Downloaded from orbit.dtu.dk on: Nov 08, 2017

Technical University of Denmark

Mathematical programming methods for large-scale topology optimization problems

Rojas Labanda, Susana; Stolpe, Mathias; Sigmund, Ole

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Rojas Labanda, S., Stolpe, M., & Sigmund, O. (2015). Mathematical programming methods for large-scale topology optimization problems. DTU Wind Energy.

DTU Library
Technical Information Center of Denmark

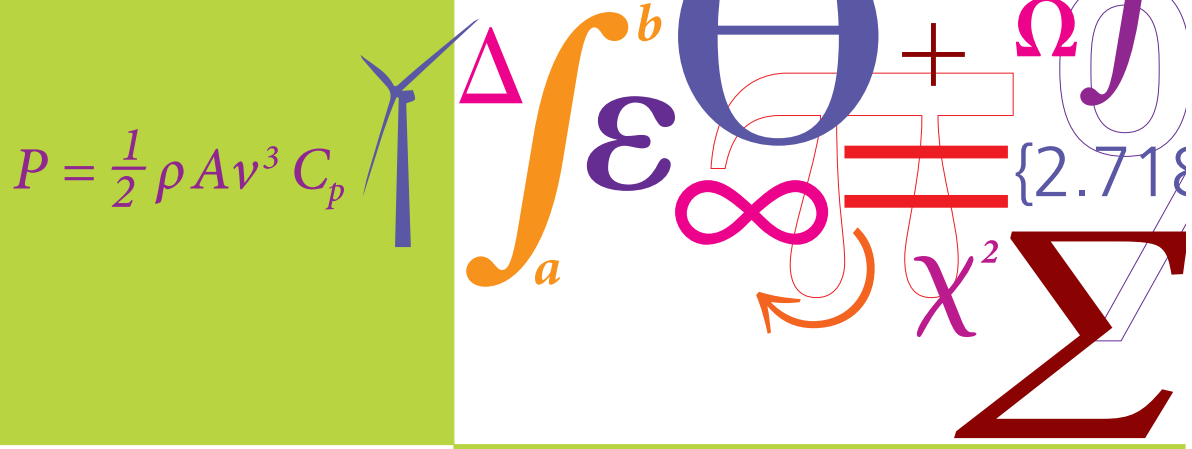
General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Mathematical programming methods for large-scale topology optimization problems

PhD Thesis



Susana Rojas Labanda
DTU Wind Energy
August 2015

DTU Wind Energy
Department of Wind Energy

Mathematical programming methods for large-scale topology optimization problems

Susana Rojas Labanda

PhD Thesis

Department of Wind Energy, DTU
August 2015

DTU Wind Energy
Department of Wind Energy

Author: Susana Rojas Labanda
Title: Mathematical programming methods for large-scale topology optimization problems
Division: Department of Wind Energy

The thesis is submitted to the Danish Technical University in partial fulfillment of the requirements for the PhD degree. The PhD project was carried out in the years 2012-2015 at the Wind Turbine Structures section of the Department of Wind Energy. The dissertation was submitted in August 2015.

August 2015

Project Period:
2012-2015

Degree:
PhD

Supervisor:
Mathias Stolpe
Ole Sigmund

Sponsorship: Villum Foundation through the research project "Topology Optimization, the Next Generation (NextTop)".

Technical University of Denmark,
Department of Wind Energy
Frederiksborgvej 399
Building 118
4000 Roskilde
Denmark
Telephone: (+45)20683230
Email: srla@dtu.dk
www.vindenergi.dtu.dk

"Life is not about waiting for the storm to pass.
It's about learning how to dance in the rain."

Vivian Greene.

Acknowledgements

During the PhD I have had the great privilege of working and sharing these three years with a lot of wonderful people.

I would firstly like to thank my supervisor, Mathias Stolpe, for his extraordinary help and support, and for the initiation into this amazing field. I will always be grateful for the amount of time he has dedicated me, for his guidance and advise. Thanks for pushing and demanding but at the same time for being so positive and close.

I would like to acknowledge Professor Michael Saunders from Stanford University for his help and advice during my external stay. I would also want to thank all the people I met in San Francisco, special thanks to Rikel and the firehouse.

Thanks to my office colleagues and Risø friends, to make these three years very easy-going, for all the moments in the Friday bar, for the moral support, the coffee breaks and for the good times in the office. Special thanks to Juan.

I would also like to thank for all the support to my "Danish family". Thanks Britta, Felix, Jorge and Cristina for all the great moments, all the conversations and for the every day life. It was very comforting to have you there.

Special mention must go to my family and friends back home. I am really grateful to have them in my lives. Without them this will never be done, and they are the reason why I am here right now. Last but not least, many many thanks to Jaime. Thanks for being always there.

Gracias a todos y a todas por hacer que la distancia no sea un obstáculo. Por todas las risas, la cercanía, las visitas, las escapadas y los skypes. Os quiero!

Especial gracias a mis padres por la ejemplar educación que me habéis dado. Porque sin vosotros no estaría hoy aquí. Gracias por la confianza, la exigencia y el amor que siempre me dais. Gracias a mis hermanas, Cristina y Elena, por ser a la vez tan distintas y tan iguales a mí. Por enseñarme algo cada día y por ser ejemplo. Y por último millones de gracias a tí, Jaime. Por ser mi compañero de viaje, por ser bastón, por hacerlo todo más fácil, por tu optimismo, tu infinita paciencia y por tu cariño. Gracias por ser mi vía de escape, por hacerme reír y por enseñarme a relativizar.

Abstract

This thesis investigates new optimization methods for structural topology optimization problems. The aim of topology optimization is finding the optimal design of a structure. The physical problem is modelled as a nonlinear optimization problem. This powerful tool was initially developed for mechanical problems, but has rapidly extended to many other disciplines, such as fluid dynamics and biomechanical problems. However, the novelty and improvements of optimization methods has been very limited. It is, indeed, necessary to develop of new optimization methods to improve the final designs, and at the same time, reduce the number of function evaluations. Nonlinear optimization methods, such as sequential quadratic programming and interior point solvers, have almost not been embraced by the topology optimization community. Thus, this work is focused on the introduction of this kind of second-order solvers to drive the field forward.

The first part of the thesis introduces, for the first time, an extensive benchmarking study of different optimization methods in structural topology optimization. This comparison uses a large test set of instance problems and three different structural topology optimization problems.

The thesis additionally investigates, based on the continuation approach, an alternative formulation of the problem to reduce the chances of ending in local minima, and at the same time, decrease the number of iterations.

The last part is focused on special purpose methods for the classical minimum compliance problem. Two of the state-of-the-art optimization algorithms are investigated and implemented for this structural topology optimization problem. A Sequential Quadratic Programming (TopSQP) and an interior point method (TopIP) are developed exploiting the specific mathematical structure of the problem. In both solvers, information of the exact Hessian is considered. A robust iterative method is implemented to efficiently solve large-scale linear systems. Both TopSQP and TopIP have successful results in terms of convergence, number of iterations, and objective function values. Thanks to the use of the iterative method implemented, TopIP is able to solve large-scale problems with more than three millions degrees of freedom.

Resumé (In Danish)

Denne afhandling undersøger nye optimeringsmetoder for strukturelle topologiske optimeringsproblemer. Målet med topologisk optimering er at finde det optimale design af en struktur. Det fysiske problem er modelleret som et ikke-lineært optimeringsproblem. Dette stærke værktøj var oprindeligt udviklet til mekaniske problemer, men har siden udviklet sig hastigt til andre discipliner såsom strømningsmekanik (fluid dynamics) og biomekaniske problemer. Ikke desto mindre har nytænkningen og forbedringerne af optimieringsmetoderne været meget begrænset. Det er i den grad nødvendigt at udvikle nye optimeringsmetoder til at forbedre det endelige design og på samme tid reducere antallet af funktionsevalueringer. Ikke-lineære optimeringsmetoder, såsom sekvensiel kvadratisk programming og indre punkts metoder, har næsten ikke fået opmærksomhed af det topologiske optimeringsfaglige fællesskab. Derfor fokuserer dette arbejde på at introducere disse anden-ordens løsningsmetoder for at drive feltet fremad.

Den første del af afhandlingen introducerer, for første gang, et omfattende benchmark studie af forskellige optimeringsmetoder indenfor strukturel topologisk optimering. Denne sammenligning anvender et stort testsæt og tre forskellige strukturelle optimeringsproblemer.

Afhandlingen undersøger desuden, baseret på kontinuerte tilgange, en alternativ formulering af problemet for at reducere risikoen for at ende i et lokalt minimum, og samtidig mindske antallet af iterationer.

Den sidste del fokuserer på skrædersyede metoder til det klassiske minimum compliance problem. To af de mest velansete optimeringsalgoritmer er undersøgt og implementeret for dette struturalle optimeringsproblem. En sekvensiel kvadratisk programmerings (TopSQP) og en indre punks metode (TopIP) er udviklet til at udnytte problemets specielle matematiske struktur. I begge løserer bruger vi eksakt Hessian information. En robust iterativ metode er implementeret til effektivt at løse lineære systemer i stor skala. Både TopSQP og TopIP opnår successfulde resultater, både hvad angår konvergens, antallet af iterationer og objektivværdien. Takket været den implementerede iterative metode, kan TopIP løse problemer i stor skala med mere end tre millioner frihedsgrader.

Preface

This thesis was submitted in partial fulfillment of the requirements for obtaining the PhD degree at the Technical University of Denmark. The PhD project was carried out from September 2012 till August 2015 at the Wind Turbine Structures section of the Department of Wind Energy. The project has been supervised by Professor Mathias Stolpe and Professor Ole Sigmund. The PhD project was funded by the Villum Foundation through the research project Topology Optimization - the Next Generation (NextTop).

The dissertation is organized as a collection of papers. The first part of the thesis gives an overview of the background needed for the investigation. The second part contains a collection of four articles.

During the PhD studies, part of the work was presented at the 11th World Congress on Computational Mechanics (WCCM XI), Barcelona, July, 2014, the 11th World Congress on Structural and Multidisciplinary Optimization (WCSMO-11), Sydney, June 2015, and the DCAMM Internal Symposium in March 2013 and in 2015. From October to December 2014, I visited Professor Michael Saunders at Stanford University (California, USA) as part of the external research stay.

A list of attended conferences and publications is collected below.

List of publications

- Rojas-Labanda, S. and Stolpe, M.: Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization* (2015). Published online. DOI : 10.1007/s00158-015-1250-z.

- Rojas-Labanda, S. and Stolpe, M.: Automatic penalty continuation in structural topology optimization. *Structural and Multidisciplinary Optimization* (2015). Published online. DOI : 10.1007/s00158-015-1277-1.

- Rojas-Labanda, S. and Stolpe, M.: An efficient second-order SQP method for structural topology optimization. *Structural and Multidisciplinary Optimization* (2015). In review.

- Rojas-Labanda, S. and Stolpe, M.: Solving large-scale structural topology optimization problems using a second-order interior point method. To be submitted.

Presentation and conferences

- 14th DCAMM Symposium, March 2013, Nyborg, Denmark. Rojas-Labanda S. and Stolpe, M.: Mathematical programming methods for large-scale topology optimization problems. Poster presentation.
- 11th World Congress on Computational Mechanics (WCCM XI), July 2014, Barcelona, Spain. Rojas-Labanda S. and Stolpe, M.: Benchmarking optimization methods for structural topology optimization problems. Oral presentation.
- Linear Algebra and Optimization Seminar, ICME Stanford University, October 2014, California, USA. Rojas-Labanda S. and Stolpe, M.: Mathematical programming methods for large-scale structural topology optimization. Oral presentation.
- 15th DCAMM Symposium, March 2015, Horsens, Denmark. Rojas-Labanda S. and Stolpe, M.: The use of second-order information in structural topology optimization. Oral presentation.
- 11th World Congress of Structural and Multidisciplinary Optimization (WCSMO 11), June 2015, Sydney, Australia. Rojas-Labanda S. and Stolpe, M.: An efficient second-order SQP method for structural topology optimization. Oral presentation.

Roskilde, August 2015

Susana Rojas Labanda

List of Acronyms

AMG	Algebraic Multigrid
BFGS	Broyden Fletcher Godfarb Shanno
CG	Conjugate Gradient
CQ	Constraint Qualification
CONLIN	Convex Linearization
ESO	Evolutionary Structural Optimization
FEM	Finite Element Method
GCMMA	Globally Convergent Method of Moving Asymptotes
GMRES	Generalized Minimal Residual
KKT	Karush Kuhn Tucker
LICQ	Linear Independence Constraint Qualification
MBB	Messerschmitt Bölkow Blohm
MFCQ	Mangasarian Fromowitz Constraint Qualification
MINRES	Minimal Residual
MMA	Method of Moving Asymptotes
OC	Optimality Criteria
PCG	Preconditioner Conjugate Gradient
PDE	Partial Differential Equation
QP	Quadratic Programming
RAMP	Rational Approximation of Material Properties
SAND	Simultaneous Analysis and Design
SCP	Separable Convex Programming
SIMP	Solid Isotropic Material with Penalization
SQP	Sequential Quadratic Programming

Contents

Part I

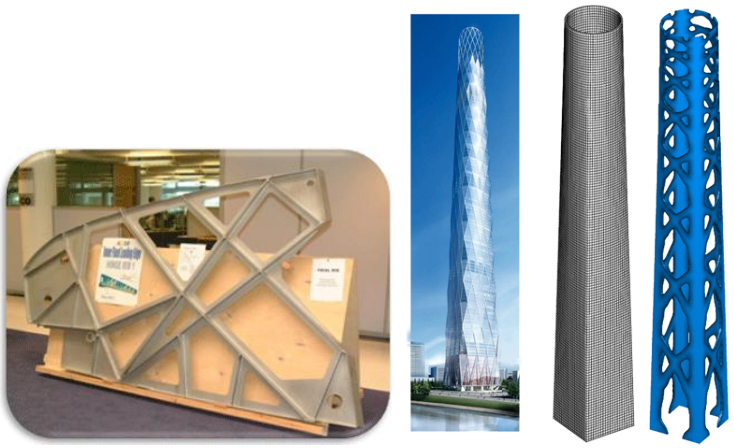
Background

1 Introduction

Structural topology optimization [74] is a relatively mature field that has rapidly expanded due to its interesting theoretical implications in mathematics, mechanics, and computer science. It has, additionally, important practical applications in the manufacturing, automotive, and aerospace industries [34]. This discipline is focused on finding the optimal distribution of material in a prescribed design domain given some boundary conditions and external loads. Topology optimization is commonly used in the conceptual design phase presenting new and innovative structures. Classical structural topology optimization problems are, for instance, maximization of the stiffness (minimize the compliance) or minimization of the total weight (volume) of the structure, subject to some constraints on the total volume, total stiffness, maximum displacements, or stresses [10]. The design domain is often discretized using finite elements, where the variables represent the density of each element.

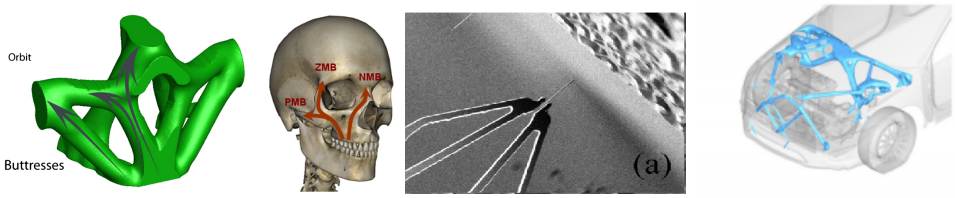
Topology optimization was first initiated in the 1960s, with the introduction of truss topology design in [39]. The continuum approach appeared in the late 1980s [8]. Topology optimization can be regarded as an extension of sizing and shape optimization. The goal of sizing optimization consists of finding the thickness of the structure for a fixed design domain, whereas shape optimization finds the optimal shape of a domain. Topology optimization is now well-establish and can be applied in many different research areas such as fluid dynamics, electromagnetic problems, nuclear physics, and biomechanical problems, among others [10]. However, the discussion in this thesis is limited to structural topology design problems.

Figure 1.1 shows some of the practical applications of structural topology optimization. In the last decade plenty of new applications have emerged in this field [34]. On the other hand, very few improvements and insights are done regarding the optimization techniques. The development of novel mathematical optimization methods to accurately solve large-scale topology optimization problems, is crucial to improve the final designs in these and in many other applications ([103]).



(a) Aerospace applications

(b) Architecture applications



(c) Medical applications

(d) Micromachine applications

(e) Automotive applications

Figure 1.1: Examples of some practical structural topology optimization applications (from [76], [111], [110], [97], and [88], respectively).

First of all, the physical problem needs to be suitably formulated in a mathematical problem. Then, it is discretized and optimized. Figure 1.2 shows the general flow used in this work for obtaining an optimized design using a mathematical programming method. In particular, standard finite element analysis and classical formulations of the topology optimization problem are considered. This thesis concerns with the optimization step rather than the pre-processing, the post-processing, and the structural analysis. New techniques are implemented and developed to improve the performance of the "black-box" that is usually considered the optimization solver.

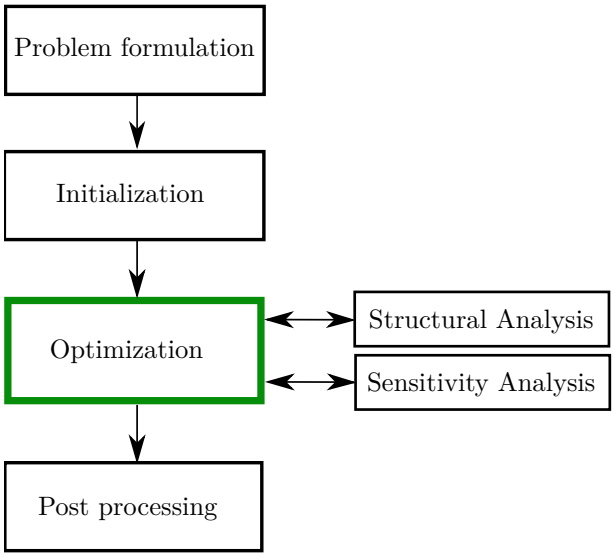


Figure 1.2: Flow chart of topology optimization design using mathematical programming methods. The research presented here is focused on the optimization step.

While a variety of large-scale nonlinear solvers could be applied, structural topology optimization problems are usually solved by sequential convex approximation methods such as the Method of Moving Asymptotes (MMA) [112] and [131], and the Convex Linearization (CONLIN) method [48], or by using Optimality Criteria (OC) methods, see e.g. [93], [130], and [4]. These methods were specially designed for optimal design purposes and are now extensively used in commercial optimal design software as well as academic research codes. However, they are first-order methods with slow convergence rates. In addition, methods such as the original CONLIN and MMA have lack of convergence proof.

Throughout this thesis, different alternatives to the classical structural topology optimization solvers based on second-order information are presented. It is well-known in the mathematical optimization community, that second-order methods converge in fewer iterations and produce more accurate solutions than first-order solvers [38]. Although second-order methods have not been embraced by the topology optimization community, this thesis will show that the introduction of this kind of solvers is necessary to drive the field forward. The use of second-order information is essential to obtain good optimized designs in few iterations.

The problem is frequently defined in its nested form, meaning that the state (nodal displacement) variables are related to the design (density) variables through the equilibrium equations [10]. When second-order solvers are applied to this formulation, the computational effort is dominated by both, the solution of the equilibrium equations and the computation of the Hessian. In such cases, efficient approaches to reduce the computational time and memory usage are needed. Thus, the implementation of the methods

presented in this thesis exploits the specific mathematical structure of the problem.

Ultimately, new techniques for large-scale problems will enable the solution of real structural topology optimization problems. More specifically, the use of iterative methods for solving large-scale linear systems such as multigrid techniques and Krylov sub-space methods is necessary to apply general nonlinear solvers to this type of problems [103].

This thesis concerns with the comparison, research, and implementation of numerical optimization methods for structural topology optimization problems, such as interior point methods [52], and Sequential Quadratic Programming (SQP) methods [15]. The discussion is almost restricted to the minimum compliance problem. Further investigations regarding different topology optimization problems or finite element analysis are out of the scope in this work.

This thesis consists of four separate journal papers related to numerical optimization programming in topology optimization problems. Part I regards with the general background of topology optimization problems (Chapter 2), numerical optimization (Chapter 3), and linear algebra methods (Chapter 4). This covers the essentials needed to fully understand the rest of the thesis. Chapter 5 includes a brief summary of the different papers collected in the thesis, with the main results and contributions. In the second part of the thesis the four research articles are included. Chapter 6 collects "Benchmarking optimization solvers for structural topology optimization", Chapter 7 presents "An automatic penalty continuation in structural topology optimization problems", Chapter 8 gathers "An efficient second-order SQP method for structural topology optimization problems". Finally, Chapter 9 deals with "Solving large-scale structural topology optimization problems using a second-order interior point method".

2

Structural topology optimization

Structural topology optimization obtains an optimal material distribution in a prescribed design domain for some boundary conditions and loads. This optimized design is found by minimizing an objective function subject to some constraints modelling technical specifications. Structural topology optimization has become a multidisciplinary field of research and has been very active since 1988 with the publication of [7] and [8]. It is also considered a very powerful tool for industrial applications, such as the construction of aircrafts and automobiles [34]. These applications require, for instance, a light structure but at the same time as stiff as possible. In topology optimization, there are no assumptions in the final design of the structure, and ideally, the goal is to decide whether to put material among all the points in the design domain. The choice of the topology of the structure in this conceptual phase gives innovative and novel designs. In general, the structure is discretized using finite element method (FEM) for design parametrization and structural analysis.

The main purpose of this chapter is to give a brief introduction and a literature review of structural topology optimization problems. Firstly, the topology optimization problem formulation considered throughout the thesis is introduced. Then, an overview of some optimization methods commonly used in this field is presented. Finally, the standard set of benchmark problems is discussed.

2.1 Problem formulation

The classical structural topology optimization problem consists of maximizing the stiffness of the structure (or minimizing the compliance) with a constraint on the total volume or weight [10]. This problem, without any type of regularization, is well-known to be ill-posed, see e.g. [18], [42], and [9]. This means, the problem lacks existence of solutions in the original design domain. To simplify the notation, the variational problem formulation is presented without regularization techniques. Thus, the problem as stated in this section is ill-posed. For the complete and correct description of the problem, see e.g. [18]. The variational problem is defined as in [10], [107], and [18].

Let $\Omega \in \mathbb{R}^{dim}$ describes the bounded domain, with a Lipschitz boundary Γ [33].

For notation purposes, the Sobolev space $W^{k,p}$ for $k=1$ and $p=2$ is defined as

$$H^1(\Omega) = \{f \in L^2(\Omega), \text{ s.t. } \nabla f \in L^2(\Omega)\},$$

with

$$L^p(\Omega) = \{f : \Omega \to \mathbb{R}, \text{ s.t. } ||f||_{L^p(\Omega)} < \infty\},$$

$$||f||_{L^p(\Omega)} = \begin{cases} \left(\int_\Omega |f(x)|^p dx\right)^{1/p} & 1 \leq p < \infty, \\ \inf\{\alpha : |f(x)| \leq \alpha \text{ for almost every point } x \in \Omega\} & p = \infty. \end{cases}$$

The standard notation $H^1(\Omega)^{dim}$ will be used for the Sobolev space of function $f : \Omega \to \mathbb{R}^{dim}$. More details on Sobolev spaces can be found in [33].

In the following formulation, the term $\mathcal{U}$ refers to the space of kinematically admissible displacements, and $\mathcal{H}$ to the set of feasible designs [18], i.e.,

$$\mathcal{U} = \{\mathbf{u} \in H^1(\Omega)^{dim}, \text{ such that } \mathbf{u} = \mathbf{0} \text{ on } \Gamma_u\},$$

$$\mathcal{H} = \{t \in L^\infty(\Omega) \cap L^1(\Omega) : \quad 0 < \underline{t} \leq t \leq 1 \text{ on } \Omega, \text{ and } \int_\Omega t d\Omega \leq V\}. \tag{2.1}$$

For any given $\underline{t} > 0$ and maximum volume $V > 0$. For convenience, the boundary is partitioned, $\Gamma = \Gamma_u \cup \Gamma_t$ such that $\Gamma_u \cap \Gamma_t = \emptyset$. Γ_t refers to the part of the boundary with non-fixed displacements, i.e., where the tractions are assigned [10].

The minimum compliance problem is stated as

$$\begin{aligned} \underset{\mathbf{u}\in\mathcal{U}, t\in\mathcal{H}}{\text{minimize}} \quad & l(\mathbf{u}) \\ \text{subject to} \quad & a(\mathbf{u},\mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{U}. \end{aligned}$$

The problem minimizes the load linear form [10], described as

$$l(\mathbf{u}) = \int_\Omega \mathbf{f}_b \cdot \mathbf{u} d\Omega + \int_{\Gamma_t} \mathbf{f}_t \cdot \mathbf{u} ds,$$

Here, the variable $\mathbf{f}_b \in L^2(\Omega)^{dim}$ represents the body forces and $\mathbf{f}_t \in L^2(\Gamma_t)^{dim}$ the boundary tractions. For a fixed and admissible design $t \in \mathcal{H}$, the displacement $\mathbf{u} \in \mathcal{U}$ satisfies the state equation in its variational form [107],

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \mathbf{f}_b \cdot \mathbf{v} d\Omega + \int_{\Gamma_t} \mathbf{f}_t \cdot \mathbf{v} ds \quad \forall \mathbf{v} \in \mathcal{U},$$

with the energy bilinear form defined as

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega t\ \mathbf{E}\ \varepsilon(\mathbf{u}) \cdot \varepsilon(\mathbf{v}) d\Omega.$$

The term $\varepsilon(\mathbf{u})$ refers to the strain tensor and $\mathbf{E}$ is the elasticity tensor. Existence of solutions to the state equations are ensured, see for instance [18]. Assuming, small displacements, the linearized strain is

$$\varepsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right).$$

For computational purposes, the space $\mathcal{U}$ (and $\mathcal{H}$) is usually discretized. Let $\mathbf{V}_h \subset \mathcal{U}$ ($\mathbf{Q}_h \subset \mathcal{H}$) be any finite dimensional sub-space. Here, h refers to the discretization parameter. Then, the finite dimensional problem finds the approximated solution $\mathbf{u}^h \in \mathbf{V}_h$ (and $\mathbf{t}^h \in \mathbf{Q}_h$) such that

$$a(\mathbf{u}^h, \mathbf{v}^h) = l(\mathbf{u}^h) \quad \forall \mathbf{v}^h \in \mathbf{V}_h.$$

In particular, the domain Ω is discretized using finite elements. In the following, the finite element method is presented for a 2D domain (assuming constant thickness and plane stress condition). More details of the finite element method (FEM) can be found in [31] and [30].

The strain-displacement relationship on the element e is

$$\boldsymbol{\varepsilon}_e = \partial \mathbf{u}_e^h.$$

In order to keep only the necessary information, the strain tensor is transformed in a vector with independent components ([31]). Here, the partial derivatives of the coordinates are gathered in ∂

$$\partial = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}.$$

The displacement on a finite element can be written as

$$\mathbf{u}_e^h = \mathbf{N}_e \mathbf{d},$$

where $\mathbf{d}$ is the vector with all nodal displacements (global level) and $\mathbf{N}_e$ is the shape functions (interpolation functions).

The strain tensor can be redefined as

$$\boldsymbol{\varepsilon}_e = \partial \mathbf{u}_e^h = \partial \mathbf{N}_e \mathbf{d} = \mathbf{B}_e \mathbf{d},$$

with $\mathbf{B}_e$, the strain-displacement matrix (element level). The principle of virtual work is applied to obtain the element stiffness matrix and forces expressions, similar to [31]. Notice that the state equations can be written as the summation of integrals over the elements. Let $\mathbf{v} = \mathbf{N}_e \hat{\mathbf{d}}$ define a virtual displacement (small perturbation of the displacement), then the discretized bilinear energy form becomes

$$\sum_e \hat{\mathbf{d}}^T \int_{\Omega_e} t_e^h \mathbf{B}_e^T \mathbf{E} \mathbf{B}_e d\Omega \mathbf{d} = \sum_e \hat{\mathbf{d}}^T \left(\int_{\Omega_e} \mathbf{N}_e^T \mathbf{f}_b^h d\Omega + \int_{(\Gamma_t)_e} \mathbf{N}_e^T \mathbf{f}_t^h ds \right). \tag{2.2}$$

Here, t_e^h represents the material of the element e. The discretized problem is defined on an element level. The vector of nodal loads applied to the element e is

$$\mathbf{f}_e = \int_{\Omega_e} \mathbf{N}_e^T \mathbf{f}_b^h d\Omega + \int_{(\Gamma_t)_e} \mathbf{N}_e^T \mathbf{f}_t^h ds,$$

and

$$\mathbf{K}_e = \int_{\Omega_e} t_e \mathbf{B}_e^T \mathbf{E} \mathbf{B}_e d\Omega, \tag{2.3}$$

the element stiffness matrix. The isotropic stiffness tensor $\mathbf{E}$ is defined based on the Hooke's law as follows [31]

$$\mathbf{E} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix},$$

with ν the Poisson's ratio of the material, and E the Young's modulus constant. From now on, the global density, displacement (state variable), and force vectors are defined with the terms $\mathbf{t} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^d$ (instead of $\mathbf{d}$) and $\mathbf{f} \in \mathbb{R}^d$, respectively, with n the number of elements and d the total number of degrees of freedom. Then, the global equilibrium equations are obtained by the assembly of all elements,

$$\mathbf{K}\mathbf{u} = \mathbf{f},$$

with

$$\mathbf{K} = \sum_{e=1}^{n} \mathbf{K}_e.$$

The summation symbol refers to the expansion to the element terms to the global vector (or matrix), i.e. $\mathbf{K} : \mathbb{R}^n \to \in \mathbb{R}^{d \times d}$.For more information about the finite element method and its expressions, see [30] and [31].

Throughout the rest of the thesis the problem is always formulated in its discretized version. Isotropic materials and a regular grid with a constant density throughout each

element are considered. In particular, bilinear quadrilateral elements are considered. In this scenario, the element stiffness matrix is assumed to be the same for all elements. Additionally, the integral (2.3) is evaluated using 2×2 Gauss-point quadrature. Finally, only design-independent loads are considered. Along the thesis, the isotropic stiffness tensor is defined based on the density of the element, i.e. $E(t_e)$ (see Section 2.2). Thus, the equilibrium equations are defined as in [10],

$$\begin{aligned} &\mathbf{K}(\mathbf{t})\mathbf{u} = \mathbf{f} \\ &\mathbf{K}(\mathbf{t}) = \sum_{e=1}^{n} E(t_e)\mathbf{K}_e. \end{aligned}$$

Finally, throughout the rest of the thesis, the stiffness matrix is assumed to be positive definite to avoid singularity. In other words, $E(t_i) > 0$ (see Section 2.2).

The discretized minimum compliance problem is formulated as

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n, \mathbf{u}\in\mathbb{R}^d}{\text{minimize}} \quad & \mathbf{f}^T\mathbf{u} \\ \text{subject to} \quad & \mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{aligned} \tag{P_S^c}$$

The volume constraint is defined as a linear inequality with $\mathbf{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$ the relative volume of each element and $0 < V < 1$ the maximum volume fraction allowed. The discretized equilibrium equations are described as explicit constraints.

In its original formulation (2.1), the density variable specifies the design of the structure, taking either $t_i = 1$, if the ith element contains material (solid), or $t_i = 0$, if the element remains void. However, in practice, the integer variables are replaced by continuous variables. The solid and void topology optimization problem is modified, so that the density variables can take any value between 0 (void) and 1 (solid) (see Section 2.2).

The minimum compliance formulation described in (P_S^c) is the so-called Simultaneous Analysis and Design (SAND) [5], since both, design and state variables, are simultaneously optimized. The main advantage of this formulation is the ease of the objective function, which is linear. On the other hand, optimization algorithms need to deal with a large number of nonlinear equality constraints and infeasible iterations.

The problem is frequently defined in a nested form, where the equilibrium equations are implicit in the objective function. Therefore, only the design variables are optimized, while the state variables are computed by solving the equilibrium equations at each objective function evaluation. The nested formulation is described as in (P_N^c).

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \qquad (\text{or } \mathbf{f}^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{f}) \\ \text{subject to} \quad & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{aligned} \tag{P_N^c}$$

with $\mathbf{u}(\mathbf{t}) = \mathbf{K}^{-1}(\mathbf{t})\mathbf{f}$. Although the objective function of the nested form is highly nonlinear (and typically nonconvex), it has the advantage of containing only linear constraints. Thus, the development and implementation of nonlinear optimization methods for this specific formulation (see Chapters 8 and 9) are focused on dealing with the nonconvexity and the computational effort of the objective function, rather than incorporating different techniques to cope with the infeasibility and unboundedness of the problem.

Two more topology optimization problems are considered in Chapters 6 and 7, namely the minimum volume and the compliant mechanism design problems (see [22] and [101]). For the former, a constraint controlling the value of the compliance of the structure is required. Problem (P_S^w) describes the minimum volume problem in the nested form.

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{minimize}} & \mathbf{v}^T\mathbf{t} \\ \text{subject to} & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \leq C \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{array} \tag{P_N^w}$$

The SAND formulation is stated as

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n,\mathbf{u}\in\mathbb{R}^d}{\text{minimize}} & \mathbf{v}^T\mathbf{t} \\ \text{subject to} & \mathbf{K}(\mathbf{t})\ \mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{f}^T\mathbf{u} \leq C \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{array} \tag{P_S^w}$$

Here $C > 0$ is a given upper bound of the compliance.

In the compliant mechanism design problem, the displacement at a given point (output spring) is maximized with a constraint on the volume. The domain contains an input force $\mathbf{f}_{\text{in}}$ and an input and output spring stiffness (k_{in}, k_{out}). The objective function is defined by the use of a unit length vector ($\mathbf{l}$) with zeros in all the degrees of freedom except at the output [10]. Assuming a linear model for the equilibrium equations, the nested and SAND formulations of this problem are described in (P_N^m) and (P_S^m), respectively.

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{maximize}} & \mathbf{l}^T\mathbf{u}(\mathbf{t}) \\ \text{subject to} & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{array} \tag{P_N^m}$$

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n,\mathbf{u}\in\mathbb{R}^d}{\text{maximize}} & \mathbf{l}^T\mathbf{u} \\ \text{subject to} & \mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{array} \tag{P_S^m}$$

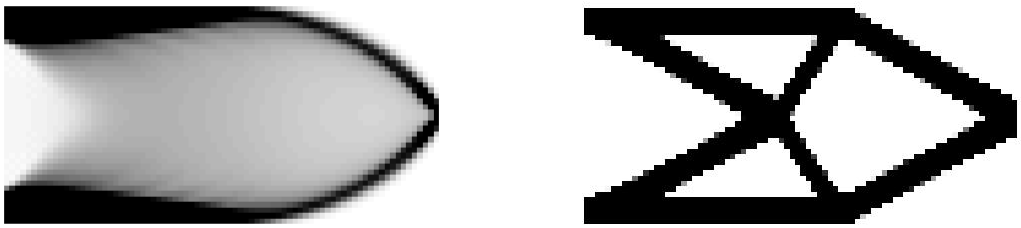
2.2 Density penalization and regularization techniques

The topology optimization problem is formulated such that the density of material $\mathbf{t}$ varies continuously between 0 (void) and 1 (solid) [103]. However, it is desirable to obtain

designs close to a 0-1 solution. A material interpolation scheme is included to penalize intermediate density values. These artificial densities are typically called *grey regions* [9]. Two of the most popular approaches to penalize these densities are the Solid Isotropic Material with Penalization (SIMP) [7] and [130], and the Rational Approximation of Material Properties (RAMP) [109]. These approaches use interpolation schemes to force the density values go to the bounds. The Young's modulus E is defined as

$$E(t_i) = \begin{cases} E_v + (E_1 - E_v)t_i^p & \text{SIMP scheme} \\ \\ E_v + (E_1 - E_v)\frac{t_i}{1+q(1-t_i)} & \text{RAMP scheme.} \end{cases} \tag{2.4}$$

Here, $E_v > 0$ and $E_1 \ggg E_v$ are the Young's modulus for the "void" and solid material respectively. In practice, the material penalization parameter is generally set to $p = 3$ and $q = 6$, for the SIMP and RAMP approaches, respectively [71]. For values of $p = 1$ ($q = 0$), the structural topology optimization problems described in Section 2.1 are convex, but, in general, the problem becomes nonconvex for values of $p > 1$ ($q > 0$). In particular, the four articles included in this thesis consider the penalization parameter greater than 1. Chapters 6, 8, and 9 use the SIMP approach with $p = 3$, while Chapter 7 uses both the SIMP (with $p = 3$) and the RAMP (with $q = 6$) approaches.



(a) No penalization parameter, $p = 1$.

(b) Penalization parameter $p = 3$.

Figure 2.1: Example of a cantilever beam design using different penalization parameter values. This figure illustrates how the material interpolation approaches affect the optimized design. A density filter technique was used to obtain the design presented in Figure 2.1b.

Figure 2.1 shows an illustrative example where the design of a cantilever beam is optimized with no penalization scheme, i.e. $p = 1$ (Figure 2.1a), and with the SIMP approach using $p = 3$ (Figure 2.1b). In the latter, the grey regions (intermediate designs) disappear. Throughout the rest of the thesis, terms such as *almost solid-and-void* and

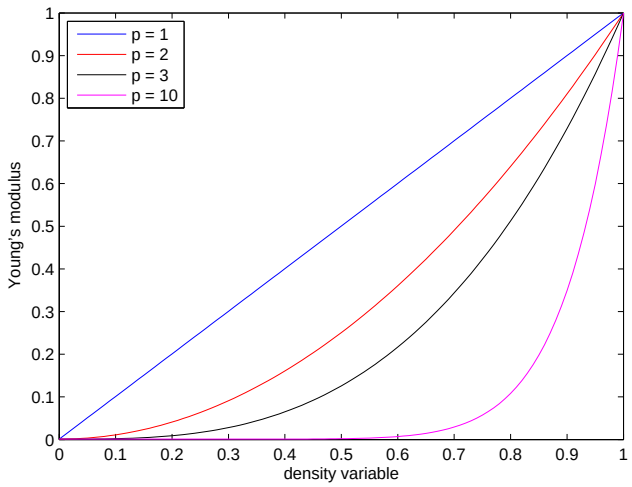


Figure 2.2: Example of the behaviour of the Young's modulus at different material penalization parameter values using the SIMP approach.

black-and-white designs are used to refer to this situation.

Additionally, Figure 2.2 shows the behaviour of the Young's modulus (2.4) using the SIMP approach with different penalization parameter values. When $p \ggg 1$, the intermediate densities are highly penalized. However, the value $p = 3$ is normally sufficient to produce almost solid-and-void designs [103].

Three numerical instabilities are detected for density-based topology optimization problems, namely, checker-boards, mesh-dependencies, and local minima [104].

It is well-known that the density-based topology optimization problem is ill-posed in the continuum setting, see [10] and [42]. In other words, the solution of the same problem at different discretization meshes is different. This non existence of solutions is commonly displayed as mesh-dependency, see Figure 2.3. Different regularization techniques emerge in the literature to prevent this issue, such as the perimeter control [2] and [63], gradient restrictions [90], and filtering techniques [18].

For the density-based topology optimization problems, the distribution of material might form a checker-board pattern. This refers to the formation of regions where solid material and "void" material are alternating forming a checker-board pattern [36]. For an illustrative example, see Figure 2.4. The cause of this anomaly comes from the finite element approximations, in which the modelling of the stiffness is overestimated [10].

In order to overcome this instability, either higher order finite elements or filtering techniques [18] can be used. Filtering strategies are more common since they solve both the mesh-dependency and the checker-boards. In contrast, if higher order finite elements are used, other regularization techniques are needed, such as perimeter control or gradient restriction techniques. Filtering methods include explicit limitations on the distribution

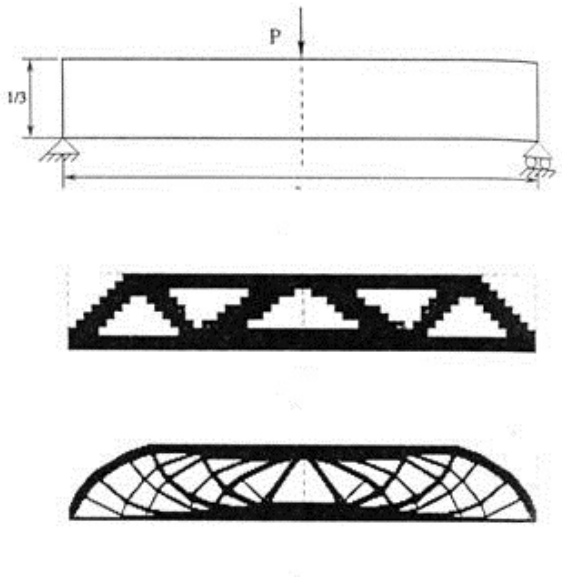


Figure 2.3: Example of mesh-dependency in a MBB beam. Different optimized designs result for different discretizations. Figure taken from [104].

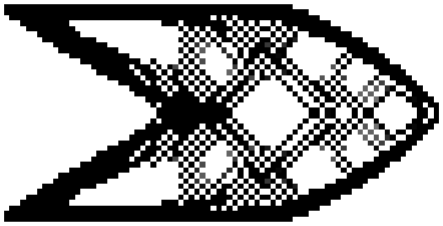


Figure 2.4: Example of a checker-board pattern in a cantilever beam.

of the density. The sensitivity filter [10], the density filter [18], and the PDE filter [77] are, nowadays, some of the most popular choices, see e.g. [100] and [104].

In particular, only the density filter is considered to ensure regularity and existence of solution [18]. It is implemented based on [4]. For a given element e, its filtered density variable $\tilde{t}_e$ depends on a weighted average over the neighbours in a radius $r_{\min}$.

$$\tilde{t}_e = \frac{1}{\sum_{i \in N_e} \overline{H}_{ei}} \sum_{i \in N_e} \overline{H}_{ei} t_i$$
$$\overline{H}_{ei} = \max(0, r_{\min} - \Delta(e,i))$$

Here, N_e the set of elements for which the distance to element i (defined by $\Delta(e,i)$) is

smaller than the filter radius $r_{\min}$.

Finally, the third numerical and theoretical challenge is the so-called local minima [104]. Since topology optimization problems are generally defined as nonlinear and non-convex problems, a global solution cannot be guaranteed. Different optimization methods can produce different local solutions for the same problem (and the same discretization). To avoid this issue, [104] suggests the use of continuation techniques. In these methods, either the radius of the filter or the material penalization parameter is gradually decreasing or increasing (respectively) to reduce the chances of ending in local minima. Thus, the continuation approach solves a sequence of optimization problems. A specific continuation in the penalization parameter strategy is implemented in Chapter 7. In this article, the performance of this method is compared with the classical formulation. The results suggest that, indeed, continuation methods help to avoid local minima. Moreover, the article proposes a new alternative to overcome this numerical issue.

In this section, the classical density-based topology optimization formulation has been introduced. Yet, other alternative formulations are emerging in this field and becoming very popular. For instance, in the review [103], topology optimization methods are principally classified in *(i)* density-based methods ([130] and [7]), *(ii)* evolutionary approaches (such as Evolutionary Structural Optimization (ESO) [127] and [128]), *(iii)* level set methods ([37] and [121]), *(iv)* phase field methods ([19] and [24]), and *(v)* topological derivatives [106].

2.3 Topology optimization methods

Since topology optimization problems can be described as a 0-1 discrete problem, it is indeed natural to solve them using discrete optimization methods [108]. Even so, the solution of the discrete problem is very difficult to obtain, and large-scale problems are nowadays impossible to solve [103]. Heuristic approaches, such as Genetic Algorithms, can estimate an optimized design of the problem without any information of the gradients, see for instance [122] and [6]. However, the computational effort of these non-gradient methods is extremely large for large-scale problems and they are not practical for real topology optimization problems [92] and [103]. Therefore, this thesis is focused on gradient-based mathematical programming methods.

The Optimality Criteria (OC) method [93], [130], and [4], and the Method of Moving Asymptotes (MMA) [112], [113], and [131], are two of the most classical first-order optimization solvers in structural topology optimization, see e.g. [8], [114], [3], and [102] among others. With the sake of completeness, their principal ideas are briefly introduced.

Ultimately, several studies investigate second-order solvers for this type of problems. Newton's type and SQP methods are implemented for structural topology optimization problems in the SAND formulation in, for instance, [70], [69], [68], and [40].

Optimality Criteria

The origins of the Optimality Criteria method go back to the 1960-70s [91] and [10]. The OC method updates the design variables of each point based on an estimation of the optimality conditions. The method updates the designs independently, adding material in those elements in which the estimation of the strain energy is high. For more details of this method, see e.g. the text book [10], where the OC method is explained for the minimum compliance and mechanism design problems.

The OC method used in Chapter 6 is based on 88-lines code implemented in MATLAB [4]. Note that this optimization method, in contrast to the rest of the solvers developed and used in this thesis, does not estimate the Lagrangian multipliers, and therefore, there is no knowledge of the Karush-Kuhn-Tucker (KKT) conditions (see Chapter 3). The KKT conditions are typically used to determine the convergence of the solvers. Thus, the stopping criterion of this method depends only on the difference between two consecutive iterate points.

The Method of Moving Asymptotes

The development of the first-order Convex Linearization (CONLIN) method [48] was the basis for the Method of Moving Asymptotes. MMA was originally developed in 1987 [112], and was specifically implemented for structural optimization problems. It is still one of the most popular solvers in the structural optimization community. MMA approximates the objective and the constraint functions with convex and separable functions. These local approximations appear from the Taylor expansion in the reciprocate and shifted variables, and they only require one objective and gradient function evaluation per iteration [107].

$$\tilde{f}(\mathbf{x})^k = r^k + \sum_{i=1}^{n} \left(\frac{p_i^k}{U_i^k - x_i} + \frac{q_i^k}{x_i - L_i^k} \right),$$

with

$$r^k = f(\mathbf{x}^k) - \sum_{i=1}^{n} \left(\frac{p_i^k}{U_i^k - x_i^k} + \frac{q_i^k}{x_i^k - L_i^k} \right),$$

$$p_i^k = \begin{cases} \left(U_i^k - x_i^k\right)^2 \frac{\partial f}{\partial x_i}(\mathbf{x}^k) & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}^k) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$q_i^k = \begin{cases} 0 & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}^k) \geq 0, \\ -\left(x_i^k - L_i^k\right)^2 \frac{\partial f}{\partial x_i}(\mathbf{x}^k) & \text{otherwise.} \end{cases}$$

The variable U_i and L_i are the asymptotes of the convex approximations. The values of these variables depend on the previous iterations. They move apart if the iterates are going in the same direction (the solver is making progress), and move closer (more

conservative approximations) if the iterates display oscillatory behaviour. The updating scheme of these variables is as follows

$$L_i^k - x_i^k = \gamma_i^k (L_i^{k-1} - x_i^{k-1}),$$
$$U_i^k - x_i^k = \gamma_i^k (U_i^{k-1} - x_i^{k-1}).$$

Here,

$$\gamma_i^k = \begin{cases} 1.2 & \text{if } (x_i^k - x_i^{k-1})(x_i^{k-1} - x_i^{k-2}) > 0, \\ 0.7 & \text{if } (x_i^k - x_i^{k-1})(x_i^{k-1} - x_i^{k-2}) < 0, \\ 1.0 & \text{if } (x_i^k - x_i^{k-1})(x_i^{k-1} - x_i^{k-2}) = 0. \end{cases}$$

Similar to other mathematical programming methods, MMA solves a sequence of easier sub-problems until the KKT conditions (see Chapter 3) are satisfied. Although a stopping criterion based only on the change between design variables is commonly used in the literature (see for instance [3] and [1]), the stopping criterion of MMA in this thesis is based on the first-order optimality conditions.

MMA is an inexpensive method in the sense that the sub-problems are normally easily solvable. However, it does not have globally convergent properties. In contrast, the globally convergent version, GCMMA, introduces conservative approximations of the functions to ensure convergence at the expense of becoming potentially slower [113].

There is plenty of literature concerning extensions of MMA and separable convex programming (SCP) methods, see for instance, [132], [131], [23], [120], and [129]. Additionally, several articles, such as [50], [43], [49], and [13], are focused on MMA-type solvers based on second-order approximations.

Both solvers are compared with other general nonlinear optimization methods in Chapter 6. GCMMA is also used for the comparison of the continuation strategy studied in Chapter 7. Since MMA and GCMMA use the Taylor expansion in the reciprocate variables, these methods cannot be applied in the automatic continuation strategy proposed in Chapter 7[1].

2.4 Benchmark test problems and numerical experiments in structural topology optimization

The numerical experiments in the topology optimization community are generally done using very few examples, see for instance [35], [16], [3], and [63]. When only two or three problems are used to compare different solvers, the results can be misleading. To illustrate this fact, the minimum compliance problem in the nested formulation is considered. Table 2.1 shows the objective function value of three different design domains optimized with GCMMA and IPOPT (interior point software) [117].

[1]This approach requires solvers in which the constraints are linearized (see Chapter 3 and Chapter 7).

Table 2.1: Comparison of the objective function value between GCMMA and IPOPT using three examples. These results are taken from the numerical experiments of Chapter 6.

Solver	**Problem 1:** **Michell** 40×40**,** $V = 0.2$	**Problem 2:** **Michell** 40×20**,** $V = 0.5$	**Problem 3:** **Michell** 40×20**,** $V = 0.1$
GCMMA	$f(\mathbf{t}) = 43.54$	$f(\mathbf{t}) = 73.72$	$f(\mathbf{t}) = 2137$
IPOPT	$f(\mathbf{t}) = 43.74$	$f(\mathbf{t}) = 73.73$	$f(\mathbf{t}) = 1618$

It is clear that if only the results of the first two problems are shown, GCMMA is a better solver choice[2]. In contrast, if the third problem is under consideration, the conclusion will be completely different. Additionally, the difference between the objective function values is very important. In problems 1 and 2, IPOPT and GCMMA obtain very similar designs, while in the third problem, the optimized design of GCMMA has an objective function value noticeably worse than the one obtained with IPOPT. Thus, it is not enough to show which solver obtains the best objective function values, but also how big the difference between the results are. IPOPT might not obtain the best designs, but it might produce good results for larger number of problems. Furthermore, there is no available big test set of problems in which the results can be based on.

The possibility of improving the comparison of formulations and optimization solvers in topology optimization motivates the introduction of performance profiles in this field. For the description of this approach, see [38] and Chapter 6. Performance profiles are nowadays the only acceptable tool used in the numerical optimization community to fairly compare different optimization methods and choices of problem formulations. As a result, a large benchmark library needs to be defined. Some of the well-known and well-establish test problems for benchmarking the minimum compliance problem are cited in [103]. In contrast, there is no standardization for compliant mechanism design problems. Chapter 6 collects for the very first time a large benchmarking test set of problems for minimum compliance, minimum volume, and compliant mechanism design problems. As an illustrative example, Figures 2.5, 2.6 and 2.7 show some of these problems. On the left side of the figures, their boundary conditions and external loads are defined. A possible optimized design is included on the right side. Note that this final design depends on the mesh discretization, length ratios, volume fraction, problem formulation, and optimization solver, among other factors.

[2]Chapter 6 will refute this statement.

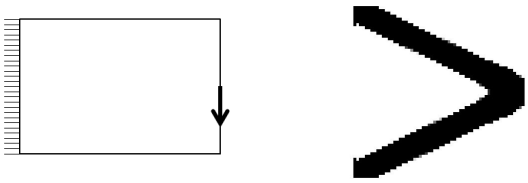


Figure 2.5: Michell 2D domain with an example of a possible optimized design.

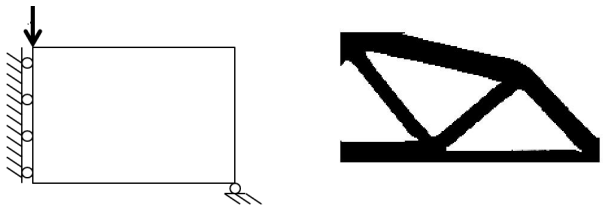


Figure 2.6: MBB 2D domain with an example of a possible optimized design.

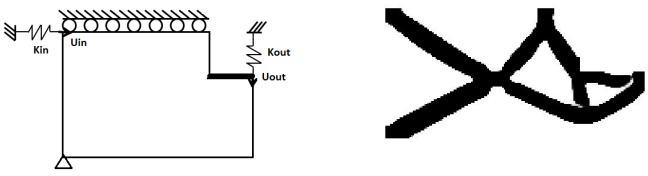


Figure 2.7: Compliant gripper 2D domain with an example of a possible optimized design.

3

Numerical Optimization

The thesis addresses numerical optimization methods for solving structural topology optimization problems efficiently. In particular, two of the state-of-the-art second-order optimization methods are developed and implemented. An efficient Sequential Quadratic Programming (TopSQP) method is implemented in Chapter 8. Chapter 9 is focused on an interior point method (TopIP). In the latter, special emphasis is given on investigating efficient linear algebraic solvers for obtaining the search direction, since large-scale problems are considered.

This chapter provides the necessary background regarding mathematical optimization theory and some optimization methods. First, some preliminary definitions and theorems are introduced. Afterwards, general nonlinear gradient-based programming techniques are discussed. More details of general numerical optimization theory can be found in the text books [87], [78], and [57]. A general review of some iterative methods for linear systems is covered in Chapter 4.

3.1 Numerical Optimization

The considered nonlinear optimization problem can be stated as

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) = 0 & i \in \mathcal{E}, \\ & g_i(\mathbf{x}) \leq 0 & i \in \hat{\mathcal{I}}, \\ & l_i \leq x_i \leq u_i & i = 1, \ldots, n, \end{array} \tag{NLP}$$

with $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ (with $i = 1, \ldots, m$) being continuously differentiable functions. The terms l_i and u_i are the lower and the upper bounds of the variable x_i, respectively. The set $\hat{\mathcal{I}}$ contains the indices i such as the constraint $g_i(\mathbf{x})$ is an inequality. The term $\mathcal{E}$ refers to the equality constraints. For notational convenience, both general and bound constraints are gathered in $c_i(\mathbf{x})$, i.e.,

$$c_i(\mathbf{x}) = \begin{cases} (g_j(\mathbf{x}))_{j=1,\ldots,m} & i = 1, \ldots, m \\ (x_j - u_j)_{j=1,\ldots,n} & i = m+1, \ldots, m+n \\ (l_j - x_j)_{j=1,\ldots,n} & i = m+n+1, \ldots, m+2n. \end{cases}$$

For this generalization, the indices i such that the constraint $c_i(\mathbf{x})$ is an inequality are gathered in $\mathcal{I}$ ($\hat{\mathcal{I}} \subset \mathcal{I}$). Thus, $\mathcal{E} \cup \mathcal{I} = \{1, \ldots, m+2n\}$ and $\mathcal{E} \cap \mathcal{I} = \emptyset$.

This section only outlines some theoretical aspects needed. The proofs of the theorems can be found in [87], [20], and [78].

Definition 3.1. *(from [87]) A **feasible set** Ω is the set of all points $\boldsymbol{x}$ for an optimization problem that satisfy all the constraints.*

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \quad | c_i(\boldsymbol{x}) \leq 0,\ i \in \mathcal{I} \text{ and } c_i(\boldsymbol{x}) = 0,\ i \in \mathcal{E}\}.$$

*A vector $\boldsymbol{x} \in \mathbb{R}^n$ is a **feasible point** if $\boldsymbol{x} \in \Omega$.*

A set is a **convex set** [20] if it contains a line segment joining any two points $\mathbf{x}$ and $\mathbf{y}$ in the set, i.e.,

$$\mathbf{x}, \mathbf{y} \in \Omega, \quad \theta \in [0,1] \Rightarrow \theta\mathbf{x} + (1-\theta)\mathbf{y} \in \Omega.$$

Examples of convex sets are, for instance, hyperplanes, halfspaces, Euclidean balls, and polyhedra [20]. The feasible set of a nonlinear optimization problem must be non-empty in order for the problem to admit a solution.

The constrained optimization problem (NLP) is described using inequality constraints. In general, nonlinear optimization solvers can be categorized based on how the inequalities are dealt with. With the aim of distinguishing whether these constraints are exactly held or not, the active set is defined as follows.

Definition 3.2. *(from [87]) The* ***active set*** *of the optimization problem (NLP) at a point* $\boldsymbol{x}$ *is defined as*

$$\mathcal{A}(\boldsymbol{x}) = \{i \in \{1,\ldots,2n+m\} \text{ such that } c_i(\boldsymbol{x}) = 0\}.$$

The active constraints restrict the possible directions from a feasible point $\mathbf{x}$. On the other hand, if a constraint is inactive in a feasible point, then any small enough perturbation will end up in another feasible point.

Convex analysis

Convex optimization is a special case of mathematical optimization. One of the great advantages of convex problems is that any local solution is a global solution. Thus, most of the nonlinear optimization solvers, approximate (NLP) by convex sub-problems. Let some preliminary concepts be introduced.

Definition 3.3. *(from [20]) A function* $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *is a* ***convex function*** *if the domain of* f *(****dom*** f*) is a convex set and*

$$f(\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leq \theta f(\boldsymbol{x}) + (1-\theta) f(\boldsymbol{y})$$
$$\forall \boldsymbol{x},\ \boldsymbol{y} \in \boldsymbol{dom} f \quad \theta \in [0,1].$$

Here, the **dom** f specifies the subset of $\mathbb{R}^n$ of points $\mathbf{x}$ for which $f(\mathbf{x})$ is defined.

Definition 3.4. *(from [20]) A problem such as (NLP) is a* ***convex optimization problem*** *if* $f(\boldsymbol{x})$ *and* $c_i(\boldsymbol{x})$ *with* $i \in \mathcal{I}$ *are convex functions, and* $c_i(\boldsymbol{x})$ *with* $i \in \mathcal{E}$ *are affine functions.*

Here, the term affine refers to functions with the form $f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$. It is important to note that the feasible set of a convex problem is convex.

Theorem 3.1 relates the convexity property with the second-order information of the function.

Theorem 3.1. *(from [20]) Let be* f *a twice differentiable function with convex domain.*

$$f \text{ is convex} \iff \nabla^2 f(\boldsymbol{x}) \succeq 0 \text{ for all } \boldsymbol{x} \in \boldsymbol{dom}\ f.$$

Here, $(\nabla^2 f(\boldsymbol{x}))_{ij} = \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}$ $i,j = 1,\ldots,n$ *is the Hessian of the function* f*. If* $\nabla^2 f(\boldsymbol{x}) \succ 0$*, then* f *is strictly convex.*

The expression "$\mathbf{A} \succeq \mathbf{B}$" ("$\mathbf{A} \succ \mathbf{B}$") means that $\mathbf{A} - \mathbf{B}$ is a positive semi-definite (positive definite) matrix.

A vector $\overline{\mathbf{x}} \in \Omega$ is a **global** solution of (NLP) if $\forall\ \mathbf{x} \in \Omega, \quad f(\overline{\mathbf{x}}) \leq f(\mathbf{x})$. In addition, it is said a **local** solution of (NLP) if $\overline{\mathbf{x}} \in \Omega$, and there is a neighbourhood $\mathcal{N} \subset \Omega$ of $\overline{\mathbf{x}}$ such that $f(\mathbf{x}) \geq f(\overline{\mathbf{x}})$ for $\mathbf{x} \in \mathcal{N}$.

Theorem 3.2 states one of the most important properties of convex optimization. For nonconvex problems, implications in only one direction are satisfied, from the top to the bottom. The proof of this theorem can be found in [20]. Definition 3.5 is included to complete the theorem.

Definition 3.5. *(from [20]) A vector* $\boldsymbol{d} \in \mathbb{R}^n$ *is*

- *A* ***feasible direction*** *at* $\boldsymbol{x} \in \Omega$ *if there exists a real number* $\epsilon_1 > 0$ *such that* $\boldsymbol{x} + t\boldsymbol{d} \in \Omega$ *for all* $t \in (0, \epsilon_1)$.
- *A* ***descent direction*** *at* $\boldsymbol{x} \in \Omega$ *if there exists a real number* $\epsilon_2 > 0$ *such that* $f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x})$ *for all* $t \in (0, \epsilon_2)$.
- *A* ***feasible descent direction*** *at* $\boldsymbol{x} \in \Omega$ *if* $\boldsymbol{d}$ *is both feasible direction and a descent direction at* $\boldsymbol{x}$.

Theorem 3.2. *(from [20]) Suppose that (NLP) is a convex problem, and that* $\overline{\boldsymbol{x}} \in \Omega$. *Then the following are equivalent:*

$$\overline{\boldsymbol{x}} \text{ is a global solution.}$$
$$\Updownarrow$$
$$\overline{\boldsymbol{x}} \text{ is a local solution.}$$
$$\Updownarrow$$
$$\text{At } \overline{\boldsymbol{x}} \text{ there is no feasible descent direction } \boldsymbol{d}.$$

Optimality conditions

The optimality conditions are some necessary and sufficient expressions to check if a given point $\mathbf{x}$ is indeed a local solution. Nonlinear optimization solvers generally stop when the first-order optimality conditions are satisfied for a given tolerance.

Constraint Qualifications (CQ) are regularity conditions in the constraints to ensure that they do not show degenerate behaviour at the Karush-Kuhn-Tucker (KKT) point $\overline{\mathbf{x}}$ (cf. below). There are plenty of CQ. Here, two of the most popular ones are cited.

