A direct approach to structural topology optimization

Dirk Pieter Munro

Dissertation presented for the degree of Doctor of Philosophy in the Faculty of Engineering at Stellenbosch University.

Promotor:
Prof. Albert A. Groenwold

March 2017

DECLARATION

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

This dissertation includes 3 original papers published in peer-reviewed journals or books and 2 unpublished publications. The development and writing of the papers (published or unpublished) were the principal responsibility of myself and, for each of the cases where this is not the case, a declaration is included in the dissertation indicating the nature and extent of the contributions of co-authors.

Date: March 2017

D.P. Munro

Abstract

This dissertation addresses various topics that emerge from the unification of conventional structural optimization—based on 'sequential approximate optimization' (SAO)—with the alternative 'direct'—or 'simultaneous analysis and design' (SAND)—formulation of the structural topology design problem. Structural topology optimization—in the form of a 'material distribution problem'—is a generalisation of structural optimization, encompassing and simultaneously addressing al the aspects of structural design. In structural optimization, SAO techniques are preferred because the number of structural analyses—which are expensive, computationally speaking—are ostensibly minimised. However, particularly in the presence of local state-based constraints—*e.g.* local stress constraints—the sensitivity analyses which accompany traditional 'nested analysis an design' (NAND) methods require a prohibitive number of structural analysis runs per design iteration.

In the alternative SAND setting, structural analysis is conducted approximately and sequentially: the finite element (FE) equilibrium equations are retained as a set of nonlinear equality constraints and the state variables—*i.e.* displacements—form part of the overall set of primal variables. Therefore, the FE equilibrium equations may only be satisfied at convergence of the optimization algorithm, and the complex and expensive sensitivity analyses associated with state-based constraints, simplify to the calculation of partial derivatives. Moreover, the equivalent of a single structural analysis only is required per design iteration, notwithstanding the imposition of a large number of state-based constraints.

Based on a dual method in theory, we propose a separable and strictly convex quadratic Lagrange-Newton approximate subproblem for use in SAO of the SAND formulated topology design problem. In classical (simply-constrained) minimum compliance design, the dual statement of the subproblem is equivalent to the ever-popular optimality criteria (OC) approach—a class of NAND methods. This relates, in turn, to the known equivalence between dual SAO-NAND algorithms based on intervening variables and the OC method.

Due to the presence of nonlinear equality constraints, the classical SAO procedure (exclusively geared, traditionally, for inequality constrained problems) is extended to a general, nonlinear and nonconvex, mathematical programming framework. It turns out that conventional techniques of enforced convergence and termination in traditional NAND-based SAO may be transplanted into the SAND setting with only minor complications.

It is demonstrated that the compounded issues of existence of solutions, mesh-dependence, local minima, and macro-scale manufacturability, may be addressed in a computationally efficient manner by the imposition of so-called 'slope constraints'—point-wise bounds on the gradient of the material distribution function. For global optimization, random multistart strategies may be pursued. A specialized version of 'linear independence constraint qualification'

(LICQ) may hold in many practical situations, and because the global stiffness matrix is not inverted *per se*, material density variables are permitted a value of *zero* on the lower bound. Hence, singular local minima are feasible and available—in both simply-constrained and local stress-constrained problems—and may be converged to with standard gradient-based optimization methods without having to resort to any relaxation or perturbation techniques whatsoever.

Opsomming

Hierdie proefskrif spreek verskeie aspekte aan wat volg op die samevoeging van konvensionele struktuur optimering—gebaseer op herhaalde benaderde optimering (*sequential approximate optimization*; SAO)—met die alternatiewe 'direkte'—of 'gelyktydige analise en ontwerp' (*simultaneous analysis and design*; SAND)—formulering van die strukturele topologie ontwerp probleem. Strukturele topologie optimering is 'n veralgemening van struktuur optimering—in die vorm van 'n materiaal verspreidings probleem—en dus kan al die aspekte van struktuur ontwerp gelyktydig aangespreek word. In struktuur optimering word SAO tegnieke verkies, want die aantal struktuur analises—wat baie duur is, in terme van berekeninge—word oënskynlik geminimeer. Nietemin, wanneer lokale spanning begrensings bygevoeg word, vereis die sensitiwiteit analise wat gepaard gaan met tradisionele 'geneste analise en ontwerp' (*nested analysis and design*; NAND) metodes, 'n groot aantal struktuur analises.

In die alternatiewe SAND raamwerk word struktuur analise benaderd en herhaaldelik uitgevoer: die eindige element (EE) ewewigs vergelykings word behou in die optimering probleem as 'n groot aantal gelykheidsbegrensings, en die verplasings vorm deel van die algehele stel primale veranderlikes. Dit wil sê, die EE ewewigs vergelykings sal net tevrede gestel word indien die optimerings algoritme konvergeer, en die duur sensitiwiteit analises wat gepaard gaan met begrensings in terme van verplasings—soos bv. spanning begrensings—vereenvoudig na die berekening van eenvoudige parsiële afgeleides. Wat meer is, net 'n enkele struktuur analise word benodig per ontwerp iterasie, al word daar 'n groot aantal lokale spanning begrensings bygevoeg.

Gebaseer op 'n duale metode in teorie, stel ons voor dat 'n skeibare en streng konvekse kwadratiese Lagrange-Newton subprobleem gebruik word vir SAO van die SAND geformuleerde probleem. In klassieke maksimum styfheid ontwerp kan aangetoon word dat die duale metode soortgelyk is aan die populêre optimale kriteria (*optimality criteria*; OC) metode—wat 'n NAND formulering is. Dit is, op sy beurt, identies aan die ooreenstemming tussen duale SAO-NAND algoritmes gebaseer op tussentydse veranderlikes en die OC metode.

Omdat die optimerings probleem nie-lineêre gelykheidsbegrensings bevat, moet die SAO prosedure (wat gewoonlik net van toepassing is op nie-gelykheid begrensde probleme) toegepas word in 'n algehele, nie-lineêre en nie-konvekse, wiskundige programmerings raamwerk. Inderdaad, konvensionele tegnieke wat konvergensie af dwing in tradisionele NAND-SAO, kan amper net so gebruik word in die SAND raamwerk.

Dit word bewys dat die ooreenstemmende uitdagings wat gepaard gaan met die bestaan van oplossings, maas-onafhanklikheid, lokale minima en makro-skaal vervaardiging tegnieke, kan geadresseer word deur middel van helling-begrensings—wat puntsgewys die gradiënt van die materiaal verspreiding bind—solank dit ook gepaard gaan met 'n lukrake multi-begin strate-

gie (vir globale optimering). ’n Gespesialiseerde soort ‘lineêre onafhanklike begrensing kwalifikasie’ (*linear independence constraint qualification*; LICQ) mag geld in baie praktiese gevalle, en omdat die inverse van die globale styfheid matriks nie *per se* uitgewerk word nie, word ’n onderste grens van presies nul toegelaat vir die materiaal veranderlikes. Dit beteken, op sy beurt, dat die singulariteite wat gepaard gaan met die lokale minima in beide eenvoudig-begrensde probleme en lokale spanning-begrensde probleme, oorkom kan word met gewone gradiënt-gebaseerde optimerings metodes, sonder dat daar enigsins verslapping- of perturbasie-tegnieke gebruik moet word.

Acknowledgements

The author wishes to thank Stellenbosch University's Department of Mechanical and Mechatronic Engineering for funding, the administrator of the Rhasatsha HPC, Mr. C. Möller, and last but by no means least, Prof. A.A. Groenwold.

The destruction of the world is mirrored in the slowing currents of our sun. The extraction of things from the ground. The emptying of reserves. Degradation. A tightening of the oesophagus. The hottest year on record. Let us embrace nothingness and try to fill it with being because in our ever-increasing specialised imaginations we trundle toward nothingness and absence not with any ceremony or sanctity but blindly. Our domain over matter, things, people, is only ever a progression downwards, away from the light of our roiling sun into the empty space of a world unfurling, a fiction that is real.

—JOHN HOLTEN
Oslo, Norway (2015)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An abridged version of this chapter, entitled 'A contemplation on topology optimization', can be found in the December 2016 issue of Digital Engineering [1]. Digital Engineering is an online magazine (not a peer-reviewed journal) which reports on high-performance computing, simulation-based modelling, and computer-aided design.

A contemplation on topology optimization

Topology optimization is an algorithmic approach to a ubiquitous engineering problem: how to distribute a limited resource (*e.g.* material), in a predefined design domain—subject to applied loads, boundary conditions, and certain design restrictions—in an optimal way? In engineering terms, the state of material is discrete: at each and every spatial position material is either *present* or *absent*. Therefore, the term *material distribution* is often used interchangeably with terms like *shape*, *geometry*, *connectivity*, or *lay-out*. The word *topology* is derived from the Greek word *topos*, literally meaning *place*. *Topology* thus refers to the study of *place*: the relative positions of the constituent parts of a structure, and the manner in which these are connected, interrelated, and arranged.

Topology optimization is typically studied as a sub-field of structural optimization—with automotive, aerospace and civil engineering applications in mind—however, topology optimization is an attempt to *generalise* structural optimization, encompassing and simultaneously addressing all the aspects of structural design[1]. In 1990 Haftka, Gürdal and Kamat wrote that 'structural optimization is still a relatively new field undergoing rapid changes in methods and focus', and that computational cost and integration with 'general-purpose software packages for structural analysis' are some major challenges [3]. For them, 'the motivation is to exploit the available limited resources in a manner that maximizes output or profit' [3, p. 1]. That is, from Haftka, Gürdal and Kamat's point of view, the emphasis is on the *optimality* of the design—exploitation of resources for maximum output and profit—defined in terms of the objective and the restrictions that form a given design scenario.

More than a quarter of a century later, the issues highlighted by Haftka, Gürdal and Kamat have remained much the same, while structural optimization methods, and their focus, have

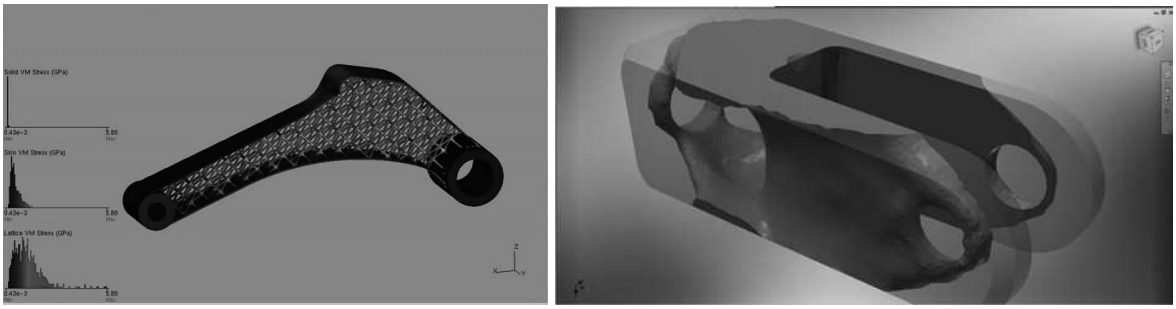
[1] See Bendsøe and Sigmund's canonical introduction [2, p. 1].

changed. On the one hand, topology optimization methods and techniques—*i.e.* very detailed structural optimizations—are plagued by issues pertaining to the control of the complexity of the design—its 'manufacturabillity'. If this aspect is not accounted for, a topologically optimized design may not be physically realizable, or the cost of manufacture may outweigh the efficiency and profit associated with the optimized design. On the other hand, with the development of additive manufacturing techniques—selective laser melting and 3D printing, for example—unprecedented levels of structural complexity may be realised (both in terms of the geometry and the material properties of the design). However, a human designer may not be able to the design on the level of structural complexity afforded by additive production processes. That is, topology optimization may have the potential to fully exploit the increased scope of complexity afforded by additive manufacturing techniques, and these algorithmic procedures may be the only way to translate and communicate the wishes of a human designer—an objective, and the restrictions, within a predefined design domain—to the additive production process. This automated avenue of design and manufacture—the so-called 'third production revolution' [4]—is buttressed by a continual increase in common computational capabilities, improvements in the efficiency of computational methods and, more recently, user-friendly integration with computer aided design (CAD) platforms.

* * *

In July 2015, the software corporation Autodesk released a topology design tool called *Within*—aimed at automotive, aerospace, and orthopaedic implant specialists—given a user defined geometry, a 'generative design' algorithm suggests a lightweight lattice structure to replace the original design—Figure 1.1a—manufacturable with additive techniques like 3-D printing [5]. In October of the same year, Autodesk released a topology optimization tool called *Shape Generator*, set within its popular CAD package—Inventor—and based on Autodesk's Nastran finite element (FE) solver, the software suggests an 'optimal' geometry based on predefined boundary conditions and applied loads [6]—Figure 1.1b. In January 2016, Graphics Systems Corporation, partnered with Dassault Systèmes, released *SIMULIA*, offering 'new capabilities' like topology optimization [7]. In February 2016, Vanderplaats Research and Development Incorporated released *GENESIS*, a structural optimization platform for ANSYS Mechanical [8]. SolidThinking's *Inspire*, first released in 2009, employs a so-called 'morphogenesis tool'—based on Altair's OptiStruct solver—to mimic the way human bones grow and thereby suggest optimal designs. The 2016 version includes a new geometry representation based on computer graphics applications, permitting simple geometry manipulation and smooth integration with additive manufacturing processes [9]. 'The upshot is', writes Anthony Lockwood, editor of Digital Engineering, using topology optimization 'early in the concept development process enhances your ability to create and study structurally efficient, optimally shaped parts and assemblies', which, 'weigh less and use fewer materials, reducing costs and development time' [9].

Since the year 2000 a number of simple and efficient 'instructional' or 'educational' topology optimization codes have been released—invariably set in Matlab, a high-level programming language many engineers are familiar with. The first '99 line topology optimization code' [10] is due to Sigmund, followed by a variety of contributions by various authors [11–14]. Notably,



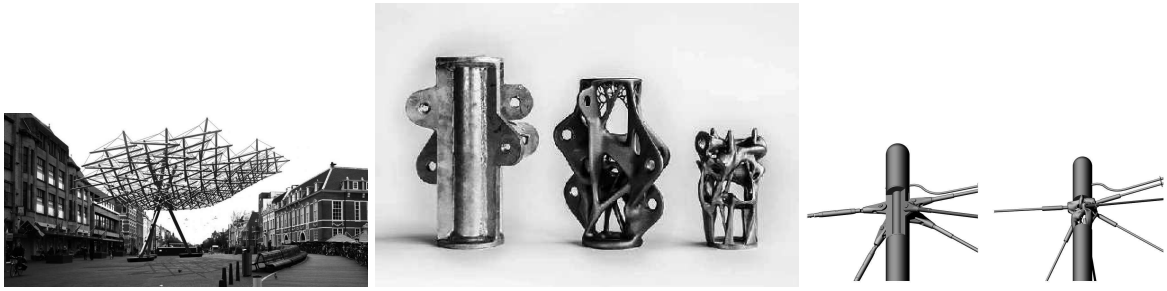
(a) Autodesk *Within* [5]. (b) Autodesk *Shape generator* [6].

Figure 1.1: CAD integrated topology optimization.

in November 2015, a team at University of Wisconsin-Madison released a free to use cloud-based CAD web application *CloudTopopt* [15]. The software has large-scale FE analysis and topology optimization capabilities—computations are carried out remotely on the cloud server. Krishnan Suresh, associate professor at UW-Madison and developer of the software, is of the opinion that 'design optimization lies at the heart of modern engineering' and 'is critical in reducing cost, reducing material, reducing weight and increasing quality, and is a driving force behind innovation' [15].

In September 2015, The Economist reports on the Amsterdam office of Arup—a global engineering consulting firm—who used topology optimization and 3D printing technology to create 'wonderful widgets'—'components become more elegant with software that produces the most efficient shape' [16]. Arup's task was to design 1 200 components (or 'brackets', each slightly different) to hold the support cables of a large outdoor lighting system. A artist rendering of the structure (which was, unfortunately, never built) can be seen in Figure 1.2a—the image is courtesy of Salomé Galjaard, team leader of the project. Three versions of the bracket is depicted in Figure 1.2b. The component to left (about a metre tall) was designed using traditional methods and manufactured by cutting, drilling, and finally welding sections together—the work would predominately be carried out by hand. The other two components (middle and right) were designed with topology optimization techniques and manufactured with an additive process. The middle component is optimized for weight, but the connection points are kept in much the same place—the result is a 40% weight saving. The component to right is also optimized for weight, but in this case the positions of the connection points were allowed to change—*cf.* Figure 1.2c—resulting in a 70% weight saving. The Economist writes that 'optimization software typically comes up with natural-looking shapes that seem to mimic nature' and that 'this is not surprising', because 'nature has had a few million years head start in designing structures like bones, stems and leaves'—this sentiment is reiterated in 'Alloy angels' (May, 2016) [17], a report on a 3D printed lightweight motorcycle (Figure 1.3a), designed and manufactured by APWorks, a division of the Airbus Group. Galjaard believes that optimised 3D-printed parts and components is an important avenue in the pursuit of weight, material, and cost savings, provided that these are accepted by contractors and standards authorities [16]. 'This seems to be happening', writes The Economist, reportedly, Stratasys, an American producer of 3D printers, was commissioned by Airbus to make more than a 1000 parts for the A350 airliner—the parts had 'enhanced performance', were '25% lighter', 'should

last five times longer' and 'met aerospace certification standards' [16].



(a) Outdoor light structure. (b) 'Wonderful widgets'. (c) Brackets in place.

Figure 1.2: Outdoor lighting structure and brackets, Grote Markstraat, Den Haag.

In 2015, Precision Casting Centre, a foundry on the south-coast of England, won the coveted British Cast Metals Federation *Component of the year* award for their design and manufacture of an aluminium wheel upright [18]. The team employed a thorough topology optimization process, leading to a part five times stiffer than the original (but manufactured from the same amount of material). The part is produced by 3D printing a model in transparent thermoplastic, in turn surrounded by metal to form a mould, before it is cast in aluminium.

In January 2016, Engineering.com reports on a team of engineers at Hanning Elektro-Werke (Germany) who used topology optimization to decrease mechanical losses in an electric motor—an efficiency improvement of 27% was realised [19].

In 2014, manufacturing giant Ford invested heavily in lightweighting efforts, making 'aggressive use of advanced simulation and optimization software' [20], in the drive to meet the rigorous emissions standards imposed by the Obama administration.

Structural and topology optimization is found in small scale and bespoke production environments too—*e.g.* chain cases and bulkheads in snowmobiles [21], and implants used in facial reconstructive surgery [22]. Alok Sutradhar, of Ohio State University, an aeronautical engineer and pioneer of topology optimization in craniofacial reconstruction, notes that 'when I saw how the surgeons are, at present, reshaping faces without any structural analysis, using simple heuristics instead, I thought: wait a minute, we can at least tell you where you need the materials so that the patient will have a structurally stable structure for basic functions: breathing, speaking, chewing, and swallowing' [23]. Sutradhar too finds confidence in the 'natural-looking' shapes suggested by topologically optimized designs: 'the optimization software came up with solutions akin to those proffered by evolution', which, although a 'surprise at first', 'gave us confidence too', for, 'whatever approach we were taking seemed to be making sense' [23].

In August 2016, the University of Pittsburgh received a $350 000 research grant to develop models for a topology optimization framework which can mitigate the distortions and stresses inherent in current additive manufacturing processes [24, 25]. This is a major challenge, according to Dr. To, lead researcher of the project: 'optimizing the design to compensate for residual distortion, residual stress, and post-machining requirements can take days or even months' [24, 25]. The team at the University of Pittsburgh are looking to create 'new, fast computational methods for better additive manufacturing'. 'The technology developed will significantly shorten the design phase during new [additive manufacturing] product development,

which will potentially lead to wider adoption of additive manufacturing by the manufacturing base in the US' [25]. Reportedly, since 2014, funding for additive manufacturing related research received by the University of Pittsburgh's Swanson School of Engineering totals about $7 million [24]. In the fall-out of the latest economic crisis, some forecasts value the 3D printing market at $40 billion by 2020, and some predict further growth into the $12 trillion global manufacturing industry [24]. Norsk Titanium, a company which specialises in aerospace-grade titanium structures, are looking to invest $1 billion in factories with large-scale 3D printing capabilities [24]. In July 2016, Alcoa, a major company in the metals industry, opened a $60 million 3D printing plant in Pittsburgh [26]. Norsk plan to have a similar facility operational, in New York, by the end of 2017 [27].

* * *

The complex, 'natural-looking' geometries produced by topology optimization procedures almost invariably necessitate additive manufacturing processes. Some of the geometries shown here—*e.g.* the 'widgets' in Figure 1.2b, and the 'Light Rider' in Figure 1.3a—are clearly not manufacturable with traditional production processes. (The wheel upright mentioned earlier is a notable exception, hence the accolade [18].) That is, advances in additive manufacturing techniques—*e.g.* modelling of residual distortions, stresses, and machining requirements—has the potential to fully exploit topology optimization techniques, and *vice versa*. Constraints that preserve the conventional manufacturability of the design—'shape controls', for example, in SolidThinking's *Inspire*—take the form of a limit on local geometric variations, leading to traditional, truss-like designs. Jaideep Bangal, SolidThinking's senior application engineer, warns that 'today we are going backwards', 'we are making very organic designs without these controls that are just 3D printable' [28]. There is however a trade-off between geometric optimization on the one hand, and the increased complexity and cost of manufacture on the other. Bruce Jenkins, president of Ora Research, a research and advisory services firm, advocates a system-level approach to optimization, writing that 'optimization technology offers great benefit when used early in product development to optimize the functional systems', which are, 'created and refined before geometry definition begins' [29].

(a) The Light Rider [17]. (b) Cauliflower [30].

Figure 1.3: Natural-looking designs.

The notion that a 'natural-looking' design necessarily points the way to an 'optimal' design should be approached with utmost caution. It is fair to say that many a complex geometry may invoke connotations with structures found in nature—the patterns described by fractals [30],

for example. Natural structures, like the cauliflower shown in Figure 1.3b, exhibit self-similar patterns of complexity on various length scales. Topology optimization may come up with 'natural-looking' designs, but this is predominantly due to the allowance of relatively detailed geometric variations. The observation that the geometries arrived at by topology optimization procedures often resemble natural structures is pertinent indeed, but for a different reason: it confirms that the associated differential equations—the equations of elasticity, in this case—are able to capture the character of the physical laws which govern the connectivity, interrelation and arrangement of the constituent parts of the structure. Moreover, the imposition of 'conventional manufacturability constraints'—like 'shape controls'; or 'slope constraints', see Figure 1.4—which may result in an optimal design in terms of the entire supply and manufacture chain, work by explicitly suppressing small-scale geometric variations. That is, considering a broader scope of optimization, 'more optimal' designs may look 'less natural'.

Figure 1.4: Geometric variations.

The *raison d'être* for topology optimization—and the motivation behind the pursuit of reliable and efficient methods—is its potential to produce components and structures which have improved performance characteristics and which consume resources more efficiently. In large-scale production environments, material and cost savings are paramount. In bespoke production environments, the objective is often defined in terms of the performance characteristics of the design, which may include material and weight savings. Nevertheless, as we have seen, topology optimization is typically used subsequent to the definition of the functional systems, therefore only marginal gains may be realised in a fairly narrow design scope. Moreover, especially with recent CAD integration—software used quite late in most design methodologies—topology optimization may act merely as 'translator' between the designer and the manufacturing process, confining the scope for optimization.

Taking a step back, it is perhaps prudent to keep in mind that a specialised procedure like topology optimization—which is often confined to a fairly narrow design scope—may inadvertently provide a semblance of 'sustainability' to industries and resources—the automotive industry, for example, and fossil fuels—which will have to be replaced, sooner or later, with fundamentally different and 'unconventional' alternatives. In 'The beginning of infinity' [31], David Deutsch, a quantum physicist, notes that *sustain* is an interestingly ambiguous word: on the one hand it can mean providing someone with what they need, but on the other hand it can also mean preventing something from changing; these are opposite meanings, for the suppression of change is seldom what human beings actually need. Deutsch warns that 'prevention and delaying tactics'—specialised techniques, confined to a narrow design scope, aiming at marginal, accumulative improvements—'are useful', 'but they can be no more than a minor part of a viable strategy for the future', for, 'problems are inevitable, and sooner or later survival will depend on being able to cope when prevention and delaying tactics have failed.' Deutsch adds that

> For this we need a large and vibrant research community, interested in explanation and problem-solving. We need the wealth to fund it, and the technological capacity

to implement what it discovers.

Undoubtedly, the marriage of topology optimization and additive manufacturing techniques will shape the machines and the structures of the future, but these algorithmic procedures should never be allowed to curb—or indeed, be proported to replace—the imaginative capacity of the design-engineer.

Outline

The form of this dissertation permits a somewhat unconventional introduction: Chapters 2 to 6, which form the body of the dissertation, are each a close reproduction of a self-contained research-paper. At the time of writing, some have seen publication, while others are still under review.

As the name suggests, *a direct approach to structural topology optimization* sets out to establish the alternative 'direct' ('simultaneous analysis and design'; SAND) formulation of the structural topology optimization problem—particularly, from a conventional sequential approximate optimization (SAO) point of view. Chapter 2 is an investigation into the relationship between the traditional 'reduced' ('nested analysis and design'; NAND) SAO setting—in classical minimum compliance design—and the SAO-SAND (SEASAND) equivalent. Chapter 3 serves to investigate the extent to which traditional techniques of enforced convergence and termination may be transplanted into the SEASAND method. Chapter 4 considers design-set restriction techniques, and how these manifest in the SAND setting. In Chapter 5 the issue of local stress constraints—and the infamous 'stress singularity problem'—is taken up, and in Chapter 6 the large-scale potential of the SEASAND method is put on trial. Chapter 7 closes with suggested topics for future research.

It should be mentioned that some of the work presented herein builds upon topics that have been touched on in the author's Master's thesis [32].

Chapter 2

On sequential approximate simultaneous analysis and design in classical topology optimization

This chapter is a reproduction of a paper entitled 'On Sequential approximate simultaneous analysis and design in classical topology optimization' [33]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical and Mechatronic Engineering at the University of Stellenbosch, South Africa.

2.1 Abstract

We study the SAND (simultaneous analysis and design) formulation of the 'classical' topology optimization problem subject to linear constraints on material density variables. Based on a dual method in theory, and a primal-dual method in practice, we propose a separable and strictly convex quadratic Lagrange-Newton subproblem for use in sequential approximate optimization (SAO) of the SAND formulated classical topology design problem.

The SAND problem is characterised by a large number of nonlinear equality constraints (the equations of equilibrium) which are linearized in the approximate convex subproblems. The availability of cheap second-order information is exploited in a Lagrange-Newton sequential quadratic programming (SQP)-like framework. In the spirit of efficient structural optimization methods the quadratic terms are restricted to the diagonal of the Hessian matrix; the subproblems have minimal storage requirements, are easy to solve, and positive definiteness of the diagonal Hessian matrix is trivially enforced.

Theoretical considerations reveal that the dual statement of the proposed subproblem for SAND minimum compliance design agrees with the ever-popular optimality criterion (OC) method—which is a NAND (nested analysis and design) formulation. This relates, in turn, to the known equivalence between rudimentary dual SAO algorithms based on reciprocal (and exponential) intervening variables and the OC method.

2.2 Introduction

In the *Elements of Structural Optimization* [3] Haftka, Gürdal and Kamat relate the following historical perspective: 'During the seventies people who worked in the field of structural optimization were divided into two distinct and somewhat belligerent camps'; the mathematical programming camp, who 'believed in employing the general methods of nonlinear programming', 'decried the lack of generality of optimality criteria methods', while the optimality criteria camp, 'sneered at the inefficiency of the general mathematical programming approach'.

Since that time common computational capabilities and the efficiency of sequential convex programming (SCP) and sequential approximate optimization (SAO) methods have increased. Originally SCP and SAO algorithms—*e.g* CONLIN [34, 35] and its generalisation, the 'method of moving asymptotes' (MMA) [36]—were based on an efficient dual statement due to Fleury [37]. The dual statement—the dual of Falk [38], a specialised version of the Lagrange dual—is in general invokable if a programming problem is strictly convex[1] and separable. It is widely intuited though that dual methods are too slow if a large number of active constraints are present—see *e.g.* Fleury's [40] exposition on 'computational time issues'—and although pure dual methods are successfully applied to large scale problem instances—see *e.g.* SAO*i* [41] in Reference [42] and the MMA in Reference [43]—primal-dual interior-point methods have recently emerged as a superior choice. See for example the recent benchmarking study by Rojas-Labanda and Stolpe [44]. Primal-dual interior-point methods can handle a large number of active constraints by exploiting the sparsity of the Karush-Kuhn-Tucker (KKT) system of equations and separable (sparse) curvature approximations too may be exploited in a superior way [45]. This has led the mathematical programming camp to a general Lagrange-Newton diagonal quadratic programming (QP) framework, in turn reminiscent of sequential quadratic programming (SQP) methods. See also Reference [46] for additional information.

Optimality criterion (OC) methods have an equally rich history in structural optimization. OC methods are especially popular in 'classical' (minimum compliance) topology design (constrained by a single linear constraint on volume). The OC method used mostly is due to Bendsøe [47], culminating in the well-known Matlab implementations by Sigmund and coworkers [10, 12]. When more than a single constraint is present however, OC methods are considered impractical and SAO methods are typically resorted to.

Fleury [37] has demonstrated, using reciprocal intervening variables to represent the sizing design variables, that dual SAO algorithms and OC methods are closely related in structural weight minimization [37]. In the *Elements of Structural Optimization* Haftka, Gürdal and Kamat [3] continue with an overview of the contributions by Fleury and others in relating OC and SAO methods for the minimum weight optimization problem. For minimum compliance design, Bendsøe and Sigmund [2] note that the OC method and the MMA 'in essence involves the same kind of computations'—it has since been demonstrated that an arbitrary choice of heuristic damping parameter in the OC method is *identical* to a rudimentary SAO subproblem with exponential intervening variables [48]. Exponential intervening variables in turn generalise the reciprocal case, which is identical to the OC method with the popular choice of 0.5 for the numerical damping factor [48].

In structural optimization it is customary to formulate the mathematical programming prob-

[1] The convexity requirement may actually be relaxed, sometimes [39].

lem in a 'nested analysis and design' (NAND) setting. The OC method, for example, is a NAND formulation. SAO methods, it can also be said, have predominantly had a 'NAND view' (a consequence no doubt of traditional 'simulation-based' optimization). NAND formulations are characterised by finite element (FE) function calls which relate design and state variables (or the 'simulation' part of the problem). In effect then the state variables are removed from the optimization problem by a procedure equivalent to the direct substitution of the equilibrium equations. This is especially prohibitive if higher-order derivatives (*i.e.* curvature information) is desired, which, in turn, led to the practice of employing intervening variables in the NAND setting, where curvature approximations based on intervening are much cheaper to calculate than having to conduct exact analyses. In the alternative 'simultaneous analysis and design' (SAND) setting the equations of equilibrium are constraints in the optimization problem, typically in the form of nonlinear equality constraints, and although the mathematical programming problem is much larger (in number of variables and constraints) and more complex than the equivalent NAND case, all gradient information is readily available in the form of simple, sparse, easy to calculate, partial derivatives. See Arora and Wang [49] for a monograph on the subject, and the 'state space' Newton method proposed by Evgrafov [50] in the context of Stokes flow problems.

