proposed. First and second order derivatives are given.

The implementation and additional features speeding up the algorithm are presented in Chapter 7. Applications arising in FMO and oil industry are presented. Feasibility constraints are identified and the corresponding MMA approximations are given. Numerical results for a test set are shown.

The Appendix contains the program documentation as well as a detailed description of the calling parameters and the reverse communication.

2. BASIC THEORY OF NONLINEAR OPTIMIZATION

Within this chapter we will briefly review basic theory of nonlinear programming, that is necessary for the subsequent chapters. The main topic of this thesis is to combine constraints that have to be satisfied in each iteration step, called feasibility constraints, and constraints that might be infeasible until the optimal solution is found, referred as regular constraints. Most nonlinear optimization methods do not guarantee feasibility during the optimization process, while some specific algorithms ensure feasibility in the main iterates, called feasible optimization methods. In the subsequent chapters we will consider both, feasible optimization methods and others. Therefore, we proceed from the nonlinear optimization problem where the constraints are divided into feasibility constraints $e_j(x)$, $j = 1, \ldots, m_f$, and regular constraints $c_j(x)$, $j = 1, \ldots, m_c$.

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j(x) = 0, & j = 1, \ldots, m_e \\ & c_j(x) \leq 0, & j = m_e + 1, \ldots, m_c \\ & e_j(x) \leq 0, & j = 1, \ldots, m_f \end{array} \tag{2.1}$$

The constraints $e_j(x)$, $j = 1, \ldots, m_f$, which have to be feasible in each iteration step are defined on $\mathbb{R}^n$ and are at least twice continuously differentiable. The objective function $f(x)$ and the constraints $c_j(x)$, $j = 1, \ldots, m_c$, are defined on the feasible region of the constraints $e_j(x)$, $j = 1, \ldots, m_f$. i.e., they may be violated within the optimization process. In addition, they are at least twice continuously differentiable on the subset given by the feasibility constraints. Note that in some chapters only feasibility constraints are present, i.e., $m_e = m_c = 0$, while they vanish in other chapters, i.e., $m_f = 0$. Moreover, the feasible region is given by Definition 1.

Definition 1. *The feasible region of (2.1) is defined by the set*

$$\begin{array}{rll} \mathbb{F} & := & \{x \in \mathbb{R}^n \mid c_j(x) = 0,\ j = 1, \ldots, m_e\} \\ & & \cap \{x \in \mathbb{R}^n \mid c_j(x) \leq 0,\ j = m_e + 1, \ldots, m_c\} \\ & & \cap \{x \in \mathbb{R}^n \mid e_j(x) \leq 0,\ j = 1, \ldots, m_f\} \end{array} \tag{2.2}$$

In general, the inequality constraints are divided into two groups, active and inactive constraints. We define

Definition 2. *The constraint c_j, $1 \leq j \leq m_c$, or e_j, $1 \leq j \leq m_f$ respectively, is active at $x \in \mathbb{F}$, if*

$$c_j(x) \;=\; 0 \tag{2.3}$$

or

$$e_j(x) \;=\; 0 \tag{2.4}$$

holds. Moreover, the active set at $x \in \mathbb{F}$ is defined by

$$\begin{aligned}\mathbb{J}(x) \;:=\; & \{j \in \{1,\ldots,m_c\} \mid c_j(x) = 0\} \\ & \cup \{j \in \{1,\ldots,m_f\} \mid e_j(x) = 0\}\,.\end{aligned} \tag{2.5}$$

Nonlinear programming deals with finding a minimum with respect to the feasible region $\mathbb{F}$, given in Definition 1. We distinguish local and global minima.

Definition 3. *$x^\star \in \mathbb{F}$ is a local minimum, if there exist a neighborhood $U(x^\star)$, such that*

$$f(x^\star) \;\leq\; f(x) \tag{2.6}$$

holds for all $x \in U(x^\star) \cap \mathbb{F}$.

$x^\star \in \mathbb{F}$ is a global minimum, if

$$f(x^\star) \;\leq\; f(x) \tag{2.7}$$

holds for all $x \in \mathbb{F}$.

The optimality criteria are based on the Lagrangian function, which combines the value of the objective function and the constraints via the Lagrangian multipliers.

Definition 4. *The Lagrangian function corresponding to optimization problem (2.1) is defined by*

$$L(x,y) \;:=\; f(x) + \sum_{j=1}^{m_c} (y_c)_j\, c_j(x) + \sum_{j=1}^{m_f} (y_e)_j\, e_j(x) \tag{2.8}$$

where

$$y := \begin{pmatrix} y_c \\ y_e \end{pmatrix} \in \mathbb{R}^{m_c+m_f} \tag{2.9}$$

are Lagrangian multipliers or dual variables, while $x \in \mathbb{R}^n$ are the primal variables.

Some nonlinear programming algorithms use second order information, motivated by Newton's method. This means that they take the Hessian of the Lagrangian (2.8) into account. In the following, the Hessian of the Lagrangian function with respect to x is denoted by

$$H(x,y) \quad := \quad \nabla^2_{xx} L(x,y)\,. \tag{2.10}$$

To save computational effort and to ensure positive definiteness, $H(x,y)$ can be approximated in a certain sense by appropriate updating schemes, e.g., the widely used BFGS update, see Schittkowski and Yuan [75]. In the following we denote the Hessian or its approximation at iteration k by

$$H^{(k)} \quad := \quad H\left(x^{(k)}, y^{(k)}\right). \tag{2.11}$$

Before we can formulate optimality criteria, we consider the linear independence constraint qualification (LICQ).

Definition 5. *The linear independence constraint qualification (LICQ) is satisfied at a feasible solution $x \in \mathbb{F}$, if the gradients of the active constraints are linearly independent at x.*

A constraint qualification is a necessary requirement for the following first order necessary optimality condition.

Lemma 2.1. *Let $x^\star \in \mathbb{R}^n$ be a local minima of (2.1) and let the LICQ hold. Then there exists a $y^\star \in \mathbb{R}^{m_c+m_f}$ such that the following Karush-Kuhn-Tucker (KKT) conditions hold*

$$\begin{aligned}
\nabla_x L(x^\star, y^\star) &= 0 && && (2.12)\\
c_j(x^\star) &= 0, && j = 1, \ldots, m_e && (2.13)\\
c_j(x^\star) &\leq 0, && j = m_e + 1, \ldots, m_c && (2.14)\\
e_j(x^\star) &\leq 0, && j = 1, \ldots, m_f && (2.15)\\
(y_c^\star)_j &\geq 0, && j = m_e + 1, \ldots, m_c && (2.16)\\
(y_e^\star)_j &\geq 0, && j = 1, \ldots, m_f && (2.17)\\
(y_c^\star)_j\, c_j(x^\star) &= 0, && j = m_e + 1, \ldots, m_c && (2.18)\\
(y_e^\star)_j\, e_j(x^\star) &= 0, && j = 1, \ldots, m_f && (2.19)
\end{aligned}$$

Definition 6. *If $(x^\star, y^\star)$ satisfies the KKT conditions (2.12) - (2.19), it is called stationary or KKT point.*

It can be shown that the Lagrangian multipliers are unique, if the LICQ holds, see Schittkowski and Yuan [75]. Otherwise, this might lead to numerical instabilities within the optimization process.

In general, optimization problem (2.1) is solved iteratively by constructing a sequence of subproblems. At each iterate $x^{(k)} \in \mathbb{R}^n$, a subproblem is formulated which can be

solved efficiently by appropriate solution methods. Typically, it possesses a special structure that can be exploited. The next iterate is determined by the solution of the current subproblem. The resulting iteration sequence converges towards a stationary point under certain assumptions.
To ensure global convergence, i.e., a stationary point is found independently of the starting point, the algorithms are equipped with globalization techniques. In general, we distinguish between

1. trust region methods, see Vardi [89], Byrd, Schnabel and Shultz [15] and Omojokun [62],
2. filter methods, see Fletcher and Leyffer [26],
3. and line search methods, see Armijo [2], Ortega and Rheinboldt [63].

This thesis focuses on line search methods only. We denote the primal solution of the subproblem in iteration k by $z^{(k)} \in \mathbb{R}^n$ and the dual solution by $v^{(k)} \in \mathbb{R}^{m_c+m_f}$. In each iteration, a sufficient descent with respect to a suitable merit function $\Phi \begin{pmatrix} x \\ y \end{pmatrix}$ is required, which combines objective function and constraints in an adequate way. If $\left(z^{(k)}, v^{(k)}\right)$ yields no sufficient descent, the so-called Armijo line search algorithm can be applied, see Armijo [2], Ortega and Rheinboldt [63]. By successive bisection of $\sigma^{(k)} \in \mathbb{R}^+$, starting from $\sigma^{(k)} = 1$, it finds the first $\sigma^{(k)}$ satisfying

$$\Phi \left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k)} d^{(k)} \right) \leq \Phi \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + r\sigma^{(k)} \nabla \Phi \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}, \tag{2.20}$$

where $r \in (0,1)$ is constant and where $d^{(k)} := \begin{pmatrix} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{pmatrix}$, $d^{(k)} \in \mathbb{R}^{n+m_f+m_c}$ is called search direction. We define

Definition 7. *A primal search direction $\Delta x^{(k)} \in \mathbb{R}^n$, $\Delta x^{(k)} := z^{(k)} - x^{(k)}$ is a descent direction of a real continuously differentiable function $f(x)$ at $x^{(k)} \in \mathbb{R}^n$, if*

$$\nabla f \left(x^{(k)} \right)^T \Delta x^{(k)} < 0 \tag{2.21}$$

holds.

The efficiency of an algorithm is given by the convergence rate, which measures how fast the iteration sequence converges towards a stationary point $x^\star$ in the neighborhood of $x^\star$. We define three different convergence rates.

Definition 8. *An iteration sequence $\left\{x^{(k)}\right\}$ is said to converge linearly towards a stationary point $x^\star \in \mathbb{R}^n$, if there exists a $c_1 \in]0,1[$ with*

$$\left\| x^{(k+1)} - x^\star \right\| \quad \leq \quad c_1 \left\| x^{(k)} - x^\star \right\| \tag{2.22}$$

for all k sufficiently large.

An iteration sequence $\left\{x^{(k)}\right\}$ *is said to converge superlinearly towards a stationary point* $x^\star \in \mathbb{R}^n$, *if there exist a sequence* $\left\{c^{(k)}\right\}$ *with* $c^{(k)} \in \mathbb{R}^+$ *converging to zero with*

$$\left\|x^{(k+1)} - x^\star\right\| \quad \leq \quad c^{(k)} \left\|x^{(k)} - x^\star\right\| \tag{2.23}$$

for all k *sufficiently large.*

An iteration sequence $\left\{x^{(k)}\right\}$ *is said to converge quadratically towards a stationary point* $x^\star \in \mathbb{R}^n$, *if there exists a* $c_2 \in \mathbb{R}^+$ *with*

$$\left\|x^{(k+1)} - x^\star\right\| \quad \leq \quad c_2 \left\|x^{(k)} - x^\star\right\|^2 \tag{2.24}$$

for all k *sufficiently large.*

Under some assumptions, sequential quadratic programming (SQP) methods, converge with local superlinear convergence rate towards a KKT point specified in Definition 6, see Schittkowski and Yuan [75]. A necessary requirement is that stepsize $\sigma^{(k)} = 1$ is accepted in the neighborhood of a solution. The globalization techniques often prevent the acceptance of stepsize one, which leads to slow convergence. This behavior is called the Maratos effect, see Maratos [57]. In the literature, several techniques can be found to prevent the Maratos effect, e.g.,

1. non-monotone techniques, see Ulbrich and Ulbrich [86] and Gould and Toint [33],
2. watch-dog techniques, see Chamberlain et al. [17],
3. smooth exact penalty functions or the augmented Lagrangian function as merit function, see Schittkowski [73], Powell and Yuan [68, 69] and Ulbrich [87],
4. second order correction techniques, see Fletcher [25], Mayne and Polak [59] and Fukushima [30].

3. FEASIBLE SEQUENTIAL QUADRATIC OPTIMIZATION METHODS

In this chapter, a brief review of existing feasible sequential quadratic optimization techniques (FSQP) is given. The methods guarantee that each main iterate is feasible, i.e., $\left\{x^{(k)}\right\} \in \mathbb{F}$, but function evaluations at infeasible points are nevertheless necessary. We proceed from the following problem formulation:

$$\begin{array}{lll} \min\limits_{x} & f\left(x\right) & x \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x\right) \leq 0, & j = 1, \ldots, m_f \end{array} \tag{3.1}$$

where $f\left(x\right)$ and $e_j\left(x\right),\ j = 1, \ldots, m_f$, are at least twice continuously differentiable. We require that the nonlinear constraints $e_j\left(x\right),\ j = 1, \ldots, m_f$, are satisfied in each iteration step after a possible restoration phase. The feasible region $\mathbb{F}$ is specified in Definition 1. Note that in this chapter $d^{(k)} \in \mathbb{R}^n$ denotes the primal search direction. In the literature, several different classes of feasible algorithms can be found. In the sequel, we focus on feasible direction sequential quadratic programming methods. First, we define a feasible direction according to Herskovits and Carvalho [38].

Definition 9. *A search direction $d \in \mathbb{R}^n$ is a feasible direction at $x \in \mathbb{F}$, if for some $\theta \in \mathbb{R}^+$*

$$x + \sigma d \quad \in \quad \mathbb{F} \tag{3.2}$$

holds for all $\sigma \in [0, \theta]$.

We will consider different feasible direction approaches, that are based on the sequential quadratic programming (SQP) method, see Schittkowski and Yuan [75] for a review of general SQP methods. SQP algorithms converge towards a stationary point, see Definition 6, by solving a sequence of quadratic programming (QP) subproblems of the form

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2} d^T H^{(k)} d + \nabla f\left(x^{(k)}\right)^T d & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq 0, & j = 1, \ldots, m_f \end{array} \tag{3.3}$$

where $H^{(k)}$ is the Hessian of the Lagrangian or an appropriate approximation, see (2.10) and (2.11). Let the solution of (3.3) in iteration k be $\left(d_0^{(k)}, y_0^{(k)}\right)$ where the corresponding dual variable is denoted by $y_0^{(k)} \in \mathbb{R}^{m_f}$. The next iterate $x^{(k+1)}$ is computed by

$$x^{(k+1)} \;:=\; x^{(k)} + \sigma^{(k)} d_0^{(k)}, \tag{3.4}$$

where the stepsize $\sigma^{(k)} \in \mathbb{R}^+$ is determined by a line search procedure, such that a descent in the corresponding merit function is obtained.

In general, feasible SQP methods compute a search direction $d_0^{(k)}$ analogously to SQP methods, which needs to be adapted as it may be infeasible. Especially, if a constraint $e_j\left(x\right),\ j = 1, \ldots, m_f$, is active, i.e., $e_j\left(x^{(k)}\right) = 0$, the corresponding search direction can be asymptotic with respect to the feasible region. Feasible SQP methods modify $d_0^{(k)}$, such that a feasible descent direction is obtained. An extended line search procedure ensures that the next iterate is feasible and yields a descent with respect to the objective function. The different methods are based on either solving QPs or least squares problems.

3.1 Modified Method of Topkis and Veinott

In 1967, Topkis and Veinott [85] formulated a feasible direction algorithm motivated by Zoutendijk's P1 and P2 [106] methods. Topkis and Veinott propose to solve the following linear subproblems iteratively,

$$\begin{array}{lll} \min\limits_{\delta,d} & \delta & d \in \mathbb{R}^n,\ \delta \in \mathbb{R} \\ \text{s.t.} & \nabla f\left(x^{(k)}\right)^T d \leq \delta & \\ & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq \delta, & j \in \mathbb{J}^{(k)} \\ & -1 \leq d_i \leq 1, & i = 1, \ldots, n \end{array} \tag{3.5}$$

with

$$\mathbb{J}^{(k)} \;:=\; \left\{ j = 1, \ldots, m_f \,\middle|\, e_j\left(x^{(k)}\right) = 0 \right\}. \tag{3.6}$$

Let the solution of (3.5) in iteration k be $\left(\delta^{(k)}, d^{(k)}\right)$. The algorithm is also equipped with an extended line search strategy that ensures both a descent direction with respect to the objective function $f\left(x\right)$ and feasibility with respect to the inequality constraints $e_j\left(x\right),\ j = 1, \ldots, m_f$. The solution of (3.5) is not unique and oscillating might slow down convergence, see Birge, Qi and Wei [11].

To speed up the observed slow convergence, Birge, Qi and Wei [11] extended the method of Topkis and Veinott by quadratic subproblems and quasi-Newton approximations. The corresponding subproblem is solved iteratively

$$\begin{array}{lll} \min\limits_{\delta,d} & \delta + \frac{1}{2} d^T H^{(k)} d & d \in \mathbb{R}^n,\ \delta \in \mathbb{R} \\ \text{s.t.} & \nabla f\left(x^{(k)}\right)^T d \leq t_0^{(k)} \delta & \\ & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq t_j^{(k)} \delta, & j \in \mathbb{J}^{(k)} \end{array} \tag{3.7}$$

with $t_j^{(k)} > 0,\ j = 0, \ldots, m_f$. $H^{(k)} \in \mathbb{R}^{n \times n}$ is a symmetric positive definite approximation of the Hessian of the Lagrangian function, see (2.10) and (2.11). We denote the solution of (3.7) by $\left(\delta^{(k)}, d^{(k)}\right)$. If the matrix $H^{(k)}$ is positive definite, i.e., $H^{(k)} \succ 0$, problem (3.7) is convex, see Geiger and Kanzow [32]. The unique solution can be computed efficiently by an appropriate QP solver. Moreover, it can be shown that $\delta^{(k)} \leq 0$, see Birge, Qi and Wei [11].

An extended line search procedure is applied to ensure a descent in the objective function $f(x)$ and feasibility with respect to the constraints $e_j(x),\ j = 1, \ldots, m_f$. Motivated by the high cost of function evaluations in many applications, the corresponding implementation forces to accept stepsize one. Whenever $\sigma^{(k)} \neq 1$, the parameter $t_j^{(k+1)},\ j = 0, \ldots, m_f$, is enlarged, which emphasizes feasibility, see Lawrence and Tits [54]. The corresponding update rules are given as follows

$$t_0^{(k+1)} := \begin{cases} t_0^{(k)}, & \text{if } f\left(x^{(k)} + d^{(k)}\right) \leq f\left(x^{(k)}\right) + r\nabla f\left(x^{(k)}\right)^T d^{(k)} \\ 2t_0^{(k)}, & \text{otherwise} \end{cases} \tag{3.8}$$

$$t_j^{(k+1)} := \begin{cases} t_j^{(k)}, & \text{if } e_j\left(x^{(k)} + d^{(k)}\right) \leq 0,\ j = 1, \ldots, m_f \\ 2t_j^{(k)}, & \text{otherwise} \end{cases} \tag{3.9}$$

with $r \in (0, 1)$.

The algorithm according to Birge, Qi and Wei [11] can be written as follows:

Algorithm 1. Modified method of Topkis and Veinott [11]

```
Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$, parameters $r \in (0,1)$, $t_j^{(0)} > 0$,
        $j = 0, \ldots, m_f$ and $H^{(0)} \in \mathbb{R}^{(n \times n)}$ symmetric and positive definite. Let $k := 0$.
Step 1: Formulate and solve (3.7) to obtain $\left(\delta^{(k)}, d^{(k)}\right)$.
Step 2: If $\left(\delta^{(k)}, d^{(k)}\right) = 0$ then STOP.
Step 3: Get stepsize $\sigma^{(k)} := \max\left\{0.5^l, l = 0, 1, 2, \ldots\right\}$ such that
        $f\left(x^{(k)} + \sigma^{(k)} d^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ and
        $e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)}\right) \leq 0,\ j = 1, \ldots, m_f$.
Step 4: Determine $t_j^{(k+1)} > 0,\ j = 0, \ldots, m_f$, according to (3.8) and (3.9).
Step 5: Compute a symmetric and positive definite matrix $H^{(k+1)} \in \mathbb{R}^{(n \times n)}$ by a
        quasi-Newton formula.
Step 6: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)}$ and $k := k + 1$, goto Step 1.
```

It can be guaranteed that $\sigma^{(k)}$ exists, such that the requirements in Step 3 are satisfied. Under certain conditions global convergence can be shown, see Birge et al. [11].

3.2 A feasible SQP method by Panier and Tits

In 1987, Panier and Tits [64] proposed a feasible direction algorithm, which is under certain conditions locally superlinear convergent. To ensure global convergence a first order feasible search direction will be used far away from the solution, see Algorithm 2 Step 5. Otherwise, two QPs and one least squares problem have to be solved in each iteration. Beginning with the solution $\left(d_0^{(k)}, y_0^{(k)}\right)$ of (3.3), the right hand side of the inequality constraints is modified to guarantee a search direction aligning to the interior of the feasible region and thus to ensure feasibility. The value of this perturbation depends on the solution $d_0^{(k)} \in \mathbb{R}^n$ of (3.3). We obtain the following second QP in iteration k,

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2} d^T H^{(k)} d + \nabla f\left(x^{(k)}\right)^T d & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq -\left\|d_0^{(k)}\right\|^{\beta_1}, & j = 1, \ldots, m_f \end{array} \tag{3.10}$$

with $\beta_1 > 2$. The resulting search direction $d^{(k)} \in \mathbb{R}^n$ is a feasible direction, see Panier and Tits [64]. Moreover, in the neighborhood of the solution of (3.1), $d^{(k)}$ is guaranteed to be a descent direction with respect to $f(x)$. To prevent the Maratos effect, it is necessary to bend the search direction and perform a line search along the resulting search arc, see Mayne and Polak [58]. Therefore, we introduce an additional search direction $\widetilde{d}^{(k)} \in \mathbb{R}^n$ which tends to zero, if $d^{(k)}$ is small. To compute $\widetilde{d}^{(k)}$ the following linear least squares problem is to be solved,

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2} \|d\|^2 & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)} + d^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d = -\left\|d_0^{(k)}\right\|^{\beta_2}, & j \in \mathbb{J}^{(k)} \end{array} \tag{3.11}$$

with $\beta_2 \in (2, 3)$ and

$$\mathbb{J}^{(k)} := \left\{ j = 1, \ldots, m_f \;\middle|\; e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d^{(k)} = -\left\|d_0^{(k)}\right\|^{\beta_1} \right\}. \tag{3.12}$$

Note that $x^{(k)} + d^{(k)}$ might be infeasible. In this case, the constraints have to be evaluated at infeasible iterates to formulate (3.11).

Solving the QPs (3.3), (3.10) and (3.11) together with a line search along the search arc $x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$ does not lead to global convergence, see Lawrence and Tits [54]. Therefore, a feasible first order search direction $d^{(k)}$ is computed far away from the solution, i.e., if one of the following conditions holds

$$\left\| d_0^{(k)} \right\| > M \tag{3.13}$$

$$\left\| H^{(k)} d_0^{(k)} \right\| > \left\| d_0^{(k)} \right\|^{1/2} \tag{3.14}$$

$$\left\| d^{(k)} \right\| > M \tag{3.15}$$

$$\nabla f \left(x^{(k)} \right)^T d^{(k)} > \min \left\{ - \left\| d_0^{(k)} \right\|^{\beta_2}, - \left\| d^{(k)} \right\|^{\beta_2} \right\} \tag{3.16}$$

$$\left\| \widetilde{d}^{(k)} \right\| > \left\| d^{(k)} \right\| \tag{3.17}$$

with $M \in \mathbb{R}^+$. The feasible first order search direction $d^{(k)}$ satisfies

$$\nabla f \left(x^{(k)} \right)^T d^{(k)} \leq -\delta \tag{3.18}$$

$$\nabla e_j \left(x^{(k)} \right)^T d^{(k)} \leq -\delta, \quad j = 1, \ldots, m_f \tag{3.19}$$

with $\delta \in \mathbb{R}^+$. One possibility is to solve (3.5) to obtain $d^{(k)}$ and set $\widetilde{d}^{(k)} = 0$. This may lead to slow convergence, if the starting point is badly chosen, see Lawrence and Tits [54].

Algorithm 2. Feasible direction SQP method by Panier and Tits [64]

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$, *parameters* $\beta_1 > 2, \beta_2 \in (2, 3)$, $\beta \in (0, 1), r \in (0, 1), M \in \mathbb{R}^+$, $H^{(0)} \succ 0$. *Let* $k := 0$.

Step 1: Solve (3.3) to obtain $d_0^{(k)}$. *If no optimal solution exists or* $\left\| d_0^{(k)} \right\| > M$ *or* $\left\| H^{(k)} d_0^{(k)} \right\| > \left\| d_0^{(k)} \right\|^{1/2}$, *goto Step 5.*

Step 2: If $d_0^{(k)} = 0$, *then STOP.*

Step 3: Solve (3.10) to obtain $d^{(k)}$. *If no optimal solution exists or* $\left\| d^{(k)} \right\| > M$ *or* $\nabla f \left(x^{(k)} \right)^T d^{(k)} > \min \left\{ - \left\| d_0^{(k)} \right\|^{\beta_2}, - \left\| d^{(k)} \right\|^{\beta_2} \right\}$, *goto Step 5.*

Step 4: Solve (3.11) to obtain $\widetilde{d}^{(k)}$. *If no optimal solution exists or* $\left\| \widetilde{d}^{(k)} \right\| > \left\| d^{(k)} \right\|$, *then goto Step 5.*
Else goto Step 6.

Step 5: Compute a feasible first order descent direction $d^{(k)}$ *satisfying (3.18) and (3.19), e.g., solve (3.5). Set* $\widetilde{d}^{(k)} = 0$.

Step 6: Compute stepsize $\sigma^{(k)}$, *i.e., the first value* $\sigma^{(k)}$ *in sequence* $\{1, \beta, \beta^2, \ldots\}$ *satisfying*
$f \left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)} \right)^2 \widetilde{d}^{(k)} \right) \leq f \left(x^{(k)} \right) + r \sigma^{(k)} \nabla f \left(x^{(k)} \right)^T d^{(k)}$ *and*
$e_j \left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)} \right)^2 \widetilde{d}^{(k)} \right) \leq 0, \quad j = 1, \ldots, m_f.$

Step 7: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$*, compute a symmetric and positive definite matrix* $H^{(k+1)} \in \mathbb{R}^{(n\times n)}$ *by a quasi-Newton formula,* $k := k+1$ *and goto Step 1.*

3.3 A feasible SQP method by Herskovits and Carvalho

Herskovits and Carvalho [38] proposed a feasible direction algorithm, which is under certain conditions locally superlinear but not globally convergent. In each iteration, two QPs have to be solved. Considering problem (3.3), the resulting descent direction $d_0^{(k)} \in \mathbb{R}^n$ might be infeasible depending on the curvature of the active constraints. To ensure that the search direction $d_0^{(k)}$ is also a feasible direction, a modified quadratic program is solved. Active constraints of the original QP (3.3) are included as equality constraints with a modified right hand side. The perturbation factor depends on the norm of the original search direction $d_0^{(k)}$. In addition, a line search is performed with respect to the Lagrangian function defined in (2.8). The solution $\left(d_0^{(k)}, y_0^{(k)}\right)$ of (3.3), defines the active set $\mathbb{J}^{(k)}$ in iteration k and its complement $\overline{\mathbb{J}}^{(k)}$

$$\begin{aligned} \mathbb{J}^{(k)} &:= \left\{ j = 1,\ldots,m_f \,\middle|\, \left(y_0^{(k)}\right)_j \neq 0 \right\}, && (3.20)\\ \overline{\mathbb{J}}^{(k)} &:= \left\{ j = 1,\ldots,m_f \,\middle|\, \left(y_0^{(k)}\right)_j = 0 \right\}. && (3.21) \end{aligned}$$

Whenever (3.3) is solved, a modified QP is formulated depending on the resulting search direction $d_0^{(k)}$ and a weighting factor $t^{(k)} \in \mathbb{R}^+$. We get $d^{(k)}$ by solving

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2} d^T H^{(k)} d + \nabla f\left(x^{(k)}\right)^T d & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d = -t^{(k)} \left\| d_0^{(k)} \right\|^2, & j \in \mathbb{J}^{(k)} \end{array} \quad (3.22)$$

The value of $t^{(k)}$ has to be chosen such that specific conditions for active and inactive constraints are satisfied. Considering the active constraints, we define

$$\begin{aligned} e_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) &:= \left[e_j\left(x^{(k)}\right)\right]^T_{\mathbb{J}^{(k)}} \in \mathbb{R}^{\left|\mathbb{J}^{(k)}\right|} && (3.23)\\ A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) &:= \left[\nabla e_j\left(x^{(k)}\right)\right]_{\mathbb{J}^{(k)}} \in \mathbb{R}^{n\times\left|\mathbb{J}^{(k)}\right|} && (3.24) \end{aligned}$$

It is shown by Herskovits and Carvalho [38] that

$$\nabla_x L\left(x^{(k)}, y_0^{(k)}\right)^T d^{(k)} \leq \nabla_x L\left(x^{(k)}, y_0^{(k)}\right)^T d_0^{(k)} \quad (3.25)$$

holds, if

$$e_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T \left(A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T \left(H^{(k)}\right)^{-1} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)\right)^{-1} \mathbb{1} > 0 \quad (3.26)$$

or

$$t^{(k)} \leq t_0^{(k)} := \frac{(1-\beta_1)\nabla_x L\left(x^{(k)}, y_0^{(k)}\right)^T d_0^{(k)}}{\left\|d_0^{(k)}\right\|^2 e_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T \left(A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T \left(H^{(k)}\right)^{-1} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)\right)^{-1} \mathbb{1}} \tag{3.27}$$

where $\beta_1 \in (0,1)$, $\mathbb{1}$ is a vector of ones of appropriate size and $L\left(x^{(k)}, y^{(k)}\right)$ is the Lagrangian function defined in (2.8). Moreover, $d_0^{(k)}$ is a descent direction, i.e.,

$$\nabla_x L\left(x^{(k)}, y_0^{(k)}\right)^T d_0^{(k)} < 0 \tag{3.28}$$

holds. It can be shown that inactive constraints, $e_j(x)$, $j \in \overline{\mathbb{J}}^{(k)}$, are still inactive in the modified QP, i.e.,

$$e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d^{(k)} < -t^{(k)} \left\|d_0^{(k)}\right\|^2, \tag{3.29}$$

for all $j \in \overline{\mathbb{J}}^{(k)}$, if

$$\nabla e_j\left(x^{(k)}\right)^T \left(H^{(k)}\right)^{-1} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \left(A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T \left(H^{(k)}\right)^{-1} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)\right)^{-1} \mathbb{1} - 1 > 0 \tag{3.30}$$

or

$$t^{(k)} \leq \left(t_1^{(k)}\right)_j \tag{3.31}$$

with

$$\left(t_1^{(k)}\right)_j := \frac{\left\|d_0^{(k)}\right\|^{-2} \left(e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d_0^{(k)}\right)}{\nabla e_j\left(x^{(k)}\right)^T \left(H^{(k)}\right)^{-1} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \left(A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T \left(H^{(k)}\right)^{-1} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)\right)^{-1} \mathbb{1} - 1} \tag{3.32}$$

holds. Therefore, we set

$$t^{(k)} := \begin{cases} 0.5 t_M^{(k)}, & \text{if } t^{(k-1)} > t_M^{(k)} \\ t^{(k-1)}, & \text{otherwise} \end{cases} \tag{3.33}$$

with

$$t_M^{(k)} := \min_{j \in \overline{\mathbb{J}}^{(k)}} \left\{\bar{t}, t_0^{(k)}, \left(t_1^{(k)}\right)_j\right\} \tag{3.34}$$

and $\bar{t} \in \mathbb{R}^+$. For a detailed description see Herskovits and Carvalho [38].

Algorithm 3. Feasible direction SQP method by Herskovits and Carvalho [38]

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$, *set parameters* $r \in (0,1)$, $\bar{t} \in \mathbb{R}^{+}$, $0 < t^{(-1)} < \bar{t}$, $\beta \in (0,1)\,, \beta_1 \in (0,1)$ *and* $H^{(0)} \in \mathbb{R}^{(n \times n)}$ *symmetric and positive definite. Let* $k := 0$.

Step 1: Solve (3.3) to obtain $\left(d_0^{(k)}, y_0^{(k)}\right)$.

Step 2: If $d_0^{(k)} = 0$ *then STOP.*

Step 3: If (3.26) does not hold, determine $t_0^{(k)}$ *according to (3.27).*

Step 4: If (3.30) does not hold, determine $\left(t_1^{(k)}\right)_j$, $j \in \overline{\mathbb{J}}^{(k)}$ *according to (3.32).*

Step 5: Update $t^{(k)}$ *according to (3.33).*

Step 6: Solve (3.22) to obtain $d^{(k)}$.

Step 7: Compute stepsize $\sigma^{(k)}$, *i.e., the first value* $\sigma^{(k)}$ *in sequence* $\{1, \beta, \beta^2, \ldots\}$ *satisfying*
$L\left(x^{(k)} + \sigma^{(k)} d^{(k)}, y_0^{(k)}\right) \leq L\left(x^{(k)}, y_0^{(k)}\right) + r\sigma^{(k)} \nabla_x L\left(x^{(k)}, y_0^{(k)}\right)^T d^{(k)}$ *and*
$e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)}\right) \leq 0, \quad j = 1, \ldots, m_f.$

Step 8: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)}$, *compute a symmetric and positive definite matrix* $H^{(k+1)} \in \mathbb{R}^{(n \times n)}$ *by a quasi-Newton formula, set* $k := k + 1$ *and goto Step 1.*

It can be shown that the resulting descent direction $d^{(k)}$ is a feasible direction according to Definition 9. In addition, the Maratos effect is avoided, see Herskovits and Carvalho [38].

3.4 A feasible SQP method by Panier and Tits

The concept of the feasible SQP method proposed by Panier and Tits [65] is based on the idea of generating an iteration sequence such that

$$f\left(x^{(k+1)}\right) \;\; < \;\; f\left(x^{(k)}\right) \tag{3.35}$$

and

$$e_j\left(x^{(k+1)}\right) \;\; \leq \;\; 0, \qquad j = 1, \ldots, m_f \tag{3.36}$$

holds. Even if the search direction $d_0^{(k)} \in \mathbb{R}^n$ resulting from (3.3) is a feasible direction according to Definition 9, local superlinear convergence is not guaranteed, since stepsize one might be infeasible. Therefore, the search direction $d_0^{(k)}$ is 'titled', i.e., replaced by a convex combination $d^{(k)} := \left(1 - t^{(k)}\right) d_0^{(k)} + t^{(k)} d_1^{(k)}$, where $d_1^{(k)} \in \mathbb{R}^n$ is an arbitrary feasible descent direction and $t^{(k)} \in [0,1]$. To ensure superlinear convergence in the neighborhood of a stationary point, one forces $t^{(k)} \to 0$ appropriately. Moreover, a line search is performed along the arc $x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$, which can be interpreted as 'bending'. Without bending, $x^{(k)} + d^{(k)}$ may neither be feasible nor yield a descent in the objective function.

After solving (3.3), a feasible descent direction $d_1^{(k)}$ is computed satisfying the following requirements

1. $d_1^{(k)} = 0$, if $x^{(k)}$ is a KKT point,
2. $\nabla f\left(x^{(k)}\right)^T d_1^{(k)} < 0$, if $x^{(k)}$ is not a KKT point,
3. $\nabla e_j\left(x^{(k)}\right)^T d_1^{(k)} < 0, \forall j = 1, \ldots, m_f$, with $e_j\left(x^{(k)}\right) = 0$, if $x^{(k)}$ is not a KKT point.

The solution $d_1^{(k)} \in \mathbb{R}^n$ of the least squares problem

$$\min_d \quad \frac{1}{2}\|d\|^2 + \max\left\{\nabla f\left(x^{(k)}\right)^T d, \max_{j=1,\ldots,m_f}\left\{e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d\right\}\right\} \tag{3.37}$$

satisfies these requirements, see Panier and Tits [65]. The feasible direction is defined by

$$d^{(k)} := \left(1 - t^{(k)}\right) d_0^{(k)} + t^{(k)} d_1^{(k)}, \tag{3.38}$$

where the weighting factor $t^{(k)} \in [0,1]$ is computed such that $t^{(k)}$ is bounded away from zero, if $d_0^{(k)} \neq 0$ and $t^{(k)} = 1$, if $d_0^{(k)}$ becomes large. There are many possibilities to compute $t^{(k)}$, for example

$$t^{(k)} := \min\left(1, \left\|d_0^{(k)}\right\|^{\beta_1}\right) \quad \text{or} \quad t^{(k)} := \frac{\left\|d_0^{(k)}\right\|^{\beta_1}}{\left(1 + \left\|d_0^{(k)}\right\|^{\beta_1}\right)}, \tag{3.39}$$

with $\beta_1 \geq 2$, see Panier and Tits [65]. As $t^{(k)}$ is bounded from below by a positive constant at every nonstationary point, $d^{(k)}$ is a feasible descent direction. To prevent the Maratos effect, an additional QP has to be solved

$$\begin{array}{lll} \min\limits_d & \frac{1}{2}\left(d^{(k)} + d\right)^T H^{(k)}\left(d^{(k)} + d\right) + \nabla f\left(x^{(k)}\right)^T\left(d^{(k)} + d\right) & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)} + d^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq -\left\|d^{(k)}\right\|^{\beta_2}, & j = 1, \ldots, m_f \end{array} \tag{3.40}$$

with $\beta_2 \in (2,3)$. Let the solution of (3.40) in iteration k be $\widetilde{d}^{(k)} \in \mathbb{R}^n$. $\widetilde{d}^{(k)}$ is set to zero, if the current iterate is far away from the solution, i.e., if

$$\left\|\widetilde{d}^{(k)}\right\| > \min\left\{\left\|d^{(k)}\right\|, C\right\} \tag{3.41}$$

holds, where $C \in \mathbb{R}^+$ is a given large number. Otherwise, global convergence can not be guaranteed. Note that in (3.40) the inequality constraints are evaluated at $x^{(k)} + d^{(k)}$, which might be infeasible.

Algorithm 4. Feasible direction SQP method by Panier and Tits [65]

Step 0: *Choose feasible starting point $x^{(0)} \in \mathbb{F}$, parameters $r \in (0,1)$, $\beta \in (0,1)$, $\beta_1 \geq 2$, $\beta_2 \in (2,3)$, $C \in \mathbb{R}^{+}$ and $H^{(0)} \in \mathbb{R}^{(n\times n)}$ symmetric and positive definite. Let $k := 0$.*

Step 1: *Solve (3.3) to obtain $d_0^{(k)}$.*

Step 2: *If $d_0^{(k)} = 0$, then STOP.*

Step 3: *Solve (3.37) to obtain $d_1^{(k)}$.*

Step 4: *Set $t^{(k)}$ according to (3.39). Let $d^{(k)} := \left(1 - t^{(k)}\right) d_0^{(k)} + t^{(k)} d_1^{(k)}$.*

Step 5: *Solve (3.40) to obtain $\widetilde{d}^{(k)}$. If $\left\|\widetilde{d}^{(k)}\right\| > \min\left\{\left\|d^{(k)}\right\|, C\right\}$, set $\widetilde{d}^{(k)} = 0$.*

Step 6: *Compute stepsize $\sigma^{(k)}$, i.e., the first value $\sigma^{(k)}$ in sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying*
$f\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ *and*
$e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq 0, \quad j = 1, \ldots, m_f.$

Step 7: *Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$, compute a symmetric and positive definite matrix $H^{(k+1)} \in \mathbb{R}^{(n\times n)}$ by a quasi-Newton formula, $k := k+1$ and goto Step 1.*

Under certain conditions, global convergence and local two-step superlinear convergence can be shown, i.e.,

$$\lim_{k\to\infty} \frac{\left\|x^{(k+2)} - x^{\star}\right\|}{\left\|x^{(k)} - x^{\star}\right\|} = 0, \tag{3.42}$$

where $x^{\star} \in \mathbb{R}^n$ is a stationary point, see Panier and Tits [65]. The algorithm can be extended, such that equality constraints can be handled, see Lawrence and Tits [53]. The resulting algorithm is implemented efficiently, see user's guide by Zhou and Tits [93].

3.5 A feasible SQP method by Lawrence and Tits

Lawrence [52] and Lawrence and Tits [54] proposed an algorithm ensuring feasibility with respect to the inequality constraints $e_j(x)$, $j = 1, \ldots, m_f$. Under certain conditions, the algorithm is globally and locally superlinear convergent. The method aims to reduce the computational work per iteration. In each iteration one QP and two related least squares problems have to be solved. Instead of solving (3.3), a QP, which is closely related to (3.7), is solved

$$\begin{array}{lll} \min\limits_{d,\delta} & \delta + \frac{1}{2} d^T H^{(k)} d & d \in \mathbb{R}^n,\ \delta \in \mathbb{R} \\ \text{s.t.} & \nabla f\left(x^{(k)}\right)^T d \leq \delta & \\ & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq t^{(k)}\delta, & j = 1, \ldots, m_f \end{array} \tag{3.43}$$

where $t^{(k)} \rightarrow 0,\ t^{(k)} \in \mathbb{R}^+$. We denote the solution of (3.43) by $\left(d^{(k)}, \delta^{(k)}\right)$. A large value of $t^{(k)}$ forces to become a feasible solution, while a small value emphasizes the descent with respect to the objective function. Moreover, $t^{(k)}$ is bounded from below by a positive constant away from a stationary point to guarantee a feasible descent direction. It can be shown that $\delta^{(k)} \leq 0$, see Birge, Qi and Wei [11]. The resulting search direction $d^{(k)} \in \mathbb{R}^n$ is supposed to converge towards the solution $d_0^{(k)} \in \mathbb{R}^n$ of (3.3) as fast as possible.

To ensure local superlinear convergence properties, a correction step $\widetilde{d}^{(k)} \in \mathbb{R}^n$ has to be computed by solving

$$\begin{array}{lll} \min\limits_{d} & \left(d^{(k)} + d\right)^T H^{(k)} \left(d^{(k)} + d\right) + \nabla f\left(x^{(k)}\right)^T \left(d^{(k)} + d\right) & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)} + d^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq -\left\|d^{(k)}\right\|^{\beta_1}, & j \in \mathbb{J}^{(k)} \end{array} \tag{3.44}$$

with $\beta_1 \in (2, 3)$ and

$$\mathbb{J}^{(k)} := \left\{ j = 1, \ldots, m_f \,\middle|\, e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d^{(k)} = t^{(k)}\delta^{(k)} \right\}. \tag{3.45}$$

(3.44) can be equivalently formulated as a least squares problem, see Lawrence and Tits [54]. If $\left\|\widetilde{d}^{(k)}\right\|$ is large, we expect to be far away from the solution. To ensure global convergence, $\widetilde{d}^{(k)}$ is set to zero in this case, see Lawrence and Tits [54]. An additional QP has to be solved to update the weighting factor $t^{(k+1)}$

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2} d^T H^{(k)} d + \nabla f\left(x^{(k)}\right)^T d & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d = 0, & j \in \mathbb{J}^{(k)} \end{array} \tag{3.46}$$

which is, after a change of variables, equivalent to a least squares problem, see Lawrence and Tits [54]. Let the solution of (3.46) be $\left(d_E^{(k)}, y_E^{(k)}\right)$. It can be shown that $d_E^{(k)}$ is an approximation of $d_0^{(k)}$ and for k sufficient large

$$d_E^{(k)} = d_0^{(k)} \tag{3.47}$$

holds. (3.46) is solved to update the perturbation parameter $t^{(k)}$ of (3.43). We get

$$t^{(k+1)} := \begin{cases} C^{(k+1)}\varepsilon^2, & \text{if } \left\|d^{(k)}\right\| \geq \varepsilon \\ C^{(k+1)} \left\|d_E^{(k+1)}\right\|^2, & \text{if } \left\|d^{(k)}\right\| < \varepsilon,\ \left\|d_E^{(k+1)}\right\| \leq \overline{D} \text{ and } y_E^{(k+1)} \geq 0 \\ C^{(k+1)} \left\|d^{(k)}\right\|^2, & \text{otherwise} \end{cases} \tag{3.48}$$

where $C^{(k+1)} \in \mathbb{R}^+,\ \overline{D} \in \mathbb{R}^+,\ \varepsilon \in \mathbb{R}^+$.

Algorithm 5. Feasible direction SQP method by Lawrence and Tits [54]

Step 0: *Choose feasible starting point* $x^{(0)} \in \mathbb{F}$*, parameters* $r \in (0,1)$, $\beta \in (0,1)$, $t^{(0)} > 0$ *and* $H^{(0)} \in \mathbb{R}^{(n \times n)}$ *symmetric and positive definite. Set* $0 < \underline{C} < \overline{C}$, $C^{(0)} \in \mathbb{R}^{+}$, $\overline{D} \in \mathbb{R}^{+}$, $\varepsilon \in \mathbb{R}^{+}$, $\beta_1 \in (2,3)$*. Let* $k := 0$.

Step 1: *Solve (3.43) to obtain* $\left(d^{(k)}, \delta^{(k)}\right)$.

Step 2: *If* $d^{(k)} = 0$*, then STOP.*

Step 3: *Solve (3.44) and denote the solution by* $\widetilde{d}^{(k)}$*. If no solution exists or* $\left\|\widetilde{d}^{(k)}\right\| > \left\|d^{(k)}\right\|$ *holds, let* $\widetilde{d}^{(k)} = 0$.

Step 4: *Compute stepsize* $\sigma^{(k)}$*, i.e., the first value* $\sigma^{(k)}$ *in sequence* $\{1, \beta, \beta^2, \ldots\}$ *satisfying*
$f\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ *and*
$e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq 0, \quad j = 1, \ldots, m_f.$

Step 5: *Set* $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$*. Select* $C^{(k+1)} \in \left[\underline{C}, \overline{C}\right]$*. Compute a symmetric and positive definite matrix* $H^{(k+1)} \in \mathbb{R}^{(n \times n)}$ *by a quasi-Newton formula.*

Step 6: *If* $\left\|d^{(k)}\right\| < \varepsilon$*, solve (3.46) to obtain* $\left(d_E^{(k+1)}, y_E^{(k+1)}\right)$.