Definition 3.6 (**Linear independence constraint qualification (LICQ)** [87])**.** *The LICQ holds at* $\overline{\boldsymbol{x}}$ *if the gradients* $\nabla c_i(\overline{\boldsymbol{x}})$, $i \in \mathcal{A}(\overline{\boldsymbol{x}})$ *are linearly independent.*

Definition 3.7 (**Mangasarian-Fromovitz constraint qualification (MFCQ)** [87])**.** *The MFCQ holds at* $\overline{\boldsymbol{x}}$ *if there exists a vector* $\boldsymbol{d} \in \mathbb{R}^n$ *such that*

$$\begin{aligned} \nabla c_i(\overline{\boldsymbol{x}})^T \boldsymbol{d} &< 0 \quad i \in \mathcal{A}(\overline{\boldsymbol{x}}) \cap \mathcal{I} \\ \nabla c_i(\overline{\boldsymbol{x}})^T \boldsymbol{d} &= 0 \quad i \in \mathcal{E} \end{aligned}$$

and the set of equality constraint gradients $\{\nabla c_i(\overline{\boldsymbol{x}}) : i \in \mathcal{E}\}$ *is linearly independent.*

In particular, if the feasible region is formed by only linear constraints, then the constraint qualifications are met (see [87]). Therefore, in Chapter 8 and 9, the CQ are satisfied for all feasible points since the minimum compliance problem is formulated in the nested form (P_N^c) (only linear constraints, see Chapter 2).

Definition 3.8. *(from [87]) The **Lagrangian function** of (NLP) is defined as*

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{m+2n} \lambda_i c_i(\boldsymbol{x}).$$

Here $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{m+2n})^T$ are the Lagrangian multipliers of all general constraints.

The first-order conditions for a point $\overline{\mathbf{x}}$ to be a local solution of the problem (NLP), are gathered Theorem 3.3.

Theorem 3.3 (**First-order necessary conditions** [87])**.** *Suppose that $\overline{\boldsymbol{x}}$ is a local solution of (NLP) and that a CQ holds at $\overline{\boldsymbol{x}}$. Then, there is a Lagrangian multiplier vector $\overline{\boldsymbol{\lambda}}$ such that the following conditions are satisfied at $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{\lambda}})$.*

$$\nabla \mathcal{L}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{\lambda}}) = \nabla f(\overline{\boldsymbol{x}}) + J(\overline{\boldsymbol{x}})^T \overline{\boldsymbol{\lambda}} = \boldsymbol{0}, \tag{3.1}$$

$$c_i(\overline{\boldsymbol{x}}) \leq 0 \quad i \in \mathcal{I}, \tag{3.2}$$

$$c_i(\overline{\boldsymbol{x}}) = 0 \quad i \in \mathcal{E}, \tag{3.3}$$

$$\overline{\lambda}_i \geq 0 \quad i \in \mathcal{I}, \tag{3.4}$$

$$c_i(\overline{\boldsymbol{x}})\overline{\lambda}_i = 0 \quad i \in \mathcal{I}, \tag{3.5}$$

where $J(\boldsymbol{x}) = [\nabla c_i(\boldsymbol{x})^T]_{i=1,\ldots,m+2n} : \mathbb{R}^n \mapsto \mathbb{R}^{m+2n \times n}$ is the Jacobian matrix of the constraints. Equation (3.1) refers to the stationarity condition, equations (3.2)-(3.3) are the primal feasibility conditions, and equation (3.5)) is the complementarity condition.

Finally, the second-order condition gathers how the second derivatives affect the optimality condition. The second-order conditions are assumed in some theoretical convergence proofs for second-order methods (see Chapter 8).

Theorem 3.4 (**Second-order sufficient conditions** [87])**.** *Suppose that for some feasible point $\overline{\boldsymbol{x}}$ there is a Lagrangian multiplier vector for which the KKT conditions are satisfied. Suppose also that*

$$\boldsymbol{p}^T \nabla^2 \mathcal{L}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{\lambda}}) \boldsymbol{p} > 0 \quad \forall \boldsymbol{p} \text{ such that } \boldsymbol{J}(\overline{\boldsymbol{x}})^T \boldsymbol{p} = \boldsymbol{0}, \text{ with } \boldsymbol{p} \neq \boldsymbol{0}.$$

Then $\overline{\boldsymbol{x}}$ is a strict local solution of (NLP).

Duality Theory

This section outlines the duality theory which consists of defining the general nonlinear optimization problem (NLP) alternatively. This new *dual problem* is defined using *dual variables* $\boldsymbol{\lambda}$ instead of the primal variable $\mathbf{x}$, as in the original problem (NLP). In some cases the dual problem is much easier to solve, and computationally less expensive. The purpose of this sub-section is to outline some theoretical details assumed in Chapter 8[1].

Definition 3.9. *(from [20]) For a given optimization problem such as (NLP), the* ***Lagrangian dual function*** ρ *is defined as the minimum value the Lagrangian primal function* $\mathcal{L}$ *can take over the primal variable* $\boldsymbol{x}$*.*

$$\rho(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x}} \quad \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}).$$

The Lagrangian dual function is the infimum of a family of affine functions (linear function of $\boldsymbol{\lambda}$). Thus, $\rho(\boldsymbol{\lambda})$ is always concave for any general problem (NLP). Let the Lagrangian dual problem (NLP_d) be defined as

$$\begin{array}{ll} \underset{\boldsymbol{\lambda}}{\text{maximize}} & \rho(\boldsymbol{\lambda}) \\ \text{subject to} & \boldsymbol{\lambda} \geq 0. \end{array} \qquad (NLP_d)$$

One important property, called *weak duality* (see for instance [20] for the proof), is that the Lagrangian dual function gives a lower bound of the optimal value of $f(\mathbf{x})$, i.e.,

$$\rho(\overline{\boldsymbol{\lambda}}) \leq f(\overline{\boldsymbol{x}}).$$

The strong duality ([20]) holds when $\rho(\overline{\boldsymbol{\lambda}}) = f(\overline{\mathbf{x}})$, i.e. when there is no gap between these two optimal values. When (NLP) is convex, the strong duality generally holds, however some constraint qualification conditions are needed to ensure it.

3.2 Methods for nonlinear constrained problems

In this section, general aspects and characteristics of some existing nonlinear optimization algorithms are described. Nonlinear optimization methods can be categorized as follows [87].

- **Penalty methods:** The constrained optimization problem is replaced by a sequence of sub-problems in which the constraints are included in the objective function using a *penalty function*.

[1] In the proposed TopSQP (Chapter 8) the inequality quadratic problem (IQP) is reformulated into its dual to avoid the storage and computation of the Hessian, and thus, to reduce memory and time demand.

The resulting unconstrained sub-problem is, for instance,

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) + \mu \left(\sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| + \sum_{i \in \mathcal{I}} [c_i(\mathbf{x})]^+ \right),$$

or

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) + \frac{\mu}{2} \left(\sum_{i \in \mathcal{E}} c_i(\mathbf{x})^2 + \sum_{i \in \mathcal{I}} ([c_i(\mathbf{x})]^+)^2 \right),$$

where the penalty parameter is $\mu > 0$, and the operation $[c_i(\mathbf{x})]^+$ denotes $\max(c_i(\mathbf{x}), 0)$.

The aim is to solve the unconstrained minimization problem for a sequence of increasing values of $\mu \uparrow \infty$.

- **Augmented Lagrangian methods:** In this type of methods, the Lagrangian multipliers are explicitly included in the objective function. The Augmented Lagrangian method combines properties of the Lagrangian function and the quadratic penalization introduced above [28]. The inequality constraints are reformulated as equality constraints using slack variables, and thus, the approximate sub-problem to solve at each outer iteration is

$$\begin{aligned} \underset{\mathbf{x},\mathbf{s}}{\text{minimize}} \quad & f(\mathbf{x}) - \sum_{i \in \mathcal{I}} \lambda_i (c_i(\mathbf{x}) + s_i) - \sum_{i \in \mathcal{E}} \lambda_i c_i(\mathbf{x}) + \frac{\mu}{2} \left(\sum_{i \in \mathcal{I}} (c_i(\mathbf{x}) + s_i)^2 + \sum_{i \in \mathcal{E}} c_i(\mathbf{x})^2 \right) \\ \text{subject to} \quad & s_i \geq 0 \quad i \in \mathcal{I}. \end{aligned}$$

  The sub-problem is solved for fixed values of $\boldsymbol{\lambda}$ and μ. Then, both parameters are updated until the KKT conditions are satisfied.

  MINOS [84], LANCELOT [29], and PENNON [75], are examples of nonlinear software based on Augmented Lagrangian methods.

- **Sequential Quadratic Programming:** This nonlinear method obtains the search direction $\mathbf{d}$ by minimizing a quadratic programming problem where the objective function is normally a convex and quadratic approximation of the Lagrangian and the constraints are linearized [15].

$$\begin{aligned} \underset{\mathbf{d}}{\text{minimize}} \quad & \nabla f(\mathbf{x})^T \mathbf{d} + \tfrac{1}{2} \mathbf{d}^T \nabla^2 \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \mathbf{d} \\ \text{subject to} \quad & c_i(\mathbf{x}) + \nabla c_i(\mathbf{x})^T \mathbf{d} \leq 0 \quad i \in \mathcal{I}, \\ & c_i(\mathbf{x}) + \nabla c_i(\mathbf{x})^T \mathbf{d} = 0 \quad i \in \mathcal{E}, \end{aligned}$$

  SQP methods solve a sequence of Quadratic Programming (QP) problems. More details of QP problems can be found in [87]. Once the search direction is estimated, the primal variable $\mathbf{x}$ and the estimates of the Lagrangian multipliers are updated until a KKT point is found.

  Examples of existing SQP algorithms are SNOPT [55], NPSOL [56], FILTERSQP [46], and KNITRO/ACTIVE [26]. Chapter 8 explains in more detail a SQP-type method.

- **Interior point methods:** Slack variables are introduced to transform the inequalities to equality constraints. In addition, the objective function is defined with a barrier function to deal with the bound constraints. For a given value of the barrier parameter $\mu > 0$, the algorithm solves the following sub-problem

$$\begin{aligned} \underset{\mathbf{x},\mathbf{s}}{\text{minimize}} \quad & f(\mathbf{x}) - \mu \left(\sum_{i \in \hat{\mathcal{I}}} \ln s_i + \sum_{i=1}^{n} \ln(x_i - l_i) + \sum_{i=1}^{n} \ln(u_i - x_i) \right) \\ \text{subject to} \quad & g_i(\mathbf{x}) + s_i = 0, \quad i \in \hat{\mathcal{I}}, \\ & g_i(\mathbf{x}) = 0, \quad i \in \mathcal{E}. \end{aligned}$$

  For a fixed μ, the goal is to obtain a local solution of the barrier problem using a Newton's method [41]. The search direction is obtained by solving the so-called KKT system[2]. Generally, these sub-problems are not solved to optimality.

  Then, the barrier parameter is decreased $\mu \to 0$ until convergence, so that $\overline{\mathbf{x}}_\mu \to \overline{\mathbf{x}}$.

  Examples of nonlinear interior point methods available in the community are LOQO [116], IPOPT [117], KNITRO/DIRECT, and KNITRO/CG [26]. Chapter 9 gathers more implementation details of an interior point method.

Interior point methods together with SQP methods are considered the most powerful solvers nowadays [59], [38], and [11]. Therefore, both algorithms are implemented for the minimum compliance problem in Chapters 8 and 9. Additionally, these methods solve sequence of sub-problems in which the constraints are linearized. The topology optimization formulation proposed in Chapter 7 is based on this property. Thus, both SQP and interior point methods are suitable for this automatic continuation approach.

Nonlinear solvers need to deal with several challenges, such as how to solve the sub-problems, how to deal with nonconvexity, how to deal with infeasible and unbounded sub-problems, and how to ensure progress towards a KKT point, among others. Throughout the rest of the section, some techniques and methods commonly used to solve these challenges are introduced.

3.2.1 Strategies for determining the step

There exist two different techniques to ensure the progress of the solvers to a KKT point, namely line search [82] and trust region [27] strategies. These strategies require the use of either merit functions [14] or filters [47] to measure the progress. In particular, a line search combined with a merit function is implemented in both the SQP in TopSQP (Chapter 8) and the interior point method in TopIP (Chapter 9).

[2]Special saddle-point system appeared from a Newton's method [83] iteration, with the form $\nabla \mathbf{F} \boldsymbol{\Delta} = -\mathbf{F}$, with $\mathbf{F}$ the KKT conditions. Here, $\boldsymbol{\Delta}$ is the search direction.

Line search methods

Line search is a technique to decide how far the algorithm should move along the given search direction $\mathbf{d}_k$. The new iterate solution is then $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, where $0 < \alpha_k \leq 1$ is the step length chosen by the line search at the kth iteration.

The aim of line search strategies is to find a step length α to give a substantial reduction of $f(\mathbf{x})$ [82]. Ideally, the goal is to find the minimizer of $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$ with $1 \geq \alpha > 0$. However, this is computationally expensive. In practice, the algorithm tries a sequence of candidates of α enforcing some sufficient decrease conditions. For instance, to ensure the *Wolfe conditions* [87],

$$f(\mathbf{x}_k + \alpha \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$
$$\nabla f(\mathbf{x}_k + \alpha \mathbf{d}_k)^T \mathbf{d}_k \geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k,$$

for some constants $c_1 \in (0,1), c_2 \in (c_1, 1)$. The first condition is commonly called *Armijo condition* [87]. Algorithm 1 outlines one of the most popular line search strategies based on a backtracking search satisfying the Armijo condition. Nevertheless, there are other conditions that a line search can follow to force a sufficient decrease, such as the Goldstein conditions [87], the 1D-Gamma, and 2D-Gamma conditions [72]. Other more sophisticated and complicated line search strategies based on finding the minimum of $\phi(\alpha)$ can be applied, for example interpolation techniques [87]. Finally, new approaches to extend the search from line to curve using arc search strategies are found in the literature, see e.g. [115] and [65].

Algorithm 1 Line search Backtracking algorithm [87].

Input: Choose $\tau \in (0,1)$ and $c \in (0,1)$.

1: Initialize $\alpha = 1$.
2: **repeat**
3: **if** $f(\mathbf{x}_k + \alpha \mathbf{d}_k) \leq f(\mathbf{x}_k) + c\alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ **then**
4: `sufficient decrease = true`
5: **else**
6: $\alpha = \tau \alpha$.
7: **end if**
8: **until** `sufficient decrease`
9: **return**

For constrained optimization problems, a sufficient decrease in the objective function is not enough. There is a need of balance between minimizing the objective function and satisfying the constraints [87]. Then, the objective function in Algorithm 1 is replaced with a merit function or with the use of filters (cf. below).

Line search strategies are used in some nonlinear software such as LOQO, KNITRO/DIRECT, IPOPT, and SNOPT.

Trust region methods

Trust region strategies are the alternative of line search methods. The main idea is to define a region around the current iterate point $\mathbf{x}_k$ such that a selected model fits adequately with the real objective function, and thus, the method can trust the approximate model in this area. The model is minimized in this region to be able to choose the step for the current iterate [27].

The main difference between line search strategies and trust region methods is that the latter finds α_k and $\mathbf{d}_k$ simultaneously. At every iteration, the size of the trust region is modified depending on the performance of the step selected. Trust region methods choose a suitable Δ_k, such that the descent direction is inside the ball of radius Δ_k, i.e.,

$$||\mathbf{d}_k|| \leq \Delta_k.$$

This inequality is included as an extra constraint in the optimization problem. At a given iteration, a ratio ρ_k is defined based on a model function m_k and the original objective function [87],

$$\rho_k = \frac{\text{actual reduction}}{\text{predicted reduction}} = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{d}_k)}{m_k(\mathbf{0}) - m_k(\mathbf{d}_k)}.$$

Algorithm 2 Outline of a trust region method [87].

Input: Choose $\Delta_{\max} > 0$, $\Delta_0 \in (0, \Delta_{\max})$ and $\kappa \in [0, \frac{1}{4})$.

repeat
  Obtain $\mathbf{d}_k$ such that the model function $m_k(\mathbf{d}_k)$ is minimized and $||\mathbf{d}_k|| \leq \Delta_k$ is satisfied.
  Evaluate ratio ρ_k.
  if $\rho_k < \frac{1}{4}$ **then**
    $\Delta_{k+1} = \frac{1}{4}\Delta_k$.
  else
    if $\rho_k > \frac{3}{4}$ and $||\mathbf{d}_k|| = \Delta_k$ **then**
      $\Delta_{k+1} = \min(2\Delta_k, \Delta_{\max})$.
    else
      $\Delta_{k+1} = \Delta_k$.
    end if
  end if
  if $\rho_k > \kappa$ **then**
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$.
  else
    $\mathbf{x}_{k+1} = \mathbf{x}_k$.
  end if
  $k = k + 1$.
until `convergence`
return

Depending on the ratio value, the size of the region will increase, decrease, or remain the same (see Algorithm 2). Typically, the model function is a quadratic approximation of the objective function. For nonlinear constraint problems, merit functions or filters are used instead (cf. below).

Nowadays, there are several nonlinear optimization methods that are based on trust regions strategies, such as KNITRO/CG, LANCELOT, and FILTERSQP.

Merit function

A merit function balances the conflicting goal of reducing the objective function and satisfying the constraints. It is defined using a penalty parameter $\pi > 0$ which represents the weight assigned to the satisfaction of the constraints [14]. Several alternative functions to use as merit function are [87]:

- l_1 **merit function:**

$$\phi(\mathbf{x}, \pi) = f(\mathbf{x}) + \pi \left(\sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| + \sum_{i \in \mathcal{I}} [c_i(\mathbf{x})]^+ \right).$$

- **Sum-of-squares merit function:**

$$\phi(\mathbf{x}, \pi) = f(\mathbf{x}) + \frac{\pi}{2} \left(\sum_{i \in \mathcal{E}} c_i(\mathbf{x})^2 + \sum_{i \in \mathcal{I}} ([c_i(\mathbf{x})]^+)^2 \right).$$

- **Fletcher's augmented Lagrangian merit function:**

$$\phi(\mathbf{x}, \pi) = f(\mathbf{x}) - \sum_{i \in \mathcal{E}} \lambda_i c_i(\mathbf{x}) - \sum_{i \in \mathcal{I}} \lambda_i [c_i(\mathbf{x})]^+ + \frac{\pi}{2} \left(\sum_{i \in \mathcal{E}} c_i(\mathbf{x})^2 + \sum_{i \in \mathcal{I}} ([c_i(\mathbf{x})]^+)^2 \right).$$

The merit function is in charge of controlling the step length α_k in line search methods, and the ratio ρ_k in trust region methods. The penalty parameter is updated at every iteration and it plays an important role in the convergence rate of the algorithm. For different updating schemes, see for instance [119] and [32]. SNOPT and LOQO are examples of nonlinear optimization software that use merit functions.

In particular, the implementation of both, TopSQP (Chapter 8) and TopIP (Chapter 9), are based on the l_1-merit function with a very simple update rule for the penalty parameter [87],

$$\pi = ||\boldsymbol{\lambda}||_\infty.$$

Filters

The second mechanism to control the acceptance or rejection of the step is the use of filters. It is based on multi-objective function since the idea is to minimize the objective

function, but at the same time, satisfy the constraints [47]. In other words, both $f(\mathbf{x})$ and $h(\mathbf{x})$ must be minimized, where

$$h(\mathbf{x}) = \sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| + \sum_{i \in \mathcal{I}} [c_i(\mathbf{x})]^+.$$

Filters will accept a trial step depending on the value of the pair (f_k, h_k).

Definition 3.10. *(from [87])*

- *A pair (f_k, h_k) is said to **dominate another pair** (f_l, h_l) if both $f_k \leq f_l$ and $h_k \leq h_l$.*
- *A filter is a list of pairs (f_l, h_l) such that no pair dominates any other.*
- *An iterate $\boldsymbol{x}_k$ is said to be acceptable to the filter if (f_k, h_k) is not dominated by any other pair in the filter.*

Examples of nonlinear solvers with filter techniques are IPOPT and FILTERSQP.

3.2.2 Existence of solution of saddle-point problems

Some mathematical programming algorithms, such as interior point methods and some QP solvers, require the solution of saddle-point systems. The saddle-point problem is defined as the following linear system

$$\mathbf{W}\boldsymbol{\Delta} = \mathbf{F} \tag{3.6}$$

with

$$\mathbf{W} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & -\mathbf{C} \end{bmatrix}.$$

Here, $\mathbf{A}$ and $\mathbf{C}$ are square matrices. The saddle-point must satisfy at least one of the following conditions [12]:

- $\mathbf{A}$ is symmetric.
- $\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$ (symmetric part) is positive semi-definite.
- $\mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B}$.
- $\mathbf{C}$ is symmetric and positive definite.
- $\mathbf{C} = \mathbf{0}$.

The most typical scenario is when all the conditions are satisfied [12]. Examples of these problems arise in Chapters 8 and 9.

For simplicity, let assume that $\mathbf{A} = \mathbf{H} \in \mathbb{R}^{n \times n}$ is the Hessian of the Lagrangian function, $\mathbf{C} = \mathbf{0} \in \mathbb{R}^{m \times m}$, and $\mathbf{B} = \mathbf{J} \in \mathbb{R}^{m \times n}$ is the Jacobian of the active constraints. In this case, the matrix $\mathbf{W}$ is called the Karush-Kuhn-Tucker matrix. The next theorems contain the requirements for a KKT matrix to ensure existence of solution of (3.6).

Theorem 3.5. *(from [87]) Let $\boldsymbol{J}$ have a full row rank, and assume that the reduced-Hessian matrix $\boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z}$ is positive definite. Then the KKT matrix $\boldsymbol{W}$ is nonsingular, and hence there is a unique vector satisfying the linear system (3.6).*

Here, $\mathbf{Z} \in \mathbb{R}^{n \times n-m}$ is a matrix which columns are a basis for the null-space of $\mathbf{J}$ [87].

Definition 3.11. *(from [61]) The **inertia** of a symmetric matrix $\boldsymbol{W}$ is the triple (i_+, i_-, i_0), where i_0, i_+ and i_- be the number of zero, positive and negative eigenvalues of $\boldsymbol{W}$, respectively.*

Theorem 3.6. *(from [87]) Suppose $\boldsymbol{W}$ is defined as (3.6) with $\boldsymbol{A} = \boldsymbol{H}$, $\boldsymbol{C} = \boldsymbol{0}$, and $\boldsymbol{B} = \boldsymbol{J}$ the Jacobian of the constraints (full rank). Then*

$$inertia(\boldsymbol{W}) = inertia(\boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z}) + (m, m, 0).$$

Therefore, if $\boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z}$ is positive definite, $inertia(\boldsymbol{W}) = (n, m, 0)$ [87].

In order to ensure the existence of solution of the saddle-point problem, the matrix $\mathbf{W}$ must have the correct inertia. In the next sub-section, some methods available in the literature to correct the inertia of these systems are cited. Nevertheless, in Chapters 8 and 9 the existence of solution of the KKT systems is assumed. The Hessian is approximated using a positive definite matrix, thus, the reduced-Hessian is also positive definite, and the inertia is always correct.

Saddle-point systems can be solved using direct methods [87], such as Schur complement, null-space methods, and LDL factorization, or using iterative methods (see Section 4). For more details of saddle-point problems and some existing techniques available to solve them see the review article [12].

3.2.3 Dealing with nonconvex problems

In the previous sub-section, the importance of the inertia in a saddle-point problem has been introduced. If the inertia of the KKT system in interior point methods is not correct, the search direction may not be a descent direction (of a merit function, for instance). Thus, the solver could end in a local maximum or a stationary point. Algorithms dealing with nonconvex problems, such as interior point methods, need to modify or perturb the saddle-point system to ensure the existence of solution. In addition, the QP sub-problems

of SQP methods should generally be convex. There are many different ways of handling the nonconvexity of the problems.

The easiest technique consists of adding a constant diagonal matrix to the Hessian of the Lagrangian, big enough, such that the eigenvalues of the reduced-Hessian become positive, i.e.,

$$\begin{aligned}&\hat{\mathbf{H}} = \mathbf{H} + \gamma \mathbf{I}\\ &\text{with}\\ &\gamma = \max(0, -\lambda_{\min}(\mathbf{Z}^T \mathbf{H} \mathbf{Z}) + \epsilon).\end{aligned}$$

Here, $\lambda_{\min}$ refers to the minimum eigenvalue, $\mathbf{I}$ the identity matrix, and $1 \gg \epsilon > 0$. More details of this inertia correction strategy can be found in [87].

In the mid 1950s, a new algorithm was implemented to accelerate the iteration of Newton's method. This quasi-Newton's method was proven to be more reliable and fast than the classical Newton's method. In particular, the BFGS (Broyden-Fletcher-Godfarb-Shanno) method is one of the most popular quasi-Newton's algorithms [85]. This method is nowadays commonly used to approximate the Hessian when there is no available second-order information (or is computationally expensive). Software such as IPOPT and SNOPT, use a limited memory BFGS approach to estimate the Hessian [117] and [55]. Equation (3.7) outlines the general iterative process for obtaining a BFGS approximation [87].

$$\begin{aligned}&\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\\ &\text{with}\\ &\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k,\\ &\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k).\end{aligned} \tag{3.7}$$

Inertia controlling methods are included in some algorithms where the linear system has an incorrect inertia. Plenty of literature can be found in this regard, for instance [45], [54], [53], [52], and [51], where LDL factorization techniques are detailed. Additionally, some modifications and perturbations to the KKT matrix are discussed in [67] and in the implementation of the interior point method in IPOPT [117]. Finally, some convexification strategies to obtain convex problems are explained in [58] and [60].

Nevertheless, Chapters 8 and 9 do not include any of the techniques mentioned above. Instead, a convex approximation of the Hessian based on its specific mathematical structure is proposed. Part of the information of the exact Hessian is lost at the expense of reducing computational effort.

3.2.4 Other implementation techniques

Some features are often added to improve the practical performance of the algorithms. For instance, it is common to include a corrector step in interior point algorithms [80]. The idea is to compensate the errors made due to the linearization by including two steps namely, predictor and corrector steps [87]. It has been proved very effective for linear and convex quadratic problems. The adaptive barrier parameter update strategy in Chapter 9 is based on [86]. Here, the *Mehrotra's predictor-corrector method* [80] is not implemented since [86] states that it is not robust for nonlinear programming. Thus, the proposed algorithm does not introduce it either.

The *Maratos effect* [79] is the phenomenon where the algorithm fails to converge fast because it rejects steps that make progress towards the solution. The implementation of general nonlinear solvers also needs the incorporation of some techniques to avoid this issue. Examples of these strategies are, the use of second-order correction, the use of different merit functions, and the flexibility of increasing the merit function in a fixed number of iterations (called *watchdog strategy*) [87]. However, since the considered problem formulation (P_N^c) contains only linear constraints, this effect does not occur in practice, see [25].

Finally, the algorithms need to deal with infeasibility and unboundedness of problems. Some techniques to detect infeasible or unbounded problems are explained in the comparison study [11]. SNOPT software, for instance, forces a feasible starting point while the interior point algorithms in KNITRO, LOQO, and IPOPT allow infeasible iterates and detect infeasibility through line search, filters, or feasibility restoration phases, [116] and [117]. Nevertheless, the proposed solvers in Chapter 8 and 9 assumed feasible and bounded problems.

Some practical implementation details are included in Chapters 8 and 9 to produce fast convergence. For instance, in TopSQP, the KKT conditions of some problems stall in a value close to the optimal. To promote convergence in those cases, the maximum possible step direction is taken without the backtracking (see Chapter 8). Furthermore, the scaling of the problem plays an extremely important role in the efficiency of the solvers. For more details of practical implementation in optimization, the text book [57] is recommended.

4

Iterative methods for solving linear systems

The computational effort of some optimization solvers, such as interior point methods, relies on the solution of large-scale linear systems. The different existing techniques to solve these systems can be classified in two main groups, namely direct and iterative methods [57]. Direct methods are characterized for their robustness, but they have difficulties in solving large-scale linear systems due to the amount of memory and time required. On the other side, for this type of problems, iterative methods are particularly interesting since they have lower storage need and are easier to parallelize [73]. Structural topology optimization problems are characterized as large-scale problems involving millions of degrees of freedom. Thus, the performance of optimization methods is highly dependent on the solution of very large-scale systems. This chapter outlines different iterative methods available in the literature. More details of these techniques can be found in [12], [95], and [73]. This introductory chapter serves as a lead to Chapter 9, where an efficient iterative method to solve the KKT system of an interior point solver is proposed for the minimum compliance problem.

Let us consider the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. Iterative methods produce a sequence of $\{\mathbf{x}_k\}$ that is expected to converge to $\overline{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$. Depending on how this sequence of iterate points is defined, different techniques emerge, such as stationary iterations, Krylov sub-space, and multigrid methods [73]. These approaches can be defined for both positive definite and indefinite linear systems.

Most of these techniques terminate when the residual $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ is sufficiently small.

In practice, the Euclidean-norm of the relative residual error is used, i.e.,

$$||\mathbf{r}||_2 = \frac{||\mathbf{b} - \mathbf{A}\mathbf{x}||_2}{||\mathbf{b}||_2} \leq \epsilon,$$

for some tolerance $1 \gg \epsilon > 0$.

4.1 Stationary iterative methods

The Richardson [98] or stationary iteration is one of the simplest iterative techniques in which the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is modified into a linear fixed-point iteration as

$$\mathbf{x} = (\mathbf{I} - \mathbf{A})\mathbf{x} + \mathbf{b}.$$

In particular, these methods solve the more general iteration

$$\mathbf{x}_{i+1} = \mathbf{M}\mathbf{x}_i + \mathbf{c}. \tag{4.1}$$

A common technique to obtain the above iteration, consists of splitting the matrix as $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$. Here, $\mathbf{A}_1$ is nonsingular, and the system $\mathbf{A}_1\mathbf{y} = \mathbf{q}$ is easy to solve. Then, the iteration matrix is defined as $\mathbf{M} = -\mathbf{A}_1^{-1}\mathbf{A}_2$ and $\mathbf{c} = \mathbf{A}_1^{-1}\mathbf{b}$. This technique is called *preconditioned* Richardson iteration [73].

Two of the most common and popular matrix splitting techniques are Jacobi and Gauss-Seidel methods [73]. The matrix $\mathbf{A}$ is partitioned in three parts; the diagonal part $\mathbf{D}$, the upper triangular part $\mathbf{U}$, and the lower triangular part $\mathbf{L}$. In the first method, the matrices are defined as $\mathbf{A}_1 = \mathbf{D}$ and $\mathbf{A}_2 = \mathbf{L} + \mathbf{U}$, while in Gauss-Seidel the partition is $\mathbf{A}_1 = \mathbf{U} + \mathbf{D}$ and $\mathbf{A}_2 = \mathbf{L}$. Thus, Jacobi and Gauss-Seidel iterations are defined as

$$\mathbf{x}_{i+1} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}_i + \mathbf{D}^{-1}\mathbf{b},$$

and

$$\mathbf{x}_{i+1} = -(\mathbf{U} + \mathbf{D})^{-1}\mathbf{L}\mathbf{x}_i + (\mathbf{U} + \mathbf{D})^{-1}\mathbf{b},$$

respectively. These stationary methods are commonly used as preconditioner of other more efficient iterative methods, such as Krylov sub-space methods, or as smoother functions in multigrid cycles (cf. below) [105] and [73].

The convergence rate of stationary iterative methods depends on the condition number[1] of the matrix $\mathbf{A}$ [73]. Preconditioner matrices will help to reduce the condition number and, therefore, decrease the number of iterations needed to converge [73].

[1] (from [95]) The condition number κ of a matrix $\mathbf{A}$ with respect to a norm is given by

$$\kappa(\mathbf{A}) = ||\mathbf{A}||||\mathbf{A}^{-1}||.$$

In particular, a stationary iteration method for structural topology optimization problems has been implemented in [68] and [69]. Here, the so-called right-transforming iteration (see e.g. [99] and [126]) is applied to solve Newton's systems in a SAND form of topology optimization problems.

4.2 Krylov sub-space methods

Unlike the stationary iteration, Krylov sub-space methods [123] do not normally have access to the whole matrix. They obtain the sequence of iterates from the history of the previous iterations. These methods minimize the residual error over the affine space $\mathbf{x}_0 + \mathcal{K}_k$, where $\mathbf{x}_0$ is the initial iterate, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is the initial residual, and $\mathcal{K}_k = \text{span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0)$ is the Krylov sub-space. The speed of convergence of Krylov sub-space methods improves with respect to classical stationary iterations, although it still depends on the condition number of the matrix [95]. Details of these algorithms, implementation, and theoretical properties can be found in [73].

Examples of this type of iterative methods are the Conjugate Gradient (CG) method [66] and the Generalized Minimal Residual (GMRES) method [96], among others. The conjugate gradient method was developed in 1952, and is one of the most efficient methods for linear systems when the matrix is symmetric and positive definite. The basic algorithm is described in Algorithm 3, where the matrix $\mathbf{P}$ is a preconditioner.

In contrast, GMRES can also be applied to non-symmetric and indefinite systems. It is a generalization of the minimal residual algorithm (MINRES) [89] based on the Arnoldi process to obtain the orthonormal vectors. Algorithm 4 outlines the GMRES method.

For large-scale structural topology optimization problems, PCG (Preconditioner Conjugate Gradient) is commonly used to solve the equilibrium equations, see for instance [3]. In addition, Chapter 9 uses flexible GMRES (FGMRES, [94]) in combination with efficient preconditioners. In this article, an indefinite linear system needs to be efficiently solved at each interior point iteration.

4.3 Multigrid methods

Multigrid methods have been developed since 1964 [44], with a huge growth in the last decades. In contrast to other methods, the number of iterations does not depend on the condition of the matrix. Thus, these methods are characterized for their great efficiency [124]. They are considered numerically scalable since the computational cost is linearly dependent on the number of variables [3]. Multigrid methods are successfully used in plenty of different applications, such as shape optimization of turbine blades [40], computational fluid dynamics [125], and optimal control [105], among others.

Algorithm 3 Conjugate Gradient using a general preconditioner [73].

Input: $\mathbf{A}$,$\mathbf{b}$,$\mathbf{x}_0$, tolerance ω, and `max iter`.

1: Initialization. $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $k = 1$.
2: Preconditioner $\mathbf{z} = \mathbf{P}^{-1}\mathbf{r}$.
3: $\mathbf{p} = \mathbf{z}$.
4: **repeat**
5: $\quad \mathbf{w} = \mathbf{A}\mathbf{p}$.
6: $\quad \gamma = \mathbf{r}^T\mathbf{z}$.
7: $\quad \alpha = \frac{\gamma}{\mathbf{w}^T\mathbf{p}}$.
8: $\quad \mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$.
9: $\quad \mathbf{r} = \mathbf{r} - \alpha\mathbf{w}$.
10: $\quad \mathbf{z} = \mathbf{P}^{-1}\mathbf{r}$.
11: $\quad \beta = \frac{\mathbf{r}^T\mathbf{z}}{\gamma}$.
12: $\quad \mathbf{p} = \mathbf{z} + \beta\mathbf{p}$.
13: $\quad k = k + 1$.
14: $\quad rn = ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2 / ||\mathbf{b}||_2$
15: $\quad$ **if** $rn < \omega$ **then**
16: $\quad\quad$ `convergence = true`
17: $\quad$ **end if**
18: **until** `convergence` or $k \geq$ `max iter`.
19: **return**

In this section, a basic overview of geometric multigrid methods is presented, in which hierarchical meshes with their corresponding discretizations are required. The theoretical properties and convergence proofs are detailed in the monographs [21], [124], and [64]. Another type of multigrid methods, namely Algebraic Multigrid (AMG) methods, are becoming very popular since unstructured meshes can efficiently be solved. More details of this technique can be found in [81], [62], and [118].

The multigrid strategy combines two key parts, the smoothing and the coarse-grid correction steps. The smoothing step reduces the high frequency error. The smooth residual is used to obtain the solution of the system in a coarse grid. This estimation is then prolonged to a finer mesh where the low frequencies are corrected again with a smoother. Smoothing methods in multigrid algorithms are usually some kind of stationary iterative methods, such as Gauss-Seidel and Jacobi [124], [105], and [133]. However, if the matrix $\mathbf{A}$ at level l is indefinite, the construction of an appropriate smoother is not obvious and very problem dependent [12].

For the coarse-grid correction, some mappings are required to go from the coarse to the fine grids. Let us consider l hierarchy meshes with the corresponding finite element spaces $V_0 \subset V_1 \subset \ldots \subset V_l$ and mesh sizes $h_0 \geq h_1 \geq \ldots \geq h_l$, respectively. The restriction and prolongation operators are defined as follows,

Algorithm 4 Generalized Minimal Residual method [73].

Input: $\mathbf{A}$,$\mathbf{b}$,$\mathbf{x}_0$, tolerance ω, `max restart iter` and `max iter`.

Initialization $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{e}_1 = (1, 0, \ldots, 0)$, and $m = 1$.
$\mathbf{v}_1 = \frac{\mathbf{r}}{||\mathbf{r}||_2}$ and $\beta = ||\mathbf{r}||_2$.
repeat
  $\mathbf{v}_{m+1} = \mathbf{A}\mathbf{v}_m$.
  $j = 1$
  repeat
    $h_{j,m} = \mathbf{v}_{m+1}^T \mathbf{v}_j$.
    $\mathbf{v}_{m+1} = \mathbf{v}_{m+1} - \mathbf{h}_{j,m}\mathbf{v}_j$.
    $j = j + 1$.
  until $j == m$
  $h_{m+1,m} = ||\mathbf{v}_{m+1}||_2$.
  $\mathbf{v}_{m+1} = \frac{\mathbf{v}_{m+1}}{h_{m+1,m}}$.
  minimize $||\beta\mathbf{e}_1 - \mathbf{H}_m\mathbf{y}||$ to obtain $\mathbf{y}_m$.
  Form the approximate solution $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m$
  $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_m$
  $rn = ||\mathbf{b} - \mathbf{A}\mathbf{x}_m||_2 / ||\mathbf{b}||_2$
  if $rn < \omega$ **then**
    `convergence = true`
  end if
  $m = m + 1$.
  if $m =$`max restart iter` **then**
    Restart: $\mathbf{x}_0 = \mathbf{x}_m$, and $\mathbf{v}_1 = \frac{\mathbf{r}}{||\mathbf{r}||_2}$.
  end if
until `convergence` or $m \geq$`max iter`
return

Definition 4.1. *(from [105]) The coarse-to-fine operator, called prolongation is*

$$\boldsymbol{I}_{l-1}^{l} : V_{l-1} \longrightarrow V_l.$$

The fine-to-coarse operator, called restriction is

$$\boldsymbol{I}_{l}^{l-1} : V_l \longrightarrow V_{l-1}.$$

One of the simplest ways to define the prolongation operation is through a linear interpolation. The restriction is the inverse operation. Examples of these operators can be found in [95] and [3]. Figure 4.1 represents both operators more visually with a hierarchical discretization in 2D. The linear system to be solved at a given level l is defined as

$$\mathbf{A}_l\mathbf{x}_l = \mathbf{b}_l.$$

Thus, the methods requires the matrix $\mathbf{A}$ at each level of discretization. This can be generated either by assembling the matrix $\mathbf{A}$ for all the levels, or by using the Galerkin's

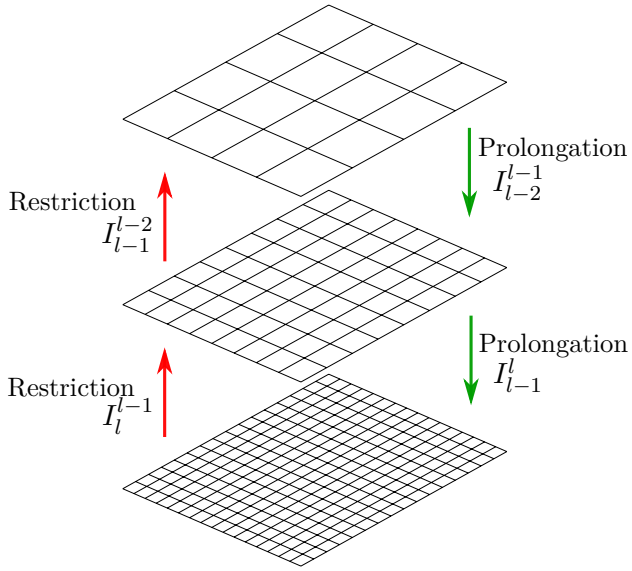


Figure 4.1: Example of three hierarchy levels of a 2D domain with the restriction and the prolongation operators.

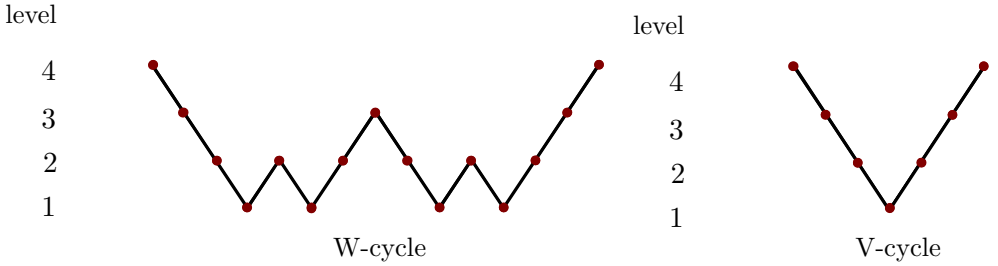


Figure 4.2: Example of two different types of cycles in a multigrid algorithm with 4 levels of hierarchy.

method, i.e.,

$$\mathbf{A}_{l-1} = \mathbf{I}_{l}^{l-1}\mathbf{A}_{l}\mathbf{I}_{l-1}^{l}.$$

Algorithm 5 outlines one cycle of the multigrid method. For a general convergent method, several cycles must be performed [124]. Figure 4.2 shows the effect of two different cycles. Here, the parameter m_c is either 2 (W-cycle) or 1 (V-cycle). The performance of multigrid methods are highly affected by the selection of smoother (function S), the number of pre-smoother and post-smoother iterations (ν_1 and ν_2), and the type of cycle (m_c).

This technique is already used as a method, but also as preconditioner in Krylov sub-space methods for structural topology optimization problems, such as in [126], [40], and [3]. Additionally, the multigrid cycle is used as a key part of the iterative method implemented in Chapter 9.

Algorithm 5 Multigrid cycle [17].

Input: $\mathbf{x}$ = MC($\mathbf{A}_l$,$\mathbf{b}_l$,$\mathbf{x}_l$, l, ν_1, ν_2, m_c)

if $l == 0$ **then**
  Solve the problem: $\mathbf{x}_0 = \mathbf{A}_0^{-1}\mathbf{b}_0$.
  `return`
end if
Pre-smoothing step: $\mathbf{x}_l$ = S($\mathbf{A}_l$,$\mathbf{b}_l$,$\mathbf{x}_l$, ν_1).
Grid correction: $\mathbf{r}_l = \mathbf{A}_l\mathbf{x}_l - \mathbf{b}_l$.
Restriction step: $\mathbf{r}_{l-1} = \mathbf{I}_l^{l-1}\mathbf{r}_l$.
$\mathbf{A}_{l-1} = \mathbf{I}_l^{l-1}\mathbf{A}_l\mathbf{I}_{l-1}^l$.
Initialize $c = 1$, $\mathbf{x}_{l-1} = \mathbf{0}$.
repeat
  $\mathbf{x}_{l-1}$ = MC($\mathbf{A}_{l-1}$,$\mathbf{r}_{l-1}$, $\mathbf{x}_{l-1}$, $l-1$,ν_1, ν_2, m_c).
  $c = c + 1$.
until $c = m_c$
Prolongation step: $\mathbf{x}_l = \mathbf{x}_l - \mathbf{I}_{l-1}^l\mathbf{x}_{l-1}$.
Post-smoothing step: $\mathbf{x}_l$ = S($\mathbf{A}_l$,$\mathbf{b}_l$,$\mathbf{x}_l$, ν_2).
return

5

Contributions and conclusions

This thesis thoroughly investigates some state-of-the-art nonlinear optimization algorithms for structural topology optimization problems. First of all, an extensive benchmarking study is carried out in order to establish whether general nonlinear algorithms outperform classical structural topology optimization methods. The results presented in Chapter 6 motivate the work of this thesis. The benchmarking study strongly recommends the use of the exact Hessian to produce designs with good objective function values and to reduce the number of iterations. The benchmarking also reinforces the initial belief, in which general nonlinear optimization methods can be at least as efficient and robust as classical structural optimization solvers.

Based on these results, a Sequential Quadratic Programming method, namely TopSQP, is implemented for the minimum compliance problem (Chapter 8). In order to promote fast convergence and reduce the objective function value, second-order information is used. The Hessian of the compliance is approximated with a positive and semi-definite matrix based on its structure. This approximation is used to properly define the sub-problems. In addition, the mathematical structure of the Hessian is exploited to reduce both computational time and memory usage. The proposed TopSQP is a robust method that obtains optimized designs using few iterations. However, its main drawback is the computational time spent in solving the inequality quadratic sub-problem. These sub-problems can be solved using several algorithms, for instance, interior point methods. Interior point algorithms solve a sequence of large-scale saddle-point systems in which they spend most of the computational effort. Chapter 9 investigates efficient

iterative methods to solve the saddle-point problem arising in interior point methods for the minimum compliance problem. Particularly, Krylov sub-space methods are combined with different preconditioners such as multigrid cycles, to develop an efficient and robust large-scale method for the KKT system. The interior point method presented in Chapter 9, namely TopIP, solves large-scale 3D minimum compliance problems.

In parallel with the implementation of state-of-the-art nonlinear optimization solvers, a comparative study of continuation methods is performed. Continuation methods are considered one of the best alternatives to decrease the chances of ending in a local minimum. Chapter 7 assesses the effectiveness of the continuation technique. This technique improves the final designs, but the computational time considerably increases. Thus, a new alternative to be able to produce better designs without consuming so much iterations (and therefore computational time) is proposed. The automatic penalty continuation approach includes the material penalization parameter as an extra variable in the optimization problem. This new formulation of the problem reduces not only the objective function value, but also the number of iterations.

The remainder of the chapter elaborates in more detail the main conclusions and contributions of the articles collected in the thesis. Furthermore, some recommendations for future work are gathered at the end of the chapter.

5.1 Contributions and conclusions

Benchmarking optimization solvers for structural topology optimization

Methods such as MMA and OC were specially implemented to be used in optimal design. They are extensively used in commercial software and in academic research codes. Since topology optimization problems are defined as nonlinear problems, we strongly believe that the use of the state-of-the-art optimization software could (and should) outperform these classical and first-order structural optimization solvers.

The goal of this paper is to perform extensive numerical tests and compare structural solvers such as MMA, GCMMA, and the OC method, with existing general purpose nonlinear optimization methods, such as the interior point methods in IPOPT and MATLAB, and the sequential quadratic programming method in SNOPT. For the first time in this field, general nonlinear solvers are compared with the classical structural topology optimization solvers on a large test set of benchmark problems.

Extensive numerical results are presented using performance profiles. In the numerical optimization field, performance profiles are used to illustratively compare optimization methods and formulations. This tool shows the results "at-a-glance" by comparing the relative ratio of performance for a certain criterion. The criteria considered to evaluate the solvers are the number of iterations, the objective function value, the number

of stiffness matrix assemblies, and the computational time. With the aim of producing representative and fair results, a large test set is defined, gathering 225 minimum compliance and minimum volume problems, and 150 compliance mechanism design problem instances.

The numerical experiments show that the interior point solver IPOPT applied to the SAND formulation (in which the exact second-order information is used), outperforms MMA and GCMMA both in terms of the objective function value and the number of function evaluations. In addition, SNOPT (SQP solver) in the nested formulation requires very few iterations.

The article motivates the investigation of general nonlinear solvers such as interior point and sequential quadratic programming methods for specific topology optimization purposes. These results emphasize the need of using second-order information to reduce the objective function values and improve the convergence rate.

Automatic penalty continuation in structural topology optimization

Structural topology optimization problems are commonly defined based on material interpolation schemes. The design variables of the discretized problem can take any value between zero and one. The topology optimization problem becomes nonconvex when the intermediate densities are penalized with techniques such as SIMP and RAMP. A continuation strategy in the material penalization parameter is often used to avoid ending in a poor local optimum. This approach solves a sequence of optimization problems with different material penalization parameter values. The value of the parameter is gradually increasing, and the solution of the optimization solver is used as starting point for the next optimization problem.

The purpose of this article can be divided in two. Firstly, a benchmarking study of representative continuation methods and the classical formulation in topology optimization problems is performed. Based on the previous article, the results are collected using performance profiles and the test set of 225 minimum compliance and 150 compliant mechanism design problems. Indeed, the numerical experiments reflect better performance of continuation methods for the objective function value. However, the number of iterations and the computational time are remarkably large.

Based on these results, the second part of the article proposes an automatic continuation method. Instead of solving a sequence of optimization problems, the automatic continuation approach solves one problem in which the penalization parameter is considered as a new variable. Thus, the design and the penalization parameter simultaneously change. This new formulation includes only one extra nonlinear constraint to enforce the increase of the penalization parameter.

Since both the objective function and the number of iterations decrease compared

to classical formulations, the automatic continuation approach is considered a very good alternative to continuation methods.

An efficient second-order SQP method for structural topology optimization

The use of second-order optimization solvers such as interior point and sequential quadratic programming methods has not been commonly adopted by the structural optimization community. Nevertheless, the results of the first article motivate the introduction of second-order optimization methods in the field.

A special-purpose SQP method, namely TopSQP, is proposed for the minimum compliance problem. It contains two phases, an inequality and an equality quadratic phase. The inequality constrained convex quadratic sub-problem estimates the set of active constraints. The equality constrained quadratic sub-problem promotes fast convergence. In both phases, the TopSQP uses second-order information.

Since the Hessian of the compliance is dense, and for some designs indefinite, an approximate positive semi-definite Hessian is defined. It is obtained by removing the potentially nonconvex part from the exact Hessian. Moreover, the sub-problems are reformulated based on their specific mathematical structure, avoiding the storage of the Hessian and consequently, significantly improving the efficiency of the method.

The performance of the proposed TopSQP method is compared with GCMMA, SNOPT, and IPOPT using the same test set of 225 medium-sized topology optimization problem instances defined in Chapter 6. Performance profiles confirm that the use of information based on the exact Hessian is decisive to produce good optimized designs (with low KKT error). TopSQP obtains better objective function values and uses fewer iterations than classical first-order structural optimization solvers. On the other hand, the proposed method demands a lot of computational time. In particular, most of the time is spent in the solution of the inequality quadratic sub-problem. Efficient solvers must be developed for solving this type of large-scale problems.

Solving large-scale structural topology optimization problems using a second-order interior point method

Based on the previous studies, we conclude that the use of second-order information reduces the number of function evaluations at the expense of increasing the computational time. The computational effort of some of these methods is principally focused on the solution of large-scale indefinite linear systems. Thus, the cost of second-order methods can be reduced to the cost of solving linear systems.

This article implements and develops an efficient iterative method integrated in an interior point method for large-scale minimum compliance problems. The proposed inte-

rior point method, namely TopIP, is based on an adaptive strategy, in which the barrier parameter is updated every iteration. The most expensive step in interior point methods is the computation of the search direction. The proposed iterative method solves the KKT system by combining some of the state-of-the-art iterative strategies, such as Krylov sub-space methods, block preconditioners, and multigrid methods. The KKT system contains the Hessian of the compliance. The same approximation and reformulation as in Chapter 8 is done to formulate the saddle-point problem with only sparse matrices.

Large-scale 3D minimum compliance problems are presented in the numerical experiments. The results show the robustness of the proposed iterative method. The number of iterative iterations remains constant along the optimization process. Additionally, TopIP converges to good designs using, in general, less than 100 iterations. Problems with more than three million degrees of freedom can now be solved using second-order methods.

General contributions and impact of the thesis

The most important contributions from our point of view, are gathered in the following points:

- Introduction of performance profiles in the topology optimization field.
- Extensive benchmarking study of the performance of classical structural optimization solvers and general nonlinear optimization methods in structural topology optimization problems.
- Reliable results concluding that second-order information is decisive to produce accurate and good designs and improve the convergence rate.
- Introduction, implementation, and benchmarking of an automatic continuation approach to reduce the chances of ending in local minima. It improves the optimized designs and at the same time reduces the number of function evaluations.
- Definition of a positive semi-definite approximate Hessian of the compliance based on its specific mathematical structure.
- Implementation and benchmarking of an efficient sequential quadratic programming method for the minimum compliance problem.
- Reformulation of quadratic sub-problems in the TopSQP method to reduce the computational effort based on the specific mathematical structure of the problem.
- Implementation and benchmarking of an interior point method for the minimum compliance problem.
- Implementation and benchmarking of an efficient and robust iterative method for solving large-scale structural topology optimization problems.

5.2 Future work

As far as solving topology optimization problems with second-order methods concerns, there are several challenges to deal with. In this dissemination we discuss how to approximate the Hessian and how to efficiently solve QP problems and saddle-point systems. The thesis thus provides a step towards the introduction and development of second-order methods in the field. There is, however, plenty of room to investigate new methods and approximations for many other problems and formulations. For instance, the methods can be implemented to handle more and possibly nonlinear constraints, such as stress constraints, or to efficiently solve SAND formulations, among other possibilities. This section presents several topics for possible future research.

The discussion is limited to the minimum compliance problem in the nested formulation. Future work must be done to answer the above questions for other topology optimization problems, such as compliant mechanism design problems in which the Hessian is more complicated to approximate.

The proposed formulation uses the SIMP material penalization approach and a density filter as regularization technique. In addition, the domain is discretized using a standard finite element model, generally used for academic purposes. The final implementation of both TopSQP and TopIP must be able to handle more general meshes as well as different regularization techniques, such as PDE filters and projection filters. For instance, the proposed solvers could handle ABAQUS or ANSYS (finite element analysis software) input files to solve more complex and industrial problems.

Regarding the benchmarking study, the solvers are compared using a medium-size test set of problems. Further investigations need to be done to assess the performance of the optimization methods in a large-scale test set in order to obtain more reliable results for practical applications. The size of the problems may significantly affect the performance of the solvers. In addition, this work could answer the question of which is the best formulation of the problem (SAND or nested) for very large-scale problems. Moreover, the methods can be tested for difficult starting points and design domains to study their robustness and convergence rate.

It seems particularly interesting to extend the solvers to be able to handle nonlinear constraints, and thus, to apply the proposed methods to real and practical applications. However, infeasibility control techniques are needed since the considered problem (minimum compliance in the nested form) is bounded and feasible, and neither TopSQP nor TopIP study how to deal with infeasible and unbounded problems. Moreover, there is plenty of room to improve the line search strategy implemented in the solvers, as well as the choice of a merit function, the penalty parameter update method, and the barrier parameter update scheme (for TopIP). In addition, the mentioned nonlinear methods can be tested for structural topology optimization problems using also trust region methods

and filter techniques.

Issues with the scaling of the problem were observed throughout the four articles. The scaling of the problem was controlled with the Young's modulus parameters E_1 and E_v. However, the Young's modulus contrast (E_1/E_v) established in the thesis is smaller than the commonly observed in the literature. Further investigation should be done to allow different Young's moduli values.

Hopefully, the promising results presented in the thesis can stimulate further research towards the development of more efficient and fast iterative methods that can facilitate the solution of very large-scale problems. For instance, the investigation of inexpensive and robust smoother functions may be interesting to develop a multigrid method for solving the saddle-point system in interior point methods, and thus, reduce the computational time of TopIP. Moreover, the iterative method needs to be extended to handle unstructured meshes. Therefore, the geometric multigrid cycle must be replaced by algebraic multigrid methods.

Additionally, TopSQP and TopIP are written in MATLAB without the use of parallel linear algebra libraries such as PETSc or ScaLAPACK. The code must be improved and parallelized in order to solve very large-scale problem efficiently. Moreover, an easy and user-friendly interface should be prepared to be able to use them for research and commercial purposes.

In parallel with these improvements, the solvers should be extended to include the automatic continuation approach, to reduce the objective function value even more. Ultimately, TopIP can be introduced in TopSQP for solving the IQP sub-problem. This combination might help to produce good and fast 3D large-scale optimized designs.

Based on the numerical experiments in this thesis, and extending the code with the mentioned suggestions, we are hopeful that the proposed methods can be accepted in the structural topology optimization community.

Bibliography

[1] N. Aage, E. Andreassen, and B. S. Lazarov. Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization*, 51(3):565–572, 2014.

[2] L. Ambrosio and G. Buttazzo. An optimal design problem with perimeter penalization. *Calculus of Variations and Partial Differential Equations*, 1(1):55–69, 1993.

[3] O. Amir, N. Aage, and B. S. Lazarov. On multigrid-CG for efficient topology optimization. *Structural and Multidisciplinary Optimization*, 49(5):815–829, 2014.

[4] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[5] J. S. Arora and Q. Wang. Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization*, 30(4):251–272, 2005.

[6] R. Balamurugan, C. V. Ramakrishnan, and N. Singh. Performance evaluation of a two stage adaptive genetic algorithm (TSAGA) in structural topology optimization. *Applied Soft Computing*, 8(4):1607–1624, 2008.

[7] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):192–202, 1989.

[8] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

[9] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9–10):635–654, 1999.

[10] M. P. Bendsøe and O. Sigmund. *Topology optimization: Theory, methods and applications*. Springer, 2003.

[11] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei. A comparative study of large-scale nonlinear optimization algorithms. Technical Report ORFE-01-04, Operations Research and Financial Engineering, Princeton University, 2002.

[12] M. Benzi, G . H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2004.

[13] K. U. Bletzinger. Extended method of moving asymptotes based on second-order information. *Structural Optimization*, 5(3):175–183, 1993.

[14] P. T. Boggs and J. W. Tolle. A family of descent functions for constrained optimization. *SIAM Journal on Numerical Analysis*, 21(6):1146–1161, 1984.

[15] P. T. Boggs and J. W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

[16] T. Borrvall and J. Petersson. Large-scale topology optimization in 3D using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6201–6229, 2001.

[17] A. Borzì and V. Schulz. Multigrid methods for PDE optimization. *SIAM Review*, 51(2):361–395, 2009.

[18] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[19] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM: Control, Optimization and Calculus of Variations*, 9:19–48, 2003.

[20] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2010.

[21] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, 3rd edition, 2000.

[22] M. Bruggi and P. Duysinx. Topology optimization for minimum weight with compliance and stress constraints. *Structural and Multidisciplinary Optimization*, 46(3):369–384, 2012.

[23] M. Bruyneel, P. Duysinx, and C. Fleury. A family of MMA approximations for structural optimization. *Structural and Multidisciplinary Optimization*, 24(4):263–276, 2002.

[24] M. Burger and R. Stainko. Phase-field relaxation of topology optimization with local stress constraints. *SIAM Journal on Control and Optimization*, 45(4):1447–1466, 2006.

[25] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[26] R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO : An Integrated Package for Nonlinear Optimization. In *Large Scale Nonlinear Optimization*, volume 83, pages 35–59. Springer, 2006.

[27] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 1987.

[28] A. R. Conn, N. I. M. Gould, and P. L. Toint. A globally convergent Augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[29] A. R Conn, N. I. M Gould, and P. L. Toint. *Lancelot: A FORTRAN Package for Large-Scale Nonlinear Optimization (Release A)*. Springer-Verlag, 1992.

[30] R. D. Cook. *Finite element modelling for stress analysis*. John Wiley & Sons, 1995.

[31] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt. *Concepts and applications of finite element analysis*. John Wiley & Sons, 4th Edition, 2002.

[32] F. E. Curtis and J. Nocedal. Flexible penalty functions for nonlinear constrained optimization. *IMA Journal of Numerical Analysis*, 25(4):749–769, 2008.

[33] B. Dacorogna. *Introduction to the calculus of variations*. Imperial College Press, 2004.

[34] J. D. Deaton and R. V. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.

[35] S. R. Deepak, M. Dinesh, D. K. Sahu, and G. K. Ananthasuresh. A comparative study of the formulations and benchmark problems for the topology optimization of compliant mechanisms. *Journal of Mechanisms and Robotics*, 1(1):1–8, 2009.

[36] A. Diaz and O. Sigmund. Checkerboard patterns in layout optimization. *Structural Optimization*, 10(1):40–45, 1995.

[37] N. P. Dijk, K. Maute, M. Langelaar, and F. Keulen. Level set methods for structural topology optimization: A review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013.

[38] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[39] W. S. Dorn, R. E. Gomory, and H. J. Greenberg. Automatic design of optimal structures. *Journal De Mecanique*, 3(1):25–52, 1964.

[40] T. Dreyer, B. Maar, and V. Schulz. Multigrid optimization in applications. *Journal of Computational and Applied Mathematics*, 120(1-2):67–84, 2000.

[41] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89(3):507–541, 1996.

[42] H. A. Eschenauer and N. Olhoff. Topology optimization of continuum structures: A review. *Applied Mechanics Reviews*, 54(4):331–390, 2001.

[43] L. F. Etman, A. A. Groenwold, and J. E. Rooda. First-order sequential convex programming using approximate diagonal QP subproblems. *Structural and Multidisciplinary Optimization*, 45(4):479–488, 2012.

[44] R. P. Fedorenko. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, 4(3):227–235, 1964.

[45] R. Fletcher. A general quadratic programming algorithm. *Journal of the Institute of Mathematics and Its Applications*, 6(4):76–91, 1970.

[46] R. Fletcher and S. Leyffer. User manual for filterSQP. Technical Report NA/181, University of Dundee Numerical Analysis Report, 1998.

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

[48] C. Fleury. CONLIN: An efficient dual optimizer based on convex approximation concepts. *Structural Optimization*, 1(2):81–89, 1989.

[49] C. Fleury. Efficient approximation concepts using second order information. *International Journal for Numerical Methods in Engineering*, 28(9):2041–2058, 1989.

[50] C. Fleury. First and second order convex approximation strategies in structural optimization. *Structural Optimization*, 1(1):3–10, 1989.

[51] A. Forsgren. Inertia-controlling factorizations for optimization algorithms. *Applied Numerical Mathematics*, 43(1-2):91–107, 2002.