Haftka [51] was amongst the first to study the SAND approach in structural optimization. He solved the problem with a penalty function formulation and a conjugate gradient method, demonstrating that preconditioning with an element-by-element approximate inverse of the stiffness matrix may improve convergence. Bendsøe, Ben-Tal and Haftka [52] studied and compared the SAND approach to a displacements-only formulation. Allowing for zero cross-sectional areas they demonstrate that the SAND approach avoids the nondifferentiability problem associated with singularity of the global stiffness matrix. Sankaranarayanan, Haftka and Kapania [53] studied the SAND minimum weight problem subject to stress and displacements constraints. They compare the penalty function approach to an augmented Lagrangian formulation and progressive elimination of members with small cross-sectional areas, reporting substantial computational savings.

More recently Rojas-Labanda and Stolpe [44] studied the SAND formulation of the topology optimization problem as part of a thorough benchmarking study. The study consists of a large set of test problems spanning minimum compliance, minimum volume and mechanism design problems, subject to 'simple' constraints. The SAND problems are solved with either primal-dual interior-point methods or a sequential quadratic programming (SQP) approach—due to the availability of cheap second-order information, the exact Hessian matrix could be used. The study demonstrates that the exact second-order information improves 'final objective function values' and that the formulation is 'reliable and robust'. However, inferior computation times are achieved when solving the SAND formulated problems, compared to the NAND equivalents. Reportedly some SAND problems had to be removed from the test set due to time and memory limitations—undoubtedly the calculation and storage of the full Hessian matrix contributed to this.

Herein we propose a *separable* and *strictly convex* Lagrange-Newton quadratic subproblem for SAND based minimum compliance design. Partly based on the findings by Rojas-Labanda and Stolpe [44], we argue that the full Hessian matrix is, in general, prohibitively expensive in computational storage requirements. Moreover, positive definiteness of a diagonal Hessian matrix is easily enforced and the consequential sparsity of the KKT linear system of equations—

the equations on which most of the computation time is spent [45]—can be exploited by primal-dual interior-point methods, continuing where separable dual methods left off.

In the spirit of the historical perspective cited above, we consider a relatively inefficient but rigorous mathematical programming formulation of a structural optimization problem. It is thought that the SAO-SAND approach (or 'SEASAND', as we like to say) may one day form part of a very large scale structural optimization methodology, or rather, at this point in time there are tantalizing reasons to suggest so. Therefore, herein we attempt to elucidate, for posterity, the workings of an SAO method applied to the SAND formulated problem, at the limit of common computational capabilities.

2.3 Topology optimization

Consider the 'classical' topology optimization problem [2]. The goal is to find the material distribution which minimizes the compliance of a predefined design domain. The amount of material is limited on the upper bound by a single linear volume constraint. It is assumed that the design domain is discretized with the finite element (FE) method. Each element is assigned a design variable t_i which represents the amount of material in the element. The vector of material (density) variables is denoted by $\boldsymbol{t} \in \mathbb{R}^e$, with e the number of elements. The vector of nodal displacements is denoted by $\boldsymbol{u} \in \mathbb{R}^d$, with d the number of degrees of freedom. The SAND formulation of the problem has a vector of primal optimization variables $\boldsymbol{x}$ composed of $\boldsymbol{x} = (\boldsymbol{t}, \boldsymbol{u}) \in \mathcal{R}^{n=e+d}$. Topological features are introduced with the solid isotropic material with penalization (SIMP) method, independently proposed by Bendsøe [54] and Rozvany and Zhou [55]. The SAND problem is expressed as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{u}) = \boldsymbol{r}^T\boldsymbol{u} \\ \text{subject to} \quad & [\boldsymbol{K}(\boldsymbol{t})]\boldsymbol{u} - \boldsymbol{r} = 0 \\ & f_1(\boldsymbol{t}) = \mathbf{1}^T\boldsymbol{t} - \bar{\nu} \leq 0 \\ & \mathbf{0} \leq \boldsymbol{t} \leq \mathbf{1} \end{aligned} \tag{2.1}$$

where $\boldsymbol{K}(\boldsymbol{t})$ represents the $d \times d$ globally assembled finite element stiffness matrix

$$[\boldsymbol{K}(\boldsymbol{t})] = \sum_{i=1}^{e} (t_i)^p [\boldsymbol{K}_i] \,, \tag{2.2}$$

with $p > 1$ the SIMP parameter (as is customary, we use $p = 3$ throughout). The prescribed limit on the volume fraction of the domain is denoted by $\bar{\nu}$. It is assumed that the FE's are equally sized. It is also assumed that the loads $\boldsymbol{r}$ are design independent.

The Lagrangian $\mathcal{L}$ of the SAND problem (2.1) can be expressed as

$$\mathcal{L}(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}, z) = \left\{ \left[\boldsymbol{r}^T\boldsymbol{u} - \boldsymbol{v}^T[[\boldsymbol{K}(\boldsymbol{t})]\boldsymbol{u} - \boldsymbol{r}] + z\left(\mathbf{1}^T\boldsymbol{t} - \bar{\nu}\right) \right] \,:\, \mathbf{0} \leq \boldsymbol{t} \leq \mathbf{1} \right\} \,, \tag{2.3}$$

with $\boldsymbol{v} \in \mathbb{R}^d$ the vector of Lagrangian multipliers associated with the equations of equilibrium and z the single multiplier associated the volume constraint. The negative in front of the 'strain energy density-like' term is standard—see *e.g.* Bendsøe and Sigmund's derivation of the OC

method [2] (pg.10)—and, importantly, it is inconsequential to the sign of the sensitivity information of $\mathcal{L}$ with respect to the primal variables $\boldsymbol{x}$. More on this in Section 2.7. The complete vector of dual variables is formed by $\boldsymbol{\lambda} = (\boldsymbol{v}, z) \in \mathcal{R}^{m=d+1}$. For now we consider displacement variables $\boldsymbol{u}$ unbounded.

Under fairly mild assumptions the discretized topology design problem has a nonempty feasible set [56]. Moreover, the feasible set of the relaxed and penalized problem is nonempty if the volume limit is enforced in the form of an equality constraint [56].

In reality discrete ('black-and-white') solutions are sought—*i.e.* $t_i \in [0, 1]$, $\forall\, i$—from an algorithmic point of view however the discrete programming problem is rather cumbersome—see *e.g* Wood [42]. The idea behind SIMP is that the penalty parameter $p > 1$ drives each material density variable t_i towards either 0 or 1, but because the problem is continuous, efficient gradient-based optimization methods can be used. However, since the discrete problem is intractable, the relaxed and penalised problem is intractable too; the penalization strategy itself introduces many local minima. In the SAND setting the nonconvexity of the problem is expected to be even worse, since it manifests in the form of the nonlinear equality constraints—the equations of equilibrium in the SAND problem (2.1)—which may only be satisfied at convergence.

There is in fact a number of difficulties which could potentially plague the relaxed and penalized problem (in either NAND or SAND form) [57]. In addition to multimodality (more severe in the SAND setting), the optimal solution is dependent on the mesh discretization and might exhibit an artificial (numerical) stiffening phenomena known as 'checkerboarding'. These are typically negated with restriction methods and higher order elements [57]. Popular approaches are based on the idea of a 'density filter'—for details see the well-known Matlab codes by Sigmund and co-workers [10, 12]—and although Bourdin [58] proved the method to be mathematically sound (*i.e.* not heuristic), it works by coupling large groups of density variables, which could be expensive in the SAND setting [59]. A density filter is used in the SAND setting by for example Rojas-Labanda and Stolpe [44]. Herein, in order to restrict the design set in a fairly simple (and 'sparse') way, local slope constraints as per Petersson [60] are resorted to. Slope constraints restrict the solution set by limiting local density variations. This is enforced by imposing linear constraints on the material density variables such that

$$\begin{aligned} f_j(\boldsymbol{t}) &= t_h - t_i \leq \mu\pi \,, \\ f_{j+1}(\boldsymbol{t}) &= t_i - t_h \leq \mu\pi \,, \end{aligned} \tag{2.4}$$

where h and i denote neighbouring elements, for $j = 1, 2, \ldots, b$, and b the number of inter element boundaries. The prescribed slope constraints parameter is μ and π is a mesh size parameter, proportional to the size of the elements. Slope constraints restrict the solution set by imposing a minimum length scale on the design domain via a limit on local density variations. For μ constant the optimal solution is mesh independent, see Petersson [60] for details.

Restriction methods go some way in mitigating multimodality, but additional measures are required to gain any insight whatsoever in the relative optimality of a stationary point. It is sometimes argued that the problem should first be solved with no penalization—*i.e.* $p = 1$—alleviating the prevalence of local minima, after which the penalization parameter is gradually increased. Although this has been implemented successfully in practice—by Petersson [60] no less—a measure of 'confidence' can not be attached to the potential global optimality of a local

minimum. To this end multistart methods can be used, see *e.g.* Bolton *et al.* [61].

We find it necessary here to draw the readers attention to the zero (0) lower bound on material density variables $\boldsymbol{t}$, *exactly*. This a novel property of the SAND formulated problem, permitted because the global stiffness matrix is not inverted *per se*—in the NAND setting a nonzero lower bound is required to avoid the singularity of the global stiffness matrix. In the SAND setting however, it turns out, this is inconsequential. Note that this is the case even though linear independence constraint qualification (LICQ) can not be expected to hold at a local minimizer of the SAND problem (2.1).

2.4 Optimality criterion (OC) methods

The well-known OC method is an expression of the necessary conditions of optimality—the KKT conditions—see *e.g.* Haftka, Gürdal and Kamat [3] and Bendsøe and Sigmund [2]. Let us elaborate on this for a while. The key to generate extremely efficient computational update schemes, writes Bendsøe and Sigmund [2], 'is to devise iterative methods which, for a previously computed design and its associated displacements, updates the design variables at each point (or rather at each element of a finite element discretization) independently from the updates at the other points, based on the necessary conditions of optimality'. One such condition is the equations of equilibrium, 'considered part of a function call'; for a previously computed design $\boldsymbol{t}^{\{k\}}$, the displacements $\boldsymbol{u}$ are obtained by

$$\boldsymbol{u} = [\boldsymbol{K}(\boldsymbol{t}^{\{k\}})]^{-1}\boldsymbol{r}\,, \tag{2.5}$$

making the OC method a NAND formulation. The NAND problem is expressed as

$$\begin{aligned} \min_{\boldsymbol{t}} \quad & f_0(\boldsymbol{t}) = \boldsymbol{u}^T[\boldsymbol{K}(\boldsymbol{t}^{\{k\}})]\boldsymbol{u} \\ \text{subject to} \quad & f_1(\boldsymbol{t}) = \mathbf{1}^T\boldsymbol{t} - \bar{\nu} \leq 0 \\ & \mathbf{0} < \boldsymbol{t}_{\min} \leq \boldsymbol{t} \leq \mathbf{1} \end{aligned} \tag{2.6}$$

with displacement variables $\boldsymbol{u}$ a function of the material density variables $\boldsymbol{t}$—*i.e.* $\boldsymbol{x} = (\boldsymbol{t}) \in \mathcal{R}^{n=e}$ and $\lambda = (z) \in \mathcal{R}^1$. The objective function f_0 is modified as per the adjoint method for calculating sensitivity derivatives [2]. In 'classical' minimum compliance design subject to a single linear volume constraint the derivatives of the compliance objective f_0 reduce to the particularity simple form

$$\frac{\partial f_0}{\partial t_i}(\boldsymbol{t}^{\{k\}}) = -p(t_i^{\{k\}})^{p-1}\boldsymbol{u}^T[\boldsymbol{K}_i]\boldsymbol{u}\,, \tag{2.7}$$

where $\boldsymbol{u}$ is obtained directly from (2.5) for a given $\boldsymbol{t}^{\{k\}}$. In general the adjoint method requires additional computations.

Another modification is the imposition of a nonzero lower bound $\boldsymbol{t}_{\min}$ to prevent singularity in the FE function call (2.5).

For (2.6) the OC updating scheme takes on the following form

$$t_i = \begin{cases} t_i^{\{k\}} B_i^{\eta} & \text{if} \quad \check{t}_i^{\{k\}} < t_i^{\{k\}} B_i^{\eta} < \hat{t}_i^{\{k\}} \\ \check{t}_i^{\{k\}} & \text{if} \quad t_i^{\{k\}} B_i^{\eta} \leq \check{t}_i^{\{k\}} \\ \hat{t}_i^{\{k\}} & \text{if} \quad t_i^{\{k\}} B_i^{\eta} \geq \hat{t}_i^{\{k\}} \end{cases} \tag{2.8}$$

for $i = 1, 2, \ldots, n$, and

$$\check{t}_i^{\{k\}} \leftarrow \max(t_i^{\{k\}} - \delta, t_{\min}) \,, \tag{2.9}$$

$$\hat{t}_i^{\{k\}} \leftarrow \min(t_i^{\{k\}} + \delta, 1) \,, \tag{2.10}$$

with $\delta > 0$ a prescribed move limit. The numerical damping factor $\eta < 1$ is introduced to overcome oscillatory behaviour, diminishing the sensitivity of t_i with respect to B_i, for $i = 1, 2, \ldots, n$. In Reference [48] it is demonstrated that the updating scheme (2.8) is *identical* to the use of a dual SAO procedure with exponential intervening variables—the common practice of $\eta = 0.5$ is in turn *identical* to reciprocal intervening variables.

The B_i are found via

$$B_i = -\left(\lambda \frac{\partial f_1}{\partial t_i}\right)^{-1} \left(\frac{\partial f_0}{\partial t_i}\right) \tag{2.11}$$

with $\lambda \geq 0$ the Lagrangian multiplier (although the equality does not hold if the constraint is active and binding). The multiplier is typically found using a simple bisectioning strategy, enforcing adherence of $\boldsymbol{t}$ to the volume constraint f_1. If the elements contribute equally to the volume of the topology, as is the case herein, (2.11) reduces to

$$B_i = \frac{p(t_i^{\{k\}})^{p-1}}{\lambda} \boldsymbol{u}^T [\boldsymbol{K}_i] \boldsymbol{u} \,. \tag{2.12}$$

This completes the presentation of the simple and efficient OC method for minimum compliance topology design subject to a single linear volume constraint. We reiterative that the OC method is a NAND formulation, characterised by the function call (2.5) in every iteration k. For additional implementation details the reader is referred to the cited literature and the well-known Matlab implementations [10, 12].

2.5 Sequential approximate optimization (SAO)

SAO is characterised by the construction of inexpensive analytical approximation functions $\tilde{f}_\alpha^{\{k\}}(\boldsymbol{x})$, $\alpha = 0, 1, \ldots, m$ to the objective $f_0(\boldsymbol{x})$ and constraint functions $f_j(\boldsymbol{x})$, $j = 1, 2, \ldots, m$ at successive iteration points $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, 3, \ldots$. Together the approximations $\tilde{f}_\alpha^{\{k\}}(\boldsymbol{x})$, $\alpha = 0, 1, \ldots, m$ form an approximate subproblem $P[k]$. The specific form of each approximation $\tilde{f}_\alpha^{\{k\}}(\boldsymbol{x})$ is determined by the consequential solubility of the subproblem $P[k]$. In structural optimization it is customary to use *strictly convex* and *separable* approximations, see *e.g.* the CONLIN algorithm [34, 35] and MMA [36]. The resulting subproblems (which are strictly convex and separable) are amenable to highly efficient dual statements when the number of constraints m is far less than the number of design variables n. When the number of constraints are relatively large, the performance of Lagrange-Newton diagonal quadratic subproblems in combination with primal-dual interior-point subsolvers is considered superior [46].

2.5.1 Function approximation

The simplest approximation is the linear Taylor series expansion

$$\tilde{f}_\alpha^{\{k\}}(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \nabla f_\alpha^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \,, \tag{2.13}$$

although of course, the success of SAO methods depend a great deal on the quality of the approximations. In the NAND problem (2.6) it is considered too costly to calculate higher-order derivatives, and motivated by the reciprocal-like functional dependency imposed by the FE function call (2.5), it has become standard practice to employ reciprocal intervening variables

$$y_i = \frac{1}{t_i}, \quad i = 1, 2, \ldots, n \, . \tag{2.14}$$

In terms of the optimization variables t_i of the NAND problem (2.6), the reciprocal approximation is written as

$$\tilde{f}_{R\alpha}(\boldsymbol{t}) = f_\alpha(\boldsymbol{t}^{\{k\}}) + \sum_{i=1}^{n} \left(\frac{\partial f_\alpha}{\partial t_i}(\boldsymbol{t}^{\{k\}}) \right) \left(\frac{t_i^{\{k\}}}{t_i} \right) (t_i - t_i^{\{k\}}) \, . \tag{2.15}$$

Exponential intervening variables $y_i = t_i^{a_i}$ are a generalization of reciprocal intervening variables (2.14). The unknown exponents a_i may be estimated with any number of strategies, see *e.g.* Fadel *et al.* [62] and the work on incomplete series expansions [63] and 'approximated approximations' [64]. For $a_i = -1$ the reciprocal case is recovered, identical to the popular choice of $\eta = 0.5$ in the OC update (2.8)—the exponential case is, in turn, equivalent to an adaptive numerical damping factor η. See Reference [48] for details.

Beyond the linear approximation (2.13), efficient and flexible approximations may be derived in terms of the above mentioned incomplete series expansions [63] and 'approximated approximations' approach [64]. Consider the general diagonal quadratic approximation [65] in terms of general design variables $\boldsymbol{x}$

$$\tilde{f}_\alpha^{\{k\}}(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \nabla f_\alpha^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) + \frac{1}{2} \sum_{i=1}^{n} c_{\alpha_i}^{\{k\}} (x_i - x_i^{\{k\}})^2 \, , \tag{2.16}$$

where the second-order terms $c_{\alpha_i}^{\{k\}}$, $i = 1, 2, \ldots, n$ are restricted to the diagonal Hessian terms—lending the approximation (2.16) separability. Linear functions may be represented exactly by setting $c_{\alpha_i} = 0$, $i = 1, 2, \ldots, n$. The second partial derivatives to, for example, the reciprocal approximation (2.15) (which implies a NAND context) with respect to t_i at $\boldsymbol{t}^{\{k\}}$ are obtained as

$$\frac{\partial^2 \tilde{f}_{R\alpha}}{\partial t_i^2}(\boldsymbol{t}^{\{k\}}) = -\frac{2}{t_i^{\{k\}}} \left(\frac{\partial f_\alpha}{\partial t_i}(\boldsymbol{t}^{\{k\}}) \right) \, , \tag{2.17}$$

clearly then, if (2.16) is constructed with

$$c_{\alpha_i}^{\{k\}} = -\frac{2}{t_i^{\{k\}}} \left(\frac{\partial f_\alpha}{\partial t_i}(\boldsymbol{t}^{\{k\}}) \right) \, , \tag{2.18}$$

the second-order Taylor series approximation to the reciprocal approximation at point $\boldsymbol{t}^{\{k\}}$ is obtained. Innumerable approximation strategies may be derived in this way [63]. The diagonal quadratic approximation (2.16) may in general be rather inaccurate with respect to the original function, capturing only the behaviour of the function in the immediate vicinity of $\boldsymbol{t}^{\{k\}}$—see *e.g.* the illustration of the reciprocal approximation (2.15) in Reference [65]. However, in SAO

methods it is standard practice to employ move limits anyway, and globally convergent methods typically rely on a measure of the accuracy of the approximations to adjust the trust region, like the Pareto-like filter algorithm by Fletcher *et al.* [66], or 'conservatism' [67] due to Svanberg.

Due to the availability of cheap second-order information in the SAND setting, one may as well, for example, use analytical second-order derivatives with respect to direct variables when approximating the functions of the SAND problem (2.1):

$$c_{\alpha_i}^{\{k\}} = \frac{\partial^2 f_\alpha}{\partial x_i^2}(\boldsymbol{x}^{\{k\}}), \tag{2.19}$$

for all functions $\alpha = 0, 1, \ldots, m$ and with respect to all variables $i = 1, 2, \ldots, n$. More sophisticated problem specific approximations, based *e.g.* on historical information, or perhaps sophisticated intervening variables, or a combinations of these, may turn out to be superior. Especially considering that the SAND problem (2.1) is actually linear in displacement variables $\boldsymbol{u}$.

2.5.2 Convexity, feasibility and constraint qualification of the approximate subproblem

We restrict ourselves to strictly convex subproblems, as is customary. This implies that subproblem $P[k]$ is naturally formulated as Lagrange-Newton quadratic program (QP) in the SAND setting; the nonlinear equality constraints made up by the equations of equilibrium are linearised for the subproblem to be convex in the first place[2][3]. On the other hand, because quadratic inequality constraints describe a convex feasible region if one feasible point exists, and if $P[k]$ contains only nonlinear inequality constraints (like in common NAND formulations), then the subproblem may take the form of a quadratic program with quadratic constraints (QCQP), soluble either by pure dual methods, see *e.g.* Reference [46], or in the form of a second-order cone program (SOCP) [68].

In the drive to solve problems with more and more active constraints, the SAO community has somewhat inadvertently arrived at methods that may be applicable to SAND formulated problems. Lagrange-Newton QPs are considered attractive anyway for problems with a large number of constraints [46]. The QP subproblems are characterised by the linearisation of all the constraint functions, with constraint curvature information retained as part of the diagonal quadratic objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ of the subproblem. The approximate objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ is thus viewed as an approximation of the Lagrangian and the SAO method may be termed a diagonal SQP type algorithm [46]. Hence, the general Lagrange-Newton subproblem

[2]Feasibility is of course assumed too. It turns out that this is a reasonable assumption in SAND minimum compliance design subject to a single linear volume constraint.

[3]Nonconvex QP's—that is QP subproblems with negative definite Hessians—may be used too, but the underlying theory, and the numerical routines, are typically quite involved [39].

$P_{QP}[k]$ for the SAND problem (2.1) can be expressed as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \tilde{f}_0^{\{k\}}(\boldsymbol{x}) = f_0(\boldsymbol{x}^{\{k\}}) + \nabla f_0^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}})^T \boldsymbol{Q}^{\{k\}}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \\ \text{subject to} \quad & \tilde{f}_j^{\{k\}}(\boldsymbol{x}) = f_j(\boldsymbol{x}^{\{k\}}) + \nabla f_j^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \; [=, \leq] \; 0 \,, \quad j = 1, 2, \ldots, m \\ & \check{x}_i^{\{k\}} \leq x_i - x_i^{\{k\}} \leq \hat{x}_i^{\{k\}} \,, \quad i = 1, 2, \ldots, n \end{aligned} \tag{2.20}$$

and

$$\check{x}_i^{\{k\}} \leftarrow \max(x_i^{\{k\}} - \delta(x_{i,\max} - x_{i,\min}), x_{i,\min}) \,, \tag{2.21}$$

$$\hat{x}_i^{\{k\}} \leftarrow \min(x_i^{\{k\}} + \delta(x_{i,\max} - x_{i,\min}), x_{i,\max}) \,, \tag{2.22}$$

for $i = 1, 2, \ldots, n$, with $x_{i,\min}$ and $x_{i,\max}$ the lower and upper bounds on variable i as per the global problem (2.1).

We have resorted to some nonstandard notation, for the sake of brevity, to denote both equality and inequality constraints—*i.e.* $[=, \leq]$.

The Hessian matrix $\boldsymbol{Q}^{\{k\}}$ of the approximate objective $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ contains only diagonal elements, constructed according to

$$Q_{ii}^{\{k\}} = c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}}, \quad i = 1, 2, \ldots, n \,, \tag{2.23}$$

where it is standard practice to retain the Lagrangian multiplier estimates from the previous iterate—*i.e.* $\boldsymbol{\lambda}^{\{k\}} = \boldsymbol{\lambda}^{\{k-1\}\star}$.

We will return shortly to the calculation of diagonal Hessian terms (2.23), first however, feasibility and constraint qualification (CQ) of subproblem $P_{QP}[k]$ deserves a word. Because 'technological' or 'behavioural' constraints typically take the form of limited quantities, and due to the prevalence of the NAND formulation, structural optimization and SAO literature is typically concerned with subproblems constrained only by inequalities—*i.e.* only $[\leq]$ in $P_{QP}[k]$. Importantly, inequality constraints can readily be relaxed if the feasible set of $P_{QP}[k]$ happens to be empty. Therefore, since problem $P_{QP}[k]$ possesses only (affine) linear constraints, Abadie's CQ (or 'regularity') is trivially satisfied. This is not the case for the general $P_{QP}[k]$—*i.e.* $[=, \leq]$ in $P_{QP}[k]$. Although Abadie's CQ holds[4], feasibility of $P_{QP}[k]$ is not readily attainable. A range of strategies exist to deal with infeasible instances of $P_{QP}[k]$: *e.g.* a simple backtracking strategy in conjunction with a trust region approach—see Fletcher *et al.* [66]—or the 'feasibility subproblem' due to Burke and Han [69]—see also Shen *et al.* [70]. Herein, in an attempt to restrict ourselves to the topic at hand, we employ the former. In the future we hope to implement the latter.

Therefore, assuming feasibility and in turn CQ, we only require positive definiteness of $\boldsymbol{Q}^{\{k\}}$ to end up with a strictly convex subproblem $P_{QP}[k]$. Because $\boldsymbol{Q}^{\{k\}}$ is diagonal—the

[4]Linear independence CQ is not necessary for Abadie's CQ. Indeed, with a zero lower bound on the material density variables ($\boldsymbol{t}_{\min} = 0$) linear independence CQ can not be expected hold at a local minimizer of the SAND problem (2.1).

approximate objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ is separable—this boils down to

$$Q_{ii}^{\{k\}} = c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}} \geq 0, \quad i = 1, 2, \ldots, n \,. \tag{2.24}$$

The positive definite requirement (2.24) may be enforced, for example, by modifying the individual contributions of the approximate objective and constraint functions

$$\begin{aligned} c_{0_i}^{\{k\}} &\leftarrow \max(\epsilon_0 > 0, c_{0_i}^{\{k\}}), \quad i = 1, 2, \ldots, n, \\ c_{j_i}^{\{k\}} &\leftarrow \max(0, c_{j_i}^{\{k\}}), \quad i = 1, 2, \ldots, n, \end{aligned} \tag{2.25}$$

and $j = 1, 2, \ldots, m$, or, on the other hand, by direct modification of the accumulated diagonal Hessian terms

$$Q_{ii}^{\{k\}} = \max\left(\epsilon > 0, c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}}\right), \quad i = 1, 2, \ldots, n \,. \tag{2.26}$$

In SAO parlance (2.25) is referred to as the *convex* strategy, and (2.26) is referred to as the *nonconvex* strategy—see Reference [71]—although of course both lead to a *strictly convex* $P_{QP}[k]$, the strategy in (2.26) could possibly retain some nonconvex curvature information. In addition, (2.26) is identical to (2.25) if the curvature terms $c_{0_i}^{\{k\}}$ and $c_{j_i}^{\{k\}}$ are respectively strictly positive and positive anyway. Moreover, numerical experiments in Reference [71] suggest that the *nonconvex* strategy (2.26) may have a notable computational advantage for problems characterised by nonconvex objective and/or constraint functions.

For reasons that will become clear in Section 2.7, we propose an additional strategy to enforce the positive definite requirement (2.24). The new strategy is referred to as the *absolutely nonconvex* strategy, obtained if (2.26) is modified with an absolute value operator

$$Q_{ii}^{\{k\}} = \max\left(\epsilon > 0, \left| c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}} \right|\right), \quad i = 1, 2, \ldots, n \,. \tag{2.27}$$

Clear to see is that the *absolutely nonconvex* strategy (2.27) could possibly retain severely nonconvex curvature information, which would have been discarded otherwise. This is actually not as far fetched as it may initially seem; (2.27) is similar to a common strategy in quasi-Newton methods of function minimization where second-order curvature information is modified to maintain the 'direction of descent' [72]. Interestingly, in both CONLIN [34, 35] and the MMA [36], the sign of the approximate curvature terms are controlled with a conditional ('if') statement. What is more, standard practice when using exponential (and reciprocal) intervening variables in problems that are not self-adjoint is the introduction of an absolute value operator in the approximate curvature equation, see *e.g.* Reference [65].

The cheap availability of second-order information in the SAND setting makes it rather tempting to employ the full approximate Hessian matrix instead, as Rojas-Labanda and Stolpe [44] have done. The advantages of using only separable (diagonal) second-order information, however, is clear: positive-definiteness of $\boldsymbol{Q}^{\{k\}}$ is easily enforced, $\boldsymbol{Q}^{\{k\}}$ has minimal storage requirements and, last but not least, the consequential sparsity of the KKT linear system is amenable to highly efficient primal-dual interior-point methods of continuous programming [45].

2.6 A dual statement for QP based 'classical' SAND topology design

For the sake of completeness we recite first the derivation of Falk's dual formulation [38] for the general subproblem $P[k]$. The section is closed with the dual statement of subproblem $P_{QP}[K]$ in the notation of the 'classical' SAND minimum compliance topology design problem (2.1).

The potential efficiency of the dual statement hinges on both *strict convexity* and *separability* of the primal approximate subproblem $P[k]$. These two properties mean that the saddle point $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ of the approximate Lagrangian $\mathcal{L}^{\{k\}}$ may conveniently be found using the Falk dual [37, 38]:

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x}} \{\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda}) \ : \ \check{\boldsymbol{x}} \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}}\} = \max_{\lambda} \gamma(\boldsymbol{\lambda}) \tag{2.28}$$

since the bound constraints represent a closed and bounded set—see Falk [38] for details. The dual function $\gamma(\boldsymbol{\lambda})$ can be expressed as

$$\gamma(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \left\{ \left[\tilde{f}_0^{\{k\}}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \tilde{f}_j^{\{k\}}(\boldsymbol{x}) \right] \ : \ \check{\boldsymbol{x}} \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}} \right\} . \tag{2.29}$$

If the primal approximate subproblem $P_{QP}[k]$ is *separable*, the minimization problem (2.29) reduces to n one-dimensional minimizations, which, in turn, relate primal and dual variables

$$\boldsymbol{x}^\star = \boldsymbol{x}(\boldsymbol{\lambda}) \ : \ \check{\boldsymbol{x}} \leq \boldsymbol{x}^\star \leq \hat{\boldsymbol{x}} \, . \tag{2.30}$$

If the minimization problem (2.29) is *strictly convex*, the stationary conditions

$$\frac{\partial \tilde{f}^{\{k\}}}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial \tilde{f}_j^{\{k\}}}{\partial x_i} = 0, \quad i = 1, 2, \ldots, n \tag{2.31}$$

yield simple analytical relationships which relate the primal and dual variables in relation (2.30). The saddle point $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ is then found by maximising the dual function $\gamma(\boldsymbol{\lambda})$—an m-dimensional minimization/maximization problem subject to simple bound constraints—which constitutes the *raison d'être* for the efficiency of pure dual methods for problems with a small number of constraints.