Step 7: *Determine* $t^{(k+1)}$ *according to (3.48).*

Step 8: *Set* $k := k + 1$ *and goto Step 1.*

3.6 A feasible SQP method by Zhu, Zhang and Jian

The feasible direction SQP algorithm presented by Zhu, Zhang and Jian [94], solves two linear systems and one QP per iteration. Under certain conditions, the resulting method is globally and locally superlinear convergent. To save computational effort, the size of subproblem (3.3) is reduced to the number of ε active constraints, i.e., only constraints contained in $\mathbb{J}_\varepsilon^{(k)}$ given by (3.49) are considered in the subproblem. We denote the ε active set in iteration k by

$$\mathbb{J}_\varepsilon^{(k)} \;:=\; \left\{ j = 1, \ldots, m_f \;\middle|\; -\varepsilon \mu_j^{(k)} \leq e_j\left(x^{(k)}\right) \leq 0 \right\}, \tag{3.49}$$

with $\varepsilon \in \mathbb{R}^{+}$ and $\mu^{(k)} \in \mathbb{R}^{m_f}$.

We define $e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|}$ and $A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{n\times\left|\mathbb{J}_\varepsilon^{(k)}\right|}$ according to (3.23) and (3.24) by

$$\begin{aligned} e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) &:= \left[e_j\left(x^{(k)}\right)\right]^T_{\mathbb{J}_\varepsilon^{(k)}} && (3.50)\\ A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) &:= \left[\nabla e_j\left(x^{(k)}\right)\right]_{\mathbb{J}_\varepsilon^{(k)}}. && (3.51)\end{aligned}$$

Moreover, if

$$\det\left(A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right) < \varepsilon, \tag{3.52}$$

holds, ε has to be adapted to ensure that the gradients of the active constraints are of full rank, see Zhu, Zhang and Jian [94]. After updating $\varepsilon = 0.5\varepsilon$ and $\mathbb{J}_\varepsilon^{(k)}$ until (3.52) is violated, the ε active set (3.49) is adjusted. This procedure, called Pivoting Operation (POP), terminates in a finite number of iterations, see Gao, He and Wu [31].
The subproblem derived from (3.3) is formulated by

$$\begin{array}{lll} \min\limits_{d} & \frac{1}{2}d^T H^{(k)} d + \nabla f\left(x^{(k)}\right)^T d & d \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \le 0, & j \in \mathbb{J}_\varepsilon^{(k)} \end{array} \tag{3.53}$$

We denote the solution of (3.53) by $\left(d_0^{(k)}, y_0^{(k)}\right)$. Moreover, we obtain the matrix $A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|\times\left|\mathbb{J}_\varepsilon^{(k)}\right|}$ by applying a permutation matrix $P^{(k)} \in \mathbb{R}^{n\times n}$ yielding

$$\begin{pmatrix} A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \\ A^2_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \end{pmatrix} := P^{(k)} A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \tag{3.54}$$

such that $A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)$ is invertible. The following linear system is solved to obtain search direction $\widetilde{d}_1^{(k)} \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|}$

$$A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T d = -\left\|d_0^{(k)}\right\|^{\beta_1} \mathbb{1} - e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)} + d_0^{(k)}\right) \tag{3.55}$$

where $\beta_1 \in (2,3)$ and $\mathbb{1}$ a vector of ones of appropriate size. Note that function evaluations at infeasible points might be necessary to formulate (3.55). We define the feasible direction $d^{(k)} \in \mathbb{R}^n$ by

$$d^{(k)} := d_0^{(k)} + d_1^{(k)}, \tag{3.56}$$

with $d_1^{(k)} \in \mathbb{R}^n$ given by

$$d_1^{(k)} := \left(P^{(k)}\right)^T \begin{pmatrix} \widetilde{d}_1^{(k)} \\ \mathbb{0} \end{pmatrix} \tag{3.57}$$

where $\mathbb{0}$ is a vector of zeros of appropriate size.

To prevent the Maratos effect, an additional correction term $\widetilde{d}^{(k)} \in \mathbb{R}^n$ has to be computed. We define

$$\begin{pmatrix} \nabla f^1\left(x^{(k)}\right) \\ \nabla f^2\left(x^{(k)}\right) \end{pmatrix} := P^{(k)} \nabla f\left(x^{(k)}\right) \tag{3.58}$$

such that $\nabla f^1\left(x^{(k)}\right) \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|}$. Moreover, $\widetilde{d}_2^{(k)} \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|}$ and $d_2^{(k)} \in \mathbb{R}^n$ is denoted by

$$\widetilde{d}_2^{(k)} := \frac{\nabla f\left(x^{(k)}\right)^T d_0^{(k)}}{1 - 2\left|\mathbb{1}^T\left(A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-1} \nabla f^1\left(x^{(k)}\right)\right|} \left(A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-T} \mathbb{1}, \tag{3.59}$$

$$d_2^{(k)} := \left(P^{(k)}\right)^T \begin{pmatrix} \widetilde{d}_2^{(k)} \\ \mathbb{0} \end{pmatrix}. \tag{3.60}$$

Note that $A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)$ has to be inverted, which increases the computational effort. The correction term preventing the Maratos effect is given by

$$\widetilde{d}^{(k)} := -\nabla f\left(x^{(k)}\right)^T d_0^{(k)} \left(d_0^{(k)} + d_2^{(k)}\right). \tag{3.61}$$

In each iteration $\mu^{(k)} \in \mathbb{R}^{m_f}$ has to be adapted dependent on the solution $\left(d_0^{(k)}, y_0^{(k)}\right)$ of subproblem (3.53). We get

$$\mu_j^{(k+1)} := \begin{cases} \min\left\{\max\left\{\left(y_0^{(k)}\right)_j, \left\|d_0^{(k)}\right\|\right\}, \overline{\mu}\right\}, & \text{if } j \in \mathbb{J}_\varepsilon^{(k)} \\ \min\left\{\left\|d_0^{(k)}\right\|, \overline{\mu}\right\}, & \text{if } j \in \overline{\mathbb{J}}_\varepsilon^{(k)} \end{cases} \tag{3.62}$$

where $\overline{\mu} \in \mathbb{R}^+$ is an appropriate upper bound.
The feasible direction SQP method can be summarized according to Zhu, Zhang and Jian [94].

Algorithm 6. Feasible direction SQP method by Zhu, Zhang and Jian [94]

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$. Set parameters $r \in (0,1)$, $\varepsilon \in \mathbb{R}^+$, $\beta = 0.5$, $\beta_1 \in (2,3)$, $\beta_2 > 2$, $\beta_3 \in (0,1)$, $\overline{\mu} \in \mathbb{R}^+$ and $H^{(0)} \in \mathbb{R}^{(n \times n)}$ symmetric and positive definite. Let $k := 0$.

Step 1: Define ε and $\mathbb{J}_\varepsilon^{(k)}$ according to (3.49).

Step 2: Formulate and solve (3.53) with respect to active set $\mathbb{J}_\varepsilon^{(k)}$ to obtain $\left(d_0^{(k)}, y_0^{(k)}\right)$.

Step 3: If $d_0^{(k)} = 0$, then STOP.

Step 4: Compute $\widetilde{d}_1^{(k)}$, $d_1^{(k)}$ and $d^{(k)}$ according to (3.55), (3.57) and (3.56).

Step 5: If $\nabla f\left(x^{(k)}\right)^T d_0^{(k)} \leq \min\left\{-\beta_3 \left\|d_0^{(k)}\right\|^{\beta_2}, -\beta_3 \left\|d^{(k)}\right\|^{\beta_2}\right\}$,
$f\left(x^{(k)}+d^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d_0^{(k)}$ *and*
$e_j\left(x^{(k)}+d^{(k)}\right) \leq 0, \quad j=1,\ldots,m_f$,
then set $\sigma^{(k)}=1, \widetilde{d}^{(k)}=d^{(k)}$ *and goto Step 8.*

Step 6: Compute $\widetilde{d}_2^{(k)}$, $d_2^{(k)}$ *and* $\widetilde{d}^{(k)}$ *according to (3.59), (3.60) and (3.61).*

Step 7: Compute stepsize $\sigma^{(k)}$*, i.e., the first value* $\sigma^{(k)}$ *in sequence* $\{1,\beta,\beta^2,\ldots\}$ *satisfying*
$f\left(x^{(k)}+\sigma^{(k)}\widetilde{d}^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ *and*
$e_j\left(x^{(k)}+\sigma^{(k)}\widetilde{d}^{(k)}\right) \leq 0, \quad j=1,\ldots,m_f.$

Step 8: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)}\widetilde{d}^{(k)}$ *and compute a symmetric and positive definite matrix* $H^{(k+1)} \in \mathbb{R}^{(n\times n)}$ *by a quasi-Newton formula. Define* $\mu^{(k+1)}$ *according to (3.62).*

Step 9: Set $k := k+1$*, goto Step 1.*

3.7 A feasible SQP method by Jian and Tang

Under certain conditions, the algorithm introduced by Jian and Tang [44] is globally and locally superlinear as well as quadratically convergent without requiring strict complementary. It solves iteratively one QP. Moreover, in each iteration the feasible search direction is determined by two formulas.
Analogue to (3.53) in Section 3.6 a QP is formulated with respect to the ε active set, where $\mu^{(k)}=1$, i.e.,

$$\mathbb{J}_\varepsilon^{(k)} \quad := \quad \left\{j=1,\ldots,m_f \left| -\varepsilon \leq e_j\left(x^{(k)}\right) \leq 0 \right.\right\}, \tag{3.63}$$

with $\varepsilon \in \mathbb{R}^+$. According to POP in Section 3.6, ε has to be reduced until

$$\det\left(A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right) > \varepsilon \tag{3.64}$$

holds, where $A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{n\times\left|\mathbb{J}_\varepsilon^{(k)}\right|}$ is defined according to (3.51). As the resulting solution $d_0^{(k)} \in \mathbb{R}^n$ of (3.53) is not necessarily a feasible direction, a correction $d^{(k)} \in \mathbb{R}^n$ has to be determined given by

$$d^{(k)} \quad := \quad d_0^{(k)} - t^{(k)} A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\left(A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-1} \mathbb{1}, \tag{3.65}$$

where $t^{(k)} \in \mathbb{R}$ is defined by

$$t^{(k)} \quad := \quad \frac{\left\|d_0^{(k)}\right\| \left(d_0^{(k)}\right)^T H^{(k)} d_0^{(k)}}{2\left|\mathbb{1}^T\left(A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-1} A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T \nabla f\left(x^{(k)}\right)\right| \left\|d_0^{(k)}\right\| + 1}. \tag{3.66}$$

Note that $A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)$ has to be inverted which increases the computational effort. It can be shown that $d^{(k)}$ is a feasible descent direction. To prevent the Maratos effect we have to define a correction direction $\widetilde{d}^{(k)} \in \mathbb{R}^n$ by

$$\begin{aligned}\widetilde{d}^{(k)} \;:=\; & -A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\left(A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-1} \\ & \left(\left\|d_0^{(k)}\right\|^{\beta_1}\mathbb{1} + e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}+d^{(k)}\right) - e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) - A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T d^{(k)}\right) \quad (3.67)\end{aligned}$$

with $\beta_1 \in (2,3)$ and $e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|}$ defined according to (3.50). To compute (3.67) active constraints have to be evaluated at iterate $x^{(k)} + d^{(k)}$. Although $d^{(k)}$ is a feasible direction, the iterate might be infeasible, if $\theta^{(k)} < 1$, see Definition 9.

Algorithm 7. Feasible direction SQP method by Jian et al. [45]

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$, set parameters $r \in (0,1)$, $\beta \in (0,1)$, $\beta_1 \in (2,3)$, $\varepsilon \in \mathbb{R}^+$ and $H^{(0)} \in \mathbb{R}^{(n \times n)}$ symmetric and positive definite.
Let $k := 0$.

Step 1: Define $\mathbb{J}_\varepsilon^{(k)}$ and ε according to (3.63).

Step 2: Solve (3.53) to obtain $d_0^{(k)}$.

Step 3: If $d_0^{(k)} = 0$, then STOP.

Step 4: Compute $d^{(k)}$ according to (3.65) and $\widetilde{d}^{(k)}$ according to (3.67).

Step 5: Compute stepsize $\sigma^{(k)}$, i.e., the first value $\sigma^{(k)}$ in sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying
$f\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \le f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ *and*
$e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \le 0, \quad j = 1, \ldots, m_f.$

Step 6: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$ and compute a symmetric and positive definite matrix $H^{(k+1)} \in \mathbb{R}^{(n \times n)}$ by a quasi-Newton formula.

Step 7: Set $k := k + 1$, goto Step 1.

The algorithm is extended such that equality constraints can be handled, see Jian, Tang, Hu, Zheng [45]. Equality constraints are added to the objective function by a corresponding penalty function, see Mayne and Polak [58]. It can be shown that the solution of the resulting penalty problem is identical to the solution of the original problem, if the penalty parameters are large enough, see Jian, Tang, Hu, Zheng [45].

3.8 A feasible SQP method by Zhu

Zhu [95] presented a feasible direction algorithm, solving one QP per iteration. To prevent the Maratos effect an additional linear system has to be solved, which contains active inequality constraints only. Under certain conditions, the algorithm is globally and locally superlinear convergent. According to Section 3.5, we start with the following quadratic subproblem

$$\begin{array}{lll} \min\limits_{\delta,d} & \delta + \frac{1}{2} d^T H^{(k)} d & d \in \mathbb{R}^n,\ \delta \in \mathbb{R} \\ \text{s.t.} & \nabla f\left(x^{(k)}\right)^T d \leq \delta & \\ & e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d \leq t^{(k)}\delta, & j = 1, \ldots, m_f \end{array} \tag{3.68}$$

where $t^{(k)} \in \mathbb{R}^+$. Let the solution of (3.68) be defined by $\left(\delta^{(k)}, d^{(k)}\right)$. It can be shown that $\delta^{(k)} \leq 0$, see Birge, Qi and Wei [11]. Moreover, the resulting descent direction is feasible, if $t^{(k)}$ is positive, see Zhu [95]. To prevent the Maratos effect it is necessary to determine a correction $\widetilde{d}^{(k)} \in \mathbb{R}^n$. We define the active set by

$$\mathbb{J}^{(k)} := \left\{ j = 1, \ldots, m_f \,\middle|\, e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T d^{(k)} = t^{(k)}\delta^{(k)} \right\}. \tag{3.69}$$

Whenever $A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{n \times \left|\mathbb{J}^{(k)}\right|}$ defined by (3.24), has full rank, the correction $\widetilde{d}^{(k)}$ is computed. We define the permutation matrix $P^{(k)} \in \mathbb{R}^{n \times n}$

$$\begin{pmatrix} A^1_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \\ A^2_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \end{pmatrix} := P^{(k)} A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \tag{3.70}$$

such that $A^1_{\mathbb{J}^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{\left|\mathbb{J}^{(k)}\right| \times \left|\mathbb{J}^{(k)}\right|}$ consist of $\left|\mathbb{J}^{(k)}\right|$ linearly independent rows and $A^2_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)$ contains the remaining rows. We denote the solution of the following linear system by $\overline{d}^{(k)} \in \mathbb{R}^{\left|\mathbb{J}^{(k)}\right|}$

$$A^1_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)^T d = -\Psi^{(k)}\mathbb{1} - e_{\mathbb{J}^{(k)}}\left(x^{(k)} + d^{(k)}\right) \tag{3.71}$$

with

$$\Psi^{(k)} := \max\left\{ \left\|d^{(k)}\right\|^{\beta_1}, -t^{(k)}\delta^{(k)} \left\|d^{(k)}\right\| \right\}, \tag{3.72}$$

$\delta^{(k)} \leq 0$, $\beta_1 \in (2,3)$ and $e_{\mathbb{J}^{(k)}}(x) \in \mathbb{R}^{\left|\mathbb{J}^{(k)}\right|}$ defined according to (3.23). The correction term $\widetilde{d}^{(k)} \in \mathbb{R}^n$, preventing the Maratos effect, is given by

$$\widetilde{d}^{(k)} := \left(P^{(k)}\right)^T \begin{pmatrix} \overline{d}^{(k)} \\ \mathbb{0} \end{pmatrix} \tag{3.73}$$

where $\mathbb{0}$ is a vector of zeros of appropriate size. Formulating the linear system (3.71), active constraints have to be evaluated at $x^{(k)} + d^{(k)}$, which might be an infeasible point, although $d^{(k)}$ is a feasible direction, see Definition 9.

In each iteration, the weighting parameter $t^{(k)} \in \mathbb{R}^+$ has to be adapted,

$$t^{(k+1)} \quad := \quad \min \left\{ \bar{t}, \left\| d^{(k)} \right\|^{\beta_2} \right\}, \tag{3.74}$$

with $\beta_2 \in (0,1)$ and $\bar{t} \in \mathbb{R}^+$.

Algorithm 8. Feasible direction SQP method by Zhu [95]

```
Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$ and set parameters $t^{(0)} \in \mathbb{R}^+$, $r \in (0,1)$,
        $\bar{t} \in \mathbb{R}^+$, $\beta \in (0,1), \beta_1 \in (2,3), \beta_2 \in (0,1)$ and $H^{(0)} \in \mathbb{R}^{(n \times n)}$ symmetric and
        positive definite. Let $k := 0$.
Step 1: Solve (3.68) to obtain $d^{(k)}$.
Step 2: If $d^{(k)} = 0$, then STOP.
Step 3: Define $\mathbb{J}^{(k)}$ according to (3.69).
Step 4: If $A_{\mathbb{J}^{(k)}}\left(x^{(k)}\right)$ has full rank, solve (3.71) to obtain $\overline{d}^{(k)}$. Determine $\widetilde{d}^{(k)}$ according
        to (3.73).
        Else $\widetilde{d}^{(k)} = 0$.
Step 5: If $\left\| \widetilde{d}^{(k)} \right\| > \left\| d^{(k)} \right\|$, set $\widetilde{d}^{(k)} = 0$.
Step 6: Compute stepsize $\sigma^{(k)}$, i.e., the first value $\sigma^{(k)}$ in sequence $\{1, \beta, \beta^2, \ldots\}$
        satisfying.
        $f\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ and
        $e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq 0, \quad j = 1, \ldots, m_f$.
Step 7: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$ and compute a symmetric and positive
        definite matrix $H^{(k+1)} \in \mathbb{R}^{(n \times n)}$ by a quasi-Newton formula. Set $t^{(k+1)}$
        according to (3.74).
Step 8: Set $k := k + 1$, goto Step 1.
```

3.9 A feasible SQP method by Zhu and Jian

Zhu and Jian [96] presented a feasible direction SQP method for which, under certain conditions, global and local superlinear convergence can be shown. In each iteration one QP and a system of linear equations need to be solved. According to Section 3.6, we formulate the reduced subproblem (3.53) with the corresponding ε active set

$$\mathbb{J}_{\varepsilon}^{(k)} \quad := \quad \left\{ j = 1, \ldots, m_f \;\middle|\; -\varepsilon \mu_j^{(k)} \leq e_j\left(x^{(k)}\right) \leq 0 \right\}, \tag{3.75}$$

with $\mu^{(k)} \in \mathbb{R}^{m_f}$. The parameter $\varepsilon \in \mathbb{R}^+$ is adapted as described in Section 3.6 until

$$\det\left(A_{\mathbb{J}_{\varepsilon}^{(k)}}\left(x^{(k)}\right)^T A_{\mathbb{J}_{\varepsilon}^{(k)}}\left(x^{(k)}\right)\right) > \varepsilon, \tag{3.76}$$

holds, where $A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{n \times \left|\mathbb{J}_\varepsilon^{(k)}\right|}$ is defined according to (3.51). The solution of (3.53) is denoted by $d_0^{(k)} \in \mathbb{R}^n$ and the corresponding Lagrangian multipliers are given by $y_0^{(k)} \in \mathbb{R}^{\left|\mathbb{J}^{(k)}\right|}$. In each iteration, $\mu^{(k+1)} \in \mathbb{R}^{m_f}$ has to be adapted, using the following update rule

$$\mu_j^{(k+1)} \quad := \quad \begin{cases} \min\left\{\max\left\{\left(y_0^{(k)}\right)_j, \left\|d_0^{(k)}\right\|\right\}, \overline{\mu}\right\}, & j \in \mathbb{J}_\varepsilon^{(k)} \\ \frac{1}{2}\mu_j^{(k)}, & \text{otherwise} \end{cases} \tag{3.77}$$

where $\overline{\mu} \in \mathbb{R}^+$ is an upper bound.
As $d_0^{(k)}$ may not be a feasible direction, a correction has to be determined. Therefore, we define a permutation matrix $P^{(k)} \in \mathbb{R}^{n \times n}$

$$\begin{pmatrix} A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \\ A^2_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \end{pmatrix} \quad := \quad P^{(k)} A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \tag{3.78}$$

such that $A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \in \mathbb{R}^{|\mathbb{J}_\varepsilon^{(k)}| \times |\mathbb{J}_\varepsilon^{(k)}|}$ is the maximal linearly independent row subset of $A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)$. We define

$$\begin{pmatrix} H^1_d\left(x^{(k)}\right) \\ H^2_d\left(x^{(k)}\right) \end{pmatrix} \quad := \quad P^{(k)}\left(H^{(k)} d_0^{(k)}\right) \tag{3.79}$$

such that $H^1_d\left(x^{(k)}\right) \in \mathbb{R}^{|\mathbb{J}_\varepsilon^{(k)}|}$. Moreover, we define $\overline{d}_1^{(k)} \in \mathbb{R}^{|\mathbb{J}_\varepsilon^{(k)}|}$

$$\overline{d}_1^{(k)} \quad := \quad -t^{(k)}\left(A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-T} \mathbb{1} \tag{3.80}$$

with $t^{(k)} \in \mathbb{R}$ given by

$$t^{(k)} \quad := \quad \frac{\left\|d_0^{(k)}\right\|^2 \nabla f\left(x^{(k)}\right)^T d_0^{(k)}}{1 + 2\left|\mathbb{1}^T\left(y_0^{(k)} + \left(A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)\right)^{-1} H^1_d\left(x^{(k)}\right)\right)\right|\left\|d_0^{(k)}\right\|^2}. \tag{3.81}$$

The correction $d_1^{(k)} \in \mathbb{R}^n$ is computed by

$$d_1^{(k)} \quad := \quad \left(P^{(k)}\right)^T \begin{pmatrix} \overline{d}_1^{(k)} \\ \mathbb{0} \end{pmatrix} \tag{3.82}$$

where $\mathbb{0}$ is a vector of zeros of appropriate size. This leads to the corresponding feasible search direction

$$d^{(k)} \quad := \quad d_0^{(k)} + d_1^{(k)}. \tag{3.83}$$

In order to prevent the Maratos effect we have to solve a linear system to obtain $\widetilde{d}_1^{(k)} \in \mathbb{R}^{\left|\mathbb{J}_\varepsilon^{(k)}\right|}$

$$A^1_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T d = -\left\|d_0^{(k)}\right\|^{\beta_1}\mathbb{1} - e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)} + d^{(k)}\right) + e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) + A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T d^{(k)} \tag{3.84}$$

with $\beta_1 \in (2,3)$. The correction term $\widetilde{d}^{(k)} \in \mathbb{R}^n$ is defined by

$$\widetilde{d}^{(k)} \ := \ \left(P^{(k)}\right)^T \begin{pmatrix} \widetilde{d}_1^{(k)} \\ \mathbb{0} \end{pmatrix}. \tag{3.85}$$

Note that we have to evaluate inequality functions at $x^{(k)} + d^{(k)}$, which might be infeasible, although $d^{(k)}$ is a feasible descent direction.

Algorithm 9. Feasible direction SQP method by Zhu and Jian [96]

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$ and set parameters $r \in (0,1)$, $\beta = 0.5$ and $H^{(0)} \in \mathbb{R}^{(n\times n)}$ symmetric and positive definite. Let $k := 0$.

Step 1: Define $\mathbb{J}_\varepsilon^{(k)}$ according to (3.75).

Step 2: Formulate and solve (3.53) with respect to active set $\mathbb{J}_\varepsilon^{(k)}$ to obtain $d_0^{(k)}$.

Step 3: If $d_0^{(k)} = 0$, then STOP.

Step 4: Compute $d_1^{(k)}$ according to (3.82) and (3.80) and $d^{(k)}$ according to (3.83).

Step 5: Solve (3.84) to obtain $\widetilde{d}_1^{(k)}$. Define $\widetilde{d}^{(k)}$ according to (3.85). If $\left\|\widetilde{d}^{(k)}\right\| > \left\|d^{(k)}\right\|$, set $\widetilde{d}^{(k)} = 0$.

Step 6: Compute stepsize $\sigma^{(k)}$, i.e., the first value $\sigma^{(k)}$ in sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying
$f\left(x^{(k)} + \sigma^{(k)}d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq f\left(x^{(k)}\right) + r\sigma^{(k)}\nabla f\left(x^{(k)}\right)^T d^{(k)}$ *and*
$e_j\left(x^{(k)} + \sigma^{(k)}d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \leq 0, \quad j = 1, \ldots, m_f.$

Step 7: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)}d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$ and compute a symmetric and positive definite matrix $H^{(k+1)} \in \mathbb{R}^{(n\times n)}$ by a quasi-Newton formula. Define $\mu^{(k+1)}$ according to (3.77).

Step 8: Set $k := k+1$, goto Step 1.

Under certain assumptions global and superlinear convergence can be shown. Based on the feasible SQP algorithm presented by Zhu and Jian [96], the problem formulation is extended by equality constraints which are included in the objective function using a penalty function, see Ren, Duan, Zhu and Luo [70].

3.10 A feasible SQP method by Hu, Chen and Xiao

Hu, Chen and Xiao [41] created a feasible direction SQP method which is under certain conditions globally and locally superlinear convergent. In each iteration, one QP and two linear systems have to be solved including active constraints only. We define $A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)$ according to (3.51) by

$$\begin{aligned} A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) &:= \left[\nabla e_j\left(x^{(k)}\right)\right]_{\mathbb{J}_\varepsilon^{(k)}} \in \mathbb{R}^{n\times\left|\mathbb{J}_\varepsilon^{(k)}\right|}, && (3.86)\\ A\left(x^{(k)}\right) &:= \left[\nabla e_j\left(x^{(k)}\right)\right]_{j=1,\dots,m_f} \in \mathbb{R}^{n\times m_f}. && (3.87)\end{aligned}$$

The corresponding active set is defined by

$$\mathbb{J}_\varepsilon^{(k)} := \left\{ j=1,\dots,m_f \;\middle|\; e_j\left(x^{(k)}\right) + \varepsilon \left\| \begin{bmatrix} \nabla_x L\left(x^{(k)},y^{(k)}\right) \\ \min\left\{-f\left(x^{(k)}\right), y_1^{(k)},\dots,y_{m_f}^{(k)}\right\} \end{bmatrix} \right\|^{1/2} \geq 0 \right\} \quad (3.88)$$

where $\varepsilon \in \mathbb{R}^+$, $L(x^{(k)},y^{(k)})$ is defined in (2.8) and $y^{(k)} \in \mathbb{R}^{m_f}$ is given by

$$y^{(k)} := -\left(A\left(x^{(k)}\right)^T A\left(x^{(k)}\right) + E\left(x^{(k)}\right)^2\right)^{-1} A\left(x^{(k)}\right)^T \nabla f\left(x^{(k)}\right), \quad (3.89)$$

with

$$E\left(x^{(k)}\right) := \operatorname{diag}\left(e_1\left(x^{(k)}\right),\dots,e_{m_f}\left(x^{(k)}\right)\right). \quad (3.90)$$

Analogue to the POP procedure presented in Section 3.6, ε has to be reduced until the gradients of the active constraints are linearly independent. The quadratic subproblem is formulated analogously to (3.53). The resulting solution is denoted by $\left(d_0^{(k)}, y_0^{(k)}\right)$. In addition, $\left(d_1^{(k)}, y_1^{(k)}\right)$ is computed by solving the following linear system

$$\begin{bmatrix} H^{(k)} & A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \\ A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T & 0 \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} = \begin{bmatrix} -\nabla f\left(x^{(k)}\right) \\ -\left\|d_0^{(k)}\right\|^{\beta_1} \mathbb{1} \end{bmatrix} \quad (3.91)$$

where $\beta_1 > 2$. To define a feasible descent direction $d^{(k)} \in \mathbb{R}^n$ we use a suitable combination of the descent direction $d_0^{(k)} \in \mathbb{R}^n$ and the feasible direction $d_1^{(k)} \in \mathbb{R}^n$. We get

$$d^{(k)} := d_0^{(k)} + t^{(k)} d_1^{(k)} \quad (3.92)$$

where $t^{(k)} \in \mathbb{R}$ is given by

$$t^{(k)} := \begin{cases} \bar{t}, & \text{if } \nabla f\left(x^{(k)}\right)^T d_1^{(k)} \leq 0 \\ \dfrac{-\nabla f\left(x^{(k)}\right)^T d_0^{(k)}}{\nabla f\left(x^{(k)}\right)^T d_1^{(k)} + \beta_2}, & \text{otherwise} \end{cases} \quad (3.93)$$

with $\beta_2 \in \mathbb{R}^+$ and a corresponding upper bound $\bar{t} \in \mathbb{R}^+$. To prevent the Maratos effect and thus ensure local superlinear convergence, an additional linear system has to be solved

$$\begin{bmatrix} H^{(k)} & A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) \\ A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T & 0 \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ -\left\|d^{(k)}\right\|^{\beta_3} \mathbb{1} - \widetilde{e}_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)} + d^{(k)}\right) \end{bmatrix} \tag{3.94}$$

where $\beta_3 \in (2,3)$ and

$$\widetilde{e}_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)} + d^{(k)}\right) \;:=\; e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)} + d^{(k)}\right) - e_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right) - A_{\mathbb{J}_\varepsilon^{(k)}}\left(x^{(k)}\right)^T d^{(k)}. \tag{3.95}$$

Let the solution be $\left(\widetilde{d}^{(k)}, \widetilde{y}^{(k)}\right)$. Note that $x^{(k)} + d^{(k)}$ might be infeasible, although $d^{(k)}$ is a feasible direction. A line search is performed along the resulting search arc $x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$. The complete algorithm is formulated as follows:

Algorithm 10. Feasible direction SQP method by Hu, Chen and Xiao [41]

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$, set parameters $\varepsilon \in \mathbb{R}^+$, $\beta_1 > 2$, $\beta_2 > 0$, $\beta_3 \in (2,3)$, $r \in (0,1)$, $\beta = (0,1)$, $\bar{t} \in \mathbb{R}^+$ and $H^{(0)} \in \mathbb{R}^{(n \times n)}$ symmetric and positive definite. Let $k := 0$.

Step 1: Reduce ε until (3.86) is linearly independent.

Step 2: Formulate and solve (3.53) with respect to active set $\mathbb{J}_\varepsilon^{(k)}$ to obtain $\left(d_0^{(k)}, y_0^{(k)}\right)$.

Step 3: If $d_0^{(k)} = 0$, then STOP.

Step 4: Solve (3.91) to obtain $\left(d_1^{(k)}, y_1^{(k)}\right)$.

Step 5: Compute $d^{(k)}$ according to (3.92).

Step 6: Solve (3.94) to obtain $\left(\widetilde{d}^{(k)}, \widetilde{y}^{(k)}\right)$. If $\left\|\widetilde{d}^{(k)}\right\| > \left\|d^{(k)}\right\|$, set $\widetilde{d}^{(k)} = 0$.

Step 7: Compute stepsize $\sigma^{(k)}$, i.e., the first value $\sigma^{(k)}$ in sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying
$f\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \le f\left(x^{(k)}\right) + r\sigma^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)}$ and
$e_j\left(x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}\right) \le 0, \quad j = 1, \ldots, m_f$.

Step 8: Set $x^{(k+1)} := x^{(k)} + \sigma^{(k)} d^{(k)} + \left(\sigma^{(k)}\right)^2 \widetilde{d}^{(k)}$ and compute a symmetric and positive definite matrix $H^{(k+1)} \in \mathbb{R}^{(n \times n)}$ by a quasi-Newton formula.

Step 9: Set $k := k + 1$, goto Step 1.

4. SEQUENTIAL CONVEX PROGRAMMING METHODS

4.1 Method of Moving Asymptotes

We consider the general nonlinear optimization problem

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j(x) \leq 0, & j = 1, \ldots, m_c \end{array} \tag{4.1}$$

where $f(x)$ and $c_j(x)\,,\ j = 1, \ldots, m_c$, are at least continuously differentiable on $\mathbb{R}^n$. First we review the so-called method of moving asymptotes (MMA), see Fleury [28] and Svanberg [80]. MMA is a nonlinear programming algorithm that creates a sequence of convex and separable subproblems, which are easy to solve due to their special structure. The resulting primal solution $x^{(k)} \in \mathbb{R}^n$ in the corresponding iteration k, is used to formulate a new subproblem. MMA achieves good results in practice, although no convergence proof is given. The algorithmic scheme of MMA is illustrated in Figure 4.1.

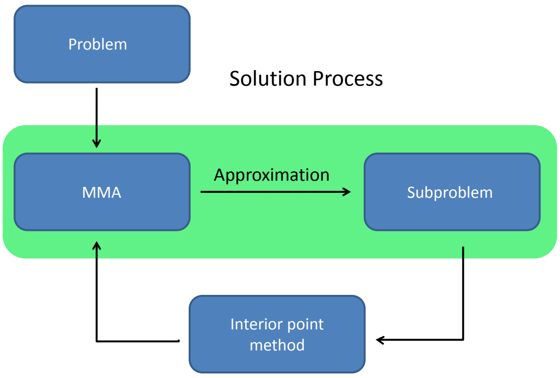


Fig. 4.1: Procedure of MMA-Algorithm

The idea behind MMA is the segmentation of the n-dimensional problem space into n one-dimensional spaces. One of the fundamental features is the introduction of two flexible asymptotes U_i and L_i, $i = 1, \ldots, n$, for each optimization variable x_i, $i = 1, \ldots, n$, which truncate the feasible region. Important additional features are

1. linearization of nonlinear inequality constraints and the objective function with respect to the lower or upper asymptote depending on the sign of the partial derivative at the current iteration point,

2. general-purpose solver applicable to any nonlinear program,

3. generation of convex and separable subproblems, i.e., diagonal Hessian matrices of the Lagrangian function, see Definition 4,

4. efficient solution of the large and sparse nonlinear subproblems by an interior point method.

In each iteration k the objective function and the inequality constraints are linearized with respect to the inverse variables $\frac{1}{U_i^{(k)} - x_i}$ and $\frac{1}{x_i - L_i^{(k)}}$ depending on the sign of the corresponding partial derivative at the current iterate. These sets are denoted by

$$I_+^{(k)} \quad := \quad \left\{ i = 1, \ldots, n \,\middle|\, \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \geq 0 \right\} \tag{4.2}$$

and

$$I_-^{(k)} \quad := \quad \left\{ i = 1, \ldots, n \,\middle|\, \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} < 0 \right\}, \tag{4.3}$$

respectively. Thus, the resulting approximation of the objective function at an iterate $x^{(k)} \in \mathbb{R}^n$ is

$$\begin{aligned} f^{(k)}(x) \quad := \quad & f\left(x^{(k)}\right) \\ + \quad & \sum_{I_+^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}} \right) \right] \\ - \quad & \sum_{I_-^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}} \right) \right] \end{aligned} \tag{4.4}$$

with $L_i^{(k)} < x_i < U_i^{(k)}$, see Svanberg [80].

The nonlinear inequality constraints $c_j(x)\,,\ j=1,\dots,m_c$, are approximated analogously to (4.4) by

$$\begin{aligned} c_j^{(k)}(x) &:= c_j\left(x^{(k)}\right) \\ &+ \sum_{I_+^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(U_i^{(k)} - x_i^{(k)}\right)^2\left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}}\right)\right] \\ &- \sum_{I_-^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(x_i^{(k)} - L_i^{(k)}\right)^2\left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}}\right)\right] \end{aligned} \tag{4.5}$$

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

$$I_+^{(j,k)} := \left\{ i=1,\dots,n \,\middle|\, \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \geq 0 \right\}, \tag{4.6}$$

$$I_-^{(j,k)} := \left\{ i=1,\dots,n \,\middle|\, \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} < 0 \right\}. \tag{4.7}$$

To prevent $U_i^{(k)} - x_i = 0,\ i=1,\dots,n$, or $x_i - L_i^{(k)} = 0,\ i=1,\dots,n$, we define suitable lower and upper bounds for $x \in \mathbb{R}^n$ within the subproblem

$$\underline{x}_i^{(k)} := x_i^{(k)} - \omega\left(x_i^{(k)} - L_i^{(k)}\right),\ i=1,\dots,n \tag{4.8}$$

and

$$\overline{x}_i^{(k)} := x_i^{(k)} + \omega\left(U_i^{(k)} - x_i^{(k)}\right),\ i=1,\dots,n \tag{4.9}$$

with $\omega \in\,]0;1[$ constant.
We obtain the subsequent subproblem by applying the approximations (4.4) and (4.5),

$$\begin{array}{lll} \min\limits_{x} & f^{(k)}(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j^{(k)}(x) \leq 0, & j=1,\dots,m_c \\ & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i=1,\dots,n \end{array} \tag{4.10}$$

The functions are defined on the subset $X^{(k)}$ given by the box constraints (4.8) and (4.9), i.e.,

$$X^{(k)} := \left\{ x \in \mathbb{R}^n \,\middle|\, \underline{x}^{(k)} \leq x \leq \overline{x}^{(k)} \right\}. \tag{4.11}$$

Moreover,

$$L_i^{(k)} < \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)} < U_i^{(k)} \tag{4.12}$$

holds. The first and second order derivatives of the convex approximations can be given analytically by

$$\frac{\partial f^{(k)}(x)}{\partial x_i} = \begin{cases} \dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} \dfrac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^2}, & \text{if } i \in I_+^{(k)} \\ \dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} \dfrac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left(x_i - L_i^{(k)}\right)^2}, & \text{otherwise} \end{cases} \tag{4.13}$$

$$\frac{\partial^2 f^{(k)}(x)}{\partial x_i \partial x_j} = 0, \qquad \forall i \neq j \tag{4.14}$$

$$\frac{\partial^2 f^{(k)}(x)}{\partial^2 x_i} = \begin{cases} 2\,\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} \dfrac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^3}, & \text{if } i \in I_+^{(k)} \\ -2\,\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} \dfrac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left(x_i - L_i^{(k)}\right)^3}, & \text{otherwise} \end{cases} \tag{4.15}$$

The derivatives for inequality constraints $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, can be obtained by replacing $f(x)$ by $c_j(x)$, $j = 1, \ldots, m_c$, and $I_+^{(k)}$ by $I_+^{(j,k)}$. It is easy to see that the functions are strictly convex, if $\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} \neq 0$ and $\dfrac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \neq 0$ holds for all $i = 1, \ldots, n$, and $j = 1, \ldots, m_c$.

The approximations (4.4) and (4.5) are convex and of first order, i.e.,

$$\begin{array}{ll} f^{(k)}\left(x^{(k)}\right) = f\left(x^{(k)}\right), & c_j^{(k)}\left(x^{(k)}\right) = c_j\left(x^{(k)}\right),\ \forall\, j = 1, \ldots, m_c \\ \nabla f^{(k)}\left(x^{(k)}\right) = \nabla f\left(x^{(k)}\right), & \nabla c_j^{(k)}\left(x^{(k)}\right) = \nabla c_j\left(x^{(k)}\right),\ \forall\, j = 1, \ldots, m_c \\ f^{(k)} \text{ convex}, & c_j^{(k)} \text{ convex},\ \forall\, j = 1, \ldots, m_c \\ f^{(k)} \text{ separable}, & c_j^{(k)} \text{ separable},\ \forall\, j = 1, \ldots, m_c \end{array} \tag{4.16}$$

The subproblem can be solved by an interior point method, where the separability of the approximations $f^{(k)}(x)$ and $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, can be exploited.

In each iteration k, the asymptotes have to be adapted. The update rules are presented in the following algorithm, according to Zillober [104] extended by some additional conditions which are essential to prove convergence of the SCP algorithm presented in Section 4.2.

Algorithm 11. Update of asymptotes
For iteration number $k = 0, 1, \ldots$ and constants $L_{min} < U_{max}$, $\xi \geq 0$, $0 < T_1 < 1$ and $T_2 > 1$ we compute for each $i = 1, \ldots, n$

$$k < 2: \quad L_i^{(k)} := x_i^{(k)} - \max\left\{1, \left|x_i^{(k)}\right|\right\},$$
$$U_i^{(k)} := x_i^{(k)} + \max\left\{1, \left|x_i^{(k)}\right|\right\}.$$

$k \geq 2:$ *If $sign\left(x_i^{(k)} - x_i^{(k-1)}\right) \neq sign\left(x_i^{(k-1)} - x_i^{(k-2)}\right)$, then*

$$L_i^{(k)} := \max\left\{x_i^{(k)} - \max\left\{\xi, T_1\left(x_i^{(k-1)} - L_i^{(k-1)}\right)\right\}, L_{min}\right\},$$
$$U_i^{(k)} := \min\left\{x_i^{(k)} + \max\left\{\xi, T_1\left(U_i^{(k-1)} - x_i^{(k-1)}\right)\right\}, U_{max}\right\},$$

else

$$L_i^{(k)} := \max\left\{x_i^{(k)} - \max\left\{\xi, T_2\left(x_i^{(k-1)} - L_i^{(k-1)}\right)\right\}, L_{min}\right\},$$
$$U_i^{(k)} := \min\left\{x_i^{(k)} + \max\left\{\xi, T_2\left(U_i^{(k-1)} - x_i^{(k-1)}\right)\right\}, U_{max}\right\},$$

Suitable values for the parameters are $T_1 = 0.7$ and $T_2 = 1/T_1$ respectively, see Svanberg [80], while Zillober et al. [105] propose $T_1 = 0.7$ and $T_2 = 1.15$. Within the MMA procedure $\xi = 0$ and $U_{\max} = -L_{\min} = \infty$. To prove global convergence of SCP, it is essential that $\xi > 0$. Within the algorithm, the values are set to $\xi = 0.5$ and $U_{\max} = -L_{\min} = 1.D5$. We distinguish two different situations. If $\text{sign}\left(x_i^{(k)} - x_i^{(k-1)}\right) \neq \text{sign}\left(x_i^{(k-1)} - x_i^{(k-2)}\right)$, the distance between the asymptotes is reduced to prevent oscillation. As a consequence the domain shrinks. Otherwise, the distance is enlarged to allow larger steps and to speed up convergence. Svanberg [80] determines the asymptotes in the first iteration dependent on box constraints of the original problem, if they exist.

We can now formulate the corresponding MMA algorithm according to Svanberg [80].

Algorithm 12. Method of Moving Asymptotes

Step 0: Choose starting point $x^{(0)} \in \mathbb{R}^n$. Set parameter $\xi = 0$, $L_{min} = -\infty$, $U_{max} = \infty$, $T_2 > 1$, $0 < T_1 < 1$ and $\omega \in]0;1[$. Compute $f\left(x^{(0)}\right), \nabla f\left(x^{(0)}\right)$, $c_j\left(x^{(0)}\right)$, $\nabla c_j\left(x^{(0)}\right)$, $j = 1, \ldots, m_c$. Let $k := 0$.

Step 1: Determine $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, by Algorithm 11 and let $f^{(k)}(x)$, $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, be defined by (4.4) and (4.5). Define $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, according to (4.8) and (4.9). Formulate subproblem (4.10) for the corresponding iteration k.

Step 2: Solve (4.10). Let $x^{(k+1)}$ be the optimal solution of the subproblem.

Step 3: If $x^{(k+1)} = x^{(k)}$, then STOP. $x^{(k+1)}$ is the solution of (4.1).

Step 4: Set $k = k + 1$ and compute $f\left(x^{(k)}\right), \nabla f\left(x^{(k)}\right)$, $c_j\left(x^{(k)}\right), \nabla c_j\left(x^{(k)}\right)$, $j = 1, \ldots, m_c$. Goto Step 1.

For this method convergence cannot be guaranteed in a formal way despite of excellent numerical results. In later papers, Svanberg presented an extension of his MMA method such that convergence can be shown, see Svanberg [81] and [82]. The resulting algorithm, called GCMMA, is presented in Section 4.3.

4.2 The SCP-Method of Zillober

The sequential convex programming method (SCP) is an extension of the method of moving asymptotes. We proceed from the following optimization problem, where equality constraints are included additionally,

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j(x) = 0, & j = 1, \ldots, m_e \\ & c_j(x) \leq 0, & j = m_e + 1, \ldots, m_c \end{array} \tag{4.17}$$

Moreover, SCP ensures convergence by introducing a merit function and a corresponding line search procedure, see Zillober [101]. The merit function combines the objective function and the constraints in a suitable way. A possible merit function for problem (4.17) is the augmented Lagrangian function $\Phi_\rho : \mathbb{R}^{n+m_c} \to \mathbb{R}$ for a given set of penalty parameters $\rho_j > 0$, $j = 1, \ldots, m_c$,

$$\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} = f(x) + \sum_{j=1}^{m_c} \begin{cases} y_j c_j(x) + \dfrac{\rho_j}{2} c_j^2(x), & \text{if } j \in J(x) \\ -\dfrac{y_j^2}{2\rho_j}, & \text{otherwise} \end{cases} \tag{4.18}$$

where $y \in \mathbb{R}^{m_c}$ are the corresponding Lagrangian multipliers. Moreover, we define the active set with respect to the augmented Lagrangian by $J(x)$ and its complement by $\overline{J}(x)$,

$$\begin{aligned} J(x) &:= \{1 \leq j \leq m_e\} \\ &\quad \cup \left\{ m_e + 1 \leq j \leq m_c \,\middle|\, -\frac{y_j}{\rho_j} \leq c_j(x) \right\} \qquad (4.19) \\ \overline{J}(x) &:= \{m_e + 1 \leq j \leq m_c \mid j \notin J\} \,. \qquad (4.20) \end{aligned}$$

It can be shown that $\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix}$ is differentiable. The penalty parameters $\rho \in \mathbb{R}^{m_c}$ must be carefully adapted during the solution process to guarantee a sufficient descent and global convergence, see Schittkowski [73]. Choosing the augmented Lagrangian merit function is motivated by the following properties, Zillober [101]:

Lemma 4.1. *1. A point* $(x^\star, y^\star)$ *is stationary for* $\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix}$ *defined by (4.18) for a positive fixed* $\rho \in \mathbb{R}^{m_c}$*, if and only if it is stationary for problem (4.17).*

2. Let $(x^\star, y^\star)$ *be stationary for problem (4.17) and let the gradients of the active constraints be linearly independent in* $x^\star$*, i.e., Definition 5 (LICQ) holds. Then there exists a positive parameter* $\rho^\star \in \mathbb{R}^{m_c}$*, such that* $x^\star$ *is a local minimizer for* $\Phi_\rho \begin{pmatrix} x \\ y^\star \end{pmatrix}$, $\forall\, \rho \geq \rho^\star$*.*

Proof. Fletcher [23, 24]. □

To ensure strict convexity of the approximated objective function $f^{(k)}(x)$ and thus an unique solution of the subproblem, an additional parameter $\tau > 0$ is introduced to ensure that $\frac{\partial^2 f^{(k)}(x)}{\partial^2 x_i} > 0$ holds for all $i = 1, \ldots, n$, see Zillober [104]. The resulting approximation of the objective function at iterate $x^{(k)}$ is

$$\begin{aligned} f^{(k)}(x) \;&:=\; f\left(x^{(k)}\right) \\ &+\; \sum_{I_+^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}} \right) \right] \\ &-\; \sum_{I_-^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}} \right) \right] \qquad (4.21) \\ &+\; \sum_{I_+^{(k)}} \tau \frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} + \sum_{I_-^{(k)}} \tau \frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \end{aligned}$$

with $L_i^{(k)} < x_i < U_i^{(k)}$, $\tau > 0$, $I_+^{(k)}$ and $I_-^{(k)}$ defined by (4.2) and (4.3). Equality constraints $c_j(x)$, $j = 1, \ldots, m_e$, are linearized by

$$c_j^{(k)}(x) \;:=\; c_j\left(x^{(k)}\right) \;+\; \sum_{i=1}^{n} \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(x_i - x_i^{(k)}\right). \qquad (4.22)$$

The inequality constraints $c_j(x)$, $j = m_e + 1, \ldots, m_c$, are approximated analogously to the MMA algorithm, see (4.5). For an illustration, Figure 4.2 presents the approximation of the objective function $f(x)$ with respect to an upper asymptote $U_i^{(k)}$ for given $i \in I_+^{(k)}$ and k.