[52] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

[53] A. Forsgren and W. Murray. Newton methods for large-scale linear inequality-constrained minimization. *SIAM Journal on Optimization*, 7(1):162–176, 1997.

[54] P. E. Gill and W. Murray. Numerically stable methods for quadratic programming. *Mathematical Programming*, 14(3):349 – 372.

[55] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large -Scale Constrained Optimization. *SIAM Journal on Optimization*, 47(4):99–131, 2005.

[56] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User's guide for NPSOL 5.0: A fortran package for nonlinear programming. Technical Report SOL 86-1, Systems Optimization Laboratory, Department of Opertations Research, Stanford University, 1998.

[57] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press, 1981.

[58] P. E. Gill and D. P. Robinson. A globally convergent stabilized SQP method. *SIAM Journal on Optimization*, 23(4):1983–2010, 2013.

[59] P. E. Gill, M. A. Saunders, and E. Wong. On the performance of SQP methods for nonlinear optimization. Technical Report CCoM 15-1, Department of Mathematics, University of California, San Diego, 2015.

[60] P. E. Gill and E. L. Wong. Convexification schemes for SQP methods. Technical Report CCoM 14-6, Center for Computational Mathematics, University of California, San Diego, 2014.

[61] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 32(1):90–99, 1985.

[62] M. Griebel, D. Oeltz, and M. A. Schweitzer. An Algebraic Multigrid Method for Linear Elasticity. *SIAM Journal on Scientific Computing*, 25(2):385–407, 2003.

[63] R. B. Haber, C. S. Jog, and M. P. Bendsøe. A new approach to variable-topology shape design using a constraint on perimeter. *Structural optimization*, 11(1–2):1–12, 1996.

[64] W. Hackbusch. *Multigrid Methods and Applications*. Springer, 1985.

[65] N. W. Henderson. *Arc Search Methods for Linearly Constrained Optimization*. PhD thesis, Stanford University, 2012.

[66] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[67] N. J. Higham and S. H. Cheng. Modifying the inertia of matrices arising in optimization. *Linear Algebra and its Applications*, 275–276:261–279, 1998.

[68] R. H. W. Hoppe, C. Linsenmann, and S. I. Petrova. Primal-dual Newton methods in structural optimization. *Computing and Visualization in Science*, 9(2):71–87, 2006.

[69] R. H. W. Hoppe and S. I. Petrova. Primal-dual Newton interior point methods in shape and topology optimization. *Numerical Linear Algebra with Applications*, 11(56):413–429, 2004.

[70] R. H. W. Hoppe, S. I. Petrova, and V. Schulz. Primal-dual Newton-type interior-point method for topology optimization. *Journal of Optimization Theory and Application*, 114(3):545–571, 2002.

[71] C. F. Hvejsel and E. Lund. Material interpolation schemes for unified topology and multi-material optimization. *Structural and Multidisciplinary Optimization*, 43(6):811–825, 2011.

[72] Kaustuv. *IPSOL: An Interior Point Solver for Nonconvex Optimization Problems*. PhD thesis, Stanford University, 2009.

[73] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, 1995.

[74] U. Kirsch. *Structural Optimization. Fundamentals and Applications*. Springer, 1993.

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

[76] L. Krog, A. Tucker, and G. Rollema. Application of topology, sizing and shape optimization methods to optimal design of aircraft components. Technical Report BS99 7AR, Altair Engineering, Airbus. Advanced numerical simulation department, Bristol, 2011.

[77] B. S. Lazarov and O. Sigmund. Filters in topology optimization based on Helmholtz-type differential equations. *International Journal for Numerical methods in engineering*, 86(6):765–781, 2011.

[78] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[79] N. Maratos. *Exact penalty function algorithms for finite dimensional and control optimization problems*. PhD thesis, University of London, 1978.

[80] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal Optimization*, 2(4):575–601, 1992.

[81] B. Metsch. *Algebraic multigrid (AMG) for saddle point systems.* PhD thesis, PhD in Mathematics, Faculty of Natural Sciences of Rheinischen Friedrich-Wilhelms-Universität Bonn, 2013.

[82] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20(3):286–307, 1994.

[83] W. Murray. Newton-type Methods. Technical report, Deparment of Management Science and Engineering, Stanford University, Stanford, CA, 2010.

[84] B. A. Murtagh and M. A. Saunders. MINOS 5.5 User's Guide. Technical Report SOL 83-20R, Stanford University Systems Optimization Laboratory, Department of Operations Research, 1998.

[85] J. Nocedal. Updating Quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[86] J. Nocedal, R. Wächter, and R. A. Waltz. Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization*, 19(4):1674–1693, 2009.

[87] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, 1999.

[88] E. Norberg and S. Lövgren. Topology optimization of vehicle body structure for improved ride and handling. Technical Report SRN LIu-IEI-TEK-A-11/01158-SE, Deparment of Management and Engineering, Linköpings universitet, 2011.

[89] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[90] J. Petersson and O. Sigmund. Slope constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 41(8):1417–1434, 1998.

[91] W. Prager and J. E. Taylor. Problems of optimal structural design. *Applied Mechanics*, 35(1):102–106, 1968.

[92] G. I. N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37(3):217–237, 2008.

[93] G. I. N. Rozvany and M. Zhou. The COC algorithm, part I: Cross-section optimization or sizing. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):281–308, 1991.

[94] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[95] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[96] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[97] O. Sardan, V. Eichhorn, D. H. Petersen, S. Fatikow, O. Sigmund, and P. Bøggild. Rapid prototyping of nanotube-based devices using topology-optimized microgrippers. *Nanotechnology*, 19(49):1–9, 2008.

[98] P. E. Saylor and D. C. Smolarski. Implementation of an adaptive algorithm for Richardson's method. *Linear Algebra and its Applications*, 154-156:615–646, 1991.

[99] V. Schulz and G. Wittum. Transforming smoothers for PDE constrained optimization problems. *Computing and Visualization in Science*, 11(4–6):207–219, 2008.

[100] O. Sigmund. *Design of material structures using topology optimization*. PhD thesis, Department of Solid Mechanics, Technical University of Denmark, 1994.

[101] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, 25(4):492–526, 1997.

[102] O. Sigmund. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica*, 25(2):227–239, 2009.

[103] O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[104] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(2):68–75, 1998.

[105] R. Simon. *Multigrid solvers for saddle point problems in PDE-constrained optimization*. PhD thesis, Johannes Kepler Universität, 2008.

[106] J. Sokolowski and A. Zochowski. On the topological derivative in shape optimization. *SIAM Journal on Control Optimization*, 37(4):1251–1272, 1999.

[107] R. Stainko. *Advanced multilevel techniques to topology optimization*. PhD thesis, Johannes Kepler Universität, 2006.

[108] M. Stolpe and M. P. Bendsøe. Global optima for the Zhou-Rozvany problem. *Structural and Multidisciplinary Optimization*, 43(2):151–164, 2011.

[109] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, 2001.

[110] L. L. Stromberg, A. Beghini, W. F. Baker, and G. H. Paulino. Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. *Structural and Multidisciplinary Optimization*, 43(2):165–180, 2011.

[111] A. Sutradhar, G. H. Paulino, M. J. Miller, and T. H. Nguyen. Topological optimization for designing patient-specific large craniofacial segmental bone replacements. *PNAS*, 107(30):13222 – 13227, 2010.

[112] K. Svanberg. The method of moving asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

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

[114] J. E. Taylor and M. P. Rossow. Optimal truss design based on an algorithm using optimality criteria. *International Journal of Solids and Structures*, 13(10):913 – 923, 1977.

[115] A. L. Tits, A. Wächter, and S. Bakhtiari. A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. *SIAM Journal on Optimization*, 14(1):173–199, 2003.

[116] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13(1-3):231–252, 1999.

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

[118] C. Wagner. Introduction to algebraic multigrid. Technical Report http://perso.uclouvain.be/alphonse.magnus/num2/amg.pdf, Course notes of an algebraic multigrid course at the University of Heidelberg, 1998/99, 1998.

[119] B. Wang and D. Pu. Flexible penalty functions for SQP algorithm with additional equality constrained phase. Technical report, Proceedings of the 2013 International Conference on Advance Mechanic System, China, September 25-27, 2013.

[120] H. Wang, Q. Ni, and H. Liu. A new method of moving asymptotes for large-scale linearly equality-constrained minimization. *Acta Mathematicae Applicatae Sinica*, 27(2):317–328, 2011.

[121] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2):227–246, 2003.

[122] S. Wang, K. Tai, and M. Y. Wang. An enhanced genetic algorithm for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(1):18–44, 2006.

[123] R. Weiss. A theoretical overview of Krylov subspace methods. *Applied Numerical Mathematics*, 9(1):207–233, 1995.

[124] P. Wesseling. *An introduction to multigrid methods*. John Wiley & Sons, 1992.

[125] P. Wesseling and C. W. Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128:311–334, 2001.

[126] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. *Numerische Mathematik*, 57(3):15–38, 1990.

[127] Y. M. Xie and G. P. Steven. A simple evolutionary procedure for structural optimization. *Computers and Structures*, 49(5):885 – 896, 1993.

[128] Y. M. Xie and G. P. Steven. *Evolutionary structural optimization*. Springer, 1997.

[129] W. H. Zhang, C. Fleury, P. Duysinx, V. H. Nguyen, and I. Laschet. A generalized method of moving asymptotes (GMMA) including equality constraints. *Structural optimization*, 12(2-3):143–146, 1996.

[130] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):309–336, 1991.

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

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

[133] W. Zulehner. A Class of Smoothers for Saddle Point Problems. *Computing*, 65(3):227–246, 2000.

Part II

Articles

6

Article I : Benchmarking optimization solvers for structural topology optimization

Published online the 17 May 2015:
Rojas-Labanda, S. and Stolpe, M.: Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization* (2015). DOI : 10.1007/s00158-015-1250-z.

Benchmarking optimization solvers for structural topology optimization*

Susana Rojas-Labanda[+] and Mathias Stolpe[+]

[+]DTU Wind Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark. E-mail: srla@dtu.dk , matst@dtu.dk

Abstract

The purpose of this article is to benchmark different optimization solvers when applied to various finite element based structural topology optimization problems. An extensive and representative library of minimum compliance, minimum volume, and mechanism design problem instances for different sizes is developed for this benchmarking. The problems are based on a material interpolation scheme combined with a density filter.

Different optimization solvers including Optimality Criteria (OC), the Method of Moving Asymptotes (MMA) and its globally convergent version GCMMA, the interior point solvers in IPOPT and FMINCON, and the sequential quadratic programming method in SNOPT, are benchmarked on the library using performance profiles. Whenever possible the methods are applied to both the nested and the Simultaneous Analysis and Design (SAND) formulations of the problem.

The performance profiles conclude that general solvers are as efficient and reliable as classical structural topology optimization solvers. Moreover, the use of the exact Hessians in SAND formulations, generally produce designs with better objective function values. However, with the benchmarked implementations, solving SAND formulations consumes more computational time than solving the corresponding nested formulations.

Keywords: Topology optimization, Benchmarking, Nonlinear optimization, Optimization methods

Mathematics Subject Classification 2010: 74P05, 74P15, 90C30, and 90C90

*This research is funded by the Villum Foundation through the research project Topology Optimization - The Next Generation (NextTop).

1 Introduction

Structural topology optimization problems determine the optimal distribution of material in a given design domain that minimizes an objective function under certain constraints. The continuum design problem is often discretized using finite elements and a design variable is associated with each finite element. A detailed description of this approach for topology optimization problems can be found in e.g [6]. Since topology optimization problems are examples of large-scale nonlinear optimization problems, it is possible to solve them using a large variety of different numerical optimization methods.

Some first-order methods which are well-established in the topology optimization community are the Method of Moving Asymptotes (MMA) [43] and [54], its globally convergent version GCMMA [44], and the Convex Linearization (CONLIN) method [22]. However, general methods such as primal-dual interior point [23] and Sequential Quadratic Programming (SQP) [8] methods, are also applicable to these optimization problems.

The main purpose of this article is to assess combinations of optimization methods and formulations in topology optimization problems to see their performance and study if they are efficient and reliable for this class of problems. The interior point methods implemented in IPOPT [47] and MATLAB's FMINCON [45], and the SQP method in SNOPT [24], are compared with the commonly used topology optimization methods Optimality Criteria (OC) [37], [53], and [2], MMA, and GCMMA.

Three classes of structural topology optimization problems, which often appear in the literature, are taken into account. We study minimum compliance, minimum volume, and compliant mechanism design problems as described in e.g. [6]. It is suggested in [41] that in order to generalize the results and have reliable conclusions, an algorithm must be tested using several problems. It is expected to obtain differences between the formulations of the problem as well as differences in the performance of the methods for the three classes of problems.

In order to generalize the results, several equivalent problem formulations are evaluated. Both the nested approach in which the displacements are functions of the design variables, and the Simultaneous Analysis and Design (SAND) approach in which the displacements and design are independent variables, are used. SAND formulations have the advantages that both gradients and Hessians of the objective and constraints functions are easily computed without the need of solving the equilibrium equations and adjoint equations. Furthermore, the Hessian of the Lagrangian is, in general, a sparse matrix. The apparent disadvantage is the potentially significant increase in the size of the problems, both in terms of variables and constraints. Nested formulations have the advantage of reduced size at the expense of more complicated and more expensive sensitivity analysis. Moreover, second-order information is, in general,

computationally and memory wise much too expensive, and can thus, disqualify fast second-order methods. More details of the advantages and disadvantages of these two approaches are listed in the review [3].

Between all possible alternatives to define the discretized topology optimization problem, we formulate it using continuous design variables. The Solid Isotropic Material with Penalization (SIMP) material interpolation scheme is chosen to penalize intermediate densities values [5]. The RAMP (Rational Approximation of Material Properties) interpolation scheme [42] was also considered and implemented but it was removed in the final benchmark since the results were very similar. Finally, the density filter technique described in e.g. [12] is used to ensure existence of solutions and to avoid checker-board patterns in the final design.

We are aware of the limitations of this benchmark. Only one approach to parametrize the topology and one material interpolation scheme with a density filter is used for the material modelling and regularization. Other possibilities are the use of level sets see e.g. [50], or other mesh-independence techniques such as perimeter control [26]. We decided to focus on the performance of the solvers using one classic approach and deepen the investigation in the behaviour of the solvers in a large test set, rather than study the performance for different topology optimization formulations.

Therefore, one of the most important targets of this article, is to produce useful, clear, and fair results. The state-of-art technique for benchmarking optimization solvers, called *performance profiles* [19], is used on an illustrative and large set of topology optimization problems. A specific benchmark library has created for this study. Moreover, in Section 4.4 this library is defined in detail. In this test set, some typical 2D test examples are collected from the literature, gathering over different problem instances for minimum compliance, minimum volume and for mechanism design problems. The design domains, boundary conditions and external loads are taken from the literature such as [40] [4], [13], [2], [14], [6], [48], [39], [18], and [38].

Performance profiles have already been extensively used in [7] for a comparative study of different large-scale nonlinear optimization algorithms such as the interior point algorithm LOQO [46] and KNITRO [16], and the SQP method SNOPT. The test set used for this benchmarking came from CUTE [10] and COPS [9], two general benchmark libraries with equality and inequality constrained nonlinear problems. Performance profiles were also used in [19] where LANCELOT [17], MINOS [34], SNOPT, and LOQO are compared on the COPS benchmarking library.

Finally, this article is focused on medium-size problems since our implementation is in MATLAB and some of the solvers require, due to their particular implementations, large amounts of memory. Several articles develop techniques for solving large-scale topology optimization problems focused mostly on 3D design domains. The computational bottleneck of these problems is the computation of the solution of

large-scale linear systems such as the equilibrium equations in the nested formulation and the saddle-point system in the SAND formulations. Techniques capable of solving large-scale problems are presented in e.g. [51], [21], and [11].

The paper is organized as follows. Section 2 presents the fundamentals of the topology optimization problems and the different formulations. Section 3 reports the implementation and methods used in the benchmarking. Section 4 introduces the test set of topology optimization problems, explains the performance profiles as well as describes some aspects to consider before the benchmark. Section 5 reports the numerical results. Finally, Sections 6 and 7 conclude with the limitations of our benchmark and collect the main results, our final recommendations, and topics for future research.

2 Problem formulations

Structural topology design problems consist of obtaining a material distribution in a fixed design domain with boundary conditions and external loads.

The classical formulation of the problem is minimizing the compliance of the structure, considering a volume constraint, see e.g. [6]. However, it is also possible, and sometimes desirable, to formulate topology optimization problems as minimizing the structural volume (or mass) subject to a compliance constraint [13]. In addition, compliant mechanism design problems are included in this benchmarking study. These problems are considered in e.g. [39]. In practice, these problems are modelled by discretizing the design domain using finite elements and coupling one design variable to each element. This section describes the mathematical formulation for these optimization problems.

2.1 Minimum compliance problems

Minimizing the compliance is equivalent to maximizing the global stiffness of the structure for the given external load. The linear elastic equilibrium equation obtained by applying the finite element method to the underlying partial differential equation must be satisfied in the final design. The equilibrium equations are modelled as

$$\mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^d$ is the state variable (nodal displacements) and $\mathbf{t} \in \mathbb{R}^n$ is the design variable. Throughout this article the design variables represent relative density of the material in each finite element. Furthermore, $\mathbf{f} \in \mathbb{R}^d$ is the design independent external load, d the degrees of freedom, and n the number of elements. The stiffness matrix is $\mathbf{K}(\mathbf{t}) : \mathbb{R}^n \to \mathbb{R}^{d \times d}$, and we assume it is positive definite for all designs satisfying the variable box

constraints in order to avoid ill-conditioning i.e. $\mathbf{K}(\mathbf{t}) \succ 0$ for all $\mathbf{t}$ such that $\mathbf{0} \leq \mathbf{t} \leq \mathbf{1}$. A small positive value is included in the definition of the stiffness matrix to avoid singularity for those densities equal to zero (cf. below).

There are different ways of modelling the optimization problem. In the first approach, the compliance is minimized over the design variables (density) and the state variables (displacements). They are considered as independent variables. In addition, the equilibrium equations are explicitly included as equality constraints. Thus, the formulation (P_S^c) is commonly called SAND (Simultaneous Analysis and Design), see e.g [3]. The discrete form of the problem is

$$\begin{aligned} \underset{\mathbf{t},\mathbf{u}}{\text{minimize}} \quad & \mathbf{f}^T\mathbf{u} \\ \text{subject to} \quad & \mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{a}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{aligned} \tag{P_S^c}$$

The relative volume of the elements is defined with $\mathbf{a} \in \mathbb{R}^n$ with $a_i > 0$. Finally, $0 < V \leq 1$ is the volume fraction upper limit. The problem (P_S^c) is defined with a linear objective function, nonlinear equality constraints and a linear inequality constraint. Topology optimization problems, such as (P_S^c) are generally characterized as nonconvex problems.

An alternative approach is to model the objective as a nonlinear function. The external load $\mathbf{f}$ is defined as $\mathbf{K}(\mathbf{t})\mathbf{u}$, by the equilibrium equation (1). This formulation, which is equivalent to (P_{SNL}^c), is called SAND nonlinear (SANDNL),

$$\begin{aligned} \underset{\mathbf{t},\mathbf{u}}{\text{minimize}} \quad & \mathbf{u}^T\mathbf{K}(\mathbf{t})\mathbf{u} \\ \text{subject to} \quad & \mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{a}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{aligned} \tag{P_{SNL}^c}$$

The name is given since the objective function is nonlinear and nonconvex, in general, due to the definition of the stiffness matrix. In the numerical experiments we do not include this formulation. The performance of the solvers is very similar to the SAND formulation (see Section 5).

The number of constraints and variables can be reduced if the problem is formulated using only the design variable. The displacement caused by the force is determined by the equilibrium equation (1),

$$\mathbf{u}(\mathbf{t}) = \mathbf{K}^{-1}(\mathbf{t})\mathbf{f}. \tag{2}$$

The state problem is thus solved during the objective function evaluation, and the minimum compliance problem can equivalently be written in the nested formulation

$$\begin{array}{ll} \underset{\mathbf{t}}{\text{minimize}} & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \qquad (\text{or } \mathbf{f}^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{f}) \\ \text{subject to} & \mathbf{a}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{array} \tag{P_N^c}$$

The formulation (P_N^c) has only linear inequality constraints with a nonlinear, and generally nonconvex, objective function, and it is the classic formulation in the topology optimization field, see e.g. [6].

The main advantage of the nested approach is that the number of variables and constraints are much smaller than in the SAND approach. However, the evaluation of the objective, gradient, and Hessian functions, is more costly. In these functions, the inverse of the stiffness matrix is involved which is computational expensive.

2.2 Minimum volume problems

Structural topology optimization problems can also be formulated as the minimization of the volume of the structure with a restriction on the compliance [13]. Similar to the minimum compliance problem (P_S^c), the SAND formulation for minimum volume (P_S^w) is

$$\begin{array}{ll} \underset{\mathbf{t},\mathbf{u}}{\text{minimize}} & \mathbf{a}^T\mathbf{t} \\ \text{subject to} & \mathbf{K}(\mathbf{t})\ \mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{f}^T\mathbf{u} \leq C \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{array} \tag{P_S^w}$$

where $C > 0$ is a given upper bound of the compliance. In the nested formulation (P_N^w), the equilibrium equation is satisfied in the nonlinear inequality constraint (compliance constraint)

$$\begin{array}{ll} \underset{\mathbf{t}}{\text{minimize}} & \mathbf{a}^T\mathbf{t} \\ \text{subject to} & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \leq C \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{array} \tag{P_N^w}$$

Finally, the SANDNL formulation (P_{SNL}^w) has nonlinear equality and inequality constraints.

$$\begin{array}{ll} \underset{\mathbf{t},\mathbf{u}}{\text{minimize}} & \mathbf{a}^T\mathbf{t} \\ \text{subject to} & \mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{u}^T\mathbf{K}(\mathbf{t})\mathbf{u} \leq C \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{array} \tag{P_{SNL}^w}$$

For minimum volume problems, the objective function is linear and the nonlinearity appears in the constraints.

2.3 Compliant mechanism design problems

Compliant mechanism design consists of building a mechanism with some flexible elements in order to gain mobility. The goal is to maximize the displacement where the output spring is located given an input force ($\mathbf{f}_{\text{in}}$) and input and output spring stiffness (k_{in}, k_{out}). The objective function is defined by the use of a unit length vector ($\mathbf{l}$) with zeros in all the degrees of freedom except in the output degree of freedom. We assume a linear model for the equilibrium equation as in the minimum compliance or minimum volume even if there are several limitations with this approach [6]. The nested formulation of the considered mechanism design problem is

$$\begin{aligned} \underset{\mathbf{t}}{\text{maximize}} \quad & \mathbf{l}^T\mathbf{u}(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{a}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{aligned} \tag{P_N^m}$$

where the objective function is nonlinear (and nonconvex) and has only one linear inequality constraint. The SAND formulation is defined with a linear objective function, linear inequality constraint and nonlinear equality constraints,

$$\begin{aligned} \underset{\mathbf{t},\mathbf{u}}{\text{maximize}} \quad & \mathbf{l}^T\mathbf{u} \\ \text{subject to} \quad & \mathbf{K}(\mathbf{t})\mathbf{u} - \mathbf{f} = \mathbf{0} \\ & \mathbf{a}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{aligned} \tag{P_S^m}$$

2.4 Topology optimization approaches

The goal of the optimization process is often to obtain a close-to solid or void design representing both the topology and the shape of a structure. Material interpolation models are very popular in topology optimization to convert the 0-1 problem into a nonlinear continuous problem. These models usually use penalization schemes to find good designs [6]. The SIMP material interpolation scheme is one of the most common approaches. In this model, the density $\mathbf{t}$ is replaced by a power law, see [5] and [6] among many others.

On the other hand, it is well-known that the continuum problem has a lack of solutions in general [6]. Another important difficulty in topology optimization problems is the appearance of checker-boards. The SIMP approach does not resolve these issues. However, an efficient technique to avoid these problems is the use of filters, that ensures regularity and existence of a solutions, see e.g. [12]. Specifically, the density filter is considered in this article. Given one element, its density variable depends on a weighted

average over the neighbours in a radius $r_{\min}$.

$$\begin{aligned} \tilde{t}_e &= \frac{1}{\sum_{i \in N_e} \overline{H}_{ei}} \sum_{i \in N_e} \overline{H}_{ei} t_i \\ \overline{H}_{ei} &= \max(0, r_{\min} - \Delta(e,i)) \end{aligned} \tag{3}$$

Here, $\tilde{t}_e$ is the transformed density variable of element e, N_e the set of elements for which the distance to element i (defined by $\Delta(e,i)$) is smaller than the filter radius $r_{\min}$. In practice, this concrete value is taken from [2], $r_{\min} = 0.04 L_x$, where L_x is the length of the design domain in the x direction.

The modified stiffness matrix using density filter and SIMP penalization is

$$\mathbf{K}(\mathbf{t}) = \sum_{e=1}^{n} (E_v + (E_1 - E_v)\tilde{t}_e^p)\mathbf{K}_e \tag{4}$$

where $p \geq 1$ and $E_v > 0$ and $E_1 \gg E_v$ are Young's modulus of the "void" and solid material, respectively. The parameter E_v is included to avoid ill-conditioning when the density variable is equal to zero, as previously mentioned. This formulation has proven particularly efficient in many cases [6]. More details about this interpolation is given in [4]. In practice, SIMP typically has a penalization parameter value chosen to be $p = 3$, see e.g [2] and [6]. This value is used in the numerical experiments (cf. below).

3 Implementation

This section contains all relevant information required to reproduce the numerical results. It includes a brief description of the solvers used in the benchmarking as well as the details of the implementation such as the finite elements, the stopping criteria, and the setting of the parameters.

3.1 Chosen optimization methods

Structural topology optimization problems are usually solved using sequential convex approximation methods such as the Method of Moving Asymptotes (MMA) [43] or the Globally Convergent Method of Moving Asymptotes (GCMMA) [44]. These methods were used for structural topology optimization in e.g. [41], [48], and [18], among many others.

These methods generally require one evaluation of the objective and constraint functions and their derivatives at each iteration. GCMMA includes inner iterations, making the approximations more conservative, to force global convergence to a KKT (Karush-Kuhn-Tucker) point. These methods are only applicable for optimization

problems with inequality constraints[1], which means that only the nested formulations can be solved with them.

In addition, the nested formulations can also be solved using the Optimality Criteria (OC) method [37], [53], and [6] among others. It is valid only when the gradient of the objective function is negative, thus it is not possible to implement it for minimum volume problems. In particular, we use the *88-lines* code from [2] and the *104-lines* code from [6] for the OC method for minimum compliance and compliant mechanism design problems, respectively.

However, since topology optimization are nonlinear optimization problems, it is possible to use state-of-the-art second-order solvers to solve them. Two implementations of primal-dual interior point solvers, IPOPT [47] and FMINCON [45], and a sequential quadratic programming solver SNOPT [24] version 7, are tested for topology optimization problems. IPOPT is a primal-dual interior point software library for large-scale nonlinear optimization problems that uses a line-search based on filter methods. SNOPT is also a general large-scale optimization solver that uses a sequential quadratic programming algorithm. Finally, FMINCON is a set of nonlinear optimization algorithms in MATLAB. These three optimization methods all have MATLAB interfaces, are popular, and have been extensively benchmarked in the optimization community, see e.g. [7] and [32]. Moreover, IPOPT and SNOPT have already used for certain topology optimization problems in e.g. [15] and [28]. These solvers accept both equality and inequality constraints, hence it is possible to benchmark the SAND formulation. For those solvers that can use exact Hessian (IPOPT and FMINCON), it is much simpler and cheaper to compute it in the SAND than in the nested approach. For the nested formulation, the default linear solver MUMPS [1] is used in IPOPT to compute the search direction while for the SAND formulation, MA27 is used [20]. The IPOPT version used is 3.11.1.

3.2 Finite element analysis

The design domains of the benchmark library created for this study (cf. below), are all 2D and discretized by square finite elements. It is assumed that the domain is rectangular and each element has four nodes with two degrees of freedom per node. A 2 by 2 Gaussian integration rule is used in the computation of the stiffness matrix. This Q4 interpolation of displacements is identical to that implemented in [2].

[1]There are implementations of these methods which do allow equality constraints see e.g. [52] and [49]. However, to the best of our knowledge there are no numerical results suggesting that these methods can be used to solve SAND formulation of topology optimization problems.

3.3 Stopping criteria

For a general nonlinear optimization problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & h_i(\mathbf{x}) = 0 \quad i = 1, \ldots, m \\ & g_i(\mathbf{x}) \leq 0 \quad i = 1, \ldots, t, \end{aligned} \tag{5}$$

the Lagrangian function is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{i}^{t} \mu_i g_i(\mathbf{x}), \tag{6}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the Lagrangian multipliers of the equality and inequality constraints, respectively. A primal-dual solution $(\overline{\mathbf{x}}, \overline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\mu}})$ of problem (5), should satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions [33].

$$\nabla \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0} \tag{7}$$

$$h_i(\mathbf{x}^*) = 0 \quad i = 1, \ldots, m \tag{8}$$

$$g_i(\mathbf{x}^*) \leq 0 \quad i = 1, \ldots, t \tag{9}$$

$$\mu_i^* \geq 0 \quad i = 1, \ldots, t \tag{10}$$

$$\mu_i^* g_i(\mathbf{x}^*) = 0 \quad i = 1, \ldots, t. \tag{11}$$

Equation (7) is the stationary condition, (8) and (9) are the primal feasibility condition, (10) is the dual feasibility condition, and (11) contains the complementarity condition. In practice, we assume that the KKT conditions are numerically satisfied if the Euclidean norm of the equations is lower than a given positive tolerance ω. In the same way, the primal feasibility conditions are satisfied if the Euclidean norm is lower than some given positive tolerance η. These tolerances will significantly affect the final design.

The solvers MMA, GCMMA, IPOPT, SNOPT and FMINCON have different implementations of the optimality conditions (different scalings and norms). Nevertheless, at the end of the optimization process the KKT error is measured for all of them using (7)-(11) to be consistent in the future comparison of the solvers.

The solvers will try to obtain a design until the feasibility error and the optimality conditions are lower than certain specified tolerances (ω and η, respectively). The value of ω is, in this benchmark, set differently for the first- and second-order methods[2], respectively. IPOPT, FMINCON and SNOPT are generally able to satisfy the KKT

[2]MMA and GCMMA are considered as first-order methods. IPOPT, SNOPT and FMINCON are considered as second-order methods even though for certain problem formulations, limited memory BFGS (Broyden-Fletcher-Goldfarb-Shanno) is used to approximate the Hessian of the Lagrangian.

Table 1: Parameter names, description and values of the convergence criteria.

Solver	**Parameter**	**Description**	**Value**
MMA, GCMMA	`kkt tol`	Euclidean norm of the KKT error	10^{-4}
IPOPT	`tol`	Tolerance of the NLP error	10^{-6}
SNOPT	`Major optimality tolerance`	Final accuracy of the dual variable	10^{-6}
FMINCON	`TolFun`	Tolerance on the function value	10^{-6}
OC	`change`	Difference in the design variable	10^{-4}
OC, MMA, GCMMA	`feas tol`	Euclidean norm of the feasibility error	10^{-8}
IPOPT	`constr viol tol`	Tolerance of the constraint violation	10^{-8}
SNOPT	`Major feasibility tolerance`	Tolerance of the nonlinear constraint violation	10^{-8}
FMINCON	`TolCon`	Tolerance of the constraint violation	10^{-8}

conditions with a very small tolerance, while first-order methods such as MMA and GCMMA generally require larger tolerances. This is highlighted in Table 1.

It is important to remark that the 88-line code OC solver is a special case because it does not compute estimates of the Lagrangian multipliers and does not compute the KKT error. This method stops because of the feasibility error and by the parameter called `change` [2]. This parameter measures the difference between the updated and the old design variable $||\mathbf{t}^k - \mathbf{t}^{k-1}||_\infty$.

Table 1 collects the tolerance parameter for the convergence of the solvers.

3.3.1 Termination control

We have established some maximum values to avoid the situation that the solvers run indefinitely in case they stop making progress towards a KKT point and they are unable to obtain a design with the requested accuracy. For instance, we limited the maximum number of iterations (number of sub-problems solved), the maximum number of functions evaluations (FMINCON), the maximum number of stiffness matrix assemblies (OC, MMA and GCMMA) and the maximum CPU time (IPOPT). The values for mechanism design problems are set to three times more than for minimum compliance and minimum volume due to their difficulty. Table 2 gathers the values of these parameters.

3.4 Starting points

An important aspect that could affect the performance of the methods is the starting point. For OC, MMA, GCMMA, and SNOPT the primal starting point is initialized as

Table 2: Parameter names, description and values of the other termination criteria.

Solver	**Parameter**	**Description**	**Value**
OC	`loop`	Maximum number of iterations	1,000
MMA, GCMMA	`max outer itn`	Maximum number iterations	1,000
IPOPT	`max iter`	Maximum number of iterations	1,000
SNOPT	`Major iterations limit`	Maximum number of iterations	1,000
FMINCON	`MaxIter`	Maximum number of iterations	1,000
OC, MMA, GCMMA	`max assemblies` matrix assemblies	Maximum number of stiffness	10,000
IPOPT	`max cpu time`	Maximum CPU time	48h
FMINCON	`MaxFunEvals` function evaluations	Maximum number of	10,000

an homogeneous design with $\mathbf{t}_0 = V\mathbf{e}$ where $\mathbf{e}$ is a vector of all ones. The displacements are set to $\mathbf{u}_0 = \mathbf{0}$ for the minimum compliance and compliant mechanism design problems. However, the starting points in IPOPT and FMINCON are always initialized in between the lower and upper bounds on the density variables, i.e $\mathbf{t}_0 = 0.5\mathbf{e}$ and $\mathbf{u}_0 = \mathbf{0}$. Other starting points could be a disadvantage due to the use of interior point algorithms. For minimum volume problems the starting point is chosen as $\mathbf{t}_0 = 0.5\mathbf{e}$ and $\mathbf{u}_0 = \mathbf{0}$. Other starting points for the displacements were examined but did not give better results. For instance, our numerical experiments did not indicate any difference between the initialization $\mathbf{u}_0 = \mathbf{K}^{-1}(\mathbf{t}_0)\mathbf{f}$ to the initialization $\mathbf{u}_0 = \mathbf{0}$.

3.5 Computation of compliance upper limit

For minimum compliance and compliant mechanism design problems, the volume constraint is limited by an upper bound given by a scalar value between 0 and 1, to indicate the percentage of material the user wants in the final design. For minimum volume problems the upper limit in the compliance constraint is chosen as

$$C = k(\mathbf{f}^T(\mathbf{K}^{-1}(\mathbf{t}_0)\mathbf{f})), \tag{12}$$

for some user defined constant $k \geq 1$.

3.6 Setting of the parameters

In general, it is desirable to tune the solver parameters as little as possible. However, the performance of the solvers for topology optimization problems can be improved if some default parameters are modified. The automatic scaling of the problem is turned off in both IPOPT and FMINCON solvers. Depending on the problem and the formulation

used, SNOPT performs better with different scaling option values. In practice, SNOPT in the SAND formulation performs better when there is no automatic scaling. The nested formulation works better with certain scaling and the default value is therefore used in this situation.

Moreover, the adaptive barrier update strategy suggested in [35] is used in IPOPT because it requires less iterations than the monotone update strategy, and the obtained designs are more accurate. In order to improve the performance of IPOPT, some options are activated, such as the full step size in the constraint multipliers and the use of the least square estimation for computing the constraints multipliers. FMINCON uses an interior point algorithm. The conjugate gradient method is chosen to determine how the iteration step is calculated. This is usually faster than the default value (LDL factorization).

Finally, in SNOPT the super basics limit, the iteration limit, the function precision, the line-search tolerance and the step limit are modified to be able to solve large-scale problems. In particular, the function precision parameter is significantly increased since too stringent values affect the built-in line-search termination criteria and significantly increase the number of function evaluations.

When the nested approach is used, IPOPT approximates the Hessian using a limited memory BFGS approach. The number of most recent iterations that are taken into account for the approximation is set to 25. The main reason of using BFGS is the high computational cost of the exact Hessian. FMINCON has an option where a matrix-vector multiplication can be defined. Using this feature, the time spent in this computation is much lower, and therefore, FMINCON can use the exact Hessian. The same option is not implemented in IPOPT.

Tables 3, 4 and 5 collect all the parameters tuned in IPOPT, FMINCON and SNOPT, respectively.

Table 3: Parameters tuned in IPOPT. The table contains the name of the parameter, the new value, the default value and a brief description.

Parameter	**New value**	**Default**	**Description**
`mu strategy`	adaptive	monotone	Update strategy for barrier parameter
`limited memory max history`	25	6	Maximum size of history in BFGS
`nlp scaling method`	none	gradient based	Technique for scaling the problem
`alpha for y`	full	primal	Method to determine the step size of constraint multipliers (full step size 1)
`recalc y`	yes	no	Tells the algorithm to recalculate the multipliers as least square estimates

Table 4: Parameters tuned in FMINCON. The table contains the name of the parameter, the new value, the default value and a brief description.

Parameter	**New value**	**Default**	**Description**
`Algorithm`	interior point	trust region reflective	Determine the optimization algorithm
`Subproblem Algorithm`	cg	ldl factorization	Determines how the iteration step is calculated

Table 5: Parameters tuned in SNOPT. The table contains the name of the parameter, the new value, the default value and a brief description.

Parameter	**New value**	**Default**	**Description**
`scale option`	0(SAND)	2	Scaling of the problem (2 LP, 0 no scale)
`Iteration limit`	10^6	10^4	Maximum number of minor iterations allowed in QP su-bproblem
`Major step limit`	10	2	Limits the change in x during the line-search
`Line search tolerance`	.99999	.9	Accuracy which a step length will be located along the search direction
`New superbasics limit`	10^4	99	Early termination of QP sub-problem if the number of free variables has increased since the first feasibility iteration
`Function precision`	10^{-4}	3.7×10^{-11}	Relative function precision

3.7 Scaling of the problems

Regarding the Young's modulus parameters, the benchmark library instances defined in Section 4.4, considers rather small contrast between the solid (E_1) and the void (E_v) values. The main reason is that the final design is similar to the results for large contrast but the time and iterations required to satisfy the stopping criteria is significantly reduced. Since the scope of interest in this article is the comparison of the solvers, this contrast is fixed to $E_1/E_v = 10^3$ for minimum compliance and minimum volume problems, and to $E_1/E_v = 10^2$ for compliant mechanism design problems. Our experience is that compliant mechanism design problems, in general, are more difficult to solve.

Moreover, the choices of void and solid Young's modulus values considerably modify the computational performances of all solvers. In particular, the number of iterations and the obtained accuracy are affected. Table 6 gathers the values of E_v and E_1 for each solver and class of problem.

Table 6: Young modulus's values for each solver and topology optimization problem.

Solver	**Problem**	$\mathbf{E}_1$	$\mathbf{E}_0$
IPOPT/SNOPT	Minimum compliance	10^3	1
FMINCON	Minimum compliance	10^2	10^{-1}
IPOPT/SNOPT	Minimum volume	10^4	10
FMINCON	Minimum volume	10	10^{-2}
MMA/GCMMA	Minimum compliance/volume	10	10^{-2}
OC	Minimum compliance	1	10^{-3}
All	Mechanism design	1	10^{-2}

Finally, the Poisson's ratio parameter is set to $\nu = 0.3$ as in [2]. The numerical behaviour of the solvers (except OC) is better if the inequality constraint is scaled by a factor of $\frac{1}{\sqrt{n}}$ for minimum compliance and minimum volume problems. Additionally, for IPOPT, FMINCON and SNOPT in the compliant mechanism design problems, it is scaled by $\frac{1}{n}$, while in MMA and GCMMA is by $\frac{1}{\sqrt{n}}$.

3.8 Code modifications

Some parts of the code for GCMMA and OC have been modified to obtain better results and to improve the performance. First of all, the maximum number of inner iterations in GCMMA is reduced from 50 to two. The main reason is to be less restrictive with the convex approximation of the problems. This means that the theoretical global convergence results from [44] are no longer certain to hold, but our numerical results indicate that this is a good compromise between robustness and efficiency.

In OC, the stopping criteria has been slightly changed to satisfy either the `change` parameter and the feasibility tolerance, or the maximum number of iterations or assemblies, with values given in Tables 1 and 2. Moreover, the inner OC loop, i.e. estimation of the Lagrange multiplier for the volume constraint by bisection, is modified such as that the difference of limits has to be lower than 10^{-6}. Previous values were 10^{-3} for minimum compliance problems and 10^{-4} for mechanism design problems. Finally, it is important to remark that the OC method in the 88-line code from [2] has been modified to mechanism design problems following instructions for earlier codes in [6].

4 Benchmarking

A method called *performance profiles* was proposed in [19] to compare different optimization solvers on a set of problem instances. This tool was also used in e.g [7] to compare large-scale nonlinear optimization algorithms.

One of the benefits of this tool of comparison is that it is possible to obtain a global idea of how the solvers perform.

4.1 Performance profiles

Performance profiles show the general performance of all the solvers in 2D plots. The x-axis represents the parameter τ that indicates how far a solver is from being the winner (regarding a specific criterion such as objective function or number of iterations), i.e. it describes the relative ratio of performance. The y-axis represents the percentage of problems that each solver is able to obtain a solution by a factor τ to the best solver for a concrete measure of performance. In the case where we are only interested in the number of problems where the solver is the best, the scope of interest is $\tau = 1$.

When τ is small, the performance profile shows the amount of problems the solver has performed similar to the best solver at each problem. While τ increases, the percentage of problems increases because the solver has more chances to obtain a solution. It is desirable to observe a high increase of percentage of problems for small variations of τ. Those solvers with fast growth are the most suitable because, in general, the performance of this solver for the whole test set is very close to the best solver for each concrete problem, for the specific criterion. When τ is large, it shows the chances of a solver to be able to solve any problem. Performance profiles, as explained in [19], are based on a set P of problem instances. The performance of a set of solvers S is evaluated and compared using the set. A measure of performance, m, such as the number of iterations, is defined for a given problem p and a solver s as

$$m_{p,s} = \text{iter}_{p,s} = \text{number of iterations required to solve the problem } p \text{ by a solver } s. \tag{13}$$

The ratio of performance $r_{p,s}$ is the specific measure compared with the best performance of all the solvers.

$$r_{p,s} = \frac{m_{p,s}}{\min\{m_{p,s} : s \in S\}}. \tag{14}$$

Moreover, the maximum value of the ratio, r_M, is defined such as $r_M > r_{p,s}$ for all p and s, and $r_M = r_{p,s}$ if and only if there is a failure of the solver. Then, the performance of the solver is defined by

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P : r_{p,s} \leq \tau\} \tag{15}$$

which represents the probability that the performance ratio for solver s is at most, a factor of τ of the best possible ratio. The term n_p represents the total number of problems in the test set.

The performance of a solver can also be defined by

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P : \log(r_{p,s}) \leq \tau\} \tag{16}$$

This logarithmic scale is considered in order to observe all the performance of the solver until $\tau = r_M$.

4.2 Performance evaluation

Before presenting the results of the benchmark, it is necessary to establish which measures are the most suitable to compare the optimization methods. It is, for example, important to use few function evaluations. For topology optimization problems on nested form, this is equivalent to the assembly of the stiffness matrix. An efficient method should, thus, assemble the matrix as few times as possible.

Moreover, the performance of a solver should be evaluated for the number of iterations (one iteration is equivalent to the solution of one sub-problem), the CPU time, and the obtained objective function value. The comparison of the objective function value is focused on the behaviour for small values of τ. This first performance will be decisive for our final recommendation, since hopefully, most the methods will achieve good designs in a small range of τ.

4.3 Penalization of inaccurate designs

It is important to be consistent and clarify under which circumstances the final design of a solver is not considered adequate and is penalized in the performance profiles.

It could happen that for some problems, a method is unable to obtain an accurate design, i.e. to sufficiently accurately satisfy the optimality conditions. In those cases, the solver fails and must be penalized in the performance profiles. We establish that the method obtains an incorrect design when some of the following conditions occurs:

- *Infeasibility.* If the feasibility error, i.e the Euclidean norm of equations (8) and (9), is greater than a threshold chosen to $\eta_{\max} = 10^{-4}$.
- *KKT conditions unsatisfied.* If the KKT error, i.e the Euclidean norm of equations (7)-(11), is greater than $\omega_{\max} = 10^{-3}$ (cf. below).
- *Incorrect sign of the objective function.* If the objective function value is greater than zero for compliant mechanism design problems and smaller than zero for minimum compliance and minimum volume problems.

In those cases, the ratio for this concrete solver s for the problem p must be $r_{p,s} = r_M$ for all the ratio criteria.

However, there are some problems that have to be removed from the test set either for computational time or for problems with the available memory. Each solver is allowed to run for 48 hours. If it cannot find a design, if MATLAB produces an exception or if IPOPT and SNOPT stop with flag of memory/time problems, the problem is removed from the final test set. If one solver is unable to solve an instance for these reasons, this problem is not solved for any solver. When these errors happen, it is not possible to consider that the solver is unable to obtain a good design because it is simply, it cannot solve such a large problem. It is not a problem of the underlying method by itself, it is an issue related to the particular choices in the implementation and the properties of the computer.

4.4 Test set of topology optimization problems

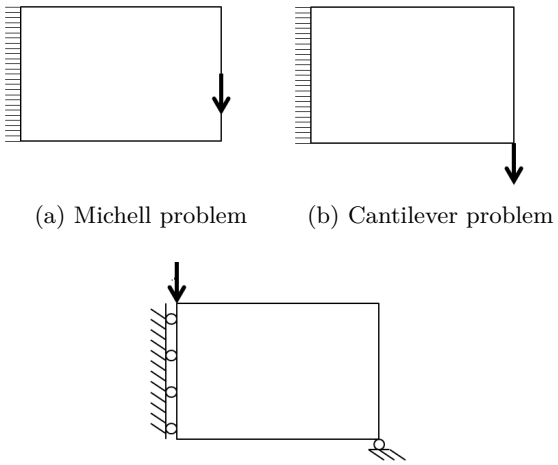
It is important to produce a large and representative test set of topology optimization problems in order to be able to conclude and state recommendations from the performance profiles. The election of which problems should be selected to obtain a fair benchmark is always a source of disagreements. The test set should be heterogeneous, interesting and difficult to solve.

Some of the most typical test sets of problems considered in the literature for linear and nonlinear optimization methods are CUTE [10], CUTEr [25], MIPLIB [29], COPS [9] and Vanderbei among others (used in [7], [19] and [47]). In contrast, in the topology optimization field, there is no publicly available big test set of problems. In general, the numerical results in research articles are made using only three or four examples, see e.g. [18], and [48].

In [41] it is noted that there are several test problems well-known for benchmarking minimum compliance problems, however, there is no standardization regarding compliant mechanism design problems. In this section we present a test set of problems in two dimensions and with a rectangular design domain. Moreover, the static external load is a single nonzero vector with value equal to one ($f_i = 1,\ f_j = 0 \quad \forall j \neq i$). Finally, for simplicity, the volume is identical for all the elements, i.e. $a_i = a_j \quad \forall i, j = 1, \ldots, n$.

4.4.1 Minimum compliance and minimum volume test problems

Three different types of design domains and loads are considered (see Figure 1) for minimum compliance and minimum volume problems. These examples are considered in plenty of articles related to topology optimization such as [40], [4], [13], [2], [14], and [6]. For each domain we consider different length ratios. We experience that the methods have difficulties in solving problems with large difference between lengths.



(a) Michell problem (b) Cantilever problem

(c) MBB problem

Figure 1: Michell, Cantilever, and MBB design domains, boundary conditions and external load definitions that are collected and used to define the benchmarking library.

Moreover, for each length, different size of discretization, N_l, are defined. N_l refers to the number of elements in the finite element analysis defined for unit of length. We decided to build a test set with a relatively low number of elements to be able to test all the solvers. Since several of the implementations are general purpose codes, which are not designed for simulation based problems, larger numbers of elements will eventually result in problems of memory. The memory usage could be an important factor when different solvers are compared. However, since each solver is implemented in different languages, the interfaces between MATLAB and these programs increase memory usage significantly. Therefore, it is not possible to produce any fair study of memory usage.

Finally, for each design domain in the minimum compliance test set, 5 different volume fraction upper bounds are considered. The volume bound is chosen from the values 0.1, 0.2, 0.3, 0.4, and 0.5. Solving problems with small volume bounds is, in our experience, generally more difficult and should give some issues. In general, when the amount of material available is reduced, the solvers have more difficulties in distributing the material so that the structure satisfies the constraints and minimizes the objective function. For minimum volume problems, three different instances, with upper bound in the compliance proportional to $k = 1$, 1.2, and 1.5 (see (12)) are generated.

A brief description of the test set is outlined in Tables 7, 9, and 8. The test set of minimum compliance problems contains 5 different problems for each characteristic's row (one for each volume bound) with a total number of problems of 225, while the minimum volume test set contains 135 problems.

Table 7: Test set of problems for the Michell design domain, see Figure 1a. L_x and L_y denote the length ratio on the x and y direction, respectively. N_l denotes the size of the discretization per unit of length, n the number of elements and d the number of degrees of freedom.

Domain	L_x	L_y	N_l	n	d
Michell	1	1	20	400	882
			40	1600	3362
			60	3600	7442
			80	6400	13122
			100	10000	20402
	2	1	20	800	1722
			40	3200	6642
			60	7200	14762
			80	12800	26082
			100	20000	40602
	3	1	20	1200	2562
			40	4800	9922
			60	10800	22082
			80	19200	39042
			100	30000	60802

Table 8: Test set of problems for the MBB design domain, see Figure 1c.

Domain	L_x	L_y	N_l	n	d
MBB	1	2	20	800	1722
			40	3200	6642
			60	7200	14762
			80	12800	26082
			100	20000	40602
	1	4	20	1600	3402
			40	6400	13202
			60	14400	29402
			80	25600	52002
			100	40000	81002
	2	1	20	800	1722
			40	3200	6642
			60	7200	14762
			80	12800	26082
			100	20000	40602
	4	1	20	1600	3402
			40	6400	13202
			60	14400	29402
			80	25600	52002
			100	40000	81002

Table 9: Test set of problems for the Cantilever design domain, see Figure 1b.

Domain	L_x	L_y	N_l	n	d
Cantilever	2	1	20	800	1722
			40	3200	6642
			60	7200	14762
			80	12800	26082
			100	20000	40602
	4	1	20	1600	3402
			40	6400	13202
			60	14400	29402
			80	25600	52002
			100	40000	81002

Table 10: Test set of 150 compliant mechanism design problems, see Figure 2.

Domain	L_x	L_y	N_l	n	d
Inverter/ Gripper/ Amplifier/ Compliant Lever/ Crimper	1	1	20	400	882
			40	1600	3362
			60	3600	7442
			80	6400	13122
			100	10000	20402
	2	1	20	800	1722
			40	3200	6642
			60	7200	14762
			80	12800	26082
			100	20000	40602

4.4.2 Compliant mechanism design problem

The most typical problem used to test the methods in the literature is the force inverter, shown in Figure 2a, see e.g. [41]. Furthermore, another typical compliant mechanism, called the compliance gripper, is included in the benchmark library, see Figure 2b. Three more examples denoted the amplifier, the compliant lever, and the crimper, shown in Figure 2c, 2d, and 2e, respectively, complete the mechanism design test set. These problems can be found in different publications such as [6], [48], [40], [14], [39], [18], [31], and [38].

For each design domain, 5 different discretizations, and three different volume fractions: 0.2, 0.3, 0.4, are considered resulting in a a final test set of 150 problems gathered in Table 10.

5 Numerical experiments

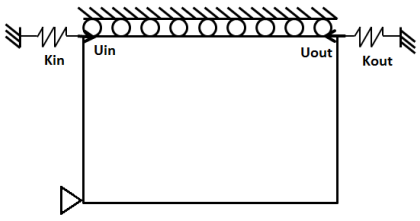
The benchmark library is based on 225 minimum compliance problems, 135 minimum volume problems and 150 mechanism design problems. However, the final benchmark library is made of a subset using 121, 64 and 124 examples respectively. For the rest of the problem instances, either IPOPT, SNOPT or FMINCON in the SAND formulation have memory problems or the computational time required is more than the maximum allowed (which is 48h). The need of research in efficient and fast large-scale methods to be able to solve large saddle-points systems is evident from this fact.

Furthermore, as the objective function value of the compliant mechanism design problems is negative, large values of ratios represent better performance than small values of ratios. In order to observe the same scale and behaviour in the profiles for compliant mechanism design problems, the ratio of the objective function value is computed as the inverse of equation (14).

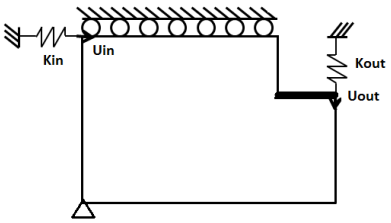
All computations were done on an Intel Xeon X5650 6-core CPUs, running at 2.66 GHz and with 4 GB Memory for each core.

It is important to highlight that the computation time is highly affected by the amount of work assigned to the nodes in the cluster. The time for the same problem using the same solver at different moments in time can vary significantly. Therefore, we do not recommend that the performance profiles for computational time are trusted unconditionally.

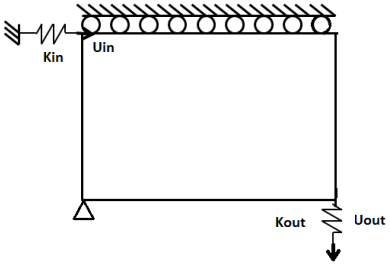
Finally, some restrictions have been made during the numerical experiments. Both the SAND and the SANDNL formulations require a lot of memory, and, for large problems, the solvers are unable to run. In a reduced test set of small problems, the performance of both formulations is compared. The results are not reported here, but both formulations obtain similar results in function objective value, number of



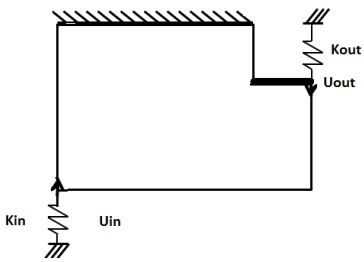
(a) Force inverter example with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.001$



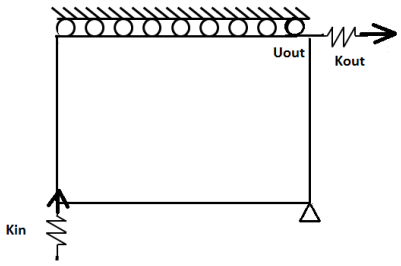
(b) Compliant gripper example with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.005$



(c) Amplifier example with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.005$



(d) Compliant lever example with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.005$



(e) Crimper example with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.05$

Figure 2: Compliant mechanism design domains, with boundary conditions and external loads definition.

iterations and also computational time. The performance of SNOPT and FMINCON on the SAND/SANDNL formulation is not competitive. In order to save time, represent more problems and produce clearer results we have decided to remove the SANDNL formulation. The SAND formulation for SNOPT and FMINCON are also excluded from the minimum volume benchmark.

Throughout this section we denote by IPOPT-N, FMINCON-N and SNOPT-N the solvers applied to a nested formulation and by IPOPT-S, FMINCON-S and SNOPT-S for SAND formulations.

5.1 Numerical experiments for minimum compliance problems

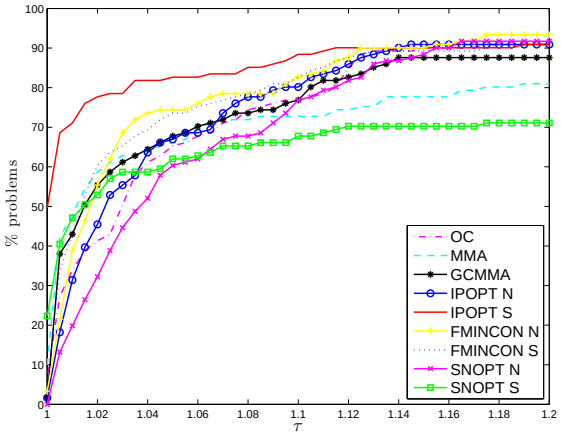
Section 4.3 considers a design to be incorrect if the Euclidean-norm of the KKT conditions is higher than a threshold set to $\omega_{\max} = 10^{-3}$. Figure 3 shows the impact of this decision. The figure contains the performance profiles for the 121 different minimum compliance problem instances for $\omega_{\max} = 10^{-2}$, 10^{-3}, and 10^{-4}, respectively. The performance is measured with the objective function value. Most of the solvers are able to obtain a design with a KKT error lower than 10^{-2}. However, for $\omega_{\max} = 10^{-4}$, it is more likely that the solvers fail. While Figure 3a shows the performance of the solvers to obtain a design ($\omega_{\max} = 10^{-2}$), Figure 3c shows the robustness of the solvers. It is clear that solvers such as MMA and GCMMA are highly affected by this threshold. Their performances decrease considerably when we enforce the optimality tolerance (10^{-4}). The performance of GCMMA decays from 87% of success (when $\omega_{\max} = 10^{-2}$) to 70% (when $\omega_{max} = 10^{-4}$). Similarly, the performance of MMA drops from 82% to 52%. On the other hand, the percentage of success for IPOPT and SNOPT are not affected by the parameter $\omega_{\max}$. IPOPT and SNOPT can either produce a design where the optimality conditions are satisfied or produce very poor results.

With this in mind, Figure 4a presents the performance profile for objective function value for small values of τ, which gives the performance of the solvers when they are close to the best design. IPOPT-S obtains the best objective function value for 50% of the problem instances. The chances for the rest of the solvers of winning are small, lower than 25%. However, as τ increases IPOPT-N, SNOPT-N, FMINCON-N, FMINCON-S and OC become more competitive. If we choose, for instance, being at a factor of $\tau = 1.12$ to the best solver, they have a probability close to 90% to obtain a design. On the other hand, SNOPT-S, MMA, and GCMMA obtain remarkably poor results. They are only able to produce a design with an objective function value 1.2 times the best value in less than the 80% of the cases.

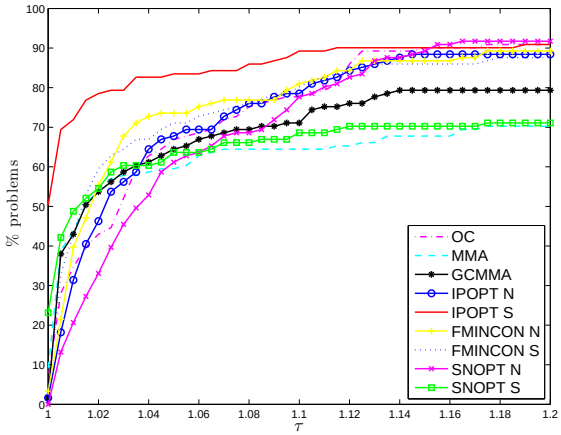
In Figure 4b[3] the robustness of each method can be identified. When $\tau \sim r_M = 10^2$ the percentage of problems is equivalent to the probability to obtain a design. MMA is able to obtain a design in only the 70% of the cases, SNOPT-S in 75%, and GCMMA in 80%.

Figure 4b also shows that FMINCON-N obtains a design using the least number of iterations, followed by SNOPT-N. Although IPOPT-S has a lower number of wins,

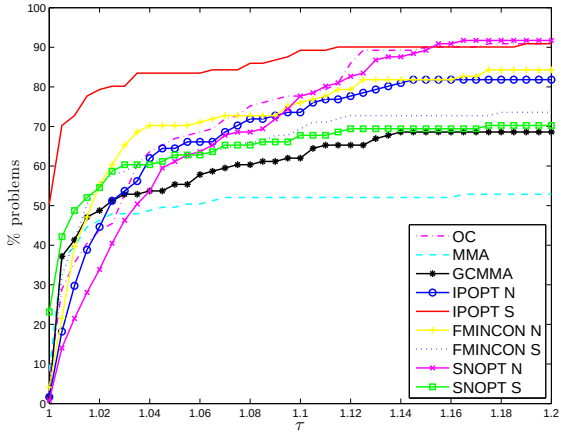
[3] A design, which is deemed incorrect by the optimality conditions, can indeed be a capable design and visually describe the correct topology. However, we experience that tight optimality conditions lead to better objective function values.