In terms of the 'classical' SAND topology design problem (2.1) the analytical relationships (2.30) can be expressed as

$$t_i(\boldsymbol{v}, z) = \prod_{t_i} \left(t_i^{\{k\}} + \frac{p(t_i^{\{k\}})^{p-1} \boldsymbol{v}^T [\boldsymbol{K}_i] \boldsymbol{u}^{\{k\}} - z}{Q_{t_i}^{\{k\}}} \right), \quad i = 1, 2, \ldots, n \, , \tag{2.32}$$

$$\boldsymbol{u}(\boldsymbol{v}) = \prod_{\boldsymbol{v}} \left(\boldsymbol{u}^{\{k\}} + [\boldsymbol{Q}_{\boldsymbol{u}}^{\{k\}}]^{-1} \left[[\boldsymbol{K}(\boldsymbol{t}^{\{k\}})] \boldsymbol{v} - \boldsymbol{r} \right] \right) . \tag{2.33}$$

The Hessian matrix $\boldsymbol{Q}^{\{k\}}$ is split into two instances $\boldsymbol{Q}_{(\cdot)}^{\{k\}}$, denoting the diagonal Hessian terms with respect to $\boldsymbol{t}$ and $\boldsymbol{u}$ respectively. Separability implies that the matrix inversions in relation (2.32) and (2.33) are trivially computed.

The $\prod_{(\cdot)}(\cdot)$ operator represents the projection onto the bounds of the material density variable i $[\check{t}_i^{\{k\}}, \hat{t}_i^{\{k\}}]$ and the vectors of displacement variable bounds $[\check{\boldsymbol{u}}^{\{k\}}, \hat{\boldsymbol{u}}^{\{k\}}]$ respectively. Here bounds on the displacement variables $[\check{\boldsymbol{u}}^{\{k\}}, \hat{\boldsymbol{u}}^{\{k\}}]$ are introduced in order to define the Falk dual, although in practice these can be chosen arbitrarily large.

2.7 The relation between OC and SEASAND methods

Let us now investigate the differences and similarities between the OC method and SAO-SAND based on the Lagrange-Newton diagonal QP subproblem $P_{QP}[k]$, in 'classical' minimum compliance topology design.

2.7.1 Equivalence

The dual update (2.32) of material density variables $\boldsymbol{t}$ reveals a similarity with the OC update (2.8) in 'classical' topology design. Consider first the OC update (2.8); because of the nonzero lower bound on material density variables $\boldsymbol{t}_{\text{min}}$ in the NAND problem (2.6) the elemental stiffness matrices $[\boldsymbol{K}_i]$ are positive definite, thus the following inequality can be derived

$$p(t_i^{\{k\}})^{p-1}\boldsymbol{u}^{\{k\},T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}} > 0\,, \tag{2.34}$$

for $i = 1, 2, \ldots, n$. Substitution of (2.34) in (2.7) reveals that the derivative of the compliance objective $\frac{\partial f_0}{\partial t_i}$ in the NAND problem (2.6) is negative always—*i.e.* $\frac{\partial f_0}{\partial t_i} < 0$. Bendsøe and Sigmund [2] put it thus: 'physical intuition is confirmed', 'additional material in any element decreases compliance, that is, makes the structure stiffer'. Substitution of (2.34) in (2.12) reveals the mechanism of the OC method: the amount of material in element i is increased if the 'strain energy density-like term' $p(t_i^{\{k\}})^{p-1}\boldsymbol{u}^{\{k\},T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}}$ is greater than the Lagrangian multiplier $\lambda = z$—the dual variable related to the volume constraint f_1, the only constraint in the NAND problem (2.6)—and decreased if the strain energy term is less than λ.

On the other hand, consider the material density dual update (2.32) in the SAND problem (2.1). Clearly the material in element i is increased if

$$p(t_i^{\{k\}})^{p-1}\boldsymbol{v}^{T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}} > z\,, \tag{2.35}$$

and decreased if

$$p(t_i^{\{k\}})^{p-1}\boldsymbol{v}^{T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}} < z\,. \tag{2.36}$$

In this case the 'strain energy density-like term' is a function of both displacement fields: the primal displacement variables $\boldsymbol{u}$ and the Lagrangian multipliers $\boldsymbol{v}$ of the equations of equilibrium. In the OC method $\boldsymbol{v}$ is the vector of adjoint variables and $\boldsymbol{v} = \boldsymbol{u}^{\{k\}}$ in every iteration k—*i.e.* the NAND problem (2.6) is self-adjoint.

Now, consider the stationary conditions which describe a saddle point of the Lagrangian $\mathcal{L}$, as defined in equation (2.3)—*i.e.* the Lagrangian of the SAND problem (2.1). The stationary condition with respect to $\boldsymbol{u}$ yields

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}(\boldsymbol{t}^\star, \boldsymbol{u}^\star, \boldsymbol{v}^\star, z^\star) = -[[\boldsymbol{K}(\boldsymbol{t}^\star)]\boldsymbol{v}^\star - \boldsymbol{r}] = \boldsymbol{0}\,, \tag{2.37}$$

and, with respect to $\boldsymbol{v}$, yields

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}}(\boldsymbol{t}^\star, \boldsymbol{u}^\star, \boldsymbol{v}^\star, z^\star) = [\boldsymbol{K}(\boldsymbol{t}^\star)]\boldsymbol{u}^\star - \boldsymbol{r} = \boldsymbol{0}\,. \tag{2.38}$$

Note that if the Lagrangian $\mathcal{L}$ is defined with a positive 'strain energy density-like' term, then the 'virtual' load vector $\boldsymbol{r}$ in (2.37) changes sign, in turn, the 'virtual' displacement vector $\boldsymbol{v}$ has to change sign, negating the sign change with respect to the complete 'strain energy density-like' term. The 'real' displacement vector $\boldsymbol{u}$ has to satisfy the 'real' equations of equilibrium in (2.38), regardless.

If every variable in the displacement vector $\boldsymbol{u}^{\{k\}}$ remains unbounded for all k ($\boldsymbol{v}$ is unbounded) , then a (feasible) local minimizer $(\boldsymbol{t}^\star, \boldsymbol{u}^\star)$ will satisfy (2.37) and (2.38). In practice this is achieved simply by setting the displacement variable bounds $[\check{\boldsymbol{u}}, \hat{\boldsymbol{u}}]$ at sufficiently large values, dependent on the scaling of the problem. Therefore, because the elemental stiffness matrices $[\boldsymbol{K}_i]$ are positive semidefinite (in SAND), at or close to a local minimum $(\boldsymbol{t}^\star, \boldsymbol{u}^\star)$ the 'strain energy density-like' term in the material density dual update (2.32) is

$$p(t_i^{\{k\}})^{p-1}\boldsymbol{v}^{\star,T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}} \geq 0\,, \tag{2.39}$$

similar to the 'strain energy density-like' in the sensitivity of compliance objective f_0 (2.34) in the NAND problem (2.6). Next we derive the sign of the separable curvature information, using the relation in (2.39).

2.7.2 Separable curvature information

Consider the diagonal second derivative of the Lagrangian $\mathcal{L}$ defined in (2.3) with respect to t_i, evaluated at a (feasible) local minimum $(\boldsymbol{t}^\star, \boldsymbol{u}^\star)$:

$$\frac{\partial^2 \mathcal{L}}{\partial t_i^2}(\boldsymbol{t}^\star, \boldsymbol{u}^\star, \boldsymbol{v}^\star, z^\star) = -p(p-1)(t_i^\star)^{p-2}\boldsymbol{v}^{\star,T}[\boldsymbol{K}_i]\boldsymbol{u}^\star \leq 0\,. \tag{2.40}$$

In conjunction with the relation in (2.39) one can see that the separable curvature with respect to a material variable $t_i > 0$ is *negative* at or close to a local minimum $(\boldsymbol{t}^\star, \boldsymbol{u}^\star)$ of the SAND problem (2.1). Thus, the diagonal second-order information of the Lagrangian (2.3) is *concave* with respect to $\boldsymbol{t}$ at (and when approaching) a local minimizer $(\boldsymbol{t}^\star, \boldsymbol{u}^\star)$ of the SAND problem (2.1).

We do the same for the compliance objective f_0 in the NAND problem (2.6). The second-order derivative of the compliance objective f_0 at $\boldsymbol{t}^k$ can expressed as

$$\frac{\partial^2 f_0}{\partial t_i^2}(\boldsymbol{t}^{\{k\}}) = -p(p-1)(t_i^{\{k\}})^{p-2}\boldsymbol{u}^{\{k\},T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}} < 0\,, \tag{2.41}$$

which, in light of the positive definiteness of $[\boldsymbol{K}_i]$ in NAND, shows that the curvature of f_0 is *concave* too with respect to direct variables $\boldsymbol{t}$. The substitution of reciprocal intervening variables (2.14) effectively convexifies the curvature information, specifically in the self-adjoint case—*i.e.* 'classical' NAND topology design subject to a single volume constraint. To illustrate

this, consider the substitution of equation (2.7) in the quadratic approximation to the reciprocal approximation (2.18), this yields

$$c_{0_i}^{\{k\}} = -\frac{2}{t_i^{\{k\}}}\left(\frac{\partial f_0(\boldsymbol{t}^{\{k\}})}{\partial t_i}\right) = 2p(t_i^{\{k\}})^{p-2}\boldsymbol{u}^{\{k\},T}[\boldsymbol{K}_i]\boldsymbol{u}^{\{k\}}\,, \tag{2.42}$$

because the elemental stiffness matrix $[\boldsymbol{K}_i]$ is positive definite in the NAND setting, the curvature quantity in (2.42) is strictly positive too.

See also the recently published 'discussion' by Evgrafov [73].

In the SAND setting the relation expressed in (2.40) is somewhat catastrophic—keep in mind that we plan to use analytical direct second-order derivatives in the Lagrange-Newton QP subproblem $P_{QP}[k]$. That is, since the problem is linear in displacement variables $\boldsymbol{u}$, using *e.g.* the *nonconvex* strategy (2.26) in order to enforce positive definiteness of $\boldsymbol{Q}^{\{k\}}$, at or close to a local minimum $(\boldsymbol{t}^\star, \boldsymbol{u}^\star)$ it can be guaranteed that all the diagonal elements in $\boldsymbol{Q}^{\{k\}}$ will be replaced with the strict convexity parameter ϵ. In effect then little or no second-order information actually enters $P_{QP}[k]$, especially when the algorithm is close to a local minimum, where it might, in fact, be needed most. This is the motivation behind the proposed *absolutely nonconvex* strategy (2.27).

How can this be though? Should one not expect positive (convex) curvature at or close to a local minimum of a problem—this is in fact a sufficient condition for a local minimum—sufficient yes, but not necessary. Lets throw away the volume constraint f_1 and consider a hypothetical material, permitted to take on values of infinite density $t_i \to \pm\infty$, for $i = 1, 2, \ldots, e$. The Lagrangian of this problem can be expressed as

$$\mathcal{L}(\boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}, z) = \boldsymbol{r}^T\boldsymbol{u} - \boldsymbol{v}^T[[\boldsymbol{K}(\boldsymbol{t})]\boldsymbol{u} - \boldsymbol{r}]\,, \tag{2.43}$$

the first derivative of (2.43) with respect to t_i is

$$\frac{\partial \mathcal{L}}{\partial t_i} = -p(t_i^{\{k\}})^{p-1}\boldsymbol{v}^T[\boldsymbol{K}_i]\boldsymbol{u}\,. \tag{2.44}$$

Except for the trivial case where one or both of the displacement vectors ($\boldsymbol{u}$ or $\boldsymbol{v}$) have all zero entries—*i.e.* an unloaded structure $\boldsymbol{r} = 0$—the Lagrangian (2.44) has a single stationary point with respect to t_i, at $t_i = 0$. That is, no stationary points exist for $t_i \neq 0$. The Lagrangian (2.43) may be equated to total potential energy of the structure; the sum of virtual work and the stored elastic energy $\boldsymbol{r}^T\boldsymbol{u}$. The structure is in equilibrium if the internal virtual work and the external virtual work is equal for an arbitrary virtual displacement $\boldsymbol{v}$—the stationary conditions of (2.43) with respect to $\boldsymbol{v}$ is the equilibrium equation. In order to minimize the compliance of the structure, we require the total potential energy (2.44) to be stationary with respect to the real displacements $\boldsymbol{u}$, the virtual displacements $\boldsymbol{v}$, and minimized with respect to t_i; (2.44) tells us that, in the direction of positively infinite density $t_i \to \infty$, the total potential energy—the Lagrangian (2.43)—may be minimized *ad infinitum*. In addition, for a SIMP density penalization of $p > 2$, the derivative of the total potential energy (2.44) changes exponentially relative to t_i, hence its concavity.

In the actual problem (2.1) local minima are constrained by the volume limit f_1 and the bounds on $\boldsymbol{t}$, hence the actual Lagrangian, as defined in (2.3), need not be convex with respect to $\boldsymbol{t}$ at a local minimum.

2.7.3 Numerical damping

The manner in which positive definiteness is enforced—*viz. convex* (2.25), *nonconvex* (2.26) or *absolutely nonconvex* (2.27)—seemingly trivializes the quantities that end up in the Hessian $\boldsymbol{Q}^{\{k\}}$ of the subproblem $P_{QP}[k]$. If, for example, the *absolutely nonconvex* strategy is used, then the quantities in $\boldsymbol{Q}^{\{k\}}$ may bear little resemblance to the actual curvature of the Lagrangian (2.3). Moreover, the SAND problem (2.1) is linear in displacement variables $\boldsymbol{u}$. Therefore, due to the strict convexity requirement, $\boldsymbol{Q}^{\{k\}}_{u_i} = \epsilon$ for $i = 1, 2, \ldots, d$ and all $k = 1, 2, 3, \ldots$.

One can however imagine that, for relatively large values of the strict convexity parameter ϵ, oscillatory behaviour may be 'damped' to some extent, similar to the effect of the numerical damping factor η in the OC update (2.8). Let us elaborate on this. The damping factor diminishes the sensitivity of $t^{\{k+1\}}$ with respect to the 'strain energy density-like' term (2.34). One can see that the diagonal Hessian terms $Q^{\{k\}}_{ii}$, $i = 1, 2, \ldots, n$ has the same effect in the dual updates (2.32) and (2.33). In material density variables the strict convexity parameter ϵ together with the strategy used to enforce positive definiteness of $\boldsymbol{Q}^{\{k\}}$ has the effect of an 'adaptive damping factor', similar to the equivalence demonstrated pertaining to exponential intervening variables and an 'adaptive numerical damping factor' in the NAND setting [48]. This is also the reason why negative definite curvature information (*i.e.* nonconvex QP's) is not expected to yield good algorithmic convergence properties, since the updates in the design variables will not be 'damped' by the curvature information (the minimum of the subproblem will often be on the variable bounds), but this should be tested in the future.

Various strategies exist to determine the unknown exponents a_i of the exponential approximation in the NAND setting. Similarly, in the SAND setting, innumerable strategies can be devised to enforce positive definiteness of the Hessian matrix $\boldsymbol{Q}^{\{k\}}$ of subproblem $P_{QP}[k]$, or to estimate the curvature terms $c^{\{k\}}_{j_i}$, $i = 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, m$ in the first place. As already pointed about, historical information (previously visited points) might be usable in order to capture and incorporate off-diagonal interactions in the diagonal terms, possibly yielding more appropriate curvature approximations (especially with respect to the displacement variables $\boldsymbol{u}$).

2.7.4 Additional constraints

From a computational point of view, due to inherent sparsity and the fact that gradient information is simple to calculate, the inclusion of additional constraints in the SAND problem (2.1) is fairly inconsequential. However, displacement-based constraints—which are the type of constraints which require additional FE function calls (2.5) in the NAND problem (2.6)—affects the relations pertaining to the 'strain energy density-like' term (2.39). In the future we hope to investigate constraints of this form—specifically local stress constraints.

On the other hand, density-based constraints—*i.e.* dependent on only $\boldsymbol{t}$—can be incorporated without substantially affecting the relations derived above, especially if these are linear. For illustrative purposes, consider the inclusion of a single slope constraint (2.4), limiting the difference between, say, t_1 and t_2—*i.e.* $t_1 - t_2 \leq \mu\pi$. It is clear that only the stationary conditions with respect to t_1 and t_2 are effected. For example, the dual update (2.32) for variable t_1

is modified to

$$t_1(\boldsymbol{v}, z, s) = \prod_{t_1} \left(t_1^{\{k\}} + \frac{p(t_1^{\{k\}})^{p-1} \boldsymbol{v}^T [\boldsymbol{K}_1] \boldsymbol{u}^{\{k\}} - z - s}{Q_{t_1}^{\{k\}}} \right) , \tag{2.45}$$

with s the Lagrangian multiplier of the slope constraint—the Lagrangian (2.3) and hence the dual function (2.29) are modified too to arrive at (2.45). In this case the 'strain energy density-like' term (2.39) has to be greater than $z + s$ for material in element 1 to be increased. For example, for both the volume constraint and the slope constraint inactive—*i.e.* $z = 0$ and $s = 0$—one can see the algorithm will increase the amount of material in element 1, in turn, reducing the compliance of the structure. Innumerable constraints of this form can be included in the SAND problem (2.1) without affecting the substance of the arguments outlined above.

OC methods have mostly been developed for problems with a single constraint. SAO subproblems on the other hand easily provide for any number of additional constraints. In the NAND setting density-based constraints do not require additional FE functions calls (2.5), but a multidimensional maximization/minimization procedure is required to determine the Lagrange multipliers $\boldsymbol{\lambda}$. A suitable example is conjugate gradient methods, as used in CONLIN [34, 35] and the MMA [36].

2.8 Numerical demonstrations and experiments

Here various properties of the Lagrange-Newton diagonal quadratic subproblem $P_{QP}[k]$ is investigated in SAO of the SAND problem (2.1). Because $P_{QP}[k]$ is convex, the solution method—*e.g.* dual *vs.* primal-dual—is inconsequential in terms of the convergence properties of the algorithm.

2.8.1 Diagonal curvature information and numerical damping

In order to demonstrate the assertions made in Section 2.7, we consider first a small and simple test problem; 'classical' minimum compliance design of the rectangular design domain $l_1 \times l_2$ depicted in Figure 2.1; F is a load distributed over l_3, the stiffness E of solid material is unity and the amount of material is limited to a fraction $\bar{\nu} = 0.5$ of the design domain. The test problem is often referred to in terms of the expected result; a 2-bar truss. The domain is discretized with 30×10 square Q8 finite elements. In this form the SAND problem (2.1) has $n = e + d = 300 + 1962 = 2262$ design variables and $m = d + 1 = 1963$ constraints. The equivalent NAND problem would have $n = e = 300$ design variables and a single constraint, but one additional $d \times d$ linear system is solved in each iteration. The displacement variables on the clamped-in boundary are fixed at 0 (derivatives with respect to these are zeroed too).

A rudimentary SAO procedure (iterations accepted unconditionally) based on the Lagrange-Newton diagonal quadratic subproblem $P_{QP}[k]$ is applied to the SAND problem (2.1). Material density variables $\boldsymbol{t}$ are limited between and including 0 and 1. Displacement variables $\boldsymbol{u}$ are bounded at $\pm 1 \times 10^6$. The move limit δ is set at 0.2 and applied as per (2.21) and (2.22); in displacement variables $\boldsymbol{u}$ this translates to a move limit of $0.2 \times (1 \times 10^6 - (-1 \times 10^6)) = 1 \times 10^5$, which has no effect. Material density variables are initialized with $t_i^{\{0\}} = \bar{\nu}$, for $i = 1, 2, \ldots, e$

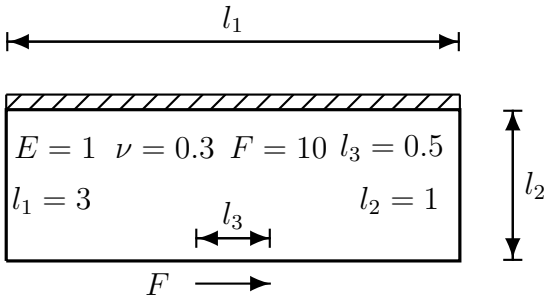


Figure 2.1: Rectangular design domain (The 2-bar truss design problem).

and displacement variables with $u_i^{\{0\}} = 0$ for $i = 1, 2, \ldots, d$. The strict convexity parameter is set $\epsilon = 1 \times 10^{-6}$ and all curvature terms are obtained as analytical partial derivatives (2.19). The algorithm is terminated if either the Euclidean norm

$$\Delta^{\{k\}} = \left\| \boldsymbol{x}^{\{k\}\star} - \boldsymbol{x}^{\{k\}} \right\| \leq 1 \times 10^{-3} \,, \tag{2.46}$$

or at a prescribed maximum number of iterations.

First we run the algorithm with the *nonconvex* strategy (2.26) to enforce positive definiteness of the diagonal Hessian matrix $\boldsymbol{Q}^{\{k\}}$. In general the *nonconvex* strategy is expected to work well for problems characterised by nonconvex functions [71]. In this case however, the algorithm fails to satisfy the convergence tolerance and is terminated after 99 iterations. In Table 2.1 the statistics of iterations 95-99 are summarised; $\Theta^{\{k\}}$ is the maximum constraint violation, $d_0^{\{k\}}$ and $d_1^{\{k\}}$ are the number of material density variables on the lower (0) and upper (1) bound respectively, $\Delta^{\{k\}}$ is calculated as per (2.46) and $\boldsymbol{H}^{\{k\}}$ is the diagonal matrix of analytical curvature terms *before* the *nonconvex* strategy (2.26) is imposed, in other words

$$H_{ii}^{\{k\}} = \frac{\partial^2 f_0}{\partial x_i^2}(\boldsymbol{x}^{\{k\}}) + \sum_{j=1}^{m} \lambda_j^{\{k\}} \frac{\partial^2 f_j}{\partial x_i^2}(\boldsymbol{x}^{\{k\}}), \quad i = 1, 2, \ldots, n \,. \tag{2.47}$$

Table 2.1: Oscillation of rudimentary SAO with $\epsilon = 1 \times 10^{-6}$ and the nonconvex strategy.

k	$f_0(\boldsymbol{u}^{\{k\}})$	$\Theta^{\{k\}}$	$d_0^{\{k\}}$	$d_1^{\{k\}}$	$\Delta^{\{k\}}$	$\max(\boldsymbol{H}^{\{k\}})$	$\min(\boldsymbol{H}^{\{k\}})$
...	...	...	...	...	...	...	...
95	9.683D+1	4.165D-2	149	146	1.256D+0	0.000D+0	-1.052D+1
96	9.683D+1	2.982D-2	149	148	1.256D+0	0.000D+0	-1.051D+1
97	9.683D+1	4.165D-2	149	146	1.256D+0	0.000D+0	-1.049D+1
98	9.683D+1	2.982D-2	149	148	1.256D+0	0.000D+0	-1.046D+1
99	9.683D+1	4.165D-2	149	146	1.256D+0	0.000D+0	-1.044D+1

The statistics in Table 2.1 confirm some of our suspicions. First, the analytical diagonal curvature terms are indeed negative close to a local minimum[5]. Consequently, the subproblem

[5]First-order accuracy of $P_{QP}[k]$ with respect to the SAND problem (2.1) permits us to assume that the algorithm is close to a local minimum.

$P_{QP}[k]$ has diagonal Hessian terms $Q_{ii}^{\{k\}} = \epsilon$, $i = 1, 2, \ldots, n$, for $k = \ldots, 95, 96, 97, 98, 99$. We can report that positive curvature information last entered $P_{QP}[k]$ in iteration number 10. The lack of curvature in $P_{QP}[k]$ seemingly results in oscillatory behaviour close to the local minimum, evidenced by the values of $\Theta^{\{k\}}$, $d_1^{\{k\}}$ and $\Delta^{\{k\}}$. The design $\boldsymbol{t}^{\{k\}}$ in iteration 99 is actually slightly asymmetric, as seen by close inspection of Figure 2.2a.

The same runs are conducted with the *convex* (2.25) and *absolutely nonconvex* (2.27) strategies to enforce positive definiteness of the diagonal Hessian matrix $\boldsymbol{Q}^{\{k\}}$, yielding the designs depicted in Figure 2.2b and Figure 2.2c. Statistics at termination are summarised in Table 2.2. Note that in this case the actual subproblem curvatures $\boldsymbol{Q}^{\{k\}}$ are given. We see that very large curvature terms enter $P_{QP}[k]$ if the *convex* strategy is used, prohibiting the algorithm from making almost any progress whatsoever. With the *absolutely nonconvex* strategy on the other hand, the algorithm terminates successfully after 19 iterations; seemingly artificially 'convexified' second-order information may aid convergence.

(a) (b) (c)

Figure 2.2: Topologies generated for the 2-bar truss design problem; rudimentary SAO with $\epsilon = 1 \times 10^{-6}$ and the (a) *nonconvex* strategy (b) *convex* strategy and (c) *absolutely nonconvex* strategy.

Table 2.2: Final iteration of rudimentary SAO with $\epsilon = 1 \times 10^{-6}$ and the (b) *convex* strategy and (c) *absolutely nonconvex* strategy.

	k	$f_0(\boldsymbol{u}^{\{k\}})$	$\Theta^{\{k\}}$	$d_0^{\{k\}}$	$d_1^{\{k\}}$	$\Delta^{\{k\}}$	$\max(\boldsymbol{Q}^{\{k\}})$	$\min(\boldsymbol{Q}^{\{k\}})$
(b)	99	4.527D+2	1.506D-5	4	0	4.227D-1	2.236D+5	1.000D-6
(c)	19	9.633D+1	3.200D-11	150	150	3.740D-4	1.044D+1	1.000D-6

We repeat all three runs but with the strict convexity parameter set at $\epsilon = 1 \times 10^{-3}$. (Up to now we had $\epsilon = 1 \times 10^{-6}$.) The results are summarised in Table 2.3. As expected, little change is observed with regard to the poor convergence properties of the *convex* strategy. However, for $\epsilon = 1 \times 10^{-3}$ the SAO algorithm with the *nonconvex* strategy not only converges, but outperforms the *absolutely nonconvex* variant—although the number of 'black-and-white' elements are fewer.

These experiments confirm some of the assertions made in Section 2.7. First, the analytical diagonal curvature information of the Lagrangian (2.3) is negative (concave) close to and at a local minimum of the SAND problem (2.1). And second, the strict convexity parameter and the positive definiteness strategy play the role of the numerical damping factor η used in the OC update (2.8), which, in turn, is equivalent to strategies pertaining to exponential intervening variables in SAO and the NAND setting [48].

Table 2.3: Final iteration of rudimentary SAO with $\epsilon = 1 \times 10^{-3}$ and the (a) *nonconvex* strategy (b) *convex* strategy and (c) *absolutely nonconvex* strategy.

	k	$f_0(\boldsymbol{u}^{\{k\}})$	$\Theta^{\{k\}}$	$d_0^{\{k\}}$	$d_1^{\{k\}}$	$\Delta^{\{k\}}$	$\max(\boldsymbol{Q}^{\{k\}})$	$\min(\boldsymbol{Q}^{\{k\}})$
(a)	33	9.633D+1	6.371D-8	123	149	6.413D-4	8.278D-3	1.000D-3
(b)	99	4.529D+2	1.116D-4	4	0	8.033D+1	2.227D+5	1.000D-3
(c)	80	9.634D+1	1.597D-8	149	144	6.630D-4	1.077D+1	1.000D-3

It is quite difficult to definitively suggest an optimal combination of ϵ and positive definiteness strategy in general—alas, this is the nature of heuristic procedures. However, based on the data in Table 2.3, it is fair to say that the *convex* approximation strategy (2.25) does not lead to good convergence properties. And although the the value of ϵ is dependent on the scaling of the problem, since we are approximating some zero terms in the diagonal Hessian matrix—the separable curvatures with respect to displacement variables $\boldsymbol{u}$—with nonzero values ϵ, it is natural to choose ϵ as small as possible. Seemingly a smaller value of ϵ leads to solutions with greater 'black-and-white' fractions.

2.8.2 The Messerschmitt-Bölkow-Blohm (MBB) beam

Next, in order to suss out the convergence and computational properties of the positive definiteness strategies and the subproblem $P_{QP}[k]$, we consider a large scale instance of the well-known MBB design domain depicted in Figure 2.3; F is a point load of unity, the stiffness E of solid material is scaled to 100—to facilitate comparisons with other works, for example Petersson [60], the objective function value herein is reported as $100 \times f_0(\boldsymbol{u}^{\{k\}})$. The amount of material is limited to a fraction $\bar{\nu} = 0.5$ of the design domain. The domain is discretized with 150×50 square Q8 finite elements. The discretization is for half the beam, due to symmetry. In this form the SAND problem (2.1) has $n = e + d = 7\,500 + 45\,802 = 53\,302$ design variables and $m = d + 1 = 45\,803$ constraints. The NAND problem will have $n = e = 7\,500$ design variables, a single constraint, but a $d \times d$ linear system is solved in each iteration.

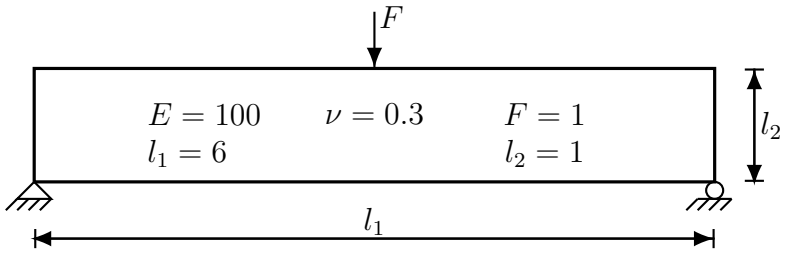


Figure 2.3: The MBB beam design domain.

First the rudimentary SAO algorithm (iterations accepted unconditionally), based on the Lagrange-Newton diagonal quadratic subproblem $P_{QP}[k]$, is applied to the SAND problem (2.1). Again, material density variables $\boldsymbol{t}$ are limited between and including 0 and 1. Displacement variables $\boldsymbol{u}$ are bounded at $\pm 1 \times 10^6$. The move limit δ is retained at 0.2. As before,

material density variables are initialized with $t_i^{\{0\}} = \bar{\nu}$, for $i = 1, 2, \ldots, e$ and displacement variables with $u_i^{\{0\}} = 0$ for $i = 1, 2, \ldots, d$. The strict convexity parameter is set $\epsilon = 1 \times 10^{-6}$ and all curvature terms are obtained as analytical partial derivatives (2.19). The algorithm is terminated if $\Delta^{\{k\}} \leq 1 \times 10^{-2}$. In the light of the results above, we test only the *nonconvex* (2.26) and *absolutely nonconvex* (2.27) strategies to enforce positive definiteness of the diagonal Hessian matrix $\boldsymbol{Q}^{\{k\}}$. The results are depicted in Figure 2.4. The SAO procedure with the *nonconvex* strategy encounters a infeasible subproblem in iteration number 10, terminating the algorithm. The variant with the *absolutely nonconvex* strategy satisfies the convergence criteria after 54 iterations.

(a) k = 10, $f_0(\boldsymbol{u}^{\{k\}})$ = -3502.6, $\Theta^{\{k\}}$ = 1.1D+1, $d_0^{\{k\}}$ = 1197, $d_1^{\{k\}}$ = 1875.

(b) k = 54, $f_0(\boldsymbol{u}^{\{k\}})$ = 203.4, $\Theta^{\{k\}}$ = 9.6D-7, $d_0^{\{k\}}$ = 3743, $d_1^{\{k\}}$ = 3683.