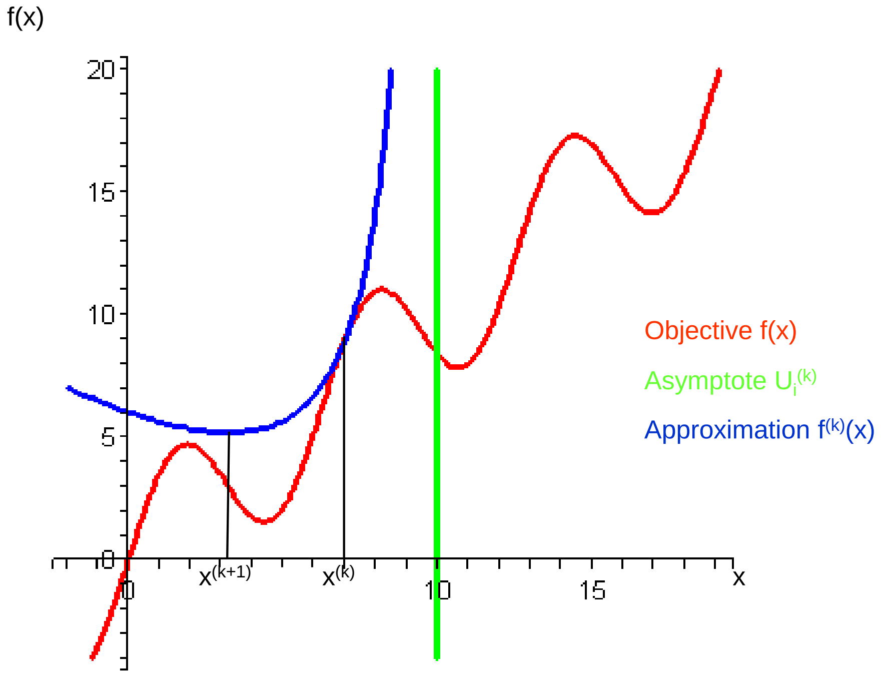


Fig. 4.2: Approximation scheme of SCP

The first and second order derivatives of the convex approximations can be given analytically, see Ertel, Schittkowski and Zillober [20].

$$\frac{\partial f^{(k)}(x)}{\partial x_i} = \begin{cases} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \frac{\left(U_i^{(k)}-x_i^{(k)}\right)^2}{\left(U_i^{(k)}-x_i\right)^2} + \tau \frac{\left(x_i-x_i^{(k)}\right)^2+2\left(x_i-x_i^{(k)}\right)\left(U_i^{(k)}-x_i\right)}{\left(U_i^{(k)}-x_i\right)^2}, & \text{if } i \in I_+^{(k)} \\ \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \frac{\left(x_i^{(k)}-L_i^{(k)}\right)^2}{\left(x_i-L_i^{(k)}\right)^2} - \tau \frac{\left(x_i-x_i^{(k)}\right)^2-2\left(x_i-x_i^{(k)}\right)\left(x_i-L_i^{(k)}\right)}{\left(x_i-L_i^{(k)}\right)^2}, & \text{otherwise} \end{cases} \tag{4.23}$$

$$\frac{\partial^2 f^{(k)}(x)}{\partial x_i \partial x_j} = 0, \qquad \forall i \neq j \tag{4.24}$$

$$\frac{\partial^2 f^{(k)}(x)}{\partial^2 x_i} = \begin{cases} 2\,\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \frac{\left(U_i^{(k)}-x_i^{(k)}\right)^2}{\left(U_i^{(k)}-x_i\right)^3} + 2\tau \frac{\left(U_i^{(k)}-x_i^{(k)}\right)^2}{\left(U_i^{(k)}-x_i\right)^3}, & \text{if } i \in I_+^{(k)} \\ -2\,\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \frac{\left(x_i^{(k)}-L_i^{(k)}\right)^2}{\left(x_i-L_i^{(k)}\right)^3} + 2\tau \frac{\left(x_i^{(k)}-L_i^{(k)}\right)^2}{\left(x_i-L_i^{(k)}\right)^3}, & \text{otherwise} \end{cases} \tag{4.25}$$

The second order derivatives are positive, as

$$\begin{aligned}
&\frac{\partial}{\partial x_i} \;\; \tau \; \frac{\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right)}{\left(U_i^{(k)} - x_i\right)^2} \\
= \;& \tau \; \frac{2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right)^2 + 2\left(U_i^{(k)} - x_i\right)^2\left(\left(U_i^{(k)} - x_i\right) - \left(x_i - x_i^{(k)}\right)\right)}{\left(U_i^{(k)} - x_i\right)^4} \\
& + \tau \; \frac{2\left(U_i^{(k)} - x_i\right)\left(\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right)\right)}{\left(U_i^{(k)} - x_i\right)^4} \\
= \;& 2\tau \; \frac{\left(U_i^{(k)} - x_i\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right) + \left(x_i - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^3} \\
= \;& 2\tau \; \frac{\left(\left(U_i^{(k)} - x_i\right) + \left(x_i - x_i^{(k)}\right)\right)^2}{\left(U_i^{(k)} - x_i\right)^3} \\
= \;& 2\tau \; \frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^3} > 0
\end{aligned}$$

holds for $i \in I_+^{(k)}$. This can be shown analogously for $i \in I_-^{(k)}$. The derivatives for inequality constraints $c_j^{(k)}(x)$, $j = m_e + 1, \ldots, m_c$, can be obtained by replacing $f(x)$ by $c_j(x)$ and $\tau = 0$. The corresponding subproblem is formulated by

$$\begin{array}{lll}
\min\limits_{x} & f^{(k)}(x) & x \in \mathbb{R}^n \\
\text{s.t.} & c_j^{(k)}(x) = 0, & j = 1, \ldots, m_e \\
& c_j^{(k)}(x) \leq 0, & j = m_e + 1, \ldots, m_c \\
& \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i = 1, \ldots, n
\end{array} \tag{4.26}$$

where $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, are defined according to (4.8) and (4.9). Due to strict convexity of the objective function the resulting subproblem (4.26) possesses an unique solution, Zillober [98].

We denote the primal solution of subproblem (4.26) in iteration k by $z^{(k)} \in \mathbb{R}^n$ and the dual solution by $v^{(k)} \in \mathbb{R}^{m_c}$. In each iteration, it is tested whether $\left(z^{(k)}, v^{(k)}\right)$ yields a sufficient descent with respect to the augmented Lagrangian merit function. Therefore, the so-called Armijo steplength algorithm, see Armijo [2], Ortega and Rheinboldt [63], is applied. In each iteration k, the stepsize $\sigma^{(k,i)}$, with $\sigma^{(k,0)} := 1$, is reduced by a constant factor $\beta \in (0,1)$ iteratively, i.e.,

$$\sigma^{(k,i+1)} \;\; := \;\; \beta\sigma^{(k,i)} \tag{4.27}$$

until the following condition is satisfied for the first time

$$\Phi_{\rho^{(k)}}\left(\left(\begin{array}{c} x^{(k)} \\ y^{(k)} \end{array}\right)+\sigma^{(k,i)}d^{(k)}\right) \leq \Phi_{\rho^{(k)}}\left(\begin{array}{c} x^{(k)} \\ y^{(k)} \end{array}\right)+r\sigma^{(k,i)}\nabla\Phi_{\rho^{(k)}}\left(\begin{array}{c} x^{(k)} \\ y^{(k)} \end{array}\right)^T d^{(k)}, \tag{4.28}$$

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

$$d^{(k)} := \left(\begin{array}{c} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{array}\right). \tag{4.29}$$

Condition (4.28) ensures a sufficient descent in the augmented Lagrangian merit function. To update the penalty parameter $\rho_i^{(k)}$, $i = 1, \ldots, m_c$, additional parameters $\eta_i^{(k)}$, $i = 1, \ldots, n$, are introduced, which estimate the curvature of the approximated objective function $f^{(k)}(x)$

$$\eta_i^{(k)} := \begin{cases} \left(\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i}+\tau\right)\dfrac{2U_i^{(k)}-z_i^{(k)}-x_i^{(k)}}{\left(U_i^{(k)}-z_i^{(k)}\right)^2}, & \text{if } i \in I_+^{(k)} \\ -\left(\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i}-\tau\right)\dfrac{-2L_i^{(k)}+z_i^{(k)}+x_i^{(k)}}{\left(z_i^{(k)}-L_i^{(k)}\right)^2}, & \text{otherwise} \end{cases} \tag{4.30}$$

and we define

$$\eta^{(k)} := \min_{i=1,\ldots,n} \eta_i^{(k)}. \tag{4.31}$$

The penalty parameters are updated until the descent property

$$\nabla\Phi_{\rho^{(k)}}\left(\begin{array}{c} x^{(k)} \\ y^{(k)} \end{array}\right)^T d^{(k)} \leq -\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{2} \tag{4.32}$$

is satisfied, where $\delta^{(k)} \in \mathbb{R}$ denotes the norm of the search direction with respect to the primal variable $x^{(k)}$, i.e.,

$$\delta^{(k)} := \left\|z^{(k)} - x^{(k)}\right\|_2. \tag{4.33}$$

Within the update loop the penalty parameters are denoted by $\rho_j^{(k,i)}$, $j = 1, \ldots, m_c$, where i denotes the i-th penalty parameter update within iteration k. $\rho_j^{(k,0)}$ is initialized by $\rho_j^{(k-1)}$, $j = 1, \ldots, m_c$. The corresponding update is described by Algorithm 13 according to Zillober [103].

Algorithm 13. Update of penalty parameters

Let $\kappa_1 > 1$, $\kappa_2 > \kappa_1$ be suitable constants and let k be the index of the current iteration. Let $x^{(k)} \in \mathbb{R}^n$ the current primal and $y^{(k)} \in \mathbb{R}^{m_c}$ the current dual variable. Moreover, $\left(z^{(k)}, v^{(k)}\right)$ is the solution of subproblem (4.26) defined in $x^{(k)}$ and $\rho_j^{(k,i)}$, $j = 1, \ldots, m_e$, is a given penalty parameter.

```
If $j \in \{1 \le j \le m_e\}$ or $j \in \left\{ m_e + 1 \le j \le m_c \;\middle|\; -\frac{y_j^{(k)}}{\rho_j^{(k,i)}} \le c_j\left(x^{(k)}\right) \right\}$ :
    if  $c_j\left(x^{(k)}\right) > 0$ and $\nabla c_j\left(x^{(k)}\right)^T \left(z^{(k)} - x^{(k)}\right) \neq 0$ or
        $c_j\left(x^{(k)}\right) < 0$ and $\nabla c_j\left(x^{(k)}\right)^T \left(z^{(k)} - x^{(k)}\right) > 0$ :
            $\rho_j^{(k,i+1)} := \min\left\{ \kappa_2 \rho_j^{(k,i)},\ \max\left\{ \kappa_1 \rho_j^{(k,i)},\ \left| \frac{2\left(v_j^{(k)} - y_j^{(k)}\right)}{c_j\left(x^{(k)}\right)} \right| \right\} \right\}$
    else:
            $\rho_j^{(k,i+1)} := \kappa_1 \rho_j^{(k,i)}$
Else:
    if  $\left(v_j^{(k)} - y_j^{(k)}\right) < 0$ :
            $\rho_j^{(k,i+1)} := \min\left\{ \kappa_2 \rho_j^{(k,i)},\ \max\left\{ \kappa_1 \rho_j^{(k,i)},\ \left| \frac{y_j^{(k)}\left(v_j^{(k)} - y_j^{(k)}\right) 4 m_c}{\eta^{(k)}\left(\delta^{(k)}\right)^2} \right| \right\} \right\}$
    else:
            $\rho_j^{(k,i+1)} := \kappa_1 \rho_j^{(k,i)}$
```

κ_1 prevents that the penalty parameters converge too slowly towards an upper bound. κ_2 ensures that the penalty parameters do not increase too quickly. Zillober [102] proposes to set $\kappa_1 = 2$ and $\kappa_2 = 10$.

To prove global convergence it is essential that the lower and upper asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, are adapted carefully and satisfy certain conditions see Zillober [98] or Ertel [19]. We define

Definition 4.1. *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. *$L_i^{(k)} \le x_i^{(k)} - \xi$, $U_i^{(k)} \ge x_i^{(k)} + \xi$, for all $i = 1, ..., n$, and $k \ge 0$.*
2. *$L_i^{(k)} \ge L_{min}$, $U_i^{(k)} \le U_{max}$, $\forall\ k\ \ge 0$, $i = 1, \ldots, n$.*

The first part of this definition prevents the curvature of the approximations from becoming too steep. The second part prevents that $L_i^{(k)} \to -\infty$ and $U_i^{(k)} \to \infty$ as

$$-\infty < L_{\min} \le L_i^{(k)} < U_i^{(k)} \le U_{\max} < \infty \tag{4.34}$$

holds. The asymptotes need to be feasible according to Definition 4.1 to ensure convergence of sequential convex programming (SCP) methods, see Zillober [102]. In general, it is possible to choose different asymptotes for the objective and (each) constraint. Although this might improve the performance of the algorithm we proceed with one pair of asymptotes for both, objective and constraints, as the computational effort is much higher otherwise, Zillober [103]. Algorithm 11 ensures a feasible sequence of asymptotes according to Definition 4.1.

The SCP algorithm introduced by Zillober [104] can be summarized as follows:

Algorithm 14. Sequential Convex Programming

Step 0: Choose starting point $x^{(0)} \in \mathbb{R}^n$ *and* $y^{(0)} \geq 0$. *Compute* $f\left(x^{(0)}\right), \nabla f\left(x^{(0)}\right),$ $c_j\left(x^{(0)}\right), \nabla c_j\left(x^{(0)}\right),\ j = 1, \ldots, m_c$. *Set parameters* $\xi > 0$, $L_{min} < U_{max}$, $T_2 > 1$, $0 < T_1 < 1$, $\omega \in]0;1[$, $r \in (0,1)$, $\beta \in (0,1)$, $\tau > 0$, $\kappa_2 > \kappa_1 > 1$ *and penalty parameters* $\rho_j^{(-1)} > 0,\ j = 1, \ldots, m_c$. *Let* $k := 0$.

Step 1: Determine $L_i^{(k)}$ *and* $U_i^{(k)},\ i = 1, \ldots, n$, *by Algorithm 11. Let* $f^{(k)}(x)$, $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, *be defined by (4.21), (4.22) and (4.5). Define* $\underline{x}_i^{(k)}$ *and* $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, *according to (4.8) and (4.9). Formulate (4.26) for the corresponding iteration* k.

Step 2: Solve (4.26). Let $z^{(k)}$ *be the optimal solution of (4.26) and* $v^{(k)}$ *the vector of the corresponding Lagrangian multipliers.*

Step 3: If $z^{(k)} = x^{(k)}$, *then STOP.* $\left(x^{(k)}, v^{(k)}\right)$ *is a KKT point of (4.17).*

Step 4: Let $d^{(k)} := \begin{pmatrix} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{pmatrix}, \delta^{(k)} := \left\| z^{(k)} - x^{(k)} \right\|_2$ *and* $\eta^{(k)}$ *as defined in (4.31). Let* $i = 0$ *and* $\rho^{(k,0)} := \rho^{(k-1)}$.

Step 5: Compute $\Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}, \nabla \Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}, \nabla \Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}$.

Step 6: If $\nabla \Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} > -\dfrac{\eta^{(k)} \left(\delta^{(k)}\right)^2}{2}$, *update penalty parameters according to Algorithm 13. Let* $i = i + 1$ *and goto Step 5.*
Otherwise, let $\rho^{(k)} := \rho^{(k,i)}$, $i = 0$ *and* $\sigma^{(k,0)} := 1$.

Step 7: 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* $\Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)}\right)$.
If (4.28) 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 8: 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 9: Compute $\nabla f\left(x^{(k)}\right), \nabla c_j\left(x^{(k)}\right),\ j = 1, \ldots, m_c$, *and goto Step 1.*

Subproblem (4.26) can be solved by an interior point method, see Zillober [99]. The size of the primal-dual system of linear equations can be reduced depending on the number of constraints and variables, see Zillober [100]. The asymptotes truncate the feasible region and allow to control the curvature of the merit function. A convergence proof is given in Zillober [97] and [102]. The line search procedure can be replaced by a filter method. A comparative numerical study of SCP-Filter and the presented SCP method is given in Ertel [19]. A trust-region SCP algorithm is introduced in Ni [60]. Some numerical results for problems resulting from topology optimization are given in Ni, Zillober and Schittkowski [61].

4.3 The Globally Convergent Method of Moving Asymptotes

As no convergence proof for the MMA algorithm presented in Section 4.1 can be given, Svanberg [83] presented an extension called globally convergent method of moving asymptotes (GCMMA). Starting from a feasible point $x^{(0)} \in \mathbb{R}^n$, GCMMA creates a sequence of feasible iterates. As GCMMA is not able to handle equality constraints, we proceed from the following problem formulation

$$\begin{array}{lll} \min\limits_{x} & f\left(x\right) & x \in \mathbb{R}^n \\ \text{s.t.} & e_j\left(x\right) \leq 0, & j = 1, \ldots, m_f \\ & \underline{x}_i \leq x_i \leq \overline{x}_i, & i = 1, \ldots, n \end{array} \tag{4.35}$$

containing box constraints. Moreover, $f(x)$ and $e_j(x)$, $j = 1, \ldots, m_f$, are at least continuously differentiable on X given by

$$X \ := \ \left\{ x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x} \right\}. \tag{4.36}$$

In contrast to MMA and SCP, the subproblems are formulated by approximating the functions with respect to both inverse variables $\frac{1}{U_i^{(k)} - x_i}$ and $\frac{1}{x_i - L_i^{(k)}}$. The resulting approximations are strictly convex and separable. To ensure a feasible iteration sequence, the method combines an inner and an outer iteration process. Within the inner iteration process a sequence of subproblems is solved, until the objective function is decreased and the constraints yield a smaller function value than their approximations at the solution of the subproblem. Otherwise, the iterate is rejected and a new subproblem is formulated and solved. Therefore, we introduce a second iteration index. We denote the approximations of the objective function by $f^{(k,p)}\left(x\right)$ and the approximations of the constraints by $e_j^{(k,p)}\left(x\right)$, $j = 1, \ldots, m_f$, where k denotes the outer iteration and p the inner iteration, respectively. The approximation scheme of the objective function in iteration (k,p) is given by

$$\begin{aligned} f^{(k,p)}(x) \; := \; & f\left(x^{(k)}\right) \\ + & \sum_{I_+^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(T_3 \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}} \right) \right.\right. \\ & \left.\left. + \; T_4 \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}} \right) \right) \right] \\ - & \sum_{I_-^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left(T_4 \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}} \right) \right.\right. \\ & \left.\left. + \; T_3 \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}} \right) \right) \right] \\ + & \sum_{i=1}^{n} \frac{\tau_0^{(k,p)}}{\overline{x}_i - \underline{x}_i} \left(\left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}} \right) \right. \\ & \left. + \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}} \right) \right) \end{aligned} \tag{4.37}$$

with $T_3 > 1$ and $0 < T_4 < 1$, $\tau_0^{(k,p)} \in \mathbb{R}^+$, $I_+^{(k)}$ and $I_-^{(k)}$ are defined according to (4.2) and (4.3). Moreover, the asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, are computed by Algorithm 11. Svanberg [84] proposes $T_3 = 1.001$ and $T_4 = 0.001$. The inequality constraints $e_j(x)$, $j = 1, \ldots, m_f$, are approximated analogously. In contrast to the MMA and SCP algorithm, see Section 4.1 and Section 4.2, a parameter $\tau_j^{(k,p)} \in \mathbb{R}^+$, $j = 1, \ldots, m_f$, is introduced for each constraint and the objective function, which leads to strictly convex approximations. The corresponding subproblem is given by

$$\begin{array}{lll} \min\limits_{x} & f^{(k,p)}(x) & x \in \mathbb{R}^n \\ \text{s.t.} & e_j^{(k,p)}(x) \le 0, & j = 1, \ldots, m_f \\ & \underline{x}_i^{(k)} \le x_i \le \overline{x}_i^{(k)}, & i = 1, \ldots, n \end{array} \tag{4.38}$$

where $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, are defined according to (4.8) and (4.9) with $\omega = 0.9$, see Svanberg [84]. As the resulting subproblem is strictly convex, see Svanberg [82], it exhibits an unique solution denoted by $z^{(k,p)} \in \mathbb{R}^n$. If the following conditions hold

$$f^{(k,p)}\left(z^{(k,p)}\right) \;\ge\; f\left(z^{(k,p)}\right) \tag{4.39}$$

$$0 \;\ge\; e_j^{(k,p)}\left(z^{(k,p)}\right) \;\ge\; e_j\left(z^{(k,p)}\right), \quad j = 1, \ldots, m_f \tag{4.40}$$

the inner iteration is terminated and the next iterate is given by

$$x^{(k+1)} := z^{(k,p)}. \tag{4.41}$$

Otherwise, the inner iteration process is continued, i.e., p is increased by one and a more restrictive subproblem (4.38) is formulated by increasing $\tau_j^{(k,p)}$, $j = 1, \ldots, m_f$, for each constraint which violates (4.40). $\tau_0^{(k,p)}$ is increased, if the objective function violates (4.39), see Svanberg [84]. $\tau_0^{(k,0)}$ is initialized in each outer iteration k by

$$\tau_0^{(k,0)} := \frac{0.1}{n} \sum_{i=1}^{n} \left| \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \right| \left(\overline{x}_i - \underline{x}_i\right) \tag{4.42}$$

and $\tau_0^{(k,p+1)}$ is updated according to

$$\tau_0^{(k,p+1)} := \begin{cases} \min\left\{10\tau_0^{(k,p)}, 1.1\left(\tau_0^{(k,p)} + \iota_0^{(k,p)}\right)\right\}, & \text{if } f^{(k,p)}\left(z^{(k,p)}\right) < f\left(z^{(k,p)}\right) \\ \tau_0^{(k,p)}, & \text{otherwise} \end{cases} \tag{4.43}$$

with $\iota_0^{(k,p)} \in \mathbb{R}$ given by

$$\begin{aligned} \iota_0^{(k,p)} := \; & \left(f\left(z^{(k,p)}\right) - f^{(k,p)}\left(z^{(k,p)}\right)\right) \\ & \left(\sum_{i=1}^{n} \frac{\left(U_i^{(k)} - z_i^{(k,p)}\right)\left(z_i^{(k,p)} - L_i^{(k)}\right)\left(\overline{x}_i - \underline{x}_i\right)}{\left(U_i^{(k)} - L_i^{(k)}\right)\left(z_i^{(k,p)} - x_i^{(k)}\right)^2}\right) \end{aligned} \tag{4.44}$$

The values of $\tau_j^{(k,p)}$, $j = 1, \ldots, m_f$, are defined analogously using $e_j(x)$ instead of $f(x)$. Within the inner iteration loop the gradients need not be adapted. This leads to the following algorithm

Algorithm 15. Globally Convergent Method of Moving Asymptotes

Step 0: Choose feasible starting point $x^{(0)} \in \mathbb{F}$. Compute $f\left(x^{(0)}\right), \nabla f\left(x^{(0)}\right)$, $e_j\left(x^{(0)}\right)$, $\nabla e_j\left(x^{(0)}\right)$, $j = 1, \ldots, m_f$. Set parameters $\xi = 0$, $-L_{min} = U_{max} = \infty$, $T_3 > 1$, $0 < T_4 < 1$, $\omega = 0.9$. Let $k := 0$ and $p := 0$.

Step 1: Determine $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, by Algorithm 11 and define $\tau_j^{(k,0)}$, $j = 0, \ldots, m_f$, according to (4.42). Define $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, according to (4.8) and (4.9).

Step 2: Define $f^{(k,p)}(x)$, $e_j^{(k,p)}(x)$, $j = 1, \ldots, m_f$, according to (4.37). Solve (4.38) for the corresponding outer iteration k and inner iteration p. Let $z^{(k,p)}$ be the optimal solution.

Step 3: If (4.39) and (4.40) are satisfied then goto Step 5.

Step 4: Update $\tau_j^{(k,p+1)}$, $j = 0, \ldots, m_f$, according to (4.43), set $p = p + 1$ and goto Step 2.

Step 5: Set $x^{(k+1)} := z^{(k,p)}$. *If* $x^{(k+1)} = x^{(k)}$, *then STOP.* $x^{(k+1)}$ *is the solution of (4.35).*

Step 6: Set $k = k + 1$, $p = 0$ *and compute* $f\left(x^{(k)}\right), \nabla f\left(x^{(k)}\right),\ e_j\left(x^{(k)}\right),\ \nabla e_j\left(x^{(k)}\right)$, $j = 1, \ldots, m_f$. *Goto Step 1.*

It can be shown that the inner iteration process, given by *Step 2* - *Step 5*, terminates after a finite number of iterations. Moreover, global convergence can be shown, see Svanberg [82]. Note that the functions might be evaluated at infeasible iterations, to test whether (4.39) and (4.40) holds.

5. A STRICTLY FEASIBLE SEQUENTIAL CONVEX PROGRAMMING METHOD

In many applications, some problem functions are only defined on a certain domain specified by other constraints. Since most nonlinear optimization methods cannot ensure feasibility during the solution process, these problems cannot be solved appropriately. Typical examples are the logarithmic or square root functions, e.g.,

$$\begin{aligned} c_1(x) &:= \log(e_1(x)), && (5.1) \\ c_2(x) &:= \sqrt{e_2(x)}, && (5.2) \end{aligned}$$

where $e_1(x)$ and $e_2(x)$ are nonlinear functions. To ensure that $c_1(x)$ and $c_2(x)$ can be evaluated, the constraints

$$\begin{aligned} e_1(x) &> 0, && (5.3) \\ e_2(x) &> 0, && (5.4) \end{aligned}$$

need to be satisfied. Note that we require $e_2(x) > 0$ such that $c_2(x)$ is continuously differentiable for all x satisfying (5.4).

We present an extended version of the SCP Algorithm 14 guaranteeing feasibility of a given subset of constraints in each iteration, which will be referred to as feasibility constraints $e_j(x) \leq 0,\ j = 1, \ldots, m_f$. The resulting method is called feasible sequential convex programming method (SCPF). 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, if all feasibility constraints $e_j(x) \leq 0,\ j = 1, \ldots, m_f$, are satisfied.

5.1 Feasible Sequential Convex Programming

In the sequel, we consider problem (4.17) extended by additional feasibility constraints $e_j(x),\ j = 1, \ldots, m_f$, that have to be satisfied whenever the objective function $f(x)$ and the constraints $c_j(x)$, $j = 1, \ldots, m_c$, need to be evaluated. We get

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j(x) = 0, & j = 1, \ldots, m_e \\ & c_j(x) \leq 0, & j = m_e + 1, \ldots, m_c \\ & e_j(x) \leq 0, & j = 1, \ldots, m_f \end{array} \qquad (5.5)$$

The objective function $f\left(x\right)$ and constraints $c_j\left(x\right),\ j=1,\dots,m_c$, are supposed to be at least continuously differentiable on the subset

$$F \quad := \quad \left\{x \in \mathbb{R}^n \ \mid e_j\left(x\right) \leq 0,\ j=1,\dots,m_f\right\}. \tag{5.6}$$

The functions $e_j\left(x\right),\ j=1,\dots,m_f$, must be convex and at least twice continuously differentiable on $\mathbb{R}^n$, see Ertel, Schittkowski and Zillober [22]. As a consequence, 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., $x^{(0)} \in F$, SCPF generates a sequence of convex subproblems, which are easy to solve due to their special structure. Moreover, they contain the nonlinear constraints $e_j\left(x\right),\ j=1,\dots,m_f$, to ensure their feasibility. The objective function $f\left(x\right)$ and the constraints $c_j\left(x\right),\ j=1,\dots,m_c$, are approximated by convex and separable functions according to (4.21), (4.22) and (4.5) yielding $f^{(k)}\left(x\right)$ and $c_j^{(k)}\left(x\right),\ j=1,\dots,m_c$, while the constraints $e_j\left(x\right),\ j=1,\dots,m_f$, are passed to the subproblem directly. We implicitly assume that the nonlinear functions $e_1\left(x\right),\dots,e_{m_f}\left(x\right)$ and their derivatives are much easier to evaluate than the functions and gradients of $f(x)$ and $c_1\left(x\right),\dots,c_{m_c}\left(x\right)$.

In each iteration k we obtain the following subproblem,

$$\begin{array}{lll}\min\limits_{x} & f^{(k)}\left(x\right) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j^{(k)}\left(x\right) = 0, & j=1,\dots,m_e \\ & c_j^{(k)}\left(x\right) \leq 0, & j=m_e+1,\dots,m_c \\ & e_j\left(x\right) \leq 0, & j=1,\dots,m_f \\ & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i=1,\dots,n \end{array} \tag{5.7}$$

The box constraints are defined by

$$\underline{x}_i^{(k)} := x_i^{(k)} - \omega\left(x_i^{(k)} - L_i^{(k)}\right),\ i=1,\dots,n \tag{5.8}$$

and

$$\overline{x}_i^{(k)} := x_i^{(k)} + \omega\left(U_i^{(k)} - x_i^{(k)}\right),\ i=1,\dots,n \tag{5.9}$$

where $\omega \in]0,1[$ is a suitable constant. The asymptotes $L_i^{(k)}$ and $U_i^{(k)},\ i=1,\dots,n$, are defined according to Algorithm 11 satisfying Definition 4.1. The solution $x^{(k)} \in \mathbb{R}^n$ of (5.7) lies in the set $F_X^{(k)}$

$$F \quad \supseteq \quad F_X^{(k)} \tag{5.10}$$

with

$$\begin{aligned} F_X^{(k)} \quad &:= \quad F \cap X^{(k)} & (5.11) \\ X^{(k)} \quad &:= \quad \left\{x \in \mathbb{R}^n \ \middle|\ \underline{x}^{(k)} \leq x \leq \overline{x}^{(k)}\right\}. & (5.12) \end{aligned}$$

To assure global convergence of the algorithm, we apply a line search procedure. The feasibility constraints have to be included in the differentiable augmented Lagrangian merit function (4.18). We get

$$\begin{aligned}\Phi_\rho\begin{pmatrix} x \\ y \end{pmatrix} \quad := \quad & f(x) + \sum_{j=1}^{m_e}\left((y_c)_j\, c_j(x) \; + \; \frac{(\rho_c)_j}{2} c_j^2(x)\right) \\ & + \sum_{j=m_e+1}^{m_c}\begin{cases} (y_c)_j\, c_j(x) \; + \; \frac{(\rho_c)_j}{2} c_j^2(x)\,, & \text{if } -\frac{(y_c)_j}{(\rho_c)_j} \le c_j(x) \\ -\frac{(y_c)_j^2}{2(\rho_c)_j}, & \text{otherwise} \end{cases} \\ & + \sum_{j=1}^{m_f}\begin{cases} (y_e)_j\, e_j(x) \; + \; \frac{(\rho_e)_j}{2} e_j^2(x)\,, & \text{if } -\frac{(y_e)_j}{(\rho_e)_j} \le e_j(x) \\ -\frac{(y_e)_j^2}{2(\rho_e)_j}, & \text{otherwise} \end{cases}\end{aligned} \tag{5.13}$$

for a given set of penalty parameters

$$\rho := \begin{pmatrix} \rho_c \\ \rho_e \end{pmatrix} \tag{5.14}$$

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

$$y := \begin{pmatrix} y_c \\ y_e \end{pmatrix} \tag{5.15}$$

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 penalty parameters are updated according to Algorithm 13, see Section 4.2.
The feasible SCP algorithm is summarized as follows:

Algorithm 16. Feasible Sequential Convex Programming

Step 0: Choose feasible starting point $x^{(0)} \in F$. Set parameters $\xi > 0$, $L_{min} < U_{max}$, $T_2 > 1$, $0 < T_1 < 1$, $\omega \in]0;1[$, $r \in (0,1)$, $\beta \in (0,1)$, $\tau > 0$, $\kappa_2 > \kappa_1 > 1$ 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 $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, by Algorithm 11. Let $f^{(k)}(x)$, $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, be defined by (4.21), (4.22) and (4.5). Define $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, according to (4.8) and (4.9). Formulate (4.26) for the corresponding iteration k.

Step 2: Solve (5.7). Let $z^{(k)}$ be the optimal solution of subproblem (5.7) 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 (5.5).

Step 4: Let $d^{(k)} := \begin{pmatrix} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{pmatrix}, \delta^{(k)} := \left\|z^{(k)} - x^{(k)}\right\|_2$ and $\eta^{(k)}$ as defined in (4.31). Let $i = 0$ and $\rho^{(k,0)} := \rho^{(k-1)}$.

Step 5: Compute $\Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}, \nabla\Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}, \nabla\Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}$.

Step 6: If $\nabla\Phi_{\rho^{(k,i)}} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} > -\frac{\eta^{(k)} \left(\delta^{(k)}\right)^2}{2}$, update penalty parameters according to Algorithm 13. Let $i = i + 1$ and goto Step 5. Otherwise, let $\rho^{(k)} := \rho^{(k,i)}$, $i = 0$ and $\sigma^{(k,0)} := 1$.

Step 7: Compute $f\left(x^{(k)} + \sigma^{(k,i)}\left(z^{(k)} - x^{(k)}\right)\right), c_j\left(x^{(k)} + \sigma^{(k,i)}\left(z^{(k)} - x^{(k)}\right)\right)$, $j = 1, \ldots, m_c$, $e_j\left(x^{(k)} + \sigma^{(k,i)}\left(z^{(k)} - x^{(k)}\right)\right)$, $j = 1, \ldots, m_f$, and $\Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)}\right)$. If (4.28) 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 8: 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 9: 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 (5.7) become inconsistent. In this case, the subproblem is extended by additional variables, see Section 7.1.3 for details. By assumption, the subset F described by the feasibility constraints is convex. This is important for the line search procedure as strict feasibility cannot be ensured on the whole interval $\left[z^{(k)}, x^{(k)}\right]$ otherwise. Moreover, subproblem (5.7) possesses an unique solution, if F is convex.

The feasibility constraints $e_j(x)$, $j = 1, \ldots, m_f$, have to be satisfied whenever the objective $f(x)$ and the constraints $c_j(x)$, $j = 1, \ldots, m_c$, are evaluated. Due to the formulation of the approximations (4.21), (4.22) and (4.5) they may be violated within the solution process of the subproblem.

In contrast to the feasible direction SQP algorithms presented in Chapter 3, stepsize $\sigma^{(k)} = 1$ ensures feasibility with respect to the feasibility constraints. In addition, the search direction need not be adapted to ensure a feasible iteration point $x^{(k+1)}$. The globally convergent MMA method, presented in Section 4.3, computes a sequence of feasible iteration points. Compared to SCPF, function evaluations are still necessary at infeasible points.

A convergence proof of the feasible sequential convex programming method is given in Section 5.2. The implementation of SCPF is based on SCPIP30.f, see Zillober [103]. It is extended by feasibility constraints that are handled separately during the optimization process. In addition, the predictor corrector interior point method, which is used in SCPIP30.f to solve the subproblem (4.26), is replaced by the general nonlinear programming solver IPOPT, see Wächter and Biegler [90]. If no feasibility constraints are present, i.e., $m_f = 0$, the iteration sequences of SCPF and SCP are identical.

5.2 Global Convergence

5.2.1 Notation and Analysis

We proceed from optimization problem (5.5) and omit equality constraints to ease the notation, i.e., $m_e = 0$. We get

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j(x) \leq 0, & j = 1, \ldots, m_c \\ & e_j(x) \leq 0, & j = 1, \ldots, m_f \end{array} \tag{5.16}$$

where $f\left(x\right)$ and $c_j\left(x\right),\ j = 1, \ldots, m_c$, are supposed to be at least continuously differentiable on the subset

$$F \ := \ \left\{x \in \mathbb{R}^n \mid e_j\left(x\right) \leq 0,\ j = 1, \ldots, m_f\right\}, \tag{5.17}$$

while $e_j\left(x\right),\ j = 1, \ldots, m_f$, are convex and at least twice continuously differentiable on $\mathbb{R}^n$. Moreover, we assume:

Assumption 1. *The feasible region described by the feasibility constraints is nonempty, i.e.,*

$$F \ := \ \{x \in \mathbb{R}^n \mid e_j(x) \leq 0,\ j = 1, \ldots, m_f\} \ \neq \emptyset. \tag{5.18}$$

and compact.

As a consequence, the functions $f(x),\ c_j(x),\ j = 1, \ldots, m_c$, and $e_j(x),\ j = 1, \ldots, m_f$, and the corresponding partial derivatives are bounded on F, which is important to show that the corresponding dual variables are bounded.

Note that we omit box constraints throughout our theoretical analysis to simplify the notation. They are introduced again for the implementation of the algorithms to guarantee that the variables retain in a compact set described by box and feasibility constraints.

By approximating the inequalities $c_j(x) \leq 0,\ j = 1, \dots, m_c$, according to (4.5) and the objective function $f(x)$ according to (4.21) we obtain the subproblem (5.7), i.e.,

$$\begin{array}{lll} \min\limits_{x} & f^{(k)}(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j^{(k)}(x) \leq 0, & j = 1, \dots, m_c \\ & e_j(x) \leq 0, & j = 1, \dots, m_f \\ & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i = 1, \dots, n \end{array} \tag{5.19}$$

The box constraints are chosen in the following way

$$\underline{x}_i^{(k)} := x_i^{(k)} - \omega\left(x_i^{(k)} - L_i^{(k)}\right),\ i = 1, \dots, n \tag{5.20}$$

$$\underline{x}^{(k)} := \left(\underline{x}_1^{(k)}, \dots, \underline{x}_n^{(k)}\right)^T \tag{5.21}$$

$$\overline{x}_i^{(k)} := x_i^{(k)} + \omega\left(U_i^{(k)} - x_i^{(k)}\right),\ i = 1, \dots, n \tag{5.22}$$

$$\overline{x}^{(k)} := \left(\overline{x}_1^{(k)}, \dots, \overline{x}_n^{(k)}\right)^T \tag{5.23}$$

with $\omega \in]0;1[$ constant. To ease the notation, the box constraints in iteration k are considered as linear inequality constraints defined by

$$b_i^{(k)}(x) := x_i - \overline{x}_i^{(k)} \leq 0, \quad i = 1, \dots, n, \tag{5.24}$$

$$b_{n+i}^{(k)}(x) := \underline{x}_i^{(k)} - x_i \leq 0, \quad i = 1, \dots, n. \tag{5.25}$$

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

$$A_{u^{(k)}}(x) := \left(\nabla b_1^{(k)}(x), \dots, \nabla b_n^{(k)}(x)\right) \in \mathbb{R}^{n \times n}, \tag{5.26}$$

$$A_{l^{(k)}}(x) := \left(\nabla b_{n+1}^{(k)}(x), \dots, \nabla b_{2n}^{(k)}(x)\right) \in \mathbb{R}^{n \times n}. \tag{5.27}$$

It is easy to see, that

$$A_{u^{(k)}}(x) = -A_{l^{(k)}}(x) = I, \tag{5.28}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix.