(a) Penalization of problems with $\omega_{\max} = 10^{-2}$

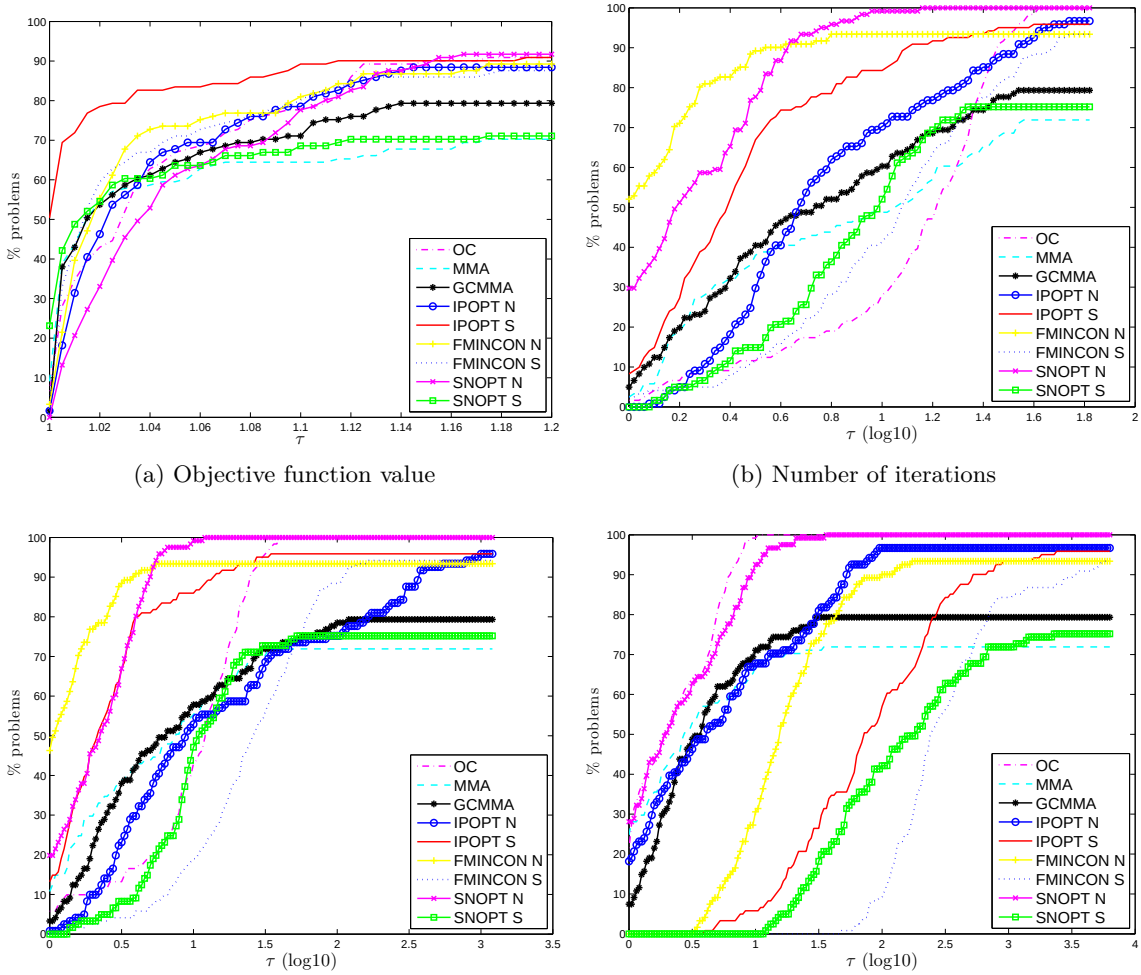


(b) Penalization of problems with $\omega_{\max} = 10^{-3}$



(c) Penalization of problems with $\omega_{\max} = 10^{-4}$

Figure 3: Performance profiles for the (reduced) test set of 121 minimum compliance problems (P_N^c) and (P_S^c). The performance is measured for objective function value. A problem is penalized in the performance profiles if the KKT error is higher than 10^{-2} (3a), 10^{-3} (3b) or 10^{-4} (3c), respectively.



(a) Objective function value

(b) Number of iterations

(c) Number of stiffness matrix assemblies

(d) Computational time

Figure 4: Performance profiles for the (reduced) test set of 121 minimum compliance problems (P_N^c) and (P_S^c) described in Tables 7, 9 and 8. The performance is measured by objective function value (4a), number of iterations (4b), number of stiffness matrix assemblies (4c), and computational time (4d).

it becomes a strong rival when τ increases. FMINCON-N, SNOPT-N and IPOPT-S perform at a factor of 10 to the best solver in around 85-100% of the instances. In general, the rest of the solvers require substantial numbers of iterations to obtain a design (the performance is lower than 80%).

The performance of the solvers for the number of stiffness matrix assemblies (Figure 4c) is very similar to for the number of iterations.

Finally, Figure 4d shows the performance profile for the computational time. It suggests that SNOPT-N or OC are the best solvers for minimum compliance problems. The rest of the solvers consume more than 10^2 times, even 10^3 times more, to achieve similar percentage of success. It is also clear that the SAND formulation requires a lot of time. However, this study is done using small-scale problems.

We observe that IPOPT-S (which uses an exact Hessian of the Lagrangian) outperforms IPOPT-N (which uses a limited memory BFGS approximation of the Hessian) with respect to the objective function value, the number of iterations, and the number of assemblies. In contrast, it is clear that the SAND formulation consumes more time. This could be either for the increase of the number of variables and constraints or because of the use of exact Hessian. On the other hand, if the computational time for FMINCON (both formulations compute the exact Hessian) or SNOPT (both formulations use limited memory BFGS approximations) is compared, we conclude that the increment of time is due to the increment of variables and constraint but not the use of second order information.

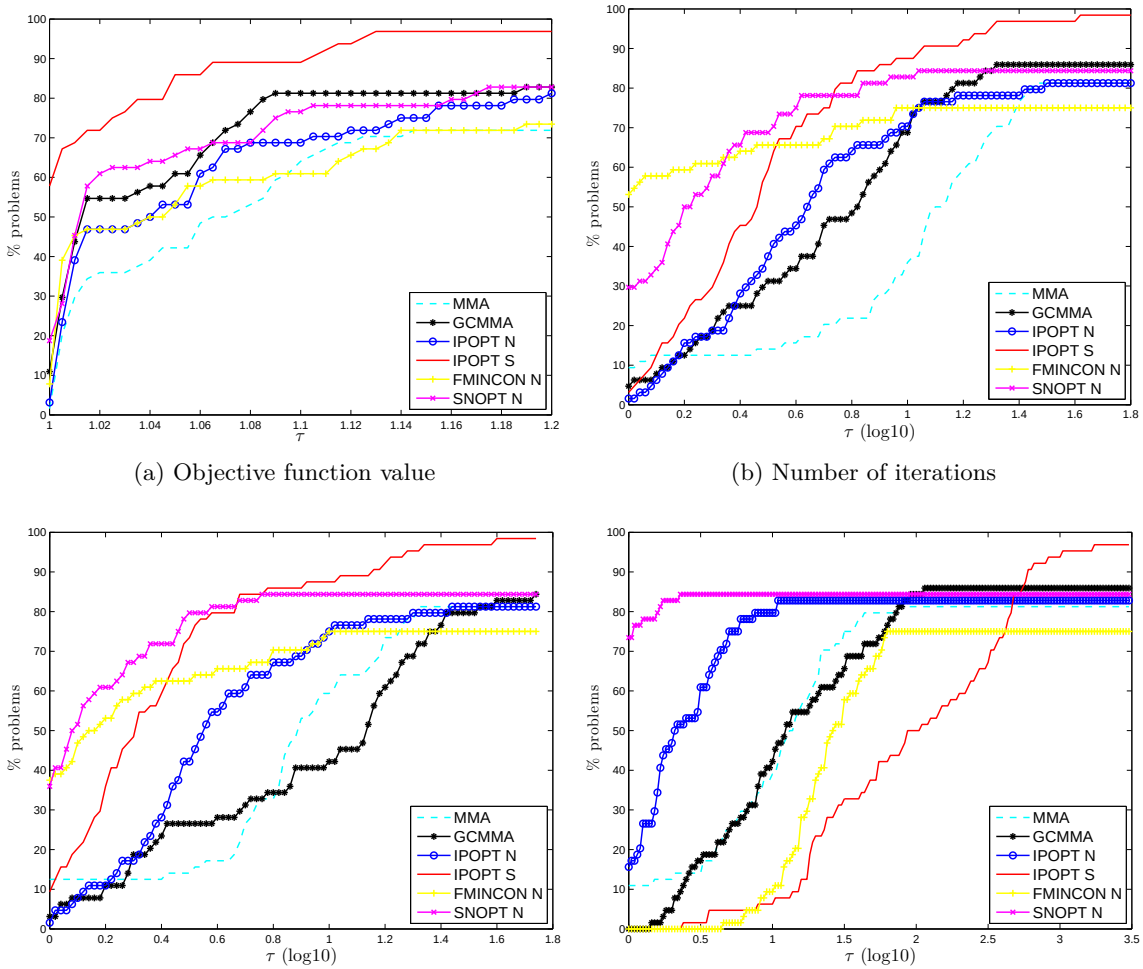
5.2 Numerical experiments for minimum volume problems

Figure 5 shows the performance profiles for the 64 minimum volume problems. This test sect is a sub-set of the problems listed in Tables 7, 9 and 8.

It is clear that IPOPT-S outperforms the rest of the solvers with respect to the obtained objective function value, see Figure 5a. The probability for IPOPT-S of winning is higher than 55%. In addition, it has at least 10% more chances than any other solver to obtain a final design with an objective function value at a factor of $\tau = 1.2$. Unlike the minimum compliance problems, FMINCON is unable to solve more than the 70% of the problems. MMA shows similar performance.

However, IPOPT-S also has the problem that it consumes more iterations and assemblies of the stiffness matrix than SNOPT or FMINCON. Nevertheless, IPOPT-S improves really fast as τ increases. If we are interested in a solver that can solve more than the 80% of the problems with the greatest efficiency as possible (for the number of iterations), then the election should be either IPOPT-S or SNOPT. Their performance for this 80% is at a factor of $\tau = 5.6$.

The performance of the solvers with respect to the number of assemblies is represented



(a) Objective function value

(b) Number of iterations

(c) Number of stiffness matrix assemblies

(d) Computational time

Figure 5: Performance profiles for the (reduced) test set of 64 minimum volume problems (P_N^w) and (P_S^w) described in Tables 7, 9 and 8. The performance is measured by objective function value (5a), number of iterations (5b), number of stiffness matrix assemblies (5c), and computational time (5d).

in Figure 5c. The robustness of IPOPT-S is again demonstrated since it can solve almost 100% of the problems. In contrast, the rest of the solvers have between 75 to 85% of chances to solve a problem. It is remarkable that MMA and GCMMA require more iterations and more stiffness matrix assemblies than the general nonlinear solvers.

The performance of the solvers, for computational time, is similar to the results for minimum compliance problems as observed in Figure 5d.

5.3 Numerical experiments for compliant mechanism design problems

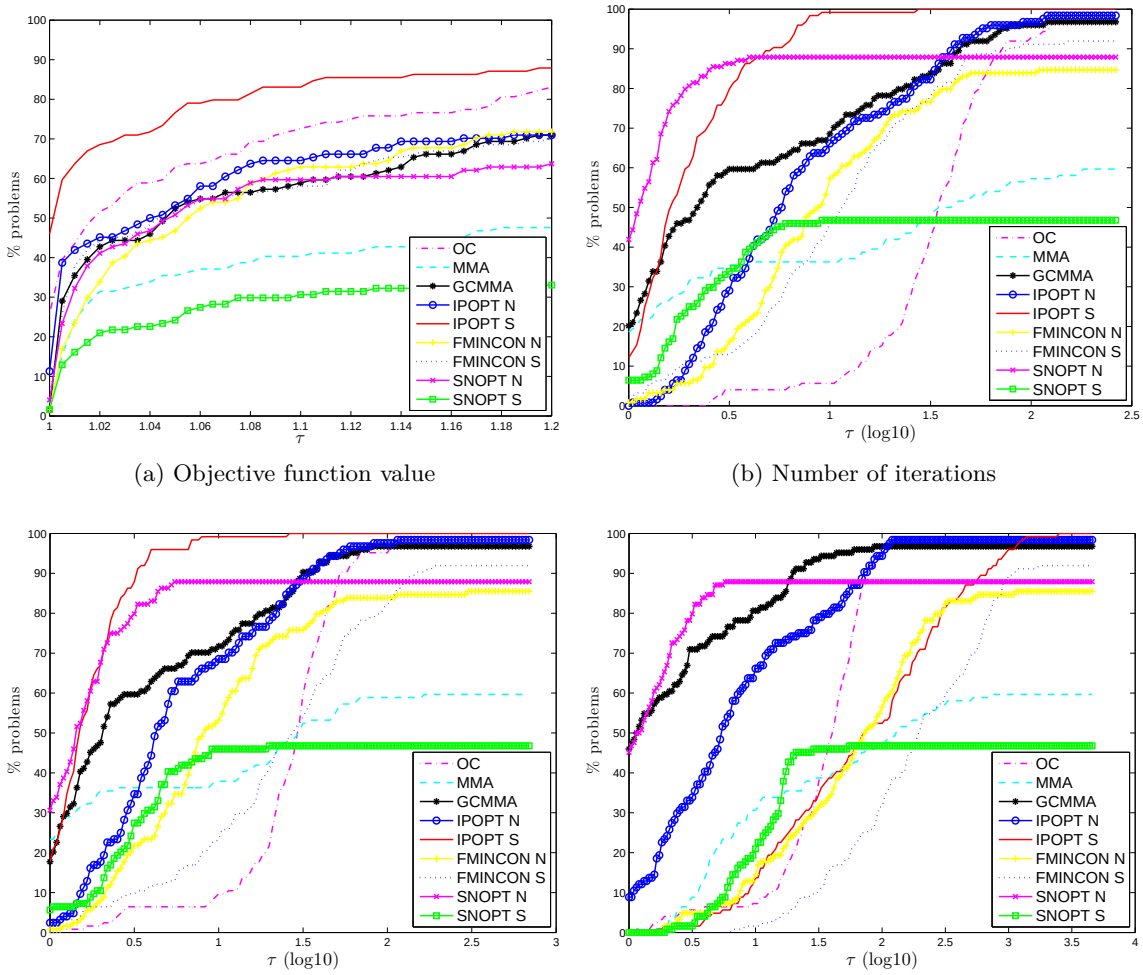
Figure 6 shows the performance profiles for the 124 compliant mechanism design problems listed in Table 10. Our experience is that these problems, in general, are more difficult to solve than minimum compliance and volume problems. This is reflected in the performance profiles. One more time, IPOPT-S highlights from the rest of the solvers. Although, OC has the best performance in 25% of the problems, its probability to obtain a design with an objective function value between the best one and 1.2 times, is not higher than 80%. The rest of the solvers have a very poor performance.

Figure 6b shows the performance of the solvers for the number of iterations. Although SNOPT-N obtains the best percentage of winners, IPOPT-S has a 100% of chances to obtain a design in at most 10 times more iterations than the best solver. At this factor, only SNOPT-N is able to obtain a success in more than the 70% of the instances. Indeed, SNOPT-S and MMA have very low performance. They solve less than 65% of the problems. Although OC has the 100% of probability to obtain a design, the performance for any value of τ is very poor. In general, OC stops because it reaches the maximum number of allowed iterations and not because the `change` parameter is satisfied (which is not penalized in these performance profiles).

The behaviour of the solvers for number of stiffness matrix assemblies is very similar to the number of iterations. For mechanism design problems, GCMMA stands out compared to FMINCON. This is unlike the situation for minimum compliance and minimum volume problems.

5.4 Conclusions from the numerical experiments

After the detailed explanation of the performance profiles for the different type of problems, we can generalize that IPOPT-S obtains good designs, it is very robust, and the performance does not depend on the problem under consideration. With more than 95% of probability, IPOPT-S is able to obtain a design satisfying the required optimality conditions. The use of second order information helps IPOPT to produce better designs using less iterations. On the other hand, it requires a lot of computational time.



(a) Objective function value

(b) Number of iterations

(c) Number of stiffness matrix assemblies

(d) Computational time

Figure 6: Performance profiles for the (reduced) test set of 124 compliant mechanism design problems (P_N^m) and (P_S^m) described in Table 10. The performance is measured by objective function value (6a), number of iterations (6b), number of stiffness matrix assemblies (6c), and computational time (6d).

In contrast, SNOPT-N produces poor designs (bad objective function value) but using very few number of iterations and computational time.

In general, the SAND formulation requires a lot of time. There is a need of improvements in the computation of the saddle-point system in order to reduce this computational time to be able to obtain a competitive solver when the SAND formulation is used.

Finally, it is curious to observe that the structural optimization methods, MMA and GCMMA, do not outperform the other solvers in neither objective function value nor number of iterations. Moreover, the performance profiles reveal that GCMMA in general obtains better designs and requires less iterations than MMA. The additional measures implemented in GCMMA compared to MMA to ensure the theoretical global convergence results in [44] apparently also have a positive effect on the numerical performance.

6 Limitations of the benchmark

We are aware of the many limitations of this benchmarking study. However, it is computationally too demanding to benchmark all possible combinations of optimization methods and problem formulations. We therefore decided to focus on a small, but relevant, range of combinations of problem formulations and methods and develop a representative and meticulous benchmark. Further research must be done in order to contemplate other alternatives to the problems.

First of all, only one topology optimization approach, namely combining the SIMP material interpolation scheme with a density filter, has been used. The RAMP material interpolation [42] and [36] has been numerically tested, giving similar results to the SIMP scheme. It could be interesting to observe the performance of the solvers when other penalizations schemes or regularization approach are used, such as for example perimeter control [26]. The problems, furthermore, only include a few mechanical requirements. Important constraints such as displacement and stress constraints, are not modelled and included in the benchmark problems.

Moreover, there are limitations regarding the choice of solvers used for the benchmarking. There is room for testing several other general nonlinear optimization solvers. This includes those implemented in the KNITRO package [16], the interior point method in LOQO [46], the sequential augmented Lagrangian algorithms in LANCELOT [17], or MINOS [34], the generalized augmented Lagrangian method in PENNON [30], and the feasible direction method in FAIPA [27], among others.

The scaling of the problems as well as different parameter values such as the Young's modulus, the magnitude of the external loads, the penalty parameter p in the SIMP penalization, and the filter radius $r_{\min}$ of the density filter have been set to only one value. It is likely that the performance of the solvers is affected by them.

Finally, the test set considers only 2D problems with simple design domains, single external loads, and the structural analysis is done using only one type of finite elements. The sizes of the problem instances in the test are generally only medium-scale. The performance of the solvers could be reduced significantly with increasing number of finite elements and degrees of freedom.

7 Conclusions and future research

An extensive benchmarking of topology optimization problems in combination with different optimization methods has been developed for the first time in the community. The benchmark is based on a specific set of test problems and uses performance profiles which has been proven a great approach for the comparison of solvers.

The main objective of this study is to investigate the performance of both general nonlinear optimization solvers and special purpose methods intended for structural topology optimization problems. The numerical experiments indicate that the use of second-order information is helpful to obtain good designs. IPOPT, applied to the SAND formulation, computes the exact Hessian of the Lagrangian. It outperforms all the other solvers for minimum compliance, minimum volume, and compliant mechanism design problems when the final objective function values are compared. This combination is also the most reliable since it solves the largest percentage of the problems in the test set. In contrast, the computational time required to obtain a design is very high for IPOPT. It is important to remark that some problems have been removed from the test set due to limitations in time and memory caused for the SAND formulation. However, the performance profiles show that this method is the most robust.

Most remarkable is, perhaps, that the performance profiles point out that structural topology optimization problems can robustly be solved using general nonlinear solvers rather than structural optimization methods. The performance of the general solvers is comparable to MMA and GCMMA, and are as efficient and reliable as structural topology optimization solvers. Notable is also that, in general, GCMMA produces designs with better objective function values than MMA and it is also more reliable.

Further research should be done in order to improve the efficiency of implementations of second-order optimization solvers applied to the SAND formulation to reduce the computational time required for topology optimization problems. The most expensive step in interior point methods is the solution of the saddle-point system to compute the search direction. Therefore, advanced and efficient iterative solvers should be developed and implemented for the resolution of these large-scale linear systems.

Acknowledgements

We would like to thank Professor Krister Svanberg at KTH in Stockholm for providing the implementations of both MMA and GCMMA.

References

[1] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2–4):501–520, 2000.

[2] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[3] J. S. Arora and Q. Wang. Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization*, 30(4):251–272, 2005.

[4] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):192–202, 1989.

[5] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9–10):635–654, 1999.

[6] M. P. Bendsøe and O. Sigmund. *Topology optimization: Theory, methods and applications*. Springer, 2003.

[7] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei. A comparative study of large-scale nonlinear optimization algorithms. Technical report, Operations Research and Financial Engineering, Princeton University, 2002.

[8] P. T. Boggs and J. W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

[9] A. S. Bondarenko, D. M. Bortz, and J. J. Moré. COPS: Large-scale nonlinearly constrained optimization problems. Technical Report ANL/MCS-TM-237, Mathematics and Computer Science Division, Argonne National Laboratory, 1999.

[10] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21(1):123–160, 1995.

[11] T. Borrvall and J. Petersson. Large-scale topology optimization in 3D using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6201–6229, 2001.

[12] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[13] M. Bruggi and P. Duysinx. Topology optimization for minimum weight with compliance and stress constraints. *Structural and Multidisciplinary Optimization*, 46(3):369–384, 2012.

[14] T. E. Bruns. A reevaluation of the SIMP method with filtering and an alternative formulation for solid void topology optimization. *Structural and Multidisciplinary Optimization*, 30(6):428–436, 2005.

[15] M. Burger and R. Stainko. Phase-field relaxation of topology optimization with local stress constraints. *SIAM Journal on Control and Optimization*, 45(4):1447–1466, 2006.

[16] R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO : An Integrated Package for Nonlinear Optimization. In *Large Scale Nonlinear Optimization*, volume 83, pages 35–59. Springer, 2006.

[17] A. R Conn, N. I. M. Gould, and P. L. Toint. *Lancelot: A FORTRAN Package for Large-Scale Nonlinear Optimization (Release A)*. Springer-Verlag, 1992.

[18] S. R. Deepak, M. Dinesh, D. K. Sahu, and G. K. Ananthasuresh. A comparative study of the formulations and benchmark problems for the topology optimization of compliant mechanisms. *Journal of Mechanisms and Robotics*, 1(1), 2009.

[19] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[20] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[21] A. Evgrafov, C. J. Rupp, K. Maute, and M. L. Dunn. Large-scale parallel topology optimization using a dual-primal substructuring solver. *Structural and Multidisciplinary Optimization*, 36(4):329–345, 2008.

[22] C. Fleury. Efficient approximation concepts using second order information. *International Journal for Numerical Methods in Engineering*, 28(9):2041–2058, 1989.

[23] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

[24] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large -Scale Constrained Optimization. *SIAM Journal on Optimization*, 47(4):99–131, 2005.

[25] N. I. M. Gould, D. Orban, and P. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[26] R. B. Haber, C. S. Jog, and M. P. Bendsøe. A new approach to variable-topology shape design using a constraint on perimeter. *Structural optimization*, 11(1–2):1–12, 1996.

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

[28] C. F. Hvejsel and E. Lund. Material interpolation schemes for unified topology and multi-material optimization. *Structural and Multidisciplinary Optimization*, 43(6):811–825, 2011.

[29] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011.

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

[31] G. K. Lau, H. Du, and M. K. Lim. Use of functional specifications as objective functions in topological optimization of compliant mechanism. *Computer Methods in Applied Mechanics and Engineering*, 190(34):4421–4433, 2001.

[32] S. Leyffer and A. Mahajan. Software for Nonlinearly Constrained Optimization. Technical Report ANS/MCS-P1768-0610, Mathematics and Computer Science Division, Argonne National Laboratory, 2010.

[33] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[34] B. A. Murtagh and M. A. Saunders. MINOS 5.5 User's Guide. Technical Report SOL 83-20R, Stanford University Systems Optimization Laboratory, Department of Operations Research, 1998.

[35] J. Nocedal, R. Wächter, and R. A. Waltz. Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization*, 19(4):1674–1693, 2009.

[36] A. Rietz. Sufficiency of a finite exponent in SIMP (power law) methods. *Structural and Multidisciplinary Optimization*, 21(2):159–163, 2001.

[37] G. I. N. Rozvany and M. Zhou. The COC algorithm, part I: Cross-section optimization or sizing. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):281–308, 1991.

[38] A. Saxena and G. K. Ananthasuresh. Topology synthesis of compliant mechanisms for nonlinear force-deflection and curved path specifications. *Journal of Mechanical Design*, 123(1):33–42, 2001.

[39] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, 25(4):492–526, 1997.

[40] O. Sigmund. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica*, 25(2):227–239, 2009.

[41] O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[42] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, 2001.

[43] K. Svanberg. The method of moving asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

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

[45] Inc. The MathWorks. Optimization Toolbox User's Guide R release 2014a, 2014.

[46] R. J. Vanderbei. LOQO User's Manual. Version 4.05. Technical report, Operations Research and Financial Engineering Princeton University, 2006.

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

[48] F. Wang, B. S. Lazarov, and O. Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43(6):767–784, 2011.

[49] H. Wang, Q. Ni, and H. Liu. A new method of moving asymptotes for large-scale linearly equality-constrained minimization. *Acta Mathematicae Applicatae Sinica*, 27(2):317–328, 2011.

[50] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2):227–246, 2003.

[51] S. Wang, E. Sturler, and G. H. Paulino. Large scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69:2441–2468, 2007.

[52] W. H. Zhang, C. Fleury, P. Duysinx, V. H. Nguyen, and I. Laschet. A generalized method of moving asymptotes (GMMA) including equality constraints. *Structural optimization*, 12(2-3):143–146, 1996.

[53] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):309–336, 1991.

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

7

Article II: Automatic penalty continuation in structural topology optimization

Published online the 28 July 2015:
Rojas-Labanda, S. and Stolpe, M.: Automatic penalty continuation in structural topology optimization. *Structural and Multidisciplinary Optimization* (2015). DOI : 10.1007/s00158-015-1277-1.

Automatic penalty continuation in structural topology optimization*

Susana Rojas-Labanda[+] and Mathias Stolpe[+]

[+]DTU Wind Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark. E-mail: srla@dtu.dk, matst@dtu.dk

Abstract

Structural topology optimization problems are often modelled using material interpolation schemes to produce almost solid-and-void designs. The problems become nonconvex due to the use of these techniques. Several articles introduce continuation approaches in the material penalization parameter to reduce the risks of ending in local minima. However, the numerical performance of continuation methods has not been studied in detail.

The first purpose of this article is to benchmark existing continuation methods and the classical formulation with fixed penalty parameter in structural topology optimization. This is done using performance profiles on 225 minimum compliance and 150 compliant mechanism design problems.

The results show that continuation methods, generally, find better designs. On the other hand, they typically require a larger number of iterations. In the second part of the article this issue is addressed. We propose an automatic continuation method, where the material penalization parameter is included as a new variable in the problem and a constraint guarantees that the requested penalty is eventually reached. The numerical results suggest that this approach is an appealing alternative to continuation methods. Automatic continuation also generally obtains better designs than the classical formulation using a reduced number of iterations.

Keywords: Topology optimization, Continuation methods, Benchmarking, Mechanism design, Minimum compliance

Mathematics Subject Classification 2010: 74P05, 74P15, and 90C90

*This research is funded by the Villum Foundation through the research project Topology Optimization - The Next Generation (NextTop).

1 Introduction

Structural topology optimization distributes the material in a design domain to minimize an objective function under certain constraints [6]. In the most common formulation, the design variable is chosen as the density $\mathbf{t}$, which is defined as a continuous variable with values between $\mathbf{0}$ (void) and $\mathbf{1}$ (solid). The design domain is discretized using finite elements and a design variable is associated with each element. The aim is often to obtain a final design which is almost solid-and-void. In order to penalize intermediate values, different interpolation schemes such as the Solid Isotropic Material with Penalization (SIMP), see e.g. [4], [25], and [38], or the Rational Approximation of Material Properties (RAMP) [30] are used. These approaches *principally* modify the stiffness matrix in the following ways,

$$\mathbf{K}(\mathbf{t}) = \begin{cases} \displaystyle\sum_{e=1}^{n} t_e^p \mathbf{K}_e & \text{in the SIMP scheme} \\ \displaystyle\sum_{e=1}^{n} \frac{t_e}{1+(p-1)(1-t_e)} \mathbf{K}_e & \text{in the RAMP scheme.} \end{cases} \tag{1}$$

Here, $\mathbf{K}_e$ is the element stiffness matrix of unit density, the stiffness matrix is $\mathbf{K}(\mathbf{t}) : \mathbb{R}^n \to \mathbb{R}^{d\times d}$, n is the number of elements, and d is the number of degrees of freedom. The density variable is defined with $\mathbf{t} \in \mathbb{R}^n$ and the material penalization parameter with $p \geq 1$ for both the SIMP and the RAMP[a]interpolation schemes.

For values of the parameter $p > 1$, the topology optimization problem generally becomes nonconvex. Thus, numerical optimization solvers could end in a local minimum. However, it is common to use a *continuation method* to avoid local minima, see e.g. [22]. This technique consists of obtaining a solution of the problem without penalization ($p = 1$), where the optimal design generally has grey regions (intermediate density values). Then, the value of the material penalization parameter is gradually increased in small steps and the problem is resolved. This is continued until the desired value is reached, which should be large enough to produce (almost) solid-and-void designs. For each new value of the material penalization parameter, the optimization problem is normally solved using the solution of the previous problem as starting point.

Continuation methods are introduced in e.g. [2] and [1] among others. In addition, [6] and [11] suggest to use them as a standard procedure.

Particularly, as the review article [24] explains, one starts from a global optimum and after some steps, the grey regions change into black-and-white regions. It is expected that the final design does not move too far from the solution of the convex problem.

[a]The material penalization parameter in the RAMP approach is commonly defined as $q = p - 1$ in the literature, see e.g. [30], [28], and [17].

Nevertheless, one of the major drawbacks of these methods is the increase in the number of iterations required for convergence, see e.g. [22] and [10].

Many articles use the technique of solving a sequence of problems with increasing value of p, such as [19], [37], [29], [22], [36], [10], and [14] among others. These articles claim this method helps to avoid ending in a local minimum. We find in [29]: "*Based on experience, it seems that continuation methods must be applied because, by construction, they take also "global" information into account and are thus more likely to ensure "global" convergence (or at least convergence to better designs)*". The review [24] states, "*On the basis of many years experience, one should start with p=1, use small increments of p...*". Moreover, it is mentioned in [10] that: "*Using this approach we experience that the "optimal" designs can be reproduced with a high accuracy and as the examples will show, it makes it possible to compare the objective functions of different designs with several digits of accuracy*". In addition, the review [11] reports: "*...continuation methods are frequently used in the literature to increase the chance of obtaining a global optimal solution.*" and "*...they nonetheless perform very well in practical applications, especially when used with a regularization scheme*".

In contrast, [31] shows some examples where the continuation approach in the material penalization parameter fails. The article concludes that "*...although the continuation approach combined with some penalization techniques may be a very good heuristic in many cases, it is not possible to prove any convergence results*". Furthermore, [36] concludes that: "*The global optimal solution cannot always be obtained by continuation with respect to the penalization parameter....*" However, they assert that "*... a good approximate solution is found in the numerical examples*".

One of the purposes of this article is to study if existing continuation methods for structural topology optimization problems help to obtain, in general, good designs. We also study if these methods are more robust[b] than the classical approach, where no continuation techniques are applied. Moreover, we give a new alternative to continuation methods, namely *automatic continuation*. This new approach includes the material penalization parameter as a variable in the definition of the optimization problem. Additionally, an extra constraint is added to force the material penalization variable towards the requested final value during the optimization process. In contrast to other continuation methods, our proposed automatic continuation approach solves only one optimization problem. In our numerical experiments, the final designs obtained by automatic continuation approach, are generally better than those provided by the classical approach. Additionally, automatic continuation uses fewer iterations and therefore, less computational time than existing continuation approaches.

Specifically, we define the topology optimization problem with a material

[b] By robustness we mean the capability to find points that satisfy first-order optimality conditions to requested accuracy within stated iteration and time limitations.

interpolation scheme combined with a density filter, see e.g. [29], and [8]. Both techniques together penalize the intermediate densities, ensure the existence of a solution, and avoid checker-boards. Two structural topology optimization problems are considered; minimum compliance and compliant mechanism design problems. Both classes of problems are often included in the literature to benchmark topology optimization solvers, see e.g. [28].

Structural topology optimization problems are commonly solved using sequential convex approximations methods such as the Method of Moving Asymptotes (MMA) [32] and its globally convergent version GCMMA [33]. In particular, in the benchmarking of our continuation methods we use GCMMA for solving the problems for each value of p. The choice of GCMMA over MMA is based on our previous experience which is reported in the extensive solver benchmarking [23]. The benchmarking study shows that GCMMA generally produces better designs than MMA.

The automatic continuation formulation is based on linearization of the constraints (cf. Section 6.2). Linearization of constraints is a very common technique and it is used in Sequential Quadratic Programming (SQP) [7], interior point methods [15], and sequential linearly constrained Lagrangian methods [21], among others. We expect a good performance using solvers such as the interior point algorithm in IPOPT [34] or the sequential quadratic programming method SNOPT [16]. It is presumed that this technique helps to obtain good designs since the material penalization variable p increases gradually. Moreover, as automatic continuation solves only one problem instead of a sequence of problems, we expect that the number of optimization iterations is reduced.

The continuation approaches and the automatic continuation method are compared to the classical formulation where the material penalization parameter is kept fixed. This comparative study is done using performance profiles as introduced in [13]. Performance profile is a state-of-the-art technique commonly chosen for comparing the performance of numerical optimization solvers and problem formulations. A test library of 2D examples are collected from the literature. The library consists of 225 minimum compliance problem instances and 150 compliant mechanism design problems.

The paper is organized as follows. Section 2 formulates the considered topology optimization problems. The implementation details of the continuation methods are collected in Section 3. Section 4 introduces the performance profiles and describes the test set used for this study. Section 5 reports the performance profiles of the continuation methods and the classical formulation. The automatic continuation approach and the details of its implementation are explained in Section 6. The numerical experiments of this approach are included in Section 7. Finally, Section 8 gathers the main conclusions of the numerical experiments.

2 Problem formulation

The classical formulation of structural topology optimization problems consists of maximizing the stiffness with a constraint on the volume of the structure [6]. This formulation is equivalent to minimizing the compliance of the structure. The topology optimization problems are stated in nested forms where the displacements $\mathbf{u}$ (state variables) depend on the density $\mathbf{t}$ (design variables). The linear elastic equilibrium equations in their discretized form relate these variables through,

$$\begin{aligned} \mathbf{f} &= \mathbf{K}(\mathbf{t})\mathbf{u} \\ \mathbf{u}(\mathbf{t}) &= \mathbf{K}^{-1}(\mathbf{t})\mathbf{f}. \end{aligned} \tag{2}$$

Here, $\mathbf{f} \in \mathbb{R}^d$ is the (design independent) static external load, and $\mathbf{u} \in \mathbb{R}^d$ the displacements. The minimum compliance problem (P^c) consists of

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{aligned} \tag{P^c}$$

where $\mathbf{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$ is the relative volume of the elements with $v_i > 0$ $i = 1, \ldots, n$ and $0 < V \leq 1$ the maximum total volume fraction of the structure.

The considered compliant mechanism problem (P^m) consists of maximizing an output displacement of a flexible structure under a volume restriction,

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{maximize}} \quad & \mathbf{l}^T\mathbf{u}(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \ \mathbf{1}. \end{aligned} \tag{P^m}$$

Here, $\mathbf{l} \in \mathbb{R}^d$ is a vector with zeros in all entries except the output degree of freedom where the displacement must be maximized. More details of compliant mechanism design problems are included in e.g. [6] and [26].

For simplicity, it is assumed that:

- In order to avoid singularity, the stiffness matrix $\mathbf{K}(\mathbf{t})$ is modified to be positive definite (see (4)) for all design variables satisfying $\mathbf{0} \leq \mathbf{t} \leq \mathbf{1}$.
- All the elements have the same volume.

3 Continuation methods

In this section we describe a class of continuation methods which is similar to those described and used in the literature. Three different variants are obtained by choosing certain parameters.

Several articles suggest to apply continuation methods in the material penalization parameter to obtain better designs. Classical continuation methods consist of increasing the material penalization parameter p in small steps. For a fixed value of the penalty parameter p_k at continuation iteration k, an optimization problem is solved for a given optimality tolerance $\omega_k > 0$. Our continuation method solves sequence of sub-problems where the optimality tolerance decreases using a parameter defined with $0 < \theta < 1$, i.e. $\omega_{k+1} \approx \theta\omega_k$. This parameter helps to reduce the number of optimization iterations. Problems with material penalization parameter below the requested value, are mainly used to estimate a good starting point for the next problem. Thus, they can be solved with lower accuracy.

The outline of the continuation method proposed in this article is explained in Algorithm 1. The obtained design variable at continuation iteration k, $\mathbf{t}_k$, is chosen as the starting point for the next sub-problem. The continuation method solves one optimization problem at each iteration k. The parameter $\Delta p > 0$ is the step increment of the material penalization parameter. The initial value $p_0 = 1$ ensures the convexity of the problem (P^c) in the first continuation iteration.

Algorithm 1 Basic continuation method.

Input: Starting point $\mathbf{t}_0$, optimality tolerance ω_0, initial material penalization parameter $p_0 = 1$.

1: Set $0 < \theta < 1$ and $\Delta p > 0$.
2: Define $\omega_{\min}$ as the minimum optimality tolerance and $p_{\max}$ the maximum material penalization parameter.
3: Initialize the continuation iteration counter $k = 0$
4: **repeat**
5: Find a KKT point of the problem (P^c) or (P^m) with $\mathbf{t}_k$ as starting point, for a given p_k and ω_k.
6: Update the penalty parameter $p_{k+1} = p_k + \Delta p$
7: Update the optimality tolerance $\omega_{k+1} = \max(\omega_k\theta, \omega_{\min})$.
8: Update the iteration counter $k = k + 1$
9: **until** $p_{k+1} = p_{\max}$
10: **return**

Although it is not commonly used in continuation methods, the material penalization parameter can be increased using a nonlinear updating scheme. Step 6 in Algorithm 1 can be modified to $p_{k+1} = p_k\gamma$ with $\gamma > 1$. In this case, the step increment at the beginning of the algorithm is small, and p_k remains closer to p_0. However, when p_k approaches $p_{\max}$, the step increments becomes larger than the linear step increment Δp.

3.1 Implementation

The well-known structural optimization solver the Globally Convergent Method of Moving Asymptotes (GCMMA) [33] is selected for solving the problems in Step 5 of Algorithm 1. The problem is, for the purpose of benchmarking continuation methods, described using the SIMP material interpolation scheme (see e.g. [4], and [5]) combined with a density filter [8]. Thus, the density variable of one element depends on a weighted average of the neighbours in a radius $r_{\min}$. The filtered density in the eth element is denoted by $\tilde{t}_e$. It is defined by

$$\begin{aligned} \tilde{t}_e &= \frac{1}{\sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} t_i \\ H_{ei} &= \max(0, r_{\min} - \text{dist}(e, i)). \end{aligned} \tag{3}$$

Here, the term dist refers to the Euclidean distance between the centers of elements e and i. The set N_e contains the elements such as the distance to element i is smaller than the filter radius $r_{\min}$. In our implementation, we define $r_{\min} = 0.04 L_x$ where L_x is the length of the design domain in the x direction. This is identical to the filter radius used in e.g. [3].

The SIMP scheme interpolates the density with a power law. The modified stiffness matrix for this density filter and the SIMP penalization is

$$\begin{aligned} \mathbf{K}(\mathbf{t}) &= \sum_{e=1}^{n} (E_v + (E_1 - E_v)\tilde{t}_e^p)\mathbf{K}_e \\ p &\geq 1. \end{aligned} \tag{4}$$

Here, $E_v > 0$ and $E_1 \gg E_v$ are the Young's modulus of the "void" and solid materials, respectively. The SIMP penalization attempts to force the design variables to the limits $\mathbf{t} = \mathbf{0}$ (void) or $\mathbf{t} = \mathbf{1}$ (solid) producing a close to 0-1 design. GCMMA combined with the SIMP scheme is selected for this benchmarking study since it is one of the most popular combinations of optimization methods and material interpolation schemes in the structural topology optimization community.

It is common to set the material penalization parameter in the SIMP interpolation to $p = 3$, see e.g. [3] and [6]. This value is, in general, large enough to produce almost solid-and-void designs. Hence, the maximum value of the material penalization parameter is equal to $p_{\max} = 3$ which is also suggested in [10]. However, other maximum values are suggested in the literature. The value $p_{\max} = 5$ is used in [22] and $p_{\max} = 10$ is used in [14]. Several articles state that the increment step Δp should be sufficiently small, with values between $\Delta p = 0.1$ and $\Delta p = 0.5$, see e.g [10], [14], and [22]. Three versions of the continuation approach with different increments of the material penalization parameter are implemented and benchmarked in this article. This allows us to study whether the performance of the continuation methods are highly affected by the increment Δp or not.

The three variants are

- *Continuation method 1.* C_1-GCMMA:
  - Updating $p_{k+1} = p_k + \Delta p$ with $\Delta p = 0.1$.
  - Updating $\omega_{k+1} = \omega_k \theta$ with $\theta = 0.7$.
  - Total number of sub-problems solved = 21.
- *Continuation method 2.* C_2-GCMMA:
  - Updating $p_{k+1} = p_k \gamma$ with $\gamma = 1.09$.
  - Updating $\omega_{k+1} = \omega_k \theta$ with $\theta = 0.7$.
  - Total number of sub-problems solved = 14.
- *Continuation method 3.* C_3-GCMMA:
  - Updating $p_{k+1} = p_k + \Delta p$ with $\Delta p = 0.3$.
  - Updating $\omega_{k+1} = \omega_k \theta$ with $\theta = 0.5$.
  - Total number of sub-problems solved = 8.

Here, the total number of continuation iterations refers to the number of sub-problems solved in the continuation method to reach $p_{\max}$. Continuation method 3 solves only 8 problems, and thus the reduction factor θ for the optimality tolerance is higher than for methods 1 and 2 to be able to reduce ω_k from ω_0 to $\omega_{\min}$ in the correct number of iterations. This term does, in our experience, not significantly affect the performance of the method.

The values of the parameters set in our implementation of the continuation method using GCMMA are collected in Table 1.

For intermediate material penalization parameter values ($p_k < p_{\max}$), two stopping criteria are implemented in GCMMA for minimum compliance problems. Either the maximum number of optimization iterations equal to `max interm iter` $= 100$ is reached or a KKT point satisfying the optimality tolerance ω_k is found. When $p_k = p_{\max}$, the stopping criteria is modified to be the same as for the classical formulation (with fixed p). In this case, the stopping criteria of GCMMA are based on the optimality tolerance `tol` $= \omega_{\min} = 10^{-4}$ and the optimization iteration limit `max iter` $= 1000$.

We experience that compliant mechanism design problems are more difficult to solve than minimum compliance problems. In order to obtain accurate designs, we increase the

[c] Here, $\mathbf{e}$ refers to a vector of all ones.

[d] We consider relatively small gaps between the solid and void Young's modulus values (i.e. E_1/E_v) since we are mostly interested in the comparison of the solvers rather than the final design. Small gaps reduce the computational time and the number of iterations (see [23] for more details.)

Table 1: Values of the parameters used in the continuation method for minimum compliance and compliant mechanism design problems when the sub-problems are solved by GCMMA.

Parameter	Description	Value
$\mathbf{t}_0$	Density vector of the initial design	$V\mathbf{e}^c$
p_0	Initial value of the material penalization parameter	1
$p_{\max}$	Final value of the material penalization parameter	3
ω_0	Initial optimality tolerance	10^{-2}
$\omega_{\min}$	Final optimality tolerance	10^{-4}
Δp	Step increment of the material penalization parameter	-
θ	Factor for reducing the optimality tolerance	-
`max interm iter`	Maximum number of opt. iterations in the intermediate steps (P^c)	100
`max interm iter`	Maximum number of opt. iterations in the intermediate steps (P^m)	300
`max iter`	Maximum number of opt. iterations for (P^c)	1000
`max iter`	Maximum number of opt. iterations for (P^m)	3000
E_1	Solid Young's modulus (P^c)	10
E_1	Solid Young's modulus (P^m)	1
E_v	Void Young's modulus (P^c) and (P^m)	$10^{-2\mathrm{d}}$
ν	Poisson's ratio	0.3

maximum number of optimization iterations to 3000 and in the intermediate steps to 300 optimization iterations. The Young's modulus gap is decreased compared to minimum compliance problems to reduce the computational time.

The accumulated number of optimization iterations required for the continuation methods to obtain the final design is the sum of all GCMMA iterations required to solve the sequence of sub-problems.

Finally, the inequality constraint is scaled in both minimum compliance and compliant mechanism design problems with a factor of $\frac{1}{\sqrt{n}}$ with n the number of elements [23].

4 Benchmarking

This section contains a brief description of performance profiles and the benchmark library. These are the tools used to evaluate and compare the performance of the continuation methods in Algorithm 1 and the classical formulation with fixed penalty parameter. Performance profiles ensure a fair conclusion whether continuation methods are a good alternative to the classical approach. The test set built for the benchmarking consists of 225 minimum compliance problems and 150 compliant mechanism design problems extracted from the literature. The large problem library ensures general and representative benchmarking results.

4.1 Performance profiles

Performance profiles are an effective tool to produce fair and representative results when comparing optimization solvers and problem formulations, see [13]. Performance profiles evaluate the overall performance of the solvers using a ratio of the performance of the solver s versus the best solver, for a given metric such as the objective function value. The goal is to represent the non-decreasing function $\rho_s(\tau)$ that shows the percentage of problems that the solver s performs at a factor of τ to the best solver for the specific metric. The factor τ represents the distance between the solver and the best one. In the mathematical formulation, the performance profiles visually represent the function

$$\rho_s(\tau) = \frac{1}{N}\text{size}\{\tilde{p} \in P : r_{\tilde{p},s} \leq \tau\}, \tag{5}$$

where N is the total number of problems in the set P. In addition, the ratio of performance of a solver s for a given problem $\tilde{p}$ is

$$r_{\tilde{p},s} = \frac{m_{\tilde{p},s}}{\min\{m_{\tilde{p},\tilde{s}} : \tilde{s} \in S\}} \tag{6}$$

The set S contains all the solvers of the benchmarking study. Here, the performance is measured with a specific criterion defined by m. For these numerical experiments, the performance of the solvers are measured with the number of iterations and the objective function value, i.e.

$$\begin{aligned} m_{\tilde{p},s} = \text{iter}_{\tilde{p},s} = &\text{ number of optimization iterations required to solve} \\ &\text{the problem } \tilde{p} \text{ by solver } s. \end{aligned} \tag{7}$$

or

$$m_{\tilde{p},s} = \text{f}_{\tilde{p},s} = \text{objective function value of problem } \tilde{p} \text{ obtained by solver } s. \tag{8}$$

Performance profiles are often represented using a logarithmic scale. This makes it easier to observe the general performance. The function $\rho_s(\tau)$ is then defined by

$$\rho_s(\tau) = \frac{1}{N}\text{size}\{\tilde{p} \in P : \log_{10}(r_{\tilde{p},s}) \leq \tau\}. \tag{9}$$

Finally, since the objective function value of our compliant mechanism designs problems is negative, the ratio of performance for this criterion is defined as the inverse of equation (6), i.e.

$$r_{\tilde{p},s} = \frac{\min\{m_{\tilde{p},\tilde{s}} : \tilde{s} \in S\}}{m_{\tilde{p},s}}. \tag{10}$$

Moreover, in our numerical experiments, we consider that a problem is not accurately solved if the Euclidean norm of the KKT conditions [20] is higher than a maximum tolerance defined with $\omega_{\max} = 10^{-3}$. In those cases, the ratio of performance is set to a

value higher than the maximum possible. This way we can identify the percentage of problems where the solver is unable to produce accurate designs. More details about the impact of this threshold can be found in [23].

There are three main aspects to consider in performance profiles:

1. The probability of being the best solver is read at $\tau = 1$ (or $\tau = 0$ for logarithmic scales).

2. The performance of any good solver must increase for ratio values close to 1. The function ρ_s is expected to grow fast for these solvers.

3. When τ is large enough, performance profiles represent the percentage of problems where the solver is able to obtain solutions. Thus, the ability of a solver to produce designs with a KKT error lower or equal to $\omega_{\max}$ is represented at large values of τ. Throughout this article, this probability of success is defined as robustness.

We refer to [13] and [23] for details on performance profiles in numerical optimization in general and structural topology optimization in particular, respectively.

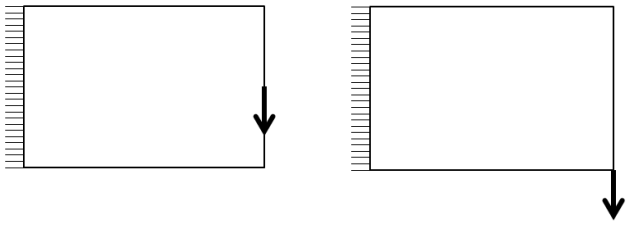
4.2 Test set

Performance profiles require an illustrative and large test set of problem instances in order to produce clear and fair results. The test set consists of 225 different instances for minimum compliance problems and 150 for compliant mechanism design problems. The design domains are represented in Figures 1 and 2, for minimum compliance and compliant mechanism design problems, respectively. These design domains and load cases are found in numerous articles related to structural optimization such as [6], [3], [9], [12], [18], [27], [35], [27], and [26] among many others. Moreover, this test set is described in detail in [23].

The upper volume fraction V in (P^c) takes the values 0.1, 0.2, 0.3, 0.4, or 0.5, respectively. For the mechanism design problem (P^m) these values are set to 0.2, 0.3, or 0.4, respectively. Moreover, we define 5 different mesh sizes (with 20, 40, 60, 80, and 100 elements per length ratio). Our numerical experiments thus only consider medium-size problem instances. The largest minimum compliance problem contains 40,000 elements, the largest mechanism design problem contains 20,000 design variables.

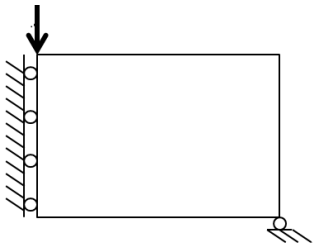
5 Numerical experiments with continuation methods

This section contains the performance profiles for the three different implementations of the continuation method (denoted C_1-GCMMA, C_2-GCMMA, and C_3-GCMMA) together with the classical formulation of topology optimization problems solved using GCMMA. In the numerical experiments we refer to this formulation as NC-GCMMA,



(a) Michell problem (with 1×1, 2×1, 3×1 length ratios)

(b) Cantilever problem (with 2×1, 4×1 length ratios)



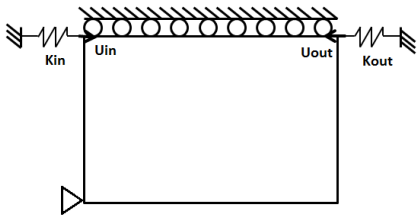
(c) MBB problem (with 2×1, 4×1, 1×2, 1×4 length ratios)

Figure 1: Michell, Cantilever, and MBB design domains, boundary conditions and external load definitions that are collected in the benchmark library for minimum compliance problems (from [23]).

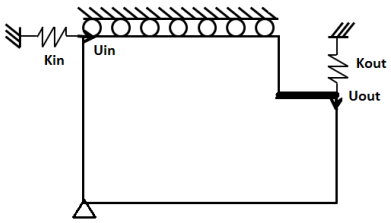
i.e. no continuation with the problem solved by GCMMA. The numerical experiments were computed on an Intel Xeon 10-core CPUs running at 2.8 GHz and with 64 GB memory.

It is expected, due to the implementation and the choices of parameters, that the number of iterations for C_1-GCMMA will be larger than for C_2-GCMMA. In turn, this will be larger than for C_3-GCMMA. However, we would like to observe how they perform in terms of objective function values since C_1-GCMMA uses small increments of the penalization parameter, C_2-GCMMA uses a nonlinear updating strategy, and C_3-GCMMA uses large increment step ($\Delta p = 0.3$).

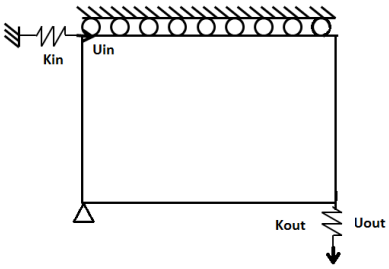
Figure 3 shows the performance profiles for the test set of 225 minimum compliance problem instances. The three continuation implementations obtain very similar objective function values, see Figure 3a. For instance, at a factor of $\tau = 1.05$ there is a difference of only 2% between them, and the percentage of success is higher than 70%. In contrast, the NC-GCMMA curve falls below all the continuation approaches. NC-GCMMA is also less robust. It has about a 5% less chances to obtain a design with a tolerance lower or equal to $\omega_{\max}$ than any of the continuation methods. Figure 3b shows that there are large differences between the performances of the solvers with respect to the number of



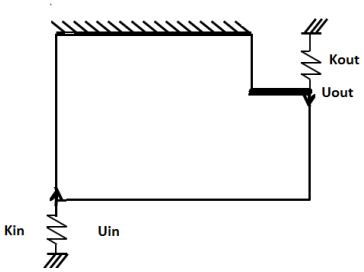
(a) Force inverter problem with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.001$.



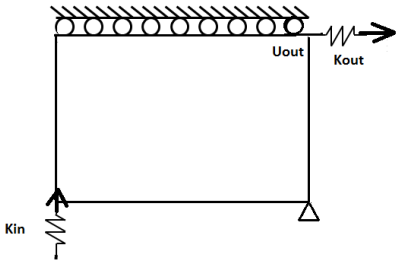
(b) Compliant gripper problem with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.005$.



(c) Amplifier problem with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.005$.

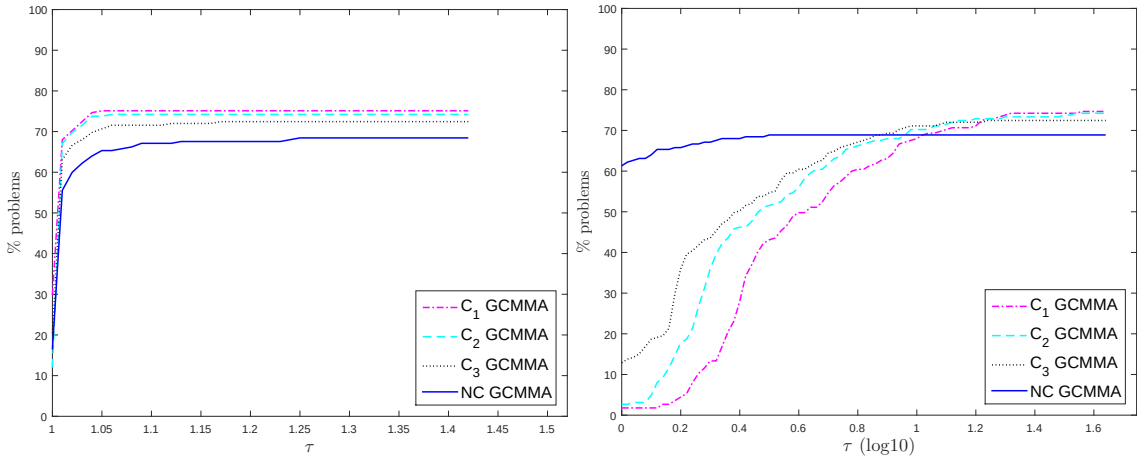


(d) Compliant lever problem with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.005$.



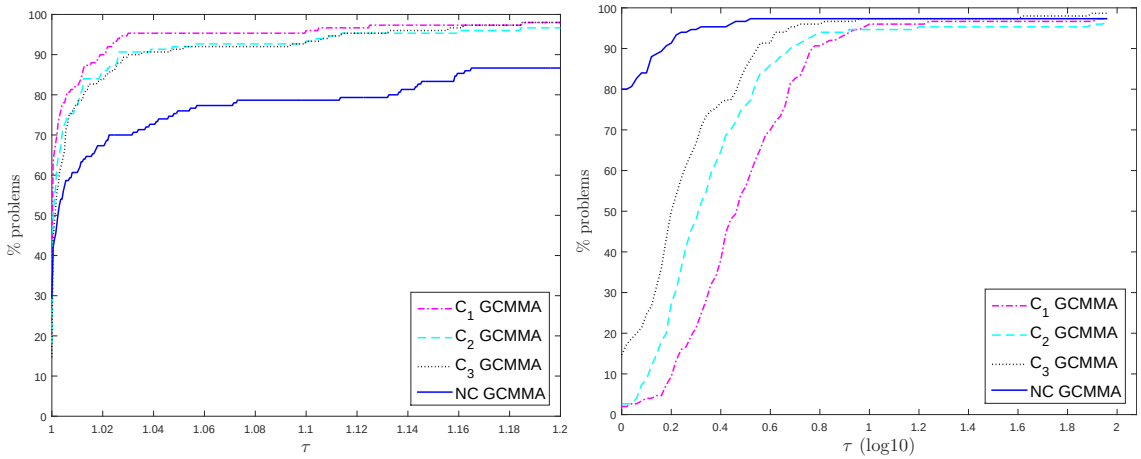
(e) Crimper problem with $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.05$

Figure 2: Design domains with boundary conditions and external loads definition for the benchmark library of compliant mechanism design problems with 1×1 and 2×1 length ratios (from [23]).



(a) Objective function value (b) Number of iterations

Figure 3: Performance profiles for various continuation methods combined with GCMMA and a classical formulation of GCMMA on a test set of 225 of minimum compliance problems (P^c). The performance is measured by the objective function value (3a) and the accumulated number of GCMMA iterations (3b).



(a) Objective function value (b) Number of iterations

Figure 4: Performance profiles for the continuation methods combined with GCMMA and a classical formulation of GCMMA on a test set of 150 compliant mechanism design problems (P^m). The performance is measured by the objective function value (4a) and the accumulated number of GCMMA iterations (4b). Note that Figure 4a represents a small range of the parameter τ.

optimization iterations, as expected. In some cases it would be impractical to use one of our continuation methods due to the large amount of iterations required.

Correspondingly, Figure 4 shows the performance profiles for the test set of 150 compliant mechanism design problems. Similar to the minimum compliance problems, the three schemes of the continuation method performs better than the classical NC-GCMMA. At $\tau = 1.02$ the chances to find a KKT point for all the continuation approaches are higher than 85% while for NC-GCMMA the chance is lower than 70%. Additionally, the performance measured in the number of optimization iterations is much worse for continuation methods than for the classical approach.

Note that since the maximum number of iterations in the compliant mechanism design problem is set to 3000, GCMMA is able to produce more designs with KKT error lower than or equal to $\omega_{\max}$ than for minimum compliance problems where the maximum number of optimization iterations is 1000.

Balancing the obtained objective function values and the required number of iterations, C_3-GCMMA (with large increments) can be considered as the winner implementation of the three suggested continuation methods.

Finally, from the study of the performance of continuation methods for minimum compliance and mechanism design problems, we conclude the following.

1. The use of continuation methods help to obtain a design with better objective function value. However, it is (of course) still possible to end in local minima.

2. The suggested continuation methods require a large number of optimization iterations and therefore, large computational time.

3. The implementation of the continuation method does not require very small increment. The experiments reflect that $\Delta p = 0.3$ produce similar results as $\Delta p = 0.1$.

6 A new approach to continuation methods

The previous numerical experiments evidence that solving a sequence of problems where the material penalization parameter is increasing from $p_{\min}$ to $p_{\max}$ helps to produce better designs. However, the main drawback of our continuation method is the large number of iterations required. In some situations this makes continuation intractable in practice.

We propose an automatic continuation method where the material penalization parameter p is considered as an additional variable of the optimization problem. Thus, the solver simultaneously finds the design and increases the material penalization variable from the initial value to $p_{\max}$. In contrast to continuation methods, the

automatic continuation approach only needs to solve one optimization problem. This gives the potential of possible reductions in the total number of iterations required while retaining some of the positive effects of continuation.

The variable p is initialized as $p_{\min}$, and then an additional nonlinear constraint $g(p)=0$ ensures that the problem is feasible only when $p=p_{\max}$, i.e. the constraint is violated for any values in $[p_{\min},p_{\max})$. Thus, we force the material penalization variable to increase from the initial value to $p_{\max}$.

6.1 Formulation of the automatic continuation problem

The new automatic continuation formulation for minimum compliance (P_{ac}^{c}) and for compliant mechanism design (P_{ac}^{m}) problems in nested formulations are

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n,p\in\mathbb{R}}{\text{minimize}} & \mathbf{u}^T(\mathbf{t},p)\mathbf{K}(\mathbf{t},p)\mathbf{u}(\mathbf{t},p) \\ \text{subject to} & \mathbf{v}^T\mathbf{t}\leq V \\ & g(p)=0 \\ & \mathbf{0}\leq\mathbf{t}\leq\mathbf{1} \\ & p_{\min}\leq p\leq p_{\max}, \end{array} \qquad (P_{ac}^{c})$$

and

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n,p\in\mathbb{R}}{\text{maximize}} & \mathbf{l}^T\mathbf{u}(\mathbf{t},p) \\ \text{subject to} & \mathbf{v}^T\mathbf{t}\leq V \\ & g(p)=0 \\ & \mathbf{0}\leq\mathbf{t}\leq\ \mathbf{1} \\ & p_{\min}\leq p\leq p_{\max}. \end{array} \qquad (P_{ac}^{m})$$

Here, p now refers to the material penalization variable, $\mathbf{u}(\mathbf{t},p)=\mathbf{K}^{-1}(\mathbf{t},p)\mathbf{f}$, $p_{\min}=1$ and $g(p)$ is the automatic penalization constraint (cf. below).

In this new formulation, the sensitivities are affected by the variable p, which is involved in the stiffness matrix. For any material interpolation scheme, the stiffness matrix can be defined as

$$\mathbf{K}(\mathbf{t},p)=\sum_{e=1}^{n}E(\mathbf{t},p)\mathbf{K}_e. \qquad (11)$$

It follows that

$$\left(\frac{\partial\mathbf{K}}{\partial p}\right)_i=\left(\frac{\partial E}{\partial p}\right)_i\mathbf{K}_e. \qquad (12)$$

More specifically,

$$\left(\frac{\partial E(\mathbf{t},p)}{\partial p}\right)_i=\begin{cases}\begin{cases}(E_1-E_v)t_i^p ln(t_i) & \text{for } t_i>0\\ 0 & \text{for } t_i=0\end{cases} & \text{(SIMP)}\\ (E_1-E_v)\dfrac{(t_i-1)t_i}{(1+(p-1)(1-t_i))^2} & \text{(RAMP).}\end{cases} \qquad (13)$$

Equation (13) for the SIMP approach is defined using the L'Hôpital rule.

It is important to highlight that the automatic continuation method only introduces one additional variable and one additional nonlinear and univariate constraint. Thus, the computational cost of the method is not significantly increased compared to the classical formulation. In contrast, the number of iterations and the final designs can potentially be improved.