Figure 2.4: Topologies generated for the MBB design problem; rudimentary SAO with the (a) *nonconvex* strategy and (b) *absolutely nonconvex* strategy.

In order to deal with potential infeasibility of subproblem $P_{QP}[k]$, evidently because too much material is removed from the design domain in areas required to connect the load to the fixing position, we employ the convergent 'trust region filter' algorithm by Fletcher *et al.* [66]. A simpler version of the algorithm can be found in Reference [74]. The idea is that the feasibility of the sequence of points $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, 3, \ldots$ may be maintained, to some extent, by the conditional acceptance of iterates, pending the satisfaction of descent and feasibility measures [66, 74]. The various parameters are set at values in agreement with those suggested in Reference [66]—details can be found in Reference [75]. In the event of an infeasible subproblem $P_{QP}[k]$, which may occur nevertheless, a simple restoration procedure is activated; the current point $\boldsymbol{x}^k$ is included in the 'filter', the trust region is enlarged by setting the move limit to $\delta = 1$ and the algorithm is restored to the previous (feasible) subproblem $P_{QP}[k-1]$. Fletcher *et al.* [66] employ a more sophisticated restoration procedure—see also Shen *et al.* [70] for another example.

The move limit is initialised with $\delta = 1$. The results of the convergent SAO method with the *nonconvex* and *absolutely nonconvex* strategies are depicted in Figure 2.5. The total number of subproblems $P_{QP}[k]$ the computational platform was tasked with is denoted by P; a measure of total computational effort, although the time actually spent in solving the subproblems has to be taken into account too, as we do below. In this case both versions of the SAO procedure converges to a feasible local minimum. The *absolutely nonconvex* variant did so with less computational effort, although of course; different local minima are found.

In an attempt to mitigate the severe multimodality of the problem linear slope constraints (2.4) as per Petersson [60][6] are imposed. For the 150×50 discretization the number of inter element boundaries is $b = 14\,800$, equating to $29\,600$ additional constraints. Therefore, the

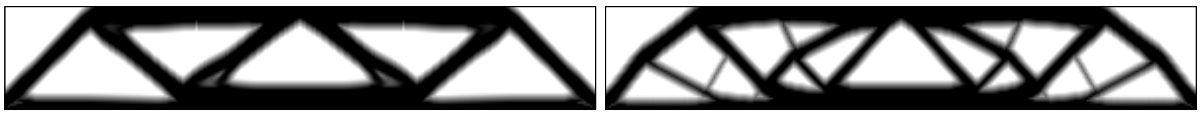
[6]Petersson [60] considered the NAND problem, solved with MMA, and used continuation on p to approach the global minimum. It is thought that the NAND problem is not as multimodal as the SAND equivalent.

(a) k = 133, P=274,
$f_0(\boldsymbol{u}^{\{k\}})$ = 210.9, $\Theta^{\{k\}}$ = 2.3D-7,
$d_0^{\{k\}}$ = 3741, $d_1^{\{k\}}$ = 3686.

(b) k = 74, P=136,
$f_0(\boldsymbol{u}^{\{k\}})$ = 202.7, $\Theta^{\{k\}}$ = 8.3D-7,
$d_0^{\{k\}}$ = 3735, $d_1^{\{k\}}$ = 3619

Figure 2.5: Topologies generated for the MBB design problem; convergent SAO with the (a) *nonconvex* strategy and (b) *absolutely nonconvex* strategy.

slope constrained SAND problem—*i.e.* problem (2.1) with b additional constraints (2.4)—has $n = e+d = 7\,500+45\,802 = 53\,302$ design variables and $m = d+b+1 = 75\,403$ constraints—which is, it should be said, a fairly large optimization problem (in NAND $n = e = 7\,500$ and $m = b + 1 = 14\,801$). The slope constraint value is set at $\mu\pi = \frac{30}{150}$, which corresponds to $c = 30$ in Petersson's notation [60]. The algorithm is terminated if $\Delta^{\{k\}} \leq 1 \times 10^{-1}$. The results are depicted in Figure 2.6. Again, the *absolutely nonconvex* variant converged to a local minimum with less computational effort, although again; different local minima are obtained.



(a) k = 248, P=507,
$f_0(\boldsymbol{u}^{\{k\}})$ = 214.5, $\Theta^{\{k\}}$ = 3.7D-6,
$d_0^{\{k\}}$ = 2873, $d_1^{\{k\}}$ = 2688.

(b) k = 127, P=241,
$f_0(\boldsymbol{u}^{\{k\}})$ = 217.4, $\Theta^{\{k\}}$ = 6.2D-6,
$d_0^{\{k\}}$ = 2529, $d_1^{\{k\}}$ = 2434

Figure 2.6: Topologies generated for the slope constrained MBB design problem with $\mu\pi = \frac{30}{150}$; convergent SAO with the (a) *nonconvex* strategy and (b) *absolutely nonconvex* strategy.

In order to facilitate a fair comparison between the *nonconvex* and *absolutely nonconvex* positive definiteness strategies, we employ a multistart procedure with random initial designs $0 \leq t_i^{\{0\}} \leq 1$, for $i = 1, 2, \ldots, d$. We do not randomize the initial displacement vector $\boldsymbol{u}^{\{0\}}$—numerical experiments indicate that the scope of the local minima actually found is not expanded a great deal with random starts in $\boldsymbol{u}$, partly because the starting position is often severely infeasible and hence untenable. A total of 200 slope constrained SAND problems (2.1) are initialized with every element in every instance of $\boldsymbol{t}^{\{0\}}$ randomized. The random number generator is seeded with the system clock each time a new instance of the problem is initialized. We can report that an interesting variety of local minima (topologies) are obtained, a detailed presentation of which is, alas, beyond the scope of this paper.

The convergent SAO algorithm is used on all 200 problem instances. One hundred are run with *nonconvex* variant and one hundred with the *absolutely nonconvex* variant. The subproblems are solved with the GUROBI Barrier QP optimizer [76]—a primal-dual interior-point method. Computational capabilities are provided by the Rhasatsha HPC [77]. The solution with the lowest objective function value $f_0(\boldsymbol{u}^{\{k\}})$ is depicted in Figure 2.7. The objective function value $f_0(\boldsymbol{u}^{\{k\}})$ and the topology is in agreement with those presented by Petersson [60]. In-

Figure 2.7: Optimal topology generated with convergent SAO of the slope constrained MBB design problem with $\mu = 30$ and 200 random starts $\mathbf{0} \leq \boldsymbol{t}^{\{k\}} \leq \mathbf{1}$: k = 286, P=600, $f_0(\boldsymbol{u}^{\{k\}})$ = 212.7, $\Theta^{\{k\}}$ = 5.3D-5, $d_0^{\{k\}}$ = 2765, $d_1^{\{k\}}$ = 2652

cidentally, it was the *absolutely nonconvex* variant which found this specific local minimum, but we may draw no conclusion from that pertaining to the relative performance of the positive definiteness strategies.

Table 2.4: Multistart statistics; 100 runs each.

Strategy	P	T	$\zeta^\dagger$	ζ^θ	$\zeta^\star$
nonconvex	36407	263	6	56	1
absolutely nonconvex	33375	123	9	20	3

We may however draw some conclusions from the statistics in Table 2.4; P is again the total numbers of subproblems the computational platform is tasked with, T is the estimated total CPU time in hours, $\zeta^\dagger$ is the number of untenable problem initializations, ζ^θ is the number of infeasible terminations[7] and $\zeta^\star$ is the number of times the best known topology—as depicted in Figure 2.7—is found. First, the number of untenable starting positions $\zeta^\dagger$ serves to suggest that the two variants of the algorithm were tested on equal footing with respect to the random starting points $\boldsymbol{t}^{\{0\}}$. Second, the SAO variant with the *absolutely nonconvex* strategy tasked the computational platform with fewer subproblems, but only marginally so. More pertinent is the gulf between the total CPU times T[8]. This suggests that the subproblems $P_{QP}[k]$ generated by the *absolutely nonconvex* strategy are much 'easier' to solve (in a numerical sense).

In terms of convergence properties, the number of infeasible terminations ζ^θ suggests that the SAO variant with the *nonconvex* strategy performs worse in terms of maintaining the feasibility of the sequence of iterates $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, 3, \ldots$. This finding is in line with that of Haftka [51], who demonstrated that a penalty function formulation preconditioned by an element-by-element approximate inverse of the stiffness matrix may lead to good convergence properties. This is undoubtedly also the reason for the large difference in total CPU times, assuming that the relative feasibility of an iterate $\boldsymbol{x}^{\{k\}}$ with respect to problem (2.1) equates to an 'easier' subproblem $P_{QP}[k]$—a reasonable proposition.

The *nonconvex* variant converged to the best known local minimum only once—likely the global minimum; in terms of a Bayesian multistart measure [61] a confidence in excess of 95%

[7]Keep in mind that even with a sophisticated restoration procedure the filter trust region algorithm [66], although it is referred to as 'convergent', is, practically speaking, only guaranteed to *terminate*—see also Ulbrich [78].

[8]This quantity is of course dependent on the total load on the computational platform during the experiment, which we can report was sufficiently constant.

may be associated with the potential global optimality of the solution depicted in Figure 2.7, specific of course to the parameters of the problem—whereas the *absolutely nonconvex* variant did so 3 times.

Although not reported here, the computational properties achieved by disregarding all second-order information and simply stating and solving the subproblem $P[k]$ as a linear program (LP) is substantially inferior to the results reported above. We tested GUROBI's primal simplex, dual simplex and Barrier optimizers for the LP case [76]. Again, it seems as though the approximate diagonal Hessian terms may be beneficial in terms of preconditioning the subproblem $P_{QP}[k]$, as argued by Haftka [51]. In terms of convergence properties, using a LP in $P[k]$ is roughly the same as using $P_{QP}[k]$ with the *nonconvex* positive definiteness strategy, which does not seem beneficial.

2.9 Concluding remarks

We have developed a strictly convex and separable Lagrange-Newton quadratic subproblem for SAND 'classical' topology design. The subproblem is characterised by the sparsity of the diagonal Hessian matrix and the ease with which positive definiteness is enforced, permitting a multitude of approximation strategies. In the SAND setting, retaining direct variables, it seems natural to use exact separable curvature terms in the diagonal quadratic approximation functions, although any form of curvature approximation is accounted for. Our investigations reveal that the separable curvature of the Lagrangian is concave close to and at a local minimum of the SAND topology design problem. It turns out that in traditional NAND formulations this is dealt with by using reciprocal intervening variables. Herein we simply 'convexify' the diagonal second-order terms. Numerical demonstrations agree with the theoretical results and suggest that artificial convexification of the approximate Lagrangian may aid computational and convergence properties of a SAO algorithm applied to the 'classical' SAND topology design problem. In the future alternative approximate subproblems—which may include off-diagonal Hessian terms, or which may be formulated and solved as nonconvex QP subproblems—should be studied and tested.

The methodology presented herein may form a platform on which tailored approximation strategies for SAND topology optimization problems can be developed. The important thing is that sensitivity information in the SAND setting takes the form of simple, sparse, easy to calculate, partial derivatives. This suggests that efficient solutions methods should be pursued in view of large scale topology design problem with many active constraints.

Chapter 3

On filtered "conservatism" in direct topology design

This chapter is the full-length version of a paper presented at the Sixth Internation Conference on Structural Engineering, Mechanics and Computation (SEMC 2016) hosted by the University of Cape Town from the 5th to the 9th of September 2016, in Cape Town, South Africa. The conference paper is entitled 'On filtered "conservatism" in direct topology design' [75]. It is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical and Mechatronic Engineering at the University of Stellenbosch, South Africa.

3.1 Abstract

This is a preliminary study into convergence and termination of a sequential approximate optimization algorithm based on convex diagonal quadratic subproblems in direct topology optimization. The direct problem is characterised by a large number of nonlinear equality constraints, hence the method forms part of a general large-scale nonlinear programming paradigm. In structural optimization globally convergent algorithms based on separable and conservative approximations are popular due to minimal storage requirements and minimal function evaluations, in addition to the fact that a trust-region need not be reduced to effect global convergence. However, these methods are restricted to inequality constrained problems. It is demonstrated that the convergence and termination properties that follow from the filtered acceptance of iterates in a trust-region framework can be effected in general nonlinear programming by conditionally increasing the curvature ("conservatism") of the approximate subproblems, without reducing the trust-region, or even, by neglecting the trust-region all-together. A large-scale instance of the direct formulated slope constrained MBB design problem is considered for numerical experimentation.

3.2 Introduction

Gradient-based sequential approximate optimization (SAO) methods based on convex and separable approximations are recognised as efficient methods for large-scale nonlinear program-

ming. Algorithms in this class are also known as sequential convex programming (SCP) methods. These algorithms appear often in structural optimization and, in particular, the subfield known as topology optimization. Originally SAO methods were based on the exploitation of efficient dual methods, permissible due to the separability of the approximations [37, 38]. Today however it is recognised that primal-dual methods are superior to pure dual methods in maintaining the sparsity of the Karush-Kuhn-Tucker (KKT) system of equations, especially for, unlike pure dual methods, problems with a large number of constraints [40, 45, 71, 79].

Distinguished algorithms in this class include CONLIN [34, 35], the method of moving asymptotes (MMA) [36], followed by SCPIP [45] and SAO*i* [41,80]. SAO methods in topology design (and structural optimization in general) are typically based on subproblems constructed from truncated linear Taylor series expansions, but, in order to model the nonlinear response of the structure, reciprocal-like intervening variables are used. In Reference [65] it is demonstrated that separable approximation functions with intervening variables can be replaced with simple diagonal quadratic Taylor series expansions; the form of the resulting subproblem is independent of the specific form of the approximations, the subproblems are easily convexified and conservatism is easily controlled [63, 74].

The effective use of reciprocal-like approximations is well established in traditional topology design and structural optimization in general [2,3,10,12,81], but the pertinence of reciprocal-like approximations is restricted to the pervasive *reduced* (nested analysis and design; NAND) formulation of the problem. See Reference [49] for alternative formulations in structural optimization. In the *direct* (simultaneous analysis and design; SAND) formulation of the problem, see *e.g* References [51, 53, 82], the equilibrium condition forms a set of nonlinear equality constraints—necessitating general, nonconvex, nonlinear programming techniques—but all sensitivity derivatives reduce to simple partial derivatives, second-order information is computationally cheap to obtain and the use of intervening variables seem somewhat redundant. That is, the direct formulation of the problem can be solved with techniques that use the exact, full Hessian matrix [44], although the convexity and computational benefits that derive from using only separable curvature information is clear [33, 45, 71]. Moreover, the dual SAO method in direct topology design can be reconciled with the traditional reduced OC approach [33], similar to the equivalence demonstrated in the reduced setting with exponential (and reciprocal) intervening variables [48].

In rudimentary form, with each iterate accepted unconditionally, these algorithms are not guaranteed to converge. Zillober adds a line search to MMA in order to guarantee convergence to a suitable solution, but this necessitates the definition of a tricky merit function [83, 84]. A 'minimalistic approach', the aim of which 'is to interfere as little as possible with the sequential quadratic programming iteration but to do enough to give a bias towards convergence' is preferred [85]. One strategy is to cast the algorithm in a trust-region SQP-filter algorithm [66, 86, 87]. The filter[1] replaces the merit function as a measure of the suitability of a given iterate, adjusting the trust-region accordingly. We denote this the *filtered trust-region* approach [74]. Importantly, this approach makes provision for nonlinear equality constraints.

Along a different line, an efficient and robust class of convergent methods can be found in Svanberg's conservative approximation framework (or 'conservative convex separable approximations'; CCSA) [67]. The CCSA method works by requiring that the approximation func-

[1]Not to be confused with the filters used in topology optimization to effect existence of solutions [58].

tions (objective and constraints) have greater values than the actual functions at the solution of a given subproblem, if not, the curvatures of the approximations are increased—*strict conservatism*—this technique ensures that the optimal solution of the subproblem is a feasible solution of the original problem with lower objective value than the previous iterate. Moreover, convergence in CCSA methods is maintained if feasible descent steps are excepted unconditionally—*relaxed conservatism* [80]—and the method is reconcilable with the filter approach mentioned above—*filtered conservatism*—in order to amalgamate the salient features of both strategies; chief among which is that the trust-region need not be reduced to effect convergence [74]. The CCSA method can only solve inequality constrained problems though. In fact, the CCSA framework was specifically developed for structural optimization problems in reduced form, for these typically contain only inequality constraints (evidenced too by the use of reciprocal approximations).

Herein we derive and test an approach that emulates the advantages of CCSA, but for general nonlinear programming; for problems that may contain nonlinear equality constraints. The strategy is denoted *filtered "conservatism"*—note the use of scare quotes—because the notion of a conservative approximation is not applicable to equality constraints.

We consider a general nonlinear mathematical programming problem $\mathcal{MP}$ written in standard negative-null form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_j(\boldsymbol{x}) = 0\,, \qquad j \in m_E\,, \\ & f_j(\boldsymbol{x}) \leq 0\,, \qquad j \in m_I\,, \\ & \boldsymbol{x} \in \mathcal{C}\,, \end{aligned} \tag{3.1}$$

where f_0 is a real valued scalar objective function, f_j, $j \in m_E$ is the set of inequality constraint functions and f_j, $j \in m_I$ the set of equality constraint functions, all of which may be nonlinear and/or nonconvex[2]. All functions f_j are typically assumed to be twice continuously differentiable. The domain of the problem is a closed and bounded subdomain of $\mathcal{R}^n$, imposed in the form of box constraints:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathcal{R}^n \mid \check{\boldsymbol{x}} \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}}\}\,, \tag{3.2}$$

where $\check{\boldsymbol{x}} = [\check{x}_1, \check{x}_2, \ldots, \check{x}_n]$ and $\hat{\boldsymbol{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]$; with $\check{x}_i$ and $\hat{x}_i$ the lower and upper bounds on each design variable x_i, $i = 1, 2, \ldots, n$.

We do not consider a relaxed version of the problem in order to ensure feasibility, as done by Svanberg [67]. It is assumed that $\mathcal{MP}$ has at least one feasible solution $\boldsymbol{x}^\star$ at which constraint qualification (CQ) holds. This is a fairly mild assumption in direct topology optimization; the issue is rather the existence of too many solutions, see *e.g* Reference [59].

[2] If $m_E = \emptyset$ it is understood that the problem is only inequality constrained.

3.3 Sequential approximate optimization

The generalized diagonal quadratic approximation [65] is used to construct a subproblem $\mathcal{QP}^k_{\text{sub}}$ in iteration k:

$$\begin{aligned} \min_{\boldsymbol{s}^k \in \mathcal{R}^n} \quad & q_0^k(\boldsymbol{s}^k) = f_0(\boldsymbol{x}^k) + \boldsymbol{\nabla} f_0^k \boldsymbol{s}^k + \boldsymbol{s}^T \frac{\boldsymbol{c}^k}{2} \boldsymbol{s}^k \\ \text{subject to} \quad & q_j^k(\boldsymbol{s}^k) = f_j^k + \boldsymbol{\nabla} f_j^k \boldsymbol{s}^k = 0 \,, \qquad j \in m_E \,, \\ & q_j^k(\boldsymbol{s}^k) = f_j^k + \boldsymbol{\nabla} f_j^k \boldsymbol{s}^k \leq 0 \,, \qquad j \in m_I \,, \\ & \boldsymbol{x}^{k+1} = (\boldsymbol{s}^k + \boldsymbol{x}^k) \in \mathcal{C} \,. \end{aligned} \tag{3.3}$$

The matrix $\boldsymbol{c}^k$ is understood to contain the n diagonal elements of the Hessian matrix of the approximate Lagrangian. The Lagrange multipliers at the solution of subproblem are unknown, hence the multipliers at the previous solution point are used:

$$\boldsymbol{c}^k = \boldsymbol{c}_0^k + \sum_{j=1}^{m} \lambda_j^k \boldsymbol{c}_j^k \,. \tag{3.4}$$

In order to ensure strict convexity (assuming feasibility) of $\mathcal{QP}^k_{\text{sub}}$ a small positive lower bound α is enforced on the elements of the diagonal Hessian matrix; in addition, nonconvex curvature information is retained [33, 71]:

$$c_{ii}{}^k \leftarrow \max(|c_{ii}{}^k|, \alpha) \qquad i = 1, 2, \ldots, n \,. \tag{3.5}$$

Herein exact second-order derivatives of the objective and constraint functions form the curvature terms:

$$c_{j,ii}^k = \frac{\partial^2 f_j^k}{\partial x_i^2} \quad i = 1, 2, \ldots, n \,, \quad j = 0, 1, \ldots, m \,. \tag{3.6}$$

It is assumed that $\mathcal{QP}^k_{\text{sub}}$ is feasible. The solution may in fact be nonunique—singular—but nevertheless attainable by dual and primal-dual methods [33, 88]. If the solution-step $\boldsymbol{s}^k$ accepted unconditionally the algorithm is referred to as *rudimentary*—Algorithm 1.

3.4 Convergence and termination

3.4.1 Conservatism

Guaranteed convergence and termination by *strict* and *relaxed conservatism* holds when $\mathcal{MP}$ has only inequality constraints: $m_E = \emptyset$. To inflict *strict conservatism Step 4* in Algorithm 1 is replaced with Algorithm 2[3]. The definition of a curvature multiplier $\chi > 1$ is required too. Under reasonable assumptions it is always possible to construct conservative approximations [67].

[3]For the sake of brevity, since Algorithms 2 and 3 are only for the purpose of illustration, the curvature terms are increased indiscriminately if any single approximation is not conservative. In practice this is done more judiciously, see *e.g.* Reference [67] and [74].

Algorithm 1 (rudimentary SAO)

Step 0: Initialization
Parameters: $\check{\boldsymbol{x}} \in \mathcal{C}$, $\hat{\boldsymbol{x}} \in \mathcal{C}$, $\boldsymbol{x_0} \in \mathcal{C}$, $\alpha > 0$, $\epsilon > 0$, $k_{\max} > 0$.
Construct $\mathcal{MP}^0_{\text{sub}}$.
for $k = 0 : k_{\max}$ **do**
  Step 1: Solve $\mathcal{QP}^k_{\text{sub}} \rightarrow \boldsymbol{s}^k, \boldsymbol{\lambda}^k$.
  Step 2: Set $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \boldsymbol{s}^k$.
  Step 3: Construct $\mathcal{QP}^{k+1}_{\text{sub}}$.
  Step 4: Null
  Step 5: Check for termination
  if $\|\boldsymbol{s}^k\| < \epsilon$ **then**
    Terminate.
  end if
end for

Algorithm 2 (substitution; *strict conservatism*)

Step 0: Initialization
Parameters: $\ldots, \chi > 1$.
$\ldots$
Step 4: Test for conservatism
if $q^k_j(\boldsymbol{x}^{k+1}) \geq f^{k+1}_j \; \forall \; j \in m_I$ **then**
  Continue.
else
  $\boldsymbol{c}^k \leftarrow \chi \boldsymbol{c}^k$.
  Return to 1.
end if

Enforced convergence based on *relaxed conservatism* is attained if *Step 4* in Algorithm 2 is substituted with Algorithm 3. (The only difference is that a feasible descent step is accepted unconditionally.) It is argued that the convergence proofs of CCSA methods rely solely on feasible descent steps, guaranteed by the conservatism of the approximations [80]. The maximum constraint violation at the new solution point $\boldsymbol{x}^{k+1}$ is denoted by Θ^{k+1}.

Algorithm 3 (substitution; *relaxed conservatism*)

Step 4: Test for feasible descent or conservatism
if $f^{k+1}_0 < f^k_0$ and $\Theta^{k+1} \leq 0$ **then**
  Continue.
else if $q^k_j(\boldsymbol{x}^{k+1}) \geq f^{k+1}_j \; \forall \; j \in m_I$ **then**
  Continue.
else
  $\boldsymbol{c}^k \leftarrow \chi \boldsymbol{c}^k$.
  Return to 1.
end if

Even though $\boldsymbol{c}^k$ is viewed as the diagonal elements of the approximate Lagrangian in its construction (3.4), from the point of view of the subproblem $\boldsymbol{c}^k$ is an approximate objective curvature. The step $\boldsymbol{s}^k$ to the stationary point of the approximate objective function q_0^k, with inflated curvature $\chi\boldsymbol{c}^k$, can be written as

$$\boldsymbol{s}^k = -\chi[\boldsymbol{c}^k]^{-1}\boldsymbol{\nabla} f_0^k \,, \tag{3.7}$$

and by taking the Euclidean norm on both sides, a measure of the step size is attained:

$$\|\boldsymbol{s}^k\| = \frac{\|\boldsymbol{\nabla} f_0^k\|}{\chi\|\boldsymbol{c}^k\|} \,. \tag{3.8}$$

That is, the curvature inflation $\chi\boldsymbol{c}^k$ might effect a step $\boldsymbol{s}^k$ of reduced size—dependent on the feasible set of $\mathcal{QP}^k_{\text{sub}}$. It is via this mechanism that an attempt is made to generate a feasible descent step in *relaxed conservatism*.

3.4.2 Filtered trust-region

It is customary to rather control the size of the generated step $\|\boldsymbol{s}^k\|$ with a trust-region—known as 'restricted step methods' [89]—delimiting a 'trusted' neighbourhood Δ^k around the current iteration point $\boldsymbol{x}^k$. This takes the form of a proportional move-limit $0 < \delta^k \leq 1$ on the design variables:

$$\Delta^k = \{\boldsymbol{x}^{k+1} \in \mathcal{C} | \boldsymbol{s}^k \leq \delta^k(\hat{\boldsymbol{x}} - \check{\boldsymbol{x}})\} \,, \tag{3.9}$$

with the resulting subproblem—wherein $\boldsymbol{s}^k \in \Delta^k$ is imposed—denoted by $\mathcal{QP}^k_\Delta$.

In order to guarantee convergence and termination the trust-region is combined with a Pareto front (or 'filter') of the objective function value f_0 and the maximum constraint violation Θ; the filter consists of the list of pairs

$$\mathcal{F}^k \subset \{(\Theta^h, f_0^h) : \; h = 1, 2, \ldots, k\} \,, \tag{3.10}$$

and using a 'slanting envelope test' [66, 85], if either

$$\Theta^{k+1} \leq \beta\Theta^h \quad \text{or} \quad f_0^{k+1} + \gamma\Theta^{k+1} \leq f_0^h \tag{3.11}$$

for all $h = 1, 2, \ldots, k$, the pair $(\Theta^{k+1}, f_0^{k+1})$ is deemed acceptable to the filter. The envelope parameters are constant and restricted to $0 < \gamma < 1$ and $\beta = 1 - \gamma$. In addition, the new iterate has to pass one of two descent conditions:

$$\rho^k := \frac{f_0^k - f_0^{k+1}}{q_0^k(\mathbf{0}) - q_0^k(\boldsymbol{s}^k)} > \eta \,, \tag{3.12}$$

or

$$q_0^k(\mathbf{0}) - q_0^k(\boldsymbol{s}^k) := \sigma^k \leq \kappa(\Theta^k)^\psi \,, \tag{3.13}$$

with constants $0 < \eta < 1$, $0 < \kappa < 1$ and $\psi > 1$, for the algorithm to progress to $k + 1$.

The *filtered trust-region* variant of the algorithm is depicted with the sequence of steps in Algorithm 4, which replace *Step 4* in Algorithm 2 or 3. Before we assumed feasibility of

each $\mathcal{QP}^k_{\text{sub}}$, with the introduction of the trust-region region however—subproblem $\mathcal{QP}^k_{\Delta}$—this assumption becomes unreasonable. Therefore, a rudimentary back-tracking strategy is activated if an infeasible subproblem is encountered (*Step 4.0*); the pair (Θ^k, f_0^k) is added to $\mathcal{F}^k$, the trust-region is expanded to the original feasible domain $\Delta^k \leftarrow C$ and the algorithm is restored to the last iterate $\boldsymbol{x}^g$ which gave rise to a feasible subproblem. The definition of the additional parameters γ, η, κ, ψ, δ_{min} and ϕ are required too, but χ is redundant.

Algorithm 4 (modifications; *filtered trust-region*)

Step 0: Initialization
Parameters: $\ldots, 0 < \gamma < 1$ ($\beta = 1 - \gamma$), $0 < \eta < 1$, $0 < \kappa < 1$, $\psi > 1$, $\delta_{\text{min}} > 0$ ($\delta^0 = 1$; $\Delta^0 = \mathcal{C}$) and $\phi > 0$ (~~$\chi \geqslant 0$~~).
$\ldots$

Step 4.0: If $\mathcal{QP}^k_{\Delta}$ is inconsistent
Add (Θ^k, f_0^k) to $\mathcal{F}^k$, $\delta^k \leftarrow \delta^0$ ($\Delta^k \leftarrow \mathcal{C}$), $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^g$.
Return to 1.

Step 4.1: Test for acceptability, relative descent and feasible descent
if $\Theta^{k+1} \leq \beta\Theta^h$ or $f_0^{k+1} + \gamma\Theta^{k+1} \leq f_0^h \; \forall \, h = 1, 2, \ldots, k$ **then**
    if $\rho^k > \eta$ or $\sigma^k < \kappa(\Theta^k)^\psi$ **then**
        Continue.
    else
        $\delta^k \leftarrow \delta^k/\phi$, terminate if $\delta^k < \delta_{\text{min}}$.
        Return to 1.
    end if
else
    $\delta^k \leftarrow \delta^k/\phi$, terminate if $\delta^k < \delta_{\text{min}}$.
    Return to 1.
end if

Step 4.2: Update the trust-region and conditionally update the filter
$\delta^{k+1} \leftarrow \phi\delta^k$, $\boldsymbol{x}^g \leftarrow \boldsymbol{x}^k$.
if $\sigma^k < \kappa(\Theta^k)^\psi$ **then**
    Add $(\Theta^{k+1}, f_0^{k+1})$ to $\mathcal{F}^{k+1}$.
end if
Continue.

Fletcher and co-workers prove a number of lemmas which demonstrate and guarantee the mechanism by which the *filtered trust-region* approach produces a sequence of convergent iterates; under reasonable assumptions the algorithm will either terminate because the restoration procedure fails—a more sophisticated strategy is used in Reference [66]—or converge to a stationary (KKT) point of $\mathcal{MP}$, or accumulate at a feasible point which fails to satisfy Mangasarian-Fromowitz CQ [78].

3.4.3 Filtered "conservatism"

Finally the *filtered conservatism* variant of the algorithm can be introduced; Algorithm 4 is modified as per Algorithm 5. *Filtered "conservatism"* is of course closely related to the *filtered conservatism* approach devised in Reference [74]. In the latter (which is restricted to inequality constrained problems) convergence and termination follows due to a combination of lemmas in Reference [66] and [67]. It has to be assumed however that conservatism has the same effect as a trust-region in generating points that are acceptable to the filter—a reasonable proposition.