We proceed from a sequence of feasible asymptotes as specified in Definition 4.1, i.e.,

$$L_{\min} \leq L_i^{(k)} \leq x_i^{(k)} - \xi, \qquad \forall i = 1, \dots, n \tag{5.29}$$

$$U_{\max} \geq U_i^{(k)} \geq x_i^{(k)} + \xi, \qquad \forall i = 1, \dots, n \tag{5.30}$$

with $\xi > 0$. We denote the primal solution of subproblem (5.19) 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

$$\Delta x^{(k)} := z^{(k)} - x^{(k)}. \tag{5.31}$$

Before proving global convergence we subsume the following definitions.

$$\begin{aligned}
c(x) &:= \left(c_1(x), \ldots, c_{m_c}(x)\right)^T && (5.32)\\
e(x) &:= \left(e_1(x), \ldots, e_{m_f}(x)\right)^T && (5.33)\\
b_u(x) &:= \left(b_1(x), \ldots, b_n(x)\right)^T && (5.34)\\
b_l(x) &:= \left(b_{n+1}(x), \ldots, b_{2n}(x)\right)^T && (5.35)\\
y_c &:= \left((y_c)_1, \ldots, (y_c)_{m_c}\right)^T && (5.36)\\
y_e &:= \left((y_e)_1, \ldots, (y_e)_{m_f}\right)^T && (5.37)\\
y &:= \begin{pmatrix} y_c \\ y_e \end{pmatrix} && (5.38)\\
v_c &:= \left((v_c)_1, \ldots, (v_c)_{m_c}\right)^T && (5.39)\\
v_e &:= \left((v_e)_1, \ldots, (v_e)_{m_f}\right)^T && (5.40)\\
v &:= \begin{pmatrix} v_c \\ v_e \end{pmatrix} && (5.41)\\
v_u &:= \left((v_u)_1, \ldots, (v_u)_n\right)^T && (5.42)\\
v_l &:= \left((v_l)_1, \ldots, (v_l)_n\right)^T && (5.43)\\
v_b &:= \begin{pmatrix} v_u \\ v_l \end{pmatrix} && (5.44)\\
\rho_c &:= \left((\rho_c)_1, \ldots, (\rho_c)_{m_c}\right)^T && (5.45)\\
\Gamma_c &:= \operatorname{diag}\left\{(\rho_c)_1, \ldots, (\rho_c)_{m_c}\right\} && (5.46)\\
\rho_e &:= \left((\rho_e)_1, \ldots, (\rho_e)_{m_f}\right)^T && (5.47)\\
\Gamma_e &:= \operatorname{diag}\left\{(\rho_e)_1, \ldots, (\rho_e)_{m_f}\right\} && (5.48)\\
\rho &:= \begin{pmatrix} \rho_c \\ \rho_e \end{pmatrix} && (5.49)\\
J_c(x) &:= \left\{ j = 1, \ldots, m_c \,\middle|\, -\frac{(y_c)_j}{(\rho_c)_j} \leq c_j(x) \right\} && (5.50)\\
\overline{J}_c(x) &:= \left\{ j = 1, \ldots, m_c \,\middle|\, -\frac{(y_c)_j}{(\rho_c)_j} > c_j(x) \right\} && (5.51)\\
J_e(x) &:= \left\{ j = 1, \ldots, m_f \,\middle|\, -\frac{(y_e)_j}{(\rho_e)_j} \leq e_j(x) \right\} && (5.52)\\
\overline{J}_e(x) &:= \left\{ j = 1, \ldots, m_f \,\middle|\, -\frac{(y_e)_j}{(\rho_e)_j} > e_j(x) \right\} && (5.53)
\end{aligned}$$

$$
\begin{aligned}
A_c(x) &:= \left(\nabla c_1(x), \ldots, \nabla c_{m_c}(x)\right) \in \mathbb{R}^{n \times m_c} && (5.54)\\
A_e(x) &:= \left(\nabla e_1(x), \ldots, \nabla e_{m_f}(x)\right) \in \mathbb{R}^{n \times m_f} && (5.55)\\
A(x) &:= \left(A_c(x), A_e(x)\right) \in \mathbb{R}^{n \times m_c + m_f} && (5.56)
\end{aligned}
$$

$$\overline{y}_c := \left((\overline{y}_c)_1, \ldots, (\overline{y}_c)_{m_c}\right)^T \quad \text{with} \quad (\overline{y}_c)_j := \begin{cases} (y_c)_j\,, & \text{if } j \in J_c(x) \\ 0, & \text{otherwise} \end{cases} \tag{5.57}$$

$$\overline{y}_e := \left((\overline{y}_e)_1, \ldots, (\overline{y}_e)_{m_f}\right)^T \quad \text{with} \quad (\overline{y}_e)_j := \begin{cases} (y_e)_j\,, & \text{if } j \in J_e(x) \\ 0, & \text{otherwise} \end{cases} \tag{5.58}$$

$$\overline{v}_c := \left((\overline{v}_c)_1, \ldots, (\overline{v}_c)_{m_c}\right)^T \quad \text{with} \quad (\overline{v}_c)_j := \begin{cases} (v_c)_j\,, & \text{if } j \in J_c(x) \\ 0, & \text{otherwise} \end{cases} \tag{5.59}$$

$$\overline{v}_e := \left((\overline{v}_e)_1, \ldots, (\overline{v}_e)_{m_f}\right)^T \quad \text{with} \quad (\overline{v}_e)_j := \begin{cases} (v_e)_j\,, & \text{if } j \in J_e(x) \\ 0, & \text{otherwise} \end{cases} \tag{5.60}$$

$$\overline{c}(x) := \left(\overline{c}_1(x), \ldots, \overline{c}_{m_c}(x)\right)^T \quad \text{with} \quad \overline{c}_j(x) := \begin{cases} c_j(x), & \text{if } j \in J_c(x) \\ 0, & \text{otherwise} \end{cases} \tag{5.61}$$

$$\overline{e}(x) := \left(\overline{e}_1(x), \ldots, \overline{e}_{m_f}(x)\right)^T \quad \text{with} \quad \overline{e}_j(x) := \begin{cases} e_j(x), & \text{if } j \in J_e(x) \\ 0, & \text{otherwise} \end{cases} \tag{5.62}$$

$$\widehat{c}(x) := \left(\widehat{c}_1(x), \ldots, \widehat{c}_{m_c}(x)\right)^T \quad \text{with} \quad \widehat{c}_j(x) := \begin{cases} c_j(x), & \text{if } j \in J_c(x) \\ -\frac{(y_c)_j}{(\rho_c)_j}, & \text{otherwise} \end{cases} \tag{5.63}$$

$$\widehat{e}(x) := \left(\widehat{e}_1(x), \ldots, \widehat{e}_{m_f}(x)\right)^T \quad \text{with} \quad \widehat{e}_j(x) := \begin{cases} e_j(x), & \text{if } j \in J_e(x) \\ -\frac{(y_e)_j}{(\rho_e)_j}, & \text{otherwise} \end{cases} \tag{5.64}$$

$$\widetilde{c}(x) := \left(\widetilde{c}_1(x), \ldots, \widetilde{c}_{m_c}(x)\right)^T \quad \text{with} \quad \widetilde{c}_j(x) := \begin{cases} c_j(x), & \text{if } j \in J_c(x) \\ -\frac{(y_c)_j}{2(\rho_c)_j}, & \text{otherwise} \end{cases} \tag{5.65}$$

$$\widetilde{e}(x) := \left(\widetilde{e}_1(x), \ldots, \widetilde{e}_{m_f}(x)\right)^T \quad \text{with} \quad \widetilde{e}_j(x) := \begin{cases} e_j(x), & \text{if } j \in J_e(x) \\ -\frac{(y_e)_j}{2(\rho_e)_j}, & \text{otherwise} \end{cases} \tag{5.66}$$

If required, we add the upper index $^{(k)}$ to denote the k-th iteration step. To ease the notation we define

$$
\begin{aligned}
J_c^{(k)} &:= J_c(x^{(k)}) && (5.67)\\
\overline{J}_c^{(k)} &:= \overline{J}_c(x^{(k)}) && (5.68)\\
J_e^{(k)} &:= J_e(x^{(k)}) && (5.69)\\
\overline{J}_e^{(k)} &:= \overline{J}_e(x^{(k)}) && (5.70)
\end{aligned}
$$

5.2.2 Preliminary Results

The convergence proof for Algorithm 16 is an extension of the convergence proof of Algorithm 14, see Zillober [97, 98] and [102]. For the nonlinear optimization problem (5.16), the Lagrangian function (2.8) can be written as

$$\begin{aligned} L\left(x,y\right) &= f(x) + y_c^T c(x) + y_e^T e(x) && (5.71)\\ \nabla_x L\left(x,y\right) &= \nabla f(x) + A_c(x) y_c + A_e(x) y_e && (5.72)\end{aligned}$$

and, respectively, the augmented Lagrangian (5.13) is given by

$$\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} = f(x) + y_c^T \widetilde{c}(x) + \frac{1}{2}\rho_c^T \overline{c}^2(x) + y_e^T \widetilde{e}(x) + \frac{1}{2}\rho_e^T \overline{e}^2(x) \qquad (5.73)$$

with the corresponding gradient

$$\nabla \Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \nabla f(x) + A_c(x)\left(\overline{y}_c + \Gamma_c \overline{c}(x)\right) + A_e(x)\left(\overline{y}_e + \Gamma_e \overline{e}(x)\right) \\ \widehat{c}(x) \\ \widehat{e}(x) \end{pmatrix} \qquad (5.74)$$

Moreover, we consider the first order necessary optimality conditions (KKT, see Definition 6 and (2.12) - (2.19)) for the subproblem. We denote the Lagrangian function (2.8) of subproblem (5.19) in iteration k by $L^{(k)}(z,v)$. As $\left(z^{(k)}, v^{(k)}\right)$ is a KKT point of subproblem (5.19) and $A_u = -A_l = I$, we get

$$\begin{aligned} \nabla_x L^{(k)}\left(z^{(k)}, v^{(k)}\right) &= \nabla f^{(k)}\left(z^{(k)}\right) + A_{c^{(k)}}\left(z^{(k)}\right) v_c^{(k)} \\ &\quad + A_e\left(z^{(k)}\right) v_e^{(k)} + v_u^{(k)} - v_l^{(k)} = 0 && (5.75)\\ \left(v_c^{(k)}\right)_j c_j^{(k)}\left(z^{(k)}\right) &= 0, \qquad j = 1, \ldots, m_c && (5.76)\\ \left(v_e^{(k)}\right)_j e_j\left(z^{(k)}\right) &= 0, \qquad j = 1, \ldots, m_f && (5.77)\\ \left(v_b^{(k)}\right)_j b_j^{(k)}\left(z^{(k)}\right) &= 0, \qquad j = 1, \ldots, 2n && (5.78)\\ c_j^{(k)}\left(z^{(k)}\right) &\leq 0, \qquad j = 1, \ldots, m_c && (5.79)\\ e_j\left(z^{(k)}\right) &\leq 0, \qquad j = 1, \ldots, m_f && (5.80)\\ b_j^{(k)}\left(z^{(k)}\right) &\leq 0, \qquad j = 1, \ldots, 2n && (5.81)\\ \left(v_c^{(k)}\right)_j &\geq 0, \qquad j = 1, \ldots, m_c && (5.82)\\ \left(v_e^{(k)}\right)_j &\geq 0, \qquad j = 1, \ldots, m_f && (5.83)\\ \left(v_b^{(k)}\right)_j &\geq 0, \qquad j = 1, \ldots, 2n && (5.84)\end{aligned}$$

Applying the Taylor approximation with residual $R_{f^{(k)}}(x)$, $R_{c_j^{(k)}}(x)$ and $R_{e_j}(x)$ for the corresponding functions, we obtain the following equations for the objective $f(x)$:

$$\begin{aligned} f^{(k)}(x) &= f\left(x^{(k)}\right) + \underbrace{\nabla f^{(k)}\left(x^{(k)}\right)^T}_{=\nabla f\left(x^{(k)}\right)^T}\left(x - x^{(k)}\right) + R_{f^{(k)}}(x) && (5.85)\\ \nabla f^{(k)}(x) &= \nabla f\left(x^{(k)}\right) + \nabla R_{f^{(k)}}(x) && (5.86)\end{aligned}$$

and respectively for the constraints $c_j(x),\ j=1,\ldots,m_c$, and $e_j(x),\ j=1,\ldots,m_f$,

$$\begin{aligned}
c_j^{(k)}(x) &= c_j\left(x^{(k)}\right) + \underbrace{\nabla c_j^{(k)}\left(x^{(k)}\right)^T}_{=\nabla c_j\left(x^{(k)}\right)^T}\left(x - x^{(k)}\right) + R_{c_j^{(k)}}(x) && (5.87)\\
\nabla c_j^{(k)}(x) &= \nabla c_j\left(x^{(k)}\right) + \nabla R_{c_j^{(k)}}(x) && (5.88)\\
e_j(x) &= e_j\left(x^{(k)}\right) + \nabla e_j\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + R_{e_j}(x) && (5.89)\\
\nabla e_j(x) &= \nabla e_j\left(x^{(k)}\right) + \nabla R_{e_j}(x) && (5.90)
\end{aligned}$$

Moreover, we can determine the gradients of each function $c_j(x),\ j=1,\ldots,m_c$, and $e_j(x),\ j=1,\ldots,m_f$, at iterate $z^{(k)}$ using (5.87) and (5.89) by

$$\begin{aligned}
c_j^{(k)}\left(z^{(k)}\right) &= c_j\left(x^{(k)}\right) + \nabla c_j\left(x^{(k)}\right)^T \Delta x^{(k)} + R_{c_j^{(k)}}\left(z^{(k)}\right) && (5.91)\\
\nabla c_j\left(x^{(k)}\right)^T \Delta x^{(k)} &= c_j^{(k)}\left(z^{(k)}\right) - c_j\left(x^{(k)}\right) - R_{c_j^{(k)}}\left(z^{(k)}\right) && (5.92)
\end{aligned}$$

with $\Delta x^{(k)} = z^{(k)} - x^{(k)}$. Analogue we obtain

$$\nabla e_j\left(x^{(k)}\right)^T \Delta x^{(k)} = e_j\left(z^{(k)}\right) - e_j\left(x^{(k)}\right) - R_{e_j}\left(z^{(k)}\right). \tag{5.93}$$

As the approximations $c_j^{(k)}(x)$ of $c_j(x),\ j=1,\ldots,m_c$, are convex, the corresponding $R_{c_j^{(k)}}(x)$ is nonnegative. The same holds for $R_{e_j}(x),\ j=1,\ldots,m_f$, respectively, as the functions $e_j(x),\ j=1,\ldots,m_f$, are convex by assumption. We consider the Taylor series of $e_j(x),\ j=1,\ldots,m_f$, evaluated in $z^{(k)}$ for the current iterate $x^{(k)}$.

$$\begin{aligned}
e_j\left(x^{(k)}\right) &= e_j\left(z^{(k)}\right) + \nabla e_j\left(z^{(k)}\right)^T\left(x^{(k)} - z^{(k)}\right) + \underbrace{R_{e_j}\left(x^{(k)}\right)}_{\geq 0}\\
&\geq e_j\left(z^{(k)}\right) + \nabla e_j\left(z^{(k)}\right)^T\left(x^{(k)} - z^{(k)}\right)\\
&= e_j\left(z^{(k)}\right) - \nabla e_j\left(z^{(k)}\right)^T\left(z^{(k)} - x^{(k)}\right)\\
e_j\left(x^{(k)}\right) - e_j\left(z^{(k)}\right) &\geq -\nabla e_j\left(z^{(k)}\right)^T \Delta x^{(k)} && (5.94)
\end{aligned}$$

We define

$$\begin{aligned}
R_{c^{(k)}}(x) &:= \left(R_{c_1^{(k)}}(x), \ldots, R_{c_{m_c}^{(k)}}(x)\right)^T \in \mathbb{R}^{m_c} && (5.95)\\
\nabla R_{c^{(k)}}(x) &:= \left(\nabla R_{c_1^{(k)}}(x), \ldots, \nabla R_{c_{m_c}^{(k)}}(x)\right) \in \mathbb{R}^{n\times m_c} && (5.96)
\end{aligned}$$

and

$$\begin{aligned}
R_e(x) &:= \left(R_{e_1}(x), \ldots, R_{e_{m_f}}(x)\right)^T \in \mathbb{R}^{m_f} && (5.97)\\
\nabla R_e(x) &:= \left(\nabla R_{e_1}(x), \ldots, \nabla R_{e_{m_f}}(x)\right) \in \mathbb{R}^{n\times m_f} && (5.98)
\end{aligned}$$

respectively. The exact values of $R_{c_j^{(k)}}(x)$, $\nabla R_{c_j^{(k)}}(x)$, $j = 1, \ldots, m_c$, and $R_{f^{(k)}}(x)$, $\nabla R_{f^{(k)}}(x)$ can be obtained according to the following lemma, see Zillober [102] Section 2.3. The lemma is used to prove Lemma 5.4, which is an essential part to show that a sufficient descent with respect to the augmented Lagrangian is obtained, see Theorem 5.2.

Lemma 5.1. *Let $f^{(k)}(x)$ and $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, be convex approximations (4.21) and (4.5) formulated in $x^{(k)} \in \mathbb{R}^n$ by Algorithm 16. Moreover, the corresponding asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, $i = 1, \ldots, n$, be feasible according to Definition 4.1. Then the following equations hold for all $k \geq 0$*

$$\begin{aligned} R_{f^{(k)}}(x) &= \sum_{I_+^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} \right] \\ &\quad - \sum_{I_-^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \right] \end{aligned} \tag{5.99}$$

$$\begin{aligned} R_{c_j^{(k)}}(x) &= \sum_{I_+^{(j,k)}} \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} \right] \\ &\quad - \sum_{I_-^{(j,k)}} \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \right] \end{aligned} \tag{5.100}$$

$$\frac{\partial R_{f^{(k)}}(x)}{\partial x_i} = \begin{cases} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)\left(2U_i^{(k)} - x_i - x_i^{(k)}\right)}{\left(U_i^{(k)} - x_i\right)^2} \right], & \textit{if } i \in I_+^{(k)} \\ -\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)\left(-2L_i^{(k)} + x_i + x_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2} \right], & \textit{if } i \in I_-^{(k)} \end{cases} \tag{5.101}$$

$$\frac{\partial R_{c_j^{(k)}}(x)}{\partial x_i} = \begin{cases} \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left[\frac{\left(x_i - x_i^{(k)}\right)\left(2U_i^{(k)} - x_i - x_i^{(k)}\right)}{\left(U_i^{(k)} - x_i\right)^2} \right], & \textit{if } i \in I_+^{(j,k)} \\ -\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left[\frac{\left(x_i - x_i^{(k)}\right)\left(-2L_i^{(k)} + x_i + x_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2} \right], & \textit{if } i \in I_-^{(j,k)} \end{cases} \tag{5.102}$$

where $\tau > 0$ and $I_+^{(k)}$, $I_-^{(k)}$,$I_+^{(j,k)}$ and $I_-^{(j,k)}$ are defined by (4.2), (4.3), (4.6) and (4.7).

Proof. We obtain the following condition by exploiting the equality of the Taylor series (5.85) and the definition of the approximation (4.21). We get:

$$
\begin{aligned}
f^{(k)}(x) &\overset{(5.85)}{=} f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + R_{f^{(k)}}(x) \\
f^{(k)}(x) &\overset{(4.21)}{=} f\left(x^{(k)}\right) + \sum_{I_+^{(k)}}\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left(U_i^{(k)} - x_i^{(k)}\right)^2\left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}}\right)\right] \\
&\quad - \sum_{I_-^{(k)}}\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left(x_i^{(k)} - L_i^{(k)}\right)^2\left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}}\right)\right] \\
&\quad + \sum_{I_+^{(k)}}\tau\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} + \sum_{I_-^{(k)}}\tau\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}}
\end{aligned}
$$

This leads to

$$
\begin{aligned}
&R_{f^{(k)}}(x) \\
&= -f\left(x^{(k)}\right) - \nabla f\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + f\left(x^{(k)}\right) \\
&\quad + \sum_{I_+^{(k)}}\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left(U_i^{(k)} - x_i^{(k)}\right)^2\left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}}\right)\right] \\
&\quad - \sum_{I_-^{(k)}}\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left(x_i^{(k)} - L_i^{(k)}\right)^2\left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}}\right)\right] \\
&\quad + \sum_{I_+^{(k)}}\tau\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} + \sum_{I_-^{(k)}}\tau\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \\
&= \sum_{I_+^{(k)}}\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left[\frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i}\underbrace{-\left(U_i^{(k)} - x_i^{(k)}\right) - \left(x_i - x_i^{(k)}\right)}_{=-U_i^{(k)}+2x_i^{(k)}-x_i}\right] \\
&\quad - \sum_{I_-^{(k)}}\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left[\frac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{x_i - L_i^{(k)}}\underbrace{-\left(x_i^{(k)} - L_i^{(k)}\right) + \left(x_i - x_i^{(k)}\right)}_{=-2x_i^{(k)}+x_i+L_i^{(k)}}\right] \\
&\quad + \sum_{I_+^{(k)}}\tau\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} + \sum_{I_-^{(k)}}\tau\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}}
\end{aligned}
$$

$$
\begin{aligned}
& R_{f^{(k)}}(x) \\
= \; & \sum_{I_+^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left[\frac{\left(U_i^{(k)}-x_i^{(k)}\right)^2}{U_i^{(k)}-x_i}-\frac{\left(U_i^{(k)}-2x_i^{(k)}+x_i\right)\left(U_i^{(k)}-x_i\right)}{U_i^{(k)}-x_i}\right] \\
& -\sum_{I_-^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left[\frac{\left(x_i^{(k)}-L_i^{(k)}\right)^2}{x_i-L_i^{(k)}}-\frac{\left(2x_i^{(k)}-x_i-L_i^{(k)}\right)\left(x_i-L_i^{(k)}\right)}{x_i-L_i^{(k)}}\right] \\
& +\sum_{I_+^{(k)}} \tau\frac{\left(x_i-x_i^{(k)}\right)^2}{U_i^{(k)}-x_i}+\sum_{I_-^{(k)}} \tau\frac{\left(x_i-x_i^{(k)}\right)^2}{x_i-L_i^{(k)}} \\
= \; & \sum_{I_+^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \\
& \left[\frac{\left(U_i^{(k)}\right)^2+\left(x_i^{(k)}\right)^2-2U_i^{(k)}x_i^{(k)}-\left(U_i^{(k)}\right)^2+2U_i^{(k)}x_i^{(k)}-2x_ix_i^{(k)}+x_i^2}{U_i^{(k)}-x_i}\right] \\
& -\sum_{I_-^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \\
& \left[\frac{\left(L_i^{(k)}\right)^2+\left(x_i^{(k)}\right)^2-2L_i^{(k)}x_i^{(k)}-\left(L_i^{(k)}\right)^2+2L_i^{(k)}x_i^{(k)}-2x_ix_i^{(k)}+x_i^2}{x_i-L_i^{(k)}}\right] \\
& +\sum_{I_+^{(k)}} \tau\frac{\left(x_i-x_i^{(k)}\right)^2}{U_i^{(k)}-x_i}+\sum_{I_-^{(k)}} \tau\frac{\left(x_i-x_i^{(k)}\right)^2}{x_i-L_i^{(k)}} \\
= \; & \sum_{I_+^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left[\frac{\left(x_i^{(k)}\right)^2-2x_ix_i^{(k)}+x_i^2}{U_i^{(k)}-x_i}\right]+\sum_{I_+^{(k)}} \tau\frac{\left(x_i-x_i^{(k)}\right)^2}{U_i^{(k)}-x_i} \\
& -\sum_{I_-^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\left[\frac{\left(x_i^{(k)}\right)^2-2x_ix_i^{(k)}+x_i^2}{x_i-L_i^{(k)}}\right]+\sum_{I_-^{(k)}} \tau\frac{\left(x_i-x_i^{(k)}\right)^2}{x_i-L_i^{(k)}}
\end{aligned}
$$

$$
\begin{aligned}
R_{f^{(k)}}(x) &= \sum_{I_+^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} \right] + \sum_{I_+^{(k)}} \tau \frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} \\
&\quad - \sum_{I_-^{(k)}} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \right] + \sum_{I_-^{(k)}} \tau \frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \\
&= \sum_{I_+^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{U_i^{(k)} - x_i} \right] \\
&\quad - \sum_{I_-^{(k)}} \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)^2}{x_i - L_i^{(k)}} \right]
\end{aligned}
$$

$R_{c_j^{(k)}}(x)$ can be computed analogously for each $c_j(x),\ j = 1, \ldots, m_c$, by setting $\tau = 0$. The derivatives of $R_{f^{(k)}}(x)$ can be determined easily, since $\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i}, U_i^{(k)}, L_i^{(k)}, x_i^{(k)}$ and τ are constant. We get for an arbitrary $i \in I_+^{(k)}$

$$
\begin{aligned}
\frac{\partial R_{f^{(k)}}(x)}{\partial x_i} &= \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau \right] \left[\frac{2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right) + \left(x_i - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^2} \right] \\
&= \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)\left(2U_i^{(k)} - 2x_i + x_i - x_i^{(k)}\right)}{\left(U_i^{(k)} - x_i\right)^2} \right] \\
&= \left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)\left(2U_i^{(k)} - x_i - x_i^{(k)}\right)}{\left(U_i^{(k)} - x_i\right)^2} \right]
\end{aligned}
$$

and respectively for $i \in I_-^{(k)}$

$$
\begin{aligned}
\frac{\partial R_{f^{(k)}}(x)}{\partial x_i} &= -\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau \right] \left[\frac{2\left(x_i - x_i^{(k)}\right)\left(x_i - L_i^{(k)}\right) - \left(x_i - x_i^{(k)}\right)^2}{\left(x_i - L_i^{(k)}\right)^2} \right] \\
&= -\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)\left(2x_i - 2L_i^{(k)} - x_i + x_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2} \right] \\
&= -\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau \right] \left[\frac{\left(x_i - x_i^{(k)}\right)\left(-2L_i^{(k)} + x_i + x_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2} \right]
\end{aligned}
$$

In total we get:

$$\frac{\partial R_{f^{(k)}}(x)}{\partial x_i} = \begin{cases} \left[\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} + \tau\right] \left[\dfrac{\left(x_i - x_i^{(k)}\right)\left(2U_i^{(k)} - x_i - x_i^{(k)}\right)}{\left(U_i^{(k)} - x_i\right)^2}\right], & \text{if } i \in I_+^{(k)} \\ -\left[\dfrac{\partial f\left(x^{(k)}\right)}{\partial x_i} - \tau\right] \left[\dfrac{\left(x_i - x_i^{(k)}\right)\left(-2L_i^{(k)} + x_i + x_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2}\right], & \text{if } i \in I_-^{(k)} \end{cases}$$

The partial derivatives of $R_{c_j^{(k)}}(x)$, $j = 1, \ldots, m_c$, can be computed analogously by setting $\tau = 0$. □

The following lemma gives an important relation between the Lagrangian multipliers of the box constraints $v_u^{(k)}$ and $v_l^{(k)}$ and the search direction $\Delta x^{(k)}$. The results are needed to prove Lemma 5.3, which is used to show that a sufficient descent with respect to the augmented Lagrangian is obtained, see Theorem 5.2.

Lemma 5.2. *Let the sequences $\left\{x^{(k)}, y^{(k)}\right\}$ and $\left\{z^{(k)}, v^{(k)}\right\}$ be computed by Algorithm 16. The box constraints of subproblem (5.19) in iteration k are given by $b_u^{(k)}(x)$ and $b_l^{(k)}(x)$ and the corresponding Lagrangian multipliers are denoted by $v_u^{(k)} \in \mathbb{R}^n$ and $v_l^{(k)} \in \mathbb{R}^n$, respectively. Let $\Delta x^{(k)} := z^{(k)} - x^{(k)}$ be the primal search direction, where $z^{(k)}$ is the primal solution of subproblem (5.19) formulated in the current iterate $x^{(k)}$. Then the following inequalities*

$$\left(v_u^{(k)}\right)^T \Delta x^{(k)} \geq 0 \tag{5.103}$$

$$\left(v_l^{(k)}\right)^T \Delta x^{(k)} \leq 0 \tag{5.104}$$

hold for all $k \geq 0$.

Proof. We consider the definition of $\underline{x}_i^{(k)}$, $i = 1, \ldots, n$, and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, see (5.20) and (5.22). We start with the proof of (5.103)

$$\begin{aligned} \left(v_u^{(k)}\right)^T \underbrace{\Delta x^{(k)}}_{=z^{(k)}-x^{(k)}} &= \left(v_u^{(k)}\right)^T \left(z^{(k)} - x^{(k)}\right) \\ &= \left(v_u^{(k)}\right)^T z^{(k)} \underbrace{- \left(v_u^{(k)}\right)^T \overline{x}^{(k)} + \left(v_u^{(k)}\right)^T \overline{x}^{(k)}}_{=0} - \left(v_u^{(k)}\right)^T x^{(k)} \\ &= \underbrace{\left(v_u^{(k)}\right)^T z^{(k)} - \left(v_u^{(k)}\right)^T \overline{x}^{(k)}}_{=\left(v_u^{(k)}\right)^T b_u^{(k)}(z^{(k)})=0,\ (5.78)} + \left(v_u^{(k)}\right)^T \overline{x}^{(k)} - \left(v_u^{(k)}\right)^T x^{(k)} \\ &= \left(v_u^{(k)}\right)^T \overline{x}^{(k)} - \left(v_u^{(k)}\right)^T x^{(k)} \\ &= \left(v_u^{(k)}\right)^T \underbrace{\left(\overline{x}^{(k)} - x^{(k)}\right)}_{>0,\ (5.22)} \geq 0 \end{aligned}$$

Analogously, we can proof (5.104)

$$\begin{aligned}
\left(v_l^{(k)}\right)^T \underbrace{\Delta x^{(k)}}_{=z^{(k)}-x^{(k)}} &= \left(v_l^{(k)}\right)^T \left(z^{(k)} - x^{(k)}\right) \\
&= \left(v_l^{(k)}\right)^T z^{(k)} \underbrace{- \left(v_l^{(k)}\right)^T \underline{x}^{(k)} + \left(v_l^{(k)}\right)^T \underline{x}^{(k)}}_{=0} - \left(v_l^{(k)}\right)^T x^{(k)} \\
&= \underbrace{\left(v_l^{(k)}\right)^T z^{(k)} - \left(v_l^{(k)}\right)^T \underline{x}^{(k)}}_{=-\left(v_l^{(k)}\right)^T b_l^{(k)}(z^{(k)})=0,\ (5.78)} + \left(v_l^{(k)}\right)^T \underline{x}^{(k)} - \left(v_l^{(k)}\right)^T x^{(k)} \\
&= \left(v_l^{(k)}\right)^T \underline{x}^{(k)} - \left(v_l^{(k)}\right)^T x^{(k)} \\
&= \left(v_l^{(k)}\right)^T \underbrace{\left(\underline{x}^{(k)} - x^{(k)}\right)}_{<0,\ (5.20)} \\
&\leq 0
\end{aligned}$$

□

The next Lemma provides an upper bound on the descent of the objective function in iteration $x^{(k)}$. The proof of Zillober [102], see Section 2.3, is extended by feasibility constraints $e_j(x),\ j = 1, \ldots, m_f$. The results are required in Theorem 5.2, which shows that a sufficient descent with respect to the augmented Lagrangian is obtained.

Lemma 5.3. *Let the sequences $\left\{x^{(k)}, y^{(k)}\right\}$ and $\left\{z^{(k)}, v^{(k)}\right\}$ be computed by Algorithm 16 and let $f^{(k)}(x)$ and $c_j^{(k)}(x),\ j = 1, \ldots, m_c$, be the corresponding convex approximations defined by (4.21) and (4.5), obtained by a sequence of feasible asymptotes according to Definition 4.1. Moreover, the primal search direction is denoted by $\Delta x^{(k)} := z^{(k)} - x^{(k)}$, where $z^{(k)}$ is the primal solution of subproblem (5.19) formulated in the current iterate $x^{(k)}$. If $e_j(x),\ j = 1, \ldots, m_f$, are convex functions, then*

$$\begin{aligned}
\nabla f\left(x^{(k)}\right)^T \Delta x^{(k)} \leq\ & -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} - \left(v_c^{(k)}\right)^T A_c(x^{(k)})^T \Delta x^{(k)} \\
& - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} \qquad (5.105) \\
& + \left(v_e^{(k)}\right)^T \left(e\left(x^{(k)}\right) - e\left(z^{(k)}\right)\right)
\end{aligned}$$

holds for all $k = 0, 1, 2, \ldots$.

Proof. Proceeding from (5.75) we get:

$$\begin{aligned}
\nabla_x L^{(k)}\left(z^{(k)}, v^{(k)}\right) &= \underbrace{\nabla f^{(k)}\left(z^{(k)}\right)}_{=\nabla f\left(x^{(k)}\right)+\nabla R_{f^{(k)}}\left(z^{(k)}\right),\ (5.86)} + \underbrace{A_{c^{(k)}}(z^{(k)})v_c^{(k)}}_{=A_c(x^{(k)})v_c^{(k)}+\nabla R_{c^{(k)}}(z^{(k)})v_c^{(k)},\ (5.88)} \\
&\quad +A_e(z^{(k)})v_e^{(k)} + v_u^{(k)} - v_l^{(k)} \\
&= \nabla f\left(x^{(k)}\right) + \nabla R_{f^{(k)}}\left(z^{(k)}\right) + A_c(x^{(k)})v_c^{(k)} + \nabla R_{c^{(k)}}(z^{(k)})v_c^{(k)} \\
&\quad +A_e(z^{(k)})v_e^{(k)} + v_u^{(k)} - v_l^{(k)} \\
&= 0
\end{aligned}$$

By reformulation and multiplication with $\Delta x^{(k)}$ we get

$$\begin{aligned}
\nabla f\left(x^{(k)}\right)^T \Delta x^{(k)} &= -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} - \left(v_c^{(k)}\right)^T A_c(x^{(k)})^T \Delta x^{(k)} \\
&\quad - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} - \left(v_e^{(k)}\right)^T \underbrace{A_e(z^{(k)})^T \Delta x^{(k)}}_{\geq e(z^{(k)})-e(x^{(k)}),\ (5.94)} \\
&\quad - \underbrace{\left(v_u^{(k)}\right)^T \Delta x^{(k)}}_{\geq 0,\ \text{see } (5.103)} + \underbrace{\left(v_l^{(k)}\right)^T \Delta x^{(k)}}_{\leq 0,\ \text{see } (5.104)} \\
&\leq -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} - \left(v_c^{(k)}\right)^T A_c(x^{(k)})^T \Delta x^{(k)} \\
&\quad - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} + \left(v_e^{(k)}\right)^T \left(e(x^{(k)}) - e(z^{(k)})\right)
\end{aligned}$$

□

To prove that the search direction is a descent direction for the augmented Lagrangian, we need the following lemma, which shows that condition (5.106) holds. We review the proof of Zillober, see Section 2.3 of [102].

Lemma 5.4. *Let the sequences* $\left\{x^{(k)}, y^{(k)}\right\}$ *and* $\left\{z^{(k)}, v^{(k)}\right\}$ *be computed by Algorithm 16 and let* $c_j^{(k)}(x)$, $j = 1, \ldots, m_c$, *be the corresponding convex approximations, obtained by a sequence of feasible asymptotes according to Definition 4.1. Let the primal search direction be* $\Delta x^{(k)} := z^{(k)} - x^{(k)}$, *where* $z^{(k)} \in \mathbb{R}^n$ *is the primal solution of subproblem (5.19) formulated in* $x^{(k)} \in \mathbb{R}^n$. *Moreover, let the corresponding Lagrangian multipliers be defined by* $v_c^{(k)} \in \mathbb{R}^{m_c}$, *then*

$$\left(v_c^{(k)}\right)^T R_{c^{(k)}}(z^{(k)}) - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} \leq 0 \tag{5.106}$$

holds for all $k = 0, 1, 2, \ldots$.

Proof. As

$$\begin{aligned}
&\left(v_c^{(k)}\right)^T R_{c^{(k)}}(z^{(k)}) - \left(v_c^{(k)}\right)^T \nabla R_{c^{(k)}}(z^{(k)})^T \Delta x^{(k)} \\
&= \sum_{j=1}^{m_c} \left[\left(v_c^{(k)}\right)_j R_{c_j^{(k)}}\left(z^{(k)}\right) - \left(v_c^{(k)}\right)_j \nabla R_{c_j^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)}\right]
\end{aligned}$$

holds, we consider each constraint $c_j(x),\ j = 1, \dots, m_c$ individually, to show that each term of the sum is less or equal than zero. For this purpose, we use the results of Lemma 5.1.

$$
\begin{aligned}
& -\left(v_c^{(k)}\right)_j \nabla R_{c_j^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} + \left(v_c^{(k)}\right)_j R_{c_j^{(k)}}\left(z^{(k)}\right) \\
\overset{(5.100),(5.102)}{=} \quad & \sum_{I_+^{(j,k)}} -\left(v_c^{(k)}\right)_j \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \\
& \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} - \frac{\left(z_i^{(k)} - x_i^{(k)}\right)^2}{U_i^{(k)} - z_i^{(k)}}\right] \\
& + \sum_{I_-^{(j,k)}} \left(v_c^{(k)}\right)_j \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \\
& \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} - \frac{\left(z_i^{(k)} - x_i^{(k)}\right)^2}{z_i^{(k)} - L_i^{(k)}}\right] \\
= \quad & \sum_{I_+^{(j,k)}} -\left(v_c^{(k)}\right)_j \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(z_i^{(k)} - x_i^{(k)}\right)^2 \\
& \left[\frac{2U_i^{(k)} - z_i^{(k)} - x_i^{(k)} - \left(U_i^{(k)} - z_i^{(k)}\right)}{\left(U_i^{(k)} - z_i^{(k)}\right)^2}\right] \\
& + \sum_{I_-^{(j,k)}} \left(v_c^{(k)}\right)_j \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(z_i^{(k)} - x_i^{(k)}\right)^2 \\
& \left[\frac{-2L_i^{(k)} + z_i^{(k)} + x_i^{(k)} - \left(z_i^{(k)} - L_i^{(k)}\right)}{\left(z_i^{(k)} - L_i^{(k)}\right)^2}\right] \\
= \quad & -\sum_{I_+^{(j,k)}} \underbrace{\left(v_c^{(k)}\right)_j}_{\geq 0} \underbrace{\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}}_{\geq 0} \underbrace{\left(z_i^{(k)} - x_i^{(k)}\right)^2}_{\geq 0} \underbrace{\left[\frac{U_i^{(k)} - x_i^{(k)}}{\left(U_i^{(k)} - z_i^{(k)}\right)^2}\right]}_{>0} \\
& + \sum_{I_-^{(j,k)}} \underbrace{\left(v_c^{(k)}\right)_j}_{\geq 0} \underbrace{\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}}_{<0} \underbrace{\left(z_i^{(k)} - x_i^{(k)}\right)^2}_{\geq 0} \underbrace{\left[\frac{x_i^{(k)} - L_i^{(k)}}{\left(z_i^{(k)} - L_i^{(k)}\right)^2}\right]}_{>0} \leq 0
\end{aligned}
$$

□

Moreover, we have to consider the parameter $\eta_i^{(k)}, i = 1, \ldots, n$, see (4.30), which gives an estimate of the curvature of the approximation of the objective function $f^{(k)}(x)$. It is shown by Zillober, in Corollary 4.14 of [97], that $\eta_i^{(k)},\ i = 1, \ldots, n$, is bounded from below, if the sequence of asymptotes is feasible. The parameter guarantees the sufficient descent of the augmented Lagrangian function, see Theorem 5.2.

Lemma 5.5. *Let the sequences $\left\{x^{(k)}, y^{(k)}\right\}$ and $\left\{z^{(k)}, v^{(k)}\right\}$ be computed by Algorithm 16 with $\eta_i^{(k)},\ i = 1, \ldots, n$, defined by (4.30). If the sequence of asymptotes is feasible according to Definition 4.1, then*

$$\min_{i=1,\ldots,n} \eta_i^{(k)} =: \eta^{(k)} > \eta > 0 \tag{5.107}$$

holds for all $k = 0, 1, 2, \ldots$ with

$$\eta \ := \ \tau \frac{(2-\omega)\,\xi}{\left(U_{max} - L_{min}\right)^2}. \tag{5.108}$$

Proof. Using (5.22) we get for a fixed $i \in I_+^{(k)}$ with $\omega \in]0, 1[$

$$\begin{aligned} z_i^{(k)} &\le \underbrace{\overline{x}_i^{(k)}}_{=x_i^{(k)}+\omega\left(U_i^{(k)}-x_i^{(k)}\right),\ (5.22)} \\ z_i^{(k)} &\le x_i^{(k)} + \omega\left(U_i^{(k)} - x_i^{(k)}\right) \\ -z_i^{(k)} &\ge -x_i^{(k)} - \omega\left(U_i^{(k)} - x_i^{(k)}\right) \end{aligned}$$

Adding $2U_i^{(k)} - x_i^{(k)}$ leads to

$$\begin{aligned} 2U_i^{(k)} - x_i^{(k)} - z_i^{(k)} &\ge 2U_i^{(k)} - x_i^{(k)} - x_i^{(k)} - \omega\left(U_i^{(k)} - x_i^{(k)}\right) \\ &= (2-\omega)\underbrace{\left(U_i^{(k)} - x_i^{(k)}\right)}_{\ge\xi,\ (5.30)} \end{aligned}$$

In total we get

$$2U_i^{(k)} - z_i^{(k)} - x_i^{(k)} \ \ge \ (2-\omega)\,\xi \tag{5.109}$$

We proceed from (4.30) and use (5.109), with a fixed $i \in I_+^{(k)}$.

$$
\begin{aligned}
\eta_i^{(k)} &= \left(\underbrace{\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}}_{\geq 0} + \tau \right) \frac{2U_i^{(k)} - z_i^{(k)} - x_i^{(k)}}{\left(U_i^{(k)} - z_i^{(k)}\right)^2} \\
&\geq \tau \frac{2U_i^{(k)} - z_i^{(k)} - x_i^{(k)}}{\left(U_i^{(k)} - z_i^{(k)}\right)^2} \\
&> \tau \frac{2U_i^{(k)} - z_i^{(k)} - x_i^{(k)}}{\left(U_{\max} - L_{\min}\right)^2} \\
&\overset{(5.109)}{\geq} \underbrace{\tau \frac{(2-\omega)\,\xi}{\left(U_{\max} - L_{\min}\right)^2}}_{=:\eta} \\
&> 0
\end{aligned}
$$

with $\tau > 0$. The corresponding proof for $i \in I_-^{(k)}$ can be given analogously with (5.20) and (5.29). All together we get:

$$
\eta_i^{(k)} \geq \eta^{(k)} > \eta > 0, \ \forall i = 1, \ldots, n. \tag{5.110}
$$

□

An important part of the convergence proof is to show that the penalty parameters and the augmented Lagrangian function are bounded. Before we can show this, we have to show that the gradients of the approximations $f^{(k)}(x)$ and $c_j^{(k)}(x), \ j = 1, \ldots, m_c$, are bounded on $X^{(k)}$. We review the proof for the objective function and the regular constraints $c_j(x), \ j = 1, \ldots, m_c$, given by Zillober [97] in Theorem 4.13.

Lemma 5.6. *Let the sequence $\left\{x^{(k)}, y^{(k)}\right\}$ be computed by Algorithm 16 and let $f^{(k)}(x)$ and $c_j^{(k)}(x), \ j = 1, \ldots, m_c$, be the corresponding convex approximations defined by (4.21) and (4.5), obtained by a feasible sequence of asymptotes $\left\{L^{(k)}\right\}$ and $\left\{U^{(k)}\right\}$ according to Definition 4.1. Let the lower and upper bounds $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$, be defined by (5.20) and (5.22).*
If F is nonempty and compact, then there exists $M_0 > 0$ and $M_j > 0, \ j = 1, \ldots, m_c$ such that

$$
\left| \frac{\partial f^{(k)}(x)}{\partial x_i} \right| < M_0, \quad i = 1, \ldots, n \tag{5.111}
$$

$$
\left| \frac{\partial c_j^{(k)}(x)}{\partial x_i} \right| \leq M_j, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m_c \tag{5.112}
$$

holds for all $x \in X^{(k)}$ and $k = 0, 1, 2, \ldots$

Proof. We start with the derivatives of the approximated objective function, given in (4.23)

$$\frac{\partial f^{(k)}(x)}{\partial x_i} = \frac{\partial h^{(k)}(x)}{\partial x_i} + \begin{cases} \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^2}, & \text{if } i \in I_+^{(k)} \\ \frac{\partial f\left(x^{(k)}\right)}{\partial x_i} \frac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left(x_i - L_i^{(k)}\right)^2}, & \text{otherwise} \end{cases} \tag{5.113}$$

with

$$\frac{\partial h^{(k)}(x)}{\partial x_i} = \begin{cases} \tau \frac{\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right)}{\left(U_i^{(k)} - x_i\right)^2}, & \text{if } i \in I_+^{(k)} \\ \tau \frac{-\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(x_i - L_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2}, & \text{otherwise.} \end{cases} \tag{5.114}$$

Moreover, we consider (5.20) and (5.22)

$$\begin{aligned} U_i^{(k)} \underbrace{-\overline{x}_i^{(k)}}_{=-x_i^{(k)}-\omega\left(U_i^{(k)}-x_i^{(k)}\right),\ (5.22)} &= U_i^{(k)} - x_i^{(k)} - \omega\left(U_i^{(k)} - x_i^{(k)}\right) \\ &= (1-\omega)\left(U_i^{(k)} - x_i^{(k)}\right) > 0 \end{aligned} \tag{5.115}$$

$$\begin{aligned} \underbrace{\underline{x}_i^{(k)}}_{=x_i^{(k)}-\omega\left(x_i^{(k)}-L_i^{(k)}\right),\ (5.20)} - L_i^{(k)} &= x_i^{(k)} - \omega\left(x_i^{(k)} - L_i^{(k)}\right) - L_i^{(k)} \\ &= (1-\omega)\left(x_i^{(k)} - L_i^{(k)}\right) > 0. \end{aligned} \tag{5.116}$$

We can show that (5.114) is bounded. Starting with an arbitrary $i \in I_+^{(k)}$ we get

$$\begin{aligned} &\left| \tau \frac{\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right)}{\left(U_i^{(k)} - x_i\right)^2} \right| \\ \leq\ & \tau \frac{\left(x_i - x_i^{(k)}\right)^2 + 2\left|\left(x_i - x_i^{(k)}\right)\right|\left(U_i^{(k)} - x_i\right)}{\left(U_i^{(k)} - x_i\right)^2} \end{aligned}$$

As

$$\begin{aligned} \left|x_i - x_i^{(k)}\right| &< U_i^{(k)} - L_i^{(k)}, && (5.117)\\ U_i^{(k)} - x_i &< U_i^{(k)} - L_i^{(k)} && (5.118) \end{aligned}$$

and

$$U_i^{(k)} - x_i \geq U_i^{(k)} - \overline{x}_i^{(k)} \qquad (5.119)$$

holds, we get

$$\begin{aligned} &\left|\tau \frac{\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(U_i^{(k)} - x_i\right)}{\left(U_i^{(k)} - x_i\right)^2}\right| \\ &< \quad \tau \frac{\left(U_i^{(k)} - L_i^{(k)}\right)^2 + 2\left(U_i^{(k)} - L_i^{(k)}\right)\left(U_i^{(k)} - L_i^{(k)}\right)}{\left(U_i^{(k)} - \overline{x}_i^{(k)}\right)^2} \\ &\overset{(5.115)}{=} \quad \tau \frac{3\left(U_i^{(k)} - L_i^{(k)}\right)^2}{(1-\omega)^2\left(U_i^{(k)} - x_i^{(k)}\right)^2} \\ &\overset{\text{Def. 4.1}}{\leq} \quad \tau \frac{3\left(U_i^{(k)} - L_i^{(k)}\right)^2}{(1-\omega)^2 \xi^2} \\ &\leq \quad \tau \frac{3\left(U_{\max} - L_{\min}\right)^2}{(1-\omega)^2 \xi^2} \quad =: \overline{M} \end{aligned}$$

And respectively for $i \in I_-^{(k)}$

$$\begin{aligned} &\left|\tau \frac{-\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(x_i - L_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2}\right| \\ &\leq \tau \frac{\left(x_i - x_i^{(k)}\right)^2 + 2\left|\left(x_i - x_i^{(k)}\right)\right|\left(x_i - L_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2} \end{aligned}$$

As

$$x_i - L_i^{(k)} < U_i^{(k)} - L_i^{(k)} \qquad (5.120)$$

and

$$x_i - L_i^{(k)} \geq \overline{x}_i^{(k)} - L_i^{(k)} \qquad (5.121)$$

holds, we get

$$\begin{aligned}
&\left|\tau\frac{-\left(x_i - x_i^{(k)}\right)^2 + 2\left(x_i - x_i^{(k)}\right)\left(x_i - L_i^{(k)}\right)}{\left(x_i - L_i^{(k)}\right)^2}\right| \\
&\quad < \quad \tau\frac{\left(U_i^{(k)} - L_i^{(k)}\right)^2 + 2\left(U_i^{(k)} - L_i^{(k)}\right)\left(U_i^{(k)} - L_i^{(k)}\right)}{\underline{x}_i^{(k)} - L_i^{(k)}} \\
&\quad \overset{(5.116)}{=} \quad \tau\frac{3\left(U_i^{(k)} - L_i^{(k)}\right)^2}{(1-\omega)^2\left(x_i^{(k)} - L_i^{(k)}\right)^2} \\
&\quad \overset{\text{Def. 4.1}}{\leq} \quad \tau\frac{3\left(U_i^{(k)} - L_i^{(k)}\right)^2}{(1-\omega)^2\,\xi^2} \\
&\quad \leq \quad \tau\frac{3\left(U_{\max} - L_{\min}\right)^2}{(1-\omega)^2\,\xi^2} \quad = \quad \overline{M}
\end{aligned}$$

Due to Assumption 1, $\left|\frac{\partial f(x)}{\partial x_i}\right|$ is bounded on F, i.e., there exists a $\overline{M}_0 \geq 0$, such that $\left|\frac{\partial f(x)}{\partial x_i}\right| \leq \overline{M}_0$ holds. Together with the previous results and (5.113), we get for each $i \in I_+^{(k)}$

$$\begin{aligned}
\left|\frac{\partial f^{(k)}(x)}{\partial x_i}\right| \quad &\leq \quad \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i\right)^2} + \underbrace{\left|\frac{\partial h^{(k)}(x)}{\partial x_i}\right|}_{<\overline{M}} \\
&< \quad \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - \overline{x}_i^{(k)}\right)^2} + \overline{M} \\
&\overset{(5.115)}{=} \quad \frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left((1-\omega)\left(U_i^{(k)} - x_i^{(k)}\right)\right)^2} + \overline{M} \\
&= \quad \underbrace{\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}}_{\leq\overline{M}_0}\frac{1}{(1-\omega)^2} + \overline{M} \\
&\leq \quad \overline{M}_0\frac{1}{(1-\omega)^2} + \overline{M} \\
&=: \quad M_0
\end{aligned}$$

The same can be shown for each $i \in I_{-}^{(k)}$,

$$\begin{aligned}
\left|\frac{\partial f^{(k)}(x)}{\partial x_i}\right| &\leq \left|\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\right| \frac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left(x_i - L_i^{(k)}\right)^2} + \underbrace{\left|\frac{\partial h^{(k)}(x)}{\partial x_i}\right|}_{<\overline{M}} \\
&< \left|\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\right| \frac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left(\underline{x}_i^{(k)} - L_i^{(k)}\right)^2} + \overline{M} \\
&\overset{(5.116)}{=} \left|\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\right| \frac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left((1-\omega)\left(x_i^{(k)} - L_i^{(k)}\right)\right)^2} + \overline{M} \\
&= \underbrace{\left|\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\right|}_{\leq \overline{M}_0} \frac{1}{(1-\omega)^2} + \overline{M} \\
&\leq \overline{M}_0 \frac{1}{(1-\omega)^2} + \overline{M} \\
&= M_0
\end{aligned}$$

The corresponding proof for $\nabla c_j^{(k)}(x),\ j = 1, \ldots, m_c$, can be given analogously with $\tau = 0$, i.e.,

$$\begin{aligned}
\left|\frac{\partial c_j^{(k)}(x)}{\partial x_i}\right| &\leq \underbrace{\left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right|}_{\leq \overline{M}_j} \frac{1}{(1-\omega)^2} \\
&\leq \overline{M}_j \frac{1}{(1-\omega)^2} \\
&=: M_j
\end{aligned}$$

□

Moreover, we need to show that the augmented Lagrangian merit function defined in (5.73) is bounded from below. The proof including the objective function $f(x)$ and regular constraints $c_j(x),\ j = 1, \ldots, m_c$, is given in Theorem 5.3 of Zillober [97]. It has to be extended by adding feasibility constraints $e_j(x),\ j = 1, \ldots, m_f$. Lemma 5.7 is needed for the main convergence Theorem 5.5.