6.2 Obtaining an adequate automatic penalization constraint

Our proposed automatic continuation formulation contains a nonlinear equality constraint $g(p) = 0$ to force the material penalization variable to increase from $p_0 = p_{\min}$ to $p_{\max}$. The constraint must satisfy that

$$g(p) = 0 \Longleftrightarrow p = p_{\max}. \tag{14}$$

In addition, for any value of $p < p_{\max}$ the constraint must be infeasible, i.e.

$$g(p) > 0 \quad \forall p < p_{\max}.$$

It is preferable that the solver uses some iterations until the constraint is satisfied. Methods such as the primal dual interior point solver [15] in IPOPT [34] or the Sequential Quadratic Programming [7] in SNOPT [16], linearize the constraints as

$$g(p_k) + g'(p_k)(p - p_k) = 0. \tag{15}$$

We now assume a linearization of the constraint as in (15), and that a line-search strategy is used in the update of the iterates. The variable p for any iteration k of the solver is updated using the search direction Δp_k and the step length $0 < \alpha_k \leq 1$ as

$$\begin{aligned} &p_{k+1} = p_k + \alpha_k \Delta p_k, \\ &\text{with} \\ &\Delta p_k = (p - p_k) = -\frac{g(p_k)}{g'(p_k)} > 0. \end{aligned} \tag{16}$$

Hence, the gradient of the automatic penalization constraint must be negative, since

$$\forall \quad p \in [p_{\min}, p_{\max}), \quad g(p) > 0 \Rightarrow g'(p) < 0. \tag{17}$$

Finally, the constraint should be nonlinear in order to use some iterations until $p = p_{\max}$.

There are many possibilities to define a function $g(p)$ satisfying the requirements outlined in (14) and (17). Some of them are

- $g_1(p) = e^{-\mu(p - p_{\max})} - 1$
- $g_2(p) = e^{\mu(p/p_{\max} - 1)^2} - 1$

- $g_3(p) = e^{\mu(p/p_{\max}-1)^2} - \frac{p}{p_{\max}}$
- $g_4(p) = e^{\mu(p/p_{\max}-1)^2} - (\frac{p}{p_{\max}})^2$
- $g_5(p) = (\frac{p}{p_{\max}})^{-q} - 1.$

Here, the parameters $\mu \geq 1$ and $q \geq 1$ are given by the user. The main difference between these possibilities is how the variable p is increased, by the solver, until $p = p_{\max}$.

As an illustrative example, Figure 5 shows how the idealized iterates for the material penalization variable behaves for different automatic penalization constraint functions. The underlying assumptions for the figure are *(i)* the step length α is one for all iterates, and *(ii)* the linearization of the automatic penalization constraint is the only thing that determines the step length and the search direction. These requirements do, in practice, normally not occur. The figure thus, shows the minimum number of iterates required to reach the requested value of the material penalization parameter, if the constraints are linearized. The scheme is thus, both solver dependent and problem dependent, and the material penalization sequence is not pre-determined.

We observe that $g_1(p)$ increases p almost linearly. Moreover as μ increases, it requires more iterations to reach $p_{\max}$. The opposite behaviour is observed in $g_2(p)$. When μ increases, fewer iterations are required. It also needs many iterations to change for values of p close to $p_{\max}$. In contrast, in $g_5(p)$ the parameter p remains closer to 1 for many iterations and then it increases rapidly to $p_{\max}$. Finally, the behaviour of $g_3(p)$ is similar to $g_4(p)$ and is also closely related to $g_2(p)$.

The automatic penalization constraint should require some iterations to increase p from $p = p_{\min}$ to $p = p_{\max}$, but at the same time, it is important not to spend too many iterations such that the performance of the solver is deteriorated. A good compromise between these requirements could be $g_2(p)$ with $\mu = 2$ or $g_5(p)$ with $q = 10$.

6.3 Implementation

The choice of the automatic penalization constraint is based on the linearization of the constraints. In order to ensure a good behaviour, we decided to use a solver that linearizes the constraints. GCMMA, in its current implementation, is not a good candidate since the constraints are dealt with using convex and separable approximations. The interior point method in IPOPT [34] is chosen for our implementation. We experienced better performance with IPOPT than with the SQP method in SNOPT while benchmarking solvers for structural topology optimization problems in [23]. In the implementation, the automatic continuation constraint is treated as an inequality constraint, i.e. $g(p) \leq 0$. This inequality, together with the bounds on p, is equivalent to the equality constraint while it gives more freedom to the solver. In addition, both the SIMP and the RAMP material interpolation schemes are

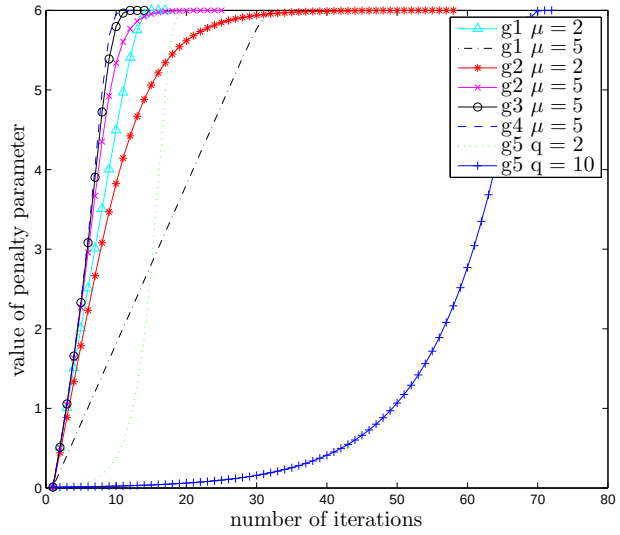


Figure 5: Example of an idealized behaviour of the material penalization variable for different constraint function assuming a fix step length $\alpha = 1$. The figure thus shows the minimum number of iterations to reach the requested value.

studied. Throughout this paper we denote by AC-IPOPT the automatic continuation formulation with the sub-problems solved by IPOPT. Similarly, as Section 5 describes, NC-IPOPT and C_3-IPOPT refer to the classical formulation and the continuation approach solved by IPOPT, respectively.

Moreover, different constraints for the automatic continuation were numerically tested using IPOPT to select the parameters for this specific solver. The best results were obtained with $g_2(p)$ with $\mu = 5$ for the SIMP interpolation scheme, and $g_5(p)$ with $q = 15$ for the RAMP scheme. A small computational study suggested that IPOPT performs better with an automatic penalization constraint that requires several iterations to achieve $p \approx p_{\max}$. Differences between SIMP and RAMP are expected since the range of possible values and the initial value of p are different.

Section 7 shows the comparative study of an automatic continuation approach (AC-IPOPT), the classic formulation (NC-IPOPT), and a continuation method (C_3-IPOPT). For this numerical experiment, the topology optimization problem is formulated in the same way as in the previous section, i.e. we use the SIMP interpolation scheme together with a density filter. Moreover, we also include results using the RAMP scheme [30] with $p_{\max} = 7$, which visually gives similar results as the SIMP interpolation with $p_{\max} = 3$, see e.g. [17].

Although it is preferable to tune as few parameters as possible, we experienced better performance of IPOPT with some parameters set differently from the default values. Table 2 collects the information of these parameters tuned for the three approaches (NC-IPOPT, AC-IPOPT and C_3-IPOPT). The Hessian of the Lagrangian

Table 2: Parameters tuned in IPOPT. The table contains the name of the parameter, the default value, and the new value used in the numerical experiments.

Parameter	**Default**	**New value**
`mu strategy`	monotone	adaptive
`limited memory max history`	6	25
`max iter`	3,000	1,000/3,000
`tol`	10^{-8}	10^{-6}
`constr viol tol`	10^{-4}	10^{-8}
`nlp scaling method`	gradient based	none
`obj scaling factor`	1	10^4 (P^m_{ac})

for the nested formulation of both the minimum compliance (P^c), (P^c_{ac}) and the compliant mechanism design (P^m), (P^m_{ac}) problems is computational expensive. Thus, a limited memory BFGS is used to approximate the Hessian in IPOPT for all approaches. The parameter `limited memory max history` determines the number of the most recent iterations that are taken into account for the BFGS approximation. We experience that with more history information than the default value (6), IPOPT performs better. A reasonable number to avoid too many iterations is 25.

It is well known in the optimization community that the performance of the solvers is highly affected by the scaling of the problems. Particularly, for topology optimization, IPOPT performs better without scaling the problem (see e.g. [23]). Thus, the parameter `nlp scaling method` is set to `none`. However, the performance of IPOPT for compliant mechanism design problems improves if the objective function is scaled by a factor given by the parameter `obj scaling factor`. The objective function value of these problems is negative and close to zero, thus, a scaling of 10^4 helps IPOPT to produce faster and more accurate results.

Moreover, in a preliminary study of the performance of IPOPT for different values of parameters, we observe that IPOPT is highly affected by the election of strategy for updating the barrier parameter (`mu strategy`). The adaptive strategy performs better than the monotone strategy for topology optimization problems, see e.g. [23].

Additionally, for IPOPT, the volume constraint is scaled in both minimum compliance and compliant mechanism design problems with a factor of $\frac{1}{\sqrt{n}}$ and $\frac{1}{n}$, respectively [23].

Finally, Table 3 gathers the values of some characteristic parameters involved in structural topology optimization problems as well as the parameters defined for the continuation approach. Section 5 revealed that continuation methods work well with $\Delta p = 0.3$ and $\theta = 0.5$ (continuation method type 3). In contrast with the implementation of the continuation approach in Section 3.1, and as IPOPT is able to produce designs with better accuracy (lower optimality tolerance) than GCMMA (see [23]), both the ω_0 and $\omega_{\min}$ tolerances, are set to smaller values than before.

Table 3: Values of some characteristic parameters of topology optimization problems.

Parameter	**Description**	**Value**
$\mathbf{t}_0$	Starting point	$0.5\mathbf{e}$
E_1	Solid Young's modulus (P^c), (P^c_{ac})	$10^{3\text{e}}$
E_1	Solid Young's modulus (P^m), (P^m_{ac})	1
E_v	Void Young's modulus (P^c), (P^c_{ac})	1
E_v	Void Young's modulus (P^m), (P^m_{ac})	10^{-2}
ν	Poisson's ratio	0.3
p_0	Initial penalization parameter value	1
$p_{\max}$	Final penalization parameter value	3/7
Δp_k	Step increment of penalization parameter	0.3
θ	Factor of reducing the tolerance at each step	0.5
ω_0	Initial tolerance for optimality (`tol`)	10^{-4}
$\omega_{\min}$	Final tolerance for optimality (`tol`)	10^{-6}

7 Numerical experiments with automatic continuation

This section is focused on the numerical experiments with automatic continuation. The performance of AC-IPOPT is compared to the classical formulation of the problem (NC-IPOPT) and the continuation method C_3-IPOPT.

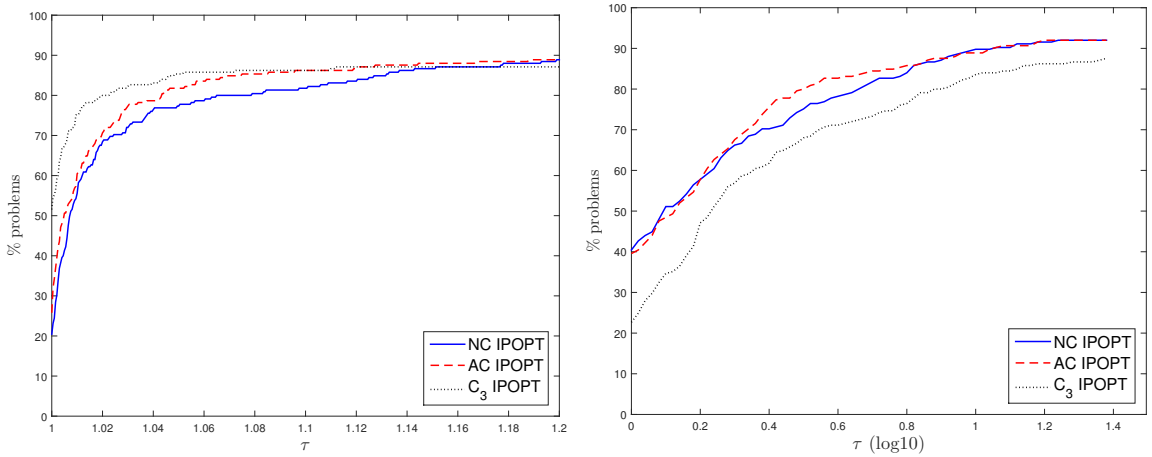
The computational time is an important aspect when comparing different solvers. However, our experiments are run on a multi-core processor in a cluster and the computational time required for the solvers may vary with the load on the cluster. Nevertheless, in structural topology optimization, one of the most expensive steps is the assembly of the stiffness matrix. Thus, the computational time required for the solvers should be proportional to the number of stiffness matrix assemblies. In this section, the performance of the solvers is compared not only with the objective function value and the number of iterations, but also with the number of stiffness matrix assemblies and the computational time.

The section is divided into two parts with the numerical results for the SIMP and the RAMP material interpolation schemes, respectively.

7.1 SIMP material interpolation scheme

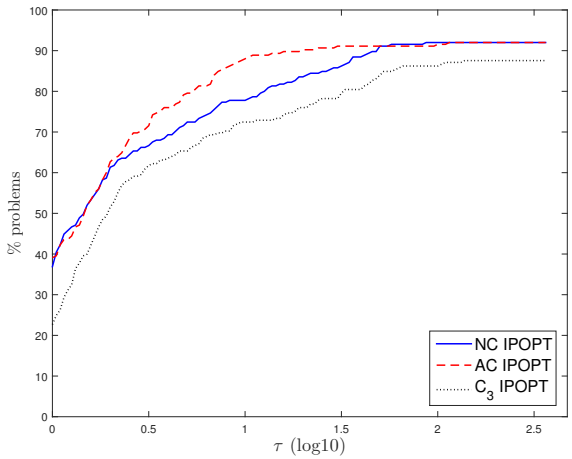
Figure 6 shows the performance profiles for minimum compliance problems based on the SIMP material interpolation scheme on a test set of 225 problem instances. The use of p as a new variable helps to produce designs with better objective function values than

[e]The choices of E_v and E_1 have changed compared to Table 1 since the performance of the solvers is highly affected by these values. The Young's modulus contrast E_1/E_v remains the same, but the scaling of the problem has changed, see [23] for more details.

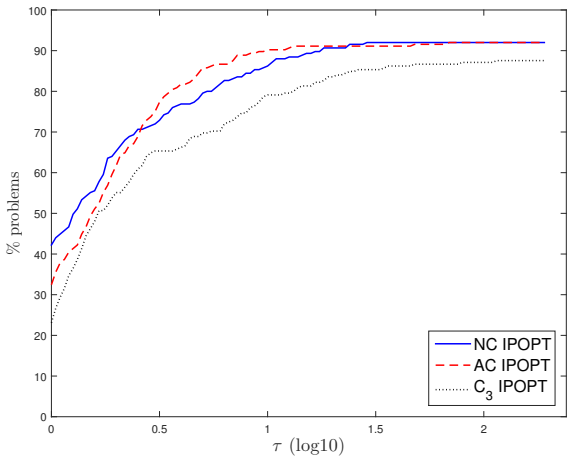


(a) Objective function value

(b) Number of iterations

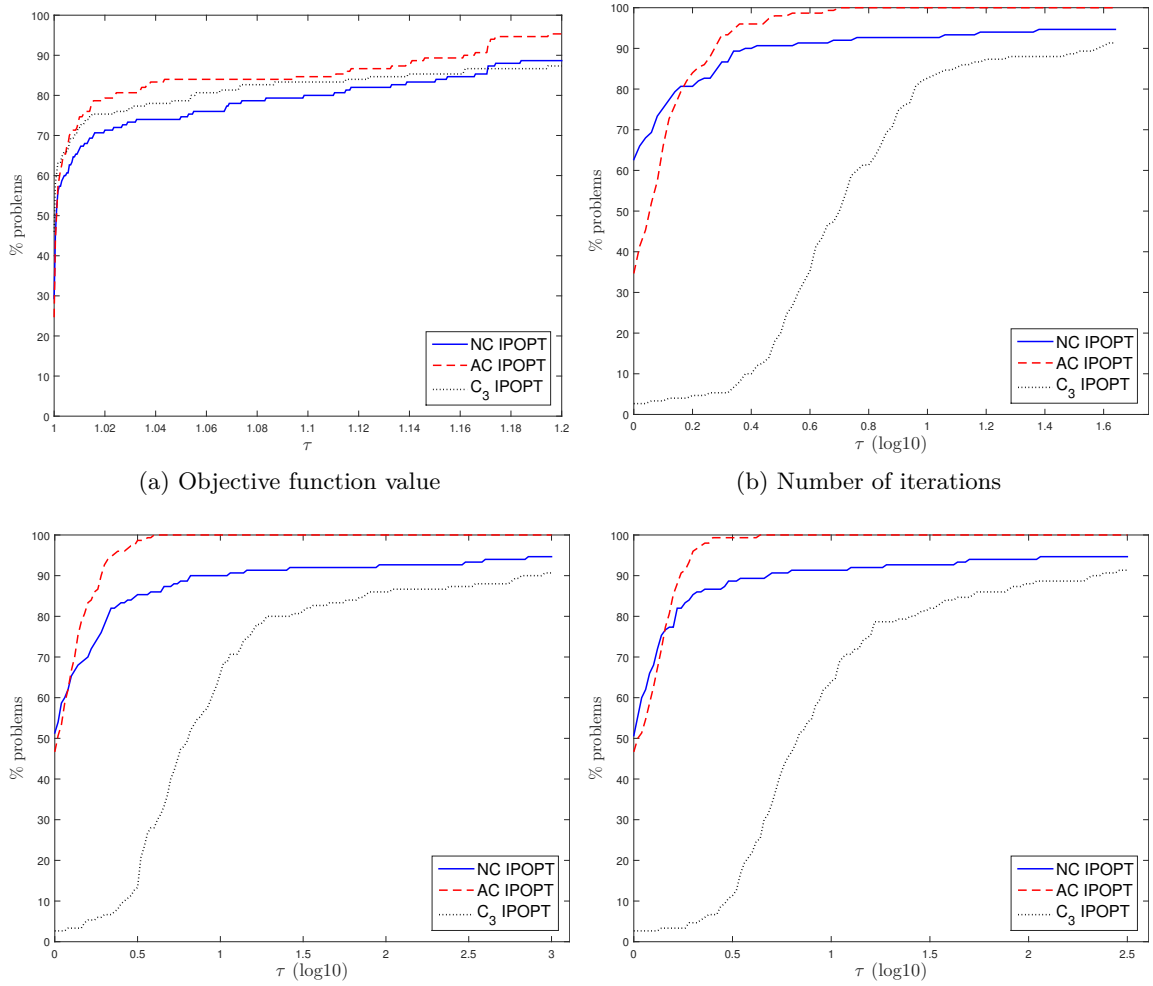


(c) Stiffness matrix assemblies



(d) Computational time

Figure 6: Performance profiles for NC-IPOPT, C_3-IPOPT and AC-IPOPT on a test set of 225 of minimum compliance problems using SIMP. The performance is measured by the objective function value (6a), the (accumulated) number of IPOPT iterations (6b), the number of stiffness matrix assemblies (6c), and the computational time (6d).



(a) Objective function value

(b) Number of iterations

(c) Stiffness matrix assemblies

(d) Computational time

Figure 7: Performance profiles for NC-IPOPT, C_3-IPOPT and AC-IPOPT on a test set of 150 compliant mechanism design problems using SIMP. The performance is measured by the objective function value (7a), the (accumulated) number of IPOPT iterations (7b), the number of stiffness matrix assemblies (7c), and the computational time (7d).

with a fixed material penalization parameter. Moreover, AC-IPOPT generally requires fewer iterations than both NC-IPOPT and C_3-IPOPT.

Figure 6a shows that even if C_3-IPOPT has more chances to win over the other solvers in terms of objective function value (with a 50% probability), AC-IPOPT reaches the same performance at a small τ. At $\tau = 1.08$, both AC-IPOPT and C_3-IPOPT have 86% chances to produce a sufficiently accurate KKT point. In contrast, the distance of NC-IPOPT to achieve the same percentage is larger. Furthermore, Figure 6b shows that both NC-IPOPT and AC-IPOPT have a 40% probability to produce a KKT point using the least number of iterations. This is in contrast to C_3-IPOPT that has a 23% chance. The performance of AC-IPOPT becomes more competitive than NC-IPOPT if we extend our interest to larger τ.

Correspondingly, the performance profiles for compliant mechanism design problems are collected in Figure 7. Figure 7a shows that C_3-IPOPT obtains the best designs for 45% of the problems. However, AC-IPOPT makes good progress, and at $\tau = 1.0025$ (i.e. very close to the best design) it performs as well as C_3-IPOPT, with a 62% success rate. From that point, AC-IPOPT outperforms the rest of the approaches. Observe that the performance of C_3-IPOPT does not improve the objective function value compared to NC-IPOPT. We can conclude that in general, IPOPT (in all of the approaches) obtains very good designs for compliant mechanism design problems.

One of the main drawbacks of these problems is the large amount of iterations required to obtain good designs. Figure 7b reflects that the use of C_3-IPOPT becomes impractical because of this. One more time, the performance of AC-IPOPT is better than NC-IPOPT and C_3-IPOPT measured by the number of optimization iterations.

The performance of the solvers measured by the number of stiffness matrix assemblies (Figure 6c and 7c) is equivalent to the performance based on the number of iterations. In addition, Figures 6d and 7d show the performance profiles for the computational time. The general performance of the three solvers is, as expected, very similar based on the performance profiles for the number of iterations and the number of stiffness matrix assemblies.

For both minimum compliance and compliant mechanism design problems, the continuation approach produces more winners (measured in objective function value) than the classical and the automatic continuation approach. However, at a very small distance to the best found design, AC-IPOPT outperforms NC-IPOPT. In contrast, C_3-IPOPT requires more iterations and stiffness matrix assemblies than AC-IPOPT. From Figures 6 and 7 we can conclude that the performance of AC-IPOPT rapidly increases and it becomes a very good competitor. For compliant mechanism design problems, AC-IPOPT outperforms C_3-IPOPT in both the number of stiffness matrix assemblies and the objective function value. It is important to point out, that there is a need to balance between obtaining the best designs and consuming few function

evaluations. Thus, AC-IPOPT is a very good alternative to C_3-IPOPT. Moreover, we suggest a new way of solving topology optimization problems where the final design is considerable better and it converges faster than with a fixed penalization parameter.

7.2 RAMP material interpolation scheme

We also briefly examine the performance of automatic continuation coupled with the RAMP interpolation scheme for both minimum compliance and compliant mechanism design problems. The results are shown in Figure 8 and Figure 9, respectively.

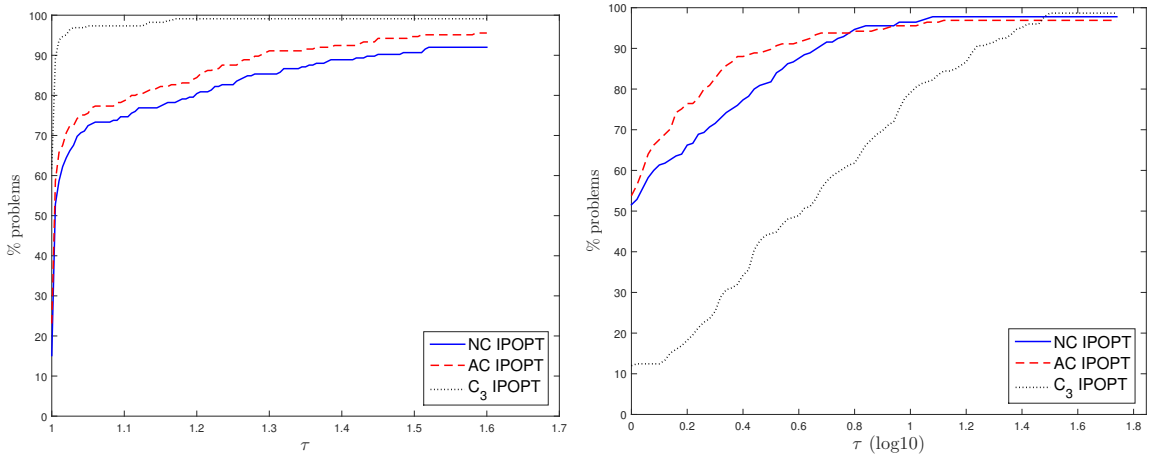
In this case, the performance of AC-IPOPT measured by the objective function value is more prominent for minimum compliance problems (as illustrated in Figure 8a) than for compliant mechanism design (see Figure 9a) where it performs similar to NC-IPOPT. Nevertheless, for the latter problems, the improvements compared to NC-IPOPT and C_3-IPOPT in the number of iterations make AC-IPOPT a very good alternative. For compliant mechanism design problems, it is more important to reduce the computational time and the number of iterations required than producing good designs, since IPOPT generally obtains very good designs for all approaches.

Similarly to the previous section, the performance of the methods measured in stiffness matrix assemblies and computational time is comparable to the number of iterations.

8 Conclusions

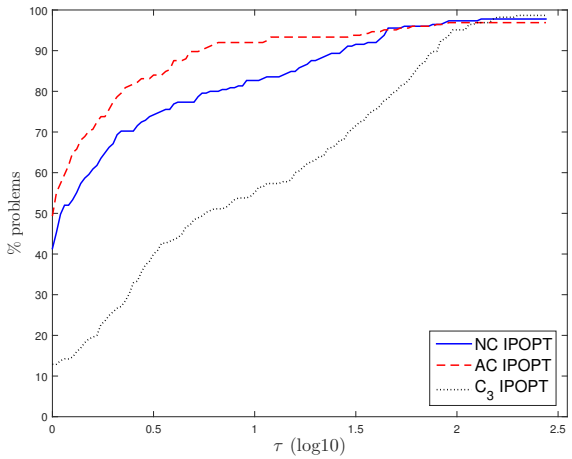
In structural topology optimization, it is common to use continuation methods in the penalization parameter of the material interpolation scheme to produce better designs. These continuation methods solve sequences of problems with increasing values of the material penalization parameter p. In this article, we presented a benchmarking study of continuation methods using GCMMA on a test set of 225 minimum compliance and 150 compliant mechanism design problems, in order to assess the general performance of this approach.

The numerical results clearly indicate that the continuation methods implemented in this article generally (but not always) obtain better designs than using a fixed material penalization parameter value. Moreover, the performance of our continuation methods is not highly affected by the increment step selected. The latter result is preliminary and requires additional investigations. However, the suggested continuation approaches are computationally expensive, and for large-scale problems or for industrial applications, our implementation of continuation methods is impractical. We present a new alternative, called automatic continuation, where the parameter of the material interpolation scheme is an explicit variable of the problem. In contrast to other continuation methods, the proposed automatic continuation approach only solves one optimization problem.

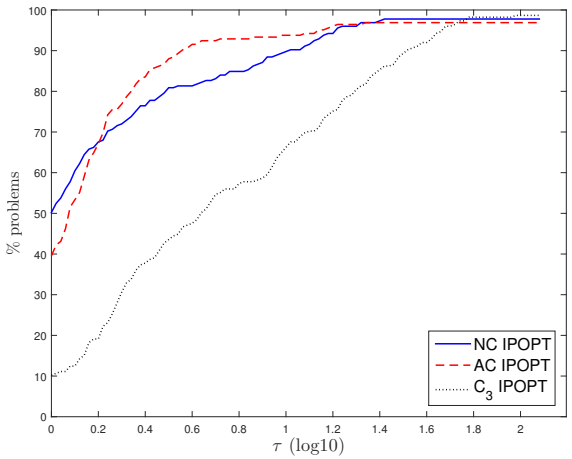


(a) Objective function value

(b) Number of iterations

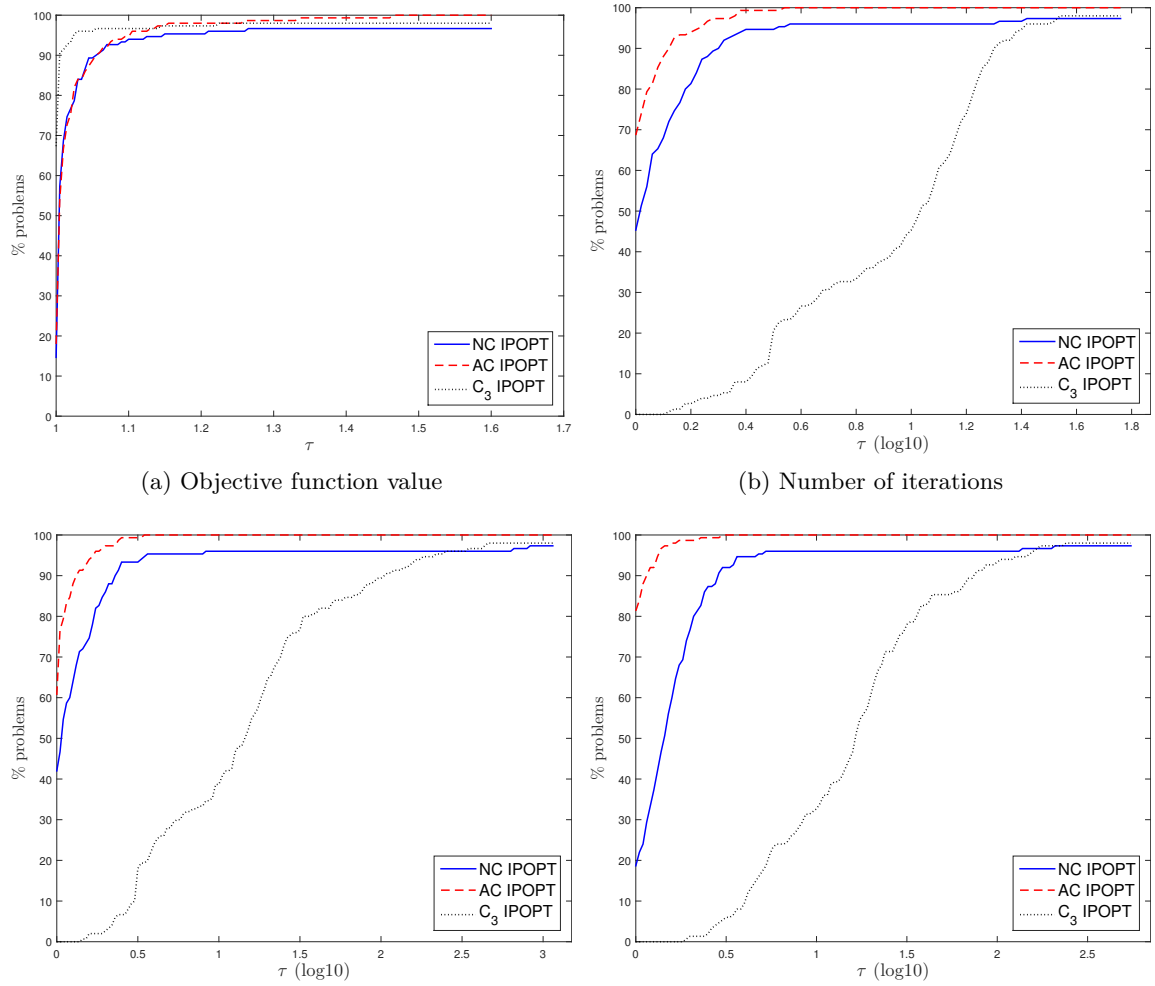


(c) Stiffness matrix assemblies



(d) Computational time

Figure 8: Performance profiles for NC-IPOPT, C_3-IPOPT and AC-IPOPT on a test set of 225 of minimum compliance problems using RAMP. The performance is measured by the objective function value (8a), the (accumulated) number of IPOPT iterations (8b), the number of stiffness matrix assemblies (8c), and the computational time (8d).



(a) Objective function value

(b) Number of iterations

(c) Stiffness matrix assemblies

(d) Computational time

Figure 9: Performance profiles for NC-IPOPT, C_3-IPOPT and AC-IPOPT on a test set of 150 compliant mechanism design problems using RAMP. The performance is measured by the objective function value (9a), the (accumulated) number of IPOPT iterations (9b), the number of stiffness matrix assemblies (9c), and the computational time (9d).

The automatic continuation approach is a good alternative to the outlined continuation methods both in terms of quality of designs and the computational effort. Even though the suggested continuation approach obtains better designs for some problems, the computational time required is high in comparison to the automatic continuation approach. Additionally, both the objective function value and the number of iterations are reduced compared to the classical formulation with fixed penalty parameter. Thus, this new formulation is a good replacement of both continuation methods and the classical formulation.

This study opens many possibilities for further research and developments of new implementations of continuation methods and new alternatives.

Acknowledgements

We would like to thank Professor Krister Svanberg at KTH in Stockholm for providing us with the MATLAB implementation of GCMMA. We extend our sincere thanks to two reviewers and the editor for providing many comments and suggestions that improved the quality of this article.

References

[1] G. Allaire and G. A. Francfort. A numerical algorithm for topology and shape optimization. In *Topology design of structures*, pages 239–248. Kluwer Academic Publishers, 1993.

[2] G. Allaire and R. V. Kohn. Topology optimization and optimal shape design using homogenization. In *Topology design of structures*, pages 207–218. Kluwer Academic Publishers, 1993.

[3] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[4] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):192–202, 1989.

[5] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9–10):635–654, 1999.

[6] M. P. Bendsøe and O. Sigmund. *Topology optimization: Theory, methods and applications*. Springer, 2003.

[7] P. T. Boggs and J. W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

[8] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[9] T. E. Bruns. A reevaluation of the SIMP method with filtering and an alternative formulation for solid void topology optimization. *Structural and Multidisciplinary Optimization*, 30(6):428–436, 2005.

[10] T. Buhl, C. B. W. Pedersen, and O. Sigmund. Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19(2):93–104, 2000.

[11] J. D. Deaton and R. V. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.

[12] S. R. Deepak, M. Dinesh, D. K. Sahu, and G. K. Ananthasuresh. A comparative study of the formulations and benchmark problems for the topology optimization of compliant mechanisms. *Journal of Mechanisms and Robotics*, 1(1), 2009.

[13] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[14] C. S. Edwards, H. A. Kim, and C. J. Budd. An evaluative study on ESO and SIMP for optimising a cantilever tie beam. *Structural and Multidisciplinary Optimization*, 34(5):403–414, 2007.

[15] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

[16] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large -Scale Constrained Optimization. *SIAM Journal on Optimization*, 47(4):99–131, 2005.

[17] C. F. Hvejsel and E. Lund. Material interpolation schemes for unified topology and multi-material optimization. *Structural and Multidisciplinary Optimization*, 43(6):811–825, 2011.

[18] G. K. Lau, H. Du, and M. K. Lim. Use of functional specifications as objective functions in topological optimization of compliant mechanism. *Computer Methods in Applied Mechanics and Engineering*, 190(34):4421–4433, 2001.

[19] L. Li and K. Khandelwal. Volume preserving projection filters and continuation methods in topology optimization. *Engineering Structures*, 85:144–161, 2015.

[20] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[21] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. *Mathematical programming*, 14(1):41–72, 1978.

[22] J. Petersson and O. Sigmund. Slope constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 41(8):1417–1434, 1998.

[23] S. Rojas-Labanda and M. Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization, In print*, 2015. DOI: 10.1007/s00158-015-1250-z.

[24] G. I. N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37(3):217–237, 2008.

[25] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural Optimization*, 4(3–4):250–252, 1992.

[26] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, 25(4):492–526, 1997.

[27] O. Sigmund. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica*, 25(2):227–239, 2009.

[28] O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[29] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(2):68–75, 1998.

[30] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, 2001.

[31] M. Stolpe and K. Svanberg. On the trajectories of penalization methods for topology optimization. *Structural and Multidisciplinary Optimization*, 21(2):128–139, 2001.

[32] K. Svanberg. The method of moving asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

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

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

[35] F. Wang, B. S. Lazarov, and O. Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43(6):767–784, 2011.

[36] R. Watada and M. Ohsaki. Continuation approach for investigation of non-uniqueness of optimal topology for minimum compliance. In *Proceedings of 8th World Congress on Structural and Multidisciplinary Optimization, June 1–5, Lisbon, Portugal*, 2009.

[37] R. Watada, M. Ohsaki, and Y. Kanno. Non-uniqueness and symmetry of optimal topology of a shell for minimum compliance. *Structural and Multidisciplinary Optimization*, 43(4):459–471, 2011.

[38] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):309–336, 1991.

8

Article III: An efficient second-order SQP method for structural topology optimization

In review:
Rojas-Labanda, S. and Stolpe, M.: An efficient second-order SQP method for structural topology optimization

An efficient second-order SQP method for structural topology optimization*

Susana Rojas-Labanda[+] and Mathias Stolpe[+]

[+]DTU Wind Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark. E-mail: srla@dtu.dk, matst@dtu.dk

Abstract

This article presents a Sequential Quadratic Programming (SQP) solver for structural topology optimization problems (TopSQP). The implementation is based on the general SQP method proposed in [40] called SQP+. The topology optimization problem is based on a density approach and thus, is classified as a nonconvex problem. More specifically, the method is designed for the classical minimum compliance problem with a constraint on the volume of the structure. The sub-problems are defined using exact second-order information and they are reformulated based on the specific mathematical properties of the problem to significantly improve the efficiency of the solver.

The performance of the TopSQP solver is compared to the special-purpose structural optimization method, the Globally Convergent Method of Moving Asymptotes (GCMMA) and the two general nonlinear solvers IPOPT and SNOPT. Numerical experiments on a large set of benchmark problems show great performance of TopSQP in terms of number of function evaluations. In addition, the use of exact second-order information helps to decrease the objective function value.

Keywords: Topology optimization, Sequential Quadratic Programming, Minimum compliance, Second-order method, Hessian approximation

Mathematics Subject Classification 2010: 74P05, 74P15, 90C30, 90C46, and 90C90

*This research is funded by the Villum Foundation through the research project Topology Optimization - The Next Generation (NextTop).

1 Introduction

Structural topology optimization determines the design in a domain by minimizing an objective function under certain constraints, for a given set of boundary conditions and loads. It is common to minimize compliance or volume subject to limitations on the displacements, volume, or stresses. Topology optimization is a mathematical approach where the design domain is often discretized using finite elements for design parametrization and structural analysis. More details of these problems can be found in the monograph [5].

Several special purpose methods have been implemented to solve structural topology optimization problems. Examples include the Method of Moving Asymptotes, (MMA) [52], its globally convergent version, (GCMMA) [53], and the Convex Linearization (CONLIN) method [21]. These first-order methods solve a sequence of convex sub-problems based on separable approximations of the objective and constraint functions. Different variations of MMA using the diagonal of the second-order derivatives are proposed in [22] and [23], evaluating the benefits of using partial second-order information.

Although it is not very commonly reported in the literature, general nonlinear optimization solvers are also applicable to topology optimization problems. The numerical experiments in [46] show that general purpose solvers, such as interior-point and sequential quadratic programming methods, can be used for solving these type of problems. In these numerical experiments, SNOPT (an SQP method) [27] requires very few iterations to converge to a Karush-Kuhn-Tucker (KKT) point. In addition, second-order information helps to produce accurate results (optimized designs with low objective function value) as demonstrated by the interior-point method IPOPT [55]. The results of the benchmarking study in [46] have motivated the implementation of the second-order SQP method for structural topology optimization problems (TopSQP). Due to the use of second-order information, TopSQP is expected to converge faster than first-order methods. In addition, the objective function values might be reduced too.

In the SQP family of methods, there are numerous variations of algorithms, but all of them are characterized by the same idea. They find approximate solutions to a sequence of normally convex sub-problems. A quadratic objective function models the Lagrangian while the original constraints are linearized. For general information of SQP see e.g. [7] and [30].

One of the main properties of SQP methods is the fast convergence when close to the solution. However, the performance of SQP depends, in general, on the starting point. It is quite difficult to use second-order information (see Section 2) when the problems are nonconvex. In practice, most of the SQP algorithms use quasi-Newton approximations of the Hessian in order to obtain a convex sub-problem, see e.g. [30].

The main differences between SQP algorithms are how the search direction is computed, how the search direction is accepted or rejected, and how inequality constraints are dealt with. Both line search [41] and trust region strategies [12] can be applied in SQP solvers. Additionally, to ensure convergence, merit functions [6] or filter methods [20] are used. Regarding how the method obtains the active inequality constraints, SQP is classified as equality constrained quadratic programming (EQP) or inequality constrained quadratic programming (IQP), see e.g. [33]. Finally, there are different ways to approximate the Hessian of the Lagrangian, using either limited-memory approximations like BFGS (Broyden - Fletcher - Goldfarb - Shanno) [15] or some information of the Hessian. Examples of different SQP algorithms can be found in [30], [29], [33], [40], [35], [13], and [49], among others.

There are several software in the optimization community based on SQP methods. For instance, SNOPT [27], NLPQLP solver [48], and NPSOL [28], where a line search is combined with different penalty functions, the trust region with a filter method in FilterSQP [19], or the new SQP implemented in KNITRO, see e.g. [11] among others. More details of nonlinear solvers can be found in e.g. [38]. Nevertheless, the use of SQP methods in topology optimization is seemingly not very popular, and very few references have been found in this regard, see e.g. [44], [17], and [18].

This article contains a detailed description of an efficient sequential quadratic programming method for maximum stiffness structural topology optimization problems. The SQP+ method introduced in [40] is implemented using second-order information based on the exact Hessian. SQP+ contains two phases, the inequality quadratic phase (IQP), where an inequality constrained convex quadratic sub-problem is solved. Then, an equality constrained quadratic phase (EQP), where the active constraints found for the IQP are used. The step generation is done using a line search strategy in conjunction with a reduction in a merit function. In the special-purpose implementation TopSQP, the IQP phase uses a convex approximation of the exact Hessian instead of the traditional BFGS. Based on the specific structure of the problem formulation, both phases are efficiently reformulated to reduce computational cost. The reformulations avoid one of the most expensive steps in topology optimization problems, which is the computation of the inverse of the stiffness matrix (involved in the Hessian of compliance).

The performance of the proposed TopSQP is compared to the specific purpose GCMMA and with two of the state-of-the-art software in numerical optimization; SNOPT and IPOPT. The comparative study is done using performance profiles [16] on a test set of 225 medium-size minimum compliance problem instances described in [46].

The paper is organized as follows. Section 2 introduces the SQP+ algorithm for a general nonlinear problem and Section 3 briefly defines the topology optimization problem under consideration. Then, the Hessian of the Lagrangian function and some possible

convex approximations are proposed in Section 4. The efficient reformulations of the IQP and EQP phases of TopSQP are gathered in Section 5 and Section 6, respectively. Some implementation details are collected in Section 7. The comparative study of the performance of TopSQP is reported in Section 8, followed by a list of the main limitations this new algorithm may have for topology optimization problems, in Section 9. Finally, Section 10 draws the main conclusions from the results and outlines recommendations for the future work.

2 Sequential Quadratic Programming with an additional equality constrained phase

A general sequential quadratic programming method generates approximate solutions using a quadratic model of the Lagrangian function and a linearization of the constraints.

The general nonlinear constrained problem under consideration is

$$\begin{array}{lll} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 & i = 1,\ldots,m, \\ & l_i \leq x_i \leq u_i & i = 1,\ldots,n, \end{array} \tag{NLP}$$

where $f(\mathbf{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $g_i(\mathbf{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}$ are assumed to be twice continuously differentiable functions.

A conventional SQP method approximates (NLP) at a given iterate $\mathbf{x}_k$, and uses the solution of the sub-problem to produce a search direction $\mathbf{d}_k$. The solver ensures convergence to a KKT point by enforcing, for instance, an improvement in a merit function. This class of methods has shown fast local convergence (see e.g. [43] and [40]), however, the theoretical properties do not hold when the Hessian is indefinite, producing some difficulties to the solver. Since nonconvex problems are NP-hard problems [42], a minimizer of the sub-problem does not guarantee the convergence of the algorithm, see e.g. [45].

The sequential quadratic programming method proposed in this article is based on the algorithm SQP+ in [40]. SQP+ tries to improve the convergence rate by including two phases. First, an inequality quadratic convex constrained sub-problem is solved. Second, the set of active constraints is estimated, and with them, an EQP sub-problem is defined and solved, ignoring the rest of the constraints. The EQP estimated solution refines the search direction, producing fast convergence. Additionally, the IQP iterate of the proposed TopSQP is expected to be more precise than in [40] since a positive definite approximation of the indefinite Hessian is used instead of the BFGS approach. The approximation is based on the exact second-order information (see Section 4).

Throughout this section, the SQP+ method, summarized in Algorithm 1, is outlined for the general nonlinear problem (NLP).

Algorithm 1 SQP+ algorithm from [40].

Input: Define the starting point $\mathbf{x}_0$, the initial Lagrangian multipliers $\boldsymbol{\lambda}_0$, $\boldsymbol{\xi}_0$ $\boldsymbol{\eta}_0$ and the optimality tolerance ω.

Set $\sigma = 10^{-4}$, $\kappa = 0.5$, $k = 1$, $\pi = 1$.
repeat
  Define an approximation of the Hessian of the Lagrangian function, $\mathbf{B}_k \succ 0$ such as $\mathbf{B}_k \approx \nabla^2 \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k)$.
  Solve the IQP sub-problem as explained in Section 2.2 where the solution is $(\mathbf{d}_k^{iq}, \boldsymbol{\lambda}_k^{iq}, \boldsymbol{\xi}_k^{iq}, \boldsymbol{\eta}_k^{iq})$.
  Determine the working set of active constraints, $\mathcal{W}_k$, defined in Section 2.3.
  Solve the EQP sub-problem as explained in Section 2.3 where the solution is $(\mathbf{d}_k^{eq}, \boldsymbol{\lambda}_k^{eq}, \boldsymbol{\xi}_k^{eq}, \boldsymbol{\eta}_k^{eq})$.
  Compute the contraction parameter $\beta \in (0, 1]$ such as the linearized contraints of the IQP sub-problem are feasible at the iterate point $\mathbf{x}_k + \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}$.
  Acceptance/rejection step. Use of line search strategy:
  if $\phi_\pi(\mathbf{x}_k + \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}) \leq \phi_\pi(\mathbf{x}_k) - \sigma q red_\pi(\mathbf{d}_k^{iq})$ **then**
    $\mathbf{d}_k = \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}$.
  else
    Find $\alpha = \{1, \kappa, \kappa^2, ...\}$ such that $\phi_\pi(\mathbf{x}_k + \alpha \mathbf{d}_k^{iq}) \leq \phi_\pi(\mathbf{x}_k) - \sigma \alpha q red_\pi(\mathbf{d}_k^{iq})$.
    $\mathbf{d}_k = \alpha \mathbf{d}_k^{iq}$.
  end if
  Update the primal iterate $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$.
  Update the Lagrangian multiplier estimates $\boldsymbol{\lambda}_{k+1}$, $\boldsymbol{\xi}_{k+1}$, $\boldsymbol{\eta}_{k+1}$ with the strategy explained in Section 2.5.
  Update the penalty parameter π.
  Compute the ∞-norm of KKT conditions of the original problem (NLP).
  Set $k \leftarrow k + 1$.
until `convergence`
return

2.1 Optimality conditions

The Lagrangian function of (NLP) is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\eta}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{n} \xi_i (x_i - u_i) + \sum_{i=1}^{n} \eta_i (l_i - x_i).$$

Here $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)^T$, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^T$, and $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^T$ are the Lagrangian multipliers of the inequality, the upper bound, and the lower bound constraints, respectively.

The first-order necessary conditions for a primal-dual point $(\overline{\mathbf{x}}, \overline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\xi}}, \overline{\boldsymbol{\eta}})$ to be a local optimal solution of the problem (NLP) are gathered in the Karush-Kuhn-Tucker (KKT) conditions (1)-(9), see e.g. [43].

$$\nabla \mathcal{L}(\overline{\mathbf{x}}, \overline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\xi}}, \overline{\boldsymbol{\eta}}) = \nabla f(\overline{\mathbf{x}}) + J(\overline{\mathbf{x}})^T \overline{\boldsymbol{\lambda}} + \overline{\boldsymbol{\xi}} - \overline{\boldsymbol{\eta}} = \mathbf{0}, \tag{1}$$

$$g_i(\overline{\mathbf{x}}) \leq 0 \quad i = 1, \ldots, m, \tag{2}$$

$$l_i \leq \overline{x}_i \leq u_i \quad i = 1, \ldots, n, \tag{3}$$

$$\overline{\lambda}_i \geq 0 \quad i = 1, \ldots, m, \tag{4}$$

$$\overline{\xi}_i \geq 0 \quad i = 1, \ldots, n, \tag{5}$$

$$\overline{\eta}_i \geq 0 \quad i = 1, \ldots, n, \tag{6}$$

$$g_i(\overline{\mathbf{x}})\overline{\lambda}_i = 0 \quad i = 1, \ldots, m, \tag{7}$$

$$(\overline{x}_i - u_i)\overline{\xi}_i = 0 \quad i = 1, \ldots, n, \tag{8}$$

$$(l_i - \overline{x}_i)\overline{\eta}_i = 0 \quad i = 1, \ldots, n. \tag{9}$$

Here, $J(\mathbf{x}) = [\nabla g_i(\mathbf{x})^T]_{i=1,\ldots,m} : \mathbb{R}^n \mapsto \mathbb{R}^{m \times n}$ is the Jacobian of the inequality constraints. Equation (1) refers to the stationarity condition, (2)-(3) are the primal feasibility conditions, and (7)-(9) are the complementarity conditions. In addition, some Constraint Qualification (CQ) must hold at $\overline{\mathbf{x}}$, see [43] and [39].

In practice, SQP+ considers that $\overline{\mathbf{x}}$ is an optimal solution if the stationarity, feasibility, and complementarity conditions are satisfied within some positive tolerance, i.e.

$$\begin{aligned}
&\left\| \nabla f(\overline{\mathbf{x}}) + J(\overline{\mathbf{x}})^T \overline{\boldsymbol{\lambda}} + \overline{\boldsymbol{\xi}} - \overline{\boldsymbol{\eta}} \right\|_\infty \leq \epsilon_1, \\
&\left\| \begin{bmatrix} \mathbf{g}(\overline{\mathbf{x}})^+ \\ \mathbf{g}_u(\overline{\mathbf{x}})^- \\ \mathbf{g}_l(\overline{\mathbf{x}})^- \end{bmatrix} \right\|_\infty \leq \epsilon_2, \\
&\left\| \begin{bmatrix} \mathbf{h}_g(\overline{\mathbf{x}}, \overline{\boldsymbol{\lambda}}) \\ \mathbf{h}_u(\overline{\mathbf{x}}, \overline{\boldsymbol{\xi}}) \\ \mathbf{h}_l(\overline{\mathbf{x}}, \overline{\boldsymbol{\eta}}) \end{bmatrix} \right\|_\infty \leq \epsilon_3,
\end{aligned}$$

for some given constants $\epsilon_1 > 0$, $\epsilon_2 > 0$ and $\epsilon_3 > 0$. Here,

$$\begin{aligned}
&\mathbf{g}(\mathbf{x})^+ \triangleq [\max\{0, g_i(\mathbf{x})\}]_{i=1,\ldots,m}, \\
&\mathbf{g}_u(\mathbf{x})^- \triangleq [\max\{0, -(u_i - x_i)\}]_{i=1,\ldots,n}, \\
&\mathbf{g}_l(\mathbf{x})^- \triangleq [\max\{0, -(x_i - l_i)\}]_{i=1,\ldots,n}, \\
&\mathbf{h}_g(\mathbf{x}, \boldsymbol{\lambda}) \triangleq \left[g_i(\mathbf{x})\lambda_i \right]_{i=1,\ldots,m}, \\
&\mathbf{h}_u(\mathbf{x}, \boldsymbol{\xi}) \triangleq \left[(x_i - u_i)\xi_i \right]_{i=1,\ldots,n}, \\
&\mathbf{h}_l(\mathbf{x}, \boldsymbol{\eta}) \triangleq \left[(l_i - x_i)\eta_i \right]_{i=1,\ldots,n}.
\end{aligned}$$

2.2 Solving the IQP sub-problem

The inequality constrained quadratic phase (IQP) approximates the problem (NLP) with a convex quadratic model of the Lagrangian function and a linearization of the

constraints. Thus, a positive definite matrix, $\mathbf{B}_k$, computed in Step 3 of Algorithm 1 is crucial to define the IQP problem.

$$\begin{aligned} \underset{\mathbf{d}\in\mathbb{R}^n}{\text{minimize}} \quad & \nabla f(\mathbf{x}_k)^T\mathbf{d} + \tfrac{1}{2}\mathbf{d}^T\mathbf{B}_k\mathbf{d} \\ \text{subject to} \quad & g(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{d} \leq \mathbf{0}, \\ & \tilde{l}_i \leq d_i \leq \tilde{u}_i \qquad\qquad i = 1,\ldots,n. \end{aligned} \tag{IQP}$$

The lower and the upper bounds of d_i are defined with $\tilde{l}_i = l_i - (\mathbf{x}_k)_i$ and $\tilde{u}_i = u_i - (\mathbf{x}_k)_i$, $i = 1,\ldots,n$, respectively.

From now on, the linearization of the constraints in (IQP) is assumed to result in a feasible problem (see Section 3). The primal-dual solution of the sub-problem ($\mathbf{d}_k^{iq}$, $\boldsymbol{\lambda}_k^{iq}$, $\boldsymbol{\xi}_k^{iq}$, $\boldsymbol{\eta}_k^{iq}$), is used to estimate first, the set of active constraints and second, the search direction.

The most important characteristic of this IQP phase is the convexity of the optimization problem. Any local optimal solution of a convex problem is a global solution. Furthermore, for the IQP in this work, existence of solution is ensured (see Section 4). In addition, the problem always has a descent direction until convergence (see e.g. [9]).

An important aspect for large-scale problems is to solve this sub-problem as fast as possible. In this context, it is solved using fast commercial solvers for large-scale quadratic problems, such as Gurobi [34] or CPLEX [37].

2.3 Solving EQP sub-problem

The working set $\mathcal{W}_k$ (Step 5 of Algorithm 1) contains all the indices of the constraints where their linearization in the sub-problem (IQP) are active at $\mathbf{d}_k^{iq}$, i.e,

$$\mathcal{W}_k = \{\mathcal{W}_k^g \cup \mathcal{W}_k^u \cup \mathcal{W}_k^l\},$$

with

$$\begin{aligned} \mathcal{W}_k^g &= \{\ i\ \mid\ g_i(\mathbf{x}_k) + \nabla g_i(\mathbf{x}_k)^T\mathbf{d}_k^{iq} = 0,\ \ i = 1,\ldots,m\}, \\ \mathcal{W}_k^u &= \{\ i\ \mid\ (\mathbf{d}_k^{iq})_i - \tilde{u}_i = 0,\ \ i = 1,\ldots,n\}, \\ \mathcal{W}_k^l &= \{\ i\ \mid\ (\mathbf{d}_k^{iq})_i - \tilde{l}_i = 0,\ \ i = 1,\ldots,n\}. \end{aligned}$$

For those active constraints, the following equality constrained sub-problem (EQP) is defined. The explanation of its final formulation can be found in [10].

$$\begin{aligned} \underset{\mathbf{d}\in\mathbb{R}^n}{\text{minimize}} \quad & (\nabla f(\mathbf{x}_k) + \mathbf{H}_k\mathbf{d}_k^{iq})^T\mathbf{d} \quad + \tfrac{1}{2}\mathbf{d}^T\mathbf{H}_k\mathbf{d} \\ \text{subject to} \quad & J_{\mathcal{W}_k^g}(\mathbf{x}_k)\mathbf{d} = \mathbf{0}, \\ & d_i = 0 \qquad\qquad i \in \{\mathcal{W}_k^u \cup \mathcal{W}_k^l\}. \end{aligned} \tag{EQP}$$

The matrix $\mathbf{H}_k$ refers to the exact Hessian of the Lagrangian function i.e. $\mathbf{H}_k = \nabla^2\mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k^{iq})$ and $J_{\mathcal{W}_k^g}(\mathbf{x}_k)$ represents the matrix whose rows are the active constraint

gradients (i.e. the $\mathcal{W}_k^g$ rows of the Jacobian). Applying Newton's method to the first-order optimality conditions of (EQP), this optimization problem is equivalent to solve the KKT system

$$\begin{bmatrix} \mathbf{H}_k & J_{\mathcal{W}_k^g}(\mathbf{x}_k)^T & \mathbf{I}_{\mathcal{W}_k^u} & -\mathbf{I}_{\mathcal{W}_k^l} \\ J_{\mathcal{W}_k^g}(\mathbf{x}_k) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{\mathcal{W}_k^u}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_{\mathcal{W}_k^l}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d}_k^{eq} \\ \boldsymbol{\lambda}_k^{eq} \\ \boldsymbol{\xi}_k^{eq} \\ \boldsymbol{\eta}_k^{eq} \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x}_k) + \mathbf{H}_k \mathbf{d}_k^{iq} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{10}$$

where $\mathbf{I}_{\mathcal{W}_k^u} \in \mathbb{R}^{n \times |\mathcal{W}_k^u|}$ and $\mathbf{I}_{\mathcal{W}_k^l} \in \mathbb{R}^{n \times |\mathcal{W}_k^l|}$ are pseudo-identity matrices. The Lagrangian multipliers $(\boldsymbol{\lambda}_k^{eq}, \boldsymbol{\xi}_k^{eq}, \boldsymbol{\eta}_k^{eq})$ refer to the active constraints. The size of the system (10) can be easily reduced since

$$\begin{aligned} \mathbf{I}_{\mathcal{W}_k^u}^T \mathbf{d}_k^{eq} &= \mathbf{0}, \\ \mathbf{I}_{\mathcal{W}_k^l}^T \mathbf{d}_k^{eq} &= \mathbf{0}. \end{aligned}$$

Therefore, only a system taking into account the nonzero direction ($(\mathbf{d}_k^{eq})_i \neq 0$) need to be solved.

$$\begin{bmatrix} \mathbf{H}_{k,\mathcal{W}_{k,c}^b} & J_{\mathcal{W}_k^g,\mathcal{W}_{k,c}^b}(\mathbf{x}_k)^T \\ J_{\mathcal{W}_k^g,\mathcal{W}_{k,c}^b}(\mathbf{x}_k) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{d}}_k^{eq} \\ \boldsymbol{\lambda}_k^{eq} \end{bmatrix} = - \begin{bmatrix} (\nabla f(\mathbf{x}_k) + \mathbf{H}_k \mathbf{d}_k^{iq})_{\mathcal{W}_{k,c}^b} \\ \mathbf{0} \end{bmatrix}. \tag{11}$$

Here, $\mathcal{W}_{k,c}^b = \{1, \ldots, n\} \setminus \{\mathcal{W}_k^u \cup \mathcal{W}_k^l\}$ is the complementarity set of the active bounds (upper and lower), $\mathbf{H}_{k,\mathcal{W}_{k,c}^b}$ refers to the $\mathcal{W}_{k,c}^b$ columns and rows of $\mathbf{H}_k$, and $J_{\mathcal{W}_k^g,\mathcal{W}_{k,c}^b}(\mathbf{x}_k)$ refers to the $\mathcal{W}_k^g$ rows and $\mathcal{W}_{k,c}^b$ columns of the Jacobian. In this case, the term $\tilde{\mathbf{d}}_k^{eq}$ represents the $\mathcal{W}_{k,c}^b$ rows of $\mathbf{d}_k^{eq}$. The Lagrangian multipliers of the active bounds are obtained afterwards[1] by

$$\begin{aligned} \mathbf{z}_k^{eq} &= -\nabla f(\mathbf{x}_k) - \mathbf{H}_k \mathbf{d}_k^{iq} - \mathbf{H}_k \mathbf{d}_k^{eq} - J_{\mathcal{W}_k^g}(\mathbf{x}_k)^T \boldsymbol{\lambda}_k^{eq}, \\ \boldsymbol{\xi}_k^{eq} &= \mathbf{z}_{k,\mathcal{W}_k^u}^{eq}, \\ \boldsymbol{\eta}_k^{eq} &= -\mathbf{z}_{k,\mathcal{W}_k^l}^{eq}. \end{aligned}$$

The computation of the EQP search direction is, generally, much faster than the IQP solution. The EQP sub-problem not only helps to produce a more accurate search direction, but also reduces the number of IQP phases (the number of iterations is decreased) (see Section 7). A direct consequence is a reduction of the total computational time.

2.3.1 Existence of solutions of the equality quadratic constraint problem

The equality constrained quadratic problem (EQP) is equivalent to solving the system of equations (11). The matrix of this system is commonly defined like

$$\mathbf{K} = \begin{bmatrix} \mathbf{H} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix},$$

[1] Note that $\mathcal{W}_k^u$ and $\mathcal{W}_k^l$ are disjoint sets by definition.

where $\mathbf{H} \in \mathbb{R}^{n\times n}$ is symmetric and $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the Jacobian of the linearized active constraints. It is assumed that $\mathbf{A}$ has full row rank. The matrix $\mathbf{K}$ is called a Karush-Kuhn-Tucker matrix. There are conditions under which the system has solution.

Theorem 1. *(from [43]) Let $\boldsymbol{A}$ have a full row rank, and assume that the reduced-Hessian matrix $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}$ is positive definite. Then the KKT matrix $\boldsymbol{K}$ is nonsingular, and hence there is a unique vector that satisfied the linear system and is the unique global solution.*

Where $\boldsymbol{Z} \in \mathbb{R}^{n\times(n-m)}$ is a matrix whose columns are the basis of the null-space of $\boldsymbol{A}$.

Thus, in order to obtain a solution of the EQP sub-problem, the reduced-Hessian must be positive definite, and the matrix $\mathbf{A}$ must have full rank.