Algorithm 5 (substitutions; *filtered "conservative"*)

Parameters: ..., $\chi > 0$ and $c_{\max} > \mathbf{0}$ ($\cancel{\delta^0}$, $\cancel{\delta^k}$, $\cancel{\delta_{\min}}$, $\cancel{\Delta^0}$, $\cancel{\Delta^k}$ and $\cancel{\phi}$).
...
Replace all $\mathcal{QP}^k_{\Delta}$ with $\mathcal{QP}^k_{\text{sub}}$.
Replace all $\delta^k \leftarrow \delta^k/\phi$ with $\boldsymbol{c}^k \leftarrow \chi \boldsymbol{c}^k$.
Replace all $\delta^k \leftarrow \phi\delta^k$ with $\boldsymbol{c}^k \leftarrow \boldsymbol{c}^k/\chi$.
Replace all $\delta^k < \delta_{\min}$ with $\|\boldsymbol{c}^k\| > c_{\max}$

Due to the presence of nonlinear equality constraints in $\mathcal{MP}$, only the lemmas of the *filtered trust-region* approach [66] can be relied upon to guarantee convergence and termination. It is assumed that increasing the curvature of $\mathcal{QP}^k_{\text{sub}}$ can generate iterates acceptable to the filter, a reasonable proposition, especially light of relation (3.8). That is, inflated curvature $\chi \boldsymbol{c}^k$ may reduce the size of the solution step $\boldsymbol{s}^k$, much like a trust-region. (Although, the advantages that may be derived from the backtracking strategy in *Step 4.0*, possibly induced by the trust-region, is lost.) Moreover, the intuition that *filtered "conservatism"* can generate acceptable steps is not far-fetched: using the Cauchy-Swartz inequality, one may write of the approximate objective function that

$$|q^k_0(\mathbf{0}) - q^k_0(\boldsymbol{s}^k)| \leq \|\boldsymbol{\nabla} f^k_0\| \|\boldsymbol{s}^k\| + \frac{1}{2}\|\boldsymbol{c}^k\| \|\boldsymbol{s}^k\|^2 \,, \tag{3.14}$$

substitution of the relation in (3.8) in (3.14) yields

$$|q^k_0(\mathbf{0}) - q^k_0(\boldsymbol{s}^k)| \leq \frac{\|\boldsymbol{\nabla} f^k_0\|^2}{\chi \|\boldsymbol{c}^k\|} + \frac{1}{2}\frac{\|\boldsymbol{\nabla} f^k_0\|^2}{\chi^2 \|\boldsymbol{c}^k\|} \,, \tag{3.15}$$

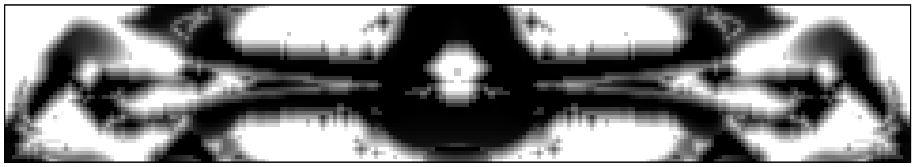
which shows that either condition (3.12) or (3.13) may eventually be satisfied for $\boldsymbol{c}^k$ large enough, much like the original *filtered conservatism* approach for inequality constrained problems. Note that guaranteed satisfaction of either condition (3.12) or condition (3.13) guarantees eventual continuation to $\boldsymbol{x}^{k+1}$, which, in conjunction with the filter and the associated assumptions, guarantees convergence and termination.

3.5 Numerical experiments

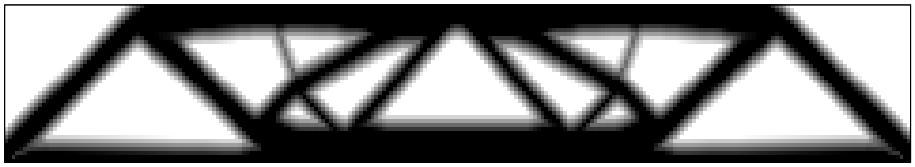
The MBB version of the direct topology design problem is considered—see *e.g.* Reference [33] for a definition of the problem—discretised with 120×40 material density variables $\boldsymbol{t} = [t_1, t_2, \ldots, t_{4800}]$. With the addition of slope constraints, required to suppress numerical instabilities [60], the direct formulated problem has $34\,242$ design variables, $29\,442$ equality constraints

and 18881 inequality constraints. Material density variables $\boldsymbol{t}$ are bounded above at 1 and below at 0 exactly. Displacement variables $\boldsymbol{u}$ are limited to $\pm 1 \times 10^6$. The vector of design variables is $\boldsymbol{x} = [\boldsymbol{u}, \boldsymbol{t}]$. Computations are performed on Stellenbosch University's Rhasatsha HPC [77].

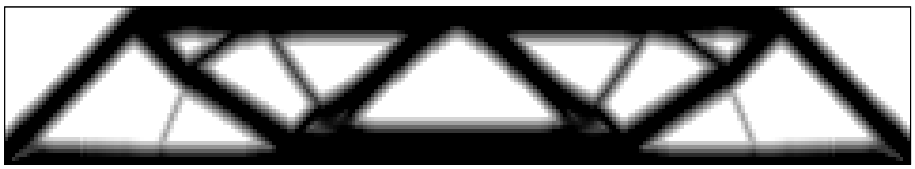
First the algorithms are initialized with an uniform material distribution $\boldsymbol{t}^0 = \mathbf{0.5}$; results are collated in Figure 3.1; l is the number of 'Return to 1' instances (inner iterations). The *rudimentary* algorithm failed to terminate. The *"conservative"* version of the algorithm compares well to the *trust-region* approach, especially considering that no trust-region is employed in the former, but a major difficulty in topology optimization is illustrated; due to the presence of many local minima the solutions are severely dependent on algorithmic parameters and barely comparable. To negate this effect 200 problem instances are initialized with a random material distribution. Each material density variable is initialized with a random value between 0.1 and 0.9. Only the convergent versions of the algorithms are tested; 100 problems each. The best solution is depicted in Figure 3.2 with statistics given in Table 3.1. In this case the *"conservative"* algorithm is more expensive (but the starting position is of course different for each algorithmic instance). Overall statistics are summarised in Table 3.2; K and L are the total number of outer and inner iterations ($K + L$ is thus the total computational effort); $\zeta_0^\star$ indicates the number of times the optimal topology is found and $\zeta_\Theta^\dagger$ is the number of infeasible terminations. The *"conservative"* version of the algorithm converged to the optimal topology more often with less computational effort.



(a) *Rudimentary:* k=999, l=0, $f_0^\star$=9763.2 $\Theta^\star$=3D+4



(b) *Filt. trust-region:* k=120, l=123, $f_0^\star$=219.87, $\Theta^\star$=3D-7



(c) *Filt."conservatism":* k=143, l=133, $f_0^\star$=221.26, $\theta^\star$=7D-6

Figure 3.1: Topologies generated for the MBB design problem; $\boldsymbol{t}^0 = \mathbf{0.5}$.

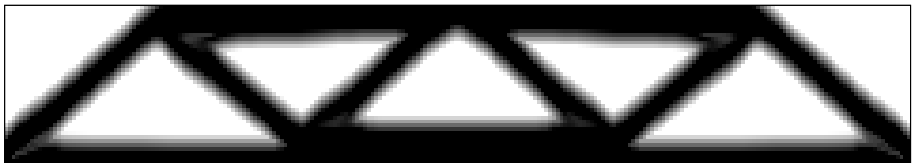


Figure 3.2: Optimal topology generated for 200 random initial material distributions; $\mathbf{0.1} \leq \boldsymbol{t}^0 \leq \mathbf{0.9}$.

Table 3.1: Multistart statistics; optimal topologies.

Algorithm	k	l	$f_0^\star$	$\Theta^\star$
Filt. trust-region	149	68	217.91	1D-6
Filt."conservatism":	248	138	218.11	2D-5

Table 3.2: Multistart statistics; 100 runs each.

Algorithm	K	L	$K+L$	$\zeta_0^\star$	$\zeta_\Theta^\dagger$
Filt. trust-region	15018	10406	25424	4	2
Filt. "conservatism"	14629	7570	22199	12	2

3.6 Concluding remarks

It is not prudent, given that the test set is rather small (and confined to a single test problem), to attempt far-reaching conclusions. However, it is fair to say that *"conservatism"* in general nonlinear programming appears to be viable notion. Moreover, the abolishment of the trust-region seems to encourage the algorithm to reach the globally optimal solution, for the entire design domain is available in each approximate subproblem. Recommended future work includes a study of the locally superlinear convergent version of the SQP-filter method devised in Reference [78].

Chapter 4

On design-set restriction in SAND topology optimization

This chapter is a reproduction of a paper which, at the time of writing, is under review for publication [59]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical and Mechatronic Engineering at the University of Stellenbosch, South Africa.

4.1 Abstract

We study design-set restriction—*i.e.* existence of solutions, mesh-dependence, local minima, and manufacturability—in the alternative SAND ('simultaneous analysis and design', or 'direct') formulation of the classical SIMP minimum compliance topology optimization problem. Straightforward computational considerations pertaining to common restriction methods suggest that so-called 'slope constraints'—point-wise bounds on the gradient of the material distribution function—are able to control the complexity of design with minimal (additional) computational requirements. A standard finite element procedure is used and the SAND optimization problem—a large-scale, general, nonlinear mathematical program—is solved with a conventional sequential approximate optimization (SAO) method based on strictly convex and separable approximate subproblems. A random multistart strategy and a simple relaxation procedure suggests that manufacturable, 'probably' globally optimal, 0-1 designs may be obtained in a reasonable amount of computation time.

4.2 Introduction

Brought about by the sophistication of modern primal-dual sequential convex programming (SCP) routines, on top of a general increase in common computational capabilities, the 'simultaneous analysis and design' (SAND, or 'direct') formulation of the topology optimization problem has recently emerged as a tenable alternative—see for example References [33,44,88]. The SAND approach is characterised by doing away with the external 'simulation' phase—*i.e.* numerical solution of the finite element (FE) discretized equilibrium equation—carried out fully in every algorithmic iteration in the conventional 'nested analysis and design' (NAND,

or 'reduced') approach. In the SAND setting the FE equations form a set of nonlinear equality constraints—in an 'unified' optimization problem—which are only fully satisfied at convergence of the SAND optimization routine. The interested reader is referred to Arora and Wang [49] for a monograph on alternative formulations in structural optimization, and the comprehensive benchmark study by Rojas-Labanda and Stolpe [44]. Reference is also made to the work by Haftka *et al.* [51, 53, 90], that of Bendsøe *et al.* [52], and Wang and Jasbir [91].

In classical minimum compliance topology optimization, sequential approximate optimization (SAO) of the SAND problem can be interpreted in the light of the ubiquitous optimality criteria (OC) approach [33], similar to the equivalence demonstrated between the OC method and traditional dual SAO of the NAND problem [48]. SAO based on efficient, diagonal quadratic approximate subproblems [65] encompass and generalise the popular dual SCP alternatives based on intervening variables—like *e.g.* CONLIN [34, 35] and the MMA [36]. However, in the SAND setting traditional convex approximation techniques—based on intervening variables—seem somewhat redundant, in part due to the cheap availability of analytical second-order sensitivity information, and in part due to the concavity of the separable Lagrangian—necessitating counter-intuitive techniques in the enforcement of strict convexity on the approximate subproblems [33]. Moreover, the SAND approach may turn out to be especially suited to local stress-constrained problems—in fact, problems with local displacement-based constraints in general—for the typically dense and computationally expensive sensitivity derivatives associated with the adjoint analysis in NAND, reduce to simple, easy to calculate, sparse, partial derivatives in the SAND formulation [88].

A very important topic that has not been sufficiently addressed in the literature thus far is how to deal with the compounded issues of *nonexistence* of solutions[1], *mesh-dependence*, *manufacturability*, and the typically severe *multimodality*[2] of the optimization problem in the unconventional SAND setting. We attempt to address these issues here.

In Section 4.3 the classical topology optimization problem is introduced. In Section 4.4 popular techniques devised to remedy or mitigate the above-mentioned issues—pertaining to restriction of the design set—are recited and analysed with a view to the SAND setting. Section 4.5 is a description of the preferred numerical method, Section 4.6 chronicles a series of numerical demonstrations and experiments, and Section 4.7 concludes.

4.3 Classical topology optimization

A solution to the classical minimum compliance topology optimization problem is a domain Ω^m, a subdomain of a predefined design domain Ω in $\mathcal{R}^2$ or $\mathcal{R}^3$. Material is either present or absent at each and every spatial position $x \in \Omega$. The subdomain Ω^m denotes the presence of material and defines the structural connectivity and shape which minimizes the compliance objective—Ω^m is thus the geometry of the stiffest structure achievable with a predefined volume, a fraction $\bar{\nu}$ of the design domain Ω. The derivation of the problem is fairly well-known—see for example the canonical work by Bendsøe and Sigmund [2]—but is repeated here to facilitate the introduction of pertinent concepts, notation conventions, and to emphasise

[1] The imposition of a minimum length scale; which relates to manufacturability.

[2] Nonconvexity; many local minima.

the similarity between the original problem statement and the SAND formulation.

In structural optimization the subdomain Ω^m is occupied by an elastic body. The internal virtual work $a(u, v)$ of the elastic body, subject to an equilibrium displacement u and a kinematically admissible virtual displacement v, is written in the energy bilinear form

$$a(u, v) = \int_\Omega E_{ijkl}(x)\epsilon_{ij}(u)\epsilon_{kl}(v)\mathrm{d}\Omega \tag{4.1}$$

with the strains ϵ linearized according to

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) . \tag{4.2}$$

The stiffness tensor $E_{ijkl}(x)$ is a function of a continuous spatial position $x \in \Omega$ and may be written as

$$E_{ijkl}(x) = \begin{cases} E^0_{ijkl} & \text{if } x \in \Omega_m \\ 0 & \text{if } x \in \Omega \backslash \Omega_m \end{cases} \tag{4.3}$$

with E^0_{ijkl} the stiffness tensor of a predefined (solid) isotropic material.

The topology may also be defined in terms of a 'material distribution function'[3] $\rho(x)$, which has a value of 1 if $x \in \Omega^m$ and 0 elsewhere. Formally, (4.3) is replaced with

$$E_{ijkl}(x) = \rho(x)E^0_{ijkl} \quad \text{with} \quad \rho(x) = \begin{cases} 1 & \text{if } x \in \Omega^m \\ 0 & \text{if } x \in \Omega \backslash \Omega^m \end{cases} \tag{4.4}$$

which, in turn, yields the classical 0-1 minimum compliance topology optimization problem:

- *The 0-1 topology optimization problem* $\mathscr{P}_d$

$$\begin{aligned} \min_{u,\rho} \quad & l(u) \\ \text{subject to} \quad & a(u, v) = l(v) \quad \forall\, v \in U \\ & \int_\Omega \rho(x)\, \mathrm{d}\Omega \leq \bar{\nu} \\ & \rho(x) = 0 \text{ or } 1 \quad \forall\, x \in \Omega \end{aligned} \tag{4.5}$$

The compliance objective is

$$l(u) = \int_\Omega ru\mathrm{d}\Omega + \int_{\Gamma_T} tu\mathrm{d}s \tag{4.6}$$

with r the body forces and t the surface tractions applied to the boundary of the domain Γ_T. The virtual compliance of the structure is denoted by $l(v)$, a function of the kinematically admissible virtual displacement field $v \in U$.

[3]The material distribution function is often referred to as a 'density' distribution function since the discrete requirement on $\rho(x)$ is typically relaxed to permit the use of efficient gradient-based optimization methods, as we do herein.

The 0-1 (discrete) problem $\mathscr{P}_d$ lacks solutions in general: for a constant amount of material $\bar{\nu}$ the stiffness of the structure can be improved *ad infinitum* by arranging the material in finer and finer topological features. Put differently, between any two spatial positions there are an infinite number of points where material can either be present or absent, and thus, there an infinite number of ways a constant amount of material can be arranged. This is related to the absence of a minimum length scale and the design set is said to lack *closedness*, resulting in the *nonexistence of solutions*—the interested reader is referred to Sigmund and Petersson [57] for an exposition of the subject.

In computational implementations, wherein a solution to problem $\mathscr{P}_d$ is sought by numerical means, the domain Ω is invariably discretized with a FE-type procedure. Each element is assigned a design variable ρ_i which represents the amount of material in the element. The vector of material (density) variables is denoted by $\boldsymbol{\rho} \in \mathcal{R}^e$, with e the number of elements. The vector of nodal displacements is denoted by $\boldsymbol{u} \in \mathcal{R}^d$, with d the number of degrees of freedom. The SAND formulation of the problem has a vector of primal optimization variables $\boldsymbol{x}$ composed of both $(\boldsymbol{\rho}, \boldsymbol{u}) \in \mathcal{R}^{n=e+d}$:

- *The FE discretized 0-1 topology optimization problem* P_d

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{\rho}} \quad & f_0(\boldsymbol{u}) = \boldsymbol{r}^T\boldsymbol{u} \\ \text{subject to} \quad & [\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} - \boldsymbol{r} = 0 \\ & f_1(\boldsymbol{\rho}) = \mathbf{1}^T\boldsymbol{\rho} - \bar{\nu} \leq 0 \\ & \rho_i \in [0,1] \quad \text{for} \quad i = 1, 2, \ldots, e \end{aligned} \tag{4.7}$$

The FE discreitization itself introduces a minimum length scale, but the *nonexistence* issue then manifests as the problem of *mesh-dependence*, for, as theory predicts, finer topological features form part of the optimum design for progressively finer FE discretizations. That is, not only is the optimal solution quantitatively different for different FE discretizations, but worse, the integrity of the FE model and manufacturability of the design might be comprised if the structural features are too detailed.

To avoid computationally expensive integer programming techniques, and, in turn, permit the use of efficient gradient-based optimization methods, it is customary to relax the discrete requirement on the material distribution function $\rho(x)$. 'Grey' material $0 < \rho(x) < 1$ may then be interpreted as equivalent to a material 'thickness' in a 'variable thickness sheet'-type (VTS) problem [2], or, alternatively, in terms of a composite microstructure [92][4]. In general however, 0-1 designs should be sought. To this end the predominant technique is SIMP, independently proposed by Bendsøe [54] and Rozvany and Zhou [55]. SIMP is designed to approximate the original discrete behaviour of the material distribution function with a continuously differentiable equivalent. Therefore, the relation in (4.4) is replaced with

$$E_{ijkl}(x) = \rho(x)^p E^0_{ijkl}\,, \quad 0 \leq \rho(x) \leq 1 \tag{4.8}$$

with p the SIMP penalization parameter. For $p > 1$ material of intermediate 'density' is penalized via the artificial material law expressed in (4.8), degrading the structural efficiency of low

[4]Relaxation of the discrete requirement on $\rho(x)$ results in the existence of solutions too [57].

'density' regions $\rho(x) \ll 1$, and, because these regions are uneconomical in the presence of the volume constraint, the material distribution $\rho(x)$ is driven to approximately 0 and 1 [93]. The interested reader is referred to Bendsøe and Sigmund [94] for a overview of material interpolation schemes in topology optimization. The classical SIMP minimum compliance topology optimization problem may be expressed as:

- *The SIMP topology optimization problem* $\mathscr{P}_c$

$$\begin{aligned} \min_{u,\rho} \quad & l(u) \\ \text{subject to} \quad & a(u,v) = l(v) \quad \forall\, v \in U \\ & E_{ijkl}(x) = \rho(x)^p E^0_{ijkl} \\ & \int_\Omega \rho(x)\, \mathrm{d}x \leq \bar{\nu} \\ & 0 \leq \rho(x) \leq 1 \quad \forall\, x \in \Omega \end{aligned} \tag{4.9}$$

Here the subscript in $\mathscr{P}_c$ indicates the *continuous* material–energy relation expressed in (4.8), as opposed to the *discrete* relation in (4.4). The associated FE discretized problem may be written as:

- *The FE discretized SIMP topology optimization problem* P_c

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{\rho}} \quad & f_0(\boldsymbol{u}) = \boldsymbol{r}^T \boldsymbol{u} \\ \text{subject to} \quad & \sum_{i=1}^{e} (\rho_i)^p [\boldsymbol{K}_i] \boldsymbol{u} - \boldsymbol{r} = 0 \\ & f_1(\boldsymbol{\rho}) = \mathbf{1}^T \boldsymbol{\rho} - \bar{\nu} \leq 0 \\ & 0 \leq \rho_i \leq 1\,, \qquad i = 1, 2, \ldots, e \end{aligned} \tag{4.10}$$

with $\sum_{i=1}^{e} (\rho_i)^p [\boldsymbol{K}_i]$ the $d \times d$ globally assembled, SIMP modified stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})]$. The feasible set of P_c is nonempty under fairly mild assumptions [56].

Before we move on to popular restriction methods and the manifestation of these in the SAND setting, we wish to emphasise that, in problem P_c, the FE equilibrium equation is part of the optimization problem in the form of a set of nonlinear equality constraints—similar to the material–energy equilibrium relation in $\mathscr{P}_c$ (and $\mathscr{P}_d$ for that matter). Moreover, computationally speaking, because the FE discretized equilibrium equation $[\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} = \boldsymbol{r}$ is not solved *per se*—the global stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})]$, which may be singular, is not 'inverted'—material variables $\boldsymbol{\rho}$ are permitted to take on a value of zero exactly in order to represent void subdomains $\Omega \backslash \Omega_m$. In the NAND setting material variables $\boldsymbol{\rho}$ are typically limited to a small nonzero value on the lower bound to avoid singularity of the global stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})]$. (Low density elements may be removed from the FE equations, but this requires fairly involved algorithmic procedures [95, 96].)

4.4 Restriction methods

Restriction methods limit the set of admissible designs, which lacks *closedness* otherwise, by imposing an extra constraint (or multiple constraints, or similar) on either $\mathscr{P}_d$ or $\mathscr{P}_c$[5]. These techniques are devised to limit the maximum oscillation of $\rho(x)$ with respect to x. If the restriction method enforces sufficient *closedness* on the set of admissible designs, solutions are said to *exist* and *mesh-independence* (of the global optimum) is ensured. Herein, for the sake of brevity, but without the loss of generality, restriction methods are studied in 2D—as is customary.

4.4.1 Perimeter control

A bound on the perimeter of the subdomain Ω^m restricts the design set by limiting the complexity of the topology—'the number of holes that can appear' [2]. Existence of solutions and FE convergence is ensured for both problems $\mathscr{P}_d$ and $\mathscr{P}_c$ [97]. A useful side effect of the continuous material–energy relation (4.8) in $\mathscr{P}_c$ is the differentiability of $\rho(x)$ with respect to x. Hence, the perimeter constraint (and other restriction methods) may be expressed and formulated in terms of the gradient of the material distribution function $\rho(x)$. In 2-D, the perimeter constraint may be expressed as an upper bound on the total variation of $\rho(x)$:

$$TV(\rho) = \int_{\mathcal{R}^2} ||\nabla\rho||\,\mathrm{d}x \leq \bar{TV}\,. \tag{4.11}$$

For an element wise constant FE discretization, the total variation may be calculated as the summation of the jumps in amount of material over each and every inter element boundary

$$TV(\rho) = \sum_{k=1}^{b} l_k \left(\sqrt{(\rho_{i(k)} - \rho_{j(k)})^2 + \epsilon^2} - \epsilon\right) \tag{4.12}$$

with l_k the associated elemental length, b the number of inter element boundaries, and $\rho_{i(k)}$ and $\rho_{j(k)}$ the elements which share inter element boundary number k. The parameter ϵ is a small positive number required in order for (4.12) to be differentiable. The global nature of the constraint has the severe drawback that the gradient of the material distribution function $\rho(x)$ is not limited on a local scale. Therefore, fine structural features—*e.g.* very thin bars—may form part of the optimal topology and the manufacturability of the design might be compromised. Moreover, Bendsøe and Sigmund [2] report that the perimeter constraint can be quite difficult to approximate in an SAO solution procedure and may cause oscillatory algorithmic behaviour [98].

In Section 4.6 the perimeter measure defined in (4.12) is used to quantify the complexity of the calculated topologies—with $\epsilon = 0$, since (4.12) is not differentiated—but the requirement of an additional heuristic parameter, reported algorithmic issues, and, most importantly, the lack of control over minimum member size, permits us to neglect design set restriction via perimeter control in the SAND setting.

[5] Sigmund and Petersson [57] warn that 'restriction' methods devised in terms of the FE discretized problems P_d or P_c, which 'generally has solutions since it is posed in finite dimension', is 'dubious', since 'one simply produces pictures that people want to see'.

4.4.2 Filtering methods

Filtering methods form the predominant class of restriction methods in SIMP based NAND topology optimization. The method is succinctly demonstrated in the well-known TopOpt Matlab codes [10, 12, 14]. Two options are available; either filtering of the sensitivity information or filtering of the density distribution itself—the first is employed in Reference [10] and the second is incorporated in Reference [12]. Rojas-Labanda and Stolpe [44] present rare and notable test cases of density filtering applied to the SAND formulated problem. The interested reader can also refer to Sigmund [99] for a novel class of filtering methods based on image processing techniques.

First, lets discuss *density* filtering—the subsection is closed with *sensitivity* filtering. Bourdin [58] provides a proof of existence of solutions and FE convergence for the SIMP based problem $\mathscr{P}_c$. Density filtering is defined as a modification to the SIMP material–energy relation in (4.8) with a convolution product

$$E_{ijkl}(x) = ((\rho \star K)(x))^p E^0_{ijkl} \tag{4.13}$$

with

$$(\rho \star K)(x) = \frac{\int_\Omega \rho(y) K(x-y) \mathrm{d}y}{\int_\Omega K(y) \mathrm{d}y} \tag{4.14}$$

and K the convolution kernel, for example

$$K(x-y) = \begin{cases} 1 - \frac{||x-y||}{\eta} & \text{if } ||x-y|| \leq \eta \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

and η the filter radius. That is, the density (and stiffness) of the structure at position x is dependent on a weighted average of all the densities within the filter radius η. For an element wise constant FE discretization, the density filter transforms the original density of element i, ρ_i, to

$$\tilde{\rho}_i = \frac{\sum_{j \in N_i} H_{ij} \rho_j}{\sum_{j \in N_i} H_{ij}} \tag{4.16}$$

with N_i the set of elements for which the center-to-center distance between element i and element j, $C(i,j)$, is less than the predefined filter radius η. The weight factor H_{ij} may be defined as, for example

$$H_{ji} = \max(0, \eta - C(i,j)) \,. \tag{4.17}$$

The density filter has the effect that the original densities ρ_i loose their physical meaning [99], and, because of the modification to the stiffness tensor, a modification to the associated sensitivity information is required too.

Density filtering requires no additional constraint functions and is therefore presented as a computationally efficient restriction method—in *e.g.* Reference [2]—but this advantage is, seemingly, confined to the NAND formulation. In order to demonstrate this, consider a standard $e_x \times e_y$ FE discretization. For the sake of argument, 10 000 Q4 FE's are used with $e_x = e_y = 100$. The constraint Jacobian matrix of problem P_c will have 20 403 rows—20 402 equilibrium

constraints[6] and 1 volume constraint—and 30 402 columns—20 402 displacement variables $\boldsymbol{u}$ and 10 000 material variables $\boldsymbol{\rho}$. In this form, the Jacobian matrix has 452 404 nonzero entries—the matrix is approximately 0.06% filled. (The well-known and freely available Matlab codes [10, 12, 14] may be used to generate these numbers.)

Density filtering works by coupling the material variables ρ_j, $\forall j \in N_i$—those elements within the filter radius η—to element i. Because this coupling is 'filtered' through a weighted average (4.16) of all the elements in the filter radius η, fine structural features are weakened and removed, reducing the complexity of the design. The extent of this coupling can be approximated with a square of $e_r \times e_r$ elements, with typically $e_r \ll e_x$ and $e_r \ll e_y$. First, each equilibrium constraint is a function of 4 material variables[7] ρ_i. Second, if density filtering is imposed, each material variable ρ_i is a function of $e_r^2 - 1$ neighbouring[8] material variables ρ_j, $\forall j \in N_i$. That is, the addition of roughly $4(e_r^2 - 1)$ nonzero terms in each equilibrium constraint is necessitated. For example, if a filter radius of $e_r = 5$ is used, about $4 \times 24 \times 20402 = 1958592$ additional nonzero terms[9] will be present in the constraint Jacobian matrix—the modified matrix is approximately 0.39% filled—that is almost a sixfold increase in computational storage requirements. For $e_r = 10$ (the filter radius is often set to about 10% of the design domain) the Jacobian matrix is 1.38% filled—about 20 times the original value.

Rojas-Labanda and Stolpe [44], who solve and compare a host of simply-constrained, density filtered NAND and SAND problems, concluded that the computational requirements associated with the SAND formulation is its main drawback—reportedly some SAND problems had to be removed from the test set due to time and memory limitations—seemingly the additional nonzero terms associated with design set restriction by density filtering (in conjunction with the calculation and storage of the full Hessian matrix, in some cases [44]) contributed to this. In NAND density filtering is computationally cheap, since it is applied directly between design iterations [10, 12, 14].

Sensitivity filtering, on the other hand, is closely related to density filtering, and equivalent to the compliance minimization of nonlocal elasticity problems [100]. The method can be justified with extensive 'computational experience' in the NAND setting [2], but it 'suffers' due to its lack of mathematical rigour [100]. In the NAND setting the method works by modifying the sensitivity derivatives of the compliance objective function $f_0(\boldsymbol{u}(\boldsymbol{\rho}))$—in NAND the displacement vector $\boldsymbol{u}$ is a function of the material variables $\boldsymbol{\rho}$. In the SAND setting however, the requirement that first-order accuracy be abandoned is somewhat dubious—since, unlike density filtering, the global stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})]$, which forms part of the constraint Jacobian matrix, is not modified too. It is therefore fair to say that *sensitivity* filtering is somewhat unjustifiable in the SAND setting.

4.4.3 Slope constraints

In Reference [60] Petersson and Sigmund introduce a restriction method which takes the form of local gradient constraints—so-called 'slope constraints'. Like filtering methods, the method

[6] In 3D, with $e_z = 100$, for example, this number is 3 090 903.
[7] In 3D this number is 6.
[8] In 3D, this is $e_r^3 - 1$.
[9] In 3D, with $e_z = 100$, this number is 2 299 631 832.

presupposes that the discrete requirement on the material distribution function $\rho(x)$ is relaxed—*i.e.* (4.8). Design set restriction is achieved by enforcing a point-wise bound on local density variations, in 2-D, this can be written as

$$\left|\frac{\partial\rho(x)}{\partial x_i}\right| \leq \mu \qquad (i=1,2)\,. \tag{4.18}$$

For a constant FE discretization and a mesh size parameter π, proportional to the size of the FE's, (4.18) may be written as

$$-\mu\pi \leq \rho_{i(k)} - \rho_{j(k)} \leq \mu\pi \quad k=1,2,\ldots,b \tag{4.19}$$

with $\rho_{i(k)}$ and $\rho_{j(k)}$ the elements which share inter element boundary number k. If slope constraints are imposed on $\mathscr{P}_c$, existence of solutions and FE convergence is ensured [60]. Moreover, the slope constraints (4.19) are linear and should be dealt with easily by a standard SAO procedure. In Reference [60] it is demonstrated that slope constraints provide a well-defined minimum length scale, lending intuitive control over the manufacturability of the design—more sophisticated manufacturing constraints, which provide for casting and extrusion processes, for example, take on a similar form [101]. However, design-set restriction with slope constraints requires the imposition of a large number of additional constraints. Petersson and Sigmund [60] report 'computational cost is extremely high' and that 'the increase in computational time lies between a factor of 100 and 1000 compared to compliance problems without slope constraints'. Bendsøe and Sigmund [2] too deems the method 'too slow for practical design problems'. In both cases the optimization problem is stated and solved in a NAND setting.