Lemma 5.7. *Let F defined by (5.18) be nonempty and compact. Then there exists a $M_\Phi \in \mathbb{R}$ such that*

$$\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix} \geq M_\Phi \tag{5.122}$$

holds for all $x \in F$, $y \in Y$ with 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 (5.73), we obtain

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

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

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

□

5.2.3 Convergence Theorems

With the results of Section 5.2.2 we can prove the convergence of Algorithm 16. In the following theorem it is shown that the primal and dual variables are bounded. This is essential to give an estimation of the descent properties of the augmented Lagrangian, see also Theorem 5.2. The proof is based on Theorem 5.6 of Zillober [97], and Theorem 2.4.1 of Zillober [102], which are extended by $e_j(x),\ j=1,\ldots,m_f$.

Theorem 5.1. *Let the sequences $\left\{x^{(k)}, y^{(k)}\right\}$ and $\left\{z^{(k)}, v^{(k)}\right\}$ be computed by Algorithm 16, where the corresponding approximations $f^{(k)}(x)$ and $c_j^{(k)}(x)$ are defined by (4.21) and (4.5). Moreover, let the smallest eigenvalue of $A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right)^T A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right)$ be larger than a lower bound $\left(\kappa^{(k)}\right)^2 > 0$. Let the asymptotes $L_i^{(k)}$ and $U_i^{(k)},\ i=1,\ldots,n$, be feasible according to Definition 4.1 and let F defined by (5.18) be nonempty and compact. Then all iterates $x^{(k)}$ are in F and the corresponding multipliers $y^{(k)}\in\mathbb{R}^{m_c+m_f}$ are bounded, i.e., there exists a $y_{max}\in\mathbb{R}$ such that $\left|v_i^{(k)}\right|\leq y_{max}$, $\left|y_i^{(k)}\right|\leq y_{max},\ i=1,\ldots,m_c+m_f$, for all $k\geq 0$.*

Proof. We assume an infinite sequence $\left\{x^{(k)}, y^{(k)}\right\}$. We start with a feasible point with respect to the feasibility constraints, i.e., $x^{(0)}\in F$. The solution $z^{(k)}$ of subproblem (5.19) lies in $F_X^{(k)}\subseteq F$, see (5.11). As F is convex and $x^{(k)}\in F$, $x^{(k+1)}\in]x^{(k)}, z^{(k)}]\subseteq F$.

According to Definition 2 we define the set of active constraints with respect to $z^{(k)}$ by

$$\begin{aligned}
\mathbb{J}_{c^{(k)}}\left(z^{(k)}\right) &:= \left\{j = 1, \ldots, m_c \,\middle|\, c_j^{(k)}\left(z^{(k)}\right) = 0\right\} && (5.123)\\
\mathbb{J}_e\left(z^{(k)}\right) &:= \left\{j = 1, \ldots, m_f \mid e_j\left(z^{(k)}\right) = 0\right\} && (5.124)\\
\mathbb{J}_{b^{(k)}}\left(z^{(k)}\right) &:= \left\{j = 1, \ldots, 2n \,\middle|\, b_j^{(k)}\left(z^{(k)}\right) = 0\right\} && (5.125)\\
\mathbb{J}^{(k)} &:= \mathbb{J}_{c^{(k)}}\left(z^{(k)}\right) \cup \mathbb{J}_e\left(z^{(k)}\right) \cup \mathbb{J}_{b^{(k)}}\left(z^{(k)}\right) && (5.126)
\end{aligned}$$

We consider the optimality condition (5.75) of subproblem (5.19)

$$\nabla_x L^{(k)}\left(z^{(k)}, v^{(k)}\right) = 0$$

which leads to

$$\begin{aligned}
\nabla f^{(k)}\left(z^{(k)}\right) &= -A_{c^{(k)}}\left(z^{(k)}\right) v_c^{(k)} - A_e\left(z^{(k)}\right) v_e^{(k)} - v_u^{(k)} + v_l^{(k)}\\
&= -\sum_{j \in \mathbb{J}_{c^{(k)}}\left(z^{(k)}\right)} \left(v_c^{(k)}\right)_j \nabla c_j^{(k)}\left(z^{(k)}\right)\\
&\quad - \sum_{j \in \mathbb{J}_e\left(z^{(k)}\right)} \left(v_e^{(k)}\right)_j \nabla e_j\left(z^{(k)}\right)\\
&\quad - \sum_{j \in \mathbb{J}_{b^{(k)}}\left(z^{(k)}\right)} \left(v_b^{(k)}\right)_j \nabla b_j^{(k)}\left(z^{(k)}\right)\\
&= -A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right) v_{\mathbb{J}^{(k)}} && (5.127)
\end{aligned}$$

where the columns of the gradients of the active constraints $c_j^{(k)}\left(z^{(k)}\right)$, $j = 1, \ldots, m_c$, $e_j\left(z^{(k)}\right)$, $j = 1, \ldots, m_f$, and $b_j^{(k)}\left(z^{(k)}\right)$, $j = 1, \ldots, 2n$, form the $n \times |\mathbb{J}^{(k)}|$ matrix $A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right) \in \mathbb{R}^{n \times |\mathbb{J}^{(k)}|}$, i.e.,

$$A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right) := \left(\left[A_{c^{(k)}}\left(z^{(k)}\right)\right]_{\mathbb{J}_{c^{(k)}}\left(z^{(k)}\right)}, \left[A_e\left(z^{(k)}\right)\right]_{\mathbb{J}_e\left(z^{(k)}\right)}, \left[A_{b^{(k)}}\left(z^{(k)}\right)\right]_{\mathbb{J}_{b^{(k)}}\left(z^{(k)}\right)}\right) \qquad (5.128)$$

and $v_{\mathbb{J}^{(k)}}$ are the corresponding Lagrangian multipliers, i.e.,

$$v_{\mathbb{J}^{(k)}} := \begin{pmatrix} \left[v_c^{(k)}\right]^T_{\mathbb{J}_{c^{(k)}}\left(z^{(k)}\right)} \\ \left[v_e^{(k)}\right]^T_{\mathbb{J}_e\left(z^{(k)}\right)} \\ \left[v_b^{(k)}\right]^T_{\mathbb{J}_{b^{(k)}}\left(z^{(k)}\right)} \end{pmatrix} \in \mathbb{R}^{|\mathbb{J}^{(k)}|} \qquad (5.129)$$

Let $\kappa^{(k)} \in \mathbb{R}$ be the smallest singular value of $-A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right)$. Then

$$\begin{aligned}
\kappa^{(k)} &:= \inf_{w \neq 0} \frac{\left\|-A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right) w\right\|_2}{\|w\|_2} \leq \frac{\left\|-A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right) v_{\mathbb{J}^{(k)}}\right\|_2}{\left\|v_{\mathbb{J}^{(k)}}\right\|_2}\\
\Longrightarrow \left\|v_{\mathbb{J}^{(k)}}\right\|_2 &\leq \frac{1}{\kappa^{(k)}} \left\|-A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right) v_{\mathbb{J}^{(k)}}\right\|_2 \overset{(5.127)}{=} \frac{1}{\kappa^{(k)}} \left\|\nabla f^{(k)}\left(z^{(k)}\right)\right\|_2,
\end{aligned}$$

if $\kappa^{(k)} \neq 0$. $\left(\kappa^{(k)}\right)^2$ is the smallest eigenvalue of $A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right)^T A_{\mathbb{J}^{(k)}}\left(z^{(k)}\right)$ which is bounded away from zero. Lemma 5.6, i.e.,

$$\left|\frac{\partial f^{(k)}(x)}{\partial x_i}\right| \ < \ M_0, \quad i = 1, \ldots, n \tag{5.130}$$

implies

$$\begin{aligned}
\left\|v_{\mathbb{J}^{(k)}}\right\|_2 \ &\leq \ \frac{1}{\kappa^{(k)}} \left\|\nabla f^{(k)}\left(z^{(k)}\right)\right\|_2 && (5.131)\\
&= \ \frac{1}{\kappa^{(k)}} \left(\sum_{i=1}^{n} \underbrace{\left(\frac{\partial f^{(k)}(z^{(k)})}{\partial x_i}\right)^2}_{<M_0^2, \text{ see } (5.111)}\right)^{1/2} && (5.132)\\
&< \ \frac{\sqrt{n}}{\kappa^{(k)}} M_0 && (5.133)
\end{aligned}$$

The same holds for $v^{(k)}$ as the remaining Lagrangian multipliers are equal to zero, i.e., $\left\|v^{(k)}\right\|_2 < \frac{\sqrt{n}}{\kappa^{(k)}} M_0$. We get the maximal value of the Lagrangian multipliers by

$$y_{\max} \ := \ \max_k \left\{\left\|v^{(k)}\right\|_\infty, \left\|y^{(k)}\right\|_\infty, \left\|y^{(0)}\right\|_\infty\right\} \tag{5.134}$$

□

In terms of the unique optimal solution $\left(z^{(k)}, v^{(k)}\right)$ of subproblem (5.19) we define

$$\delta^{(k)} \ := \ \left\|z^{(k)} - x^{(k)}\right\|_2 . \tag{5.135}$$

We want to show that the resulting search direction

$$d^{(k)} := \begin{pmatrix} \Delta x^{(k)} \\ \Delta y_c^{(k)} \\ \Delta y_e^{(k)} \end{pmatrix} \tag{5.136}$$

with

$$\begin{aligned}
\Delta x^{(k)} \ &:= \ z^{(k)} - x^{(k)} && (5.137)\\
\Delta y_c^{(k)} \ &:= \ v_c^{(k)} - y_c^{(k)} && (5.138)\\
\Delta y_e^{(k)} \ &:= \ v_e^{(k)} - y_e^{(k)} && (5.139)
\end{aligned}$$

is a descent direction for the augmented Lagrangian function $\Phi_\rho \begin{pmatrix} x \\ y \end{pmatrix}$, i.e., there are penalty parameters $\rho \in \mathbb{R}^{m_c + m_f}$ such that

$$\nabla \Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} < 0 \tag{5.140}$$

holds. This leads to the following theorem, see Theorem 5.7 of Zillober [97] and Theorem 2.4.2 of Zillober [102], respectively. Moreover, we include feasibility constraints $e_j(x), \ j = 1, \ldots, m_f$.

Theorem 5.2. *Let the assumptions of Theorem 5.1 hold. Consider $d^{(k)} \in \mathbb{R}^{n+m_c+m_f}$ defined by (5.136) and $\eta^{(k)}$ defined by (4.30) with lower bound $\eta > 0$, see Lemma 5.5. Let $\delta^{(k)} := \|z^{(k)} - x^{(k)}\|_2 \neq 0$ hold and let the augmented Lagrangian be given by (5.73). For each $\left(x^{(k)}, y^{(k)}\right)$ the following properties hold:*

1. *The loop consisting of Step 5 to Step 7 of Algorithm 16 is finite, i.e., there are penalty parameters $\overline{\rho}^{(k)} := \begin{pmatrix} \overline{\rho}_c^{(k)} \\ \overline{\rho}_e^{(k)} \end{pmatrix} > 0$, $\overline{\rho}^{(k)} \in \mathbb{R}^{m_c+m_f}$ such that $d^{(k)}$ is a descent direction for the augmented Lagrangian function, i.e.,*
$$\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \leq -\frac{\eta^{(k)} \left(\delta^{(k)}\right)^2}{2} < 0, \tag{5.141}$$
for all $\rho_c \geq \overline{\rho}_c^{(k)}$ and $\rho_e \geq \overline{\rho}_e^{(k)}$.

2. *For each $\delta > 0$ there exists a $\overline{\rho}^\delta := \begin{pmatrix} \overline{\rho}_c^\delta \\ \overline{\rho}_e^\delta \end{pmatrix} > 0$, $\overline{\rho}^\delta \in \mathbb{R}^{m_c+m_f}$ such that for all $\left(x^{(k)}, y^{(k)}\right)$, with $\delta^{(k)} \geq \delta$,*
$$\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \leq -\frac{\eta^{(k)} \left(\delta^{(k)}\right)^2}{2} < -\frac{\eta\delta^2}{2} < 0, \tag{5.142}$$
holds for all $\rho \geq \overline{\rho}^\delta$.

Proof. According to Theorem 5.1, there exists an upper bound $y_{\max} \in \mathbb{R}$ such that
$$y_{\max} \quad := \quad \max_k \left\{\left\|v^{(k)}\right\|_\infty, \left\|y^{(k)}\right\|_\infty, \left\|y^{(0)}\right\|_\infty\right\} \tag{5.143}$$
holds.

We will start with the proof of the first part, which guarantees that we get a descent direction, if the penalty parameters are large enough. In the second part of the theorem it is shown, that the resulting penalty parameters are bounded for each $\delta^{(k)} \geq \delta > 0$.

$$\begin{aligned}
&\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
\overset{(5.74)}{=} \quad & \nabla f\left(x^{(k)}\right)^T \Delta x^{(k)} + \left(\overline{y}_c^{(k)} + \Gamma_c \overline{c}\left(x^{(k)}\right)\right)^T \qquad \underbrace{A_c\left(x^{(k)}\right)^T \Delta x^{(k)}}_{=c^{(k)}\left(z^{(k)}\right) - c\left(x^{(k)}\right) - R_{c^{(k)}}\left(z^{(k)}\right),\ (5.92)} \\
& + \widehat{c}\left(x^{(k)}\right)^T \Delta y_c^{(k)} + \left(\overline{y}_e^{(k)} + \Gamma_e \overline{e}\left(x^{(k)}\right)\right)^T \qquad \underbrace{A_e\left(x^{(k)}\right)^T \Delta x^{(k)}}_{=e\left(z^{(k)}\right) - e\left(x^{(k)}\right) - R_e\left(z^{(k)}\right),\ (5.93)} \\
& + \widehat{e}\left(x^{(k)}\right)^T \Delta y_e^{(k)}
\end{aligned}$$

$$
\begin{aligned}
&\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
\overset{(5.105)}{\leq}\quad & -\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \\
& - \left(v_c^{(k)}\right)^T \qquad \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),\ (5.92)} \qquad - \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 (5.50)}} \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 (5.52)}} \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)} \\
= \quad & -\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 \quad & -\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,\ (5.76)} + \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,\ (5.106)} \\
& + \left(v_e^{(k)}\right)^T e\left(x^{(k)}\right) - \underbrace{\left(v_e^{(k)}\right)^T e\left(z^{(k)}\right)}_{=0,\ (5.77)} \\
& - \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 (5.57)-(5.66) 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)}} \left(\rho_c\right)_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}{\left(\rho_c\right)_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)}} \left(\rho_e\right)_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}{\left(\rho_e\right)_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)}} \left(\rho_c\right)_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 (5.144) \\
& - \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 + \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{\left(\rho_e\right)_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}{\left(\rho_e\right)_j}\left(\Delta y_e^{(k)}\right)_j
\end{aligned}
$$

All together this leads to

$$\begin{aligned}\nabla\Phi_\rho\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \leq\ & \underbrace{-\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)}}_{\text{(a)}} \\ & \underbrace{+\sum_{j\in J_c^{(k)}} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \sum_{j\in J_c^{(k)}} \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)}_{\text{(b)}} \\ & \underbrace{-\sum_{j\in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{\left(\rho_c\right)_j}\left(\Delta y_c^{(k)}\right)_j}_{\text{(c)}} \underbrace{-\sum_{j\in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)}_{\text{(d)}} \qquad (5.145) \\ & \underbrace{+\sum_{j\in J_e^{(k)}} e_j\left(x^{(k)}\right)\left(\Delta y_e^{(k)}\right)_j}_{\text{(e)}} \underbrace{-\sum_{j\in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j}\left(\Delta y_e^{(k)}\right)_j}_{\text{(f)}}\end{aligned}$$

We now have to show that (5.145) is less than $-\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{2}$. Therefore, we consider each part individually.

Considering (a):
Using (5.101) we get:

$$\begin{aligned}&\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \\ =\ & \sum_{I_+^{(k)}}\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}+\tau\right]\left[\frac{\left(z_i^{(k)}-x_i^{(k)}\right)^2\left(2U_i^{(k)}-z_i^{(k)}-x_i^{(k)}\right)}{\left(U_i^{(k)}-z_i^{(k)}\right)^2}\right] \qquad (5.146) \\ & -\sum_{I_-^{(k)}}\left[\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}-\tau\right]\left[\frac{\left(z_i^{(k)}-x_i^{(k)}\right)^2\left(-2L_i^{(k)}+z_i^{(k)}+x_i^{(k)}\right)}{\left(z_i^{(k)}-L_i^{(k)}\right)^2}\right] \\ \overset{(4.30)}{=}\ & \sum_{I_+^{(k)}}\eta_i^{(k)}\left(z_i^{(k)}-x_i^{(k)}\right)^2+\sum_{I_-^{(k)}}\eta_i^{(k)}\left(z_i^{(k)}-x_i^{(k)}\right)^2\end{aligned}$$

Together with Definition $\eta^{(k)} := \min\limits_{i=1,\dots,n} \eta_i^{(k)}$ given in (4.31) we get

$$-\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)} \leq -\eta^{(k)}\left\|\Delta x^{(k)}\right\|_2^2 = -\eta^{(k)}\left(\delta^{(k)}\right)^2 < 0 \qquad (5.147)$$

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

$$\left|\Delta\left(y_c^{(k)}\right)_j\right| = \left|\left(v_c^{(k)}\right)_j - \left(y_c^{(k)}\right)_j\right| \leq y_{\max} \tag{5.148}$$

holds. This leads to

$$\begin{aligned}
&\sum_{j \in J_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - (\rho_c)_j\, c_j^2\left(x^{(k)}\right)\right] \\
\leq & \sum_{j \in J_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| \underbrace{\left|\left(\Delta y_c^{(k)}\right)_j\right|}_{\leq y_{\max},\ (5.148)} - (\rho_c)_j\, c_j^2\left(x^{(k)}\right)\right] \\
\leq & \sum_{j \in J_c^{(k)}} \left[2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - (\rho_c)_j\, c_j^2\left(x^{(k)}\right)\right] \qquad (5.149)
\end{aligned}$$

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

$$Z_c^{(k)} \; := \; \left\{j \in J_c^{(k)} \mid c_j\left(x^{(k)}\right) = 0\right\}. \tag{5.150}$$

To ensure property (5.141), we assume that the penalty parameters $(\rho_c)_j$, $j \in J_c^{(k)} \backslash Z_c^{(k)}$ are larger than $\left(\rho_1^{(k)}\right)_j$, $j \in J_c^{(k)}$ given by

$$\left(\rho_1^{(k)}\right)_j := \frac{2y_{\max}}{\left|c_j\left(x^{(k)}\right)\right|}, \text{ i.e., } \left(\rho_1^{(k)}\right)_j \leq (\rho_c)_j\,,\ \forall j \in J_c^{(k)} \backslash Z_c^{(k)}. \tag{5.151}$$

With (5.151) in (5.149) we get

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

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

In total we get

$$\sum_{j\in J_c^{(k)}}\left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j-(\rho_c)_j\,c_j^2\left(x^{(k)}\right)\right]\leq 0 \tag{5.152}$$

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

$$\begin{aligned}
\left|\sum_{j\in\overline{J}_c^{(k)}}\frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\left(\Delta y_c^{(k)}\right)_j\right| &\leq \sum_{j\in\overline{J}_c^{(k)}}\frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\underbrace{\left|\left(v_c^{(k)}\right)_j-\left(y_c^{(k)}\right)_j\right|}_{\leq y_{\max},\ (5.148)}\\
&\leq y_{\max}\sum_{j\in\overline{J}_c^{(k)}}\frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\\
&\leq (y_{\max})^2\sum_{j\in\overline{J}_c^{(k)}}\frac{1}{(\rho_c)_j}
\end{aligned}$$

This can be summarized by

$$\left|\sum_{j\in\overline{J}_c^{(k)}}\frac{\left(y_c^{(k)}\right)_j}{(\rho_c)_j}\left(\Delta y_c^{(k)}\right)_j\right| \leq (y_{\max})^2\sum_{j\in\overline{J}_c^{(k)}}\frac{1}{(\rho_c)_j}. \tag{5.153}$$

To ensure property (5.141), we assume that the penalty parameters $(\rho_c)_j$, $j\in\overline{J}_c^{(k)}$ are larger than $\rho_2^{(k)}$ given by:

$$\rho_2^{(k)} := m_c\frac{10\,(y_{\max})^2}{\eta^{(k)}\left(\delta^{(k)}\right)^2},\ \text{i.e., } \rho_2^{(k)}\leq(\rho_c)_j\,,\ j\in\overline{J}_c^{(k)}. \tag{5.154}$$

Using (5.153) and the definition of $\rho_2^{(k)}$ in (5.154) we get

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

As result we get

$$\left| \sum_{j \in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{\left(\rho_c\right)_j} \left(\Delta y_c^{(k)}\right)_j \right| \leq \frac{\eta^{(k)} \left(\delta^{(k)}\right)^2}{10}, \tag{5.155}$$

for each $\left(\rho_c\right)_j \geq \rho_2^{(k)}, \ j \in \overline{J}_c^{(k)}$.

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

$$e_j\left(x^{(k)}\right) \leq 0, \qquad j = 1, \ldots, m_f \tag{5.156}$$

$$e_j\left(z^{(k)}\right) \leq 0, \qquad j = 1, \ldots, m_f \tag{5.157}$$

Considering (d):
Due to (5.52) 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 (5.141), we assume that the penalty parameters $\left(\rho_e\right)_j,\ j \in J_e^{(k)}$ are larger than $\rho_3^{(k)}$ given by:

$$\rho_3^{(k)} \;:=\; m_f \frac{10\left(y_{\max}\right)^2}{\eta^{(k)}\left(\delta^{(k)}\right)^2},\ \text{i.e.,}\ \rho_3^{(k)} \leq \left(\rho_e\right)_j,\ j \in J_e^{(k)} \tag{5.158}$$

which leads to

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

As result we get

$$\left| \sum_{j \in J_e^{(k)}} \left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right) \right| \;\leq\; \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} \tag{5.159}$$

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

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

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

To ensure property (5.141), we assume that the penalty parameters $\left(\rho_e\right)_j,\ j\in J_e^{(k)}$ are larger than $\rho_3^{(k)}$ given by (5.158) which leads to

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

As result we get

$$\left|\sum_{j\in J_e^{(k)}}\left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)\right| \leq \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} \tag{5.160}$$

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

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

$$\left|\sum_{j\in\overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j}\left(\Delta y_e^{(k)}\right)_j\right| \leq \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} \tag{5.161}$$

for

$$\left(\rho_e\right)_j \geq \rho_3^{(k)} := m_f \frac{10\left(y_{\max}\right)^2}{\eta^{(k)}\left(\delta^{(k)}\right)^2},\ j\in\overline{J}_e^{(k)}. \tag{5.162}$$

Proceeding from (5.145) we can summarize previous calculations as follows:

$$\begin{aligned}
&\nabla\Phi_\rho\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
\leq\ & \underbrace{-\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)}}_{\leq -\eta^{(k)}\left(\delta^{(k)}\right)^2,\ (5.147)} \\
&\underbrace{+\sum_{j\in J_c^{(k)}} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \sum_{j\in J_c^{(k)}}\left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)}_{\leq 0,\ (5.152)} \\
&\underbrace{-\sum_{j\in\overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j}{\left(\rho_c\right)_j}\left(\Delta y_c^{(k)}\right)_j}_{\leq\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10},\ (5.155)} \underbrace{-\sum_{j\in J_e^{(k)}}\left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)}_{\leq\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10},\ (5.159)} \\
&\underbrace{+\sum_{j\in J_e^{(k)}} e_j\left(x^{(k)}\right)\left(\Delta y_e^{(k)}\right)_j}_{\leq\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10},\ (5.160)} \underbrace{-\sum_{j\in\overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j}{\left(\rho_e\right)_j}\left(\Delta y_e^{(k)}\right)_j}_{\leq\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10},\ (5.161)} \\
\leq\ & -\eta^{(k)}\left(\delta^{(k)}\right)^2 + 0 + \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} + \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} \\
& + \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} + \frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{10} \\
<\ & -\frac{\eta^{(k)}\left(\delta^{(k)}\right)^2}{2} \\
<\ & 0
\end{aligned}$$

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

In the second part of the theorem we have to show that the resulting penalty parameters are bounded, if $\delta^{(k)} \geq \delta$. We have already shown that we get a descent direction, if the penalty parameters are larger than

$$\begin{aligned} (\rho_c)_j &\geq \rho_1^{(k)} = \max_{j \in J_c^{(k)} \setminus Z_c^{(k)}} \left\{ \frac{2y_{\max}}{\left|c_j\left(x^{(k)}\right)\right|} \right\}, && \forall \;\; j = 1, \ldots, m_c, && (5.163) \\ (\rho_c)_j &\geq \rho_2^{(k)} = m_c \frac{10\left(y_{\max}\right)^2}{\eta^{(k)}\left(\delta^{(k)}\right)^2}, && \forall \; j = 1, \ldots, m_c, && (5.164) \\ (\rho_e)_j &\geq \rho_3^{(k)} = m_f \frac{10\left(y_{\max}\right)^2}{\eta^{(k)}\left(\delta^{(k)}\right)^2}, && \forall \; j = 1, \ldots, m_f. && (5.165) \end{aligned}$$

As proved in Theorem 5.1 the Lagrangian multipliers are bounded by $y_{\max}$. Moreover, $\delta^{(k)}$ is bounded away from zero due to $\delta^{(k)} \geq \delta > 0$. In Lemma 5.5 the lower bound on $\eta^{(k)} > \eta > 0$ is identified. In addition, the number of inequality constraints m_c and m_f is finite. As a consequence the values of $\rho_2^{(k)}$ and $\rho_3^{(k)}$ are bounded. Therefore, we only have to consider penalty parameter $\rho_1^{(k)}$ which depends on $c_j(x^{(k)}),\ j \in J_c^{(k)}$.

We consider (b) and $\rho_1^{(k)}$:
This proof is divided into two cases. First we assume that the absolute value of each constraint of $J_c^{(k)}$ is bounded by $\frac{1}{m_c}\frac{\eta\delta^2}{20y_{\max}}$. In the second case we assume that it exceeds this threshold. Let for each $j \in J_c^{(k)}$

$$\left|c_j\left(x^{(k)}\right)\right| \;\leq\; \frac{1}{m_c}\frac{\eta\delta^2}{20y_{\max}} \qquad (5.166)$$

hold with $\eta > 0$, see (5.107). According to (5.149) we get for each $j \in J_c^{(k)}$

$$\begin{aligned} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - (\rho_c)_j\, c_j^2\left(x^{(k)}\right) &\leq 2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \underbrace{(\rho_c)_j\, c_j^2\left(x^{(k)}\right)}_{\geq 0} \\ &\leq 2\left|c_j\left(x^{(k)}\right)\right| y_{\max} \end{aligned}$$

Assuming (5.166) we get

$$\begin{aligned} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - (\rho_c)_j\, c_j^2\left(x^{(k)}\right) &\leq 2 \underbrace{\left|c_j\left(x^{(k)}\right)\right|}_{\leq \frac{\eta\delta^2}{20m_c y_{\max}},\ \text{see (5.166)}} y_{\max} \\ &\leq 2\frac{\eta\delta^2}{20m_c y_{\max}} y_{\max} \\ &= \frac{\eta\delta^2}{10m_c}, \end{aligned}$$

Considering all constraints $c_j(x),\ j \in J_c^{(k)}$, we obtain

$$\sum_{j \in J_c^{(k)}} \left[2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - (\rho_c)_j\, c_j^2\left(x^{(k)}\right)\right] \;\leq\; \frac{\eta\delta^2}{10}. \qquad (5.167)$$

In addition, we have to show that the penalty parameter is bounded, if

$$\left|c_j\left(x^{(k)}\right)\right| \;>\; \frac{1}{m_c}\frac{\eta\delta^2}{20 y_{\max}} \tag{5.168}$$

holds for $j \in J_c^{(k)}$. To ensure property (5.142), we assume that the penalty parameters $\left(\rho_c\right)_j,\ j \in J_c^{(k)}$ are larger than ρ_4 given by:

$$\left(\rho_c\right)_j \;\geq\; \rho_4 \;\text{ with } \rho_4 \;:=\; \frac{40 m_c \left(y_{\max}\right)^2}{\eta\delta^2},\; j \in J_c^{(k)}. \tag{5.169}$$

Using (5.169) we get:

$$\begin{aligned}
&2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)\\
\leq\;\; & 2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \underbrace{\left(\rho_c\right)_j}_{\geq\rho_4} c_j^2\left(x^{(k)}\right)\\
\leq\;\; & 2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \rho_4 c_j^2\left(x^{(k)}\right)\\
\overset{(5.169)}{=}\;\; & 2\left|c_j\left(x^{(k)}\right)\right| y_{\max} - \frac{40 m_c \left(y_{\max}\right)^2}{\eta\delta^2} c_j^2\left(x^{(k)}\right)\\
=\;\; & \left|c_j\left(x^{(k)}\right)\right| \left(2 y_{\max} - \frac{40 m_c \left(y_{\max}\right)^2}{\eta\delta^2} \underbrace{\left|c_j\left(x^{(k)}\right)\right|}_{>\frac{1}{m_c}\frac{\eta\delta^2}{20 y_{\max}}}\right)\\
<\;\; & \left|c_j\left(x^{(k)}\right)\right| \left(2 y_{\max} - \frac{40 m_c \left(y_{\max}\right)^2}{\eta\delta^2}\frac{\eta\delta^2}{20 m_c y_{\max}}\right)\\
=\;\; & \left|c_j\left(x^{(k)}\right)\right| \left(2 y_{\max} - 2 y_{\max}\right)\\
=\;\; & 0.
\end{aligned}$$

which shows that

$$2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right) \;<\; 0 \tag{5.170}$$

holds for all $\left|c_j\left(x^{(k)}\right)\right| > \frac{1}{m_c}\frac{\eta\delta^2}{20 y_{\max}}$ and $\left(\rho_c\right)_j \geq \rho_4$.
Combining (5.167) and (5.170) leads to

$$\sum_{j\in J_c^{(k)}} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j - \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right) \;\leq\; \frac{\eta\delta^2}{10}, \quad \forall \left(\rho_c\right)_j \geq \rho_4 \tag{5.171}$$

Together we get

$$\begin{aligned}
\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \;&<\; -\eta\delta^2 + \frac{\eta\delta^2}{10} + \frac{\eta\delta^2}{10} + \frac{\eta\delta^2}{10} + \frac{\eta\delta^2}{10} + \frac{\eta\delta^2}{10}\\
&=\; -\frac{\eta\delta^2}{2}\\
&<\; 0
\end{aligned}$$

for $(\rho_c)_j \geq \overline{\rho}_c^\delta := \max\{\rho_4, \rho_5\} = \rho_4, \forall\ j = 1, \ldots, m_c$, and $(\rho_e)_j \geq \overline{\rho}_e^\delta := \rho_6$, for all $j = 1, \ldots, m_f$, with

$$\begin{aligned} \rho_4 &:= m_c \frac{40\,(y_{\max})^2}{\eta\delta^2}, && \forall\ j = 1, \ldots, m_c && (5.172)\\ \rho_5 &:= m_c \frac{10\,(y_{\max})^2}{\eta\delta^2}, && \forall\ j = 1, \ldots, m_c && (5.173)\\ \rho_6 &:= m_f \frac{10\,(y_{\max})^2}{\eta\delta^2}, && \forall\ j = 1, \ldots, m_f && (5.174) \end{aligned}$$

□

In the next theorem we show that the penalty parameters are bounded, even if we are in the neighborhood of a stationary point. The proof of Zillober [97, 102], see Theorem 5.9 and 2.4.3, respectively, is extended by feasibility constraints $c_j(x),\ j = 1, \ldots, m_f$. Moreover, we omit the assumption $\left(y_c^{(k)}\right)_j = 0$ for all $j \in \overline{J}_c^{(k)}$ and $\left(y_e^{(k)}\right)_j = 0$ for all $j \in \overline{J}_e^{(k)}$.

Theorem 5.3. *Let the assumptions of Theorem 5.1 hold. Consider $d^{(k)} \in \mathbb{R}^{n+m_c+m_f}$ defined by (5.136) and $\eta^{(k)}$ defined by (4.30) with lower bound $\eta > 0$, see Lemma 5.5. Let $\delta^{(k)} := \|z^{(k)} - x^{(k)}\|_2 \neq 0$ hold and let the augmented Lagrangian function be given by (5.73). We denote*

$$\begin{aligned} \vartheta_c^{(k)} &:= \frac{\left\| v_c^{(k)} - y_c^{(k)} \right\|^2}{\left(\delta^{(k)}\right)^2}, && (5.175)\\ \vartheta_e^{(k)} &:= \frac{\left\| v_e^{(k)} - y_e^{(k)} \right\|^2}{\left(\delta^{(k)}\right)^2}. && (5.176) \end{aligned}$$

Let the sequences $\left\{x^{(k)}, y^{(k)}\right\}$ and $\left\{z^{(k)}, v^{(k)}\right\}$ fulfill the following η conditions:

1. *$j \in J_c^{(k)}$, if and only if $c_j^{(k)}\left(z^{(k)}\right) = 0,\ j = 1, \ldots, m_c$.*
2. *$j \in J_e^{(k)}$, if and only if $e_j\left(z^{(k)}\right) = 0,\ j = 1, \ldots, m_f$.*
3. *There exists a $\vartheta \in \mathbb{R}$, such that $\vartheta_c^{(k)} \leq \vartheta$ and $\vartheta_e^{(k)} \leq \vartheta$.*

Then there exists a $\delta_r \in \mathbb{R}^+$, such that

$$\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} < -\frac{\eta\left(\delta^{(k)}\right)^2}{2} < 0 \qquad (5.177)$$

holds for all $\left(x^{(k)}, y^{(k)}\right)$, $\delta^{(k)} \leq \delta_r$, $\rho \geq \rho_r$, *with* $\rho_r := \begin{pmatrix} \rho_{r_c} \\ \rho_{r_e} \end{pmatrix}$, $\rho_{r_c} \in \mathbb{R}^{m_c}$, $\rho_{r_e} \in \mathbb{R}^{m_f}$, *and*

$$\min_{j=1,\ldots,m_c+m_f} \left(\rho_r\right)_j \geq \frac{4\vartheta}{\eta}. \tag{5.178}$$

Proof. Due to the first assumption and the complementary conditions (5.76) and (5.77) we know that $\left(v_c^{(k)}\right)_j = 0$ for all $j \in \overline{J}_c^{(k)}$ and $\left(v_e^{(k)}\right)_j = 0$ for all $j \in \overline{J}_e^{(k)}$. We define

$$\underline{v}_c^{(k)} := \left(\left(\underline{v}_c^{(k)}\right)_1, \ldots, \left(\underline{v}_c^{(k)}\right)_{m_c}\right)^T \quad \text{with} \quad \left(\underline{v}_c^{(k)}\right)_j := \begin{cases} 0, & \text{if } j \in J_c^{(k)} \\ \left(v_c^{(k)}\right)_j, & \text{otherwise} \end{cases} \tag{5.179}$$

$$\underline{v}_e^{(k)} := \left(\left(\underline{v}_e^{(k)}\right)_1, \ldots, \left(\underline{v}_e^{(k)}\right)_{m_f}\right)^T \quad \text{with} \quad \left(\underline{v}_e^{(k)}\right)_j := \begin{cases} 0, & \text{if } j \in J_e^{(k)} \\ \left(v_e^{(k)}\right)_j, & \text{otherwise} \end{cases} \tag{5.180}$$

$$\underline{y}_c^{(k)} := \left(\left(\underline{y}_c^{(k)}\right)_1, \ldots, \left(\underline{y}_c^{(k)}\right)_{m_c}\right)^T \quad \text{with} \quad \left(\underline{y}_c^{(k)}\right)_j := \begin{cases} 0, & \text{if } j \in J_c^{(k)} \\ \left(y_c^{(k)}\right)_j, & \text{otherwise} \end{cases} \tag{5.181}$$

$$\underline{y}_e^{(k)} := \left(\left(\underline{y}_e^{(k)}\right)_1, \ldots, \left(\underline{y}_e^{(k)}\right)_{m_f}\right)^T \quad \text{with} \quad \left(\underline{y}_e^{(k)}\right)_j := \begin{cases} 0, & \text{if } j \in J_e^{(k)} \\ \left(y_e^{(k)}\right)_j, & \text{otherwise} \end{cases} \tag{5.182}$$

We proceed from (5.144)

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

$$
\begin{aligned}
&\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
&\leq \quad \underbrace{-\nabla R_{f^{(k)}}\left(z^{(k)}\right)^T \Delta x^{(k)}}_{\leq -\eta^{(k)}\left(\delta^{(k)}\right)^2 < -\eta\left(\delta^{(k)}\right)^2,\ (5.147)} \quad + \quad \underbrace{\sum_{j \in J_c^{(k)}} 2c_j\left(x^{(k)}\right)\left(\Delta y_c^{(k)}\right)_j}_{=2\overline{c}\left(x^{(k)}\right)^T\left(\overline{v}_c^{(k)}-\overline{y}_c^{(k)}\right),\ (5.61),\ (5.59)} \quad - \quad \underbrace{\sum_{j \in J_c^{(k)}} \left(\rho_c\right)_j c_j^2\left(x^{(k)}\right)}_{=\overline{c}\left(x^{(k)}\right)^T\Gamma_c\overline{c}\left(x^{(k)}\right),\ (5.61),\ (5.60)} \\
&\quad - \underbrace{\sum_{j \in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j \left(v_c^{(k)}\right)_j}{\left(\rho_c\right)_j}}_{=0} + \sum_{j \in \overline{J}_c^{(k)}} \frac{\left(y_c^{(k)}\right)_j^2}{\left(\rho_c\right)_j} \\
&\quad + \underbrace{\sum_{j \in J_e^{(k)}} 2e_j\left(x^{(k)}\right)\left(\Delta y_e^{(k)}\right)_j}_{=2\overline{e}\left(x^{(k)}\right)^T\left(\overline{v}_e^{(k)}-\overline{y}_e^{(k)}\right),\ (5.62)} - \underbrace{\sum_{j \in J_e^{(k)}} \left(\rho_e\right)_j e_j^2\left(x^{(k)}\right)}_{=\overline{e}\left(x^{(k)}\right)^T\Gamma_e\overline{e}\left(x^{(k)}\right),\ (5.62)} \\
&\quad - \underbrace{\sum_{j \in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j \left(v_e^{(k)}\right)_j}{\left(\rho_e\right)_j}}_{=0} + \sum_{j \in \overline{J}_e^{(k)}} \frac{\left(y_e^{(k)}\right)_j^2}{\left(\rho_e\right)_j}
\end{aligned}
$$

Due to assumption, $\left(v_c^{(k)}\right)_j = 0$ holds for all $j \in \overline{J}_c^{(k)}$ and $\left(v_e^{(k)}\right)_j = 0$ holds for all $j \in \overline{J}_e^{(k)}$. Using

$$
\begin{aligned}
\left\|\Gamma_c^{1/2}\overline{c}\left(x^{(k)}\right) - \Gamma_c^{-1/2}\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)\right\|^2 \quad = \quad & \overline{c}\left(x^{(k)}\right)^T \Gamma_c \overline{c}\left(x^{(k)}\right) \\
& + \left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)^T \Gamma_c^{-1}\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right) \\
& -2\overline{c}\left(x^{(k)}\right)^T\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)
\end{aligned}
$$

we get

$$
\begin{aligned}
& 2\overline{c}\left(x^{(k)}\right)^T\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right) - \overline{c}\left(x^{(k)}\right)^T \Gamma_c \overline{c}\left(x^{(k)}\right) \qquad (5.183) \\
& \quad = \quad -\left\|\Gamma_c^{1/2}\overline{c}\left(x^{(k)}\right) - \Gamma_c^{-1/2}\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)\right\|^2 + \left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)^T \Gamma_c^{-1}\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)
\end{aligned}
$$

The same holds for the constraints $e_j(x),\ j = 1, \ldots, m_f$, i.e.,

$$
\begin{aligned}
& 2\overline{e}\left(x^{(k)}\right)^T\left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right) - \overline{e}\left(x^{(k)}\right)^T \Gamma_e \overline{e}\left(x^{(k)}\right) \qquad (5.184) \\
& \quad = \quad -\left\|\Gamma_e^{1/2}\overline{e}\left(x^{(k)}\right) - \Gamma_e^{-1/2}\left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right)\right\|^2 + \left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right)^T \Gamma_e^{-1}\left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right)
\end{aligned}
$$

All together this leads to

$$
\begin{aligned}
&\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
<\;& -\eta \left(\delta^{(k)}\right)^2 + \underbrace{2\overline{c}\left(x^{(k)}\right)^T \left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right) - \overline{c}\left(x^{(k)}\right)^T \Gamma_c \overline{c}\left(x^{(k)}\right)}_{=-\left\|\Gamma_c^{1/2}\overline{c}\left(x^{(k)}\right)-\Gamma_c^{-1/2}\left(\overline{v}_c^{(k)}-\overline{y}_c^{(k)}\right)\right\|^2+\left(\overline{v}_c^{(k)}-\overline{y}_c^{(k)}\right)^T\Gamma_c^{-1}\left(\overline{v}_c^{(k)}-\overline{y}_c^{(k)}\right),\ (5.183)} \\
&+ \sum_{j\in\overline{J}_c^{(k)}} \underbrace{\frac{\left(y_c^{(k)}\right)_j^2}{\left(\rho_c\right)_j}}_{=\frac{\left(v_c^{(k)}-y_c^{(k)}\right)_j^2}{\left(\rho_c\right)_j},\ \left(v_c^{(k)}\right)_j=0} \\
&+ \underbrace{2\overline{e}\left(x^{(k)}\right)^T \left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right) - \overline{e}\left(x^{(k)}\right)^T \Gamma_e \overline{e}\left(x^{(k)}\right)}_{=-\left\|\Gamma_e^{1/2}\overline{e}\left(x^{(k)}\right)-\Gamma_e^{-1/2}\left(\overline{v}_e^{(k)}-\overline{y}_e^{(k)}\right)\right\|^2+\left(\overline{v}_e^{(k)}-\overline{y}_e^{(k)}\right)^T\Gamma_e^{-1}\left(\overline{v}_e^{(k)}-\overline{y}_e^{(k)}\right),\ (5.184)} \\
&+ \sum_{j\in\overline{J}_e^{(k)}} \underbrace{\frac{\left(y_e^{(k)}\right)_j^2}{\left(\rho_e\right)_j}}_{=\frac{\left(v_e^{(k)}-y_e^{(k)}\right)_j^2}{\left(\rho_e\right)_j},\ \left(v_e^{(k)}\right)_j=0} \\
=\;& -\eta \left(\delta^{(k)}\right)^2 - \underbrace{\left\|\Gamma_c^{1/2}\overline{c}\left(x^{(k)}\right) - \Gamma_c^{-1/2}\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)\right\|^2}_{\geq 0} \\
&+ \left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)^T \Gamma_c^{-1} \left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right) + \underbrace{\sum_{j\in\overline{J}_c^{(k)}} \frac{\left(v_c^{(k)} - y_c^{(k)}\right)_j^2}{\left(\rho_c\right)_j}}_{=\left(\underline{v}_c^{(k)}-\underline{y}_c^{(k)}\right)^T\Gamma_c^{-1}\left(\underline{v}_c^{(k)}-\underline{y}_c^{(k)}\right)} \\
&- \underbrace{\left\|\Gamma_e^{1/2}\overline{e}\left(x^{(k)}\right) - \Gamma_e^{-1/2}\left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right)\right\|^2}_{\geq 0} \\
&+ \left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right)^T \Gamma_e^{-1} \left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right) + \underbrace{\sum_{j\in\overline{J}_e^{(k)}} \frac{\left(v_e^{(k)} - y_e^{(k)}\right)_j^2}{\left(\rho_e\right)_j}}_{=\left(\underline{v}_e^{(k)}-\underline{y}_e^{(k)}\right)^T\Gamma_e^{-1}\left(\underline{v}_e^{(k)}-\underline{y}_e^{(k)}\right)}
\end{aligned}
$$

$$
\begin{aligned}
&\nabla\Phi_\rho \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} \\
<\ & -\eta \left(\delta^{(k)}\right)^2 \\
& + \underbrace{\left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right)^T \Gamma_c^{-1} \left(\overline{v}_c^{(k)} - \overline{y}_c^{(k)}\right) + \left(\underline{v}_c^{(k)} - \underline{y}_c^{(k)}\right)^T \Gamma_c^{-1} \left(\underline{v}_c^{(k)} - \underline{y}_c^{(k)}\right)}_{=\left(v_c^{(k)} - y_c^{(k)}\right)^T \Gamma_c^{-1} \left(v_c^{(k)} - y_c^{(k)}\right),\ (5.57),(5.59),(5.179) \text{ and } (5.181)} \\
& + \underbrace{\left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right)^T \Gamma_e^{-1} \left(\overline{v}_e^{(k)} - \overline{y}_e^{(k)}\right) + \left(\underline{v}_e^{(k)} - \underline{y}_e^{(k)}\right)^T \Gamma_e^{-1} \left(\underline{v}_e^{(k)} - \underline{y}_e^{(k)}\right)}_{=\left(v_e^{(k)} - y_e^{(k)}\right)^T \Gamma_e^{-1} \left(v_e^{(k)} - y_e^{(k)}\right),\ (5.58),(5.60),(5.180) \text{ and } (5.182)} \\
=\ & -\eta \left(\delta^{(k)}\right)^2 + \left(v_c^{(k)} - y_c^{(k)}\right)^T \Gamma_c^{-1} \left(v_c^{(k)} - y_c^{(k)}\right) + \left(v_e^{(k)} - y_e^{(k)}\right)^T \Gamma_e^{-1} \left(v_e^{(k)} - y_e^{(k)}\right) \\
\leq\ & -\eta \left(\delta^{(k)}\right)^2 + \left\|\Gamma_c^{-1}\right\| \underbrace{\left\|v_c^{(k)} - y_c^{(k)}\right\|^2}_{=\vartheta_c^{(k)}\left(\delta^{(k)}\right)^2,\ (5.175)} + \left\|\Gamma_e^{-1}\right\| \underbrace{\left\|v_e^{(k)} - y_e^{(k)}\right\|^2}_{=\vartheta_e^{(k)}\left(\delta^{(k)}\right)^2,\ (5.176)} \\
=\ & -\eta \left(\delta^{(k)}\right)^2 + \underbrace{\left\|\Gamma_c^{-1}\right\|}_{\leq \frac{\eta}{4\vartheta},\ (5.178)} \underbrace{\vartheta_c^{(k)} \left(\delta^{(k)}\right)^2}_{\leq\vartheta\left(\delta^{(k)}\right)^2} + \underbrace{\left\|\Gamma_e^{-1}\right\|}_{\leq \frac{\eta}{4\vartheta},\ (5.178)} \underbrace{\vartheta_e^{(k)} \left(\delta^{(k)}\right)^2}_{\leq\vartheta\left(\delta^{(k)}\right)^2} \\
\leq\ & -\eta \left(\delta^{(k)}\right)^2 + \frac{1}{4}\eta \left(\delta^{(k)}\right)^2 + \frac{1}{4}\eta \left(\delta^{(k)}\right)^2 \\
=\ & -\frac{1}{2}\eta \left(\delta^{(k)}\right)^2
\end{aligned}
$$

□

The first and the second assumption ensure that the set of active constraints is identical for the subproblem and the original problem. Close to a stationary point, this is no restriction. The third assumption requires that the ratio of the change in the dual variables and in the primal variables is bounded, i.e., if the change in the primal variables is small the same holds for the dual variables. This cannot be guaranteed in practice.