Definition 1. *(from [32]) Suppose a matrix $\boldsymbol{K}$. Let i_0, i_+ and i_- be the number of zero, positive and negative eigenvalues of $\boldsymbol{K}$, respectively. The trial (i_+, i_-, i_0) is called **inertia**.*

Theorem 2. ***Gould Theorem ([36]).*** *Suppose that $\boldsymbol{A}$ has full rank m. The condition $\boldsymbol{p}^T\boldsymbol{H}\boldsymbol{p} > 0$, $\forall \boldsymbol{p} \neq \boldsymbol{0}$, such that $\boldsymbol{A}^T\boldsymbol{p} = \boldsymbol{0}$ holds if and only if*

$$inertia(\boldsymbol{K}) = (n, m, 0).$$

In addition,

$$inertia(\boldsymbol{K}) = inertia(\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}) + (m, m, 0).$$

Therefore, if $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}$ is positive definite, $inertia(\boldsymbol{K}) = (n, m, 0)$.

An option to compute the inertia of a KKT matrix is by the LDL factorization, see e.g. [26] and [43]. If the inertia is not correct, it is necessary to modify the KKT matrix to ensure the existence of solution. There are many different alternatives to modify the nonconvex sub-problem into a local convex approximation, such as using different inertia correction based on LDL factorizations, see e.g. [26], [25], and [24], or using other techniques such as in [36] and [55]. In addition, it is possible to apply different convexification approaches proposed in [29] and [31].

When the computation of the reduced-Hessian is computationally cheap, the system can be easily modified to guarantee a positive definite reduced-Hessian, see [43]. For instance, $\mathbf{H}_z = \mathbf{Z}_k^T\mathbf{H}_k\mathbf{Z}_k$ is perturbed using its eigenvalues, so that $\hat{\mathbf{H}}_z = \mathbf{H}_z + \gamma\mathbf{I} \succ 0$, with $\gamma = \min(|\lambda_{eig}(\mathbf{H}_z)|) + \epsilon$, where λ_{eig} refers the the eigenvalues of the matrix. This perturbation is then used for the KKT matrix,

$$\begin{bmatrix} \mathbf{H} + \gamma\mathbf{I} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}.$$

The EQP sub-problem in [40] is solved only in those cases where the inertia of the system is correct. However, TopSQP modifies the KKT matrix to guarantee a correct

inertia. If the same approach as in [40] is used, TopSQP would essentially be a classical SQP with only an IQP step since the indefiniteness of the Hessian of the compliance (see Section 3) usually produces an incorrect inertia.

2.4 Acceptance/rejection of the step

Once the IQP and EQP search directions are computed, it is necessary to verify whether these estimates improve the iterate $\mathbf{x}_k$. First of all, a contraction parameter β (Step 7) secures that the linearization of all the constraints (active and inactive) are satisfied at the point with maximum search direction $\mathbf{d} = \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}$. In other words, the constraints of the IQP phase must remain feasible.

The largest value of $\beta \in (0,1]$ is computed such that

$$\begin{array}{ll} g_i(\mathbf{x}_k) + \nabla g_i(\mathbf{x}_k)^T \mathbf{d} \leq 0 & i \in \mathcal{W}_{k,c}^g, \\ d_i \leq \tilde{u}_i & i \in \mathcal{W}_{k,c}^u, \\ \tilde{l}_i \leq d_i & i \in \mathcal{W}_{k,c}^l. \end{array}$$

Here,

$$\begin{array}{l} \mathcal{W}_{k,c}^g = \{1,\ldots,m\} \setminus \mathcal{W}_k^g, \\ \mathcal{W}_{k,c}^u = \{1,\ldots,n\} \setminus \mathcal{W}_k^u, \\ \mathcal{W}_{k,c}^l = \{1,\ldots,n\} \setminus \mathcal{W}_k^l, \end{array}$$

are the complementarity working set. Once the contraction parameter is determined, a line search estimates the step length α for the new step direction (Step 9 and 12 in Algorithm 1). The TopSQP line search is implemented as [40], Section 5. The line search procedure is slightly modified compared to the theoretical procedure used for the theoretical results. The acceptance criterion is based on the merit function (12) and the model reduction from $\mathbf{x}_k$ to $\mathbf{x}_k + \mathbf{d}$ (13) of the original problem (NLP). These functions are defined as

$$\phi_\pi(\mathbf{x}) = f(\mathbf{x}) + \pi(||\mathbf{g}(\mathbf{x})^+||_1 + ||\mathbf{g}_l(\mathbf{x})^-||_1 + ||\mathbf{g}_u(\mathbf{x})^-||_1) \tag{12}$$

$$\begin{aligned} qred_\pi(\mathbf{d}) = \quad & -(\nabla f(\mathbf{x}_k)^T \mathbf{d} + \tfrac{1}{2}\mathbf{d}^T \mathbf{B}_k \mathbf{d}) + \pi(||\mathbf{g}(\mathbf{x}_k)^+||_1 + \\ & ||\mathbf{g}_l(\mathbf{x}_k)^-||_1 + ||\mathbf{g}_u(\mathbf{x}_k)^-||_1). \end{aligned} \tag{13}$$

If the sufficient decrease condition (14) is satisfied, then the iterate $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}$ is accepted.

$$\phi_\pi(\mathbf{x}_k + \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}) \leq \phi_\pi(\mathbf{x}_k) - \sigma qred_\pi(\mathbf{d}_k^{iq}). \tag{14}$$

Otherwise, $\alpha \in \{1, \kappa, \kappa^2, \ldots\}$ is found such that condition (15) is satisfied. In this case, the new iterate is defined as $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k^{iq}$.

$$\phi_\pi(\mathbf{x}_k + \alpha \mathbf{d}_k^{iq}) \leq \phi_\pi(\mathbf{x}_k) - \sigma \alpha qred_\pi(\mathbf{d}_k^{iq}). \tag{15}$$

There are several techniques to update the penalty parameter π in order to improve the convergence rate, for instance, the strategy explained in e.g. [56] and [14]. In practice, due to the feasibility of the sub-problems (see Section 3), the term affected by π is always close to 0. The penalty parameter π is updated very simple by just using the Lagrangian multipliers, i.e. $\pi = ||\boldsymbol{\lambda}||_\infty$. Finally, the parameters $\sigma = 10^{-4}$ and $\kappa = 0.5$ are taken from [40].

2.5 Updating the Lagrangian multipliers

The SQP+ algorithm in [40] updates the estimates of the Lagrangian multipliers depending on the final step direction. The updating scheme of the SQP+ is

$$\boldsymbol{\lambda}_{k+1} = \begin{cases} \max(\mathbf{0}, \boldsymbol{\lambda}_k^{eq}) & \text{if } \mathbf{d}_k = \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq} \\ (1-\alpha)\boldsymbol{\lambda}_k + \alpha \boldsymbol{\lambda}_k^{iq} & \text{if } \mathbf{d}_k = \alpha \mathbf{d}_k^{iq}. \end{cases}$$

Although [40] suggest that both the equality and the inequality Lagrangian multipliers can be good candidates, the former is considered since a BFGS is used in the IQP phase. Nevertheless, the proposed IQP, detailed in Section 5, gives also accurate Lagrangian estimates due to the use of part of the exact Hessian (see Section 4). In addition, the working set of active constraints is numerically approximated (see Section 7), and the EQP might be defined with constraints that are inactive. A preliminary study was performed to investigate how the election of the Lagrangian multipliers was affecting the convergence. The performance was slightly better when the inequality Lagrangian multipliers were used, i.e.,

$$\begin{aligned} \boldsymbol{\lambda}_{k+1} &= \alpha \boldsymbol{\lambda}_k^{iq} + (1-\alpha)\boldsymbol{\lambda}_k, \\ \boldsymbol{\xi}_{k+1} &= \alpha \boldsymbol{\xi}_k^{iq} + (1-\alpha)\boldsymbol{\xi}_k, \\ \boldsymbol{\eta}_{k+1} &= \alpha \boldsymbol{\eta}_k^{iq} + (1-\alpha)\boldsymbol{\eta}_k. \end{aligned}$$

2.6 Convergence properties

In [40] the global and local convergence properties of SQP+ are explained in detail. The IQP phase gives the global convergence while the EQP phase helps to produce fast local convergence, when the active set is correct. For the theoretical proof, some assumptions are established. In addition, the quadratic convergence proof relies on the second-order sufficient conditions,[2] at $\mathbf{x}_k$ sufficiently close to the optimal point. A full step ($\mathbf{d}_k = \mathbf{d}_k^{iq} + \mathbf{d}_k^{eq}$) is also assumed at those points [40]. Even, if it is not the goal of this article to demonstrate the convergence of the solver, it is important to point out that some of the assumptions made in [40] are easily proven for our specific problem,

[2] (from [43]) For a given feasible point $\overline{\mathbf{x}}$, there are some Lagrangian multipliers $(\overline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\xi}}, \overline{\boldsymbol{\eta}})$ such that the KKT conditions are satisfied. Suppose that $\mathbf{p}^T \mathbf{H} \mathbf{p} > 0$, such that $\mathbf{A}^T \mathbf{p} = \mathbf{0}$, with $\mathbf{A}$ the Jacobian of the active constraints. Then, $\overline{\mathbf{x}}$ is a strict local solution.

such as the convexity of the feasible set, the feasibility of the IQP sub-problem and the Lipschitz continuous property of the objective and constraint functions. On the other hand, there are several assumptions that cannot be proven for general topology optimization problems, such as the strict complementarity assumption. In general, these assumptions are quite strict and they exist to prove general global and local convergence.

However, TopSQP lost the theoretical quadratic convergence property. Due to numerical tolerances, the optimization method, at iterates close to the optimal solution, does not take the full step, i.e. $\beta < 1$. The strict complementarity is not satisfied, and at some of these iterations, the second-order sufficient conditions are not satisfied. Thus, Theorem 4.3 in [40] cannot be applied. Nevertheless, the numerical experiments in Section 8 will show that the proposed implementation has a great robustness.

3 Problem formulation

The minimum compliance problem is considered as one of the most typical structural topology optimization problems. The classical formulation consists of maximizing the stiffness of the structure (minimizing compliance) subject to a volume constraint, see more details in e.g. [5]. This article considers the nested approach, where the displacements (state variables, $\mathbf{u}$) depend on the design variables ($\mathbf{t}$), related with the linear elastic equilibrium equations in their discretized form

$$\begin{aligned} &\mathbf{K}(\mathbf{t})\mathbf{u} = \mathbf{f}, \\ &\mathbf{u}(\mathbf{t}) = \mathbf{K}^{-1}(\mathbf{t})\mathbf{f}. \end{aligned}$$

Here $\mathbf{t} \in \mathbb{R}^n$ is the density variable, n is the number of elements, $\mathbf{K}(\mathbf{t}) : \mathbb{R}^n \to \mathbb{R}^{d\times d}$ is the stiffness matrix, with d the number of degrees of freedom, and $\mathbf{f} \in \mathbb{R}^d$ the static design-independent force vector.

In particular, the density-based approach is used to penalize intermediate densities to produce an almost solid-and-void design. More specifically, the Solid Isotropic of Material Penalization (SIMP) approach is chosen (see e.g. [4], [47], and [57]). For this interpolation, the stiffness matrix is defined as

$$\begin{aligned} &\mathbf{K}(\mathbf{t}) = \sum_{e=1}^{n} E(t_e)\mathbf{K}_e, \\ &\text{with} \\ &E(t_e) = E_v + (E_1 - E_v)\tilde{t}_e^p, \\ &\text{where the SIMP penalty parameter is } p \geq 1. \end{aligned}$$

Here, $E_v > 0$ and $E_1 \gg E_v$ are the "void" and solid Young's modulus, respectively, and $\mathbf{K}_e$ the element stiffness matrix. The stiffness matrix is assumed to be positive definite for all design vectors satisfying the bound constraints to avoid singularity. Finally, the variable

$\tilde{t}_e$ refers to the design variable with a density filter [8], and [51], defined analogous to [46].

The minimum compliance problem in its discrete version is

$$\begin{array}{ll} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{minimize}} & \mathbf{u}(\mathbf{t})^T\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \\ \text{subject to} & \mathbf{a}^T\mathbf{t} \leq V, \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{array} \tag{P^c}$$

Here $\mathbf{a} = (a_1, \ldots, a_n)^T \in \mathbb{R}^n$ with $a_i > 0 \quad i = 1, \ldots, n$ the relative element volume, and $V > 0$ the total volume fraction. For simplicity, $a_i = a_j \quad \forall i, j$.

The nonlinear optimization problem (P^c) contains only one linear inequality constraint and bound constraints. Denote the feasible set of (P^c) by

$$\Omega = \{t_i \in [0,1] \quad i = 1, \ldots, n, \sum_{i=1}^{n} a_i t_i \leq V\}. \tag{16}$$

The set Ω (16) is convex, nonempty under natural assumptions, closed, bounded and thus compact, [9].

Both Ω and the constraint functions are convex. However, the optimization problem is, in general, nonconvex [9], since the Hessian of the Lagrangian function $\nabla^2\mathcal{L}(\mathbf{t}, \boldsymbol{\lambda}) = \nabla^2 f(\mathbf{x})$ is not positive semi-definite (cf. below). The feasible set is nonempty, i.e. there is, at least, one local solution for (NLP). Certain CQs hold at every point due to the linearity of the constraints. The authors emphasize the importance of the CQ because, in general, the numerical optimization theory assumes that they are satisfied (see e.g. [43] and [39]).

4 Approximation of the Hessian of the Lagrangian

The Hessian is defined by using sensitivity analysis on the objective function,

$$\nabla^2\mathcal{L}(\mathbf{t}, \boldsymbol{\lambda}) = 2\mathbf{F}(\mathbf{t})^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) - \mathbf{Q}(\mathbf{t})$$

with

$$\begin{aligned} \mathbf{Q}(\mathbf{t}) &= \operatorname{diag}(\mathbf{u}^T(\mathbf{t})\frac{\partial^2\mathbf{K}_i(t_i)}{\partial t_i^2}\mathbf{u}(\mathbf{t})) : \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n} \\ \mathbf{F}(\mathbf{t}) &= \left(\frac{\partial\mathbf{K}_1(t_1)}{\partial t_1}\mathbf{u}(\mathbf{t}) \quad \cdots \quad \frac{\partial\mathbf{K}_n(t_n)}{\partial t_n}\mathbf{u}(\mathbf{t})\right) : \mathbb{R}^n \longrightarrow \mathbb{R}^{d\times n}. \end{aligned}$$

For $p = 1$ (SIMP penalization parameter), the term $\mathbf{Q}(\mathbf{t})$ is zero, and the problem is convex with

$$\nabla^2\mathcal{L}(\mathbf{t}, \boldsymbol{\lambda}) = 2\mathbf{F}(\mathbf{t})^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) \succeq 0.$$

The problem generally becomes nonconvex for $p > 1$.

The IQP phase requires a positive definite approximation of the Hessian, $\mathbf{B}_k$ (step 3 Algorithm 1). Instead of using a BFGS approximation as in [40], the exact second-order

information is used as much as possible. A convex (positive semi-definite) approximation could be for instance,

$$\hat{\mathbf{H}}_1 = 2\mathbf{F}(\mathbf{t})^T \mathbf{K}^{-1}(\mathbf{t})\ \mathbf{F}(\mathbf{t}). \tag{17}$$

From a theoretical point of view, the approximate matrix $\mathbf{B}_k$ must be positive definite, i.e. $\mathbf{B}_k = \hat{\mathbf{H}}_1 + \epsilon\mathbf{I} \succ 0$. In practice, $\mathbf{B}_k = \hat{\mathbf{H}}_1 \succeq 0$ is used, and thus, in some iterations, some theoretical properties might be lost. In the numerical experiments, the approximate Hessian $\hat{\mathbf{H}}_1$ is strictly positive definite, though.

There are other alternatives to modify the Hessian so that it becomes positive definite, such as equation (18) where an extra term is added, or equation (19) where the nonconvex part is contracted by a factor γ_2.

$$\hat{\mathbf{H}}_2 = 2\mathbf{F}(\mathbf{t})^T \mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) - \mathbf{Q}(\mathbf{t}) + \gamma_1\mathbf{I}, \tag{18}$$

$$\hat{\mathbf{H}}_3 = 2\mathbf{F}(\mathbf{t})^T \mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) - \gamma_2\mathbf{Q}(\mathbf{t}). \tag{19}$$

There are some other alternatives to $\mathbf{B}_k$, for instance, the identity matrix, or the positive diagonal terms of the exact Hessian, as it is suggested in [22]. Nevertheless, we would like to take advantage of the structure of the exact Hessian. The positive semi-definite $\hat{\mathbf{H}}_1$ (17) matrix is chosen to be the $\mathbf{B}_k$ of IQP step, since it is the fastest approximation (there is no need to estimate γ_1 or γ_2).

In the EQP step, either the exact Hessian with an inertia correction method is used or the KKT matrix is perturbed so that the reduced-Hessian becomes positive definite.

The minimum compliance problem (P^c) has only one linear inequality constraint, therefore, a basis ($\mathbf{Z}$) of the null-space of the gradient of the active constraints is very easy to compute. A perturbation of the KKT matrix is easily obtained by enforcing a positive definite reduced-Hessian introduced in Section 2.3.1. However, the reduced-Hessian, $\mathbf{Z}^T\mathbf{H}\mathbf{Z}$, becomes dense and the computation of γ will be expensive.

On the other hand, it is also possible to directly use the same positive definite approximate Hessian as in the IQP. This positive definite approximation guarantees the correctness of the inertia of the system without examines it. However, once the working set of active constraints is identified, both the EQP and the IQP will theoretically be identical [40]. In practice, the set of active constraints of the considered problem is identified at the very end of the optimization since the density variables are constantly moving. Moreover, the numerical errors in both, the tolerance of the IQP and EQP steps and the selection of the active set (see Section 7), produce that the IQP and EQP steps at points sufficiently close to the optimal are different.

5 Alternatives to the primal IQP formulation

The IQP sub-problem of (P^c) is formulated using an approximate Hessian $\mathbf{B}_k$ defined in (17) as

$$\begin{array}{ll} \underset{\mathbf{d}\in\mathbb{R}^n}{\text{minimize}} & \hat{f}(\mathbf{d}) = -(\mathbf{F}_k^T\mathbf{u}_k)^T\mathbf{d} + \mathbf{d}^T\mathbf{F}_k^T\mathbf{K}_k^{-1}\mathbf{F}_k\mathbf{d} \\ \text{subject to} & \mathbf{a}^T\mathbf{t}_k - V + \mathbf{a}^T\mathbf{d} \leq 0 \\ & -\mathbf{t}_k \leq \mathbf{d} \leq \mathbf{1} - \mathbf{t}_k. \end{array} \qquad (IQP_p)$$

For simplicity, any function or matrix with the form $A(\mathbf{t}_k)$ is represented by A_k.

The feasible set of (IQP_p) is convex (hyperplanes), nonempty, closed, and bounded. By construction the (IQP_p) is convex. Moreover some CQs hold. Thus, this sub-problem has an optimal solution and any local solution that satisfies the KKT condition is also a global minimizer.

5.1 Reformulation of the primal IQP formulation

TopSQP spends most of the computational time in the IQP phase. In addition, (IQP_p) is extremely expensive since the inverse of the stiffness matrix is involved, and the matrix $\mathbf{B}_k$ is dense.

However, it is possible to reformulate the inequality sub-problem such that the computation and the storage of the approximate Hessian are no longer required. First of all, a Cholesky factorization is used for the stiffness matrix. Then, a new variable $\tilde{\mathbf{z}} = (\mathbf{R}_k^T)^{-1}\mathbf{F}_k\mathbf{d} \in \mathbb{R}^d$ is included to rename some terms of the problem so that any inverse matrix is removed from the objective function of (IQP_p).

$$\begin{aligned} \hat{f}(\mathbf{d}) &= -(\mathbf{F}_k^T\mathbf{u}_k)^T\mathbf{d} + \mathbf{d}^T\mathbf{F}_k^T(\mathbf{R}_k^T\mathbf{R}_k)^{-1}\mathbf{F}_k\mathbf{d}, \\ \hat{f}(\mathbf{d},\tilde{\mathbf{z}}) &= -(\mathbf{F}_k^T\mathbf{u}_k)^T\mathbf{d} + \tilde{\mathbf{z}}^T\tilde{\mathbf{z}}. \end{aligned}$$

The introduction of the new variable $\tilde{\mathbf{z}}$ leads to an enlargement of the number of constraints. The alternative IQP formulation is

$$\begin{array}{ll} \underset{\mathbf{d}\in\mathbb{R}^n,\tilde{\mathbf{z}}\in\mathbb{R}^d}{\text{minimize}} & -(\mathbf{F}_k^T\mathbf{u}_k)^T\mathbf{d} + \tilde{\mathbf{z}}^T\tilde{\mathbf{z}} \\ \text{subject to} & \mathbf{R}_k^T\tilde{\mathbf{z}} - \mathbf{F}_k\mathbf{d} = \mathbf{0}, \\ & \mathbf{A}_k\mathbf{d} \leq \mathbf{b}_k, \end{array} \qquad (IQP_{p-2})$$

where the linear inequality constraints are condensed into a system $\mathbf{A}_k\mathbf{d} \leq \mathbf{b}_k$, to simplify the notation. Here, $m = 2n+1$, and

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{a}^T \\ \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \in \mathbb{R}^{m\times n},$$

$$\mathbf{b}_k = \begin{bmatrix} -(\mathbf{a}^T\mathbf{t}_k - V) \\ \mathbf{1} - \mathbf{t}_k \\ \mathbf{t}_k \end{bmatrix} \in \mathbb{R}^m.$$

Using this new formulation, the number of variables and linear constraints are increased. In contrast, the computational time can, due to sparsity, significantly be reduced.

5.2 Dual problem of the IQP formulation

Using Lagrangian duality theory (see e.g. [9]), an optimization problem can be reformulated using the *dual variables* $(\boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\eta})$. In some cases this new *dual problem* is much easier to solve and computationally less expensive than the primal problem. Thus, the problem (IQP_p) can also be formulated in its dual problem.

A new variable $\mathbf{z} \in \mathbb{R}^d$ is introduced to rename some terms of the objective function. Let

$$\mathbf{z} = \mathbf{F}_k \mathbf{d}, \tag{20}$$

then, the (IQP_p) problem is equivalent to

$$\begin{array}{ll} \underset{\mathbf{d}\in\mathbb{R}^n, \mathbf{z}\in\mathbb{R}^d}{\text{minimize}} & -(\mathbf{F}_k^T \mathbf{u}_k)^T \mathbf{d} + \mathbf{z}^T \mathbf{K}_k^{-1} \mathbf{z} \\ \text{subject to} & \mathbf{A}_k \mathbf{d} \leq \mathbf{b}_k, \\ & \mathbf{z} = \mathbf{F}_k \mathbf{d}. \end{array} \tag{21}$$

The Lagrangian function of the IQP sub-problem (21) is described in terms of the primal variables $\mathbf{d}$ and $\mathbf{z}$, and the dual variables $\boldsymbol{\nu} = (\boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\eta}) \in \mathbb{R}^m$ (for the inequality and bound constraints) and $\boldsymbol{\theta} \in \mathbb{R}^d$ (for the new equality constraints).

$$\begin{aligned} \mathcal{L}_p(\mathbf{d}, \mathbf{z}, \boldsymbol{\nu}, \boldsymbol{\theta}) = \; & -(\mathbf{F}_k^T \mathbf{u}_k)^T \mathbf{d} + \mathbf{z}^T \mathbf{K}_k^{-1} \mathbf{z} + \\ & \boldsymbol{\nu}^T (\mathbf{A}_k \mathbf{d} - \mathbf{b}_k) + \boldsymbol{\theta}^T (\mathbf{z} - \mathbf{F}_k \mathbf{d}). \end{aligned}$$

The dual problem consists of maximizing

$$\varphi(\boldsymbol{\nu}, \boldsymbol{\theta}) = \inf_{\mathbf{d}, \mathbf{z}} \mathcal{L}_p(\mathbf{d}, \mathbf{z}, \boldsymbol{\nu}, \boldsymbol{\theta})$$

respect to the dual variables $\boldsymbol{\nu}, \boldsymbol{\theta}$ [9]. The formulation of the dual problem is obtained by satisfying the optimality conditions of (21),

$$\begin{aligned} \nabla_{\mathbf{d}} \mathcal{L}_p(\overline{\mathbf{d}}, \overline{\mathbf{z}}, \overline{\boldsymbol{\nu}}, \overline{\boldsymbol{\theta}}) &= -\mathbf{F}_k^T \mathbf{u}_k + \mathbf{A}_k^T \overline{\boldsymbol{\nu}} - \mathbf{F}_k^T \overline{\boldsymbol{\theta}} = \mathbf{0}, \\ \nabla_{\mathbf{z}} \mathcal{L}_p(\overline{\mathbf{d}}, \overline{\mathbf{z}}, \overline{\boldsymbol{\nu}}, \overline{\boldsymbol{\theta}}) &= 2\mathbf{K}_k^{-1} \overline{\mathbf{z}} + \overline{\boldsymbol{\theta}} = \mathbf{0}. \end{aligned}$$

Then, the solution of the dual problem must satisfy:

$$\overline{\mathbf{z}} = -\frac{1}{2} \mathbf{K}_k \overline{\boldsymbol{\theta}}, \tag{22}$$

$$-\mathbf{F}_k^T \mathbf{u}_k + \mathbf{A}_k^T \overline{\boldsymbol{\nu}} - \mathbf{F}_k^T \overline{\boldsymbol{\theta}} = \mathbf{0}. \tag{23}$$

Based on the above equations, the primal Lagrangian function at $(\overline{\mathbf{d}},\overline{\mathbf{z}},\overline{\boldsymbol{\nu}},\overline{\boldsymbol{\theta}})$ is

$$\begin{aligned}\mathcal{L}_p(\overline{\mathbf{d}},\overline{\mathbf{z}},\overline{\boldsymbol{\nu}},\overline{\boldsymbol{\theta}}) &= -(\mathbf{F}_k^T\mathbf{u}_k)^T\overline{\mathbf{d}}+\overline{\mathbf{z}}^T\mathbf{K}_k^{-1}\overline{\mathbf{z}}+\overline{\boldsymbol{\nu}}^T(\mathbf{A}_k\overline{\mathbf{d}}-\mathbf{b}_k)+\\ &\quad \overline{\boldsymbol{\theta}}^T(\overline{\mathbf{z}}-\mathbf{F}_k\overline{\mathbf{d}})\\ &= (-\mathbf{F}_k^T\mathbf{u}_k+\mathbf{A}_k^T\overline{\boldsymbol{\nu}}-\mathbf{F}_k^T\overline{\boldsymbol{\theta}})^T\overline{\mathbf{d}}-\\ &\quad \tfrac{1}{4}\overline{\boldsymbol{\theta}}^T\mathbf{K}_k\overline{\boldsymbol{\theta}}-\overline{\boldsymbol{\nu}}^T\mathbf{b}_k.\end{aligned} \tag{24}$$

Thus, the dual IQP problem is defined by merging (23) and (24) resulting in the quadratic problem

$$\begin{aligned}\underset{\boldsymbol{\nu}\in\mathbb{R}^m,\boldsymbol{\theta}\in\mathbb{R}^d}{\text{maximize}} \quad & \varphi(\boldsymbol{\nu},\boldsymbol{\theta}) = -\tfrac{1}{4}\boldsymbol{\theta}^T\mathbf{K}_k\boldsymbol{\theta}-\boldsymbol{\nu}^T\mathbf{b}_k\\ \text{subject to} \quad & \mathbf{A}_k^T\boldsymbol{\nu}-\mathbf{F}_k^T\boldsymbol{\theta}=\mathbf{F}_k^T\mathbf{u}_k,\\ & \boldsymbol{\nu}\geq\mathbf{0},\end{aligned}$$

which is equivalent to

$$\begin{aligned}\underset{\boldsymbol{\nu},\boldsymbol{\theta}}{\text{minimize}} \quad & \tfrac{1}{4}\boldsymbol{\theta}^T\mathbf{K}_k\boldsymbol{\theta}+\boldsymbol{\nu}^T\mathbf{b}_k\\ \text{subject to} \quad & \mathbf{A}_k^T\boldsymbol{\nu}-\mathbf{F}_k^T\boldsymbol{\theta}=\mathbf{F}_k^T\mathbf{u}_k,\\ & \boldsymbol{\nu}\geq\mathbf{0}.\end{aligned} \tag{IQP_d}$$

The dual problem (IQP_d) is defined with a quadratic convex objective function, n linear equality constraints and $m=2n+1$ bound constraints. The strong duality property for convex problems ensures that $\varphi(\overline{\boldsymbol{\nu}},\overline{\boldsymbol{\theta}})=\hat{f}(\overline{\mathbf{d}})$ see e.g. [9].

In order to recover the primal variables, the optimality conditions of the dual problem (IQP_d) are explicitly obtained. Given the dual Lagrangian function

$$\begin{aligned}\mathcal{L}_d(\boldsymbol{\nu},\boldsymbol{\theta},\boldsymbol{\chi},\boldsymbol{\zeta}) &= \tfrac{1}{4}\boldsymbol{\theta}^T\mathbf{K}_k\boldsymbol{\theta}+\boldsymbol{\nu}^T\mathbf{b}_k\\ &\quad +\boldsymbol{\chi}^T(-\mathbf{F}_k^T\mathbf{u}_k+\mathbf{A}_k^T\boldsymbol{\nu}-\mathbf{F}_k^T\boldsymbol{\theta})-\boldsymbol{\zeta}^T\boldsymbol{\nu},\end{aligned}$$

the optimality conditions are satisfied at $(\overline{\boldsymbol{\nu}},\overline{\boldsymbol{\theta}},\overline{\boldsymbol{\chi}},\overline{\boldsymbol{\zeta}})$

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\mathcal{L}_d(\overline{\boldsymbol{\nu}},\overline{\boldsymbol{\theta}},\overline{\boldsymbol{\chi}},\overline{\boldsymbol{\zeta}}) &= \mathbf{b}_k+\mathbf{A}_k\overline{\boldsymbol{\chi}}-\overline{\boldsymbol{\zeta}}=\mathbf{0}\\ \nabla_{\boldsymbol{\theta}}\mathcal{L}_d(\overline{\boldsymbol{\nu}},\overline{\boldsymbol{\theta}},\overline{\boldsymbol{\chi}},\overline{\boldsymbol{\zeta}}) &= \tfrac{1}{2}\mathbf{K}_k\overline{\boldsymbol{\theta}}-\mathbf{F}_k\overline{\boldsymbol{\chi}}=\mathbf{0}.\end{aligned} \tag{25}$$

Here, $\chi\in\mathbb{R}^n$ and $\boldsymbol{\zeta}\in\mathbb{R}^m$ are the Lagrangian multipliers of the equality and the bound constraints of (IQP_d), respectively.

From Equation (25), $\overline{\boldsymbol{\theta}}=2\mathbf{K}_k^{-1}\mathbf{F}_k\overline{\boldsymbol{\chi}}$ is obtained. In addition, the primal variable $\mathbf{z}$ is related with the dual variable $\boldsymbol{\theta}$ by equation (22). Thus, $\overline{\mathbf{z}}=-\mathbf{F}_k\overline{\boldsymbol{\chi}}$. At the same time, $\mathbf{z}$ was previously defined as $\mathbf{z}=\mathbf{F}_k\mathbf{d}$ (20), then, the optimal primal variable $\overline{\mathbf{d}}$ is equivalent to the negative value of the optimal dual variable of (IQP_d), i.e. $\overline{\mathbf{d}}_k^{iq}=-\overline{\boldsymbol{\chi}}$.

The variable $\overline{\boldsymbol{\nu}}$ collects the inequality, the upper bound, and the lower bound Lagrangian multipliers, i.e. $\overline{\boldsymbol{\nu}}=(\overline{\boldsymbol{\lambda}_k^{iq}},\overline{\boldsymbol{\xi}_k^{iq}},\overline{\boldsymbol{\eta}_k^{iq}})$ with $\boldsymbol{\lambda}_k^{iq}\in\mathbb{R}$, and $\boldsymbol{\xi}_k^{iq},\boldsymbol{\eta}_k^{iq}\in\mathbb{R}^n$.

The main advantage of solving the IQP sub-problem using the dual formulation is the elimination of the inverse of the stiffness matrix. This new formulation is expected to be faster than the alternative primal formulation since there are fewer number of variables and constraints. Therefore, it is chosen for the implementation of TopSQP.

6 Alternative to the EQP system

Analogous to the IQP, the KKT system for the minimum compliance problem is impractical since to the computation of the Hessian is needed. Throughout this section, and with the aim of simplifying the notation, the sub-indices referring to the working set or the complementary working set are omitted, see Section 2.3 for the sake of completeness.

The original EQP system for minimum compliance problem is

$$\begin{bmatrix} \hat{\mathbf{H}}_k & \mathbf{a} \\ \mathbf{a}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_k^{eq} \\ \lambda_k^{eq} \end{bmatrix} = - \begin{bmatrix} -\mathbf{F}_k^T \mathbf{u}_k + \mathbf{H}_k \mathbf{d}_k^{iq} \\ 0 \end{bmatrix} \tag{EQP_0}$$

With $\mathbf{H}_k = \nabla^2 \mathcal{L}(\mathbf{x}_k, \lambda_k^{iq})$, and $\hat{\mathbf{H}}_k$ an approximation of the Hessian such that the inertia of the system (EQP_0) is correct.

Let assume $\hat{\mathbf{H}}_k = 2\mathbf{F}_k^T \mathbf{K}_k^{-1} \mathbf{F}_k - \hat{\mathbf{Q}}_k$ for any $\hat{\mathbf{Q}}_k$ such that $\hat{\mathbf{H}}_k \succ 0$.

The first system of equations for the minimum compliance problem is

$$\begin{aligned} \hat{\mathbf{H}}_k \mathbf{d}_k^{eq} + \mathbf{a}\lambda_k^{eq} \quad &= -(-\mathbf{F}_k^T \mathbf{u}_k + \mathbf{H}_k \mathbf{d}_k^{iq}), \\ &\Downarrow \\ 2\mathbf{F}_k^T \mathbf{K}_k^{-1} \mathbf{F}_k \mathbf{d}_k^{eq} \quad &- \hat{\mathbf{Q}}_k \mathbf{d}_k^{eq} + \mathbf{a}\lambda_k^{eq} = -(-\mathbf{F}_k^T \mathbf{u}_k + \mathbf{H}_k \mathbf{d}_k^{iq}). \end{aligned} \tag{26}$$

In order to reduce the computational cost caused for the dense Hessian, the system is expanded. A new variable $\mathbf{v}_k = 2\mathbf{K}_k^{-1} \mathbf{F}_k \mathbf{d}_k^{eq}$ is included to split equation (26) in two,

$$\begin{aligned} &\tfrac{1}{2}\mathbf{K}_k \mathbf{v}_k - \mathbf{F}_k \mathbf{d}_k^{eq} = \mathbf{0}, \\ &\mathbf{F}_k^T \mathbf{v}_k - \hat{\mathbf{Q}}_k \mathbf{d}_k^{eq} + \mathbf{a}\lambda_k^{eq} = -(-\mathbf{F}_k^T \mathbf{u}_k + \mathbf{H}_k \mathbf{d}_k^{iq}). \end{aligned}$$

It enables to define an expanded EQP symmetric system

$$\begin{bmatrix} -\hat{\mathbf{Q}}_k & \mathbf{F}_k^T & \mathbf{a} \\ \mathbf{F}_k & -1/2\mathbf{K}_k & \mathbf{0} \\ \mathbf{a}^T & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_k^{eq} \\ \mathbf{v}_k \\ \lambda_k^{eq} \end{bmatrix} = - \begin{bmatrix} -\mathbf{F}_k^T \mathbf{u}_k + \mathbf{H}_k \mathbf{d}_k^{iq} \\ \mathbf{0} \\ 0 \end{bmatrix}. \tag{EQP_e}$$

Although the size of the system is increased from $|\mathcal{W}_{k,c}^b| + |\mathcal{W}_k^g|$ to $d + |\mathcal{W}_{k,c}^b| + |\mathcal{W}_k^g|$, it is much faster to solve than (EQP_0) since there are only sparse matrices.

The system can be solved using direct methods such as the null-space method [43]. The computation of a matrix $\mathbf{Z}$ with columns are the basis of the null-space of the Jacobian is cheap, easy, and fast. However, as there is only one constraint, the time reduction is negligible. In addition, the computational cost of the EQP step is not significant for the overall algorithm due to the sparsity of the KKT matrix. Moreover, the density variables tend fast to the bounds, and most of the bound constraints will be active. This produces a meaningful reduction of the size of the system (EQP_e), see (11).

Table 1: Study of the number of iterations required for TopSQP to converge using both EQP+IQP and using only the IQP phase on a test set of 10 small-size problems. The table contains the description of the problem (design domain, length ratios, discretization, and volume fraction) and the number of iterations required for both approaches.

Domain	**Length ratio**	**Discretization**	**Volume**	**TopIQP iterations**	**TopSQP iterations**
Michell	1×1	20×20	0.1	63	36
Michell	1×1	40×40	0.3	85	74
Michell	2×1	40×20	0.1	256	151
Michell	2×1	80×40	0.5	37	30
Michell	3×1	60×20	0.4	137	114
MBB	1×2	40×80	0.3	88	59
MBB	1×4	40×160	0.5	124	113
MBB	2×1	80×40	0.2	169	141
Cantilever	2×1	120×60	0.5	81	78
Cantilever	4×1	80×20	0.2	131	92

7 Implementation

The same approximate Hessian is used for both, the IQP and the EQP phase (see (17)), since preliminary results show that the performance of $\hat{\mathbf{H}}_1$ in the EQP was very similar to $\hat{\mathbf{H}}_2$ and $\hat{\mathbf{H}}_3$. However, the computational time required for $\hat{\mathbf{H}}_2$ and $\hat{\mathbf{H}}_3$ is higher than $\hat{\mathbf{H}}_1$ due to the estimation of γ_1 and γ_2. In addition any inertia correction in the original system will increase the computational time of the algorithm considerably.

The implementation of the proposed algorithm is written in MATLAB [54], and the IQP sub-problem is solved using Gurobi optimizer software [34]. The default method in Gurobi is used in which the QP problem is solved with a barrier algorithm. Although the Gurobi software is very efficient, the IQP phase makes the method expensive. The EQP is, thus, very important to reduce the number of IQP iterations.

The numerical experiments in [40] show the benefits of the EQP in terms of number of iterations. Since the proposed IQP is defined with the same approximate Hessian as the EQP phase, a small preliminary study was performed to study the effects of the EQP step. Table 1 shows the number of iterations needed for TopSQP using only the IQP (namely TopIQP) and using both phases. Although, only ten small problems are considered, similar behaviour was observed for the whole test set of problems. The EQP reduces the total number of iterations. In addition, the cost of the EQP phase is negligible compared to the IQP.

For the estimation of the working set, the linearized constraint $g(\mathbf{x})$ are considered active if:

$$-\epsilon_4 < g(\mathbf{x}_k) + \nabla g(\mathbf{x}_k)^T \mathbf{d}_k^{iq} < \epsilon_4,$$

with $\epsilon_4 = 10^{-4}$. In the same way, the line search is, in practice, more flexible. First of all, a smaller reduction in the merit function than in the original SQP+ algorithm is allowed, using the parameter $\epsilon_5 = 10^{-6}$.

$$\phi_\pi(\mathbf{x}_k + \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}) \leq \phi_\pi(\mathbf{x}_k) - \sigma qred_\pi(\mathbf{d}_k^{iq}) - \epsilon_5.$$

Secondly, in those cases where the previous condition is not satisfied (Step 9 Algorithm 1), instead of doing a line search with only $\mathbf{d}_k^{iq}$, up to 5 consecutive times the full IQP step ($\alpha = 1$) is allowed. In practice, the algorithm will take in most of the iterations a step direction involving both phases: $\mathbf{d}_k = \mathbf{d}_k^{iq} + \beta \mathbf{d}_k^{eq}$. However, in very few examples, this descent direction does not reduce the merit function and it stalls in a local minimum with a small KKT error but not sufficiently small for convergence. In those situations the KKT conditions are satisfied when a full inequality step is forced.

Regarding the finite element analysis, the design domain is discretized using the same size of plane stress elements (with 4 nodes per element), and then the element stiffness matrix is the same for all elements. The code of the finite element analysis is based on [2].

8 Numerical Experiments

The specific purpose TopSQP solver is compared to the first-order structural topology optimization solver, GCMMA [53], and two general purpose solvers, SNOPT [27] and IPOPT [55]. These last two solvers (namely SNOPT and IPOPT-N) use limited memory BFGS approximation of the Hessian. Moreover, the best solver (in terms of objective function value) according to [46], IPOPT-S, is under consideration. IPOPT-S[3] solves the SAND (Simultaneous Analysis and Design, see e.g. [3]) formulation using the exact Hessian. More information about these solvers and their specific parameter settings can be found in [46]. The parameter values set for the TopSQP and Gurobi are gathered in Tables 2 and 3, respectively. In addition, Table 4 contains the parameter values of the minimum compliance problem used for TopSQP. Since IPOPT and SNOPT are also second-order methods, the optimality conditions are set as in TopSQP, i.e. `feas norm` $= 10^{-8}$ and `kkt norm` $= 10^{-6}$. More details of how the KKT norm is obtained in these solvers can be found in [55] and [27]. The stopping criterion of GCMMA is `kkt norm` $= 10^{-4}$ and `feas norm` $= 10^{-8}$ (first-order method). The maximum number of iterations for all the solvers is set to `max iter` $= 1{,}000$. See more details in [46].

[3]For simplicity, the default linear algebra package MUMPS [1] is used in both nested (IPOPT-N) and SAND (IPOPT-S) formulation.

All the computations were done on a Intel Xeon e5-2680v2 ten-core processor, running at 2.8GHz with 64 GB RAM. Only Gurobi (IQP phase) runs in parallel using four threads. TopSQP, IPOPT, SNOPT, and GCMMA all run in serial.

Table 2: Parameter setting for TopSQP. The table contains the name of the parameter, a brief description and the value.

Parameter	**Description**	**Value**
$\mathbf{t}_0$	Starting point	$V\mathbf{e}$[4]
`stat tol`	Stationarity error ϵ_1 (see 2.1)	10^{-6}
`feas tol`	Feasibility error ϵ_2 (see 2.1)	10^{-8}
`comp tol`	Complementarity error ϵ_3 (see 2.1)	10^{-6}
`max iter`	Maximum number of iterations	1,000

Table 3: Parameter setting for Gurobi for solving the IQP sub-problem. The table contains the name of the parameter, a brief description and the value.

Parameter	**Description**	**Value**
`OptimalityTol`	Optimality tolerance	10^{-9}
`FeasibilityTol`	Feasibility tolerance	10^{-9}
`threads`	Number of OMP threads	4
`presolve`	Presolve level	0 (off)

Table 4: Values of charateristic parameters of topology optimization problems solve by TopSQP.

Parameter	**Description**	**Value**
E_v	Young's modulus value for void material	10^{-1}
E_1	Young's modulus value for solid material[5]	10^2
p	SIMP penalization parameter	3
$r_{\min}$	radius for the density filter (L_x is the length in the x direction)	$0.04L_x$

[4]Here, $\mathbf{e}$ refers to a vector of all ones.

[5]A small contrast of E_1/E_v is considered since we are mostly interested in the behaviour of the solvers. In addition, the values of E_v and E_1 are chosen to well-scaled problems for the solvers (see [46] for more details).

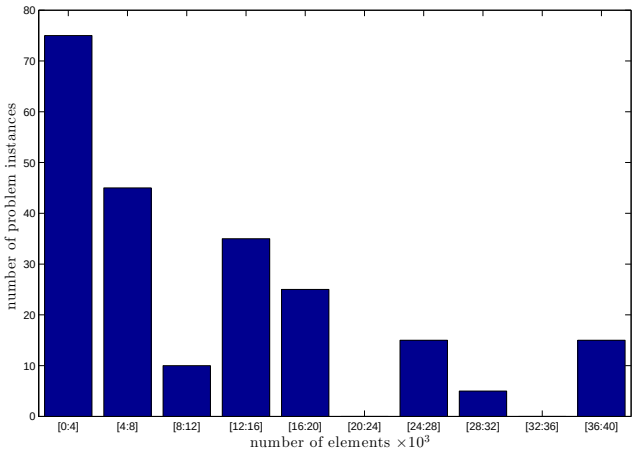


Figure 1: Distribution of the number of instances of the test set for different number of elements ranges.

The numerical experiments to assess the performance of TopSQP are presented using performance profiles, see [16]. The specific minimum compliance test set consists of 225 2D medium-size instances (with 400-40,000 finite elements) as defined in [46]. Figure 1 shows how the problem instances are distributed with regards to the number of elements (size of the problem). Since a time limit is set to 300 hours, the execution of TopSQP (MATLAB general purpose implementation) is not finished[6] for 18 problem instances. In addition, IPOPT-S (SAND formulation) has some issues in the linear algebra[7] for 12 problem instances. Nevertheless, the last intermediate design obtained in these instances is considered as the final design in the benchmarking study.

The performance profiles show the percentage of problems (in the test set) where a solver s obtains different relative ratios of performance (defined with the parameter τ). In other words, for a given solver s, the function ρ_s defined is represented as

$$\rho_s(\tau) = \frac{1}{N}\text{size}\{\tilde{p} \in P : r_{\tilde{p},s} \leq \tau\},$$

or

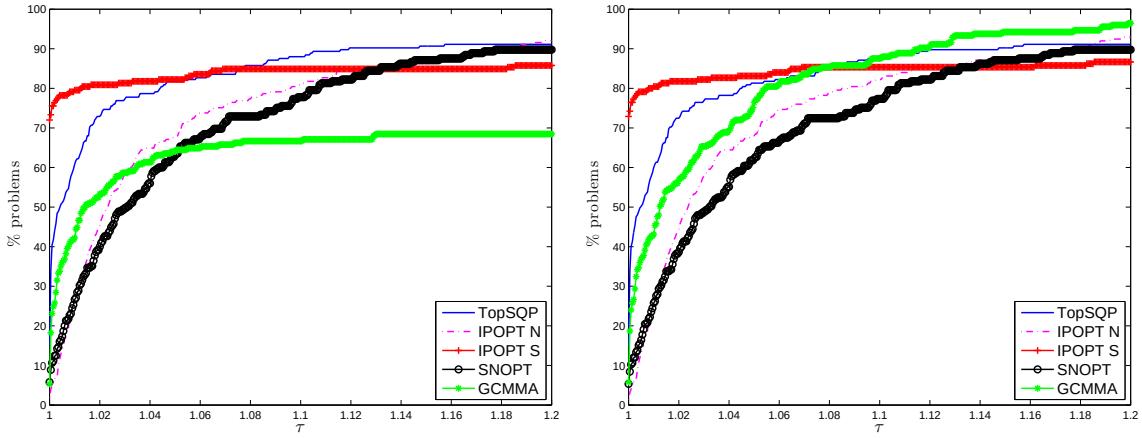
$$\rho_s(\tau) = \frac{1}{N}\text{size}\{\tilde{p} \in P : \log_{10}(r_{\tilde{p},s}) \leq \tau\}.$$

Here, P is the set of problems with $\tilde{p} \in P$ and N the size of P. The ratio of performance for a solver s for each problem $\tilde{p}$ is defined as

$$r_{\tilde{p},s} = \frac{m_{\tilde{p},s}}{\min\{m_{\tilde{p},s} : s \in S\}},$$

[6]The computational time required for the solver is highly dependent on the number of processors, the method, as well as the number of threads used during the execution.

[7]IPOPT needs to reallocate memory and thus, it has difficulties to converge.



(a) Objective function value ($\omega_{\max} = 10^{-3}$). (b) Objective function value ($\omega_{\max} \gg 0$).

Figure 2: Performance profile for a test set of 225 minimum compliance problems. The performance is measured by the objective function value. Figure 2a shows the performance when designs with KKT error higher than $\omega_{\max} = 10^{-3}$ are penalized. Figure 2b shows the performance without any penalization measured, i.e., with $\omega_{\max} \gg 0$.

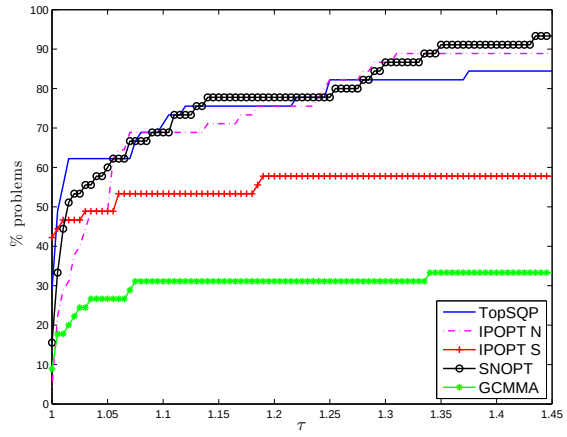
with m a measure of performance, such as

$$m_{\tilde{p},s} = \text{iter}_{\tilde{p},s} = \{ \text{ number of iterations required to solve the problem } \tilde{p} \text{ by a solver } s\}.$$

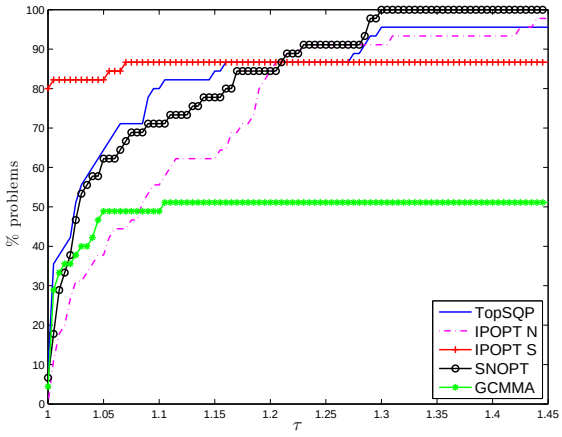
In these numerical experiments, the solvers are compared using four different criteria: the objective function value, the number of iterations (outer iterations), the number of stiffness matrix assemblies, and the computational time.

Furthermore, at the maximum value ratio r_M the performance profiles reflect the robustness of the solvers. In these numerical experiments, a solver fails if the KKT error is larger than $\omega_{\max} = 10^{-3}$. Thus, the term robustness refers to the capability of obtaining a design with a KKT accuracy lower than or equal to 10^{-3}. More details about the impact of this threshold can be found in [46].

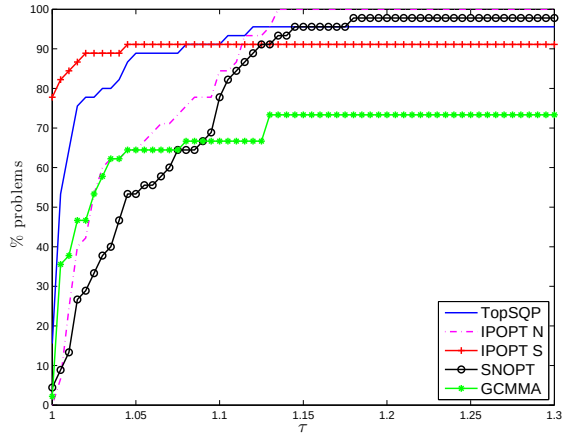
Figure 2a shows the performance profile for the objective function value. IPOPT-S still has the highest number of problems where the designs have the minimum objective function value (70% of the cases). Nevertheless, when τ is relatively small, ($\tau = 1.05$), the performance of TopSQP is the same as IPOPT-S with a success of 81%. TopSQP outperforms the other solvers in terms of objective function value for τ close to 1. Some of the methods presented in the study produce feasible iterates. Although the first-order optimality conditions are not satisfied (KKT error higher than $\omega_{\max}$), the objective function values of feasible designs might still be acceptable. Figure 2b shows the performance profile (for objective function value) when no penalization is applied. The difference is meaningful for GCMMA. In general, its feasible iterates produce reasonably good approximations. Although, the performance of IPOPT-S and TopSQP



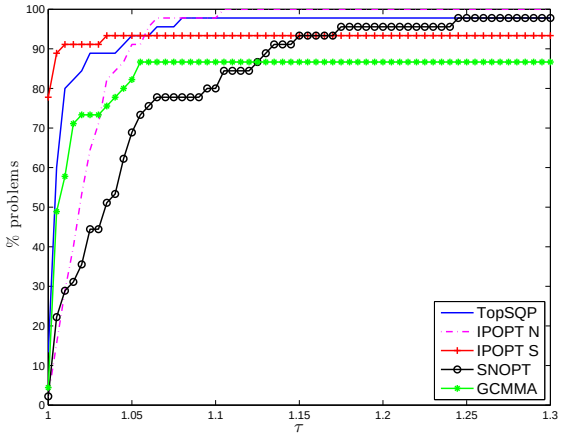
(a) Objective function value ($V = 0.1$)



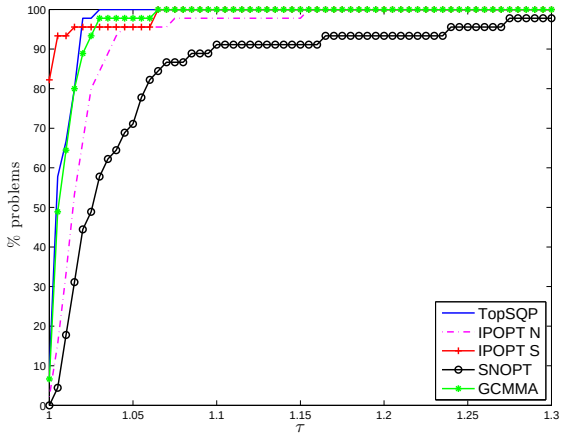
(b) Objective function value ($V = 0.2$)



(c) Objective function value ($V = 0.3$)

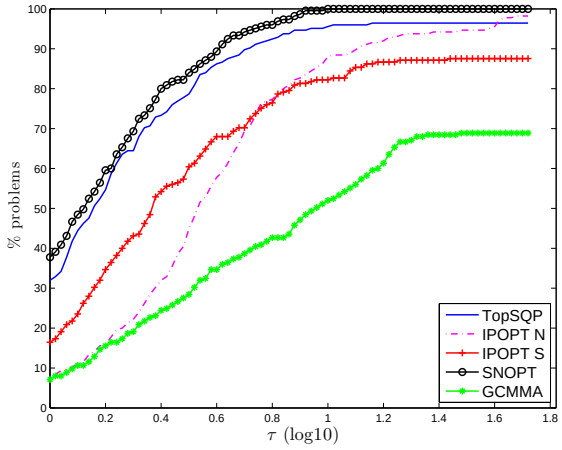


(d) Objective function value ($V = 0.4$)

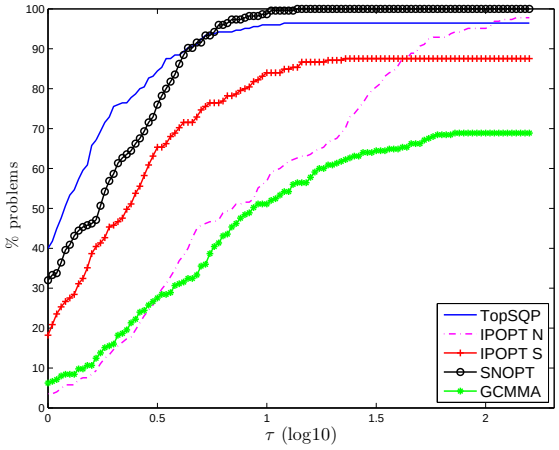


(e) Objective function value ($V = 0.5$)

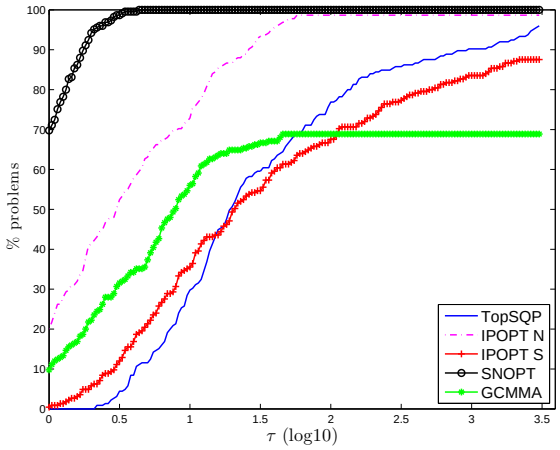
Figure 3: Performance profiles for five different subsets of the minimum compliance problems (45 instances each). The problems are divided depending on their volume fraction. The performance is measured by the objective function value.



(a) Number of iterations



(b) Stiffness matrix assemblies



(c) Computational time

Figure 4: Performance profiles for a test set of 225 minimum compliance problems. The performance is measured by the objective function value (2a), the number of iterations (4a), the number of stiffness matrix assemblies (4b) and the computational time (4c).

is still better, in general all the solvers produce designs with similar objective function values. In 80% of the problems, the difference of objective function values is smaller than 12% (i.e. $\tau = 1.12$).

In almost the 30% of the problems solved with GCMMA, the KKT error is higher than 10^{-3}. Before analysing the performance of the methods for different criteria, it is important to study in detail why GCMMA has difficulties to converge. Figure 3 shows the performance profiles (for objective function value) for smaller test sets of problems. The original test set is partitioned into five. Each subset gathers 45 problems in which the volume fraction is the same ($V = 0.1$, 0.2, 0.3, 0.4, and 0.5). It seems clear that GCMMA has difficulties on solving problems with low volume fractions, see Figures 3a and 3b in which the test set collects problems with $V = 0.1$ and $V = 0.2$, respectively.

Regarding the performance for the number of iterations (Figure 4a), TopSQP produces designs using the smallest number of iterations, with a very similar performance to SNOPT. However, the designs for the latter solver have large objective function values (see Figure 2a). Since TopSQP has two phases, it is also important to compare the performance using other criteria to check the cost of every major iteration. Figure 4b shows the performance profiles when the solvers are compared with the number of stiffness matrix assemblies, which is equivalent to the number of function evaluations. It is outstanding the few number of stiffness matrix assembled for TopSQP. In contrast to SNOPT or IPOPT-S, TopSQP usually evaluates the stiffness matrix once per iteration. This is due to the definition of the line search.

A very important aspect in the comparison of solvers is the computational time required for obtaining a solution. On the other hand, we need to be cautious since the solvers have different interfaces, they are programmed in different languages, and can be linked to different linear algebra packages. The computational time could be highly affected by this. Although it is preferable to compare the computational cost of the solvers using a more objective criterion such as the number of iterations or the number of stiffness matrix assemblies, it is remarkable the amount of time required for TopSQP (Figure 4c). It performs slightly worse than IPOPT-S (SAND formulation). The major amount of time in the proposed method is spent in the IQP sub-problem. More sophisticated and advanced convex quadratic methods must be developed to solve this sub-problem in order to produce an efficient and fast method. Nevertheless, TopSQP is considered a good compromise between accurate and good designs (good objective function values) and computational time.

Finally, TopSQP, IPOPT-N, and SNOPT have excellent robustness properties, i.e. the KKT error is lower than 10^{-3} in about the 96% of the test set. In contrast, the robustness of GCMMA is highly dependent on the problems. GCMMA performs very well with respect to the objective function value when the volume fraction is large, with a robustness of 100% (see Figure 3e). On the other hand, the robustness of GCMMA

drops drastically to 30% for a volume fraction equal to $V = 0.1$.

The use of efficient exact second-order methods (such as IPOPT-S and TopSQP) is essential in topology optimization to produce accurate designs[8] in few iterations. The results confirm that they are better not only than the classical methods but also than other (second-order) methods where the Hessian is approximated using BFGS (SNOPT and IPOPT-N). Only IPOPT-S beats TopSQP for objective function value, although at a small ratio of performance TopSQP is as competent as IPOPT-S in this aspect. In contrast, TopSQP requires fewer function evaluations. In addition, TopSQP has the benefits of solving the nested formulation, for instance it has feasible solutions at intermediate steps, less number of variables, and less memory usage.

9 Limitations of TopSQP in structural topology optimization

The main advantage of the IQP phase for minimum compliance problems is the ease of finding a good positive semi-definite approximation of the Hessian, where its dual is much faster to solve than the primal optimization problem. Moreover, the improvement of the EQP relies on the symmetry of the approximate Hessian. However, these benefits are not satisfied for all classes of problems. For instance, in compliant mechanism design problems [5] and [50], the exact Hessian is

$$\begin{aligned}\nabla^2\mathcal{L}(\mathbf{t},\mathbf{A}) \quad &= \mathbf{F}(\mathbf{t},\mathbf{A})^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t},\mathbf{u}) + \\ &\quad \mathbf{F}(\mathbf{t},\mathbf{u})^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t},\mathbf{A}) - \mathbf{Q}(\mathbf{t},\mathbf{A},\mathbf{u})\end{aligned}$$

where

$$\begin{aligned}\mathbf{Q}(\mathbf{t},\mathbf{A},\mathbf{u}) &= \operatorname{diag}(\mathbf{A}^T(\mathbf{t})\frac{\partial^2\mathbf{K}_i(t_i)}{\partial t_i^2}\mathbf{u}(\mathbf{t})) \\ \mathbf{F}(\mathbf{t},\mathbf{u}) &= \left(\frac{\partial\mathbf{K}_1(t_1)}{\partial t_1}\mathbf{u}(\mathbf{t}) \quad \cdots \quad \frac{\partial\mathbf{K}_n(t_n)}{\partial t_n}\mathbf{u}(\mathbf{t})\right).\end{aligned}$$

and $\mathbf{A}$ the adjoint variable used for the sensitivity analysis.

For this class of problems, the Hessian is more difficult to approximate. It is possible to use simple approximations, but the convergence rate could drop significantly.

Nevertheless, the major limitation of the proposed TopSQP, even for minimum compliance problems, is the computational time required to obtain the optimal solution of the sub-problems. Most of this time is spent in the IQP phase, where a quadratic convex sub-problem is solved. Efficient linear solvers, such as iterative methods are essential to be able to use TopSQP in large-scale topology optimization problems.