The character of the SAND problem, however—computationally speaking—is somewhat different. Consider the same $e_x \times e_y$ Q4 FE mesh from before. The FE mesh has about $4\times e_y \times e_x$ inter element boundaries, hence $8 \times e_y \times e_x$ additional constraints are required in order to enforce the slope relations[10] in (4.19). Clearly, each constraint contributes two nonzero terms to the constraint Jacobian matrix. For $e_x = e_y = 100$, 80 000 additional (linear) inequality constraints are necessitated—of which many may be inactive. This equates to the addition of 160 000 nonzero entries in the constraint Jacobian matrix—compared to millions of additional terms in the density filtering example outlined above[11]. Moreover, the number of additional nonzero terms is not dependent on the severity of the restriction—only adjacent elements are coupled—and similar relationships are demonstrable in $\mathcal{R}^3$.

The slope relation (4.18) may also take the form of a single global constraint, typically based on some norm [102], for example

$$\|\nabla\rho(x)\| \leq \bar{\mu}\,. \tag{4.20}$$

For an element-wise constant FE discretization, in 2D, (4.20) may be written as

$$\begin{aligned}\frac{1}{\sqrt{2}}\sum_{i=1}^{e_x-1}\sum_{j=1}^{e_y-1}\Big[&(\rho_{i+1,j}-\rho_{i,j})^2 + (\rho_{i+1,j+1}-\rho_{i,j+1})^2 \\ &+(\rho_{i,j+1}-\rho_{i,j})^2 + (\rho_{i+1,j+1}-\rho_{i+1,j})^2\Big]^{\frac{1}{2}} \leq \bar{\mu}\end{aligned} \tag{4.21}$$

[10] In 3D, for an $e_x \times e_y \times e_z = 100$ mesh, about $12 \times e_y \times e_x \times e_z$ additional constraints are required.

[11] In 3D, for $e_z = 100$, about 24 000 000 additional nonzero terms are required—in the equivalent density filtered case, this number is 2 299 631 832!

with (i, j) a single position in the FE grid $e_x \times e_y$. Existence of solutions may be proved, but it is apparently quite difficult to relate the limiting value $\bar{\mu}$ to the complexity of the design [2, 102]. Moreover, closer inspection reveals that, although only a single additional constraint is required, the number of additional nonzero terms in the constraint Jacobian matrix is of roughly the same order as the *local* slope constrained case (4.19). (Interestingly, considering the form of (4.12), restriction via perimeter control leads to roughly the same amount of additional nonzero terms.)

4.4.4 'Grey' material and local minima

In summary, the choice of restriction method in the SAND setting turns out to be rather straightforward. Perimeter control does not enforce a minimum length scale on the design domain and thus manufacturability might be compromised. Density filtering necessitates the coupling of large groups of material variables in order to control the complexity of the design—in the SAND setting this leads to a substantial increase in the computational storage requirements. Slope constraints, on the other hand, enforce the same effect by coupling only adjacent material variables. Moreover, it is fairly simple to modify slope constraints to account for more sophisticated manufacturing considerations [101].

Two issues remain: the imposition of slope constraints presupposes a *continuous* material–energy relation—*e.g.* SIMP in P_c—hence purely 0-1 designs can not be obtained, and secondly, the SAND problem is severely multimodal[12]. Thus, from a pragmatic point of view, mesh-independence may only be achieved with a global optimization strategy. Sigmund and Petersson [57] highlight the multimodality of the problem (nonconvexity; many local minima) as one of the 'numerical instabilities' in topology optimization. In Reference [60] the same authors implement a continuation procedure on the SIMP penalisation (for $p = 1$, P_c^s is the VTS problem, which is convex) in order to approach the global optimum of the problem. The idea behind continuation methods is to gradually change the problem from convex (having a single optimal topology) to the original (nonconvex) problem, for which a solution is actually sought.

In the NAND setting continuation on the SIMP parameter p seems to work well. In SAND however, continuation on p manifests as a gradual alteration of the feasible region of P_c^s, specifically the feasible region described by the equilibrium constraints. Formally, convergence may not be demonstrable, for, as Stolpe and Svanberg [103] have shown, the trajectory of the global optimal topology may be discontinuous even for continuously increasing penalization. Therefore, we implement a random multistart strategy to cope with the multimodality of the problem. The idea is that multiple instances of the problem can be solved, each initialized with a random material distribution $0 \leq \rho_i \leq 1$, $i = 1, 2, \ldots, e$, the topology found with the best objective function value $f_0(\boldsymbol{u})$ may then be compared to all the other topologies, if the 'best known' topology was found multiple times—assuming the algorithm is just as likely to find the global minimum as any local minimum—a measure of confidence may be associated with the global optimality of the best known solution [61]—hence, a '*probably*' globally optimal design. To address the prevalence of grey material, the algorithm may be initialized at the suspected global optimum, and by applying a continuation strategy to the slope constraint parameter μ, a globally optimal, 0-1 design may be obtained.

[12] The NAND problem is multimodal too, but seemingly less so.

4.5 The numerical method

Sequential approximate optimization (SAO) as solution strategy for general nonlinear programming involves the construction of inexpensive analytical approximation functions $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 0, 1, \ldots, m$ to the objective $f_0(\boldsymbol{x})$ and constraint functions $f_j(\boldsymbol{x})$, $j = 1, 2, \ldots, m$ at successive iteration points $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, 3, \ldots$. Together the approximations $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 0, 1, \ldots, m$ form an approximate subproblem $P[k]$. The form of the approximations $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$ are determined by the consequential solubility of the subproblem. In structural optimization it is customary to use *strictly convex* and *separable* approximations, as for example in CONLIN [34, 35] and the MMA [36]. The resulting subproblems are amenable to highly efficient dual methods when the number of constraints m are far less than the number of design variables n. When the number of constraints are relatively large, the performance of Lagrange-Newton diagonal quadratic subproblems in combination with primal-dual interior-point subsolvers are considered superior [33, 45, 46].

The general diagonal quadratic approximation [65] is used to construct the objective and constraint function approximations $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 0, 1, \ldots, m$. The approximate constraint functions $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 1, 2, \ldots, m$ are linearised before entering $P[k]$, resulting in a diagonal quadratic program (QP)—constraint curvature information retained as part of the diagonal quadratic objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$. The approximate objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ is thus viewed as an approximation of the Lagrangian and the SAO method may be termed a diagonal Lagrange-Newton sequential quadratic programming algorithm [46]. The Lagrange-Newton QP subproblem may be expressed as:

- *The QP subproblem* $P[k]$

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \tilde{f}_0^{\{k\}}(\boldsymbol{x}) = f_0(\boldsymbol{x}^{\{k\}}) + \nabla f_0^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}})^T \boldsymbol{Q}^{\{k\}}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \\ \text{subject to} \quad & \tilde{f}_j^{\{k\}}(\boldsymbol{x}) = f_j(\boldsymbol{x}^{\{k\}}) + \nabla f_j^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \; [=, \leq] \; 0 \,, \quad j = 1, 2, \ldots, m \\ & \check{x}_i^{\{k\}} \leq x_i - x_i^{\{k\}} \leq \hat{x}_i^{\{k\}} \,, \quad i = 1, 2, \ldots, n \end{aligned} \tag{4.22}$$

  with the subproblem variable bounds calculated according to

$$\check{x}_i^{\{k\}} \leftarrow \max(x_i^{\{k\}} - \delta(x_{i,\max} - x_{i,\min}), x_{i,\min}) \tag{4.23}$$

$$\hat{x}_i^{\{k\}} \leftarrow \min(x_i^{\{k\}} + \delta(x_{i,\max} - x_{i,\min}), x_{i,\max}) \tag{4.24}$$

for $i = 1, 2, \ldots, n$, with $x_{i,\min}$ and $x_{i,\max}$ the lower and upper bounds on variable i respectively. We have resorted to some nonstandard notation to denote both equality and inequality constraints—*i.e.* $[=, \leq]$.

The move limit is denoted by δ. Some numerical experiments in Section 4.6 are done with a fixed move limit—a standard value of 0.1 is used—in other experiments convergence and termination is enforced with an adaptive move limit as per the convergent trust-region algorithm due to Fletcher *et al.* [66].

The Hessian matrix $\boldsymbol{Q}^{\{k\}}$ of the approximate objective $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ is restricted to the diagonal entries—necessitating minimal storage requirements—and constructed according to

$$Q_{ii}^{\{k\}} = c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}}, \quad i = 1, 2, \ldots, n \tag{4.25}$$

with the curvature terms $c_{i,j}$ obtained as analytic, second-order partial derivatives of the objective $f_0(\boldsymbol{x})$ and the constraint functions $f_j(\boldsymbol{x})$, $j = 1, 2, \ldots, m$—second-order information is trivially computed in the SAND setting. The Lagrangian multiplier estimates from the previous iterate are retained—*i.e.* $\boldsymbol{\lambda}^{\{k\}} = \boldsymbol{\lambda}^{\{k-1\}\star}$. Assuming feasibility and constraint qualification, which turn out to be fairly mild assumptions in the SAND setting [33, 88], the only requirement is that $Q_{ii}^{\{k\}} > 0 \, \forall \, i$ for the subproblem $P[k]$ to be strictly convex. Herein the diagonal Hessian terms are constructed according to

$$Q_{ii}^{\{k\}} = \max\left(\gamma > 0, \left| c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}} \right|\right) \tag{4.26}$$

with γ a small positive number which ensures strict convexity—a standard value of 1×10^{-6} is used throughout. If the full Hessian matrix is used—like Rojas-Labanda and Stolpe [44] have done—computational storage requirements may become substantial and the enforcement of positive definiteness on the Hessian matrix (*i.e.* strict convexity) is non-trivial.

4.6 Numerical demonstrations and experiments

Based on the arguments presented in Section 4.4, we choose to furnish the SAND formulation of the SIMP topology optimization problem $\mathscr{P}_c$ with slope constraints:

- *The slope constrained SIMP topology optimization problem* $\mathscr{P}_c^s$

$$\begin{aligned} \min_{u,\rho} \quad & l(u) \\ \text{subject to} \quad & a(u,v) = l(v) \quad \forall \, v \in U \\ & E_{ijkl}(x) = \rho(x)^p E_{ijkl}^0 \\ & \int_\Omega \rho(x) \, \mathrm{d}x \leq \bar{\nu} \\ & \left| \frac{\partial \rho(x)}{\partial x_i} \right| \leq \mu \quad (i = 1, 2) \\ & 0 \leq \rho(x) \leq 1 \quad \forall \, x \in \Omega \end{aligned} \tag{4.27}$$

The FE discretized version of the problem is written as:

- *The FE discretized, slope constrained SIMP topology optimization problem* P_c^s

$$
\begin{aligned}
\min_{\boldsymbol{u},\boldsymbol{\rho}} \quad & f_0(\boldsymbol{u}) = \boldsymbol{r}^T\boldsymbol{u} \\
\text{subject to} \quad & \sum_{i=1}^{e}(\rho_i)^p[\boldsymbol{K}_i]\boldsymbol{u} - \boldsymbol{r} = 0 \\
& f_1(\boldsymbol{\rho}) = \mathbf{1}^T\boldsymbol{\rho} - \bar{\nu} \leq 0 \\
& \rho_{i(k)} - \rho_{j(k)} \leq \mu\pi\,, \quad k = 1, 2, \ldots, b \\
& \rho_{j(k)} - \rho_{i(k)} \leq \mu\pi\,, \quad k = 1, 2, \ldots, b \\
& 0 \leq \rho_i \leq 1\,, \qquad\quad i = 1, 2, \ldots, e
\end{aligned}
\tag{4.28}
$$

The stiffness design of the well-known MBB beam is considered for numerical experiment. The design domain is depicted in Figure 4.1; F is a point load of unity, the Possion ratio ν is set to 0.3 and the Young's modulus E is scaled to 100—to facilitate comparisons with other works, for example Reference [60], the objective function values $f_0(\boldsymbol{u})$ are premultiplied with 100. The amount of material is limited to a fraction $\bar{\nu} = 0.5$ of the design domain. The domain is discretized with $e_x \times e_y$ square Q8 finite elements[13]. The discretization is for half the beam, due to symmetry. The slope constraint mesh size parameter π is set to $1/e_x$. Material variables $\boldsymbol{\rho}$ are limited to 0 (*exactly*) and 1 on the lower and upper bound respectively. Displacement variables $\boldsymbol{u}$ are bounded at $\pm 1 \times 10^6$. SIMP penalisation of $p = 3$ is used throughout. The foregoing SAO procedure is employed and the subproblems $P[k]$ are solved with the GUROBI Barrier QP optimizer [76]. Computational capabilities are provided by the Rhasatsha HPC [77].

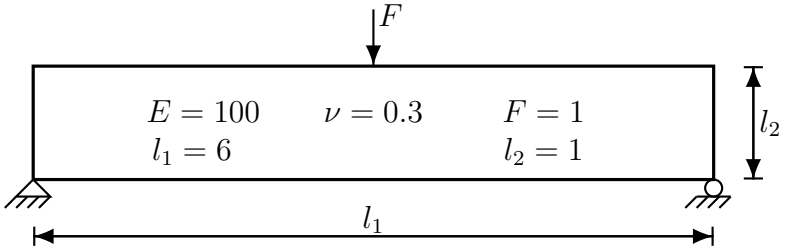


Figure 4.1: The MBB beam design domain

4.6.1 Storage and computational requirements

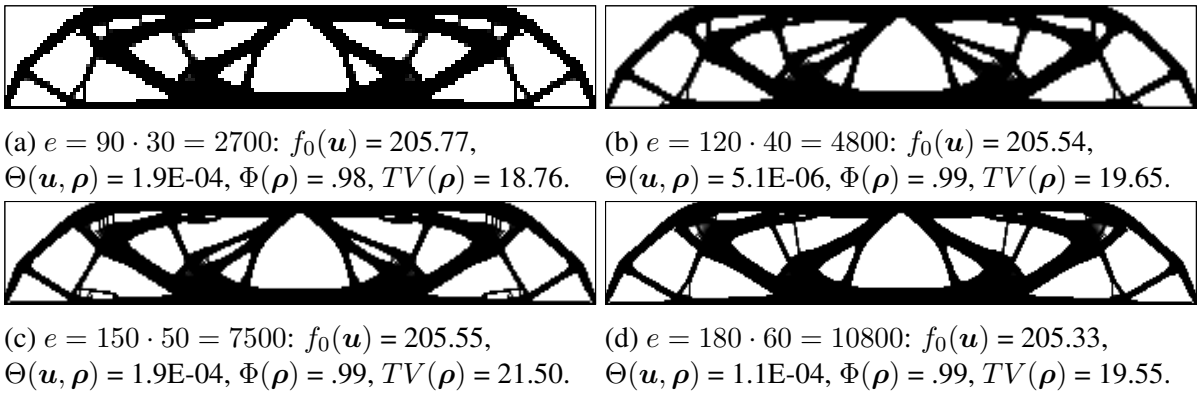
Consider the four mesh discretizations summarised in Table 4.1. In each case, the total number of elements is denoted by $e = e_x \times e_y$, b is the number of inter element boundaries, n is the number of optimization variables $\boldsymbol{x} = (\boldsymbol{\rho}, \boldsymbol{u}) \in \mathcal{R}^{n=e+d}$, m is the total number of constraints and J_{nz} is the number of nonzero entries in the constraint Jacobian matrix. The move limit is fixed at $\delta = 0.1$ and the problems are initialized at $\boldsymbol{u} = \mathbf{0}$ and $\boldsymbol{\rho} = \mathbf{0.5}$. The number of iterations

[13] We have found that the slope constraints have to be very restrictive in order to suppress checkerboards when Q4's are used.

required to achieve a convergence tolerance in terms of the Euclidean norm $\|\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}}\| \leq 1 \times 10^{-1}$ is denoted by k, T is the CPU time in seconds and T_k is the average time per iteration. The generated topologies are depicted in Figure 4.2. The largest constraint violation is denoted by $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$, $\Phi(\boldsymbol{\rho})$ is the solid-void fraction and $TV(\boldsymbol{\rho})$ is a measure of the complexity of the design—the 'total variation'— calculated as per (4.11) with $\epsilon = 0$.

Table 4.1: Storage and computational properties.

Unrestricted problem P_c									
Fig.	e	b	n	m	J_{nz}	k	T	T_k	
4.2a	2700	5280	19282	16683	557344	42	136	3.24	
4.2b	4800	9440	34242	29443	989124	48	299	6.23	
4.2c	7500	14800	53302	45803	1543904	50	526	10.52	
4.2d	10800	21360	76562	65763	2221684	46	759	16.50	
Slope constrained problem P_c^s									
Fig.	e	b	n	m	J_{nz}	k	T	T_k	
4.3a	2700	5280	19282	27243 (1.63)	578464 (1.04)	155	311	3.93	(1.21)
4.3b	4800	9440	34242	48323 (1.64)	1026884 (1.04)	168	1380	8.21	(1.32)
4.3c	7500	14800	53302	75403 (1.65)	1603104 (1.04)	222	3075	13.85	(1.32)
4.3d	10800	21360	76562	108483 (1.65)	2307124 (1.04)	272	5846	21.49	(1.30)



(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 205.77, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.9E-04, $\Phi(\boldsymbol{\rho})$ = .98, $TV(\boldsymbol{\rho})$ = 18.76.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 205.54, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 5.1E-06, $\Phi(\boldsymbol{\rho})$ = .99, $TV(\boldsymbol{\rho})$ = 19.65.

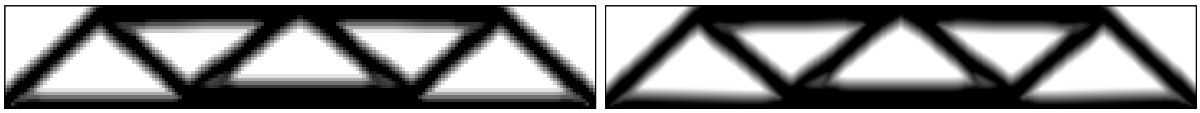
(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 205.55, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.9E-04, $\Phi(\boldsymbol{\rho})$ = .99, $TV(\boldsymbol{\rho})$ = 21.50.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 205.33, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.1E-04, $\Phi(\boldsymbol{\rho})$ = .99, $TV(\boldsymbol{\rho})$ = 19.55.

Figure 4.2: Solutions to the unrestricted problem P_c.

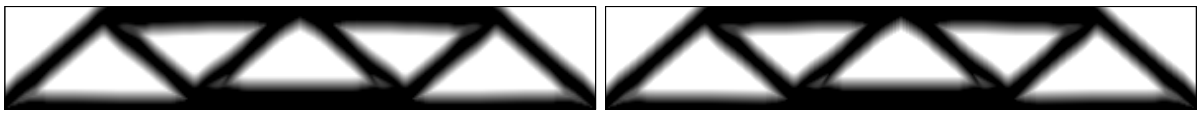
In the bottom half of Table 4.1 the storage and computational requirements of the slope constrained problem P_c^s is reported. The slope parameter μ is set to 20. Incidentally, for this set of parameters, the same minimum is approached in each case (as can be seen in Figure 4.3). In Table 4.1 The numbers in brackets are multiplication factors, denoting the relative increase *vs.* the unrestricted problem P_c. We see that the number of constraints m is increased by about 60%, although the additional nonzero terms in the constraint Jacobian matrix J_{nz} is barely noticeable. The subproblems are indeed larger and more complex, and solved with more difficulty—in the sense that the computation time per iteration T_k is increased—but this is at most a difference of about 30%. Due perhaps to the prevalence of 'grey' material and the fixed move-limit $\delta = 0.1$, the algorithm requires a substantial number of iterations to achieve convergence. (It is

suspected that the convergence properties may be improved with more sophisticated second-order approximation schemes.) Petersson and Sigmund's algorithm [60] requires 'as many as 300 iterations to convergence' (the convergence tolerance is not reported), but this is attributed, in part, to the continuation strategy on SIMP penalization p. Clearly though, in the SAND setting, the computational requirements are not increased by a factor of 100 to 1000 relative to the unrestricted case. Moreover, the storage and computational requirements for this set of test problems suggest that the algorithm scales in polynomial time—storage requirements scale linearly and average times per iteration T_k scale to roughly $\mathcal{O}(n^{1.2})$.



(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 225.95, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 5.15E-05, $\Phi(\boldsymbol{\rho})$ = .62, $TV(\boldsymbol{\rho})$ = 11.46.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 226.36, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 9.4E-06, $\Phi(\boldsymbol{\rho})$ = .62, $TV(\boldsymbol{\rho})$ = 11.57.



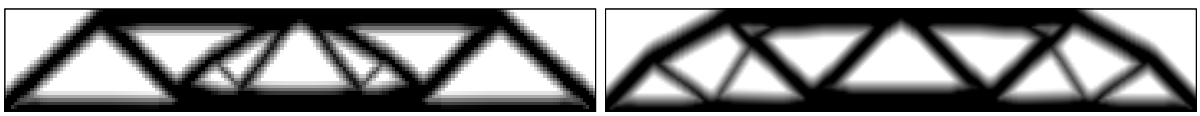
(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 226.30, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.1E-04, $\Phi(\boldsymbol{\rho})$ = .61, $TV(\boldsymbol{\rho})$ = 11.49.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 226.65, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.2E-05, $\Phi(\boldsymbol{\rho})$ = .62, $TV(\boldsymbol{\rho})$ = 11.55.

Figure 4.3: Solutions to the restricted problem P_c^s with $\mu = 20$.

4.6.2 'Probably' globally optimal designs

In both the NAND and SAND setting, small variations in algorithmic parameters (and starting positions) may result in convergence to different local minima. (It is suspected that the issue of local minima is more severe in the SAND setting.) For example, consider the slope constrained problem P_c^s with $p = 3$ and $\mu = 20$—exactly the same problem as considered in Section 4.6.1—using different starting positions, the algorithm may converge to the topologies depicted in Figure 4.4. For higher values of μ, which allows for increased complexity in the optimum design—thus, an increased number of topologies are feasible—it is quite difficult to reliably find the suspected global optimum. To remedy this, the random multistart strategy outlined in Section 4.4.4 is employed. Because the algorithm may struggle to maintain feasibility from a random starting position, convergence and termination is enforced with an adaptive move-limit δ, as per the convergent trust-region algorithm due to Fletcher *et al.* [66].



(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 228.51, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.2E-05, $\Phi(\boldsymbol{\rho})$ = .58, $TV(\boldsymbol{\rho})$ = 13.01.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 229.39, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.0E-06, $\Phi(\boldsymbol{\rho})$ = .57, $TV(\boldsymbol{\rho})$ = 13.44.

Figure 4.4: Suboptimal solutions to the restricted problem P_c^s with $\mu = 20$

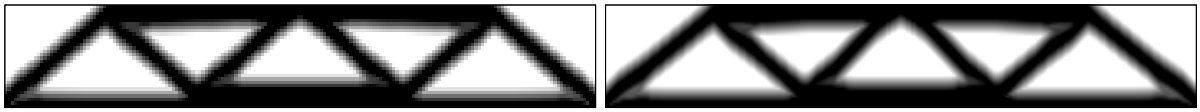
Table 4.2 is a summary of the data obtained from a total of 4600 experiments—*i.e.* randomly initialized instances of P_c^s—for 4 mesh discretizations and a range of slope parameters μ. (These experiments may of course be run in parallel.) The number of experiments per mesh size $e_x \times e_y$ and for a specific slope parameter μ is denoted by ζ^0, $\zeta^\star$ is the number of times the algorithm converged to the best known solution—measured in terms of the absolute difference in objective function values $|f_0^{\text{best}} - f_0(\boldsymbol{u})| \leq 0.1$—the total number of subproblems the computational platform is tasked with is denoted by P, and T_P is the average CPU time required to solve a subproblem (in seconds)[14]. The optimum topology associated with each set of parameters is depicted in Figures 4.5 through 4.10.

Table 4.2: Multistart statistics; mesh-independnce

Figure	$e_x \cdot e_y = e$	$\mu\pi$	$f_0(\boldsymbol{u})$	ζ^0	$\zeta^\star$	P	T_P
4.5a	$90 \cdot 30 = 2700$	$20/90 = 0.22$	225.71	200	7	32890	7
4.6a	$90 \cdot 30 = 2700$	$30/90 = 0.33$	212.51	200	1	42284	6
4.7a	$90 \cdot 30 = 2700$	$35/90 = 0.39$	210.19	300	2	55792	4
4.10a	$90 \cdot 30 = 2700$	$55/90 = 0.61$	202.12	300	2	71699	6
4.5b	$120 \cdot 40 = 4800$	$20/120 = 0.17$	225.70	200	11	44329	14
4.6b	$120 \cdot 40 = 4800$	$30/120 = 0.25$	212.90	200	1	52154	9
4.8a	$120 \cdot 40 = 4800$	$40/120 = 0.33$	206.93	300	2	83887	11
4.9a	$120 \cdot 40 = 4800$	$45/120 = 0.38$	204.81	300	2	73373	8
4.10b	$120 \cdot 40 = 4800$	$55/120 = 0.46$	201.61	300	4	61261	10
4.5c	$150 \cdot 50 = 7500$	$20/150 = 0.13$	226.21	200	23	65368	15
4.6c	$150 \cdot 50 = 7500$	$30/150 = 0.20$	213.05	200	2	69326	20
4.8b	$150 \cdot 50 = 7500$	$40/150 = 0.27$	207.34	300	5	87028	19
4.9b	$150 \cdot 50 = 7500$	$45/150 = 0.30$	204.92	300	2	83248	17
4.10c	$150 \cdot 50 = 7500$	$55/150 = 0.37$	201.63	300	4	93178	19
4.5d	$180 \cdot 60 = 10800$	$20/180 = 0.11$	226.51	200	12	80533	37
4.6d	$180 \cdot 60 = 10800$	$30/180 = 0.17$	213.13	200	2	78983	20
4.7b	$180 \cdot 60 = 10800$	$35/180 = 0.19$	210.21	300	2	94569	29
4.10d	$180 \cdot 60 = 10800$	$55/180 = 0.31$	201.41	300	1	85894	29

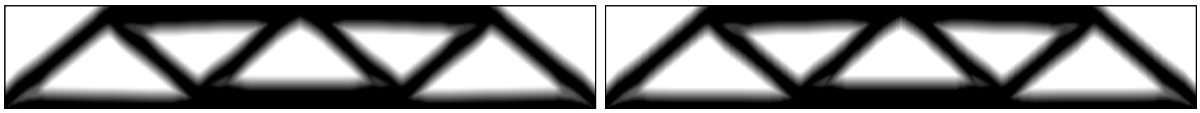
A number of observations may be made from the data reported in Table 4.2, and by inspection of Figures 4.5 to 4.10. First, as expected, more complex and more efficient topologies—characterised by an increased number of members—are available for less restrictive values of the slope parameter μ—as can be seen from the objective function value $f_0(\boldsymbol{u})$ and perimeter measure $TV(\boldsymbol{\rho})$ of the best-known solution in each case. However, there is a trade-off, because an increased number of topologies are available for less restrictive values of μ, the frequency with which the best-known solution is found, diminishes: for $\mu = 20$, the best-known solution is obtained, on average, about 13 times out of 200; for $\mu = 30$, the frequency is once or twice for every 200 experiments. For larger values of μ, more experiments are required to gain confidence in the suspected global optimality of the best-known solution, for $\mu = 35, 40, 45$ and

[14]Here we refer to 'subproblems' rather than 'iterations', because the convergent trust-region algorithm [66] involves 'inner iterations'.



(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 225.71, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 7.2E-06, $\Phi(\boldsymbol{\rho})$ = .62, $TV(\boldsymbol{\rho})$ = 11.40.

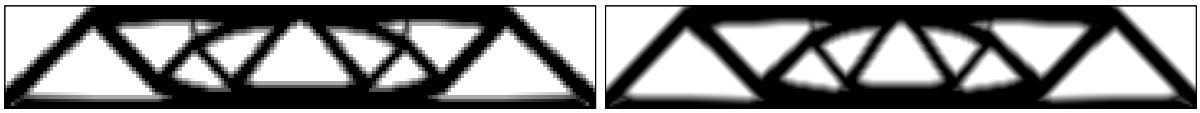
(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 225.70, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.7E-06, $\Phi(\boldsymbol{\rho})$ = .64, $TV(\boldsymbol{\rho})$ = 11.34.



(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 226.21, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.2E-05, $\Phi(\boldsymbol{\rho})$ = .61, $TV(\boldsymbol{\rho})$ = 11.43.

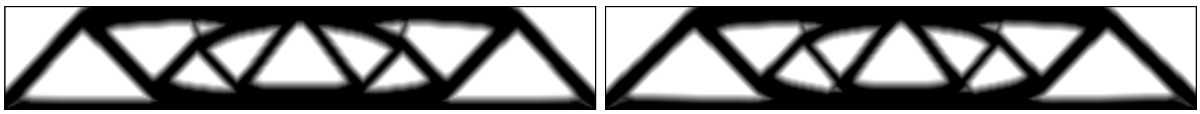
(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 226.51, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.3E-06, $\Phi(\boldsymbol{\rho})$ = .62, $TV(\boldsymbol{\rho})$ = 11.49.

Figure 4.5: Probably globally optimal solutions to the restricted problem P_c^s with $\mu = 20$.



(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 212.51, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 8.8E-05, $\Phi(\boldsymbol{\rho})$ = .75, $TV(\boldsymbol{\rho})$ = 14.11.

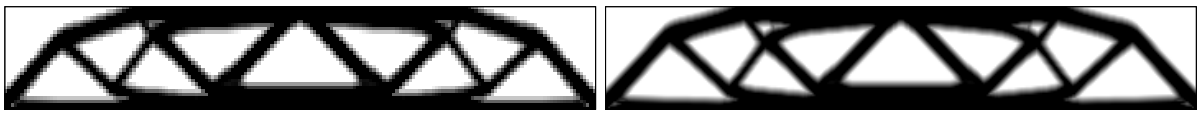
(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 212.90, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 6.1E-06, $\Phi(\boldsymbol{\rho})$ = .73, $TV(\boldsymbol{\rho})$ = 14.14.



(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 213.05, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 9.0E-05, $\Phi(\boldsymbol{\rho})$ = .72, $TV(\boldsymbol{\rho})$ = 14.42.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 213.13, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.0E-05, $\Phi(\boldsymbol{\rho})$ = .72, $TV(\boldsymbol{\rho})$ = 14.12.

Figure 4.6: Probably globally optimal solutions to the restricted problem P_c^s with $\mu = 30$.



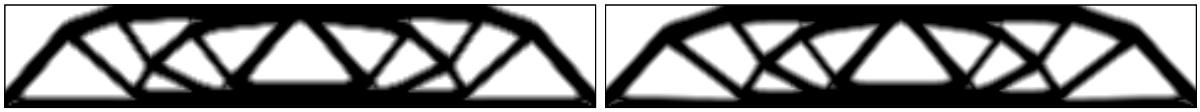
(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 210.19, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.4E-04, $\Phi(\boldsymbol{\rho})$ = .75, $TV(\boldsymbol{\rho})$ = 14.83.