As a consequence of Theorem 5.2 and Theorem 5.3, there exists a penalty parameter $\overline{\rho}_r \in \mathbb{R}^{m_c+m_f}$ such that a descent in the augmented Lagrangian function is obtained without increasing the penalty parameters. The penalty update presented in Algorithm 13 ensures that $\overline{\rho}_r$ is reached. Thereafter the penalty parameters are not adapted anymore. These results are summarized in the next corollary, see Zillober [102] Corollary 2.4.4 and Zillober [97] Corollary 5.10.

Corollary 5.1. *Let the sequence $\left\{x^{(k)}, y^{(k)}\right\}$ be computed by Algorithm 16. Let the assumptions of Theorem 5.2 and Theorem 5.3 be valid. Then the sequence of penalty parameter vectors is bounded, i.e., there exists a $\overline{\rho}_r \in \mathbb{R}^{m_c+m_f}$ such that $\rho^{(k)} \leq \overline{\rho}_r$ for all $k = 0, 1, \ldots$. As a consequence there exists a $\overline{k} \in \mathbb{N}$ such that $\rho^{(k)} = \overline{\rho}_r$ for all $k \geq \overline{k} \geq 0$.*

To prove the main convergence Theorem 5.5, it is essential to show that there exist a subsequence with $\left\|\Delta x^{(k)}\right\| \leq \varepsilon$, see Schittkowski [73].

Theorem 5.4. *Let the assumptions of Theorem 5.3 hold. Let $\left\{x^{(k)}, y^{(k)}\right\}$ be computed by Algorithm 16. Then there exists for each $\varepsilon > 0$ at least one k such that*

$$\left\|\Delta x^{(k)}\right\| \leq \varepsilon. \tag{5.185}$$

Proof. As a consequence of Theorem 5.2 and Theorem 5.3 the penalty parameters are bounded, see Corollary 5.1. We prove by contradiction, that (5.185) holds for at least one k. We assume that $\left\|\Delta x^{(k)}\right\| > \varepsilon$ for a fixed ε and each k. The penalty parameters are constant after a certain iteration $\overline{k}$, see Corollary 5.1. 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,

$$\nabla \Phi_{\overline{\rho}_r} \left(\begin{array}{c} x^{(k)} \\ y^{(k)} \end{array} \right)^T d^{(k)} < -\frac{\eta \varepsilon^2}{2} < 0 \tag{5.186}$$

holds with $d^{(k)} := \left(\begin{array}{c} z^{(k)} - x^{(k)} \\ v^{(k)} - y^{(k)} \end{array} \right) \neq \mathbb{0}$, see Theorem 5.2 and Theorem 5.3. There exists a i_0 independent from k such that the Armijo condition (4.28) is satisfied for all $i \geq i_0$, see Schittkowski [73]. As a consequence $\sigma^{(k)} \geq \underline{\sigma} := \beta^{i_0}$ and we get

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

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

$$\lim_{k \to \infty} \Phi_{\overline{\rho}_r} \left(\begin{array}{c} x^{(k)} \\ y^{(k)} \end{array} \right) = -\infty, \tag{5.187}$$

which is a contradiction to Lemma 5.7. Therefore, the assumption is wrong and (5.185) holds for at least one k.

□

With these results it is possible to formulate and prove the main convergence theorem, see Zillober [97] Theorem 5.12, and Schittkowski [73] respectively.

Theorem 5.5. *Let the assumptions of Theorem 5.3 hold. Let* $\left\{x^{(k)}\right\}$, $\left\{y^{(k)}\right\}$, $\left\{z^{(k)}\right\}$, $\left\{v^{(k)}\right\}$, *be computed by Algorithm 16. Then there exists an accumulation point* $(x^\star, v^\star)$ *of* $\left\{x^{(k)}, v^{(k)}\right\}$ *satisfying the KKT conditions for problem (5.16).*

Proof. Theorem 5.1 shows that each element of the sequence $\left\{x^{(k)}, v^{(k)}\right\}$ lies in the compact 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 (5.18). The results of Theorem 5.4 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

$$\lim_{k\in S} \Delta x^{(k)} = 0, \tag{5.188}$$

$$\lim_{k\in S} x^{(k)} = x^\star, \tag{5.189}$$

$$\lim_{k\in S} v^{(k)} = v^\star. \tag{5.190}$$

We will show that the KKT conditions are satisfied for $x^\star$ if $\Delta x_i^{(k)} \longrightarrow 0$, $k \in S$ holds. We consider the KKT system of the subproblem

$$\begin{aligned} \nabla_x L^{(k)}\left(z^{(k)}, v^{(k)}\right) &= \nabla f^{(k)}\left(z^{(k)}\right) + A_{c^{(k)}}\left(z^{(k)}\right) v_c^{(k)} \\ &\quad + A_e\left(z^{(k)}\right) v_e^{(k)} + v_u^{(k)} - v_l^{(k)} = 0 \end{aligned} \tag{5.191}$$

$$\left(v_c^{(k)}\right)_j c_j^{(k)}\left(z^{(k)}\right) = 0, \qquad j = 1, \ldots, m_c \tag{5.192}$$

$$\left(v_e^{(k)}\right)_j e_j\left(z^{(k)}\right) = 0, \qquad j = 1, \ldots, m_f \tag{5.193}$$

$$\left(v_b^{(k)}\right)_j b_j^{(k)}\left(z^{(k)}\right) = 0, \qquad j = 1, \ldots, 2n \tag{5.194}$$

$$c_j^{(k)}\left(z^{(k)}\right) \le 0, \qquad j = 1, \ldots, m_c \tag{5.195}$$

$$e_j\left(z^{(k)}\right) \le 0, \qquad j = 1, \ldots, m_f \tag{5.196}$$

$$b_j^{(k)}\left(z^{(k)}\right) \le 0, \qquad j = 1, \ldots, 2n \tag{5.197}$$

$$\left(v_c^{(k)}\right)_j \ge 0, \qquad j = 1, \ldots, m_c \tag{5.198}$$

$$\left(v_e^{(k)}\right)_j \ge 0, \qquad j = 1, \ldots, m_f \tag{5.199}$$

$$\left(v_b^{(k)}\right)_j \ge 0, \qquad j = 1, \ldots, 2n \tag{5.200}$$

We prove that $\left|c_j^{(k)}\left(z^{(k)}\right) - c_j\left(x^{(k)}\right)\right| \longrightarrow 0$ holds, for $\Delta x_i^{(k)} \longrightarrow 0$, $k \in S$, with $z^{(k)} = x^{(k)} + \Delta x^{(k)}$.

$$\begin{aligned}
&\left|c_j^{(k)}\left(z^{(k)}\right) - c_j\left(x^{(k)}\right)\right| \\
=\quad & \Bigg| c_j\left(x^{(k)}\right) \\
&+ \sum_{I_+^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{1}{U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}} - \frac{1}{U_i^{(k)} - x_i^{(k)}} \right) \right] \\
&- \sum_{I_-^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{1}{x_i^{(k)} + \Delta x_i^{(k)} - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}} \right) \right] \\
&- c_j\left(x^{(k)}\right) \Bigg| \\
=\quad & \Bigg| \sum_{I_+^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right)^2 \left(\frac{\left(U_i^{(k)} - x_i^{(k)}\right) - \left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)\left(U_i^{(k)} - x_i^{(k)}\right)} \right) \right] \\
&- \sum_{I_-^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right)^2 \left(\frac{\left(x_i^{(k)} - L_i^{(k)}\right) - \left(x_i^{(k)} + \Delta x_i^{(k)} - L_i^{(k)}\right)}{\left(x_i^{(k)} + \Delta x_i^{(k)} - L_i^{(k)}\right)\left(x_i^{(k)} - L_i^{(k)}\right)} \right) \right] \Bigg| \\
=\quad & \Bigg| \sum_{I_+^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(U_i^{(k)} - x_i^{(k)}\right) \left(\frac{\Delta x_i^{(k)}}{U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}} \right) \right] \\
&- \sum_{I_-^{(j,k)}} \left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(x_i^{(k)} - L_i^{(k)}\right) \left(\frac{-\Delta x_i^{(k)}}{x_i^{(k)} - L_i^{(k)} + \Delta x_i^{(k)}} \right) \right] \Bigg|
\end{aligned}$$

As the asymptotes $U_i^{(k)}$ and $L_i^{(k)}$, $i = 1, \ldots, n$, are feasible, there exists a fixed ξ such that

$$0 < \xi \leq x_i^{(k)} - L_i^{(k)}, \tag{5.201}$$

$$0 < \xi \leq U_i^{(k)} - x_i^{(k)} \tag{5.202}$$

holds for all $i = 1, \ldots, n$ and $k = 0, 1, \ldots$.

Moreover, $\left|\Delta x_i^{(k)}\right| \leq \frac{1}{2}\xi$ holds for $k \in S$ sufficiently large. We get

$$\begin{aligned}
&\left|c_j^{(k)}\left(z^{(k)}\right) - c_j\left(x^{(k)}\right)\right| \\
&= \left|\sum_{I_+^{(j,k)}}\left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(U_i^{(k)} - x_i^{(k)}\right)\left(\frac{\Delta x_i^{(k)}}{U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}}\right)\right]\right. \\
&\quad \left. - \sum_{I_-^{(j,k)}}\left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(x_i^{(k)} - L_i^{(k)}\right)\left(\frac{-\Delta x_i^{(k)}}{x_i^{(k)} - L_i^{(k)} + \Delta x_i^{(k)}}\right)\right]\right| \\
&\leq \sum_{I_+^{(j,k)}}\left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(U_i^{(k)} - x_i^{(k)}\right)\left(\frac{\left|\Delta x_i^{(k)}\right|}{\xi - \Delta x_i^{(k)}}\right)\right] \\
&\quad + \sum_{I_-^{(j,k)}}\left[\left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right|\left(x_i^{(k)} - L_i^{(k)}\right)\left(\frac{\left|\Delta x_i^{(k)}\right|}{\xi + \Delta x_i^{(k)}}\right)\right] \\
&\leq \sum_{I_+^{(j,k)}}\left[\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(U_i^{(k)} - x_i^{(k)}\right)\left(\frac{2\left|\Delta x_i^{(k)}\right|}{\xi}\right)\right] \\
&\quad + \sum_{I_-^{(j,k)}}\left[\left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right|\left(x_i^{(k)} - L_i^{(k)}\right)\left(\frac{2\left|\Delta x_i^{(k)}\right|}{\xi}\right)\right]
\end{aligned}$$

As $\left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right|$, $\left(U_i^{(k)} - x_i^{(k)}\right)$, $\left(x_i^{(k)} - L_i^{(k)}\right)$ are bounded and $\Delta x_i^{(k)} \longrightarrow 0$ holds, we get $\left|c_j^{(k)}\left(z^{(k)}\right) - c_j\left(x^{(k)}\right)\right| \longrightarrow 0$. As the KKT conditions hold for $c_j^{(k)}\left(z^{(k)}\right)$, i.e., $c_j^{(k)}\left(z^{(k)}\right) \leq 0$, we get $c_j\left(x^\star\right) \leq 0$.

Moreover, we consider the box constraints of the subproblem for $k \in S$.

$$b_u^{(k)}(x) = x - \overline{x}^{(k)} \leq 0 \tag{5.203}$$

$$b_l^{(k)}(x) = \underline{x}^{(k)} - x \leq 0 \tag{5.204}$$

The definitions of $\underline{x}_i^{(k)}$ and $\overline{x}_i^{(k)}$, $i = 1, \ldots, n$ given by (5.20) and (5.22) lead to

$$\begin{aligned}
b_i^{(k)}(x^{(k)} + \Delta x^{(k)}) &= x_i^{(k)} + \Delta x_i^{(k)} - \overline{x}_i^{(k)} \\
&\overset{(5.22)}{=} x_i^{(k)} + \Delta x_i^{(k)} - x_i^{(k)} - \omega \underbrace{\left(U_i^{(k)} - x_i^{(k)}\right)}_{\geq \xi} \\
&\leq \Delta x_i^{(k)} - \omega\xi \\
&\leq \frac{1}{2}\omega\xi - \omega\xi \;<\; 0 \qquad (5.205)\\
b_{n+i}^{(k)}(x^{(k)} + \Delta x^{(k)}) &= \underline{x}_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)} \\
&\overset{(5.20)}{=} x_i^{(k)} - \omega \underbrace{\left(x_i^{(k)} - L_i^{(k)}\right)}_{\geq \xi} - x_i^{(k)} - \Delta x_i^{(k)} \\
&\leq -\omega\xi - \Delta x_i^{(k)} \\
&\leq -\omega\xi + \frac{1}{2}\omega\xi \;<\; 0 \qquad (5.206)
\end{aligned}$$

for $\left|\Delta x_i^{(k)}\right| \leq \frac{1}{2}\omega\xi$ and $i = 1, \ldots, n$. The complementary condition (5.194) is satisfied since

$$v_u^{(k)} = v_l^{(k)} = 0. \qquad (5.207)$$

It is easy to see that $e_j(x^\star) \leq 0$, $j = 1, \ldots, m_f$, and $v_e^\star \geq 0$, due to (5.196) and (5.199). From (5.198) we get

$$(v_c^\star)_j \;\geq\; 0, \quad j = 1, \ldots, m_c. \qquad (5.208)$$

The complementarity conditions are satisfied as (5.192) and (5.193) holds, i.e.,

$$\begin{aligned}
(v_c^\star)_j \, c_j(x^\star) &= 0, \quad j = 1, \ldots, m_c \qquad (5.209)\\
(v_e^\star)_j \, e_j(x^\star) &= 0, \quad j = 1, \ldots, m_f. \qquad (5.210)
\end{aligned}$$

To show that $\nabla_x L(x^\star, v^\star) = 0$, we consider the gradients of the constraints $c_j^{(k)}(z^{(k)})$, $j = 1, \ldots, m_c$, with $k \in S$, see (4.23),

$$\frac{\partial c_j^{(k)}\left(z^{(k)}\right)}{\partial x_i} = \begin{cases} \dfrac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \dfrac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - z_i^{(k)}\right)^2}, & \text{if } i \in I_+^{(j,k)} \\[2ex] \dfrac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \dfrac{\left(x_i^{(k)} - L_i^{(k)}\right)^2}{\left(z_i^{(k)} - L_i^{(k)}\right)^2}, & \text{otherwise.} \end{cases}$$

We consider the index set $I_+^{(j,k)}$, i.e.,

$$
\begin{aligned}
&\frac{\partial c_j^{(k)}\left(z^{(k)}\right)}{\partial x_i} \\
= {} & \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} \\
= {} & \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(\frac{\left(U_i^{(k)} - x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} - \frac{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} \right) \\
& + \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \frac{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} \\
= {} & \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(\frac{\left(U_i^{(k)}\right)^2 + \left(x_i^{(k)}\right)^2 - 2U_i^{(k)} x_i^{(k)}}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} \right. \\
& + \frac{2U_i^{(k)} x_i^{(k)} - 2\Delta x_i^{(k)} x_i^{(k)} + 2\Delta x_i^{(k)} U_i^{(k)} - \left(U_i^{(k)}\right)^2 - \left(x_i^{(k)}\right)^2 - \left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} \\
& \left. + 1 \right) \\
= {} & \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(\frac{2\Delta x_i^{(k)} U_i^{(k)} - 2\Delta x_i^{(k)} x_i^{(k)} - \left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} + 1 \right) \\
= {} & \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(\frac{2\Delta x_i^{(k)} \left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right) + \left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} + 1 \right) \\
= {} & \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i} \left(\frac{2\Delta x_i^{(k)}}{U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}} + \frac{\left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} + 1 \right)
\end{aligned}
$$

As the sequence of asymptotes is feasible and $\left|\Delta x_i^{(k)}\right| \leq \frac{1}{2}\xi$ holds for $k \in S$ sufficiently large, we get

$$\begin{aligned}
&\left|\frac{\partial c_j^{(k)}\left(z^{(k)}\right)}{\partial x_i} - \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right| \\
&= \left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(\frac{2\Delta x_i^{(k)}}{U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}} + \frac{\left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)} - x_i^{(k)} - \Delta x_i^{(k)}\right)^2} + 1\right)\right. \\
&\qquad \left. - \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right| \\
&\leq \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(\frac{2\left|\Delta x_i^{(k)}\right|}{\xi - \Delta x_i^{(k)}} + \frac{\left(\Delta x_i^{(k)}\right)^2}{\left(\xi - \Delta x_i^{(k)}\right)^2}\right) \\
&\leq \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\left(\frac{4\left|\Delta x_i^{(k)}\right|}{\xi} + \frac{4\left(\Delta x_i^{(k)}\right)^2}{\xi^2}\right)
\end{aligned}$$

The gradients for the index set $I_-^{(j,k)}$ can be computed analogously. We get

$$\left|\frac{\partial c_j^{(k)}\left(z^{(k)}\right)}{\partial x_i} - \frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right| \leq \left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right|\left(\frac{4\left|\Delta x_i^{(k)}\right|}{\xi} + \frac{4\left(\Delta x_i^{(k)}\right)^2}{\xi^2}\right)$$

As $\left|\frac{\partial c_j\left(x^{(k)}\right)}{\partial x_i}\right|$ is bounded and $\Delta x_i^{(k)} \longrightarrow 0$ holds, we get for $k \in S$

$$\left|\frac{\partial c_j^{(k)}\left(z^{(k)}\right)}{\partial x_i} - \frac{\partial c_j\left(x^\star\right)}{\partial x_i}\right| \longrightarrow 0. \tag{5.211}$$

Moreover, we consider the gradient of the approximated objective function $f^{(k)}(z^{(k)})$ with $k \in S$,

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

The previous calculations equivalently hold for the first term. Moreover, we consider the term dependent on τ. For $i \in I_+^{(k)}$ we get

$$\begin{aligned}
&\tau\left|\frac{\left(x_i^{(k)}+\Delta x_i^{(k)}-x_i^{(k)}\right)^2+2\left(x_i^{(k)}+\Delta x_i^{(k)}-x_i^{(k)}\right)\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)^2}\right| \\
&= \tau\left|\frac{\left(\Delta x_i^{(k)}\right)^2+2\Delta x_i^{(k)}\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)^2}\right| \\
&= \tau\left|\frac{\left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)^2}+\frac{2\Delta x_i^{(k)}\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)^2}\right| \\
&= \tau\left|\frac{\left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)^2}+\frac{2\Delta x_i^{(k)}}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)}\right|
\end{aligned}$$

For $k \in S$ sufficiently large, $\left|\Delta x_i^{(k)}\right| \leq \frac{1}{2}\xi$ holds. Moreover, the sequence of asymptotes is feasible. This leads to

$$\begin{aligned}
&\tau\left|\frac{\left(\Delta x_i^{(k)}\right)^2}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)^2}+\frac{2\Delta x_i^{(k)}}{\left(U_i^{(k)}-x_i^{(k)}-\Delta x_i^{(k)}\right)}\right| \\
&\leq \tau\left(\frac{\left(\Delta x_i^{(k)}\right)^2}{\left(\xi-\Delta x_i^{(k)}\right)^2}+\frac{2\left|\Delta x_i^{(k)}\right|}{\left(\xi-\Delta x_i^{(k)}\right)}\right) \\
&\leq \tau\left(\frac{4\left(\Delta x_i^{(k)}\right)^2}{\xi^2}+\frac{4\left|\Delta x_i^{(k)}\right|}{\xi}\right)
\end{aligned}$$

And analogously for $i \in I_-^{(k)}$

$$\begin{aligned}
&\tau\quad\left|-\frac{\left(x_i^{(k)}+\Delta x_i^{(k)}-x_i^{(k)}\right)^2-2\left(x_i^{(k)}+\Delta x_i^{(k)}-x_i^{(k)}\right)\left(x_i^{(k)}+\Delta x_i^{(k)}-L_i^{(k)}\right)}{\left(x_i^{(k)}+\Delta x_i^{(k)}-L_i^{(k)}\right)^2}\right| \\
&\leq \tau\left(\frac{4\left(\Delta x_i^{(k)}\right)^2}{\xi^2}+\frac{4\left|\Delta x_i^{(k)}\right|}{\xi}\right)
\end{aligned}$$

As $\left|\frac{\partial f\left(x^{(k)}\right)}{\partial x_i}\right|$ is bounded and $\Delta x_i^{(k)} \longrightarrow 0$ holds, we get for $k \in S$,

$$\left|\frac{\partial f^{(k)}\left(z^{(k)}\right)}{\partial x_i} - \frac{\partial f\left(x^\star\right)}{\partial x_i}\right| \longrightarrow 0. \tag{5.212}$$

All together this leads to

$$\begin{aligned}
\|\nabla_x L(x^\star, v^\star)\| &= \left\| \underbrace{\nabla_x L^{(k)}(z^{(k)}, v^{(k)})}_{=0} - \nabla_x L(x^\star, v^\star) \right\| \\
&= \Big\| \nabla f^{(k)}\left(z^{(k)}\right) + A_{c^{(k)}}\left(z^{(k)}\right) v_c^{(k)} + A_e\left(z^{(k)}\right) v_e^{(k)} + \underbrace{v_u^{(k)} - v_l^{(k)}}_{=0,\ (5.207)} \\
&\quad -\nabla f\left(x^\star\right) - A_c\left(x^\star\right) v_c^\star - A_e\left(x^\star\right) v_e^\star \Big\| \\
&\leq \underbrace{\left\| \nabla f^{(k)}\left(z^{(k)}\right) - \nabla f\left(x^\star\right) \right\|}_{\longrightarrow 0,\ (5.212)} + \sum_{j=1}^{m_c} \underbrace{\left\| \nabla c_j^{(k)}\left(z^{(k)}\right)\left(v_c^{(k)}\right)_j - \nabla c_j\left(x^\star\right)\left(v_c^\star\right)_j \right\|}_{\longrightarrow 0,\ (5.211)} \\
&\quad + \sum_{j=1}^{m_f} \underbrace{\left\| \nabla e_j\left(z^{(k)}\right)\left(v_e^{(k)}\right)_j - \nabla e_j\left(x^\star\right)\left(v_e^\star\right)_j \right\|}_{\longrightarrow 0,\ \Delta x^{(k)} \longrightarrow 0}
\end{aligned}$$

We have shown that $\|\nabla_x L(x^\star, v^\star)\| = 0$. Together with the previous results, the KKT conditions (2.12) - (2.19) are satisfied for $(x^\star, v^\star)$. □

6. FREE MATERIAL OPTIMIZATION

The goal of free material optimization (FMO), see Bendsøe et. al. [7] and Zowe, Kočvara and Bendsøe [107], 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 [50], 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 [47].

FMO was first introduced by Bendsøe et al. [7], Bendsøe and Díaz [6], Bendsøe [5] and Zowe, Kočvara and Bendsøe [107]. The continuous problem formulation leads to a saddle-point problem for which the existence of a solution can be shown, see Mach [56] and Werner [91]. 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.

In this section, we use the following notation. 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 $\mathbb{S}^p_{++}$.

6.1 Theory and Problem Formulation

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 [91] and for the FE theory to, e.g., Mach [56].

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. Figure 6.1 shows an example of a design space and the corresponding finite element discretization. A single load is acting at the upper right corner and the nodes on the left hand side are fixed, i.e., they are not allowed to move in any direction. They are denoted by ▶. In each finite element, we determine the material properties and thus identify the corresponding material.

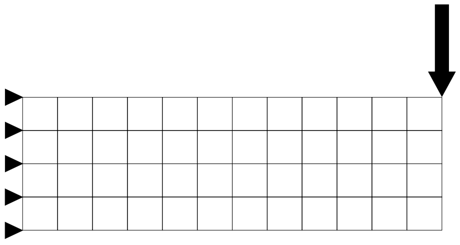


Fig. 6.1: Design space given by a finite element discretization.

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

$$E := \begin{pmatrix} E_1 & & & 0 \\ & E_2 & & \\ & & \ddots & \\ 0 & & & E_m \end{pmatrix} \tag{6.1}$$

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

$$E_i := \begin{pmatrix} e_{i1} & e_{i2} & e_{i4} \\ e_{i2} & e_{i3} & e_{i5} \\ e_{i4} & e_{i5} & e_{i6} \end{pmatrix} \succeq 0, \quad i = 1, \ldots, m, \tag{6.2}$$

see, e.g., Werner [91] 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

$$E_i := \begin{pmatrix} e_{i1} & e_{i2} & e_{i4} & e_{i7} & e_{i11} & e_{i16} \\ e_{i2} & e_{i3} & e_{i5} & e_{i8} & e_{i12} & e_{i17} \\ e_{i4} & e_{i5} & e_{i6} & e_{i9} & e_{i13} & e_{i18} \\ e_{i7} & e_{i8} & e_{i9} & e_{i10} & e_{i14} & e_{i19} \\ e_{i11} & e_{i12} & e_{i13} & e_{i14} & e_{i15} & e_{i20} \\ e_{i16} & e_{i17} & e_{i18} & e_{i19} & e_{i20} & e_{i21} \end{pmatrix} \succeq 0, \quad i = 1, \ldots, m. \tag{6.3}$$

Therefore,

$$E \in \mathbb{S}_+^{3m}, \tag{6.4}$$

$$E_i \in \mathbb{S}_+^3, \quad i = 1, \ldots, m \tag{6.5}$$

holds for the two dimensional case and

$$
\begin{aligned}
E &\in \mathbb{S}_{+}^{6m}, && (6.6)\\
E_i &\in \mathbb{S}_{+}^{6}, \quad i = 1, \ldots, m && (6.7)
\end{aligned}
$$

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 [18],

$$
\begin{aligned}
K(E) &:= \sum_{i=1}^{m} K_i(E) && (6.8)\\
K_i(E) &:= \sum_{k=1}^{n_g} B_{i,k}^T E_i B_{i,k} && (6.9)
\end{aligned}
$$

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 [40], Kočvara and Zowe [51], Zowe, Kočvara and Bendsøe [107]. The behavior of the structure with respect to loads f_j, $j = 1, \ldots, l$, is given by

$$
f_j^T u_j(E), \quad j = 1, \ldots, l, \qquad (6.10)
$$

where $u_j(E) \in \mathbb{R}^{2q}$, $j = 1, \ldots, l$, is the displacement vector. Its value is illustrated in Figure 6.2, given by the sum of the deflection of the nodes caused by the acting forces, which is marked by the red line.

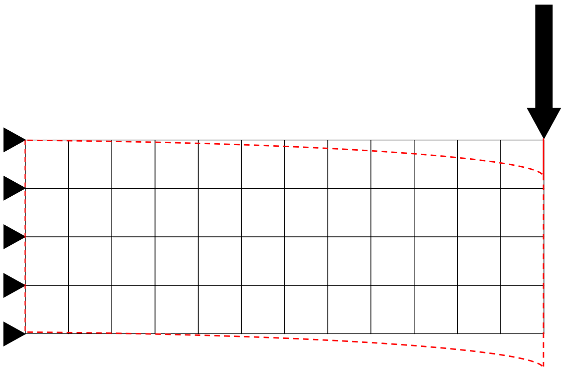


Fig. 6.2: Compliance with respect to a given load.

The displacement vector is determined by the equilibrium condition

$$K(E)\,u_j(E) = f_j, \quad j = 1,\dots,l \tag{6.11}$$

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

$$\begin{aligned} K(E)\,u_j(E) = f_j \iff & u_j(E) = K(E)^{-1} f_j, & j = 1,\dots,l, \quad (6.12)\\ \implies & f_j^T u_j(E) = f_j^T K(E)^{-1} f_j, & j = 1,\dots,l, \quad (6.13) \end{aligned}$$

where $K^{-1}(E)\,f_j$ can be computed by solving the linear system (6.11) 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 $E_i,\ i = 1,\dots,m$ is positive definite, since

$$\begin{aligned} K(E) \in \mathbb{S}^{2r}_{++} \iff & E \in \mathbb{S}^{3m}_{++} & (6.14)\\ \iff & E_i \in \mathbb{S}^{3}_{++}, \quad i = 1,\dots,m & (6.15) \end{aligned}$$

Therefore, we require $E_i - \underline{\nu} I \succeq 0,\ i = 1,\dots,m$, where I is the identity matrix and $\underline{\nu} \in \mathbb{R}^+$ is a small positive value, see Kočvara and Stingl [49]. We have to ensure, that the semidefinite constraints $E_i - \underline{\nu} I \succeq 0$, $i = 1,\dots,m$, are satisfied, whenever the linear system (6.11) 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

$$f_j^T K^{-1}(E)\,f_j \le \alpha, \quad j = 1,\dots,l, \tag{6.16}$$

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

The sum of the diagonal elements of the matrices $E_i,\ i = 1,\dots,m$, is a measure for stiffness of the material in coordinate directions. The trace of $E_i,\ i = 1,\dots,m$, 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

$$\begin{aligned} \sum_{i=1}^{m} \operatorname{Trace}(E_i) \quad & \le \quad V, & (6.17)\\ \operatorname{Trace}(E_i) \quad & := \quad e_{i1} + e_{i3} + e_{i6}, \quad i = 1,\dots,m & (6.18) \end{aligned}$$

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

$$\operatorname{Trace}(E_i) \quad \le \quad \overline{\nu}, \quad i = 1,\dots,m, \tag{6.19}$$

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

$$3\underline{\nu} \le \operatorname{Trace}(E_i), \quad i = 1,\dots,m. \tag{6.20}$$

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

$$\begin{aligned} e_{i1} &\geq \underline{\nu}, && (6.21)\\ e_{i3} &\geq \underline{\nu}, && (6.22)\\ e_{i6} &\geq \underline{\nu}. && (6.23) \end{aligned}$$

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 semidefinite problem

$$\begin{array}{lll} \min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\ \text{s.t.} & \sum\limits_{i=1}^{m} \mathrm{Trace}\,(E_i) - V \leq 0 & \\ & \mathrm{Trace}\,(E_i) - \overline{\nu} \leq 0, & i = 1, \ldots, m \\ & E_i - \underline{\nu} I \succeq 0, & i = 1, \ldots, m \\ & f_j^T K\,(E)^{-1} f_j - \alpha \leq 0, & j = 1, \ldots, l \\ & e_{i1} \geq \underline{\nu}, & i = 1, \ldots, m \\ & e_{i3} \geq \underline{\nu}, & i = 1, \ldots, m \\ & e_{i6} \geq \underline{\nu}, & i = 1, \ldots, m \end{array} \qquad (6.24)$$

The optimization variables are the entries $e_{ip},\ p = 1, \ldots, 6$ of the elementary stiffness matrices $E_i,\ i = 1, \ldots, m$. The derivatives are specified in Ertel, Schittkowski and Zillober [21] for all $i = 1, \ldots, m$, $p = 1, \ldots, 6$ by

$$\frac{\partial}{\partial e_{ip}} \left(\sum_{i=1}^{m} \mathrm{Trace}\,(E_i) - V \right) = \begin{cases} 1, & \text{if } p = 1, 3, 6 \\ 0, & \text{otherwise} \end{cases} \qquad (6.25)$$

$$\frac{\partial}{\partial e_{ip}} \left(\mathrm{Trace}\,(E_i) - \overline{\nu} \right) = \begin{cases} 1, & \text{if } p = 1, 3, 6 \\ 0, & \text{otherwise} \end{cases} \qquad (6.26)$$

$$\frac{\partial}{\partial e_{ip}} \left(f_j^T K\,(E)^{-1} f_j - \alpha \right) = -u_j\,(E)^T \left(\frac{\partial K\,(E)}{\partial e_{ip}} \right) u_j\,(E) \qquad (6.27)$$

$$\begin{aligned} \frac{\partial}{\partial e_{ip}} K\,(E) &= \frac{\partial}{\partial e_{ip}} \left(\sum_{i=1}^{m} \sum_{k=1}^{n_g} B_{i,k}^T E_i B_{i,k} \right) \\ &= \sum_{i=1}^{m} \sum_{k=1}^{n_g} B_{i,k}^T \frac{\partial E_i}{\partial e_{ip}} B_{i,k} \qquad (6.28) \end{aligned}$$

with

$$\frac{\partial E_i}{\partial e_{ip}} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^T \tag{6.29}$$

Moreover, additional constraints can be added to optimization problem (6.24). Especially stress constraints, such as ’von Mises stress conditions’, see, e.g., Li, Steven and Xie [55], are very important from the engineering point of view. In the two dimensional space the von Mises stress in an element $i \in \{1, \ldots, m\}$, and load case $j \in \{1, \ldots, l\}$, can be formulated as

$$s_{i,j}(E) \ := \ \sum_{k=1}^{n_g} u_j(E)^T B_{i,k}^T E_i I E_i B_{i,k} u_j(E) \tag{6.30}$$

with

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.31}$$

see Kočvara and Stingl [50]. The integration of stress constraints lead to numerical problems for the optimization method, as constraint qualifications, such as LICQ see Definition 5 are not satisfied, see Achtziger and Kanzow [1] and Stingl [78].
To ensure stability $s_{i,j}(E)$ may not exceed a given threshold $s_\sigma \in \mathbb{R}^+$. For each load case $j \in \{1, \ldots, l\}$, and each element $i \in \{1, \ldots, m\}$, we get one additional constraint that can be added to (6.24):

$$s_{i,j}(E) \ \leq \ s_\sigma, \quad i = 1, \ldots, m, \ j = 1, \ldots, l. \tag{6.32}$$

The corresponding derivatives are given by

$$\begin{aligned} & \frac{\partial}{\partial e_{ip}} \left(\sum_{k=1}^{n_g} u_j(E)^T B_{i,k}^T E_i I E_i B_{i,k} u_j(E) - s_\sigma \right) \\ = \ & \sum_{k=1}^{n_g} -2u_j(E)^T K^{-1}(E) \frac{\partial K(E)}{\partial e_{ip}} B_{i,k}^T E_i I E_i B_{i,k} u_j(E) \\ & + \sum_{k=1}^{n_g} 2u_j(E)^T B_{i,k}^T \frac{\partial E_i}{\partial e_{ip}} I E_i B_{i,k} u_j(E), \end{aligned} \tag{6.33}$$

see Ertel, Schittkowski and Zillober [21].
Our goal is to solve (6.24) by Algorithm 16. As SCPF is not able to handle semidefinite constraints, they are reformulated by nonlinear constraints. Two reformulation approaches can be used, which are presented in Section 6.2 and Section 6.3

6.2 Reformulation according to Benson and Vanderbei

As SCPF is not able to handle semidefinite constraints they are replaced by nonlinear ones such that Algorithm 16 can be applied. To ease the notation we define

$$E' \;:=\; E - \underline{\nu} I. \tag{6.34}$$

Benson and Vanderbei [9] propose a reformulation which is only applicable for positive definite constraints, i.e., $E' \succ 0,\ E' \in \mathbb{S}^{3m}_{++}$. As FMO requires positive semidefiniteness, the reformulation of problem (6.24) by the approach of Benson and Vanderbei is not exactly equivalent. The resulting $3m$ smooth constraints are nonlinear and convex. This approach is based on the observation that each positive definite matrix $E' \in \mathbb{S}^{3m}_{++}$ can be decomposed by

$$E' \;=\; L(E')\,\Lambda(E')\,L(E')^T\,, \tag{6.35}$$

where the lower triangular matrix $L(E') \in \mathbb{R}^{3m\times 3m}$ as well as the diagonal matrix $\Lambda(E') \in \mathbb{S}^{3m}_{++}$ are unique, if E' is positive definite. The diagonal entries $\lambda_j(E')$, $j = 1, \ldots, 3m$, of $\Lambda(E')$ represent the eigenvalues of the matrix E'. To ease the notation we define

$$L \;:=\; L(E')\,. \tag{6.36}$$

The free material optimization problem (6.24) is reformulated exploiting the following correlation

$$E' \succ 0 \iff \lambda_j(E') > 0,\ j = 1, \ldots, 3m. \tag{6.37}$$

Benson and Vanderbei replace each entry of $\Lambda(E')$ by a nonlinear smooth convex function. As a result, the nonlinear convex semidefinite program (6.24) becomes a nonlinear convex program.

We consider an arbitrary submatrix $E_j,\ j \in \{1, \ldots, 3m\}$ and the corresponding decomposition to compute the j-th diagonal entry of $\Lambda(E')$, denoted by $\lambda_j(E')$.

$$E' \;=\; L\Lambda(E')\,L^T \tag{6.38}$$

$$E' \;=:\; \left[\begin{array}{c|c|c} E'_{j-1} & k^j & \star \\ \hline (k^j)^T & k_{jj} & \star \\ \hline \star & \star & \star \end{array}\right] \tag{6.39}$$

$$L\Lambda(E')\,L^T \;=:\; \left[\begin{array}{c|c|c} L_{j-1} & 0 & 0 \\ \hline (l^j)^T & 1 & 0 \\ \hline \star & \star & \star \end{array}\right] \left[\begin{array}{c|c|c} \Lambda_{j-1}(E') & 0 & 0 \\ \hline 0 & \lambda_j(E') & 0 \\ \hline 0 & 0 & \star \end{array}\right] \left[\begin{array}{c|c|c} L^T_{j-1} & l^j & \star \\ \hline 0 & 1 & \star \\ \hline 0 & 0 & \star \end{array}\right] \tag{6.40}$$

where $E'_{j-1} \in \mathbb{S}^{j-1}_{++}$ is the $(j-1)\times(j-1)$ submatrix of E', k_{jj} is the j-th diagonal entry of E' and k^j is the vector of the first $j-1$ elements of j-th column of E'. Moreover, $L_{j-1} \in \mathbb{R}^{(j-1)\times(j-1)}$ is the $(j-1)\times(j-1)$ submatrix of L, $\Lambda_{j-1}(E') \in \mathbb{S}^{j-1}_{++}$ is the $(j-1)\times(j-1)$ submatrix of $\Lambda(E')$ and l^j is the vector of the first $j-1$ elements of j-th column of L. In addition, $\star$ denotes the remaining entries of corresponding size. By multiplying the matrices in (6.40), we get the following relations:

$$\begin{aligned} E'_{j-1} &= L_{j-1}\Lambda_{j-1}(E')\,L^T_{j-1}, && (6.41)\\ k^j &= L_{j-1}\Lambda_{j-1}(E')\,l^j, && (6.42)\\ k_{jj} &= \lambda_j(E') + \left(l^j\right)^T \Lambda_{j-1}(E')\,l^j. && (6.43) \end{aligned}$$

From (6.42) we obtain

$$l^j = \Lambda^{-1}_{j-1}(E')\,L^{-1}_{j-1}k^j. \qquad (6.44)$$

With (6.43) and (6.44) we get:

$$\begin{aligned} k_{jj} &= \lambda_j(E') + \underbrace{\left(l^j\right)^T}_{=(k^j)^T L^{-T}_{j-1}\Lambda^{-1}_{j-1}(E'),\ (6.44)} \Lambda_{j-1}(E') \underbrace{l^j}_{=\Lambda^{-1}_{j-1}(E')L^{-1}_{j-1}k^j,\ (6.44)}\\ &= \lambda_j(E') + \left(k^j\right)^T L^{-T}_{j-1}\Lambda^{-1}_{j-1}(E')\,\Lambda_{j-1}(E')\,\Lambda^{-1}_{j-1}(E')\,L^{-1}_{j-1}k^j\\ &= \lambda_j(E') + \left(k^j\right)^T \underbrace{L^{-T}_{j-1}\Lambda^{-1}_{j-1}(E')\,L^{-1}_{j-1}}_{=\left(E'_{j-1}\right)^{-1},\ \text{see (6.41)}} k^j\\ &= \lambda_j(E') + \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j && (6.45) \end{aligned}$$

By reformulating (6.45) we can define each eigenvalue $\lambda_j(E')$, $j = 1, \ldots, 3m$, of E' as a function of the entries of E. This leads to $3m$ nonlinear constraints given by

$$\lambda_j(E') = k_{jj} - \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j, \quad j = 1, \ldots, 3m. \qquad (6.46)$$

It can be shown that the functions $\lambda_j(E')$, $j = 1, \ldots, 3m$, are convex. The first and second order derivatives are given analytically by Vanderbei and Benson [88]. We review:

$$\frac{\partial \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j}{\partial k^j_p} = \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_p + \mathbb{1}_p^T \left(E'_{j-1}\right)^{-1} k^j \tag{6.47}$$

$$\frac{\partial^2 \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j}{\partial k^j_p \partial k^j_q} = \mathbb{1}_q^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_p + \mathbb{1}_p^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_q \tag{6.48}$$

$$\frac{\partial \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j}{\partial k_{pq}} = -\left(k^j\right)^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_p \mathbb{1}_q^T \left(E'_{j-1}\right)^{-1} k^j \tag{6.49}$$

$$\begin{aligned}\frac{\partial^2 \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j}{\partial k_{pq} \partial k_{kl}} &= \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_k \mathbb{1}_l^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_p \mathbb{1}_q^T \left(E'_{j-1}\right)^{-1} k^j \\ &\quad + \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_p \mathbb{1}_q^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_k \mathbb{1}_l^T \left(E'_{j-1}\right)^{-1} k^j\end{aligned} \tag{6.50}$$

$$\begin{aligned}\frac{\partial^2 \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j}{\partial k^j_p \partial k_{kl}} &= -\left(k^j\right)^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_k \mathbb{1}_l^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_p \\ &\quad -\mathbb{1}_p^T \left(E'_{j-1}\right)^{-1} \mathbb{1}_k \mathbb{1}_l^T \left(E'_{j-1}\right)^{-1} k^j\end{aligned} \tag{6.51}$$

with $\mathbb{1}_p$ is the p-th unity vector, k_{pq} the entry of E' in row p and column q and k^j_p is the pth entry of the vector k^j. We can summarize the results as follows:

$$\begin{aligned} E' \succ 0 \Longleftrightarrow \lambda_j\left(E'\right) &> 0, \quad j = 1, \ldots, 3m \\ \lambda_j\left(E'\right) &= k_{jj} - \left(k^j\right)^T \left(E'_{j-1}\right)^{-1} k^j. \end{aligned} \tag{6.52}$$

As

$$E' \succ 0 \iff E_i - \underline{\nu} I \succ 0, \quad i = 1, \ldots, m \tag{6.53}$$

holds, we consider the eigenvalues for each matrix $E_i - \underline{\nu} I,\ i = 1, \ldots, m$. With (6.2) and (6.46) we get three inequality constraints for each finite element

$$\lambda_{i1}\left(E_i - \underline{\nu} I\right) := e_{i1} - \underline{\nu} \tag{6.54}$$

$$\lambda_{i2}\left(E_i - \underline{\nu} I\right) := \left(e_{i3} - \underline{\nu}\right) - \frac{e_{i2}^2}{e_{i1} - \underline{\nu}} \tag{6.55}$$

$$\lambda_{i3}\left(E_i - \underline{\nu} I\right) := \left(e_{i6} - \underline{\nu}\right) + \frac{2 e_{i2} e_{i4} e_{i5} - e_{i4}^2 \left(e_{i3} - \underline{\nu}\right) - e_{i5}^2 \left(e_{i1} - \underline{\nu}\right)}{\left(e_{i3} - \underline{\nu}\right)\left(e_{i1} - \underline{\nu}\right) - e_{i2}^2} \tag{6.56}$$

(6.54) can be handled as box constraint, while (6.55) and (6.56) are additional inequality constraints. This leads to the following nonlinear optimization problem arising from problem (6.24)

$$\begin{array}{lll}
\min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\
\text{s.t.} & \sum\limits_{i=1}^{m} \operatorname{Trace}(E_i) - V \leq 0 & \\
 & \operatorname{Trace}(E_i) - \overline{\nu} \leq 0, & i = 1, \dots, m \\
 & f_j^T K(E)^{-1} f_j - \alpha \leq 0, & j = 1, \dots, l \\
 & (e_{i3} - \underline{\nu}) - \frac{e_{i2}^2}{e_{i1} - \underline{\nu}} > 0, & i = 1, \dots, m \\
 & (e_{i6} - \underline{\nu}) + \frac{2e_{i2}e_{i4}e_{i5} - e_{i4}^2(e_{i3} - \underline{\nu}) - e_{i5}^2(e_{i1} - \underline{\nu})}{(e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2} > 0, & i = 1, \dots, m \\
 & e_{i1} > \underline{\nu}, & i = 1, \dots, m \\
 & e_{i3} > \underline{\nu}, & i = 1, \dots, m \\
 & e_{i6} > \underline{\nu}, & i = 1, \dots, m
\end{array} \tag{6.57}$$

Previous calculations show that semidefinite matrices can be replaced by nonlinear inequality constraints that are smooth and at least twice continuously differentiable. It has to be noted that the evaluation of these functions is only possible on the interior of $\mathbb{S}_+$, i.e., $\mathbb{S}_{++}$. Vanderbei et al. [88] propose to use an interior point method in combination with a line search strategy. The stepsize is to be shortened until a descent in the merit function is achieved and the current iterate becomes feasible. As long as $\lambda_j(E') > 0,\ j \in \{1, \dots, 3m\}$, holds, $\left(E'_j\right)^{-1}$ can be evaluated and the step is accepted. If a $j \in \{1, \dots, 3m\}$ is detected such that $\lambda_j(E') \leq 0$, the evaluation of $\lambda_{j+1}(E')$ is not possible, as E'_j is not invertible. Thus, the stepsize is reduced and the functions are evaluated at the new iterate.