[8] Designs with good objective function values and low KKT error

10 Conclusions and further research

An efficient second-order sequential quadratic programming method based on SQP+ from [40] is presented for topology optimization problems (TopSQP). More specifically, the minimum compliance problem is solved using exact information of the Hessian in both, the IQP and EQP phases. An efficient approximate Hessian is used in both phases, producing very accurate estimations of the search direction. These sub-problems are efficiently improved, since they are reformulated taking advantages of the structure of the problem.

The numerical experiments confirm the benefits of using second-order information not only for reducing the number of iterations but also for decreasing the objective function values. IPOPT-S and TopSQP are the most competent methods to produce designs with good objective function value. The comparison of the performance between IPOPT-S and TopSQP (exact Hessian) and SNOPT and IPOPT-N (BFGS approximation), reinforces that the use of information based on the exact Hessian is important to obtain good designs.

Additionally, TopSQP outperforms GCMMA not only in the objective function value but also in the number of iterations, the number of stiffness matrix assemblies, and the overall robustness.

Although IPOPT-S outperforms the other solvers when measuring the objective function value, all the solvers are able to produce similar results. Indeed the objective function value of TopSQP is very close to the minimum possible. In contrast, IPOPT-S requires more iterations and function evaluations than TopSQP. Finally, TopSQP solves the nested formulation with all the advantages that brings with it.

Further work must be done in order to extend and generalize the proposed TopSQP method to solve more general topology optimization problems such as compliant mechanism design problems. Additional investigations are needed to extend the code to be able to solve large-scale problems.

Acknowledgements

We would like to thank Professor Krister Svanberg at KTH in Stockholm for providing the implementation of GCMMA. We extend our sincere thanks to two reviewers and the editor for providing many comments and suggestions that improved the quality of this article.

References

[1] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2–4):501–520, 2000.

[2] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[3] J. S. Arora and Q. Wang. Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization*, 30(4):251–272, 2005.

[4] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):192–202, 1989.

[5] M. P. Bendsøe and O. Sigmund. *Topology optimization: Theory, methods and applications*. Springer, 2003.

[6] P. T. Boggs and J. W. Tolle. A family of descent functions for constrained optimization. *SIAM Journal on Numerical Analysis*, 21(6):1146–1161, 1984.

[7] P. T. Boggs and J. W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

[8] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2010.

[10] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Mathematical Programming*, 48:27–48, 2004.

[11] R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO : An Integrated Package for Nonlinear Optimization. In *Large Scale Nonlinear Optimization*, volume 83, pages 35–59. Springer, 2006.

[12] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 1987.

[13] F. E. Curtis, T. C. Johnson, D. P. Robinson, and A. Wächter. An inexact sequential quadratic optimization algorithm for nonlinear optimization. *SIAM Journal on Optimization*, 24(3):1041–1074, 2014.

[14] F. E. Curtis and J. Nocedal. Flexible penalty functions for nonlinear constrained optimization. *IMA Journal of Numerical Analysis*, 25(4):749–769, 2008.

[15] J. E. Dennis and J. J. Moré. Quasi-Newton Methods, Motivation and Theory. *SIAM Review*, 19(1):46–89, 1977.

[16] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[17] T. Dreyer, B. Maar, and V. Schulz. Multigrid optimization in applications. *Journal of Computational and Applied Mathematics*, 120(1-2):67–84, 2000.

[18] L. F. Etman, A. A. Groenwold, and J. E. Rooda. First-order sequential convex programming using approximate diagonal QP subproblems. *Structural and Multidisciplinary Optimization*, 45(4):479–488, 2012.

[19] R. Fletcher and S. Leyffer. User manual for filterSQP. Technical Report NA/181, University of Dundee Numerical Analysis, 1998.

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

[21] C. Fleury. CONLIN: An efficient dual optimizer based on convex approximation concepts. *Structural Optimization*, 1(2):81–89, 1989.

[22] C. Fleury. Efficient approximation concepts using second order information. *International Journal for Numerical Methods in Engineering*, 28(9):2041–2058, 1989.

[23] C. Fleury. First and second order convex approximation strategies in structural optimization. *Structural Optimization*, 1(1):3–10, 1989.

[24] A. Forsgren. Inertia-controlling factorizations for optimization algorithms. *Applied Numerical Mathematics*, 43(1-2):91–107, 2002.

[25] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

[26] A. Forsgren and W. Murray. Newton methods for large-scale linear inequality-constrained minimization. *SIAM Journal on Optimization*, 7(1):162–176, 1997.

[27] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large -Scale Constrained Optimization. *SIAM Journal on Optimization*, 47(4):99–131, 2005.

[28] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User's guide for NPSOL 5.0: A fortran package for nonlinear programming. Technical report, Systems Optimization Laboratory, Department of Opertations Research, Stanford University, 1998.

[29] P. E. Gill and D. P. Robinson. A globally convergent stabilized SQP method. *SIAM Journal on Optimization*, 23(4):1983–2010, 2013.

[30] P. E. Gill and E. L. Wong. Sequential quadratic programming methods. *Mixed Integer Nonlinear Programming, in the IMA Volumes in Mathematics and its Applications*, 154:147–224, 2012.

[31] P. E Gill and E. L. Wong. Convexification schemes for SQP methods. Technical Report CCoM 14-6, Center for Computational Mathematics, University of California, San Diego, 2014.

[32] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 32(1):90–99, 1985.

[33] N. I. M. Gould and D. P. Robinson. A second derivative SQP method: Global convergence. *SIAM Journal on Optimization*, 20(4):2023–2048, 2010.

[34] Inc. Gurobi Optimization. Gurobi 12.6.0 Reference Manual, 2010.

[35] W. W. Hager. Stabilized sequential quadratic programming. In *Computational Optimization*, pages 253–273. Springer, 1999.

[36] N. J. Higham and S. H. Cheng. Modifying the inertia of matrices arising in optimization. *Linear Algebra and its Applications*, 275–276:261–279, 1998.

[37] IBM. ILOG. ILOG CPLEX, Reference Manual, 2007.

[38] S. Leyffer and A. Mahajan. Software for Nonlinearly Constrained Optimization. Technical Report ANS/MCS-P1768-0610, Mathematics and Computer Science Division, Argonne National Laboratory, 2010.

[39] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[40] J. L. Morales, J. Nocedal, and Y. Wu. A sequential quadratic programming algorithm with an additional equality constrained phase. *Journal of Numerical Analysis*, 32(2):553–579, 2010.

[41] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20(3):286–307, 1994.

[42] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117 – 129, 1987.

[43] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[44] C. E. Orozco and O. N. Ghattas. A reduced SAND method for optimal design of nonlinear structures. *International Journal for Numerical Methods in Engineering*, 40(15):2759–2774, 1997.

[45] F. J. Prieto. Sequential quadratic programming algorithms for optimization. Technical Report SOL 89-7, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1989.

[46] S. Rojas-Labanda and M. Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization, In print*, 2015. DOI: 10.1007/s00158-015-1250-z.

[47] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural Optimization*, 4(3–4):250–252, 1992.

[48] K. Schittkowski. NLPQLP : A new fortran implementation of a Sequential Quadratic Programming algorithm. Technical report, Department of Mathematics, University of Bayreuth, 2002.

[49] C. Shen, W. Xue, and X. Chen. Global convergence of a robust filter SQP algorithm. *European Journal of Operational Research*, 206(1):34–45, 2010.

[50] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, 25(4):492–526, 1997.

[51] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(2):68–75, 1998.

[52] K. Svanberg. The method of moving asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

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

[54] Inc. The MathWorks. Optimization Toolbox User's Guide R 2013 b, 2013.

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

[56] B. Wang and D. Pu. Flexible penalty functions for SQP algorithm with additional equality constrained phase. Technical report, Proceedings of the 2013 International Conference on Advance Mechanic System, China, September 25-27, 2013.

[57] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):309–336, 1991.

9

Article IV: Solving large-scale structural topology optimization problems using a second-order interior point method

To be submitted:
Rojas-Labanda, S. and Stolpe, M.: Solving large-scale structural topology optimization problems using a second-order interior point method

Solving large-scale structural topology optimization problems using a second-order interior point method*

Susana Rojas-Labanda[+] and Mathias Stolpe[+]

[+]DTU Wind Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark. E-mail: srla@dtu.dk, matst@dtu.dk

Abstract

The article presents an efficient iterative method for the indefinite saddle-point systems that arise in optimization methods when solving structural topology optimization problems. In particular, the density-based minimum compliance problem is solved with an interior point method based on an adaptive barrier parameter scheme.

Interior point methods are one of the most powerful nonlinear solvers, but for large-scale problems, the computational bottleneck is the solution of the saddle-point system. The proposed interior point method TopIP reduces the computational time by solving this system with an specific purpose iterative method that combines Krylov sub-space methods and multigrid cycles for preconditioners.

TopIP is numerically tested on a test set of large-scale 3D topology optimization problems. The results show good convergence and robustness properties. The performance of TopIP is comparable to GCMMA in objective function values with a better convergence rate. The proposed iterative method allows TopIP to solve problems with more than three million degrees of freedom.

Keywords: Topology optimization, Minimum compliance, Interior point methods, Saddle-point systems, Iterative methods

Mathematics Subject Classification 2010: 74P05, 74P15, 90C30, 90C46, and 90C90

*This research is funded by the Villum Foundation through the research project Topology Optimization - The Next Generation (NextTop).

1 Introduction

Structural topology optimization finds an optimal distribution of the material in a design domain by minimizing an objective function under certain constraints. The design domain is usually discretized using finite elements and a design variable is associated with each element. For more details of topology optimization problems see e.g. the text book [10].

More specifically, this study is focused on the minimum compliance problem with a constraint on the total volume of the structure. The topology optimization problem is solved in the nested formulation based on the Solid Isotropic Material with Penalization (SIMP) approach (see e.g. [8], [59], and [80]) combined with a density filter for regularization [19] and [64]. This article develops and benchmarks iterative methods for solving the saddle-point problems arising in interior point methods for structural topology optimization. A primal dual line search interior point method is implemented in which an approximate positive semi-definite Hessian is used to ensure descent directions for a merit function [56].

General purpose nonlinear optimization methods such as Sequential Quadratic Programming (SQP) [16] and interior point methods [31], can be used to solve topology optimization problems, see for instance the benchmarking study in [57]. Here, the SQP method in SNOPT [32] and the interior point method in IPOPT [72] produce better results than the classical structural topology optimization methods in terms of number of iterations and objective function values, respectively. The numerical results obtained by a special-purpose SQP method in [56] conclude that the use of second-order information indeed reduces the number of optimization iterations compared to first-order methods. The objective function value is lower but the computational time is, in general, very large (compared to first-order methods).

Both interior point and SQP methods solve sequences of sub-problems where the second-order information is involved. In fact, the solution of the sub-problem in interior point methods is equivalent to the solution of saddle-point systems[1]. This is the main computational bottleneck of the algorithm. Topology optimization problems typically

[1]Saddle-point systems are a particular type of indefinite linear systems. They are commonly formulated as a 2×2 block linear systems like

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix},$$

where the matrices $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{C}$ satisfy at least one of these conditions [13]:

- $\mathbf{A}$ is symmetric or $\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$ is positive definite.
- $\mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B}$.
- $\mathbf{C}$ is symmetric and positive semi-definite or $\mathbf{C} = \mathbf{0}$.

contain a large number of variables. Therefore, efficient techniques must be developed to solve these indefinite linear systems. Theory and numerical methods to solve saddle-point problems are reviewed in [13].

Direct and iterative methods can be used to solve linear systems. Although the former are still preferable in optimization due to their robustness, iterative methods have become very popular [61]. Large-scale problems provide challenges to direct methods because of memory and time demands. In contrast, iterative methods have a lower storage need and are parallelizable. For more details of iterative solvers, see for instance [42] and [61]. An important development was the combination of Krylov sub-space methods and preconditioners, providing efficient techniques comparable to direct solvers, see e.g. [15], [61], and [68]. Classical examples of preconditioning are the incomplete Cholesky factorization [43], sparse approximate inverse [14], and Richardson iteration type methods such as Jacobi or Gauss-Seidel [42] and [13]. Nowadays, multigrid methods are becoming very popular. Their theoretical convergence rate is asymptotically linearly to the number of unknowns and does not depend on the mesh discretization (conditioning of the matrix) [61], [76], [18], and [65].

Several articles present new alternatives to reduce the computational time in topology optimization problems, see for instance [5], [75], [4], and [1]. In all these articles, the topology optimization problem is solved in the nested formulation using first-order methods such as the Optimality Criteria (OC) method (see e.g. [58] and [6]) or the Method of Moving Asymptotes (MMA) [66]. Most of them are focused on reducing the computational time spent in the solution of the equilibrium equations (which are implicitly used in the objective function) since this is the most expensive step in those algorithms. Krylov sub-space methods, such as the Conjugate Gradient (CG) method [37], the Minimal Residual (MINRES) method [55], and the Generalized Minimal Residual (GMRES) method [62], are presented for topology optimization. In addition, several articles discuss the use of parallel computing in combination with domain decomposition for solving these equations, see e.g. [71], [17], [48], [29], and [2].

However, very few articles investigate the reduction of the computational time required in the saddle-point systems arising in interior point or SQP methods in topology optimization. . For instance, [47] and [26] present multigrid strategies to solve the Simultaneous Analysis and Design (SAND) formulation ([7] and [54]).

The article is organized as follows. The topology optimization problem is described in Section 2. Section 3 contains a brief description of the implemented interior point method while Sections 4 and 5 are focused on the iterative methods developed for the saddle-point problem. Section 6 collects all the implementation details needed to reproduce the numerical experiments gathered in Section 7. Finally, Section 8 contains the main conclusions and a brief description of future work.

Notation

Throughout this article, matrices are denoted with capital bold letter such as $\mathbf{H}$ and $\mathbf{A}$. Lower case bold letters represent vectors, for instance $\mathbf{u}$, $\mathbf{v}$, and $\boldsymbol{\lambda}$. Correspondingly, scalars are denoted with lower case letters. For convenience, matrices and vectors sometimes depend on other vectors, for instance, $\mathbf{J}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$. $\mathbf{I}$ and $\mathbf{e}$ denote the identity matrix and a vector of all ones of suitable size, respectively. The expression "$\mathbf{A} \succ \mathbf{B}$" ("$\mathbf{A} \succeq \mathbf{B}$") means that $\mathbf{A} - \mathbf{B}$ is positive definite (or semi-definite) matrix. Finally, the diagonal matrix formed by placing the elements of a given vector on the diagonal is denoted by $\mathrm{diag}(\mathbf{x})$.

2 Problem formulation

Topology optimization determines the optimal distribution of material in a prescribed design domain given a set of loads and boundary conditions. Topology optimization problems are generally defined as nonlinear constrained optimization problems. They are in applications often characterized as large-scale optimization problems. A detailed description of topology optimization problems and associated applications can be found in [10].

The design domain is usually discretized using finite element analysis. The design variables are related to the finite elements as thickness, densities, or material properties. Thus, the elements describe the topology and are also used to evaluate the objective function and the constraints. In particular, this article is focused on the minimum compliance problem with a limitation on the total volume of the structure. For simplicity, distribution of an isotropic material, linear elasticity, and a constant Young's modulus and Poisson's ratio values in each element, are assumed.

A material interpolation approach [9] is used to penalize intermediate densities to produce an almost solid-and-void design. The SIMP approach is chosen in combination with a density filter to avoid numerical instabilities such as mesh-dependency and checkerboards [64]. Thus, the stiffness matrix is defined as

$$\mathbf{K}(\mathbf{t}) = \sum_{e=1}^{n} E(t_e)\mathbf{K}_e,$$

with

$$E(t_e) = E_v + (E_1 - E_v)\tilde{t}_e^p,$$

and $p \geq 1$ is the SIMP penalty parameter.

Here, n is the number of elements, $E_v > 0$, and $E_1 \gg E_v$ represent the "void" and solid Young's modulus, respectively, and $\mathbf{K}_e$ is the element stiffness matrix of unit density. The density variable is represented with $\mathbf{t} \in \mathbb{R}^n$ and $\tilde{t}_e$ refers to the filtered density of the eth element (see e.g. [57], [6], and [45]). The stiffness matrix $\mathbf{K}(\mathbf{t}) : \mathbb{R}^n \mapsto \mathbb{R}^{d \times d}$ is assumed to

be positive definite for all $t_i \geq 0$, to avoid singularity, with d being the number of degrees of freedom.

Throughout this article, the problem is described in a nested formulation. This means, the displacements $\mathbf{u} \in \mathbb{R}^d$ are related to the design variable, $\mathbf{t}$, through the discretized equilibrium equations

$$\mathbf{K}(\mathbf{t})\ \mathbf{u} = \mathbf{f}. \tag{1}$$

Here, $\mathbf{f} \in \mathbb{R}^d$ is the static and design-independent external load vector. The minimum compliance problem is

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{v}^T\mathbf{t} \leq V, \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}. \end{aligned} \tag{P_N^c}$$

The relative volume of the elements is defined by $\mathbf{v} = (v_1, \ldots, v_n)^T$ with $v_i > 0$. For simplicity, all elements have the same volume, i.e., $v_i = v_j \quad \forall i, j$. Finally, $1 > V > 0$ is the volume fraction upper limit.

The minimum compliance problem can alternatively be described in a SAND formulation. Here, $\mathbf{u}$ and $\mathbf{t}$ are treated as independent variables, and the equilibrium equations are explicitly considered as equality constraints. The objective function in the SAND formulation is a linear function whereas in the nested form is nonlinear and generally (for $p > 1$) nonconvex. In the latter, the equilibrium equations, implicit in the objective function, need to be solved at each function evaluation. Thus, the objective function becomes computationally expensive for problems with many degrees of freedom. Nevertheless, the problem is described as (P_N^c) since only linear constraints are involved. Therefore, the implementation of the proposed optimization solver does not need to handle infeasibility and unboundedness issues. Another reason of choosing a nested instead of a SAND formulation is due to the large demand of memory and time the later requires [57].

The major drawback of solving the problem (P_N^c) using first-order methods, such as OC and MMA, is the solution of the equilibrium equations. Krylov sub-space methods combined with multigrid techniques are developed to solve (1), see e.g. [5] and [75] among others. The bottleneck of second-order solvers, such as interior point and SQP methods, in addition to the solution of the equilibrium equations, is the solution of the sub-problems where the Hessian (or an approximation of it) is required.

The Hessian of the compliance ([28]) is given by,

$$\mathbf{H}(\mathbf{t}) = 2\mathbf{F}(\mathbf{t})^T\mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) - \mathbf{Q}(\mathbf{t})$$

with

$$\begin{aligned} \mathbf{Q}(\mathbf{t}) &= \operatorname{diag}(\mathbf{u}^T(\mathbf{t})\frac{\partial^2\mathbf{K}_i(t_i)}{\partial t_i^2}\mathbf{u}(\mathbf{t})) : \mathbb{R}^n \to \mathbb{R}^{n\times n}, \\ \mathbf{F}(\mathbf{t}) &= \begin{pmatrix} \frac{\partial\mathbf{K}_1(t_1)}{\partial t_1}\mathbf{u}(\mathbf{t}) & \cdots & \frac{\partial\mathbf{K}_n(t_n)}{\partial t_n}\mathbf{u}(\mathbf{t}) \end{pmatrix} : \mathbb{R}^n \to \mathbb{R}^{d\times n}. \end{aligned}$$

The computation and storage of the Hessian are, in practice, impossible, since the inverse of the stiffness matrix is involved and the Hessian becomes dense. In addition, the Hessian is, in general, indefinite. This produces theoretical and numerical difficulties to second-order optimization methods. With the aim to overcome these issues, [56] proposed an approximate Hessian $\hat{\mathbf{H}}(\mathbf{t})$, that is positive semi-definite for all $\mathbf{0} \leq \mathbf{t} \leq \mathbf{1}$ under the natural assumption that $\mathbf{K}(\mathbf{t}) \succ 0$,

$$\hat{\mathbf{H}}(\mathbf{t}) = 2\mathbf{F}(\mathbf{t})^T \mathbf{K}^{-1}(\mathbf{t})\ \mathbf{F}(\mathbf{t}). \tag{2}$$

Although the sub-problems with $\mathbf{H}(\mathbf{t})$ are replaced by $\hat{\mathbf{H}}(\mathbf{t})$ are convex, they are still computationally very expensive since $\hat{\mathbf{H}}(\mathbf{t})$ is also a dense matrix. The iterate sub-problems can be reformulated thanks to the specific mathematical structure of $\hat{\mathbf{H}}(\mathbf{t})$, avoiding its computation and storage [56].

3 An interior point method for topology optimization

Interior point methods have become very popular for solving nonlinear constrained optimization problems, and they are considered one of the most powerful algorithms for large-scale problems, see for instance the comparative studies in [25] and [12]. Several articles expose the theoretical properties of interior point methods for nonlinear and nonconvex programming, see e.g. [31], [79], and [27] among others. Furthermore, plenty of literature can be found regarding different primal dual interior point implementations for nonconvex optimization problems, see for instance [72] (IPOPT), [41] (IPSOL), [11], and [70] (LOQO).

As far as solving topology optimization problems concerns, interior point methods can produce designs with very good objective function values, when the second-order information is used, see for instance the great performance of IPOPT (solving the SAND formulation) in the benchmarking study [57]. Moreover, this algorithm has already been applied in topology optimization problems in the SAND formulation, see e.g. [40], [39], [38], and [47]. Similar performance is expected for an efficient interior point method applied to the nested form using the second-order information.

Throughout this section, an interior point method is introduced, in which a line search combined with a merit function is used to guarantee convergence to a KKT (Karush-Kuhn-Tucker) point. The implementation of the interior point solver is based on a combination of the adaptive barrier parameter update in [70] and the line search in [74]. It also includes a monotone safeguard version to ensure robustness. The most complex operation of interior point methods is the solution of a saddle-point system to compute the search direction. The goal of this article is the implementation of efficient techniques for solving large-scale indefinite linear systems rather than the discussion of a novel interior point method.

The performance of the interior point method in terms of time, number of iterations, and accuracy is expected to be good since *(i)* the saddle-point system is solved fast and with low memory requirements due to the use of iterative methods, *(ii)* an adaptive barrier parameter updated scheme is implemented improving the convergence rate of the algorithm, and *(iii)* second-order information of the Hessian is used.

3.1 Interior point approach

Consider the general optimization problem

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0 \qquad i = 1,\dots,m, \\ & l_i \leq x_i \leq u_i \quad i = 1,\dots,n. \end{aligned} \tag{3}$$

Here, the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the inequality constraints $g_i : \mathbb{R}^n \to \mathbb{R}$ are assumed to be smooth. The terms $l_i > -\infty$ and $u_i < +\infty$ represent the lower and the upper bounds of the variables, respectively. The formulation of the problem assumes bound constraints on all variables to resemble the particular problem under consideration, i.e., the minimum compliance problem (P_N^c) in which the design variables are bounded.

Slack variables $\mathbf{s} \geq \mathbf{0} \in \mathbb{R}^m$ are introduced to transform the inequality constraints $g_i(\mathbf{x}) \leq 0$ to equality constraints $g_i(\mathbf{x}) + s_i = 0$. The *barrier problem* associated with the problem is

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n, \mathbf{s}\in\mathbb{R}^m}{\text{minimize}} \quad & f(\mathbf{x}) + \mu\phi(\mathbf{x},\mathbf{s}) \\ \text{subject to} \quad & g_i(\mathbf{x}) + s_i = 0 \qquad i = 1,\dots,m. \end{aligned} \tag{4}$$

Here, $\mu > 0$ is the barrier parameter, and

$$\phi(\mathbf{x},\mathbf{s}) = -\sum_{i=1}^{m} \ln(s_i) - \sum_{i=1}^{n} \ln(u_i - x_i) - \sum_{i=1}^{n} \ln(x_i - l_i)$$

the barrier function (with ln the natural logarithm).

There are (at least) three main aspects to take into account for a globally convergent interior point method; how to treat nonconvexity, how to update the barrier parameter, and how to ensure progress towards a KKT point. The first aspect is dealt with in Section 5, where the solution of the saddle-point system is discussed for the minimum compliance problem. The remaining questions are detailed throughout this section.

Algorithm 1 describes the interior point strategy based on the adaptive barrier parameter updated scheme from [52]. Since bounded variables and feasible problems are assumed, the algorithm does not need to include control techniques as in other general frameworks, such as [72]. Adaptive schemes allow some flexibility in both, the barrier parameter sequence and the reduction in the merit function. This approach updates the barrier parameter every iteration, allowing μ to both increase and decrease.

Algorithm 1 Adaptive interior point algorithm [52].

Input: Define starting point $(\mathbf{x}_0, \mathbf{s}_0, \boldsymbol{\lambda}_0, \boldsymbol{\xi}_0, \boldsymbol{\eta}_0)$, $\mu_{\min} = 10^{-7}$, $\tau = 0.995$,$\tau_a = 0.9999$, $\rho = 10^{-8}$, $\theta = 0.5$, $\gamma = 0.2$, $\delta = 1.5$, $l_{\max} = 15$, $\kappa = 1$, and the tolerances ϵ_1, ϵ_2 and ϵ_3.

repeat
  Adaptive scheme
  Compute μ_k as in (15).
  Solve the linear system (10) to compute the primal dual search direction $\boldsymbol{\Delta}$.
  Determine the maximum step size $\alpha_p^{\max}$ and $\alpha_d^{\max}$ as in (11).
  Compute φ_{k+1}; the Euclidean norm of the KKT conditions of the sub-problem.
  Obtain $M_k = \max\{\varphi_{k-l_{\max}}, \ldots, \varphi_k\}$.
  if $\varphi_{k+1} \leq \tau_a M_k$ **then**
    Accept the new iterate $(\mathbf{x}_{k+1}, \mathbf{s}_{k+1}, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\eta}_{k+1})$.
    Check convergence of the problem as in (18).
    Include φ_{k+1} in M_{k+1}.
  else
    Monotone scheme
    Define starting point as $(\mathbf{x}_{k+1}, \mathbf{s}_{k+1}, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\eta}_{k+1})$, and initialize $i, j = 0$.
    Define μ_i as in (17).
    repeat
      repeat
        Solve the linear system (10) to compute the primal dual search direction $\boldsymbol{\Delta}$.
        Determine the maximum step size $\alpha_p^{\max}$ and $\alpha_d^{\max}$ as in (11).
        Perform a backtracking line search to compute the final steps α_p and α_d as in (12).
        Accept the new iterate $(\mathbf{x}_{j+1}, \mathbf{s}_{j+1}, \boldsymbol{\lambda}_{j+1}, \boldsymbol{\xi}_{j+1}, \boldsymbol{\eta}_{j+1})$.
        Compute φ_{j+1}; the Euclidean norm of the KKT conditions of the sub-problem.
        Check convergence of sub-problem for ϵ_{μ_1}, ϵ_{μ_2}, and ϵ_{μ_3} as in (18).
        if $\varphi_{j+1} \leq \tau_a M_k$ **then**
          `go-to-adaptive = true`.
        end if
        $j = j + 1$.
      until `convergence sub-problem` or `go-to-adaptive`
      Decrease μ_{i+1} as in (16).
      $i = i + 1$.
      Check convergence of the problem as in (18).
    until `convergence` or `go-to-adaptive`
    Include φ_{j+1} in M_{k+1}.
  end if
  $k = k + 1$.
until `convergence`

They are usually more efficient than the monotone version (faster convergence rate). The main drawback of these strategies is that they do not ensure global convergence. On the other hand, the monotone approach solves the sub-problem (4) for a fixed μ, and then monotonically, the barrier parameter is decreased until convergence. The presented interior point method combines an adaptive and a monotone barrier parameter update. Once the adaptive approach does not accept a point, the monotone version is used to guarantee global convergence. It is similarly defined as in [52] with a line search strategy inspired by [74].

3.2 Obtaining the search direction

The Lagrangian function associated with problem (4) is

$$\mathcal{L}(\mathbf{x},\mathbf{s},\boldsymbol{\lambda};\mu) = f(\mathbf{x}) + \mu\phi(\mathbf{x},\mathbf{s}) + \sum_{i=1}^{m}\lambda_i(g_i(\mathbf{x})+s_i).$$

Here, $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_m)^T$ are the Lagrangian multipliers of the constraints. The perturbed first-order KKT conditions (5)-(9) gather the first-order necessary conditions for a primal dual point ($\overline{\mathbf{x}}$, $\overline{\mathbf{s}}$, $\overline{\boldsymbol{\lambda}}$, $\overline{\boldsymbol{\xi}}$, $\overline{\boldsymbol{\eta}}$) to be an optimal solution of (4) for a given μ, [53] and [46].

$$\nabla_{\mathbf{x}}\mathcal{L} = \nabla f(\overline{\mathbf{x}}) + \overline{\boldsymbol{\xi}} - \overline{\boldsymbol{\eta}} + \mathbf{J}(\overline{\mathbf{x}})^T\overline{\boldsymbol{\lambda}} = \mathbf{0}, \tag{5}$$

$$\nabla_{\mathbf{s}}\mathcal{L} = \mathbf{D}(\overline{\mathbf{s}})\overline{\boldsymbol{\lambda}} - \mu\mathbf{e} = \mathbf{0}, \tag{6}$$

$$\mathbf{g}(\overline{\mathbf{x}}) + \overline{\mathbf{s}} = \mathbf{0}, \tag{7}$$

$$\mathbf{U}(\overline{\mathbf{x}})\overline{\boldsymbol{\xi}} - \mu\mathbf{e} = \mathbf{0}, \tag{8}$$

$$\mathbf{L}(\overline{\mathbf{x}})\overline{\boldsymbol{\eta}} - \mu\mathbf{e} = \mathbf{0}. \tag{9}$$

Here, $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_n)^T \geq \mathbf{0}$, $\boldsymbol{\eta} = (\eta_1,\ldots,\eta_n)^T \geq \mathbf{0}$ are the Lagrangian multipliers of the upper and the lower bounds respectively, and $\mathbf{g}(\mathbf{x}) = [g_i(\mathbf{x})]_{i=1,\ldots,m}$. The Jacobian of the inequality constraints is denoted with $\mathbf{J}(\mathbf{x}) = [\nabla g_i(\mathbf{x})^T]_{i=1,\ldots,m} : \mathbb{R}^n \longmapsto \mathbb{R}^{m\times n}$, and the remaining of the matrices are defined as $\mathbf{D}(\mathbf{s}) = \text{diag}(\mathbf{s})$, $\mathbf{U}(\mathbf{x}) = \text{diag}(\tilde{\mathbf{u}})$, and $\mathbf{L}(\mathbf{x}) = \text{diag}(\tilde{\mathbf{l}})$, with $\tilde{u}_i = u_i - x_i$ and $\tilde{l}_i = x_i - l_i$. A point satisfying the KKT conditions is commonly called a KKT point.

The sub-problem (4) is solved by applying Newton's method to the first-order optimality conditions (5)-(9) for a given value of μ. The search direction $\boldsymbol{\Delta}$, at the kth interior point iteration, is obtained by solving a KKT system[2] evaluated at the iterate ($\mathbf{x}_k$,$\mathbf{s}_k$,$\boldsymbol{\lambda}_k$, $\boldsymbol{\xi}_k$,$\boldsymbol{\eta}_k$). For notational convenience, the sub-index kth has been removed.

[2]Saddle-point problem $\nabla\mathbf{F}\boldsymbol{\Delta} = -\mathbf{F}$, where $\mathbf{F}$ represents the KKT conditions.

The KKT system to be solved is condensed to a symmetric matrix resulting in

$$\begin{bmatrix} \mathbf{H}(\mathbf{x})+\boldsymbol{\Sigma}_u+\boldsymbol{\Sigma}_l & \mathbf{0} & \mathbf{J}(\mathbf{x})^T \\ \mathbf{0} & \boldsymbol{\Sigma}_s & \mathbf{I} \\ \mathbf{J}(\mathbf{x}) & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{s} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x})+\mathbf{J}(\mathbf{x})^T\boldsymbol{\lambda}+\mu\mathbf{U}(\mathbf{x})^{-1}\mathbf{e}-\mu\mathbf{L}(\mathbf{x})^{-1}\mathbf{e} \\ \boldsymbol{\lambda}-\mu\mathbf{D}(\mathbf{s})^{-1}\mathbf{e} \\ \mathbf{g}(\mathbf{x})+\mathbf{s} \end{bmatrix}. \tag{10}$$

Here, $\boldsymbol{\Sigma}_s = \mathbf{D}(\mathbf{s})^{-1}\boldsymbol{\lambda}$, $\boldsymbol{\Sigma}_u = \mathbf{U}(\mathbf{x})^{-1}\boldsymbol{\xi}$, $\boldsymbol{\Sigma}_l = \mathbf{L}(\mathbf{x})^{-1}\boldsymbol{\eta}$, and $\mathbf{H}(\mathbf{x})$ the Hessian of the Lagrangian (which for this problem coincides with the Hessian of the objective function). After solving (10), the search direction of the bound Lagrangian multipliers are obtained,

$$\begin{aligned} \Delta\boldsymbol{\xi} &= -\boldsymbol{\xi}+\mu\mathbf{U}(\mathbf{x})^{-1}\mathbf{e}+\boldsymbol{\Sigma}_u\Delta\mathbf{x}, \\ \Delta\boldsymbol{\eta} &= -\boldsymbol{\eta}+\mu\mathbf{L}(\mathbf{x})^{-1}\mathbf{e}-\boldsymbol{\Sigma}_l\Delta\mathbf{x}. \end{aligned}$$

The computational time of an interior point iteration is dominated by the solution of the system (10). Section 4 is focused on presenting an efficient iterative method to solve it.

3.3 Updating the iterates

Line search [51] or trust region [23] methods are generally included in nonlinear optimization solvers to guarantee convergence to a KKT point [53]. In particular, the proposed interior point method is based on a line search strategy combined with a reduction in a merit function. Once the search direction $\boldsymbol{\Delta}$ is determined, the primal and dual step lengths, α_p and α_d, are estimated in order to obtain the new iterate $(\mathbf{x}_{k+1}, \mathbf{s}_{k+1}, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\eta}_{k+1})$. First, the maximum step lengths are computed, $\alpha_p^{\max}$ and $\alpha_d^{\max}$, such that the new iterate remains feasible. The fraction to the boundary rule (11) is used for estimating these values in order to prevent that the primal and dual variables approach their bounds too quickly [53]. The step lengths are determined from

$$\begin{aligned} \alpha_p^{\max} &= \max\{\alpha\in(0,1] : \tilde{\mathbf{z}}+\alpha\Delta\tilde{\mathbf{z}} \geq (1-\tau)\tilde{\mathbf{z}}\} \\ \alpha_d^{\max} &= \max\{\alpha\in(0,1] : \boldsymbol{\chi}+\alpha\Delta\boldsymbol{\chi} \geq (1-\tau)\boldsymbol{\chi}\}, \end{aligned} \tag{11}$$

with $\tau\in(0,1)$. The maximum operator is defined as component-wise, the dual variables are gathered in $\boldsymbol{\chi} = (\boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\eta})$, and $\tilde{\mathbf{z}}$ denotes $\tilde{\mathbf{z}} = (\mathbf{s}, \tilde{\mathbf{u}}, \tilde{\mathbf{l}})$.

Additionally, the Armijo condition (see e.g. [53] and [41]) must be satisfied for the values α_p and α_d obtained within the interval

$$\alpha_p\in(0,\alpha_p^{\max}] \qquad \alpha_d\in(0,\alpha_d^{\max}].$$

A backtracking line search is performed to compute the α_p and α_d such that a merit function ψ is reduced. The line search strategy is based on [74]. A maximum possible

value of $\alpha_T = \{1, \theta, \theta^2 ...\}$ with $\theta \in (0,1)$ is selected such that the following inequality is satisfied,

$$\psi_\pi(\mathbf{z} + \alpha_T \alpha_p^{\max} \Delta\mathbf{z}) \leq \psi_\pi(\mathbf{z}) + \rho \alpha_T \alpha_p^{\max} D\psi_\pi(\mathbf{z}, \Delta\mathbf{z}), \tag{12}$$

with $\mathbf{z} = (\mathbf{x}, \mathbf{s})$ the primal variables. Here, $\rho \in (0,1)$, and $D\psi_\pi(\mathbf{z}, \Delta\mathbf{z})$ is the directional derivative of the merit function with respect to the direction $\Delta\mathbf{z}$. After the backtracking, $\alpha_p = \alpha_T \alpha_p^{\max}$ and $\alpha_d = \alpha_T \alpha_d^{\max}$ are defined.

A popular choice of merit function is the l_1-penalty function (see e.g. [53], [74], and [24])

$$\psi_\pi(\mathbf{x}) = f(\mathbf{x}) + \mu\phi(\mathbf{x}, \mathbf{s}) + \pi(\sum_{i=1}^{m} |g_i(\mathbf{x}) + s_i|).$$

The penalty parameter $\pi > 0$ is updated every iteration. There are several approaches to update this parameter so that the convergence is accelerated, see for instance [21]. Nevertheless, the penalty parameter is updated following a simple rule described in [53] as

$$\pi = ||\boldsymbol{\lambda}||_\infty.$$

This rule was successfully used for minimum compliance problems in [56]. A major drawback of this merit function is the lack of differentiability and thus, the potentially Maratos effect [49]. This can be avoided either with a watchdog strategy or with a second-order correction step, see for instance [53], [74], and [31]. However, since the problem (P_N^c) has only linear constraints, the Maratos effect cannot occur [20].

The adaptive interior point scheme is more flexible than the monotone version in regards to the reduction of the merit function. The algorithm allows the merit function to increase some iterations, i.e. the Armijo condition is not imposed. The method forces a reduction in the optimality conditions of the sub-problem within the $l_{\max}$ previous iterations [52]. Before an iterate is accepted, the interior point method verifies the following condition

$$\varphi_{k+1} \leq \tau_a M_k, \tag{13}$$

with φ_{k+1} the Euclidean norm of the KKT conditions of the sub-problem, and

$$M_k = \max\{\varphi_{k-l_{\max}}, \ldots, \varphi_k\}.$$

In addition, the algorithm switches from monotone to adaptive strategy when this condition is satisfied for a given iterate. Finally, the new primal and dual iterates are given by

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{z}_k + \alpha_p \Delta\mathbf{z}, \\ \boldsymbol{\chi}_{k+1} &= \boldsymbol{\chi}_k + \alpha_d \Delta\boldsymbol{\chi}. \end{aligned} \tag{14}$$

3.4 Updating the barrier parameter

The sequence of the barrier parameter values $\{\mu_k\}$ must converge to zero through the optimization process. The Fiacco and McCormick strategy [30], or monotone strategy, consists of fixing the barrier parameter to one value until the sub-problem converges for a given tolerance ϵ_{μ_k} (see Section 3.5). Then, the barrier parameter is decreased. It provides global convergence. On the other hand, the adaptive strategy consists of updating the barrier parameter at each iteration. The adaptive barrier parameter update scheme is based on [70] and described by

$$\mu_{k+1} = \sigma_k \frac{\tilde{\mathbf{z}}_k^T \boldsymbol{\chi}_k}{n_z}. \tag{15}$$

Here, n_z is the number of element in $\tilde{\mathbf{z}}$ ($n_z = 2n + m$), and σ_k is

$$\sigma_k = 0.1 \min\left(0.05\frac{1-\beta}{\beta}, 2\right)^3,$$

with

$$\beta = \frac{\min_i((\tilde{\mathbf{z}}_k)_i(\boldsymbol{\chi}_k)_i)}{\tilde{\mathbf{z}}_k^T \boldsymbol{\chi}_k / n_z}.$$

State-of-the-art software such as IPOPT and LOQO, already use this adaptive updating scheme. Finally, the barrier update strategy in the monotone scheme is inspired by [72]. The barrier parameter is decreased based on the previous value of μ_i and some constants, $\gamma \in (0,1)$ and $\delta \in (1,2)$,

$$\mu_{i+1} = \min(\gamma\mu_i, \mu_i^\delta). \tag{16}$$

Here, the index i refers to the outer loop of the monotone version (see Algorithm 1). Before the monotone strategy starts, the initial barrier parameter is estimated as in [52], i.e.,

$$\mu_0 = 0.8 \frac{\tilde{\mathbf{z}}_k^T \boldsymbol{\chi}_k}{n_z}. \tag{17}$$

3.5 Stopping criteria

In practice, the KKT conditions of the sub-problem converge to the original problem when μ is close to zero. Thus, the algorithm stops and reports to have found a KKT point, when the following inequalities are satisfied,

$$\begin{aligned}
&||\nabla f(\mathbf{x}) + \boldsymbol{\xi} - \boldsymbol{\eta} + \mathbf{J}(\mathbf{x})^T \boldsymbol{\lambda}||_2 \leq \epsilon_1, \\
&||\mathbf{g}(\mathbf{x}) + \mathbf{s}||_2 \leq \epsilon_2, \\
&\left|\left|\begin{bmatrix} \mathbf{D}(\mathbf{s})\boldsymbol{\lambda} - \mu\mathbf{e} \\ \mathbf{U}(\mathbf{x})\boldsymbol{\xi} - \mu\mathbf{e} \\ \mathbf{L}(\mathbf{x})\boldsymbol{\eta} - \mu\mathbf{e} \end{bmatrix}\right|\right|_2 \leq \epsilon_3.
\end{aligned} \tag{18}$$

The stationarity, feasibility, and complementarity tolerances are denoted with ϵ_1, ϵ_2, and ϵ_3, respectively. In the monotone version, the sub-problem converges when (18) is satisfied for $\epsilon_{\mu_1} = \min(\epsilon_1, \kappa\mu_i)$, $\epsilon_{\mu_2} = \min(\epsilon_2, \kappa\mu_i)$, and $\epsilon_{\mu_3} = \min(\epsilon_3, \kappa\mu_i)$ ([72]).

4 An iterative approach for solving saddle-point problems

The aim of this article is to develop efficient methods for solving the large-scale KKT systems arising in interior point methods for the minimum compliance problem (P_N^c). The proposed iterative methods are based on a combination of different existing techniques.

The saddle-point system (10) is solved with a Krylov sub-space method. In particular, flexible GMRES (FGMRES) [60] is chosen since is able to handle indefinite systems and it has good robustness properties. FGMRES is specially accurate and robust when iterative and inaccurate preconditioners are used. It is well-known that the convergence rate of this type of solvers improves with the use of preconditioners [42]. Therefore, an incomplete block triangulation preconditioner inspired by the preconditioner matrix in the right-transforming iteration is defined, see e.g. [63] and [44].

Remark 1. The right-transforming iteration is a Richardson iteration-type method. It can be used both, as smoother in multigrid methods for saddle-point problems (see for instance [63], [78], and [77]), and as an iterate method by itself [40], [39], and [38]. In these articles, structural optimization problems are defined in the SAND formulation. However, preliminary studies suggest that FGMRES is more robust and stable for the minimum compliance problem (P_N^c).

4.1 Block triangular preconditioner description

The proposed block triangular preconditioner requires the solution of smaller linear systems. In the same way, these equations are efficiently solved with FGMRES. Either block-diagonal matrices or multigrid cycles are used as preconditioners for those systems. In the following, the proposed preconditioner is explained in detail.

Let consider a general saddle-point problem

$$\mathbf{W}\boldsymbol{\Delta} = \mathbf{b}, \tag{19}$$

with

$$\mathbf{W} = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\mathbf{C} \end{bmatrix},$$

and matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{C} \in \mathbb{R}^{m\times m}$, and $\mathbf{B} \in \mathbb{R}^{m\times n}$. These problems arise in e.g. nonlinear constrained optimization solvers. The KKT system (10) in the interior point method is a particular case of saddle-point problem, with $\mathbf{A} = \mathbf{H} + \mathbf{D}$, $\mathbf{H}$ being the

Hessian of the Lagrangian function and $\mathbf{D}$ a diagonal matrix. In our application, $\mathbf{C} = \mathbf{0}$, and $\mathbf{B}$ is the Jacobian of the constraints.

Theorems 1, 2, and 3 give some conditions under which existence of solution of the system (19) can be proved.

Theorem 1. *(from [13]) Assuming $\boldsymbol{A}$ nonsingular, $\boldsymbol{W}$ is nonsingular if and only if $\boldsymbol{S} = -(\boldsymbol{C} + \boldsymbol{B}\boldsymbol{A}^{-1}\boldsymbol{B})$ is.*

Theorem 2. *(from [13]) Assume $\boldsymbol{A}$ symmetric positive definite and $\boldsymbol{C}$ symmetric positive semi-definite. If $ker\{\boldsymbol{C}\} \cap ker\{\boldsymbol{B}^T\} = \{0\}$ then the saddle-point matrix $\boldsymbol{W}$ is nonsingular. In particular $\boldsymbol{W}$ is invertible if $\boldsymbol{B}$ has full rank.*

Theorem 3. *(from [53]) For simplicity, let assume $\boldsymbol{C} = \boldsymbol{0}$, $\boldsymbol{A} = \boldsymbol{H}$. Let $\boldsymbol{B} = \boldsymbol{J}$ have a full row rank (Jacobian of the constraints), and assume that the reduced-Hessian matrix $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}$ is positive definite. Then the KKT matrix $\boldsymbol{W}$ is nonsingular, and hence there is a unique vector satisfying the linear system (19).*

The matrix $\mathbf{W}$ is partitioned in two nonsingular matrices, $\mathbf{M}$ and $\mathbf{N}$, using right-transformation, $\mathbf{W}^R$, and left-transformation, $\mathbf{W}^L$, matrices. Here, $\mathbf{M}$ is relatively easy to invert and $\mathbf{N} \simeq \mathbf{0}$.

$$\mathbf{W}^L\mathbf{W}\mathbf{W}^R = \mathbf{M} - \mathbf{N}.$$

The right-transformation iteration is defined when $\mathbf{W}^L = \mathbf{I}$ [44]. Since a good preconditioner should approximate the matrix as much as possible, $\mathbf{P}_\mathbf{W}$ is defined as follows

$$\begin{aligned}
&\mathbf{W} = \mathbf{W}\mathbf{W}^R\mathbf{W}^{-R} \simeq \mathbf{M}\mathbf{W}^{-R} \\
&\mathbf{P}_\mathbf{W} \simeq \mathbf{W} \\
&\mathbf{P}_\mathbf{W}^{-1} \equiv (\mathbf{M}\mathbf{W}^{-R})^{-1} = \mathbf{W}^R\mathbf{M}^{-1}.
\end{aligned}$$

In particular, the right-transformation matrix is

$$\mathbf{W}^R = \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{A}}^{-1}\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Here, $\tilde{\mathbf{A}}$ is assumed to be a good approximation of $\mathbf{A}$. Then, from a regular splitting

$$\mathbf{M} = \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{B} & \tilde{\mathbf{S}} \end{bmatrix},$$

with $\tilde{\mathbf{S}} = -(\mathbf{C} + \mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T)$ an approximation of the Schur complement (see [63] for more details). Based on block triangular matrix theory, the inverse of $\mathbf{M}$ is

$$\mathbf{M}^{-1} = \begin{bmatrix} \tilde{\mathbf{A}}^{-1} & \mathbf{0} \\ -\tilde{\mathbf{S}}^{-1}\mathbf{B}\tilde{\mathbf{A}}^{-1} & \tilde{\mathbf{S}}^{-1} \end{bmatrix}. \tag{20}$$

Then, the preconditioner matrix is

$$\mathbf{P}_{\mathbf{W}}^{-1} = \mathbf{W}^R\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{A}}^{-1}\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{A}}^{-1} & \mathbf{0} \\ -\tilde{\mathbf{S}}^{-1}\mathbf{B}\tilde{\mathbf{A}}^{-1} & \tilde{\mathbf{S}}^{-1} \end{bmatrix}$$
$$= \begin{bmatrix} \tilde{\mathbf{A}}^{-1} + \tilde{\mathbf{A}}^{-1}\mathbf{B}^T\tilde{\mathbf{S}}^{-1}\mathbf{B}\tilde{\mathbf{A}}^{-1} & -\tilde{\mathbf{A}}^{-1}\mathbf{B}^T\tilde{\mathbf{S}}^{-1} \\ -\tilde{\mathbf{S}}^{-1}\mathbf{B}\tilde{\mathbf{A}}^{-1} & \tilde{\mathbf{S}}^{-1} \end{bmatrix}.$$

For the rest of the article, the term $\tilde{\mathbf{b}}$ refers to a general known vector. Here, $\tilde{\mathbf{b}}_1 \in \mathbb{R}^n$ and $\tilde{\mathbf{b}}_2 \in \mathbb{R}^m$ are defined such that

$$\tilde{\mathbf{b}} = \begin{bmatrix} \tilde{\mathbf{b}}_1 \\ \tilde{\mathbf{b}}_2 \end{bmatrix}.$$

To reduce the computational cost of the preconditioner operation, a matrix-vector multiplication is defined as

$$\mathbf{P}_{\mathbf{W}}^{-1}\tilde{\mathbf{b}} = \begin{bmatrix} \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}_1 + \tilde{\mathbf{A}}^{-1}(\mathbf{B}^T\tilde{\mathbf{S}}^{-1}(\mathbf{B}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}_1 - \tilde{\mathbf{b}}_2)) \\ -\tilde{\mathbf{S}}^{-1}(\mathbf{B}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}_1 - \tilde{\mathbf{b}}_2) \end{bmatrix}. \tag{21}$$

The preconditioner operator (21) requires the solution of three linear systems where $\tilde{\mathbf{A}}$ or $\tilde{\mathbf{S}}$ appear.

Remark 2. $\mathbf{P}_{\mathbf{W}}$ is chosen as preconditioner matrix instead of classical block diagonal preconditioners since, in preliminary numerical studies, it showed better robustness properties.

Figure 1 illustrates the procedure for solving the saddle-point system (19) with the proposed iterative method. FGMRES needs two operations; a matrix-vector multiplication ($\mathbf{W}\tilde{\mathbf{b}}$) and the preconditioner operator ($\mathbf{P}_{\mathbf{W}}^{-1}\tilde{\mathbf{b}}$). These functions are represented as branches in Figure 1. The preconditioner, in turn, is detailed in five steps.

4.2 Using a multigrid cycle as preconditioner of Krylov sub-space methods

The solution of $\mathbf{P}_{\mathbf{W}}\mathbf{x} = \tilde{\mathbf{b}}$ needs the solution of systems with the form $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$ and $\tilde{\mathbf{S}}\mathbf{x} = \tilde{\mathbf{b}}$. Direct methods can be used, although for large enough systems, iterative methods, such as Krylov sub-space methods, are recommended. In these solvers, the choice of a good preconditioner that does not increase the computational effort is essential. In this article, a multigrid cycle (MC) is defined as preconditioner of Krylov sub-space methods in positive definite linear systems due to its great properties [4], [36], and [76].

Multigrid methods solve the problem on a coarse mesh. Then, the solution is interpolated from coarse to fine meshes until the original discretization is achieved.

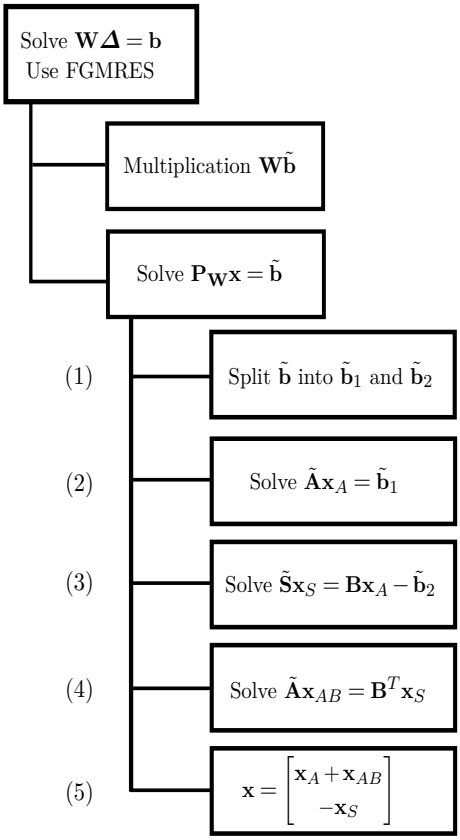


Figure 1: Iterative scheme for solving a saddle-point system with the form (19).

The good performance of the multigrid cycle is due to the use of smoother functions to remove the high frequency of the errors, and the coarse-grid correction to transfer the residuals from coarse to fine and from fine to coarse mesh [18].

The computational effort (memory and time) is significantly reduced and, unlike other iterative methods, the convergence rate does not depend on the condition number of the matrix [4]. Typically, the condition of the matrix increases with the discretization of the mesh, thus, it is considered a very powerful technique due to its scalability[3] properties. For more details of multigrid techniques, the text books [76] and [35] are recommended.

Algorithm 2 outlines the general scheme of a multigrid cycle (MC) for solving a linear system with the form $\mathbf{W}\mathbf{x} = \mathbf{b}$. Matrices $\mathbf{R}$ and $\mathbf{P}$ correspond to the restriction and prolongation operators to move from the fine to the coarse mesh, and from the coarse to the fine mesh, respectively. Depending on the number of m_c cycles the multigrid method computes V-cycles, W-cycles, or other different variations [65] and [76]. The performance of the method is affected by the number of pre-smoothers (ν_1), the type of cycle (m_c), the number of post-smoothers (ν_2), but most importantly, the smoothing

[3]A solver is *numerically scalable* if the complexity of the method grows asymptotically linearly with the size of the problem [29].

Algorithm 2 Multigrid cycle [18].

Input: $\mathbf{x}$ = MC($\mathbf{W}$,$\mathbf{b}$,$\mathbf{x}$, `level`, ν_1, ν_2, m_c)

1: **if** `level` = `coarsest-level` **then**
2: Solve the problem: $\mathbf{x} = \mathbf{W}^{-1}\mathbf{b}$.
3: `return`
4: **end if**
5: Pre-smoothing step: $\mathbf{x}$ = S($\mathbf{W}$,$\mathbf{b}$,$\mathbf{x}$, ν_1).
6: Grid correction: $\mathbf{r} = \mathbf{W}\mathbf{x} - \mathbf{b}$.
7: Restriction step: $\mathbf{r} = \mathbf{R}\mathbf{r}$.
8: $\mathbf{W} = \mathbf{R}\mathbf{W}\mathbf{R}^T$.
9: Initialize $mc = 1$, $\mathbf{x}_{mc} = \mathbf{0}$.
10: **repeat**
11: $\mathbf{x}_{mc}$ = MC($\mathbf{W}$,$\mathbf{r}$, $\mathbf{x}_{mc}$, `level-1`,ν_1, ν_2, m_c).
12: $mc = mc + 1$.
13: **until** $mc = m_c$
14: Prolongation step: $\mathbf{x} = \mathbf{x} - \mathbf{P}\mathbf{x}_{mc}$.
15: Post-smoothing step: $\mathbf{x}$ = S($\mathbf{W}$,$\mathbf{b}$,$\mathbf{x}$, ν_2).
16: **return**

method selected. This is the computationally most expensive step in the multigrid cycle. Moreover, it is the most expensive step in terms of time and memory of the proposed iterative method. For positive definite linear systems, methods such as Jacobi and Gauss-Seidel are typically used as smoothers. In preliminary studies, better performance and robustness were achieved with Gauss-Seidel than with damped Jacobi (suggested in [4]). Multigrid methods can be also applied to indefinite systems, although the selection of a smoother function is more complicated and problem dependent [13].

5 On the solution of a KKT system in structural topology optimization

Throughout this section, different techniques are introduced to reproduce each step of the iterative method presented in Section 4 (Figure 1) for the specific minimum compliance problem. In particular, two different iterative approaches are presented. They differ in the choice of the stiffness matrix preconditioner (see Section 5.1). The iterative method implemented in this article solves, at each interior point iteration, the KKT system described as

$$\begin{bmatrix} 2\mathbf{F}(\mathbf{t})^T\mathbf{K}^{-1}(\mathbf{t}) & \mathbf{F}(\mathbf{t}) + \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l & \mathbf{0} & \mathbf{v} \\ \mathbf{0} & & \Sigma_s & 1 \\ \mathbf{v}^T & & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{t} \\ \Delta s \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{t}) + \lambda\mathbf{v} + \mu\mathbf{U}(\mathbf{t})^{-1}\mathbf{e} - \mu\mathbf{L}(\mathbf{t})^{-1}\mathbf{e} \\ \lambda - \frac{\mu}{s} \\ \mathbf{v}^T\mathbf{t} - V + s \end{bmatrix}. \tag{22}$$

Here, the approximate positive semi-definite Hessian of the compliance (2) is used. With this $\hat{\mathbf{H}}(\mathbf{t})$, the existence of solution of (22) is ensured. Theorem 3 is satisfied for $\mathbf{C} = 0$, $\mathbf{B} = [\mathbf{v}^T 1]$ and

$$\mathbf{H} = \begin{bmatrix} \hat{\mathbf{H}}(\mathbf{t}) + \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l & \mathbf{0} \\ \mathbf{0} & \Sigma_s \end{bmatrix}.$$

The reduced-Hessian $\mathbf{Z}^T\mathbf{H}\mathbf{Z}$ is positive definite ($\hat{\mathbf{H}}(\mathbf{t}) \succeq 0$ and $\boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l \succ 0$ and $\Sigma_s \succ 0$) and $\mathbf{B}$ has full row rank, then, the KKT system has a unique solution and is a descent direction of the merit function [56], [53], and [46]. Thus, the inertia[4] of the KKT matrix is correct in all the iterations and there is no need to perturb or modify the system as in, for instance IPOPT [72]. Due to the symmetry of $\hat{\mathbf{H}}(\mathbf{t})$, the system is expanded to get rid of the dense Hessian [56].

$$\begin{bmatrix} \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l & \mathbf{F}(\mathbf{t})^T & \mathbf{0} & \mathbf{v} \\ \mathbf{F}(\mathbf{t}) & -\frac{1}{2}\mathbf{K}(\mathbf{t}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_s & 1 \\ \mathbf{v}^T & \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{t} \\ \Delta\mathbf{w} \\ \Delta s \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{t}) + \lambda\mathbf{v} + \mu\mathbf{U}(\mathbf{t})^{-1}\mathbf{e} - \mu\mathbf{L}(\mathbf{t})^{-1}\mathbf{e} \\ \mathbf{0} \\ \lambda - \frac{\mu}{s} \\ \mathbf{v}^T\mathbf{t} - V + s \end{bmatrix}. \quad (23)$$

Here, $\Delta\mathbf{w}$ is an auxiliary variable. Moreover, the KKT matrix in (23) is modified to simplify the construction of $\mathbf{P}_\mathbf{W}$. The matrix $\mathbf{W}$ is regrouped as follows

$$\mathbf{W} = \begin{bmatrix} -\frac{1}{2}\mathbf{K}(\mathbf{t}) & \mathbf{0} & \mathbf{F}(\mathbf{t}) & \mathbf{0} \\ \mathbf{0} & \Sigma_s & \mathbf{0} & 1 \\ \mathbf{F}(\mathbf{t})^T & \mathbf{0} & \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l & \mathbf{v} \\ \mathbf{0} & 1 & \mathbf{v}^T & 0 \end{bmatrix}. \quad (24)$$

Identification and approximation give the block matrices

$$\tilde{\mathbf{A}} = \begin{bmatrix} -\frac{1}{2}\tilde{\mathbf{K}}(\mathbf{t}) & \mathbf{0} \\ \mathbf{0} & \Sigma_s \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{F}(\mathbf{t}) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix},$$

$$\mathbf{C} = -\begin{bmatrix} \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l & \mathbf{v} \\ \mathbf{v}^T & 0 \end{bmatrix},$$

and the approximate Schur complement

$$\tilde{\mathbf{S}} = -(\mathbf{C} + \mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T) = \begin{bmatrix} \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l + 2\mathbf{F}^T(\mathbf{t})\tilde{\mathbf{K}}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) & \mathbf{v} \\ \mathbf{v}^T & -\Sigma_s^{-1} \end{bmatrix}.$$

[4]For a given matrix $\mathbf{W}$, the trial (i_+, i_-, i_0) is called inertia, where i_0, i_+, and i_- are the number of zeros, positive and negative eigenvalues, respectively.

Remark 3. Based on Theorem 1, the saddle-point problem (23) is well defined and solvable since $\mathbf{A}$ and $\mathbf{S}$ are nonsingular. For this particular matrix (24), $\mathbf{S}$ is a saddle-point problem satisfying Theorem 2;

$$\mathbf{S} = \begin{bmatrix} \mathbf{A_S} & \mathbf{B_S^T} \\ \mathbf{B_S} & -C_\mathbf{S} \end{bmatrix},$$

with $\mathbf{A_S} = \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l + 2\mathbf{F}^T(\mathbf{t})\mathbf{K}^{-1}(\mathbf{t})\mathbf{F}(\mathbf{t}) \succ 0$, $C_\mathbf{S} = \Sigma_s^{-1} \succ 0$, and $\mathbf{B_S} = \mathbf{v}^T$ is linearly independent (full rank). Therefore, $\mathbf{S}$ is nonsingular.

Remark 4. Both $\mathbf{W}^R$ and $\mathbf{M}^{-1}$ are well defined due to the nonsingularity of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{S}}$.

Since $\mathbf{W}$ is regrouped, the vector $\tilde{\mathbf{b}}$ (Figure 1 Step 1) and the output vector $\mathbf{x}$ (Step 5 in Figure 1) need to be reorganized in the same way.

Two linear systems with the form

$$\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}} \in \mathbb{R}^{d+1} \tag{25}$$

are needed for $\mathbf{P_W}$ (Steps 2 and 4 in Figure 1). This system of equations is relatively easy to solve because $\tilde{\mathbf{A}}$ is block diagonal. Nevertheless, it requires the solution of

$$-\frac{1}{2}\tilde{\mathbf{K}}(\mathbf{t})\mathbf{x} = \tilde{\mathbf{b}}_1 \in \mathbb{R}^d. \tag{26}$$

Here, $\tilde{\mathbf{b}}_1$ is another general known vector. In this case, $\tilde{\mathbf{b}}_1$ is part of the $\tilde{\mathbf{b}}$ in (25). In Section 5.1, different alternatives for solving this system of equations are detailed.