(b) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 210.21, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.5E-05, $\Phi(\boldsymbol{\rho})$ = .72, $TV(\boldsymbol{\rho})$ = 15.22.

Figure 4.7: Probably globally optimal solutions to the restricted problem P_c^s with $\mu = 35$.

55, 300 experiments are conducted. As per the methodology presented in Reference [61], 2 successful runs out of 300 equate to a confidence measure of about 87%, 3 out of 300 is about 94%, and 4 out of 300 is about 97%.

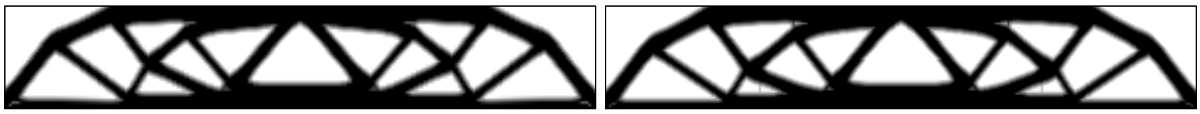
Figures 4.5 through 4.10 illustrate how the suspected optimal topology changes relative to the slope parameter μ. The topological layouts correspond well with those reported in Reference [60]. Moreover, as expected, 'grey' material is less prevalent for less restrictive values of μ (since 'grey' material is uneconomical in the presence of the volume constraint).



(a) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 206.93, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 9.5E-06, $\Phi(\boldsymbol{\rho})$ = .77, $TV(\boldsymbol{\rho})$ = 16.78.

(b) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 207.34, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.2E-04, $\Phi(\boldsymbol{\rho})$ = .75, $TV(\boldsymbol{\rho})$ = 16.78.

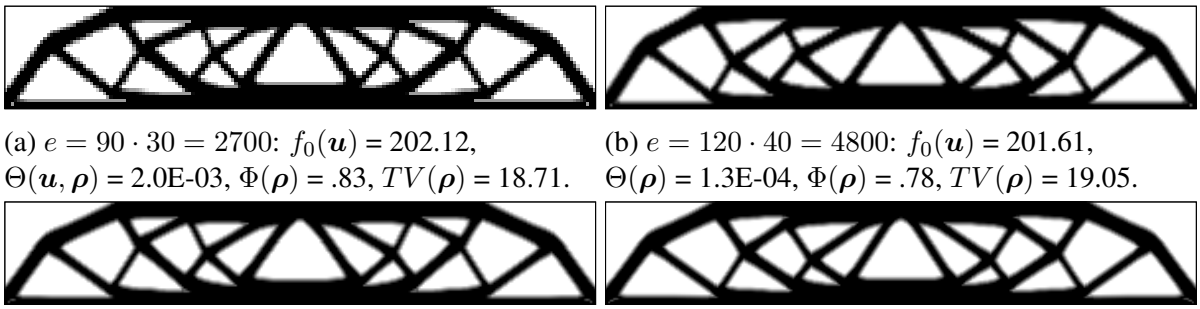
Figure 4.8: Probably globally optimal solutions to the restricted problem P_c^s with $\mu = 40$.



(a) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 204.81, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.6E-05, $\Phi(\boldsymbol{\rho})$ = .77, $TV(\boldsymbol{\rho})$ = 17.18.

(b) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 204.92, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.3E-04, $\Phi(\boldsymbol{\rho})$ = .76, $TV(\boldsymbol{\rho})$ = 16.90.

Figure 4.9: Probably globally optimal solutions to the restricted problem P_c^s with $\mu = 45$.



(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 202.12, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.0E-03, $\Phi(\boldsymbol{\rho})$ = .83, $TV(\boldsymbol{\rho})$ = 18.71.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 201.61, $\Theta(\boldsymbol{\rho})$ = 1.3E-04, $\Phi(\boldsymbol{\rho})$ = .78, $TV(\boldsymbol{\rho})$ = 19.05.

(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 201.63, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.8E-05, $\Phi(\boldsymbol{\rho})$ = .80, $TV(\boldsymbol{\rho})$ = 18.68.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 201.41, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.5E-05, $\Phi(\boldsymbol{\rho})$ = .78, $TV(\boldsymbol{\rho})$ = 19.04.

Figure 4.10: Probably globally optimal solutions to the restricted problem P_c^s with $\mu = 55$.

4.6.3 'Probably' globally optimal 0-1 designs

To reduce the prevalence of 'grey' material, the slope parameter μ may be relaxed with a continuation strategy. To this end, the algorithm is initialized with a 'probably' globally optimal solution to P_c^s (for some value of μ), and the slope parameter is increased by 1.1μ if the update in the optimization variables satisfy the convergence tolerance $\|\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}}\| \leq 1 \times 10^{-1}$. Once $\mu\pi \geq 1$ (and problem P_c is actually solved) the algorithm is allowed to terminate.

Initializing the algorithm with some of the restricted global optima presented in the preceding section, and following the continuation strategy outlined above, the designs depicted in Figures 4.11 to 4.13 are generated. As expected, the objective function $f_0(\boldsymbol{u})$ values and solid-void fractions $\Phi(\boldsymbol{\rho})$ are much improved, while the perimeter $TV(\boldsymbol{\rho})$—a measure of the complexity of the design—remains constant, roughly, in terms of the restricted *vs.* unrestricted designs.

(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 195.62, k = 153, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 8.4E-05, $\Phi(\boldsymbol{\rho})$ = .99, $TV(\boldsymbol{\rho})$ = 11.73.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 196.11, k = 202, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.0E-06, $\Phi(\boldsymbol{\rho})$ = .93, $TV(\boldsymbol{\rho})$ = 11.95.

(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 194.89, k = 233, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 3.0E-04, $\Phi(\boldsymbol{\rho})$ = .98, $TV(\boldsymbol{\rho})$ = 12.27.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 194.80, k = 229, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 8.1E-06, $\Phi(\boldsymbol{\rho})$ = .94, $TV(\boldsymbol{\rho})$ = 12.29.

Figure 4.11: Probably globally optimal, approximately 0-1 solutions to the unrestricted problem P_c, for an initial slope parameter $\mu = 20$.

(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 193.25, k = 117, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.2E-05, $\Phi(\boldsymbol{\rho})$ = .89, $TV(\boldsymbol{\rho})$ = 14.64.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 192.31, k = 200, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 5.7E-07, $\Phi(\boldsymbol{\rho})$ = .92, $TV(\boldsymbol{\rho})$ = 15.03.

(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 190.69, k = 213, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 7.2E-04, $\Phi(\boldsymbol{\rho})$ = .99, $TV(\boldsymbol{\rho})$ = 15.38.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 190.71, k = 268, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 5.3E-04, $\Phi(\boldsymbol{\rho})$ = .99, $TV(\boldsymbol{\rho})$ = 15.35.

Figure 4.12: Probably globally optimal, approximately 0-1 solutions to the unrestricted problem P_c, for an initial slope parameter $\mu = 30$.

(a) $e = 90 \cdot 30 = 2700$: $f_0(\boldsymbol{u})$ = 191.87, k = 59 $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 6.0E-04, $\Phi(\boldsymbol{\rho})$ = .86, $TV(\boldsymbol{\rho})$ = 18.79.

(b) $e = 120 \cdot 40 = 4800$: $f_0(\boldsymbol{u})$ = 189.85, k = 134 $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 7.6E-04, $\Phi(\boldsymbol{\rho})$ = .89, $TV(\boldsymbol{\rho})$ = 19.46.

(c) $e = 150 \cdot 50 = 7500$: $f_0(\boldsymbol{u})$ = 189.84, k = 116 $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 6.6E-07, $\Phi(\boldsymbol{\rho})$ = .91, $TV(\boldsymbol{\rho})$ = 19.24.

(d) $e = 180 \cdot 60 = 10800$: $f_0(\boldsymbol{u})$ = 188.01, k = 203 $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.8E-04, $\Phi(\boldsymbol{\rho})$ = .98, $TV(\boldsymbol{\rho})$ = 19.62.

Figure 4.13: Probably globally optimal, approximately 0-1 solutions to the unrestricted problem P_c, for an initial slope parameter $\mu = 55$.

4.7 Conclusion

Based on straightforward computational considerations, it is argued that the SAND topology optimization problem should be equipped with so-called 'slope constraints' in order to ensure existence of solutions, mesh-independence, and manufacturability of the design. Density filtering techniques, which are very popular and successful in the traditional NAND setting, may be unnecessarily expensive in storage and computational requirements in the unconventional SAND setting. Moreover, computational storage requirements of the slope constrained SAND problem scale linearly with problem size, and using an efficient (sparse) sequential approximate optimization method, computation times too may scale favourably.

It is demonstrated that a random multistart strategy may be used to cope with the multimodality of the SAND optimization problem—and thereby demonstrate mesh-independence. The multistart technique allows one to associate a measure of confidence with the suspected global optimality of the best-known solution—hence a 'probably' globally optimal solution is obtained. Moreover, the slope constraints may be relaxed with a continuation strategy to arrive at manufacturable, 'probably' globally optimal, 0-1 design in a reasonable amount of computation time.

Chapter 5

Local stress– and slope-constrained SAND topology optimization

This chapter is a reproduction of a paper entitled 'Local stress- and slope-constrained SAND topology optimization' [88]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical and Mechatronic Engineering at the University of Stellenbosch, South Africa.

5.1 Abstract

We study the alternative 'simultaneous analysis and design' (SAND) formulation of the local stress- and slope-constrained topology design problem. It is demonstrated that a standard trust-region Lagrange-Newton sequential quadratic programming (SQP)-type algorithm—based, in this case, on strictly convex and separable approximate subproblems—may converge to singular optima of the local stress-constrained problem without having to resort to relaxation or perturbation techniques. Moreover, due to the negation of the sensitivity analyses—in SAND the density and displacement variables are independent—and the immense sparsity of the SAND problem, solutions to large-scale problem instances may be obtained in a reasonable amount of computation time.

5.2 Introduction

Topology optimization is an algorithmic approach to a pervasive engineering problem: the distribution of a limited resource to an optimal end. In classical topology design of structures the goal is to determine the least compliant material distribution given a limited amount of material. In general however the *least compliant* material distribution is not the *strongest* material distribution: the structural topology of minimum weight (lightest) which can support the applied loads. In order to support the applied loads the topology has to prevent material failure—typically the end of the elastic regime—at each and every point in the structure. This may be achieved with local stress constraints. Bendsøe and Sigmund [2] have noted that 'imposing stress constraints on topology optimization problems is an extremely important topic', and that

'the best way to solve stress constrained problems has probably yet to be suggested' (2003). The interested reader can refer to Duysinx *et al.* [104] for 'why stress constraints are so important' and Pereira *et al.* [105], for example, to view comparative experiments with compliance and strength objectives.

In principle the topology design problem is a mixed integer program (material is either present or absent, solid or void, 1 or 0, at each and every spatial position). In the 0-1 formulation stress constraints are well-defined, and although notable progress has been made by Stolpe *et al.* [106–108] in terms of the problem formulation and numerical techniques, the highly combinatorial nature of the problem seems to prohibit reasonable computation times. In order to avoid expensive integer programming techniques, and in turn permit the use of efficient gradient-based optimization methods, it is customary to relax the discrete requirement on the material variables and implement an intermediate material stiffness interpolation—typically SIMP [54, 55]. Material variables are then interpreted as material 'density' variables which denote the amount of material present at each and every spatial position. If stress constraints are to be physically consistent, the limited stress quantity should mimic that of a porous, composite microstructure: local stresses are finite and nonzero as material is removed to the point of zero density; or the 'coherency condition' due to Duysinx and Bendsøe [109].

Duysinx and Bendsøe [109] effectively generalized stress-constrained topology design of truss-like structures to the continuum (or 'bidimensional') setting—based in turn on the material distribution problem due to Bendsøe [54]. It is demonstrated that the stress singularity problem—the problem of 'singular optima', traditionally observed in the context of truss structures [110–114]—transpires in the continuum setting too [109]. The issue is that stress is undefined at spatial positions where no material is present ('zero density' or 'void material') and the indeterminant (undefined) stresses might exceed the stress limit at suitable optima. In other words, the stress constraint function is discontinuous: stress constraints have to be 'switched off' as material is removed to the point of zero density to permit void stresses to exceed the stress limit. The effect is that standard, gradient-based optimization methods typically converge to unsuitable local minima characterised by fully stressed low density 'membranes' in between structural elements. It is typically said that suitable (singular) optima are located in 'degenerate subdomains' of 'zero measure'—or 'jelly-fish like' feasible domains [114]—where standard constraint qualifications (CQ's) do not hold. To remedy the singularity problem it is customary to relax (perturb) stress constraints, opening up the feasible domain in order for gradient-based optimization algorithms to reach suitable optima. The predominant techniques are known as ϵ-relaxation [113] (typically accompanied by a continuation strategy) and qp-relaxation [115] (a relaxation of the coherency condition itself).

Stolpe and Svanberg [116] have demonstrated that by using a 'disaggregated' formulation—which is related to 'direct', 'simultaneous analysis and design' (SAND) [49, 91] and 'displacement-based' formulations [52]—gradient-based optimization methods may find 'singular optima' without having to resort to perturbation techniques. Recently studies dedicated to the formulation of stress constraints as 'vanishing constraints' in the SAND setting—wherein zero density material is permitted because the singularity of the global stiffness matrix does not arise as an issue—have appeared too, detailing an ϵ-like relaxation method, specific CQ's, and optimality conditions for mathematical programs with vanishing constraints (MPVC) [117–119]. These findings form the first motivation for the SAND local stress-constrained formulation

presented herein—the possibility to do away with cumbersome relaxation and continuation strategies—the second motivation is related to the inherent large-scale nature of the local stress-constrained topology design problem.

In spite of efficient gradient-based optimization techniques, the local stress-constrained SIMP topology design problem is characterised by a massive computational burden. In order to limit stresses on a local scale, a stress constraint has to be imposed at each and every spatial position. Alternatively one may resort to global, aggregated or regional stress constraints [120–122], but material failure is a local phenomenon, and it is often difficult to guarantee that all the local stresses will be limited to the critical value. The computational burden related to the 'local nature' of stress-constrained topology design is one of two main issues identified by Duysinx and Bendsøe [109]—the other is the coherency requirement—and since that time published numerical experiments have progressively increased in size, with authors making use of a wide range of strategies and procedures to reduce the computational burden and improve the quality of the solutions. Pereira *et al.* [105], for example, employed an efficient three-node Lagrangian finite element (FE) mesh discretization and an augmented Lagrangian solution procedure. Bruggi and Venini [123] devised a mixed FE approach and París *et al.* [124] were able to solve large-scale problems—about 10 000 Q8 FE's—using parallel computation techniques. Bruggi and Duysinx [43] too have considered discretizations in the order of 10 000 FE's. Other recent developments include the level set method by Emmendoerfer and Fancello [125] and the so-called 'damage approach' by Verbart *et al.* [126].

Each and every large-scale study cited above is based on the 'reduced', 'nested analysis and design' (NAND) or 'conventional' formulation of the problem, as it is referred to by Arora and Wang [49, 91]. In the NAND formulation nodal displacement (state) variables are removed from the optimization problem by a procedure equivalent to the analytic substitution of the FE equilibrium equation. In common computational implementations, this manifests as an independent (nested) numerical procedure devoted to the solution of the FE system—the 'structural analysis phase' or 'FE simulation call'. In local stress-constrained problems this causes a dense coupling between displacement variables and material density variables, necessitating multiple (computationally expensive) sensitivity analyses in the nested algorithm. The sensitivity analysis is typically achieved with the adjoint variable method [2]. Bruggi and Duysinx [43] note that, in spite of the imposition of an extra compliance constraint devised in order to reduce the number of active stress constraints, the 'level of reduction might not be be enough to provide an efficient methodology for such kind of problems (*mainly due to the cost of the adjoint analysis*)' [our emphasis]. In fact, more than a decade prior, Duysinx and Bendsøe [109] concluded that 'new implementations' dedicated to 'very large-scale problems' should be pursued.

The alternative SAND formulation may turn out to be a tenable alternative for large-scale stress-constrained topology design. Wang and Arora [91] recognized and demonstrated this in the context of massive truss structures. They note that 'SAND represents a fundamental shift in the way analysis and design problems are currently treated' [91]. The main reason for this is that the FE system is part of a unified SAND optimization problem in the form of a set of nonlinear equality constraints. That is, material density variables and nodal displacement variables are independent, hence all gradient information reduces to easy to calculate (and typically sparse) partial derivatives. Furthermore, as mentioned above, in the SAND setting stress constraints may be formulated as vanishing constraints, and because a lower bound of exactly zero is per-

mitted on the material density variables, singular optima may be converged to without having to resort to relaxation or perturbation techniques. The work presented here is based on our previous efforts in the context of classical topology design: in Reference [33] the equivalences between sequential approximate optimization (SAO) in the SAND setting and conventional NAND SAO are demonstrated; in Reference [59] the compounded issues of existence of solutions, mesh dependency and local minima are addressed in the SAND setting.

It is not our intention to argue that the SAND formulation is definitively superior to the conventional NAND formulation—for one, SAND problems are more complex, and characterised by a larger number of optimization variables and constraints, which includes a large set of nonlinear equality constraints. However, it will be disingenuous to ignore some theoretical considerations and numerical results which suggest that the SAND setting may be inherently more suitable to local stress-constrained topology design, simply because it is characterised by the alleviation of the immense computational burden associated with the sensitivity analyses, in addition to the negation of the stress singularity problem. The rest of the paper reads as follows: in Section 5.3.1 the minimum weight problem subject to local stress constraints is defined, in Section 5.3.2 the infamous stress singularity problem is recited, followed by a summary of relevant CQ's in Section 5.3.3, and a simple complexity analysis in Section 5.3.4, in Section 5.4 the preferred numerical method is outlined, Section 5.5 chronicles a range of numerical demonstrations and experiments, and Section 5.6 concludes.

5.3 The problem of minimum weight subject to local stress constraints

5.3.1 Problem formulation

Herein the original (SAND-like) formulation of the SIMP minimum weight problem subject to local stress constraints—Problem $\mathscr{P}_{\mathrm{S}}$ (below)—is considered. Slope constraints [60] are introduced to ensure existence of solutions and mesh-independence of the FE discretized problem—Problem P_{S} (further below). For a constant FE discretization and a mesh size parameter π, proportional to the size of the FE's, the slope parameter μ controls the complexity of the design [59, 60]. The predefined design domain is denoted by Ω, x is the spatial position and $\rho(x)$ is the material distribution function. The equilibrium condition is shown here in the energy bilinear form $a(u, v) = l(v)$. For an equilibrium displacement u and an arbitrary, kinematically admissible virtual displacement v, $a(u, v)$ is the internal virtual work; $l(v)$ is the virtual compliance. The local stress measure is denoted by $\sigma(u, \rho)$ and $\bar{\sigma}$ is the stress limit. As is customary, the SIMP [54, 55] proportional stiffness model $\rho(x)^p E^0_{ijkl}$ is employed: E^0_{ijkl} is the stiffness tensor of a given (solid) isotropic material and p is the SIMP exponent.

□ *The SIMP minimum weight problem subject to local stress- and slope constraints* $\mathscr{P}_{\text{S}}$:

$$\begin{aligned} \min_{u,\rho} \quad & \int_\Omega \rho(x)\,\mathrm{d}x \\ \text{subject to} \quad & a(u,v) = l(v) \quad \forall\, v \in U \\ & \sigma(u,\rho) \leq \bar{\sigma} \\ & E_{ijkl}(x) = \rho(x)^p E^0_{ijkl} \\ & \left|\frac{\partial \rho(x)}{\partial x_i}\right| \leq \mu \quad (i=1,2) \\ & 0 \leq \rho(x) \leq 1 \quad \forall\, x \in \Omega \end{aligned} \tag{5.1}$$

□ *The FE discretized, SIMP minimum weight problem subject to local stress- and slope constraints* P_{S} :

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{\rho}} \quad & \sum_{i=1}^{e} \frac{\rho_i}{e} \\ \text{subject to} \quad & [\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} - \boldsymbol{r} = \boldsymbol{0} \\ & \rho_i \left[\frac{\sigma_i(\boldsymbol{u})}{\bar{\sigma}} - 1\right] \leq 0 \qquad i = 1,2,\ldots,e \\ & \rho_{i(k)} - \rho_{j(k)} \leq \mu\pi \qquad k = 1,2,\ldots,b \\ & \rho_{j(k)} - \rho_{i(k)} \leq \mu\pi \qquad k = 1,2,\ldots,b \\ & 0 \leq \rho_i \leq 1 \qquad i = 1,2,\ldots,e \end{aligned} \tag{5.2}$$

The design domain is discretized with e FE's with b the number of inter element boundaries. Each FE is assigned a material density variable ρ_i and a stress measure $\sigma_i(\boldsymbol{u})$, $i = 1,2,\ldots,e$. The $\rho_{i(k)}$ and $\rho_{j(k)}$ represent the elements which share inter element boundary number k for $k = 1,2,\ldots,b$. The vector of material density variables is denoted by $\boldsymbol{\rho} \in \mathbb{R}^e$. The vector of nodal displacements is $\boldsymbol{u} \in \mathbb{R}^d$, where d is the number of degrees of freedom. The globally assembled, SIMP modified stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})] = \sum_{i=1}^{e}(\rho_i)^p[\boldsymbol{K}_i]$ has dimension $d \times d$.

The vector of primal optimization variables is $(\boldsymbol{u},\boldsymbol{\rho}) \in \mathbb{R}^{n=d+e}$. Problem P_{S} has $m = d+e+2b$ constraints, of which d are (nonlinear) equality constraints: $[\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} - \boldsymbol{r} = \boldsymbol{0} \in \mathbb{R}^d$.

5.3.2 The stress singularity problem

A singularity is a point where a mathematical object is ill-defined, or fails to be well-behaved in a particular way. Structural stress is undefined in 'void' material—material with 'zero density' $\rho_i = 0$. Therefore, the stress constraint on element i may only be imposed if $\rho_i > 0$. In other words, one may write the local stress constraint in conditional form

$$\frac{\sigma_i(\boldsymbol{u})}{\bar{\sigma}} - 1 \leq 0 \quad \text{if} \quad \rho_i > 0\,. \tag{5.3}$$

That is, $\sigma_i(\boldsymbol{u})$ can take on *any* value if $\rho_i = 0$. The same effect is achieved—but with nicer properties since a derivative with respect to ρ_i is introduced—if the local stress constraint (5.3) is rather formulated as a 'vanishing constraint' [117–119]

$$\rho_i \left[\frac{\sigma_i(\boldsymbol{u})}{\bar{\sigma}} - 1 \right] \leq 0 \,. \tag{5.4}$$

Constraint (5.4) is reminiscent of a 'force constraint'-like formulation typically found in the context of truss structures [111], for continuum structures the additional ρ_i term is grouped with a larger class of 'quality functions' [112]. Cheng and Guo study trusses and show that the discontinuity of the stress constraint function (5.4) at $\rho_i = 0$ gives rise to degenerate 'jellyfish-like' feasible subdomains [114]. As mentioned earlier, Duysinx and Bendsøe [109] demonstrate that local stress is finite and nonzero in the continuum setting for $\rho_i \to 0$, giving rise to the same issue.

Material density variables in NAND-based topology design formulations typically have to be limited to a small, nonzero value $\rho_i \geq \rho_{\text{min}} > 0$, $i = 1, 2, \ldots, e$, in order to to avoid the singularity of the global stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})]$. Therefore, the feasible region of the local stress constraint (5.4) at $\rho_i = 0$, albeit degenerate, is not part of the feasible domain of the NAND formulated problem. To remedy this the stress constraint (5.4) may be relaxed (perturbed) with a parameter dependent on the density lower bound ρ_{min}. The ϵ-relaxation approach [113] proposes the following modification

$$\rho_i \left[\frac{\sigma_i(\boldsymbol{u})}{\bar{\sigma}} - 1 \right] \leq \epsilon \quad \text{with} \quad \epsilon^2 = \rho_{\text{min}} \,. \tag{5.5}$$

This approach is also employed by Duysinx and Bendsøe [109][1]. The perturbation of the stress constraint opens up the feasible domain, permitting general-purpose optimization algorithms to reach suitable solutions. Typically a continuation strategy is employed so that a solution to the original, unperturbed problem (or a problem of close resemblance) may be obtained[2]. A similar relaxation technique devised for purely vanishing constraints (5.5) (*i.e.* $\rho_{\text{min}} = 0$) can be found in References [118, 119]. The authors demonstrate convergence of the sequence of stationary points of the perturbed problems to a suitable stationary point of the original problem, however, numerical demonstrations rely on judicious choices of the relaxation parameter.

An alternative perturbation technique due to Bruggi [115] is employed in the consideration of large-scale problem instances in Reference [43]. The formulation tackles the discontinuity of the stress constraint function by relaxing the coherency condition [109]. The stress limit $\bar{\sigma}$ is in fact also interpolated with SIMP power-law $\rho_i^q \bar{\sigma}$

$$\rho_i^{p-q} \frac{\sigma_i(\boldsymbol{u})}{\bar{\sigma}} - 1 \leq 0 \,. \tag{5.7}$$

[1] In the NAND setting the nonzero lower bound on material density variables ρ_{min} allows for a reformulation of the stress constraint function which suites the use of popular reciprocal intervening variables:

$$\frac{\sigma_i(\boldsymbol{u})}{\bar{\sigma}} - \frac{\epsilon}{\rho_i} - 1 \leq 0 \,. \tag{5.6}$$

If zero density material is permitted $\rho_{\text{min}} = 0$ this formulation is of course untenable (due to the division by zero). Moreover, the use of intervening variables seems somewhat misplaced in the SAND setting [33].

[2] Stolpe and Svanberg [127] show that ϵ continuation may be problematic in the presence of many local minima.

The coherency condition is satisfied if $q = p$ [109]. The 'qp-approach' [115] works by setting $q < p$. Thus, the first term in (5.7) is zero at zero density $\rho_i = 0$. Bruggi [115] notes that the formulation is not physically consistent and purely a mathematical manipulation of the problem. Nevertheless, the method can be assimilated with an adaptive ϵ-relaxation approach, stress constraints on full density elements are not perturbed and with appropriate initialization and continuation of q the method is shown to have good convergence properties. We hope to test various stress constraint formulations in the SAND setting in future, herein however we limit our investigation to the preperturbed—and physically consistent—vanishing local stress constraint (5.4).

5.3.3 Constraint qualification

As per References [117, 118], let $\boldsymbol{x}^\star = (\boldsymbol{u}^\star, \boldsymbol{\rho}^\star)$ be a feasible point of P_S, and define the index sets of nonzero and zero material density elements:

$$I_+ = \{i = 1, 2, \ldots, e \mid \rho_i^\star > 0\} \tag{5.8}$$
$$I_0 = \{i = 1, 2, \ldots, e \mid \rho_i^\star = 0\}\,. \tag{5.9}$$

The set of nonzero elements I_+ can be partitioned into subsets of active and feasible local stress constraints:

$$I_{+0} = \{i = 1, 2, \ldots, e \mid \rho_i^\star > 0, [\sigma_i(\boldsymbol{u})/\bar{\sigma} - 1] = 0\} \tag{5.10}$$
$$I_{+-} = \{i = 1, 2, \ldots, e \mid \rho_i^\star > 0, [\sigma_i(\boldsymbol{u})/\bar{\sigma} - 1] < 0\}\,. \tag{5.11}$$

and similarly, the set of void elements I_0 can be partitioned into subsets of infeasible, active and feasible local stress constraints:

$$I_{0+} = \{i = 1, 2, \ldots, e \mid \rho_i^\star = 0, [\sigma_i(\boldsymbol{u})/\bar{\sigma} - 1] > 0\} \tag{5.12}$$
$$I_{00} = \{i = 1, 2, \ldots, e \mid \rho_i^\star = 0, [\sigma_i(\boldsymbol{u})/\bar{\sigma} - 1] = 0\} \tag{5.13}$$
$$I_{0-} = \{i = 1, 2, \ldots, e \mid \rho_i^\star = 0, [\sigma_i(\boldsymbol{u})/\bar{\sigma} - 1] < 0\}. \tag{5.14}$$

Achtziger and Kanzow [117] show that if $I_0 \neq \emptyset$, then $\boldsymbol{x}^\star$ violates linear independence constraint qualification (LICQ). Furthermore, if $I_{00} \cup I_{0+} \neq \emptyset$ then $\boldsymbol{x}^\star$ violates Mangasarian-Fromovitz constraint qualification (MFCQ). In other words, LICQ only has a change to hold if no void material is present, whereas MFCQ might only hold if all the local stress constraints which belong to void material elements are feasible (inactive)—which is unlikely given nature of the stress-constrained problem. The authors go on to show that specialised forms of Adabie CQ, MFCQ and LICQ can be expected to hold in many practical situations. In particular, if $\boldsymbol{x}^\star$ is a local minimum of problem P_S, then the particular gradients

$$\frac{\partial}{\partial(\boldsymbol{u}, \boldsymbol{\rho})} \left([\boldsymbol{K}(\boldsymbol{\rho}^\star)]\boldsymbol{u}^\star\right) \tag{5.15}$$
$$\frac{\partial}{\partial \rho_i} \rho_i^\star \qquad i \in I_0 \tag{5.16}$$
$$\frac{\partial}{\partial \boldsymbol{u}} \left[\sigma_i(\boldsymbol{u})/\bar{\sigma} - 1\right] \qquad i \in I_{00} \cup I_{+0} \tag{5.17}$$

have to be linearly independent for a tailored version of Adabie CQ to hold[3]. The second vector (5.16) and third set of vectors (5.17) are linearly independent (trivially) due to the simple form of the quality function in the vanishing constraint (5.4), and the fact that no two local stress constraints (5.4) are dependent on the same subset of the displacement vector $\boldsymbol{u}$. Furthermore, it turns out that the gradients of the FE equation (5.15) may be linearly dependent without violating CQ [33, 119]. In general the conditions stated above are simply that 'the gradients of all the *active* constraints are linearly independent' [117]. (It seems as though 'vanished' equilibrium constraints may be viewed as 'inactive' constraints too [33].) Vanishing stress constraints (5.4) are deemed inactive if $\rho_i = 0$, even though the value of the constraint—*i.e.* zero exactly—would cause its selection in a standard active set strategy. Achtziger and Kanzow [117] note that this is a 'natural modification of the standard LICQ assumption'.

5.3.4 Computational complexity

Consider the NAND setting. In every iteration the FE system

$$[\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} = \boldsymbol{r} \tag{5.18}$$

is solved to obtain the displacement vector $\boldsymbol{u} \in \mathbb{R}^d$. Say we have at our disposal a solver which requires $\mathcal{O}(n^\phi)$ operations to solve a linear system with n unknowns. That is, $\mathcal{O}(d^\phi)$ operations are required in every iteration in the NAND setting in order to obtain the displacements $\boldsymbol{u} \in \mathbb{R}^d$ for a given topological design $\boldsymbol{\rho}$.