6.3 Reformulation Based on Determinants

The disadvantage of Vanderbei and Benson's approach presented in Section 6.2 is that we can only compute the inverse matrix of $E'_j,\ j = 1, \dots, 3m$, if it is positive definite, i.e., within the feasible region. Applying Algorithm 16 the feasibility constraints are passed to the subproblem directly. The evaluation of infeasible iterates within the subproblem solution process would fail, if some of the submatrices are not positive definite. In addition, we have to ensure that $e_{i1} - \underline{\nu} \neq 0$ and $(e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \neq 0$, since the functions (6.55) and (6.56) are not well defined otherwise.

This is the reason why we are looking for a modification of Benson and Vanderbei's approach. The idea is to consider the submatrices of matrix E'. A matrix is positive semidefinite, if each submatrix is positive semidefinite, i.e.,

$$E' \succeq 0 \iff E'_j \succeq 0, \quad \forall j = 1, \dots, 3m \tag{6.58}$$

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

$$E' \succeq 0 \iff d_j\left(E'\right) \geq 0, \quad \forall j = 1, \ldots, 3m \tag{6.59}$$

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

$$d_j\left(E'\right) := \det\left(E'_j\right) = \sum_{q=1}^{j} (-1)^{p+q}\, k_{pq} \det\left(\left(E'_j\right)_{pq}\right), \tag{6.60}$$

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.,

$$E'_{pq} \ := \begin{pmatrix} k_{11} & \ldots & k_{1\ q-1} & k_{1\ q+1} & \ldots & k_{1\ 3m} \\ \vdots & & \vdots & \vdots & & \vdots \\ k_{p-1\ 1} & \ldots & k_{p-1\ q-1} & k_{p-1\ q+1} & \ldots & k_{p-1\ 3m} \\ k_{p+1\ 1} & \ldots & k_{p+1\ q-1} & k_{p+1\ q+1} & \ldots & k_{p+1\ 3m} \\ \vdots & & \vdots & \vdots & & \vdots \\ k_{3m\ 1} & \ldots & k_{3m\ q-1} & k_{3m\ q+1} & \ldots & k_{3m\ 3m} \end{pmatrix} \tag{6.61}$$

It can be shown that the resulting functions $d_j\left(E'\right),\ j = 1, \ldots, 3m$ are nonconvex and polynomial. As (6.59) holds, the feasible region defined by $d_j\left(E'\right) \geq 0,\ j = 1, \ldots, 3m$, is convex. The design variable E' is block diagonal, i.e., of the form

$$E' \ := \begin{pmatrix} \blacksquare & 0 & 0 & \cdots & 0 \\ 0 & \blacksquare & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \blacksquare & 0 \\ 0 & \cdots & \cdots & 0 & \blacksquare \end{pmatrix} \tag{6.62}$$

Therefore it is sufficient to show that each 3×3 block $\blacksquare$ of E' is positive semidefinite. The blocks are given by $E_i - \underline{\nu} I,\ i = 1, \ldots, m$. We get the following three inequality constraints from (6.60) and (6.2)

$$d_{i1}\left(E_i - \underline{\nu} I\right) \ := \ e_{i1} - \underline{\nu} \geq 0 \tag{6.63}$$

$$d_{i2}\left(E_i - \underline{\nu} I\right) \ := \ \left(e_{i3} - \underline{\nu}\right)\left(e_{i1} - \underline{\nu}\right) - e_{i2}^2 \geq 0 \tag{6.64}$$

$$\begin{aligned} d_{i3}\left(E_i - \underline{\nu} I\right) \ := \ & \left(e_{i6} - \underline{\nu}\right)\left(\left(e_{i3} - \underline{\nu}\right)\left(e_{i1} - \underline{\nu}\right) - e_{i2}^2\right) \\ & - e_{i4}^2\left(e_{i3} - \underline{\nu}\right) - e_{i5}^2\left(e_{i1} - \underline{\nu}\right) + 2 e_{i2} e_{i4} e_{i5} \geq 0 \end{aligned} \tag{6.65}$$

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

The following problem formulation (6.66) is equivalent to (6.24) and can be solved efficiently by Algorithm 16.

$$\begin{array}{lll}
\min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\
\text{s.t.} & \sum\limits_{i=1}^{m} \operatorname{Trace}(E_i) - V \leq 0 & \\
& \operatorname{Trace}(E_i) - \overline{\nu} \leq 0, & i = 1, \ldots, m \\
& f_j^T K(E)^{-1} f_j - \alpha \leq 0, & j = 1, \ldots, l \\
& (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \geq 0, & i = 1, \ldots, m \\
& (e_{i6} - \underline{\nu})\left((e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2\right) & \\
& -e_{i4}^2 (e_{i3} - \underline{\nu}) - e_{i5}^2 (e_{i1} - \underline{\nu}) + 2e_{i2}e_{i4}e_{i5} \geq 0, & i = 1, \ldots, m \\
& e_{i1} \geq \underline{\nu}, & i = 1, \ldots, m \\
& e_{i3} \geq \underline{\nu}, & i = 1, \ldots, m \\
& e_{i6} \geq \underline{\nu}, & i = 1, \ldots, m
\end{array} \tag{6.66}$$

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

6.4 Evaluations of Functions and Derivatives

Applying Algorithm 16 to solve (6.66), 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 $d_{i3}(E_i - \underline{\nu} I),\ i = 1, \ldots, m$, are given explicitly. We consider an arbitrary finite element $i \in \{1, \ldots, m\}$ and the corresponding matrix E_i. The determinant of the (2×2) submatrix is dependent on three variables. We get

$$d_{i2}(E_i - \underline{\nu} I) = (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \tag{6.67}$$

$$\frac{\partial d_{i2}(E_i - \underline{\nu} I)}{\partial e_{ip}} = \begin{pmatrix} e_{i3} - \underline{\nu} \\ -2e_{i2} \\ e_{i1} - \underline{\nu} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{6.68}$$

$$\frac{\partial d_{i2}(E_i - \underline{\nu} I)}{\partial e_{ip} \partial e_{iq}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.69}$$

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

$$\begin{aligned}
d_{i3}\left(E_i - \underline{\nu} I\right) &:= \left(e_{i6} - \underline{\nu}\right)\left(\left(e_{i3} - \underline{\nu}\right)\left(e_{i1} - \underline{\nu}\right) - e_{i2}^2\right) \\
&\quad - e_{i4}^2\left(e_{i3} - \underline{\nu}\right) - e_{i5}^2\left(e_{i1} - \underline{\nu}\right) + 2e_{i2}e_{i4}e_{i5} \\
&= \left(e_{i6} - \underline{\nu}\right) d_{i2}\left(E_i - \underline{\nu} I\right) - e_{i4}^2\left(e_{i3} - \underline{\nu}\right) - e_{i5}^2\left(e_{i1} - \underline{\nu}\right) \\
&\quad + 2e_{i2}e_{i4}e_{i5}
\end{aligned} \tag{6.70}$$

$$\frac{\partial d_{i3}\left(E_i - \underline{\nu} I\right)}{\partial e_{ip}} = \begin{pmatrix} \left(e_{i3} - \underline{\nu}\right)\left(e_{i6} - \underline{\nu}\right) - e_{i5}^2 \\ 2e_{i4}e_{i5} - 2e_{i2}\left(e_{i6} - \underline{\nu}\right) \\ \left(e_{i1} - \underline{\nu}\right)\left(e_{i6} - \underline{\nu}\right) - e_{i4}^2 \\ 2e_{i2}e_{i5} - 2\left(e_{i3} - \underline{\nu}\right) e_{i4} \\ 2e_{i2}e_{i4} - 2e_{i5}\left(e_{i1} - \underline{\nu}\right) \\ d_{i2}\left(E_i - \underline{\nu} I\right) \end{pmatrix} \tag{6.71}$$

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

$$\begin{pmatrix} 0 & 0 & e_{i6} - \underline{\nu} & 0 & -2e_{i5} & e_{i3} - \underline{\nu} \\ 0 & -2\left(e_{i6} - \underline{\nu}\right) & 0 & 2e_{i5} & 2e_{i4} & -2e_{i2} \\ e_{i6} - \underline{\nu} & 0 & 0 & -2e_{i4} & 0 & e_{i1} - \underline{\nu} \\ 0 & 2e_{i5} & -2e_{i4} & -2\left(e_{i3} - \underline{\nu}\right) & 2e_{i2} & 0 \\ -2e_{i5} & 2e_{i4} & 0 & 2e_{i2} & -2\left(e_{i1} - \underline{\nu}\right) & 0 \\ e_{i3} - \underline{\nu} & -2e_{i2} & e_{i1} - \underline{\nu} & 0 & 0 & 0 \end{pmatrix} \tag{6.72}$$

7. NUMERICAL IMPLEMENTATION AND RESULTS

7.1 Implementation

The implementation of Algorithm 16 is the FORTRAN code SCPF10.f. In this section, we present details of the implementation of Algorithm 16. We introduce an active set strategy, which reduces the size of the subproblem and thus enhance the computation time, see Section 7.1.1. Moreover, the linear constraints can be passed to the subproblem directly, see Section 7.1.2. In addition, subproblem (5.7) might be infeasible. In Section 7.1.3, the feasible region of the subproblem is extended, such that feasible solutions always exist. Moreover, a procedure for stepsize reduction is presented, which is more efficient than the Armijo line search algorithm, see Section 7.1.4. Some stopping criteria are presented in Section 7.1.5.

7.1.1 Active Set Strategy

An important feature improving the performance of the SCP Algorithm 14 and the SCPF Algorithm 16 significantly is the usage of an active set strategy, Zillober [104]. The active set strategy reduces the dimension of the subproblem (5.7). Only those constraints are integrated, that are active or violated at the current iterate or which are supposed to become active or violated in the subsequent iteration. The other constraints are neglected during the solution of the subproblem. As a consequence, gradients have to be computed only for the constraints belonging to the active set. This reduces the computational effort significantly, especially if the evaluation of gradients is expensive. The selection of the constraints, which should be part of the active set is difficult. If too many constraints are included in the active set, the computational benefit is low. If only few constraints are considered, the solution process might cycle. Equality constraints are included in the active set in each iteration. An inequality constraint $c_j\left(x\right),\ j \in \{m_e+1,\ldots,m_c\}$, is included in iteration k, if

$$c_j\left(x^{(k)}\right) \geq -a, \qquad \forall j = m_e+1,\ldots,m_c \tag{7.1}$$

holds, where $a \in \mathbb{R}^+$ is to be specified by the user. In addition, an inequality constraint $c_j\left(x\right),\ j = m_e{+}1,\ldots,m_c$, which was included in the active set during the last iteration remains in the active set, if

$$\left(y_c^{(k)}\right)_j \neq 0 \tag{7.2}$$

holds.

It can be beneficial that some specific constraints are part of the active set in each iteration step, although they do not satisfy (7.1) or (7.2). The reason is that these constraints are expected to be active in the optimal solution $x^\star \in \mathbb{R}^n$, i.e.,

$$c_j\left(x^\star\right) \geq -a, \qquad \forall j = m_e + 1, \dots, m_c. \tag{7.3}$$

These constraints are added to the active set permanently, if they are identified by the user. Therefore, the active set is extended by this additional condition. We define the active set by

$$\begin{aligned} \mathbb{A}^{(k)} \quad := \quad & \left\{ j = m_e + 1, \dots, m_c \,\middle|\, c_j\left(x^{(k)}\right) \geq -a \right\} \cup \\ & \left\{ j = m_e + 1, \dots, m_c \,\middle|\, c_j\left(x^{(k-1)}\right) \geq -a \text{ and } \left(y_c^{(k)}\right)_j \neq 0 \right\} \cup \\ & \left\{ j = m_e + 1, \dots, m_c \,\middle|\, c_j\left(x^\star\right) \geq -a \right\}. \end{aligned} \tag{7.4}$$

It has to be noticed that the feasibility constraints $e_j\left(x\right),\ j = 1, \dots, m_f$, have to be included in the active set in each iteration step. Otherwise, their feasibility cannot be assured. This leads to the following reduced subproblem in iteration k derived from (5.7)

$$\begin{array}{lll} \min\limits_{x} & f^{(k)}\left(x\right) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j^{(k)}\left(x\right) = 0, & j = 1, \dots, m_e \\ & c_j^{(k)}\left(x\right) \leq 0, & j \in \mathbb{A}^{(k)} \\ & e_j\left(x\right) \leq 0, & j = 1, \dots, m_f \\ & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i = 1, \dots, n \end{array} \tag{7.5}$$

Using an active set strategy on subproblem level allows to neglect feasibility constraints during the solution process of the subproblem.

7.1.2 Linear Constraints

Both the MMA and SCP method presented in Section 4.1 and 4.2 respectively, approximate linear inequality constraints in the same way as nonlinear ones, see (4.5). SCPF10.f provides the opportunity to pass linear constraints directly to the subproblem, which can be exploited on subproblem level. We define the set of linear constraints by

$$\mathbb{L} \quad := \quad \left\{ j = m_e + 1, \dots, m_c \,\middle|\, c_j\left(x\right) \text{ linear} \right\} \tag{7.6}$$

and the set of nonlinear constraints by

$$\overline{\mathbb{L}} \quad := \quad \left\{ j = m_e + 1, \dots, m_c \,\middle|\, j \notin \mathbb{L} \right\}. \tag{7.7}$$

Taking the active set strategy presented in Section 7.1.1 into account, the subproblem in iteration k is given by

$$\begin{array}{lll}\min\limits_{x} & f^{(k)}(x) & x \in \mathbb{R}^n \\ \text{s.t.} & c_j^{(k)}(x) = 0, & j = 1, \dots, m_e \\ & c_j(x) \le 0, & j \in \mathbb{L} \cap \mathbb{A}^{(k)} \\ & c_j^{(k)}(x) \le 0, & j \in \overline{\mathbb{L}} \cap \mathbb{A}^{(k)} \\ & e_j(x) \le 0, & j = 1, \dots, m_f \\ & \underline{x}_i^{(k)} \le x_i \le \overline{x}_i^{(k)}, & i = 1, \dots, n \end{array} \tag{7.8}$$

7.1.3 Infeasible Subproblems

The formulation of subproblem (5.7) does not ensure that the feasible region is nonempty, if the current iterate $x^{(k)}$ is infeasible. For this reason we have to consider an extended problem formulation, whenever an infeasible subproblem is detected. Therefore, the feasible region is enlarged by introducing additional variables for each violated constraint. Due to strict feasibility of $e_j(x)$, $j = 1, \dots, m_f$, this is only necessary for the constraints $c_j(x)$, $j = 1, \dots, m_c$. We consider the violated constraints and define

$$\mathbb{M}_e^{(k)} := \left\{ j = 1, \dots, m_e \,\middle|\, c_j^{(k)}\left(x^{(k)}\right) \neq 0 \right\}, \tag{7.9}$$

$$\overline{\mathbb{M}}_e^{(k)} := \left\{ j = 1, \dots, m_e \,\middle|\, c_j^{(k)}\left(x^{(k)}\right) = 0 \right\}, \tag{7.10}$$

$$\mathbb{M}_c^{(k)} := \left\{ j \in \overline{\mathbb{L}} \cap \mathbb{A}^{(k)} \,\middle|\, c_j^{(k)}\left(x^{(k)}\right) > 0 \right\} \cup \left\{ j \in \mathbb{L} \cap \mathbb{A}^{(k)} \mid c_j\left(x^{(k)}\right) > 0 \right\}, \tag{7.11}$$

$$\overline{\mathbb{M}}_c^{(k)} := \left\{ j \in \mathbb{A}^{(k)} \mid j \notin \mathbb{M}_c^{(k)} \right\}, \tag{7.12}$$

$$\mathbb{M}^{(k)} := \mathbb{M}_e^{(k)} \cup \mathbb{M}_c^{(k)} \tag{7.13}$$

The number of violated constraints in iterate $x^{(k)}$ is given by $|\mathbb{M}^{(k)}|$. We extend each constraint $c_j^{(k)}(x)$, $j \in \mathbb{M}^{(k)}$, by

$$c_j^{(k)}(x) - \gamma_j c_j\left(x^{(k)}\right) = 0, \quad j \in \mathbb{M}_e^{(k)} \tag{7.14}$$

$$c_j(x) - \gamma_j c_j\left(x^{(k)}\right) \le 0, \quad j \in \mathbb{M}_c^{(k)} \cap \mathbb{L} \tag{7.15}$$

$$c_j^{(k)}(x) - \gamma_j c_j\left(x^{(k)}\right) \le 0, \quad j \in \mathbb{M}_c^{(k)} \cap \overline{\mathbb{L}} \tag{7.16}$$

with $-1 \le \gamma_j \le 1$, $j \in \mathbb{M}_e^{(k)}$ and $0 \le \gamma_j \le 1$, $j \in \mathbb{M}_c^{(k)}$. It is easy to see that for $\gamma_j = 1, j \in \mathbb{M}^{(k)}$, the current iterate $x^{(k)}$ becomes feasible for the extended subproblem. Forcing the value of the additional variables γ_j, $j \in \mathbb{M}^{(k)}$, to be as small as possible, they are penalized in the objective function by the penalty parameter $\rho_c^{(k)} \in \mathbb{R}^{m_c}$, see (5.14) and the augmented Lagrangian function (5.13). We obtain the extended subproblem

$$
\begin{array}{lll}
\min\limits_{x,\gamma} & f^{(k)}(x) + \dfrac{1}{2} \displaystyle\sum_{j \in \mathbb{M}^{(k)}} \left(\rho_c^{(k)}\right)_j \gamma_j^2 & x \in \mathbb{R}^n, \gamma \in \mathbb{R}^{|\mathbb{M}^{(k)}|} \\
\text{s.t.} & c_j^{(k)}(x) - \gamma_j c_j\left(x^{(k)}\right) = 0, & j \in \mathbb{M}_e^{(k)} \\
 & c_j^{(k)}(x) = 0, & j \in \overline{\mathbb{M}}_e^{(k)} \\
 & c_j(x) - \gamma_j c_j\left(x^{(k)}\right) \leq 0, & j \in \mathbb{M}_c^{(k)} \cap \mathbb{L} \\
 & c_j^{(k)}(x) - \gamma_j c_j\left(x^{(k)}\right) \leq 0, & j \in \mathbb{M}_c^{(k)} \cap \overline{\mathbb{L}} \\
 & c_j(x) \leq 0, & j \in \overline{\mathbb{M}}_c^{(k)} \cap \mathbb{L} \\
 & c_j^{(k)}(x) \leq 0, & j \in \overline{\mathbb{M}}_c^{(k)} \cap \overline{\mathbb{L}} \\
 & e_j(x) \leq 0, & j = 1, \ldots, m_f \\
 & -1 \leq \gamma_j \leq 1, & j \in \mathbb{M}_e^{(k)} \\
 & 0 \leq \gamma_j \leq 1, & j \in \mathbb{M}_c^{(k)} \\
 & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i = 1, \ldots, n
\end{array}
\tag{7.17}
$$

Note that linear constraints might be violated in iteration k , if they were not included in the active set of the previous iteration. Zillober [97] shows that under certain conditions a descent in the augmented Lagrangian is still guaranteed for a similar problem.

7.1.4 Line Search Procedure

The implementation of the line search procedure has a major impact on the performance of the optimization algorithm. Frequently, the procedure is based on a quadratic interpolation of the merit function, which is in some cases combined with the stepsize reduction according to Armijo, see Schittkowski [74]. We use a quadratic interpolation based on $\nabla\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}$, $\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}$ and $\Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)}\right)$ and define

$$
\overline{\sigma} := \frac{0.5\left(\sigma^{(k,i)}\right)^2 \nabla\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}}{\sigma^{(k,i)} \nabla\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)} - \Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)}\right) + \Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}} \tag{7.18}
$$

To avoid numerical instabilities in special cases of an unsuitable interpolation, the procedure is combined with the Armijo step stepsize reduction rule (4.27), see Schittkowski [74]. We define

$$
\sigma^{(k,i+1)} := \begin{cases} \max\left\{\overline{\sigma}; \beta_2 \sigma^{(k,i)}\right\}, & \text{if } \overline{\sigma} < \sigma^{(k,i)} \\ \beta \sigma^{(k,i)}, & \text{otherwise} \end{cases} \tag{7.19}
$$

with $\beta \in (0,1)$ and $0 < \beta_2 < \beta$.
The stepsize $\sigma^{(k,i)}$ is reduced until the Armijo condition is satisfied for the first time, i.e.,

$$\Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)}\right) \leq \Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + r\sigma^{(k,i)} \nabla\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}, \tag{7.20}$$

holds with $r \in (0,1)$. This leads to the following algorithm

Algorithm 17. Line Search Procedure

Step 0: Set $\sigma^{(k,0)} := 1$ and choose parameters $r \in (0,1)$, $\beta \in (0,1)$, $0 < \beta_2 < \beta$.
Compute $\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}$, $\Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,0)} d^{(k)}\right)$, $\nabla\Phi_{\rho^{(k)}}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}^T d^{(k)}$.
Set $i := 0$.

Step 1: If (7.20) is satisfied, then STOP, $\sigma^{(k)} = \sigma^{(k,i)}$.

Step 2: Compute $\overline{\sigma}$ according to (7.18).

Step 3: Update $\sigma^{(k,i+1)}$ according to (7.19).

Step 4: Set $i := i+1$, compute $\Phi_{\rho^{(k)}}\left(\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \sigma^{(k,i)} d^{(k)}\right)$ and goto Step 1.

7.1.5 Stopping Criteria

Algorithm 16 seeks to find a stationary point of (5.5). Different stopping criteria can be used, see Schittkowski [74]. The algorithm terminates, if an iterate $\left(x^{(k)}, y^{(k)}\right)$ is considered as feasible, i.e.,

$$\left|c_j\left(x^{(k)}\right)\right| \leq \varepsilon, \qquad j = 1, \ldots, m_e \tag{7.21}$$

$$c_j\left(x^{(k)}\right) \leq \varepsilon, \qquad j = m_e + 1, \ldots, m_c \tag{7.22}$$

$$e_j\left(x^{(k)}\right) \leq \varepsilon, \qquad j = 1, \ldots, m_f \tag{7.23}$$

where $\varepsilon \in \mathbb{R}^+$ is the feasibility tolerance and one of the following conditions holds,

1. $\left|\nabla f\left(x^{(k)}\right)^T \Delta x^{(k)}\right| + \sum_{i=1}^{m_c}\left|\left(y_c^{(k)}\right)_j c_j\left(x^{(k)}\right)\right| + \sum_{i=1}^{m_f}\left|\left(y_e^{(k)}\right)_j e_j\left(x^{(k)}\right)\right| \leq \varepsilon$ (7.24)
2. $\left\|\nabla f\left(x^{(k)}\right)\right\| \leq \varepsilon$ (7.25)
3. $\left\|\dfrac{x^{(k)} - x^{(k-1)}}{x^{(k-1)}}\right\|_\infty \leq \varepsilon,\ |f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)| \leq \varepsilon$
   and $\left|\dfrac{f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)}{f\left(x^{(k-1)}\right)}\right| \leq \varepsilon$ (7.26)
4. $\left\|\nabla\Phi_\rho\left(x^{(k)}, y^{(k)}\right)\right\|_\infty \leq \varepsilon$ (7.27)
5. $|f\left(x^{(k-i)}\right) - f\left(x^{(k-i-1)}\right)| \leq \varepsilon,\ i = 0, \ldots, 15$ (7.28)

7.2 Program Organization

The implementation of Algorithm 16 is the FORTRAN code SCPF10.f. The program is called via reverse communication. The user has to provide function and gradient evaluations on request. Moreover, the Hessian evaluations must be provided for feasibility constraints. In general, it is distinguished between the inner and the outer iteration sequence. The outer iteration sequence consists of the iterates computed by Algorithm 16. Function and gradient evaluations are required for the objective function $f(x)$, the constraints $c_j(x)\,,\ j = 1,\dots,m_c$, and feasibility constraints $e_j(x)\,,\ j = 1,\dots,m_f$. Note that the gradients are to be computed for the constraints included in the active set only, see Section 7.1.1. In addition, the resulting subproblem (5.7) is solved iteratively. During this inner iterations, functions, gradients and second order derivatives are to be evaluated only for the feasibility constraints $e_j(x)\,,\ j = 1,\dots,m_f$. The corresponding evaluations of the objective function $f^{(k)}(x)$ and the constraints $c_j^{(k)}(x)\,,\ j = 1,\dots,m_c$, are determined internally in analytical form by the approximation schemes (4.23) and (4.25). The subproblems are solved by IPOPT, see Wächter and Biegler [90]. As it can only handle equality constraints, we extend the subproblem (5.7) presented in Section 5.1 by introducing slack variables $s \in \mathbb{R}^{m_c+m_f-m_e}$. We get

$$\begin{array}{lll} \min\limits_{x,s} & f^{(k)}(x) & x \in \mathbb{R}^n, s \in \mathbb{R}^{m_c+m_f-m_e} \\ \text{s.t.} & c_j^{(k)}(x) = 0, & j = 1,\dots,m_e \\ & c_j^{(k)}(x) + s_{j-m_e} = 0, & j = m_e+1,\dots,m_c \\ & e_j'(x) = 0, & j = 1,\dots,m_f \\ & \underline{x}_i^{(k)} \le x_i \le \overline{x}_i^{(k)}, & i = 1,\dots,n \\ & 0 \le s_i, & i = 1,\dots,m_c - m_e + m_f \end{array} \tag{7.29}$$

where

$$e_j'(x) \ := \ e_j(x) + s_{j+m_c-m_e}, \quad j = 1,\dots,m_f. \tag{7.30}$$

A graphical presentation of the program organization is given in Figure 7.1. The algorithm creates an output of the form

```
IT   ITSUB   ACT   FEASIBILITY   OBJECTIVE   SIGMA   NORM(DX)   NORM(L_X)
```

with

IT	-	iteration number
ITSUB	-	number of iterations to solve the subproblem
ACT	-	number of constraints within the active set
FEASIBILITY	-	sum of constraint violations, i.e., $\|c(x)\|_1$
OBJECTIVE	-	objective function value
SIGMA	-	stepsize
NORM(DX)	-	norm of primal search direction, i.e., $\|\Delta x\|_2$
NORM(L_X)	-	norm of gradient of the Lagrangian function, i.e., $\|\nabla_x L(x,y)\|_\infty$

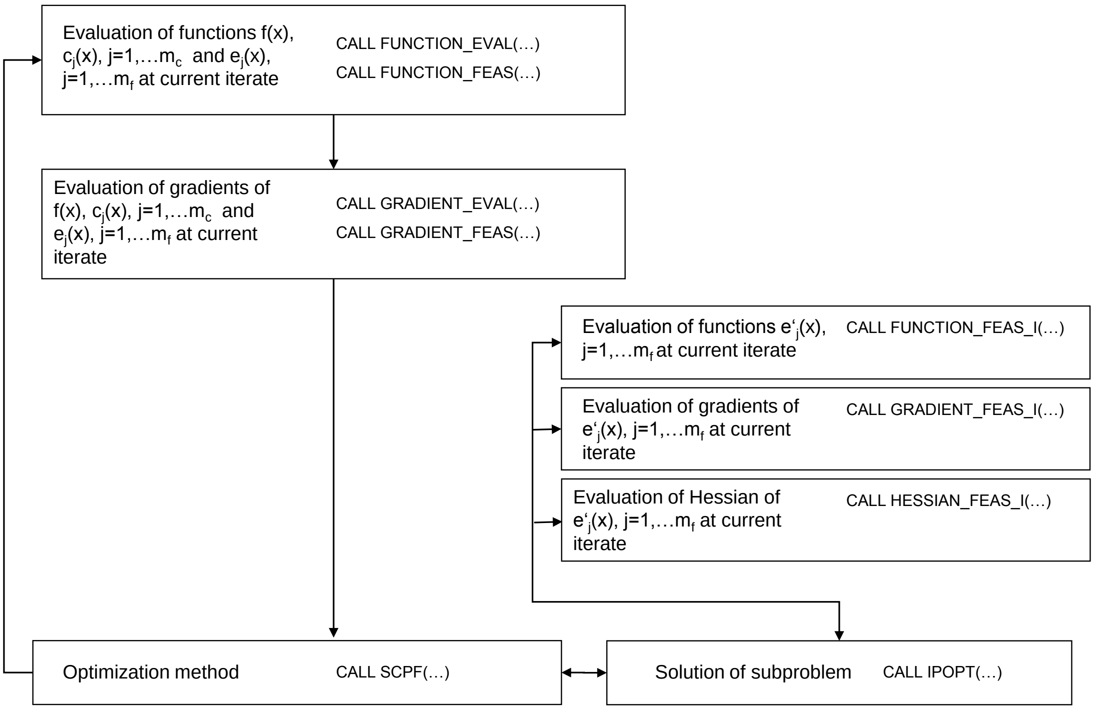

Evaluation of functions f(x), c_j(x), j=1,…m_c and e_j(x), j=1,…m_f at current iterate
CALL FUNCTION_EVAL(…)
CALL FUNCTION_FEAS(…)
Evaluation of gradients of f(x), c_j(x), j=1,…m_c and e_j(x), j=1,…m_f at current iterate
CALL GRADIENT_EVAL(…)
CALL GRADIENT_FEAS(…)
Evaluation of functions e'_j(x), j=1,…m_f at current iterate
CALL FUNCTION_FEAS_I(…)
Evaluation of gradients of e'_j(x), j=1,…m_f at current iterate
CALL GRADIENT_FEAS_I(…)
Evaluation of Hessian of e'_j(x), j=1,…m_f at current iterate
CALL HESSIAN_FEAS_I(…)
Optimization method
CALL SCPF(…)
Solution of subproblem
CALL IPOPT(…)

Fig. 7.1: Reverse communication for SCPF-Algorithm

7.3 Numerical Results

7.3.1 Free Material Optimization

Free material optimization was introduced in Chapter 6. We proceed from problem formulation (6.66) in Section 6.3, i.e.,

$$\begin{array}{lll} \min\limits_{E,\alpha} & \alpha & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R} \\ \text{s.t.} & \sum\limits_{i=1}^{m} \operatorname{Trace}(E_i) - V \le 0 & \\ & \operatorname{Trace}(E_i) - \overline{\nu} \le 0, & i = 1,\dots,m \\ & f_j^T K(E)^{-1} f_j - \alpha \le 0, & j = 1,\dots,l \\ & -d_{i2}(E_i - \underline{\nu} I) \le -\varepsilon, & i = 1,\dots,m \\ & -d_{i3}(E_i - \underline{\nu} I) \le -\varepsilon, & i = 1,\dots,m \\ & e_{ip} \le \overline{e}, & i = 1,\dots,m, \quad p = 1,\dots,6 \\ & e_{ip} \ge \underline{e}, & i = 1,\dots,m, \quad p = 2,4,5 \\ & e_{ip} \ge \underline{e}_d, & i = 1,\dots,m, \quad p = 1,3,4 \\ & \underline{\alpha} \le \alpha \le \overline{\alpha} & \end{array} \tag{7.31}$$

with

$$d_{i2}(E_i - \underline{\nu} I) := (e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2 \tag{7.32}$$

$$\begin{aligned} d_{i3}(E_i - \underline{\nu} I) := {} & (e_{i6} - \underline{\nu})\big((e_{i3} - \underline{\nu})(e_{i1} - \underline{\nu}) - e_{i2}^2\big) \\ & - e_{i4}^2 (e_{i3} - \underline{\nu}) - e_{i5}^2 (e_{i1} - \underline{\nu}) + 2 e_{i2} e_{i4} e_{i5} \end{aligned} \tag{7.33}$$

The small parameter $\varepsilon \in \mathbb{R}^+$ is introduced to prevent numerical instabilities in case of vanishing material. Otherwise, the LICQ, see Definition 5, is violated, as the gradients become zero and thus linear dependent. Moreover, we introduce box constraints for each variable. This is necessary to ensure a compact feasible set F, defined by

$$\begin{aligned} F := {} & \{e_{ip} \in \mathbb{R},\ i = 1,\dots,m,\ p = 1,\dots,6 \mid -d_{i2}(E_i - \underline{\nu} I) \le -\varepsilon\} \\ & \cap \{e_{ip} \in \mathbb{R},\ i = 1,\dots,m,\ p = 1,\dots,6 \mid -d_{i3}(E_i - \underline{\nu} I) \le -\varepsilon\} \\ & \cap X \end{aligned} \tag{7.34}$$

with

$$\begin{aligned} X := {} & \{e_{ip} \in \mathbb{R},\ i = 1,\dots,m,\quad p = 2,4,5 \mid \underline{e} \le e_{ip} \le \overline{e}\} \\ & \cap \{e_{ip} \in \mathbb{R},\ i = 1,\dots,m,\quad p = 1,3,4 \mid \underline{e}_d \le e_{ip} \le \overline{e}\} \\ & \cap \{\alpha \in \mathbb{R} \mid \underline{\alpha} \le \alpha \le \overline{\alpha}\} \end{aligned} \tag{7.35}$$

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

The subproblem is generated by approximating the objective function as well as nonlinear inequalities. To ensure strict feasibility with respect to the subdeterminants, the corresponding nonlinear functions are passed as feasibility constraints directly to the subproblem. Within our test set the linear constraints are also passed to the subproblem without approximation, see Section 7.1.2. This leads to the following subproblem

$$\begin{array}{lll}
\min\limits_{E,\alpha} & f^{(k)}(E,\alpha) & E \in \mathbb{S}^{3m},\ \alpha \in \mathbb{R},\ s \in \mathbb{R}^{1+3m+l} \\
\text{s.t.} & \sum\limits_{i=1}^{m} \operatorname{Trace}(E_i) - V + s_1 = 0 & \\
 & \operatorname{Trace}(E_i) - \overline{\nu} + s_{1+i} = 0, & i = 1,\dots,m \\
 & c_j^{(k)}(E,\alpha) + s_{1+m+j} = 0, & j = 1,\dots,l \\
 & -d_{i2}(E_i - \underline{\nu} I) + s_{1+m+l+i} = -\varepsilon, & i = 1,\dots,m \\
 & -d_{i3}(E_i - \underline{\nu} I) + s_{1+2m+l+i} = -\varepsilon, & i = 1,\dots,m \\
 & \underline{e}_{ip}^{(k)} \le e_{ip} \le \overline{e}_{ip}^{(k)}, & i = 1,\dots,m,\ p = 1,\dots,6 \\
 & \underline{\alpha}^{(k)} \le \alpha \le \overline{\alpha}^{(k)} & \\
 & 0 \le s_i, & i = 1,\dots,3m+l+1
\end{array} \tag{7.36}$$

where

$$\begin{aligned}
f^{(k)}(E,\alpha) \quad = \quad & \alpha^{(k)} + \frac{\left(U_\alpha^{(k)} - \alpha^{(k)}\right)^2}{U_\alpha^{(k)} - \alpha} - \left(U_\alpha^{(k)} - \alpha^{(k)}\right) + \tau \frac{\left(\alpha - \alpha^{(k)}\right)^2}{U_\alpha^{(k)} - \alpha} \\
& + \sum_{i=1}^{m} \sum_{p=1}^{6} \tau \frac{\left(e_{ip} - e_{ip}^{(k)}\right)^2}{U_{ip}^{(k)} - e_{ip}}
\end{aligned} \tag{7.37}$$

is the strictly convex separable approximation of the objective function $f(E,\alpha) = \alpha$. The optimization variable e_{ip} corresponds to element i, while p defines the entry of the corresponding elasticity matrix E_i. Note that $U_{ip}^{(k)}$ is the corresponding upper asymptote for e_{ip}, $i = 1,\dots,n$, $p = 1,\dots,6$ and $U_\alpha^{(k)}$ the corresponding upper asymptote for variable α. The lower asymptotes $L_{ip}^{(k)}$ are defined analogously. Moreover, $\tau > 0$ holds. The convex approximation of the compliance function (6.16) for a load case $j \in \{1,\dots,l\}$ is given by

$$\begin{aligned}
c_j^{(k)}\left(E,\alpha\right) &= c_j\left(E^{(k)},\alpha^{(k)}\right) \\
&+ \sum_{I_+^{(j,k)}} \frac{\partial c_j\left(E^{(k)},\alpha^{(k)}\right)}{\partial e_{ip}} \left(\frac{\left(U_{ip}^{(k)} - e_{ip}^{(k)}\right)^2}{U_{ip}^{(k)} - e_{ip}} - \left(U_{ip}^{(k)} - e_{ip}^{(k)}\right) \right) \\
&- \sum_{I_-^{(j,k)}} \frac{\partial c_j\left(E^{(k)},\alpha^{(k)}\right)}{\partial e_{ip}} \left(\frac{\left(e_{ip}^{(k)} - L_{ip}^{(k)}\right)^2}{e_{ip} - L_{ip}^{(k)}} - \left(e_{ip}^{(k)} - L_{ip}^{(k)}\right) \right) \\
&+ \frac{\left(\alpha^{(k)} - L_\alpha^{(k)}\right)^2}{\alpha - L_\alpha^{(k)}} - \left(\alpha^{(k)} - L_\alpha^{(k)}\right)
\end{aligned} \tag{7.38}$$

with

$$I_+^{(j,k)} := \left\{ i = 1,\dots,m,\ p = 1,\dots,6 \;\middle|\; \frac{\partial c_j\left(E^{(k)},\alpha^{(k)}\right)}{\partial e_{ip}} \geq 0 \right\} \tag{7.39}$$

$$I_-^{(j,k)} := \left\{ i = 1,\dots,m,\ p = 1,\dots,6 \;\middle|\; \frac{\partial c_j\left(E^{(k)},\alpha^{(k)}\right)}{\partial e_{ip}} < 0 \right\} \tag{7.40}$$

and

$$c_j\left(E,\alpha\right) := f_j^T K\left(E\right)^{-1} f_j - \alpha, \qquad j = 1,\dots,l \tag{7.41}$$

$$\frac{\partial c_j\left(E,\alpha\right)}{\partial e_{ip}} := -u_j(E)^T \frac{\partial K\left(E\right)}{\partial e_{ip}} u_j(E), \qquad i = 1,\dots,m,\quad p = 1,\dots,6 \tag{7.42}$$

The computation of the Hessian is important to ensure local superlinear convergence on subproblem level. Therefore, the Hessian of the Lagrangian function of the subproblem has to be determined. As the linear constraints are not approximated, the corresponding Hessian becomes zero. The second order derivatives of the approximated functions $c_j^{(k)}\left(E,\alpha\right)$, $j = m_e + 1,\dots,m_c$, and $f^{(k)}\left(E,\alpha\right)$ are given by (4.25). They are separable, i.e., the Hessian is diagonal. The Hessian of the compliance function is block separable, see (6.69) and (6.72). All together this leads to a $(6m \times 6m)$ blockdiagonal matrix of the form

$$\frac{\partial^2 L^{(k)}\left(x,y\right)}{\partial x_p \partial x_q} = \begin{pmatrix} \blacksquare & 0 & 0 & \cdots & 0 \\ 0 & \blacksquare & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \blacksquare & 0 \\ 0 & \cdots & \cdots & 0 & \blacksquare \end{pmatrix} \tag{7.43}$$

where $\blacksquare$ is a (6×6) matrix. The implementation of function, gradient and Hessian evaluation of the feasibility constraints was part of a diploma thesis, see Werner [92].

Within the solution process, the variables are stored as a vector $x \in \mathbb{R}^{6m+1}$, i.e., six variables for each $E_i,\ i = 1, \ldots, m$, and the additional variable α. We get

$$x \ := \ (e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{21}, e_{22}, \ldots, e_{m6}, \alpha)^T . \tag{7.44}$$

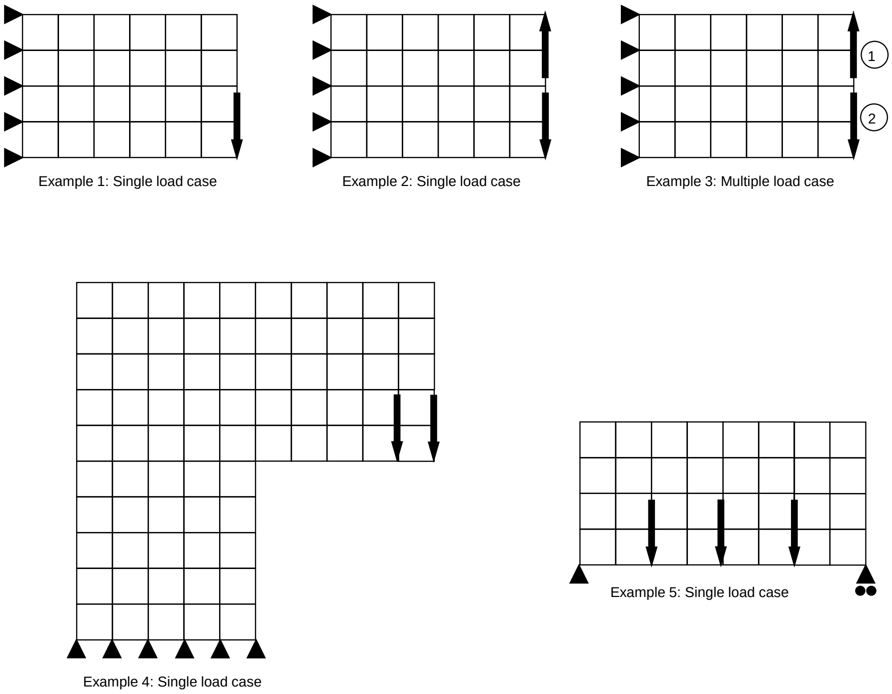
The internal parameters of the Algorithm are given in Table 9.3, see Appendix. The specific constants of problem (7.31) are summarized in the following table.

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

Tab. 7.1: 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 (7.31) is to be solved by Algorithm 16. The algorithm was integrated into the PLATO-N interface, see Boyd [14]. We consider given test cases of the PLATO-N academic test case library, see Bogomolny [13]. An overview of the test set is given in Figure 7.2. Fixed nodes are denoted by ▶ or ▲ , while ♣ allows to move in horizontal direction. Moreover, the 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 contain the same discretization for each test case, e.g., there exists no discretization with 9.500 elements for test case 1, see Table 7.2.



Example 1: Single load case

Example 2: Single load case

Example 3: Multiple load case

Example 4: Single load case

Example 5: Single load case

Fig. 7.2: Test cases, see Bogomolny [13]

The results are illustrated, by plotting the traces in each finite element, see Bodnár [12]. The palette is given in Figure 7.3, where red denotes stiff material while the material vanishes in dark blue regions. The data of the corresponding test cases is given in Table 7.2.

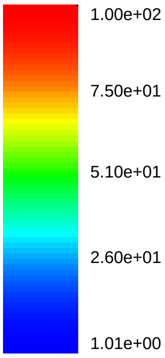


Fig. 7.3: Palette of graphical presentation

The graphical presentation of the optimal material for test cases given in Figure 7.2 are presented in Figure 7.4 - 7.16.