On the other hand, the solution of the linear system (Step 3 in Figure 1),

$$\tilde{\mathbf{S}}\mathbf{x} = \tilde{\mathbf{a}} \in \mathbb{R}^{n+1} \tag{27}$$

is more expensive, and the use of an iterative solver (FGMRES) is needed.

Figure 2 shows the scheme to solve (27). Another general known vector is denoted with $\tilde{\mathbf{a}}$ to emphasize that $\tilde{\mathbf{a}} \neq \tilde{\mathbf{b}}$. In the matrix-vector multiplication operation ($\tilde{\mathbf{S}}\tilde{\mathbf{b}}$), a system like (26) is involved. In addition, the following block diagonal preconditioner matrix of $\tilde{\mathbf{S}}$ is defined,

$$\mathbf{P_S} = \begin{bmatrix} \tilde{\mathbf{A}}_\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \tilde{S}_\mathbf{S} \end{bmatrix}.$$

With

$$\tilde{\mathbf{A}}_\mathbf{S} = \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_l + 2\mathbf{F}^T(\mathbf{t})\hat{\mathbf{K}}(\mathbf{t})\mathbf{F}(\mathbf{t}) \succ 0 \in \mathbb{R}^n,$$

and

$$\tilde{S}_\mathbf{S} = -\Sigma_s^{-1} - \mathbf{v}^T\tilde{\mathbf{A}}_\mathbf{S}^{-1}\mathbf{v} \in \mathbb{R}.$$

Here, the preconditioner of the stiffness matrix is just the inverse of its diagonal terms,

$$\hat{\mathbf{K}}(\mathbf{t}) = \operatorname{diag}\left(\frac{1}{\mathbf{K}_{i,i}(\mathbf{t})}\right)_{i=1,\dots,d}.$$

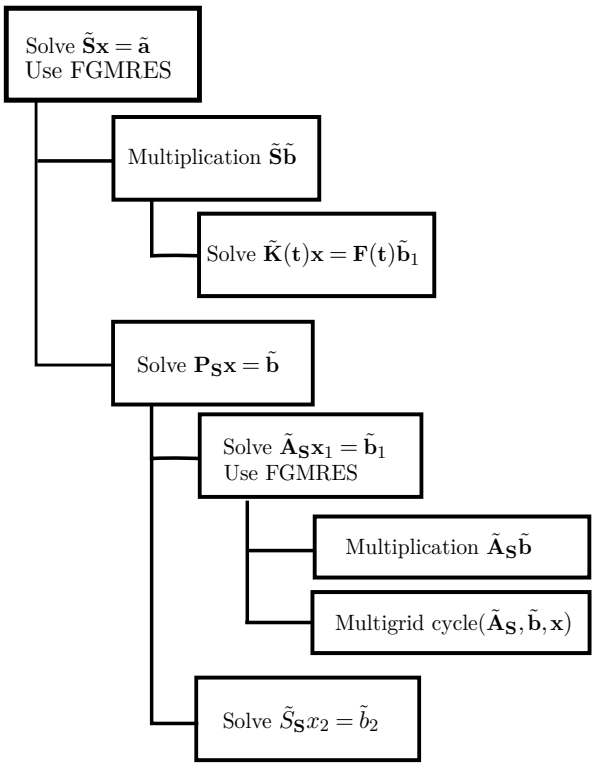


Figure 2: Iterative scheme for solving the indefinite linear system (27).

This fast and cheap preconditioner of the stiffness matrix is used since $\hat{\mathbf{K}}(\mathbf{t})$ is just part of $\mathbf{P_S}$. More advanced stiffness matrix preconditioners are detailed in Section 5.1. The preconditioner operation of $\tilde{\mathbf{S}}$ consists of solving

$$\mathbf{P_S}\mathbf{x} = \tilde{\mathbf{b}},$$

such that

$$\tilde{\mathbf{A}}_\mathbf{S}\mathbf{x}_1 = \tilde{\mathbf{b}}_1, \tag{28}$$

$$\tilde{S}_\mathbf{S} x_2 = \tilde{b}_2, \tag{29}$$

with

$$\tilde{\mathbf{b}} = \begin{bmatrix} \tilde{\mathbf{b}}_1 \\ \tilde{b}_2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ x_2 \end{bmatrix}.$$

The system of equations (28) is still expensive (the dimension of the matrix is n). The computational cost of the iterative solver will be highly dependent on the efficiency of the method used to solve it[5]. Since the matrix $\tilde{\mathbf{A}}_\mathbf{S}$ is positive definite, a multigrid cycle (MC) is chosen as preconditioner of FGMRES.

[5]This system is solved one time per FGMRES preconditioner iteration for solving (27). At the same time, (27) is solved one per FGMRES preconditioner iteration for (23).

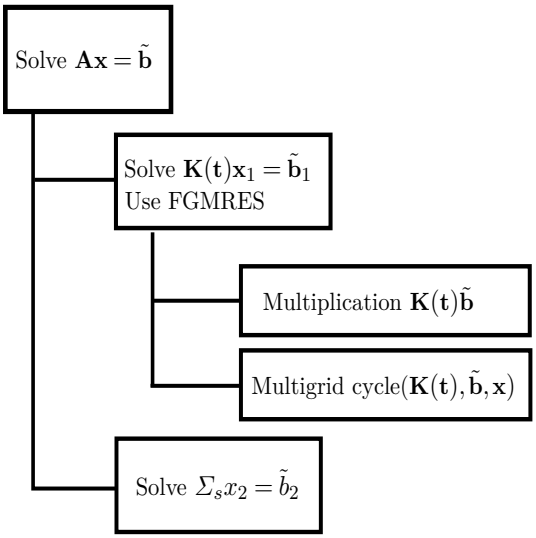


Figure 3: Iterative scheme for solving the positive definite system (25) using the first approach.

5.1 Good preconditioners for the stiffness matrix

The iterative method needs, at two different steps, the solution of the linear system (26) in which the stiffness matrix is involved. Additionally, the equilibrium equations are solved every optimization iteration. Plenty of literature can be found regarding iterative methods for the equilibrium equations (1), see e.g. [75], [5], [4], and [1] among others. Based on [4], the equilibrium equations are solved with a Krylov sub-space method and a multigrid cycle as preconditioner. However, FGMRES is chosen since, in preliminary studies, it seems more robust and requires less iterations than PCG.

The same technique can be used for (26) to obtain the solution of $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$ and $\tilde{\mathbf{S}}\tilde{\mathbf{b}}$. In this case, the exact matrix $\mathbf{K(t)}$ is used (and thus, $\mathbf{A}$ and $\mathbf{S}$). For this approach, the solution of (25) (Steps 2 and 4 in Figure 1) is illustrated in Figure 3. Another alternative preconditioner consists of using the Separate Displacement Component (SDC) as in [34]. The stiffness matrix $\mathbf{K(t)}$ is reorganized and approximated using block diagonal sub-matrices as

$$\tilde{\mathbf{K}}(\mathbf{t}) = \begin{bmatrix} \mathbf{K}_{xx}(\mathbf{t}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yy}(\mathbf{t}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{zz}(\mathbf{t}) \end{bmatrix}. \tag{30}$$

Here, $\mathbf{K}_{xx}(\mathbf{t})$, $\mathbf{K}_{yy}(\mathbf{t})$, and $\mathbf{K}_{zz}(\mathbf{t})$ are sub-matrices of the degrees of freedom components in the x-direction, y-direction, and z-direction, respectively.

Remark 5. The approximation (30) of $\mathbf{K(t)} \sim \tilde{\mathbf{K}}(\mathbf{t})$ is well-defined and it remains positive definite, see [34].

Remark 6. Preliminary numerical experiments have proven that preconditioners such as the diagonal matrix and the incomplete Cholesky factorization for (26), are not good

enough to produce a robust and convergent iterative method.

In this second approach (SDC), $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$ and $\tilde{\mathbf{S}}\tilde{\mathbf{b}}$ are solved exactly (direct solvers) using a simplified approximation of the stiffness matrix, $\tilde{\mathbf{K}}(\mathbf{t})$.

These two possible preconditioners are tested and compared in the numerical experiments. In the following the *first approach* denotes the use of FGMRES+MC to solve (26). The *second approach* refers to the use of SDC.

Both preconditioners have advantages and disadvantages. In the first case (FGMRES+MC), the preconditioner of the KKT matrix, $\mathbf{P}_{\mathbf{W}}^{-1}$, is exactly the inverse of $\mathbf{W}$. However, (21) is approximately obtained (concatenation of several iterative methods with different tolerances). Errors arising from the iterative methods may produce a failure. Nevertheless, in practice, this approach is robust and efficient (see Section 7). On the other hand, the second preconditioner has all the mathematical properties to ensure convergence (see [44] and [34]) but the computational effort and memory storage are higher. The stiffness matrix approximation (30) requires three Cholesky factorizations, the storage of several matrices, and the solution of three small systems.

6 Implementation

Section 7 shows the numerical experiments on examples where the 3D design domain is discretized using brick elements. Each rectangular solid element contains 8 nodes (24 degrees of freedom) and the scaled element stiffness matrix is assumed to be the same for all elements in the examples. The code is written in MATLAB release 2014a [69].

Regarding the pre-processing required for the multigrid cycle, two different restriction matrices, $\mathbf{R_t}$ and $\mathbf{R_u}$, are built before the optimization process takes place. The restriction matrix $\mathbf{R_t}$ refers to the full weight average of the elements, while $\mathbf{R_u}$ refers to the nodes, i.e.,

$$(\mathbf{R_t})_h^H : \mathbb{R}^{n_h} \longrightarrow \mathbb{R}^{n_h \times n_H},$$

$$(\mathbf{R_u})_h^H : \mathbb{R}^{d_h} \longrightarrow \mathbb{R}^{d_h \times d_H}.$$

Here, h and H refer to the mesh size of the fine and the coarse mesh, respectively. Similarly, n_h and n_H refer to the number of elements and d_h and d_H are the degrees of freedom at each mesh. The restriction matrix $\mathbf{R_t}$ is used in the multigrid cycle in (28) and $\mathbf{R_u}$ is applied in (1) and in (26) (first approach). The prolongation matrix is just the transpose of the restriction matrix [4] and [76].

In particular, the restriction matrix for the density variables collects the parent-children relationship. Given a coarse mesh with n_H elements, the next level contains $n_h = 2^3 n_H$ elements. Thus, given an element i in a coarse level, $\mathbf{R_t}$ contains the value of $1/8$ in those positions corresponding with its children, i.e., the elements from the

fine mesh that are "inside" the coarse element. The restriction matrix for the nodes is built based on the shape functions. Each node of the fine mesh gives the corresponding contribution to the nodes of the coarse mesh. For more details see [36].

Step 8 of Algorithm 2 is, in practice, omitted to save time. In addition, since MC is chosen as preconditioner at several steps of the proposed iterative method, all the matrices needed in Gauss-Seidel are stored to reduce the computational time. Although it will lead to an increment in memory usage, it is preferable to focus on reducing the computational time as much as possible. Every time a new stiffness matrix is assembled, Galerkin's method is used to store the stiffness matrices of each level,

$$\mathbf{K}(\mathbf{t})_H = (\mathbf{R_u})_h^H \mathbf{K}(\mathbf{t})_h ((\mathbf{R_u})_h^H)^T.$$

Moreover, the upper triangular (with the diagonal) and the lower triangular matrices of each $\mathbf{K}(\mathbf{t})_H$ are stored (using `triu` and `tril` MATLAB functions). Similarly, before the iterative method starts, $\tilde{\mathbf{A}}_\mathbf{S}$, the upper triangular (and diagonal), and the lower triangular matrices for every level are stored,

$$(\tilde{\mathbf{A}}_\mathbf{S})_H = (\mathbf{R_t})_h^H (\tilde{\mathbf{A}}_\mathbf{S})_h ((\mathbf{R_t})_h^H)^T.$$

If the second approach (SDC) is used, the block matrices $\mathbf{K}_{xx}$, $\mathbf{K}_{yy}$, and $\mathbf{K}_{zz}$ are needed. More precisely, their Cholesky factorizations and the transposes of them are stored to reduce the computational time of the direct solver. The MATLAB function `symamd` [3] is used to permute the order of the matrices and create sparser Cholesky factorizations (using `chol` [22]).

Tables 1, 2, and 3 contain the name, description, and value of every parameter involved in the interior point method for topology optimization problems. Throughout the rest of the article, due to brevity reasons, the optimization method is cited as TopIP. The parameters needed in the interior point method detailed in Algorithm 1 are collected in Table 1.

Preliminary studies suggest that multigrid cycles perform very accurately with a W-type cycle. In TopIP, depending on the linear system solved, FGMRES+MC has different parameters. It is important to obtain an accurate solution in the equilibrium equations (1). Thus, a W-type with $\nu_1 = \nu_2 = 2$ is set for the multigrid. The stopping criteria for FGMRES are a relative residual error equal to $\omega_e = 10^{-8}$ and a maximum number of iterations equal to 200. In contrast, (28) and (26) (first approach) are part of the preconditioner, and then, it is possible to be less strict. A simple V-cycle, with $\nu_1 = \nu_2 = 1$ and $\omega_s = 10^{-2}$ is used. The number of smoother iterations is chosen so that there is a balance between accuracy of solutions and computational effort. The multigrid cycle is built with 4 levels of discretizations.

Table 1: Parameter setting for the interior point method. The table contains the name of the parameter, a brief description, and the value.

Parameter	**Description**	**Value**
$\mathbf{t}_0$	Starting point	$0.5\mathbf{e}$
s_0	Starting slack variable	$\max(V - \mathbf{v}^T\mathbf{t}_0, 10^{-5})$
ϵ_1	Stationarity tolerance	10^{-6}
ϵ_2	Feasibility tolerance	10^{-8}
ϵ_3	Complementarity tolerance	10^{-6}
κ	Factor parameter for convergence of sub-problem (monotone version)	1
`max iter`	Maximum number of interior point iterations	1,000
τ	Fraction to the boundary factor [74]	0.995
ρ	Backtracking factor (monotone version) [74]	10^{-8}
θ	Backtracking reduction factor (monotone version) [74]	0.5
γ	Factor parameter to update μ (monotone version) [72]	0.2
δ	Power parameter to update μ (monotone version) [72]	1.5
τ_a	Factor parameter to accept the iterate (adaptive version) [52]	0.9999
$l_{\max}$	Number of previous iterations to take into account to accept iterate (adaptive version)	15
$\mu_{\min}$	Minimum barrier parameter value	10^{-7}

Remark 7. The Krylov sub-space methods presented in this article stop when the Euclidean norm of the relative residual error is lower than a tolerance ω, i.e.

$$||\mathbf{r}||_2 = \frac{||\mathbf{b} - \mathbf{W}\boldsymbol{\Delta}||_2}{||\mathbf{b}||_2} \leq \omega.$$

The principal (top level) FGMRES method, which solves the KKT system (23), and the intermediates (second and third level) FGMRES to solve (27), (28), and (26) (first approach), are differentiated in regards to the maximum number of iterations. For the former, it is set to 300, for the second level it is set to 100 and finally, for the last level it is set to 30.

Nonlinear optimization methods require some practical implementation details to improve their efficiency. In particular, the scaling of the problem will significantly affect the performance of TopIP. Therefore, the Young's modulus are set to $E_v = 1$ and $E_1 = 10^4$. Moreover, the objective function is scaled as

$$\tilde{f}(\mathbf{t}) = \frac{f(\mathbf{t})}{||\nabla f(\mathbf{t}_0)||_2}.$$

Table 2: Parameter setting for the iterative method. The table contains the name of the parameter, a brief description, and the value.

Parameter	Description	Value
ω	Norm of the relative residual error in FGMRES for the top level	10^{-6}
ω_e	Norm of the relative residual error in FGMRES for the equilibrium equations	10^{-8}
ω_s	Norm of the relative residual error in FGMRES for the second and third level	10^{-2}
`max iterative iter`	Maximum number of iterations of FGMRES in the top level	300
`max gmres1 iter`	Maximum number of iterations of FGMRES in the second level	100
`max gmres2 iter`	Maximum number of iterations of FGMRES in the third level	30
`max eq iter`	Maximum number of iterations of FGMRES for the equilibrium equations	200
`restart step`	Number of FGMRES restart iterations	20
m_c	Multigrid cycle type for the equilibrium equations	2 (W-cycle)
m_c	Multigrid cycle type for the iterative method	1 (V-cycle)
ν_1, ν_2	Number of pre-smoothing/post smoothing iterations in MC for the equilibrium equations	2
ν_1, ν_2	Number of pre-smoothing/post-smoothing iterations in MC for the iterative method	1
`element coarse`	Number of elements (per unit of length) in the coarsest mesh	2
`level`	Number of levels in MC	4

Table 3: Values of characteristic parameters of topology optimization problems.

Parameter	Description	Value
E_v	Young's modulus value for "void" material (TopIP)	10
E_1	Young's modulus value for solid material (TopIP)	10^4
E_v	Young's modulus value for "void" material (GCMMA)	10^{-2}
E_1	Young's modulus value for solid material (GCMMA)	10
p	SIMP penalization parameter	3
$r_{\min}$	Radius for the density filter (L_x is the length in the x direction od the design domain)	$0.04L_x$
υ	Poisson's ratio	0.3
v_i	Relative volume of element	$\frac{1}{\sqrt{n}}$

Finally, the relative volume of the elements is set to $v_i = \frac{1}{\sqrt{n}}$. In the numerical experiments, the performance of TopIP is compared to GCMMA. The settings of the parameters for GCMMA are slightly different to appropiately scale the problem (see Table 3 and [57]). In particular, $E_v = 10^{-2}$, $E_1 = 10$, and the optimality tolerance is $\omega = 10^{-4}$ (first-order method). The starting point is initialized with a homogeneous design, $\mathbf{t}_0 = 0.5\mathbf{e}$ (middle value) for TopIP and $\mathbf{t}_0 = V\mathbf{e}$ for GCMMA.

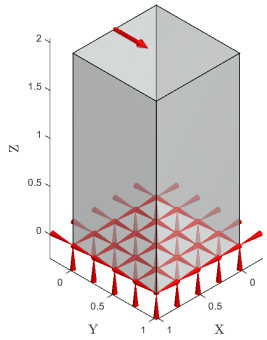
7 Numerical experiments

All the computations were done on Intel Xeon E5-2680v2 ten-core CPUs, running at 2.8GHz with 128 GB RAM.

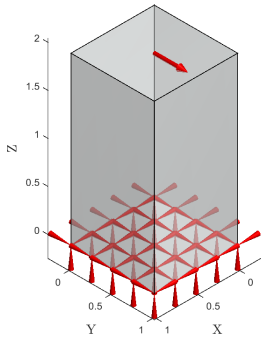
Figure 4 contains the six different boundary conditions and external load definitions used for the numerical experiments. The domain is either fully clamped on one side (D1, D2, and D3) or clamped at the corners (D4, D5, and D6). Regarding the loads, three different possibilities are collected. On the free face, the force is defined either as a point load in the middle of an edge (D1 and D4), a point load on the center (D2 and D5), or a distributed load on an edge (D3 and D6). Some of these design domains are optimized in, for instance, [4], [5], [75], and [45]. The length ratios will vary as well as the mesh size and the volume fraction.

A test set of 13 large-scale 3D problems is gathered for the numerical experiments. Table 4 collects the description of the optimized designs, such as the length ratio, the volume fraction, the type of domain, the discretization of the mesh, the number of elements (n), and the number of degrees of freedom (d). It also gathers the objective function value obtained by TopIP ($f(\mathbf{t})$), the number of optimization (outer) iterations required to find a KKT point (Iter), and the computational time consumed by the solver. For convenience, the problems are denoted with a number (N) from 1 to 13. The final designs and the performance (except computational time) of TopIP are independent of the preconditioner chosen for the stiffness matrix in the iterative method. Thus, the table shows the results using the first approach (FGMRES+MC). The computational time for the second approach (SDC) is presented in parenthesis. The biggest problem in this test set contains more than one million elements (more than three million degrees of freedom). Larger problems are not included due to time restrictions on the cluster.

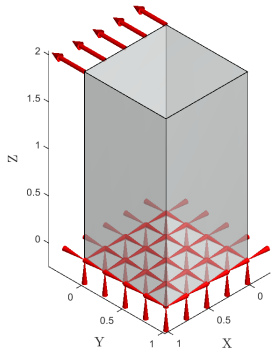
During the numerical experiments, the execution of some instances is not finished due to problems of memory or time limitation. For the former, the time is marked with a dash. When the problems are not finished due to the time limitation, the time is set to infinite. The remaining of the problems in the test set are run until the Euclidean norm of the KKT conditions is lower or equal to 10^{-6} or until the maximum number of iterations is reached (`max iter` $= 1{,}000$).



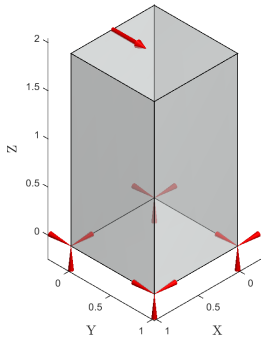
(a) Type D1



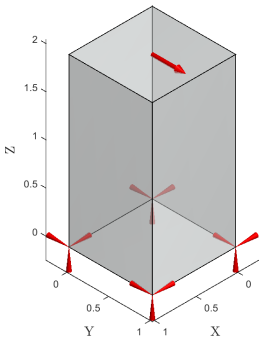
(b) Type D2



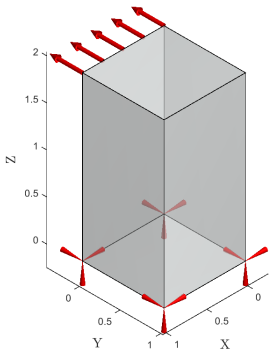
(c) Type D3



(d) Type D4



(e) Type D5



(f) Type D6

Figure 4: Collection of six 3D design domains with different boundary conditions and external loads.

Table 4: Results of the optimized design obtained with TopIP for 13 3D minimum compliance problems. The table contains the number of the problem (N), its description (length ratio, volume fraction, and domain type), the mesh discretization, the number of elements, and the number of degrees of freedom. In addition, the objective function value, the number of optimization iterations, and the computational time (first approach) obtained with TopIP are collected. The computational time for the second approach is included in parenthesis. In some problems, the time is marked with a dash (-) or with infinite (Inf) to indicate two possible types of error.

N	**Description**	**Mesh**	**n**	**d**	$f(\mathbf{t})$	**Iter**	**Time [hh:mm:ss]**
1	$1\times1\times2$, $v=0.4$, $D1$	$16\times16\times32$	8192	28611	$1.296e+02$	36	00:10:04 (00:10:47)
2	$2\times2\times2$, $v=0.2$, $D3$	$32\times32\times32$	32768	107811	$2.760e+04$	425	07:16:05 (65:13:37)
3	$2\times2\times4$, $v=0.4$, $D1$	$32\times32\times64$	65536	212355	$9.998e+01$	78	01:54:45 (07:58:35)
4	$3\times3\times3$, $v=0.3$, $D2$	$48\times48\times48$	110592	352947	$3.233e+01$	48	01:55:37 (04:58:52)
5	$2\times4\times6$, $v=0.3$, $D5$	$32\times64\times96$	196608	624195	$2.935e+02$	94	12:48:47 (96:54:17)
6	$3\times3\times6$, $v=0.4$, $D1$	$48\times48\times96$	221184	698691	$9.301e+01$	70	05:40:37 (44:55:48)
7	$4\times4\times4$, $v=0.2$, $D5$	$64\times64\times64$	262144	823875	$1.406e+02$	78	17:29:22 (133:45:58)
8	$4\times5\times5$, $v=0.3$, $D4$	$64\times80\times80$	409600	1279395	$1.767e+02$	51	30:48:47 (235:15:29)
9	$5\times5\times5$, $v=0.4$, $D6$	$80\times80\times80$	512000	1594323	$7.423e+05$	37	69:15:50 (Inf)
10	$4\times4\times8$, $v=0.4$, $D1$	$64\times64\times128$	524288	1635075	$9.041e+01$	64	50:39:41 (242:05:32)
11	$4\times6\times6$, $v=0.2$, $D2$	$64\times96\times96$	589824	1834755	$3.690e+01$	77	33:25:49 (Inf)
12	$6\times6\times6$, $v=0.4$, $D1$	$96\times96\times96$	884736	2738019	$6.917e+01$	40	76:23:04 (-)
13	$4\times8\times8$, $v=0.3$, $D4$	$64\times128\times128$	1048576	3244995	$1.845e+02$	51	74:20:51 (-)

Indeed, Table 4 gives some hints to differentiate between the first and the second approach. Since the SDC (second approach) needs three Cholesky factorization (and their transposes), problems number 12 and 13 (problems bigger than $1,834,755$ degrees of freedom) cannot be solved. Moreover, due to time limitations, the execution of problems number 9 and 11, for the same approach, is not finished. In these problems, one optimization iteration requires at least 27 hours.

The modest number of iterations that TopIP needs for finding a KKT point is remarkable. Most of the problems, in this numerical experiment, are solved using less than 100 iterations. However, it seems that TopIP has more difficulties in finding an optimal design on problems with design domain D3. In fact, problem 2 requires 425 iterations. For this problem, the convergence rate of TopIP has decreased because it switches to the monotone approach. Thus, further investigations need to be done to improve the monotone version.

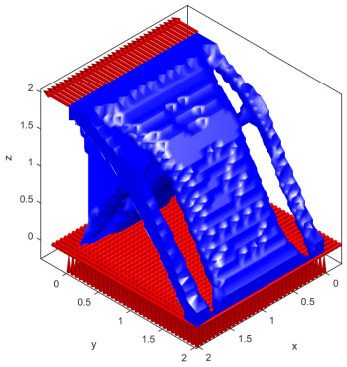
The optimized designs of some problems of the test set are represented in Figure 5. The density value of the isosurface displayed is 0.8.

The rest of the section deals with an extensive study of the performance of TopIP. First of all, the relation between the number of TopIP iterations and the size of the problem (number of elements) is represented in Figure 6a. It points out, more visually, that the convergence of TopIP is not affected by the size of the problem.

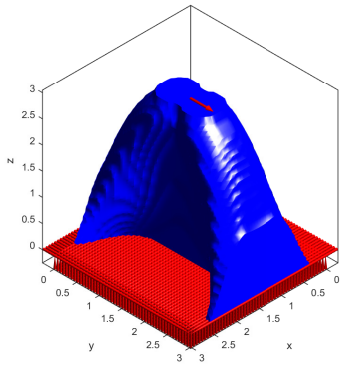
Similarly, Figure 6b shows a comparative study of the computational time of TopIP with respect to the number of elements, using the two approaches. The dotted black line is included as a reference to the linear behaviour. The computational effort of the second approach (SDC) is noticeably larger than the first approach (FGMRES+MC), but more important, it increases more than linearly. In contrast, the growth of the computational time of the first approach is very slow with respect to the number of elements.

Since the number of optimization iterations is independent of the size of the problem, it is important to study the effort of each iteration. In other words, the number of iterative iterations required at every TopIP iteration. Figure 7 shows the number of FGMRES iterations (top level) at each interior point iteration for some problems. In the first approach, the iterations remain constant between two and 7 (Figure 7a). In contrast, the second approach needs slightly more iterations as the optimization advances (Figure 7b). In addition, the number of FGMRES goes up to 250. Ultimately, there is almost no relation between the number of iterations and the size of the problem. This observation is clearer in Figure 8.

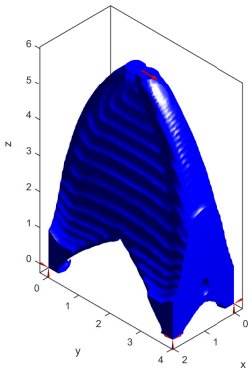
The average number of iterations and computational time of FGMRES over the TopIP iterations at different problem sizes are represented in Figures 8 and 9. Figures 8a and 8b represent the mean (and standard deviation) of the number of iterations needed to solve one KKT system, for the first and the second approaches, respectively. The average of FGMRES iterations is not affected by the size of the problem. In the



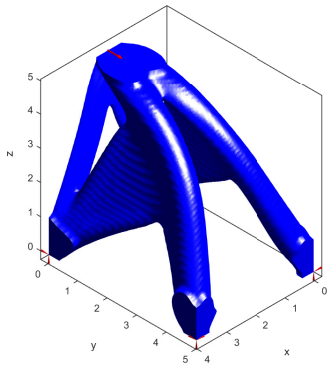
(a) Domain D3, $2 \times 2 \times 2$ mesh, with $V = 0.2$



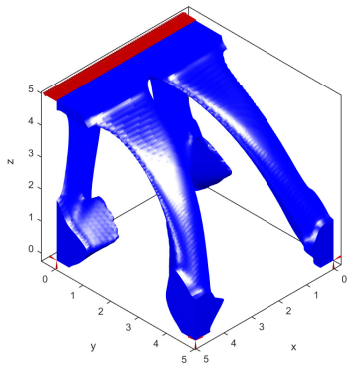
(b) Domain D2, $3 \times 3 \times 3$ mesh, with $V = 0.3$



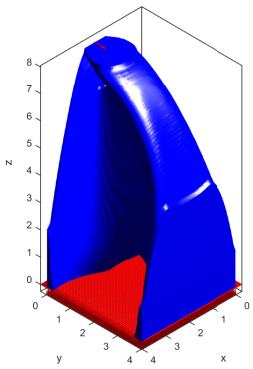
(c) Domain D5, $2 \times 4 \times 6$ mesh, with $V = 0.3$



(d) Domain D4, $4 \times 5 \times 5$ mesh, with $V = 0.3$

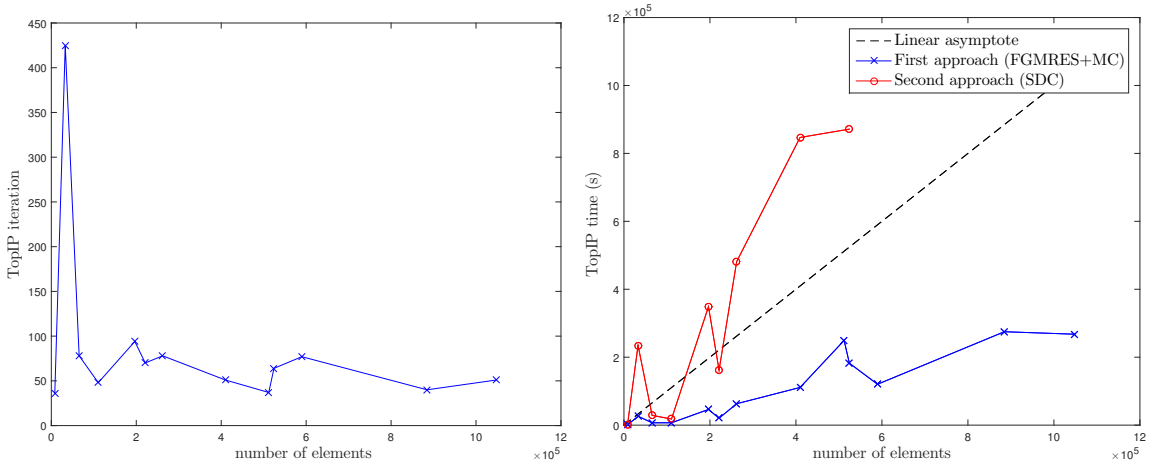


(e) Domain D6, $5 \times 5 \times 5$ mesh, with $V = 0.4$



(f) Domain D1, $4 \times 4 \times 8$ mesh, with $V = 0.4$

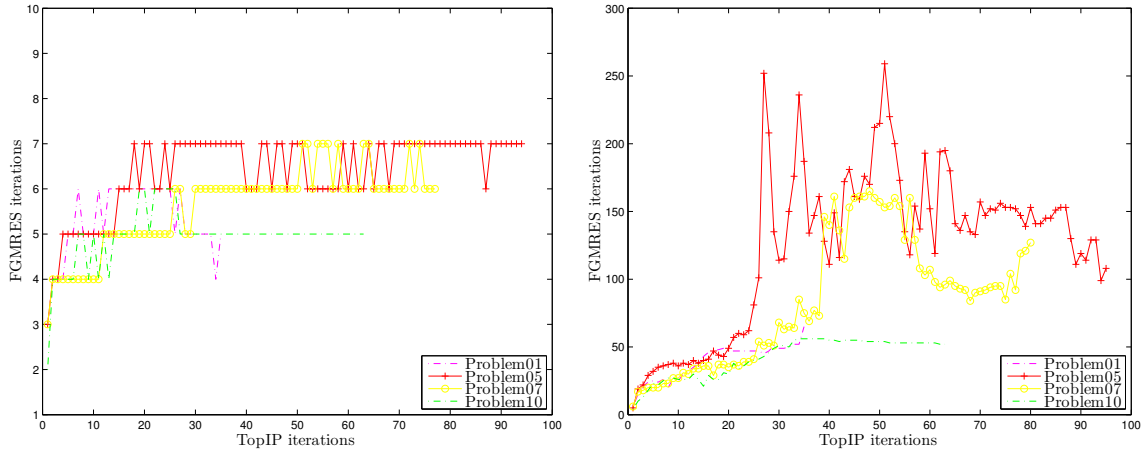
Figure 5: Optimized designs examples of some problems in the test set.



(a) Number of TopIP iterations versus the size of the problem.

(b) Computational time of TopIP versus the size of the problem

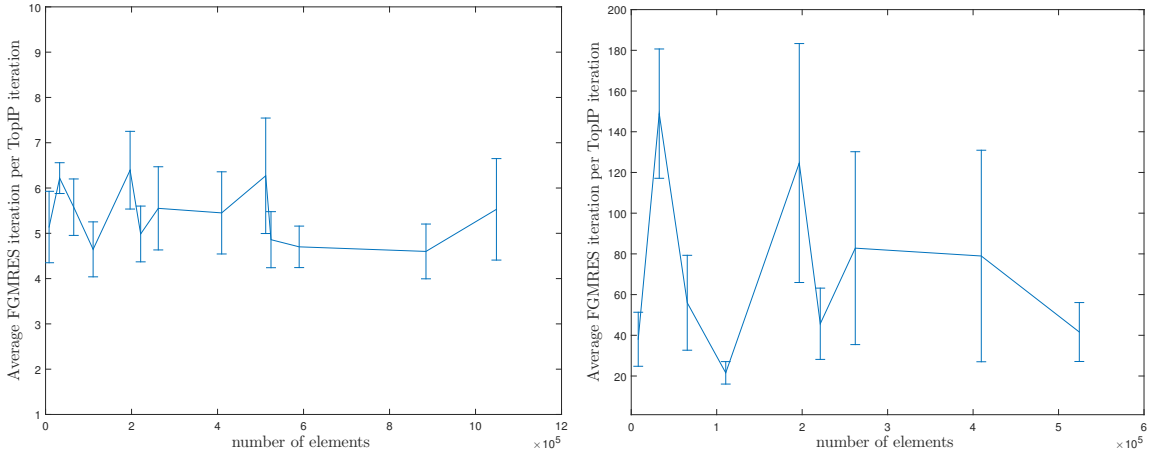
Figure 6: Number of optimization iterations (6a) and computational time of TopIP (6b) versus the size of the problem (number of elements). Figure 6b shows the total time of TopIP using both, the first (blue line) and the second (red line) iterative approaches. The black line represents a linear behaviour for reference.



(a) Iterative method using the first approach

(b) Iterative method using the second approach

Figure 7: Number of FGMRES iterations required to solve the KKT system at each TopIP iteration. Figures 7a and 7b show the results for the first and second iterative approach, respectively for some of the problems in the test set.



(a) Iterative method using the first approach (b) Iterative method using the second approach

Figure 8: Comparative study between the FGMRES iterations required to solve one KKT system (on average) versus the size of the problem (number of elements), for the first (Figure 8a) and second (8b) approaches.

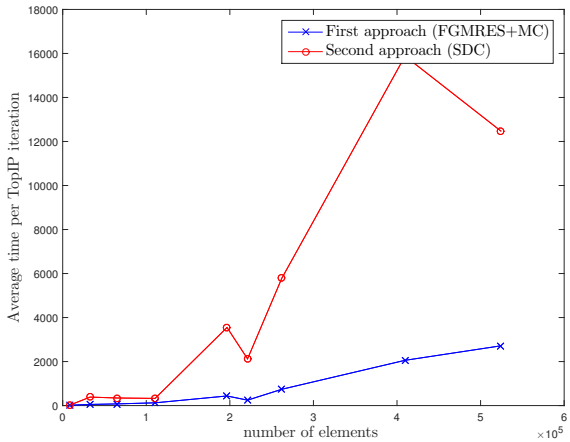


Figure 9: Comparative study between the computational time required to solve one KKT system (on average) versus the size of the problem (number of elements). The blue line represents the first approach (FGMRES+MC) and the red line represents the second approach (SDC). The figure shows the results for the subset test of problems solved using the second approach.

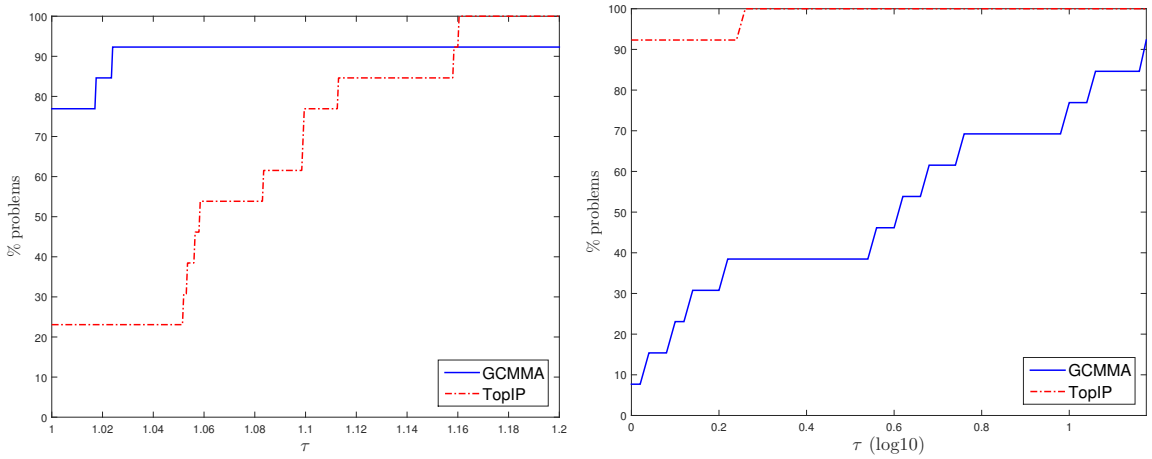
first approach, the iterations remain constant (average of 5-6), while the range of iterations in the second approach is substantially large, between 25 to 150 (on average). Additionally, the standard deviation is also greater, i.e., the number of iterations varies a lot between optimization iterations. In other words, the second approach is more sensitive to the optimization process. These fluctuations in the number of iterations make this approach more unstable and less robust than the first approach. Figure 9 shows that the computational time spent in solving one KKT system (on average) using SDC is also larger than using FGMRES+MC. This is due to the computational effort needed in the factorization of $\mathbf{K}_{xx}$, $\mathbf{K}_{yy}$, and $\mathbf{K}_{zz}$, and the solution of linear systems with them.

Since the second approach needs more memory and time, the use of FGMRES+MC as preconditioner of the stiffness matrix in the iterative method is highly recommended.

Finally, the performance of TopIP (first approach) is compared with the first-order structural optimization method GCMMA (Global Convergent Method of Moving Asymptotes) [67]. The performance of both solvers is evaluated using the objective function values, the number of iterations, and the total computational time. This comparative study is done using performance profiles ([25]) similar to [57] and [56]. The test set used for this comparison is the 13 problems detailed in Table 4. Those problems with a KKT error higher than $\omega_{\max} = 10^{-4}$ are penalized since the final design is considered incorrect (the problem is not solved). For this numerical study, only one problem solved with GCMMA has a KKT error higher than $\omega_{\max}$. All details about performance profiles in topology optimization problems can be found in [57].

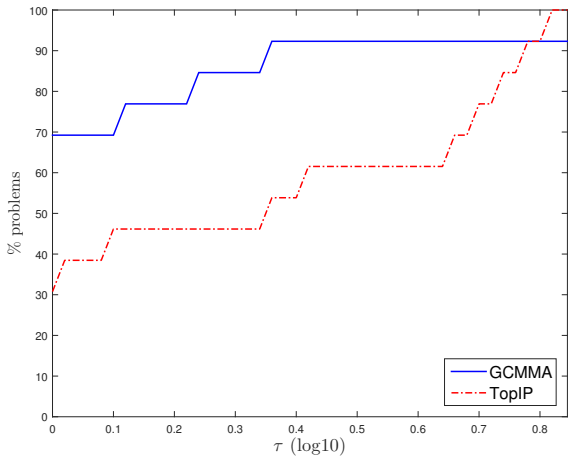
Figure 10 shows that the second-order (TopIP) method clearly needs less iterations than the first-order (GCMMA) method. Although, the computational time of TopIP is still higher than GCMMA, the maximum difference between cpu times is $\tau_{\max} = 6.02$. More importantly, the 30% of TopIP winners refers to the biggest problems in the test set. Figure 11 shows the computational time of both GCMMA and TopIP, for varying size of the problem. The computational time of TopIP grows slower than GCMMA. Thus, for larger problems, TopIP is expected to outperform first-order methods with respect to time. The computational time can be reduced, even more, if the implementation is parallelized. Ultimately, the objective function value obtained with TopIP is not as good as it was expected for second-order solvers [57]. In fact, GCMMA produces designs with lower objective function values than TopIP. Nevertheless, the results show the efficiency and robustness of the proposed iterative method. TopIP will not be able to solve problems with more than three million degrees of freedom without iterative solvers.

In general, the performance of nonlinear optimization methods is highly affected by the selection of the parameters. The scaling of the problem may cause TopIP to converge to local minima. Another possible reason might be the excessive flexibility of the adaptive strategy. Moreover, the tolerance in the iterative method is quite large,



(a) Objective function value

(b) Number of iterations



(c) Computational time

Figure 10: Performance profiles for a test set of 13 3D minimum compliance problems. The performance is measured by the objective function value (10a), the number of iterations (10b), and the computational time (10c).

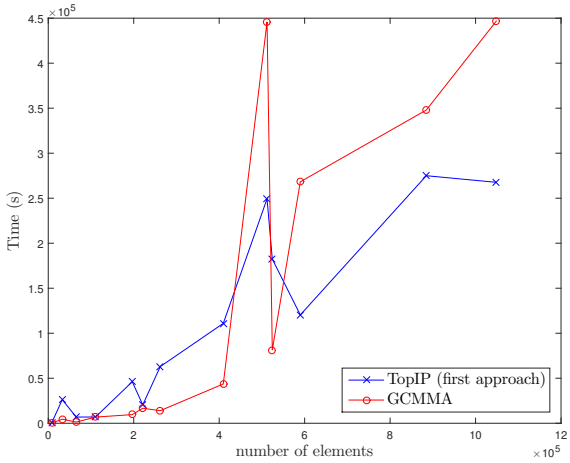


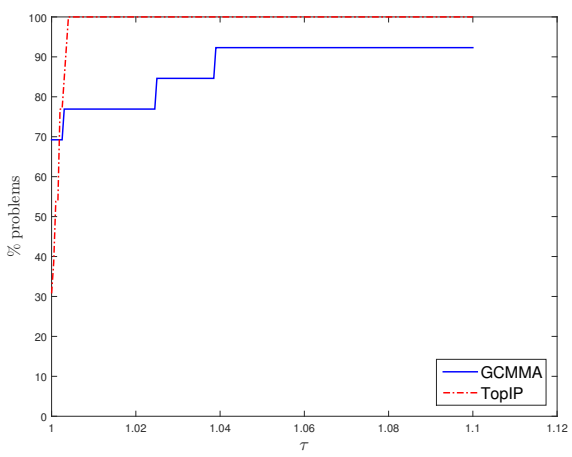
Figure 11: Comparative study of the computational time required to solve the problem using TopIP and GCMMA with respect to the number of elements.

$\omega = 10^{-6}$, producing some errors in the descent direction that may be significant to the final objective function value. The stopping criterion of TopIP is measured with the KKT conditions of the sub-problem, thus, the high value of $\mu_{\min} = 10^{-7}$ is the most likely reason of these large objective function values. TopIP could stop without actually converging. In order to obtain good results, TopIP is run again with $\mu_{\min} = 10^{-9}$, $E_0 = 10$, and $E_{\min} = 10^{-2}$, showing promising results gathered in Figure 12.

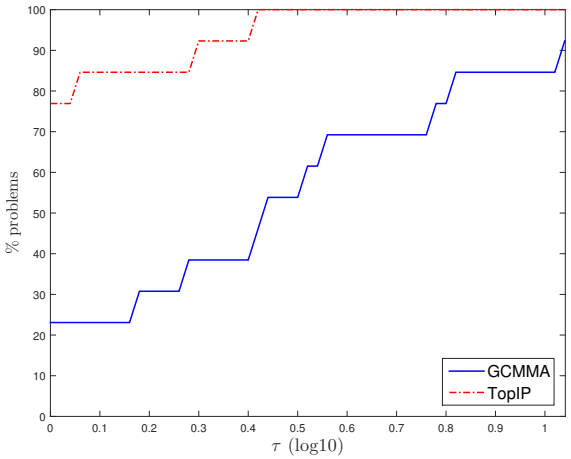
With these new parameter settings, the objective function value of TopIP has significantly improved, outperforming GCMMA for a very small τ ($\tau = 1.0018$). In contrast, the computational effort and the number of iterations have slightly increased, but not critically. For instance, TopIP still converges, generally, in about 100 iterations, with 78% chances of winning and a maximum ratio value of $\tau_{\max} = 11$. However, the percentage of success with respect to computational time has decreased from 30% to 22%. In addition, the behaviour of Figure 11 is no longer observed for this parameter selection (see Figure 13). Thus, the parallelization of the code is important to outperform GCMMA with respect to time.

Table 5 shows the results for this new TopIP configuration. The study of the performance of the two iterative approaches does not change. The number of iterative iterations (and computational time) is almost identical for $\mu_{\min} = 10^{-7}$ and for $\mu_{\min} = 10^{-9}$.

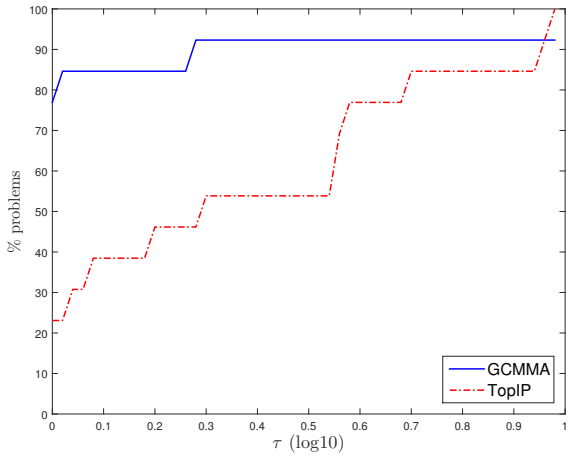
This result encourages further investigation to obtain an even better and more accurate implementation of the interior point method. The previous results evidence that the performance of TopIP is sensitive to the parameter selection, and thus, further work can be done to obtain an optimal parameter setting.



(a) Objective function value



(b) Number of iterations



(c) Computational time

Figure 12: Performance profiles for a test set of 13 3D minimum compliance problems. The performance is measured by the objective function value (12a), the number of iterations (12b), and the computational time (12c) (with $\mu_{\min} = 10^{-9}$, $E_0 = 10$, and $E_{\min} = 10^{-2}$).

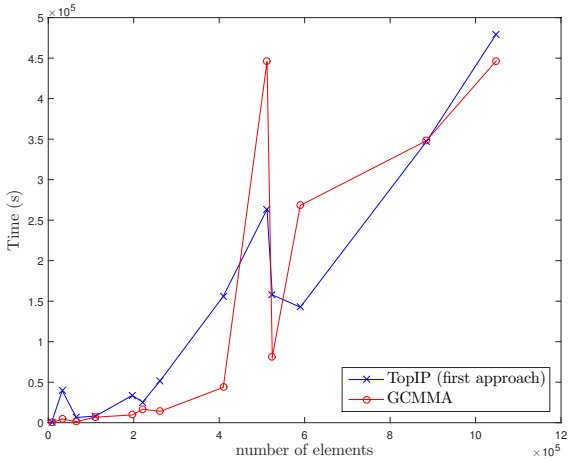


Figure 13: Comparative study of the computational time required to solve the problem using TopIP and GCMMA with respect to the number of elements (with $\mu_{\min} = 10^{-9}$, $E_0 = 10$, and $E_{\min} = 10^{-2}$).

Table 5: Results of the optimized design obtained with TopIP for 13 3D minimum compliance problems. The table contains the number of the problem (N), its description (length ratio, volume fraction, and domain type), the mesh discretization, the number of elements, and the number of degrees of freedom. In addition, the objective function value, the number of optimization iterations, and the computational time obtained with TopIP (first approach) are collected, for $\mu_{\min} = 10^{-9}$ ($E_0 = 10$ and $E_{\min} = 10^{-2}$).

N	**Description**	**Mesh**	**n**	**d**	$f(\mathbf{t})$	**Iter**	**Time [hh:mm:ss]**
1	$1\times1\times2$, $v=0.4$, $D1$	$16\times16\times32$	8192	28611	$1.269e+02$	95	00:14:32
2	$2\times2\times2$, $v=0.2$, $D3$	$32\times32\times32$	32768	107811	$2.757e+04$	616	11:02:19
3	$2\times2\times4$, $v=0.4$, $D1$	$32\times32\times64$	65536	212355	$9.469e+01$	91	01:46:43
4	$3\times3\times3$, $v=0.3$, $D2$	$48\times48\times48$	110592	352947	$3.057e+01$	77	02:16:59
5	$2\times4\times6$, $v=0.3$, $D5$	$32\times64\times96$	196608	624195	$2.714e+02$	76	09:20:20
6	$3\times3\times6$, $v=0.4$, $D1$	$48\times48\times96$	221184	698691	$8.370e+01$	94	07:04:57
7	$4\times4\times4$, $v=0.2$, $D5$	$64\times64\times64$	262144	823875	$1.338e+02$	70	14:24:16
8	$4\times5\times5$, $v=0.3$, $D4$	$64\times80\times80$	409600	1279395	$1.676e+02$	62	43:21:48
9	$5\times5\times5$, $v=0.4$, $D6$	$80\times80\times80$	512000	1594323	$7.041e+05$	47	73:09:59
10	$4\times4\times8$, $v=0.4$, $D1$	$64\times64\times128$	524288	1635075	$7.813e+01$	114	43:49:55
11	$4\times6\times6$, $v=0.2$, $D2$	$64\times96\times96$	589824	1834755	$3.187e+01$	132	39:36:47
12	$6\times6\times6$ $v=0.4$ $D1$	$96\times96\times96$	884736	2738019	$6.314e+01$	54	96:21:04
13	$4\times8\times8$, $v=0.3$, $D4$	$64\times128\times128$	1048576	3244995	$1.686e+02$	83	133:16:10

8 Conclusion

The article presents a robust and efficient iterative method for solving the large-scale indefinite linear systems arising in interior point methods for structural topology optimization problems. The proposed iterative method solves the most expensive step in interior point algorithms (the solution of a KKT system), reducing, significantly, the amount of time and memory required to direct solvers. The saddle-point system is solved using a combination of the state-of-the-art iterative methods, exploiting the structure of the problem. In particular, the interior point method (TopIP) is implemented for the minimum compliance problem based on a density-based approach in the nested form. TopIP approximates the Hessian using part of the exact second-order information.

Both adaptive and monotone barrier parameter updates are part of TopIP to guarantee global convergence. The iterative method is based on flexible GMRES with an incomplete block preconditioner matrix. This preconditioner needs the solution of smaller systems where different techniques such as FGMRES, multigrid cycles, and block diagonal preconditioners are used.

The numerical results show the robustness and efficiency of TopIP, where large 3D topology optimization problems are solved. TopIP is able to optimize designs with more than three million degrees of freedom.

The number of TopIP iterations is constant, independently on the size of the problem. Moreover, TopIP requires fewer iterations than GCMMA, in general. The number of iterative iterations remains also constant through the optimization process and at different problem sizes. Finally, the computational time of TopIP (first approach) increases slowly (lower than linearly) with respect to the size of the problem, outperforming even GCMMA for the largest problems. All these characteristics give to TopIP excellent properties for solving very large-scale problems.

At every preconditioner operation, several systems are solved where only the right side vector is modified. Thus, additional investigations should be done to develop efficient solvers to reduce the computational time even more. Future work should be also done regarding the extension of the geometric multigrid to algebraic multigrid (AMG) [50], [33], and [73], to be able to apply this interior point method to unstructured meshes. Finally, further research in regards to the adaptive strategy and to the parameter selection are recommended to speed up the convergence rate and the accuracy of TopIP.

Acknowledgements

We would like to thank Professor Krister Svanberg at KTH in Stockholm for providing the MATLAB implementation of GCMMA.

References

[1] N. Aage, E. Andreassen, and B. S. Lazarov. Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization*, 51(3):565–572, 2014.

[2] N. Aage and B. S. Lazarov. Parallel framework for topology optimization using the method of moving asymptotes. *Structural and Multidisciplinary Optimization*, 47(4):493–505, 2013.

[3] P. R. Amestoy, T. A. Davis, and I. S. Duff. Algorithm 837: AMD, an Approximate Minimum Degree Ordering Algorithm. *ACM Transactions on Mathematical Software*, 30(3):381–388, 2004.

[4] O. Amir, N. Aage, and B. S. Lazarov. On multigrid-CG for efficient topology optimization. *Structural and Multidisciplinary Optimization*, 49(5):815–829, 2014.

[5] O. Amir, M. Stolpe, and O. Sigmund. Efficient use of iterative solvers in nested topology optimization. *Structural and Multidisciplinary Optimization*, 42(1):55–72, 2009.

[6] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[7] J. S. Arora and Q. Wang. Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization*, 30(4):251–272, 2005.

[8] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):192–202, 1989.

[9] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9–10):635–654, 1999.

[10] M. P. Bendsøe and O. Sigmund. *Topology optimization: Theory, methods and applications*. Springer, 2003.

[11] H. Y. Benson and D. F. Shanno. Interior-point methods for nonconvex nonlinear programming: cubic regularization. *Computational Optimization and Applications*, 58(2):323–346, 2013.

[12] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei. A comparative study of large-scale nonlinear optimization algorithms. Technical Report ORFE-01-04, Operations Research and Financial Engineering, Princeton University, 2002.

[13] M. Benzi, G . H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2004.

[14] M. Benzi and M. Tuma. A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2):305–340, 1999.

[15] M. Benzi and A. J. Wathen. Some preconditioning techniques for saddle point problems. In *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, pages 195–211. Springer, 2008.

[16] P. T. Boggs and J. W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

[17] T. Borrvall and J. Petersson. Large-scale topology optimization in 3D using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6201–6229, 2001.

[18] A. Borzì and V. Schulz. Multigrid methods for PDE optimization. *SIAM Review*, 51(2):361–395, 2009.

[19] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[20] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[21] R. H. Byrd, G. Lopez-Calva, and J. Nocedal. A line search exact penalty method using steering rules. *Mathematical Programming*, 133(1-2):39–73, 2012.

[22] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Transactions on Mathematical Software*, 35(3):22:1–22:14, 2008.

[23] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 1987.

[24] F. E. Curtis, O. Schenk, and A. Wächter. An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. *SIAM Journal on Scientific Computing*, 32(6):3447–3475, 2010.

[25] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[26] T. Dreyer, B. Maar, and V. Schulz. Multigrid optimization in applications. *Journal of Computational and Applied Mathematics*, 120(1-2):67–84, 2000.

[27] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89(3):507–541, 1996.

[28] A. Evgrafov. On the reduced Hessian of the compliance. *Structural and Multidisciplinary Optimization*, 50(5):1197–1199, 2014.

[29] A. Evgrafov, C. J. Rupp, K. Maute, and M. L. Dunn. Large-scale parallel topology optimization using a dual-primal substructuring solver. *Structural and Multidisciplinary Optimization*, 36(4):329–345, 2008.

[30] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons. Reprinted by SIAM Publications, 1990.

[31] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

[32] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large Scale Constrained Optimization. *SIAM Journal on Optimization*, 47(4):99–131, 2005.

[33] M. Griebel, D. Oeltz, and M. A. Schweitzer. An Algebraic Multigrid Method for Linear Elasticity. *SIAM Journal on Scientific Computing*, 25(2):385–407, 2003.

[34] I. Gustafsson and G. Lindskog. On Parallel Solution of Linear Elasticity Problems. Part I : Theory. *Numerical Linear Algebra with Applications*, 5:123–139, 1998.

[35] W. Hackbusch. *Multigrid Methods and Applications*. Springer, 1985.

[36] S. Häfner and C. Könke. Multigrid preconditioned conjugate gradient method in the mechanical analysis of heterogeneous solids. Technical report, 17th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering, Germany, 2006.

[37] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[38] R. H. W. Hoppe, C. Linsenmann, and S. I. Petrova. Primal-dual Newton methods in structural optimization. *Computing and Visualization in Science*, 9(2):71–87, 2006.

[39] R. H. W. Hoppe and S. I. Petrova. Primal-dual Newton interior point methods in shape and topology optimization. *Numerical Linear Algebra with Applications*, 11(56):413–429, 2004.

[40] R. H. W. Hoppe, S. I. Petrova, and V. Schulz. Primal-dual Newton-type interior-point method for topology optimization. *Journal of Optimization Theory and Application*, 114(3):545–571, 2002.

[41] Kaustuv. *IPSOL: An Interior Point Solver for Nonconvex Optimization Problems*. PhD thesis, Stanford University, 2009.

[42] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1995.

[43] C. J. Lin and J. J. Moré. Incomplete Cholesky Factorizations with limited memory. *SIAM Journal on Scientific Computing*, 21(1):24–45, 2006.

[44] C. Linsenmann. On the convergence of right transforming iterations for the numerical solution of PDE constrained optimization problems. *Numerical Linear Algebra with Applications*, 19(4):621–638, 2012.

[45] K. Liu and A. Tovar. An efficient 3D topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 5(6):1175–1196, 2014.

[46] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[47] B. Maar and V. Schulz. Interior point multigrid methods for topology optimization. *Structural and Multidisciplinary Optimization*, 19(3):214–224, 2000.

[48] A. Mahdavi, R. Balaji, M. Frecker, and E. M. Mockensturm. Topology optimization of 2D continua for minimum compliance using parallel computing. *Structural and Multidisciplinary Optimization*, 32(2):121–132, 2006.

[49] N. Maratos. *Exact penalty function algorithms for finite dimensional and control optimization problems*. PhD thesis, University of London, 1978.

[50] B. Metsch. *Algebraic multigrid (AMG) for saddle point systems*. PhD thesis, PhD in Mathematics, Faculty of Natural Sciences of Rheinischen Friedrich-Wilhelms-Universität Bonn, 2013.

[51] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20(3):286–307, 1994.

[52] J. Nocedal, R. Wächter, and R. A. Waltz. Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization*, 19(4):1674–1693, 2009.

[53] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[54] C. E. Orozco and O. N. Ghattas. A reduced SAND method for optimal design of nonlinear structures. *International Journal for Numerical Methods in Engineering*, 40(15):2759–2774, 1997.

[55] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[56] S. Rojas-Labanda and M. Stolpe. An efficient second-order SQP method for structural topology optimization. Technical report, Technical University of Denmark, Department of Wind Energy. Submitted, 2015.

[57] S. Rojas-Labanda and M. Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization, In print*, 2015. DOI: 10.1007/s00158-015-1250-z.

[58] G. I. N. Rozvany and M. Zhou. The COC algorithm, part I: Cross-section optimization or sizing. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):281–308, 1991.

[59] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural Optimization*, 4(3–4):250–252, 1992.

[60] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[61] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[62] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[63] V. Schulz and G. Wittum. Transforming smoothers for PDE constrained optimization problems. *Computing and Visualization in Science*, 11(4–6):207–219, 2008.

[64] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(2):68–75, 1998.

[65] R. Simon. *Multigrid solvers for saddle point problems in PDE-constrained optimization*. PhD thesis, Johannes Kepler Universität, 2008.

[66] K. Svanberg. The method of moving asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

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

[68] O. Tatebe. The multigrid preconditioned conjugate gradient method. Technical report, 6th Copper Mountain Conference on Multigrid Methods, 1993.

[69] Inc. The MathWorks. Optimization Toolbox User's Guide R release 2014a, 2014.

[70] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13(1-3):231–252, 1999.

[71] K. Vemaganti and W. E. Lawrence. Parallel methods for optimality criteria-based topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 194(34-35):3637–3667, 2005.

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

[73] C. Wagner. Introduction to algebraic multigrid. Technical Report http://perso.uclouvain.be/alphonse.magnus/num2/amg.pdf, Course notes of an algebraic multigrid course at the University of Heidelberg, 1998/99, 1998.

[74] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3):391–408, 2006.

[75] S. Wang, E. Sturler, and G. H. Paulino. Large scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69:2441–2468, 2007.

[76] P. Wesseling. *An introduction to multigrid methods*. John Wiley & Sons, 1992.

[77] G. Wittum. Multigrid methods for Stokes and Navier-Stokes equations. *Numerische Mathematik*, 54(5):543–563, 1989.

[78] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. *Numerische Mathematik*, 57(3):15–38, 1990.

[79] H. Yamashita. A globally convergent primal-dual interior point method for constrained optimization. *Optimization Methods and Software*, 10(2):2–4, 1998.

[80] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):309–336, 1991.

DTU Wind Energy
Department of Wind Energy
Technical University of Denmark

Frederiksborgvej 399
4000 Roskilde
www.vindenergi.dtu.dk