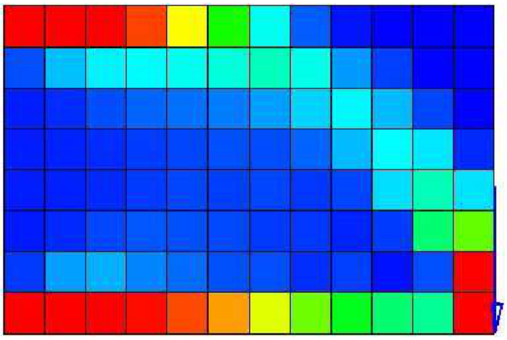


Fig. 7.4: Single load problem 1.1 with 96 elements

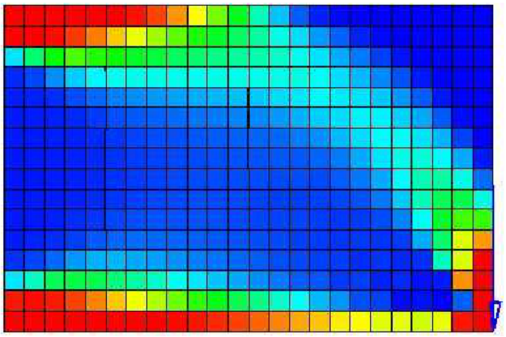


Fig. 7.5: Single load problem 1.2 with 384 elements

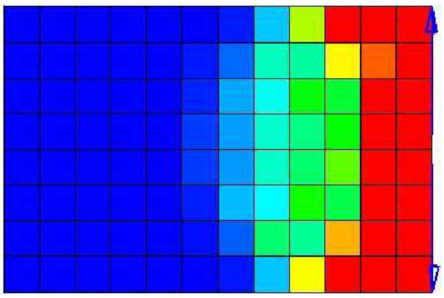


Fig. 7.6: Single load problem 2.1 with 96 elements

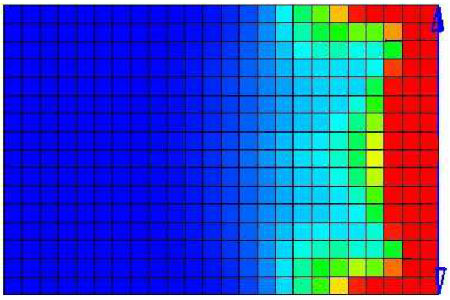


Fig. 7.7: Single load problem 2.2 with 384 elements

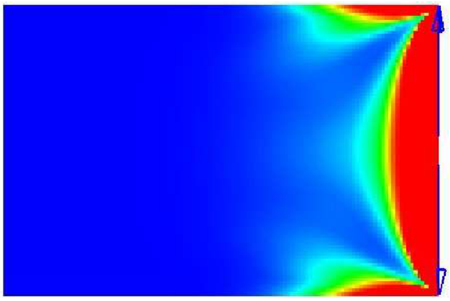


Fig. 7.8: Single load problem 2.3 with 9.500 elements

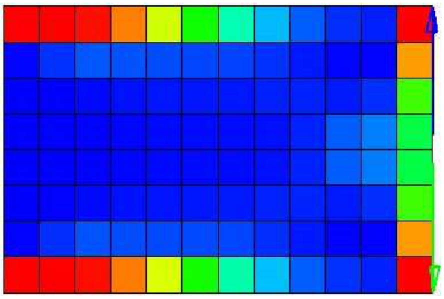


Fig. 7.9: Single load problem 3.1 with 96 elements

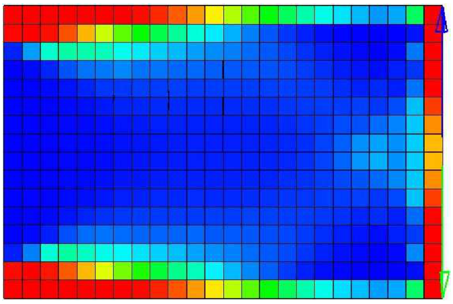


Fig. 7.10: Single load problem 3.2 with 384 elements

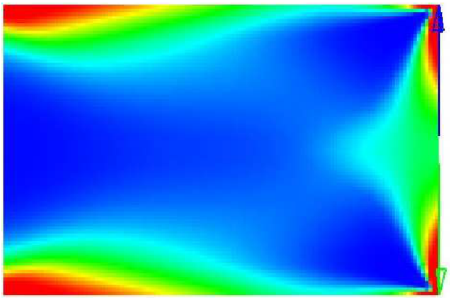


Fig. 7.11: Single load problem 3.3 with 9.500 elements

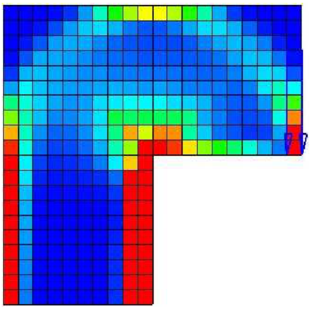


Fig. 7.12: Single load problem 4.1 with 300 elements

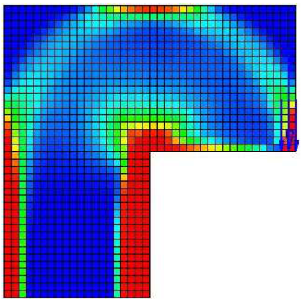


Fig. 7.13: Single load problem 4.2 with 1.300 elements

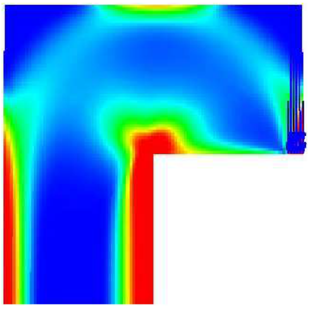


Fig. 7.14: Single load problem 4.3 with 7.500 elements

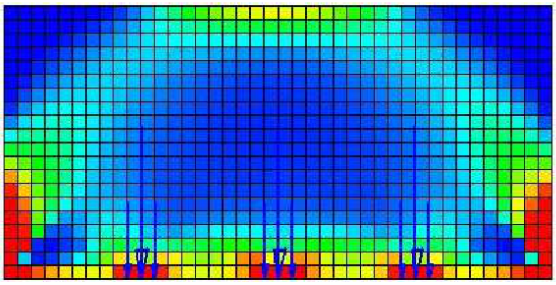


Fig. 7.15: Single load problem 5.1 with 800 elements

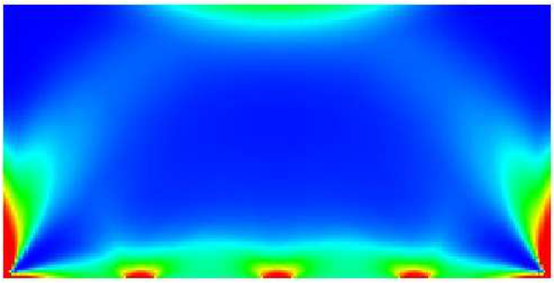


Fig. 7.16: Single load problem 5.2 with 20.000 elements

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

Tab. 7.2: Data of numerical results

Moreover, problem 4.1 is extended by stress constraints (6.32), yielding an increase of the number of regular constraints to 602. A graphical presentation of the stresses and the resulting optimal structure is given in Figure 7.17. The plot on the left hand side shows the density of the material in each finite element according to the values of the traces, analogously to the previous results. On the right hand side, the values of the stress constraints and the corresponding palette are shown for each finite element.

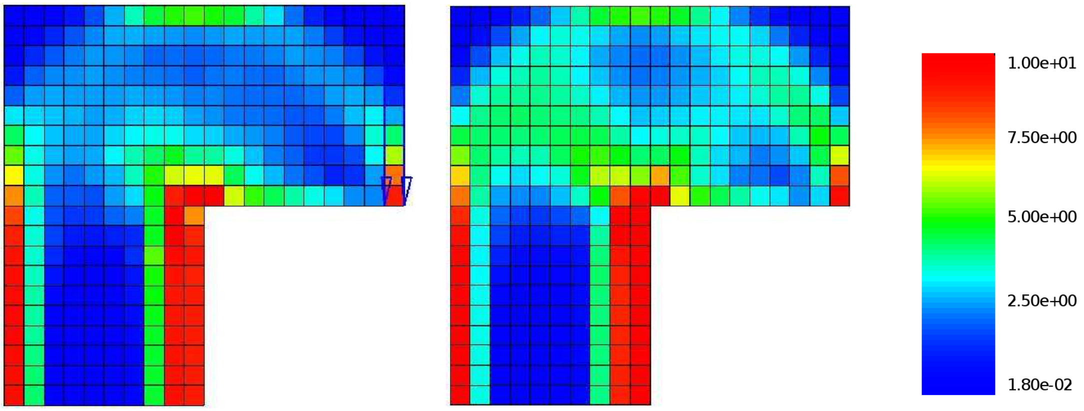


Fig. 7.17: Single load problem 4.1 including stress constraints

Comparing the results of test case 4.1, see Figure 7.12, and test 4.1stress, see Figure 7.17, we observe a significant difference caused by the introduction of stress constraints. As stresses are additional constraints, the objective function is increased yielding a structure which is less stiff. As the given amount of material is equivalent for both test cases, only the distribution of the material differs in the solution. The main difference are the bars on the left and right hand side, which stabilizes the structure and thus ensure that the stress constraints are satisfied.
As the gradients of the stresses are dense and the number of constraints are increasing dependent on the discretization it is not possible to consider problem 4.2 and 4.3 including stresses.

7.3.2 Application in Petroleum Engineering

Apart from free material optimization, there are other applications where infeasibility of some constraints needs to be prevented in order to successfully solve the corresponding optimization problem. This difficulty arises for example in petroleum engineering for Lift Gas optimization, see Camponogara and Nakashima [16].
We proceed from an idealized optimization problem of the form

$$\begin{array}{lll} \min\limits_{x} & f(x) & x \in \mathbb{R}^{12} \\ \text{s.t.} & c_j(x) \leq 0, & j = 1, \ldots, 9 \\ & \underline{x}_i \leq x_i \leq \overline{x}_i, & i = 1, \ldots, 12 \end{array} \tag{7.45}$$

with

$$\begin{aligned} f(x) &:= -\sum_{j=1}^{4} g_j(x) && (7.46)\\ c_j(x) &:= r_{17} + r_{18} f(x) - x_{j+4}, \quad j = 1, \ldots, 4 && (7.47)\\ c_5(x) &:= x_1 x_9 + x_2 x_{10} + x_3 x_{11} + x_4 x_{12} - r_{19} && (7.48)\\ c_{j+5}(x) &:= x_{j+8}\left(x_{j+4} - r_{j+4}\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}}\right), \quad j = 1, \ldots, 4 && (7.49) \end{aligned}$$

$$\begin{aligned} 0 \;\le\; & x_i \le r_{19}, && i = 1, \ldots, 4 && (7.50)\\ r_{17} \;\le\; & x_i \le r_i, && i = 5, \ldots, 8 && (7.51)\\ 0 \;\le\; & x_i \le 1, && i = 9, \ldots, 12 && (7.52) \end{aligned}$$

with constant values $r_i > 0,\ i = 1, \ldots, 19$. The functions $g_j(x),\ j = 1, \ldots, 4$, are defined by

$$g_j(x) := r_{j+12}\left[\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8}}} + \left(1 - \frac{x_j x_{j+8} + r_j}{r_{j+8}}\right)\right]\sqrt{\widetilde{e}_j(x)} \qquad (7.53)$$

with

$$\begin{aligned} \widetilde{e}_j(x) &:= \frac{e_j(x)}{\overline{e}_j(x)} && (7.54)\\ e_j(x) &:= x_{j+4} - r_{j+4}\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}} && (7.55)\\ \overline{e}_j(x) &:= -r_{j+4}\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}} < 0. && (7.56) \end{aligned}$$

$\widetilde{e}_j(x),\ j = 1, \ldots, 4$, might become negative, within the optimization process. To prevent negative values of $\widetilde{e}_j(x),\ j = 1, \ldots, 4$, and thus ensure that (7.53) is well defined, feasibility constraints are introduced. The constant values $r_i,\ i = 1, \ldots, 19$, are larger than zero, ensuring that

$$\frac{x_j x_{j+8} + r_j}{r_{j+8}} \ge 0 \qquad (7.57)$$

holds. As $\overline{e}_j(x) < 0,\ j = 1, \ldots, 4$, we introduce $e_j(x),\ j = 1, \ldots, 4$, given by (7.55), as feasibility constraints. This leads to four feasibility constraints

$$e_j(x) := x_{j+4} - r_{j+4}\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}} \le -\varepsilon \qquad (7.58)$$

where $\varepsilon > 0$ is introduced to prevent numerical instabilities. Otherwise, the computation of the gradient of $\sqrt{\widetilde{e}_j\left(x\right)}$, $j = 1, \ldots, 4$, see (7.53), fails, if the denominator becomes zero.
Moreover, we can omit the inequality constraints $c_6\left(x\right), \ldots, c_9\left(x\right)$, as

$$c_{j+5}\left(x\right) \quad = \quad x_{j+8} e_j\left(x\right), \quad j = 1, \ldots, 4 \tag{7.59}$$

and $x_{j+8} \in [0, 1]$ holds.
The first and second order derivatives of the feasibility constraint $e_j\left(x\right), j = 1, \ldots, 4$, are given by

$$\frac{\partial e_j\left(x\right)}{\partial x_i} \quad = \quad \begin{pmatrix} -\dfrac{r_{j+4} x_{j+8}}{2\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}}\left(r_{j+8} + r_j\right)} \\ 1 \\ -\dfrac{r_{j+4} x_j}{2\sqrt{\frac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}}\left(r_{j+8} + r_j\right)} \end{pmatrix} \tag{7.60}$$

$$\frac{\partial^2 e_j\left(x\right)}{\partial x_i \partial x_q} \quad = \quad \begin{pmatrix} a & 0 & b \\ 0 & 0 & 0 \\ b & 0 & c \end{pmatrix} \tag{7.61}$$

with

$$a \quad := \quad \frac{r_{j+4} x_{j+8}^2}{4\left(\dfrac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}\right)^{3/2}\left(r_{j+8} + r_j\right)^2}$$

$$b \quad := \quad \frac{r_{j+4} x_j x_{j+8}}{4\left(\dfrac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}\right)^{3/2}\left(r_{j+8} + r_j\right)^2} - \frac{r_{j+4}}{2\sqrt{\dfrac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}}\left(r_{j+8} + r_j\right)}$$

$$c \quad := \quad \frac{r_{j+4} x_j^2}{4\left(\dfrac{x_j x_{j+8} + r_j}{r_{j+8} + r_j}\right)^{3/2}\left(r_{j+8} + r_j\right)^2}$$

It can be seen easily, that $e_j\left(x\right), \ j = 1, \ldots, 4$, is convex as the corresponding Hessian matrix is positive semidefinite. In total, we can reformulate (7.45) by

$$\begin{array}{lll} \min\limits_{x} & f\left(x\right) & x \in \mathbb{R}^{12} \\ \text{s.t.} & c_j\left(x\right) \leq 0, & j = 1, \ldots, 5 \\ & e_j\left(x\right) \leq -\varepsilon, & j = 1, \ldots, 4 \\ & \underline{x}_i \leq x_i \leq \overline{x}_i, & i = 1, \ldots, 12 \end{array} \tag{7.62}$$

This leads to the following subproblem in iteration k

$$\begin{array}{lll}\min\limits_{x,s} & f^{(k)}(x) & x \in \mathbb{R}^{12}, s \in \mathbb{R}^{9} \\ \text{s.t.} & c_j^{(k)}(x) + s_j = 0, & j = 1, \ldots, 5 \\ & e_j(x) + s_{j+5} = -\varepsilon, & j = 1, \ldots, 4 \\ & \underline{x}_i^{(k)} \leq x_i \leq \overline{x}_i^{(k)}, & i = 1, \ldots, 12 \\ & 0 \leq s_i, & i = 1, \ldots, 9 \end{array} \tag{7.63}$$

where $f^{(k)}(x)$ is the convex approximation of $f(x)$ according to (4.21) and $c_j^{(k)}(x)$, $j = 1, \ldots, 5$, is the approximation of $c_j(x)$, $j = 1, \ldots, 5$, given by (4.5). Moreover, $\underline{x}^{(k)}$, $\overline{x}^{(k)} \in \mathbb{R}^n$ are defined by (4.8) and (4.9).

Algorithm 16 is applied to solve problem (7.62), using the constant values

$$\begin{array}{llll} r_1 := 500 & r_2 := 400 & r_3 := 100 & r_4 := 100 \\ r_5 := 1500 & r_6 := 1400 & r_7 := 1000 & r_8 := 900 \\ r_9 := 3000 & r_{10} := 3000 & r_{11} := 3000 & r_{12} := 3000 \\ r_{13} := 2500 & r_{14} := 2700 & r_{15} := 2000 & r_{16} := 2400 \\ r_{17} := 250 & r_{18} := 0.02 & r_{19} := 400 & \end{array} \tag{7.64}$$

The internal parameters of the Algorithm are given in Table 9.3, see Appendix. The specific constants of problem (7.62) are summarized in the following table.

Parameter	Value
$x_i^{(0)}, \ i = 1, 2$	10
$x_i^{(0)}, \ i = 3, 4$	100
$x_i^{(0)}, \ i = 5, \ldots, 9$	250
$x_i^{(0)}, \ i = 10, \ldots, 12$	1
ε	1.D-5

Tab. 7.3: Parameters solving Lift-Gas problems

The active set parameter a is set to 1.D10, while the stopping accuracy is 1.D-5. The algorithm creates the following iteration sequence within 1 second.

IT	ITSUB	ACT	FEASIBILITY	OBJECTIVE	SIGMA	NORM(DX)	NORM(L_X)
0	—	—	0.4000D+03	-.3190D+02	—	—	0.14D+01
1	75	9	0.2598D+02	−.3090D+02	0.10D+01	0.42D+02	0.27D+00
2	58	9	0.4092D+00	−.3124D+02	0.10D+01	0.53D+01	0.10D+00
3	70	9	0.1961D−01	−.3123D+02	0.10D+01	0.15D+01	0.22D−02
4	90	9	0.0000D+00	−.3123D+02	0.10D+01	0.57D−01	0.18D−03
5	69	9	0.0000D+00	−.3123D+02	0.10D+01	0.66D−01	0.58D−03
6	72	9	0.0000D+00	−.3123D+02	0.10D+01	0.16D−01	0.42D−03
7	66	9	0.0000D+00	−.3123D+02	0.10D+01	0.18D−01	0.41D−03
8	62	9	0.0000D+00	−.3123D+02	0.10D+01	0.37D−01	0.56D−03
9	77	9	0.0000D+00	−.3123D+02	0.10D+01	0.17D−01	0.39D−03
10	62	9	0.0000D+00	−.3123D+02	0.10D+01	0.61D−01	0.56D−03
11	58	9	0.0000D+00	−.3123D+02	0.10D+01	0.16D−01	0.26D−03
12	49	9	0.0000D+00	−.3123D+02	0.10D+01	0.25D−01	0.22D−03
13	49	9	0.0000D+00	−.3123D+02	0.10D+01	0.26D−01	0.21D−03
14	44	9	0.0000D+00	−.3123D+02	0.10D+01	0.59D−01	0.24D−03
15	62	9	0.0000D+00	−.3123D+02	0.10D+01	0.82D−02	0.94D−04
16	60	9	0.0000D+00	−.3123D+02	0.10D+01	0.58D−02	0.11D−03
17	53	9	0.0000D+00	−.3123D+02	0.10D+01	0.14D−01	0.71D−04
18	49	9	0.0000D+00	−.3123D+02	0.10D+01	0.90D−02	0.43D−04
19	47	9	0.0000D+00	−.3123D+02	0.10D+01	0.16D−01	0.69D−04
20	64	9	0.0000D+00	−.3123D+02	0.10D+01	0.71D−02	0.28D−04
21	58	9	0.0000D+00	−.3123D+02	0.10D+01	0.76D−02	0.20D−04
22	61	9	0.0000D+00	−.3123D+02	0.10D+01	0.85D−02	0.33D−04
23	54	9	0.0000D+00	−.3123D+02	0.10D+01	0.38D−02	0.41D−04
24	62	9	0.0000D+00	−.3123D+02	0.10D+01	0.21D−02	0.15D−04
25	57	9	0.0000D+00	−.3123D+02	0.10D+01	0.30D−02	0.48D−04
26	67	9	0.0000D+00	−.3123D+02	0.10D+01	0.33D−02	0.22D−04
27	90	9	0.0000D+00	−.3123D+02	0.10D+01	0.14D−02	0.39D−04
28	66	9	0.0000D+00	−.3123D+02	0.10D+01	0.27D−02	0.53D−04
29	80	9	0.0000D+00	−.3123D+02	0.10D+01	0.18D−02	0.58D−05

Note that from iteration 4 onwards, only the Lagrangian multipliers change until the stopping criterion based on the KKT conditions is satisfied.
The optimal solution is given by

$$\begin{array}{lll} x_1 := 368.68646 & x_2 := 369.73634 & x_3 := 394.87975 \\ x_4 := 399.41895 & x_5 := 256.24802 & x_6 := 256.24762 \\ x_7 := 256.24708 & x_8 := 256.24703 & x_9 := 0.18448106E-05 \\ x_{10} := 0.19999825E-05 & x_{11} := 0.42875094 & x_{12} := 0.57757105 \end{array} \tag{7.65}$$

8. CONCLUSION AND OUTLOOK

The main focus of this dissertation is the development of a strictly feasible sequential convex programming (SCPF) method. SCPF guarantees that all iterates retain feasible subject to a subset of special constraints called feasibility constraints. The remaining constraints, so-called regular constraints, may be violated. In contrast to feasible direction SQP methods and other algorithms yielding a feasible sequence of iterates, SCPF is applicable, if the regular constraints are only well defined on the feasible set of the feasibility constraints. Although these algorithms generate a sequence of feasible iterates, they require function and gradient evaluations at infeasible points, i.e., the feasibility constraints are not necessarily satisfied whenever function and gradient evaluations are needed. SCPF solves continuous nonlinear programs iteratively, by a sequence of convex subproblems, where the special structure can be exploited. On subproblem level, the objective function as well as the regular constraints are replaced by convex and separable approximations, while the feasibility constraints are included directly. A line search procedure guarantees global convergence. To ensure feasibility with respect to feasibility constraints, even if the stepsize is reduced, we require convexity for these special constraints. A convergence proof is given for convex feasibility constraints.

The corresponding computer code SCPF10.f was implemented efficiently such that large scale problems can be solved. If exists, the sparse structure of the gradients and the Hessian is exploited. Moreover, linear constraints are approximated optionally. An active set strategy is applied to reduce the size of the subproblems and thus speed up the solution process. To satisfy feasibility constraints in every main iteration, the active set strategy is only applied for regular constraints. The subproblems are solved by IPOPT, where second order information ensures superlinear convergence on subproblem level.

The main application of SCPF is free material optimization (FMO), where sparse, large-scale optimization problems are to be solved. Proceeding from a finite element discretization the design of a structure is to be optimized, such that it becomes as stiff as possible. The compliance function measures the stiffness of the resulting structure dependent on the material properties in each finite element. In addition, the total amount of material is bounded. Moreover, feasibility constraints are introduced to ensure positive definiteness of the elementary stiffness matrices.

Several different test cases with up to 20.000 finite elements yielding 120.000 variables were successfully solved. Moreover, some of these test cases contain stress constraints, which often cause numerical problems, since they violate constraint qualifications.

Other applications arise in petroleum industry, where square roots of some analytical expressions, need to be computed. The introduction of feasibility constraints prevents these expressions from becoming negative and thus ensures that they are well defined.

From the theoretical as well as the practical point of view, the requirement of convex feasibility constraints is restrictive, but essential if a line search procedure is applied. Using other globalization techniques, convexity is not necessarily needed. An appropriate convergence proof based on filter methods and a corresponding implementation could be the result of further research.

Although SCPF has turned out to be very efficient, additional modifications might improve the computational performance. A reduction of the calculation time is essential to allow the solution of three dimensional FMO problems or a finer discretization. The computation time is expected to be reduced further by introducing an active set strategy on subproblem level.

9. APPENDIX

9.1 Program Documentation

SCPF10.f is a FORTRAN subroutine, which is an extension of SCPIP30.f, see Zillober [104]. To execute the program, the corresponding file has to be compiled and linked with the object codes for function and gradient evaluations provided by the user. All calculations within these subroutines are performed in double precision arithmetic.

```
CALL SCPF (  N,         MIE,       MEQ,      MF,       IEMAX,
             EQMAX,     X0,        X_L,      X_U,      F_ORG,
             H_ORG,     G_ORG,     DF,       Y_IE,     Y_EQ,
             Y_L,       Y_U,       ICNTL,    RCNTL,    INFO,
             RINFO,     NOUT,      R_SCP,    RDIM,     R_SUB,
             RSUBDIM,   I_SCP,     IDIM,     I_SUB,    ISUBDIM,
             ACTIVE,    IERR,      IERN,     IECN,     IEDERV,
             IELPAR,    IELENG,    EQRN,     EQCN,     EQCOEF,
             EQLPAR,    EQLENG,    MACTIV,   SPIW,     SPIWDIM,
             SPDW,      SPDWDIM,   LINEAR,   LACT,     SETACT   )
```

In the following table, the meaning of the parameters is described, which is an adaptation of the manual of SCPIP30.f, see Zillober [104]. Values that have to be set by the user are marked with * in the first column. (*) stands for a value that has to be set by the user on request (reverse communication). The name of the parameters and the corresponding size can be found in the second column. Integer values are identified by 'I' in the third column, double precision values by 'D'. Variables which are not allowed to alter during the optimization process are denoted by (NA).

*	N	I	Number of variables, at least ≥ 1 (NA).
*	MIE	I	Number of inequality constraints (NA).
*	MEQ	I	Number of equality constraints (NA).
*	MF	I	Number of feasibility constraints (NA).
*	IEMAX	I	Dimension of arrays H_ORG, Y_IE, ACTIVE. Must be at least MIE and ≥ 1 (NA).
*	EQMAX	I	Dimension of arrays G_ORG, Y_EQ. Must be at least MEQ and ≥ 1 (NA).
*	X0(N)	D	Current iterate.
*	X_L(N)	D	Lower bounds on the variables (NA).

*	X_U(N)	D	Upper bounds on the variables (NA).
(*)	F_ORG	D	Objective function value evaluated at X0.
(*)	H_ORG(IEMAX)	D	Function values of inequality and feasibility constraints evaluated at X0.
(*)	G_ORG(EQMAX)	D	Function values of equality constraints evaluated at X0.
(*)	DF(N)	D	Gradient of the objective function at X0.
	Y_IE(IEMAX)	D	Lagrange multipliers for inequality and feasibility constraints in current iterate.
	Y_EQ(EQMAX)	D	Lagrange multipliers for equality constraints in current iterate.
	Y_L(N)	D	Lagrange multipliers for the lower bounds on the variables in current iterate.
	Y_U(N)	D	Lagrange multipliers for the upper bounds on the variables in current iterate.
*	ICNTL(13)	I	Integer array to be set by the user. A value 0 indicates that the default values should be chosen (NA). ICNTL(1): desired optimization method: 1: method of moving asymptotes (default) 2: sequential convex programming ICNTL(2): currently not used ICNTL(3): maximum number of iterations. ICNTL(3) ≥ 1. Default: 100 ICNTL(4): desired output level 1: no output 2: only final convergence analysis 3: one line of intermediate results (default) 4: more detailed results ICNTL(5): maximum number of function calls in the line-search procedure (≥ 1). Default: 10 ICNTL(6): Relaxed convergence check. The program terminates, if A)-D) hold A) the current iterate is feasible B) the relative change of the last two succeeding iteration points is less than RCNTL(6) C) the absolute change of the last two objective function values is less than RCNTL(5) D) the relative change of the last two objective function values is less than RCNTL(4). Default: 0 ICNTL(7-13): internal use.

*	RCNTL(6)	D	Double precision array to be set by the user (NA). A value 0 indicates that the default values should be chosen. RCNTL(1): desired final accuracy. Default: 1.D-7 RCNTL(2): double precision number that indicates infinity. RCNTL(2) $\geq$ 1.D10. Default: 1.D30 RCNTL(3): Active set parameter a, see Section 7.1.1. Default: RCNTL(2) RCNTL(4,5,6): see ICNTL(6). If ICNTL(6) $\neq$ 1, then RCNTL(4,5,6) are not used. Defaults: RCNTL(4) = 1.D-2 RCNTL(5) = 1.D-2 RCNTL(6) = 1.D-2
	INFO(23)	I	Integer array containing problem information. INFO(1): number of evaluations of Lagrangian function values INFO(2): number of evaluations of Lagrangian gradients INFO(3): necessary value for RDIM INFO(4): necessary value for RSUBDIM INFO(5): necessary value for IDIM INFO(6): necessary value for SPIWDIM INFO(7): necessary value for SPDWDIM INFO(8): necessary value for ISUBDIM INFO(9-19): internal use INFO(20): current iteration number INFO(21): number of iterations performed to solve last subproblem
	RINFO(5)	D	Double precision array containing some problem information. RINFO(1): residual of the subproblem of the last outer iteration RINFO(2): maximum violation of constraints RINFO(3): stepsize in last main iteration RINFO(4): norm of the difference of the last two iteration points RINFO(5): norm of the gradient of the Lagrangian computed at the last iterate
*	NOUT	I	Output unit number (NA).
	R_SCP(RDIM)	D	Double precision working array of dimension at least RDIM.
*	RDIM	I	$\geq$ 44*N+18*IEMAX+10*EQMAX+2*IELPAR +20+ICNTL(5) (NA).

	R_SUB(RSUBDIM)	D	Double precision working array of dimension at least RSUBDIM.
*	RSUBDIM	I	≥22*N+41*IEMAX+2*IELPAR+30 (NA).
	I_SCP(IDIM)	I	Integer working array of dimension at least IDIM.
*	IDIM	I	≥7*N+8*IEMAX+2*EQMAX+3*IELPAR+15 (NA).
	I_SUB(ISUBDIM)	I	Integer working array of dimension at least ISUBDIM.
*	ISUBDIM	I	≥2*N+3*IEMAX + IELPAR+5 (NA).
	ACTIVE(IEMAX)	I	Active set. Constraint i is part of the active set ↔ ACTIVE(i) = 1. Only gradients corresponding to the active set have to be updated.
*	IERR	I	Initialization: 0 (NA). On return: <0 : Reverse communication: -1: function values are requested -2: gradients are requested 0: successful computation 1: maximum number of iterations exceeded 2: N ≤ 0 3: MIE < 0 or MEQ < 0 or IEMAX < 1 or EQMAX < 1 4: MIE > IEMAX or MEQ > EQMAX 5: IELPAR or EQLPAR < 1 6: for at least one component, the lower bound is greater or equal to the upper bound. 8: RDIM too small, cf. INFO(3) 9: RSUBDIM too small, cf. INFO(4) 901: ISUBDIM too small, cf. INFO(8) 10: IDIM too small, cf. INFO(5) 120, 121 (12/0,12/1): SPDWDIM too small, see INFO(7). 130, 131, 132, 133, 134, 135 (13/0..5): SPIWDIM too small, see INFO(6) 14: the user provided for at least one constraint an empty column in the Jacobian. 15: the Jacobian matrices are not stored columnwise 16: the Jacobian matrices are not stored correctly. For at least one column the components are out of order. 161: (16/1) the Jacobian matrices are not stored correctly. More columns than constraints are provided.

20: the feasible region of the current subproblem is a singleton. One possible reason is, that the feasible region of the original problem is empty!

21: for at least one component, lower and upper bound of a subproblem are almost equal. The interior-point subproblem solver is not applicable!

22: line-search needs too much function evaluations

23: the norm of the gradient of the Lagrangian is close to 0 and no feasible solution of the subproblem is found. Together, it is very likely, that the feasible region is empty.

24: the current value for IELENG/EQLENG is larger than IELPAR/EQLPAR

30: error during solution of subproblem.

31: the subproblem could not be solved within the maximum number of iterations.

(*)	IERN(IELPAR)	I	Row indices of the entries of the Jacobian of the inequality and feasibility constraints (NA).
(*)	IECN(IELPAR)	I	Column indices of the entries of the Jacobian of the inequality and feasibility constraints (NA).
(*)	IEDERV(IELPAR)	D	Values of the entries of the Jacobian of the inequality and feasibility constraints. Only nonzero elements have to be stored, but zero elements are allowed. The three arrays IERN, IECN and IEDERV are expected to be sorted by function numbers and inside one function by component numbers, i.e., it is expected that the Jacobian is stored columnwise (NA).
*	IELPAR	I	Dimension of arrays IERN, IECN and IEDERV. Must be at least IELENG and ≥ 1 (NA).
(*)	IELENG	I	Current number of entries in IEDERV (NA).
(*)	EQRN(EQLPAR)	I	Row indices of the entries of the Jacobian of the equality constraints (NA).
(*)	EQCN(EQLPAR)	I	Column indices of the entries of the Jacobian of the equality constraints (NA).
(*)	EQCOEF(EQLPAR)	D	Values of the Jacobian of the equality constraints. Only nonzero elements have to be stored, but zero elements are allowed. The three arrays EQRN, EQCN and EQCOEF are expected to be sorted by function numbers and inside one function by component numbers, i.e., it is expected that the Jacobian is stored columnwise (NA).

*	EQLPAR	I	Dimension of arrays EQRN, EQCN and EQCOEF. Must be at least EQLENG and ≥1 (NA).
(*)	EQLENG	I	Current number of entries in EQCOEF (NA).
	MACTIV	I	Number of constraints included in the active set.
	SPIW(SPIWDIM)	I	Integer working array of dimension at least SPIWDIM.
*	SPIWDIM	I	Has to be at least (NA):
	SPDW(SPDWDIM)	D	Double precision working array of dimension at least SPDWDIM. For the handling see SPIW.
*	SPDWDIM	I	Has to be at least (NA):
*	LINEAR(IEMAX)	I	Input array indicating linear constraints, LINEAR(i) = 0 ↔ constraint i is linear.
*	SETACT(IEMAX)	I	Input array specifying constraints to be active in every iteration, see Section 7.1.1.
*	LACT	I	Length of SETACT during the optimization process

The algorithm is based on reverse communication. The parameter IERR indicates the required type of evaluation.

IERR = -1 Function evaluation for all constraints and objective, stored in H_ORG(1),…,H_ORG(MIE+MF),G_ORG(1),…,G_ORG(MEQ) and F_ORG respectively.

IERR = -2 Gradient evaluation for the objective and for all constraints included in the active set. The gradients of $c_j(x),\ j = 1,\dots,m_e$ are stored at positions 1,…,EQLENG, of EQRN, EQCN and EQCOEF. The gradients of $c_j(x),\ j = m_e+1,\dots,m_c$ and $e_j(x),\ j = 1,\dots,m_f$ are both stored in IERN, IECN and IEDERV, starting with the regular inequalities. The total number of entries is denoted by IELENG. Active constraints are indicated by the array ACTIVE(j)=1. The gradient of the objective is stored in dense format in DF.

Note, that whenever gradients are required, a previous call of function values was made at the same iterate X0 and F_ORG, H_ORG and G_ORG contain function values of objective and constraints at X0. X0, F_ORG, H_ORG and G_ORG are not allowed to be altered during the computation of gradients.

Moreover, the function and gradient evaluations have to be computed on subproblem level. Due to the approximation scheme of SCPF, the user has to adapt the following routines, to provide the evaluation of the feasibility constraints in the current iterate.

Function evaluation:

```
SUBROUTINE EV_C(  N1,   XWS,   MIE2,   H_APP,   DAT,   IDAT  )
```

Gradient evaluation:

```
SUBROUTINE EV_A(  TASK,  N1,    XWS,   NZ,   VAL,   COL,   ROW,
                  DAT,   IDAT  )
```

Evaluation of Hessian:

```
SUBROUTINE EV_H(  TASK,   N1,     XWS,    M,      LAM,   NNZH,
                  VOUT,   IRNH,   ICNH,   DAT,    IDAT   )
```

The variables are defined as follows

	N1	I	Number of variables including slack variables (NA).
	XWS(N)	D	Current iterate.
	M	I	Number of equality constraints (NA).
(*)	H_APP(M)	D	Value of constraints at current iterate.
	DAT(*)	D	Double precision working array (NA).
	IDAT(*)	I	Integer working array (NA).
	TASK	I	= 0, Compute maximal number of gradient or Hessian entries = 1, Compute gradients or Hessian
(*)	NZ	I	Maximal number of gradient entries (NA).
(*)	VAL(NZ)	D	Values of gradients. Only nonzero elements have to be stored, but zero elements are allowed.
(*)	COL(NZ)	I	Column indices of gradient.
(*)	ROW(NZ)	I	Row indices of gradient.
	LAM(M)	D	Lagrangian multipliers.
(*)	NNZH	I	Maximal number of Hessian entries (NA).
(*)	VOUT(NNZH)	D	Values of the Hessian. Only nonzero elements have to be stored, but zero elements are allowed.
(*)	IRNH(NNZH)	I	Row indices of the Hessian.
(*)	ICNH(NNZH)	I	Column indices of the Hessian

For a more detailed description see Wächter and Biegler [90]. Note, that IPOPT solves an extended subproblem including slack variables. Therefore, the subroutine to compute the gradients of the feasibility constraints within the subproblem solution process has to be adapted.

The internally used parameters are given in the subsequent table.

Name	Value
$L_{\min}$	-1.D5
$U_{\max}$	1.D5
ξ	0.5
$T1$	0.7
$T2$	1.15
ω	0.9
r	1.D-2
$y_j^{(0)},\ j = 1, \ldots, m_c + m_f$	0.D0
$\rho_j^{(-1)},\ j = 1, \ldots, m_c + m_f$	1.D0
κ_1	2
κ_2	10
β	5.D-1
β_2	1.D-2
τ	$\tau^{(k)} = \max\left\{1.\text{D-}6,\ 1.\text{D-}5\left\|\nabla f(x^{(k)})\right\|_\infty\right\}$

Tab. 9.3: Internally used parameters

BIBLIOGRAPHY

[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] J.R. Birge, L. Qi, and Z. Wei. A variant of the Topkis-Veinott method for solving inequality constrained optimization problems. Technical report, School of Mathematics, University of New South Wales, Sydney, Australien, 1997.

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

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

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

[15] R.H. Byrd, R.B. Schnabel, and G.A. Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal of Numerical Analysis*, 24:1152–1170, 1987.

[16] E. Camponogara and P.H.R. Nakashima. Solving a gas-lift optimization problem by dynamic programming. *European Journal of Operational Research*, 174, 2006.

[17] R.M. Chamberlain, C. Lemarechal, H.C. Pedersen, and M.J.D. Powell. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Study*, 16:1–17, 1982.

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

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

[20] S. Ertel, K. Schittkowski, and Ch. Zillober. D26 - sequential convex programming for free material optimization. Technical report, PLATO-N public report PU-R-4-2007, 2007.

[21] 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.

[22] S. Ertel, K. Schittkowski, and Ch. Zillober. A feasible sequential convex programming method for free material optimization. In *8th World Congress on Structural and Multidisciplinary Optimization*, 2009.

[23] R. Fletcher. An exact penalty function for nonlinear programming with equalities. *Mathematical Programming 5*, pages 129–150, 1973.

[24] R. Fletcher. An ideal penalty function for constrained optimization. *Nonlinear Programming 2*, pages 121–164, 1975.

[25] R. Fletcher. *Second order corrections for non-differentiable optimization*, volume 912 of *Lecture Notes in Mathematics*, pages 85–114. Springer Berlin / Heidelberg, 1982.

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

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

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

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

[30] M. Fukushima. A succesive quadratic programming algorithm with global and superlinear convergence properties. *Mathematical Programming*, 35:253–264, 1986.

[31] Z.Y. Gao, G.P. He, and F. Wu. A method of sequential systems of linear equations with arbitrary initial point. *Science in China Series A: Mathematics*, 40(6):561–571, 1997.

[32] C. Geiger and C. Kanzow. *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben.* Springer, 1999.

[33] N.I.M Gould and Ph.L. Toint. *Global convergence of a non-monotone trustregion filter algorithm for nonlinear programming*, pages 125–150. Multiscale Optimization Methods and Applications, 2006.

[34] C. Großmann and H. Kleinmichel. *Verfahren der nichtlinearen Optimierung.* Teubner-Verlag, 1976.

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

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

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

[38] J. Herskovits and L.A.V. Carvalho. A successive quadratic programming based feasible directions algorithm. *Analysis and Optimization Systems*, 83:93–101, 1986.

[39] J. Herskovits and G. Santos. Feasible arc interior point algorithms for nonlinear optimization. *Computational Mechanics, New Trends and Applications*, 1998.

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

[41] Q.-J. Hu, W.Y. Chen, and Y.-H. Xiao. An improved active set feasible SQP algorithm for the solution of inequality constrained optimization problems. *Journal of Computational Analysis and Applications*, 11(1):54–63, 2009.

[42] V.S. Ishutkin and C. Großmann. Verfahren der zulässigen Richtungen unter Benutzung reduzierter Gradienten für nichtlineare Optimierungsprobleme. *Optimization*, (16):374–390, 1985.

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

[44] J.-B. Jian and C.-M. Tang. An SQP feasible descent algorithm for nonlinear inequality constrained optimization without strict complementarity. *Computers and Mathematics with Applications*, 49:223–238, 2005.

[45] J.-B. Jian, C.-M. Tang, Q.-J. Hu, and H.Y. Zheng. A feasible descent SQP algorithm for general constrained optimization without strict complementarity. *Journal of Computational and Applied Mathematics*, 180:391–412, 2005.

[46] 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.

[47] 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.

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

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

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

[51] 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.

[52] C.T. Lawrence. *A computational efficient feasible sequential quadratic programming algorithm.* PhD thesis, University of Maryland, 1998.

[53] C.T. Lawrence and A.L. Tits. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software*, 6:265–282, 1996.

[54] C.T. Lawrence and A.L. Tits. A computationally efficient feasible sequential quadratic programming algorithm. *SIAM J. OPTIM.*, 11(4):1092–1118, 2001.

[55] 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.

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

[57] N. Maratos. *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems.* PhD thesis, Imperial College, London, 1978.

[58] D.Q. Mayne and E. Polak. Feasible directions algorithms for optimization problems with equaltiy and inequality constraints. *Mathematical Programming*, 11:67–80, 1976.

[59] D.Q. Mayne and E. Polak. A superlinearly convergent algorithm for constrained optimization problems. *Mathematical Programming Study*, 16:45–61, 1982.

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

[61] 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.

[62] E.O. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints.* PhD thesis, University of Colorado at Boulder, USA, 1989.

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

[64] E.R. Panier and A.L. Tits. A superlinearly convergent feasible method for the solution of inequality constrained optimization. *SIAM J. Control and Optimization*, 25(4):934–950, 1987.

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

[66] 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.

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

[68] M.J.D. Powell and Y. Yuan. A recursive quadratic programming algorithm that uses differentiable exact penalty function. *Mathematical Programming*, 35:265–278, 1986.

[69] M.J.D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49:189–211, 1991.

[70] A. Ren, F. Duan, Z. Zhu, and Z. Luo. A feasible SQP method with superlinear convergence for general constrained optimization. *Journal of Applied Sciences*, 7(10):1422–1427, 2007.

[71] 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.

[72] 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), 1961.

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

[74] K. Schittkowski. Mathematische Grundlagen von Optimierungsverfahren, 1999. Script.

[75] K. Schittkowski and Y. Yuan. Sequential quadratic programming methods. *Wiley Encyclopedia of Operations Research and Management Science*, 2010. To appear.

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

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

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

[79] 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.

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

[81] 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.

[82] 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.

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

[84] K. Svanberg. MMA and GCMMA, versions september 2007. Technical report, Department of Mathematics, KTH, Stockholm, 2007.

[85] D.M. Topkis and A.F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM Journal on Control*, 5:268–279, 1967.

[86] M. Ulbrich and S. Ulbrich. Non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function. *Mathematical Programming Ser B*, 95:103–135, 2003.

[87] S. Ulbrich. On the superlinear local convergence of a filter-SQP method. *Mathematical Programming*, 100:217–245, 2004.

[88] R.J. Vanderbei and H.Y. Benson. On formulating semidefinite programming problems as smooth convex nonlinear optimization problems. Technical report, 2000.

[89] A. Vardi. A trust region algorithm for equaltiy constrained minimization: convergence properties and implementation. *SIAM Journal on Numerical Analysis*, 22:575–591, 1985.

[90] 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.

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

[92] T. Werner. Interior point methods with second order derivatives for solving large scale nonlinear programming problems. Diplomarbeit, Department of Computer Science, University of Bayreuth, 2006.

[93] J.L. Zhou and A.L. Tits. User's guide for FSQP version 3.5: A fortran code for solving nonlinear (minimax) optimization problems, generating iterates satisfying all inequality and linear constraints. Technical report, 1995.

[94] Z. Zhou, K. Zhang, and J. Jian. An improved SQP algorithm for inequality constrained optimization. *Mathematical Methods of Operation Research*, 58:271–282, 2003.

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

[96] Z. Zhu and J. Jian. An efficient feasible SQP algorithm for inequality constrained optimization. *Nonlinear Analysis: Real World Applications*, 10:1220–1228, 2009.

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

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

[99] Ch. Zillober. A practical interior-point method for a nonlinear programming problem arising in sequential convex programming. Technical report, 1998.

[100] Ch. Zillober. A combined convex approximation – interior point approach for large scale nonlinear programming. *Optimization and Engineering*, 2(1):51–73, 2001.

[101] Ch. Zillober. Global convergence of a nonlinear programming method using convex approximations. *Numerical Algorithms*, 27(3):256–289, 2001.

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

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

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

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

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

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

Acknowledgement

Finally, I would like to thank all those people who supported and accompanied me during my time as a PhD student.

Especially, I am very thankful to my supervisor Prof. Dr. Klaus Schittkowski, who supported and encouraged me during the last years. His expertise and helpful mentoring allowed me to provide this scientific work and to finalize this thesis.
I am also very grateful to my co-advisor PD Dr. Christian Zillober. I learned and profited a lot from his knowledge. His detailed and constructive comments have been a great value throughout this study.
Further, I would like to thank Prof. Dr. Michael Stingl for the academic discussions and the collaboration. His expertise, guidance and valuable advice have been very helpful for this study.

My academic work was funded by the European project PLATO-N and a scholarship of the 'Bayerische Staatsregierung'. I am very thankful for this support and the great opportunity of a cooperation with industrial partners and scientific institutions. I enjoyed the work with each partner very much and I profited a lot due to the support that I obtained.

I am especially grateful to Prof. Dr. Dr. Martin Bendsøe and Prof. Dr. Mathias Stolpe, who invited me to work at the DTU as part of a student exchange program. Their profound knowledge and the productive collaboration (not only during this time) had a strong influence on my research.
It has been a pleasure to join the mathematical institute of the DTU. I enjoyed the exchange of knowledge and to learn more about Denmark. Thank you to everyone who made the time in Lyngby so exciting: Steffi, Julia, Eduardo, Geoffrey, Mathias and Patrick.
Tak Steffi, for your never ending patience, your encouraging support and the great time we spent together with Anja.

I would like to thank all the members of our department for the very kind and friendly atmosphere during work: Mrs. Lachmann, Axel, Björn, Oliver, Thomas and Thorsten. Special thanks to Oliver for the endless and helpful discussions about convergence proofs and algorithms, but especially for the daily coffee break, that helped to get the head clear.

Freunde sind einzigartig und man sollte sie an dieser Stelle nicht vergessen, obwohl sie nicht unbedingt direkt zur Entstehung einer Doktorarbeit beitragen. Deswegen möchte ich mich bei meinen lieben Freunden bedanken - weil ich immer auf Euch zählen kann:

'meine Bayreuther Mathematiker und Physiker', für die zahlreichen Diskussionen, aber vor allem für die Zeit in der es nicht um Mathematik ging.

die Erlanger-Clique - Alex, Jutta, Kuddel und Tine - weil ihr mich seit Jahren auf meinem Weg begleitet.

Mein größter Dank gilt aber meiner Familie, besonders

meinem Mann Thomas, weil Du stets aufbauende Worte und Zeit für meine Probleme findest und ich für Dich immer an erster Stelle stehe.

meinem Bruder Dirk, weil Du bei wissenschaftlichen Fragen aber besonders als großer Bruder immer für mich da bist.

und vor allem meinen Eltern, weil ihr mir immer zur Seite steht und mich mit viel Geduld und guten Ratschlägen unterstützt.

Sonja

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Bayreuth, den 9. Juni 2011

Sonja Lehmann
Ludwig-Thoma-Str 32e
95447 Bayreuth