Assume that the sensitivity analysis is done with a standard adjoint variable method [2, 109]. Assume also that an active set strategy is employed. In the stress-constrained problem the number of active constraints will typically be a portion of the number of elements in the FE mesh: say χe with $0 < \chi \leq 1$. To obtain the sensitivity derivatives an additional $\chi e \mathcal{O}(d^\phi)$ operations are required—one additional load case per active stress constraint [2, 109]. The number of elements e are related to the number of degrees of freedom d by a constant factor ψ. Therefore, the total computational complexity per iteration in the NAND setting is at least

$$T_{\text{NAND}}(e) = \mathcal{O}((\psi e)^\phi) + \chi e \mathcal{O}((\psi e)^\phi) \,. \tag{5.19}$$

Neglecting all but the higher order terms, equation (5.19) reduces to

$$T_{\text{NAND}}(e) = \mathcal{O}(e^{\phi+1}) \,. \tag{5.20}$$

For example, say we have an exceptionally efficient linear system solver with practical scaling properties of $\mathcal{O}(e^1)$. The structural analysis phase of the NAND formulated local stress-constrained problem will then require at least $\mathcal{O}(e^2)$ operations, and although the optimization phase itself is typically quite cheap in the NAND setting, the overall computational complexity will be that of which ever procedure dominates. In practice the computational complexity of a NAND algorithm may be closer to $\mathcal{O}(e^3)$ [40, 42]. This is rather problematic with regard to the large-scale nature of the local stress-constrained topology design problem—especially in the limit of FE discretization.

[3] This is presented as Corollary 2 in Reference [117] and Definition 3.1 (MPVC-LICQ) in Reference [119].

In the SAND setting the 'computational kernel' is formed by the Karush-Kuhn-Tucker (KKT) primal-dual linear system itself, see for example Reference [33]. The KKT system of the local stress- and slope-constrained problem has at most $n+m = 2d+2e+2b$ unknowns. The number of inter element boundaries b is proportional to the number of elements e by ω, and like above, the number of degrees of freedom is $d = \psi e$. That is, a linear system with $n+m = 2\psi e+2e+2\omega e$ unknowns is constructed and solved in every iteration. Neglecting all but the higher order terms, the number of operations of a SAND algorithm could scale to

$$T_{\text{SAND}}(e) = \mathcal{O}(e^{\phi}) \,. \tag{5.21}$$

Hence, even though the SAND problem is larger and more complex than the equivalent NAND problem, the base complexity of the 'computational kernel' may be preserved in the overarching SAO algorithm. Compare this to the NAND setting wherein the sensitivity analyses necessarily cause an exponential increase in algorithmic complexity relative to problem size—*cf.* equation (5.20).

It is of course rather difficult to compare NAND and SAND solution procedures on an equal footing[4]. A rigorous and considered benchmark study—like that due to Wang and Arora [91] in the context of truss structures, or Rojas-Labanda and Stolpe [44] in the context of simply constrained problems—requires a substantial research effort and a fair amount of computational resources. (We hope to devote some of our attention to this in the future.) In the following section a numerical method is outlined which has its roots in conventional NAND-based structural optimization—based on strictly convex and separable approximate subproblems—see for example Reference [33]. A benchmark study should of course include an evaluation of the relative performances of readily available algorithms like IPOPT, SNOPT, KNITRO and SQP+.

5.4 The numerical method

Sequential approximate optimization (SAO) as solution strategy for general nonlinear programming involves the construction of inexpensive analytical approximation functions $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 0, 1, \ldots, m$ to the objective $f_0(\boldsymbol{x})$ and constraint functions $f_j(\boldsymbol{x})$, $j = 1, 2, \ldots, m$ at successive iteration points $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, 3, \ldots$. Together the approximations $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 0, 1, \ldots, m$ form an approximate subproblem $P[k]$. The form of the approximations $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$ are determined by the consequential solubility of the subproblem. In structural optimization it is customary to use *strictly convex* and *separable* approximations, as for example in CONLIN [34, 35] and the MMA [36]. The resulting subproblems are amenable to highly efficient dual statements when the number of constraints m is far less than the number of design variables n. When the number of constraints is relatively large, the performance of Lagrange-Newton diagonal

[4]For example, which restriction methods should be used in the respective settings—*e.g.* density filtering or slope constraints? How should the stress constraints be formulated—*e.g.* vanishing, ϵ-relaxed (maybe a modified reciprocal form) or qp-relaxed? Should the density variables in the SAND formulation be restricted with a nonzero lower bound too—like the NAND formulation—in order for the comparison to be valid? These considerations are only a few within the scope of the problem formulation, even more tricky is deciding on and implementing comparable numerical routines for each: the SAND optimization, the NAND optimization and the NAND structural analyses, and finally taking into account all the algorithmic parameters.

quadratic subproblems in combination with primal-dual interior-point subsolvers is considered superior [33, 45, 46].

The general diagonal quadratic approximation [65] is used to construct the objective and constraint function approximations $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 0, 1, \ldots, m$. The approximate constraint functions $\tilde{f}_j^{\{k\}}(\boldsymbol{x})$, $j = 1, 2, \ldots, m$ are then linearised, giving $P[k]$ the form of a diagonal quadratic program (QP). The constraint curvature information is retained as part of the diagonal quadratic objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$. The approximate objective function $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ is viewed as an approximation of the Lagrangian and the SAO method may be termed a diagonal Lagrange-Newton SQP-like algorithm [46]. The Lagrange-Newton subproblem may be expressed as

□ *The QP subproblem* $P[k]$:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \tilde{f}_0^{\{k\}}(\boldsymbol{x}) = f_0(\boldsymbol{x}^{\{k\}}) + \nabla f_0^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}})^T \boldsymbol{Q}^{\{k\}}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \\ \text{subject to} \quad & \tilde{f}_j^{\{k\}}(\boldsymbol{x}) = f_j(\boldsymbol{x}^{\{k\}}) + \nabla f_j^T(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) \; [=, \leq] \; 0 \; , \qquad j = 1, 2, \ldots, m \\ & \check{x}_i^{\{k\}} \leq x_i - x_i^{\{k\}} \leq \hat{x}_i^{\{k\}} \; , \qquad i = 1, 2, \ldots, n \end{aligned} \tag{5.22}$$

and

$$\check{x}_i^{\{k\}} \leftarrow \max(x_i^{\{k\}} - \delta(x_{i,\max} - x_{i,\min}), x_{i,\min}) \tag{5.23}$$

$$\hat{x}_i^{\{k\}} \leftarrow \min(x_i^{\{k\}} + \delta(x_{i,\max} - x_{i,\min}), x_{i,\max}) \tag{5.24}$$

for $i = 1, 2, \ldots, n$, with $x_{i,\min}$ and $x_{i,\max}$ the lower and upper bounds on variable i. For the sake of brevity we have resorted to some nonstandard notation—*i.e.* '$[=, \leq]$'—to denote both equality and inequality constraints.

The move limit is denoted by $0 < \delta < 1$ and manifests via the subproblem variable bounds (5.23) and (5.24). Convergence and termination is enforced with an adaptive move limit strategy, as per the convergent trust-region algorithm due to Fletcher *et al.* [66]. If subproblem $P[k]$ is infeasible, the point $\boldsymbol{x}^{\{k\}}$ is included in the filter list, the algorithm is backtracked to the previous feasible subproblem, and the move limit is expanded to the maximum value $\delta = 1$. Numerical experience indicates this is a viable 'restoration procedure' in SAND topology design; intuitively, subproblem $P[k]$ is typically infeasible if too much material is removed in $P[k-1]$ and/or if the move limit is too restrictive. Backtracking the algorithm means that the said material might be replaced, the inclusion of the infeasible subproblem (point) in the filter-list forces the algorithm to find a point with less severe constraint violations, and the expanded move limit may cause the availability of such a point in the first place—*e.g.* $\rho_i = 0$, or some large displacement variables in $\boldsymbol{u}$—which might have been outside the subproblem variable bounds for $\delta << 1$. In Section 5.5 the performance of this procedure is tested in a random multistart framework.

The Hessian matrix $\boldsymbol{Q}^{\{k\}}$ of the approximate objective $\tilde{f}_0^{\{k\}}(\boldsymbol{x})$ contains only diagonal elements—which have *minimal* storage requirements—constructed according to

$$Q_{ii}^{\{k\}} = c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}}, \quad i = 1, 2, \ldots, n \; . \tag{5.25}$$

The curvature terms $c_{j_i}^{\{k\}}$, $j = 1, 2, \ldots, m$, $i = 0, 1, \ldots, n$ are obtained as analytic, second-order partial derivatives of the objective $f_0(\boldsymbol{x})$ and constraint functions $f_j(\boldsymbol{x})$, $j = 1, 2, \ldots, m$ respectively

$$c_{j_i}^{\{k\}} = \frac{\partial^2 f_j(\boldsymbol{x}^{\{k\}})}{\partial x_i^2} \tag{5.26}$$

for all functions $j = 0, 1, \ldots, m$ and with respect to all variables $i = 1, 2, \ldots, n$. In the construction of (5.25) the Lagrangian multiplier estimates from the previous iterate are retained—*i.e.* $\boldsymbol{\lambda}^{\{k\}} = \boldsymbol{\lambda}^{\{k-1\}\star}$. Assuming feasibility and CQ, it is only required that $Q_{ii}^{\{k\}} > 0$, for $i = 1, 2, \ldots, n$, for the subproblem $P[k]$ to be strictly convex—if the Hessian is nondiagonal this is more involved. Herein the diagonal Hessian is constructed according to

$$Q_{ii}^{\{k\}} = \max\left(\alpha > 0, \left| c_{0_i}^{\{k\}} + \sum_{j=1}^{m} \lambda_j^{\{k\}} c_{j_i}^{\{k\}} \right|\right), \quad i = 1, 2, \ldots, n \tag{5.27}$$

with α a small positive number which ensures strict convexity—a standard value of 1×10^{-6} is used throughout. See Reference [33] for details. It should be noted here that the approximation strategy as per (5.27) with analytical curvature terms (5.26) is one of many possible approximation strategies in stress-constrained SAND topology optimization, and there thus seems to be plenty of scope for future work on second-order approximation strategies (which may include nondiagonal terms).

Subproblem $P[k]$ is first-order accurate with respect to problem P_S at every iteration point $\boldsymbol{x}^{\{k\}} = (\boldsymbol{u}^k, \boldsymbol{\rho}^k)$. Therefore, $(\boldsymbol{u}^\star, \boldsymbol{\rho}^\star)$ is a solution of problem P_S if and only if $(\boldsymbol{u}^\star, \boldsymbol{\rho}^\star) = (\boldsymbol{u}^k, \boldsymbol{\rho}^k)$ solves $P[k]$. Since problem $P[k]$ possesses only (affine) linear constraints, Abadie's CQ (or 'regularity') is trivially satisfied if $P[k]$ is feasible.

Fletcher and co-workers [66] prove a number of lemmas which guarantee the convergence of the sequence of iterates $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, 3, \ldots$. Under reasonable assumptions the algorithm will either terminate because a feasible subproblem can not be found (the 'restoration procedure' fails), or converge to a stationary (KKT) point, or accumulate at a feasible point which fails to satisfy MFCQ [78].

5.5 Numerical demonstrations and experiments

5.5.1 Two-bar truss topology design

This section is opened with the well-known two-bar truss topology design problem—Figure 5.1—which, 'despite its apparent simplicity', 'illustrates clearly the difficulties of topology design with stress constraints' [109]. First, slope constraints are not enforced, and two rather coarse mesh discretizations are considered—30×10 FE's and 60×20 FE's. Q8 FE's and a SIMP exponent $p = 3$ are used throughout. The numerical method outlined above is implemented with $\rho_{i,\text{min}} = 0$ and $\rho_{i,\text{max}} = 1$ for $i = 1, 2, \ldots, e$, and $u_{i,\text{min}} = -10^2$ and $u_{i,\text{max}} = -10^2$ for $i = 1, 2, \ldots, d$. The problems are initialized at $(\boldsymbol{u}^0, \boldsymbol{\rho}^0) = (\boldsymbol{0}, \boldsymbol{1})$ with an initial move limit of $\delta^0 = 0.2$ in each case. The von Mises stress criterion is used for $\sigma_i(\boldsymbol{u})$, evaluated at the centre of each element. The stress limit is set to $\bar{\sigma} = 20$. The ILOG CPLEX Barrier QP optimizer [128],

set to use up to 8 parallel threads, is employed as solver of each $P[k]$. The computational platform is the Rhasatsha HPC [77].

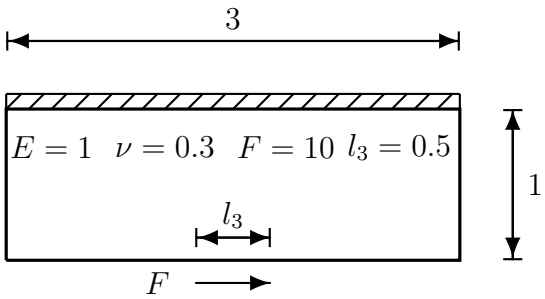
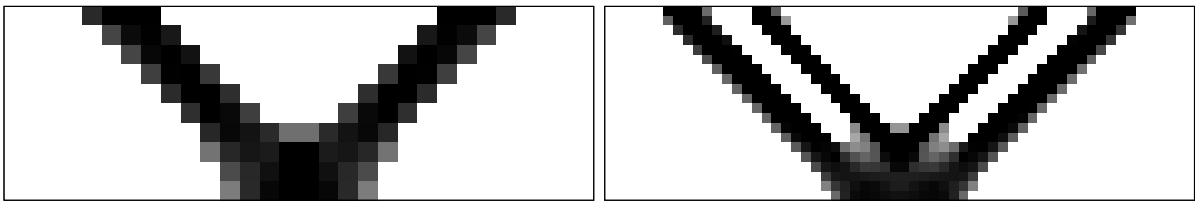


Figure 5.1: The two-bar truss problem.

Figure 5.2 shows the topologies converged to by the algorithm. The volume fraction objective is denoted by $f_0(\boldsymbol{\rho})$, the largest constraint violation is denoted by $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$, n_0 and n_1 are the number of elements at exactly 0 and 1 respectively; $\Phi(\boldsymbol{\rho})$ is the solid-void fraction. One can see that it is possible to remove the material membrane, symptomatic of the stress singularity problem [109], without resorting to perturbation or relaxation of the stress constraints. For the 60×20 FE mesh—Figure 5.2b—the same effect is observed, but the problem of mesh-dependency and local minima is encountered. This illustrates a disadvantage of the SAND formulated problem, which may be more multimodal than the equivalent NAND case.



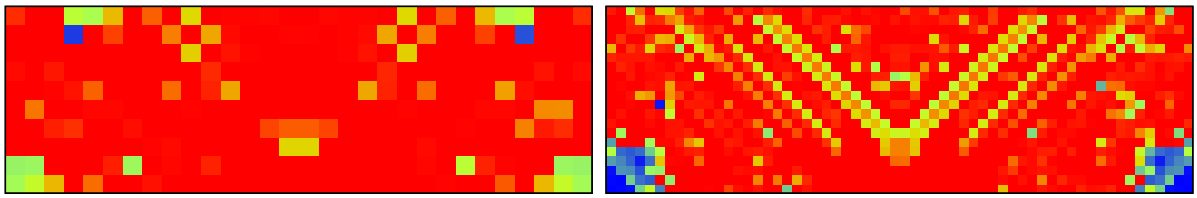
(a) $f_0(\boldsymbol{\rho})$ = .246, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 8.0E-07, $n_0(\boldsymbol{\rho})$ = 216, $n_1(\boldsymbol{\rho})$ = 18, $\Phi(\boldsymbol{\rho})$ = .78.

(b) $f_0(\boldsymbol{\rho})$ = .255, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 3.0E-04, $n_0(\boldsymbol{\rho})$ = 860, $n_1(\boldsymbol{\rho})$ = 186, $\Phi(\boldsymbol{\rho})$ = .87.

Figure 5.2: Two-bar truss: material distributions without stress relaxation (300 FE's and 1200 FE's).

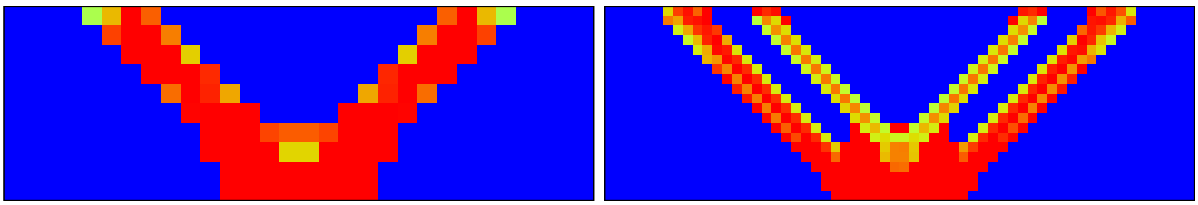
In Figure 5.3 the respective stress maps are displayed. For the sake of readability all the von Mises stresses equal to or greater than $\bar{\sigma} = 20$ are coloured red. Blue regions represent von Mises stresses of 0. The maximum stress $\max \sigma_i(\boldsymbol{u})$ is given in the caption of each figure. The number of active stress constraints, neglecting whether material is present or absent, is given by m^σ. Note that these values are collected with a post-processing routine—vanishing stress constrains (5.4) with values greater than -10^{-3} are deemed active, but active constraints are not identified or selected in any way external to the QP subsolver. In Figure 5.4 the same von Mises stress maps are given, but with the stresses in void elements $\rho_i = 0$ ignored. The maximum stress in all the nonzero elements $\rho_i > 0$ is denoted by $\max_{i \in I_+} \sigma_i(\boldsymbol{u})$. The number of nonvoid-active stress constraints is given by m^σ_+. Together Figures 5.3 and 5.4 illustrate the mechanism of vanishing stress constraints (5.4): if $\rho_i = 0$ the stress criterion $\sigma_i(\boldsymbol{u})$ can take on *any* value

without violating the constraint $\rho_i[\sigma_i/\bar{\sigma} - 1] \leq 0$; if $\rho_i > 0$ the stress constraint can only be satisfied if the stress criterion is feasible with respect to the stress limit $\sigma_i \leq \bar{\sigma}$.



(a) $\max \sigma_i(\boldsymbol{u})$ = 126.92, m^σ = 272. (b) $\max \sigma_i(\boldsymbol{u})$ = 202.70, m^σ = 987.

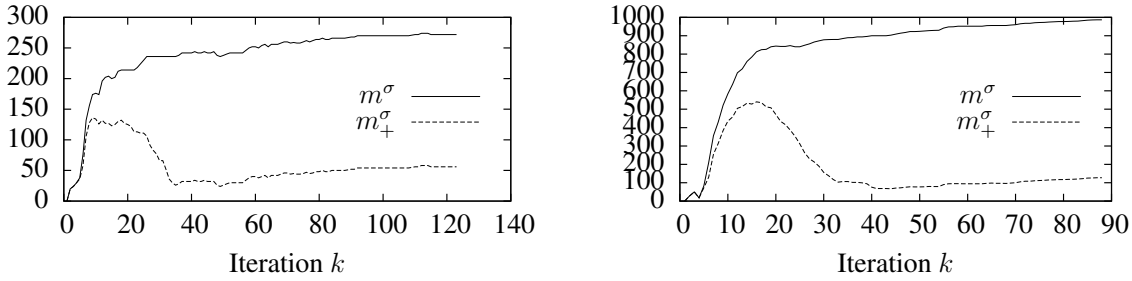
Figure 5.3: Two-bar truss: von Mises stress $\sigma_i(\boldsymbol{u})$ in each and every element $i = 1, 2, \ldots e$ (300 FE's and 1200 FE's).



(a) $\max_{i \in I_+} \sigma_i(\boldsymbol{u})$ = 20.00, m^σ_+ = 56. (b) $\max_{i \in I_+} \sigma_i(\boldsymbol{u})$ = 20.00, m^σ_+ = 127.

Figure 5.4: Two-bar truss: von Mises stress $\sigma_i(\boldsymbol{u})$ in nonzero elements $\rho_i > 0$, $i = 1, 2, \ldots, e$ (300 FE's and 1200 FE's).

Figure 5.5 illustrates the iteration histories of the number of active stress constraints m^σ and the number of nonvoid-active stress constraints m^σ_+. One can see that the iteration histories of nonvoid-active stress constraints (*i.e.* m^σ_+) correspond fairly well to the iteration histories of active (relaxed) stress constraints in the conventional NAND setting—*cf.* Figure 13 in Reference [109] and Figure 4 in Reference [43].



(a) 300 FE discretization (b) 1200 FE discretization

Figure 5.5: Number of active stress constraints.

Next the computational properties of the SAO algorithm outlined in Section 5.4 and the local stress- and slope-constrained SAND problem P_S is investigated. For this we remain with the two-bar truss problem, but now slope constrains [60] are imposed to mitigate the problem

Table 5.1: Large-scale storage and computational requirements

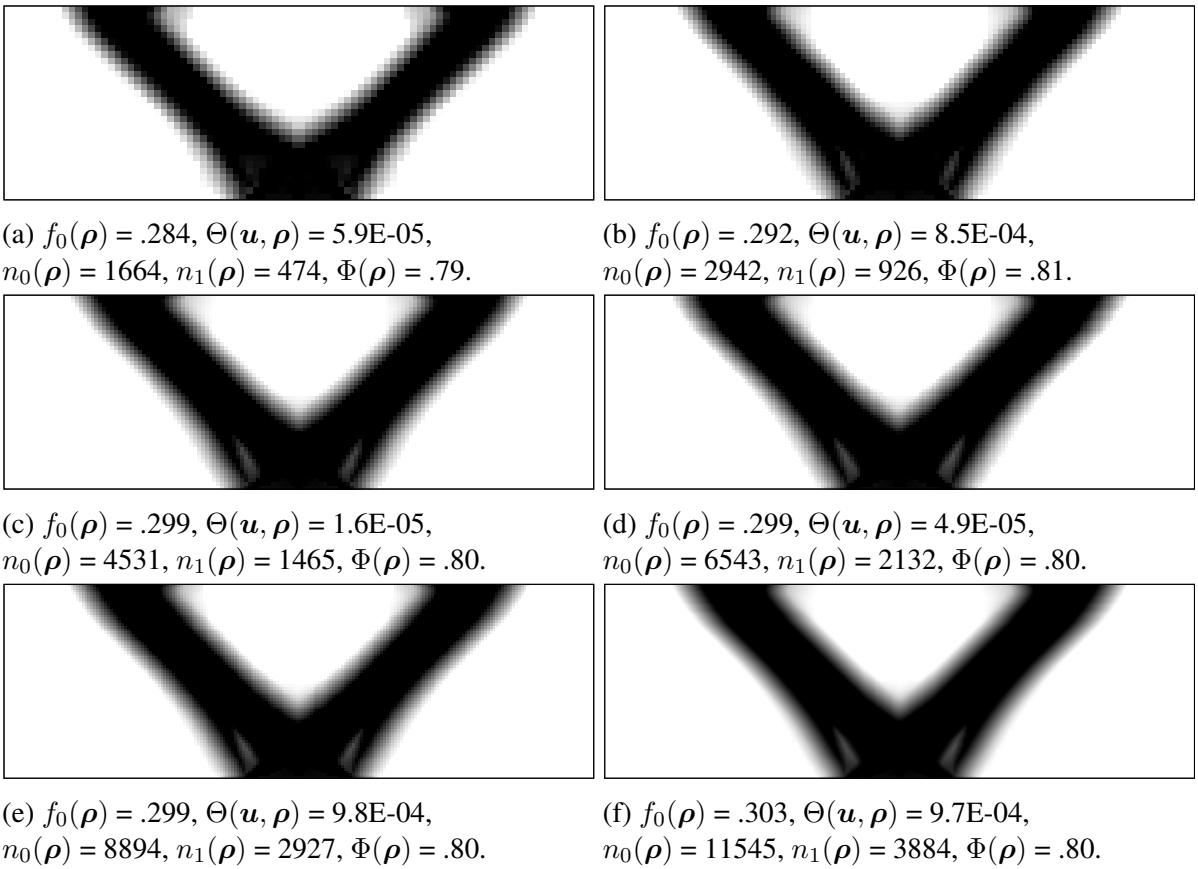
Figs.	$e_x \cdot e_y = e$	n	m	J_{nz}		m^σ	m^σ_+	k	T_k
	$60 \cdot 20 = 1200$	8722	13362	267444	(0.23%)	842	84	76	1.2
5.6a	$90 \cdot 30 = 2600$	19382	29942	600064	(0.10%)	1820	156	99	3.4
5.6b	$120 \cdot 40 = 4800$	34242	53122	1065284	(0.06%)	3206	264	140	9.2
5.6c	$150 \cdot 50 = 7500$	53302	82902	1663104	(0.04%)	4910	379	119	21.3
5.6d	$180 \cdot 60 = 10800$	76562	119282	2393524	(0.03%)	7055	512	146	28.8
5.6e	$210 \cdot 70 = 14700$	104022	162262	3256544	(0.02%)	9497	603	174	41.3
5.6f	$240 \cdot 80 = 19200$	135682	211842	4252164	(0.02%)	12322	777	199	50.9

of mesh-dependency and local minima. The mesh parameter is set to $\pi = 1/e_x$, with e_x the number of elements in the horizontal direction of the domain. The slope parameter is set at $\mu = 20$.

The mesh disretization range is summarised in Table 5.1, the largest problem size has 19200 FE's: n is the total number of optimization variables $(e + d)$, m the total number of constraints $(d+e+2b)$ and J_{nz} is the number of nonzero terms in the constraint Jacobian matrix (the number in brackets is the percentage filled). The algorithm is terminated if the Euclidean norm of the step in all the optimization variables $(\boldsymbol{u}, \boldsymbol{\rho})$ is less than 1: k is the number of iterations to convergence and T_k is the average time per iteration (in seconds). The topologies generated can be seen in Figure 5.6. In large-scale problem instances numerical difficulties may occur—reported by the QP solver—which we have found can be mitigated by setting insubstantial material density variables to zero exactly just before the construction of each subproblem $P[k]$—here a threshold of $\rho_i < 1 \times 10^{-3}$ is used, which, in terms of SIMP with $p = 3$, equates to an insubstantial equivalent density/stiffness.

The storage requirements of problem P_S scale in the linear with respect to the number of elements e. More surprising however, based on the empirical evidence in Table 5.1, the average times per iteration T_k scale favourably too—practically linear. A number of runs during various load levels on the computational platform confirmed the results. Due to the increased number of iterations required for convergence for larger problem sizes, the total time to convergence scales slightly worse than average time per iteration T_k. The numerics of the problems is not scaled in any way in the external algorithm—the subsolver of course has its own preconditioning and scaling routines.

A potential pitfall of the slope-constrained problem is the necessity of 'grey' transition zones between solid and void regions—resulting in fairly poor solid-void fractions $\Phi(\boldsymbol{\rho})$. To remedy this, slope constraints may relaxed with a continuation strategy, ending up with an unrestricted problem P_S with $\pi\mu \geq 1$. The slope parameter μ is adjusted with 1.1μ if the Euclidean norm of the step in all the optimization variables $(\boldsymbol{u}, \boldsymbol{\rho})$ is less than 1. Also, the trust-region filter-list is cleared. (Effectively then, a new optimization problem P_S is embarked upon each time the convergence condition is satisfied, initialized at a fairly good starting position.) Once $\pi\mu \geq 1$, the algorithm is allowed to terminate. The results of this procedure are displayed in Figure 5.7 and the accompanying nonvoid-active stress maps are given in Figure 5.8. It is clear that fairly well-defined topologies can be obtained in this way.



(a) $f_0(\boldsymbol{\rho})$ = .284, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 5.9E-05, $n_0(\boldsymbol{\rho})$ = 1664, $n_1(\boldsymbol{\rho})$ = 474, $\Phi(\boldsymbol{\rho})$ = .79.

(b) $f_0(\boldsymbol{\rho})$ = .292, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 8.5E-04, $n_0(\boldsymbol{\rho})$ = 2942, $n_1(\boldsymbol{\rho})$ = 926, $\Phi(\boldsymbol{\rho})$ = .81.

(c) $f_0(\boldsymbol{\rho})$ = .299, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.6E-05, $n_0(\boldsymbol{\rho})$ = 4531, $n_1(\boldsymbol{\rho})$ = 1465, $\Phi(\boldsymbol{\rho})$ = .80.

(d) $f_0(\boldsymbol{\rho})$ = .299, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.9E-05, $n_0(\boldsymbol{\rho})$ = 6543, $n_1(\boldsymbol{\rho})$ = 2132, $\Phi(\boldsymbol{\rho})$ = .80.

(e) $f_0(\boldsymbol{\rho})$ = .299, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 9.8E-04, $n_0(\boldsymbol{\rho})$ = 8894, $n_1(\boldsymbol{\rho})$ = 2927, $\Phi(\boldsymbol{\rho})$ = .80.

(f) $f_0(\boldsymbol{\rho})$ = .303, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 9.7E-04, $n_0(\boldsymbol{\rho})$ = 11545, $n_1(\boldsymbol{\rho})$ = 3884, $\Phi(\boldsymbol{\rho})$ = .80.

Figure 5.6: Large-scale two-bar truss design without stress relaxation, $\mu = 20$.

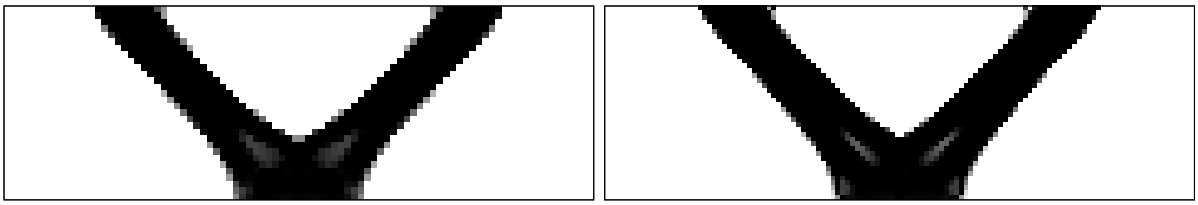
5.5.2 The MBB beam problem

The second experimental problem is the MBB beam—Figure 5.9—a problem well-known for its severe multimodality. In an attempt to cope with the problem of local minima, a random multistart procedure is implemented. The idea is that the algorithm may be initialized at any number of random starting positions, and based on the fitness of the solutions obtained, and the frequency with which the fittest known solution is converged to, one may hazard a guess—in terms of a confidence measure—pertaining to the globally optimality of the fittest known solution [61]. In Reference [59] we implement a similar multistart procedure (and the same algorithm) to generate 'probably globally optimal, approximately 0-1' solutions to the classical minimum compliance topology design problem.

A 180×60 mesh discretization is used—half the beam is modelled, due to symmetry. The bounds on the displacement variables $\boldsymbol{u}$ are expanded to $\pm 10^3$. All the other algorithmic settings are retained from before.

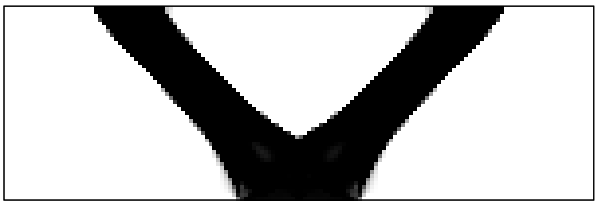
One hundred instances of problem P_S with $\mu = 18$ are initialized, each with a random material distribution such that $0.5 \leq \rho_i^0 \leq 1$, $i = 1, 2, \ldots, e$, with every element in $\boldsymbol{\rho}^0$ random[5]. The same is done for random initial material distributions restricted to $0.75 \leq \rho_i^0 \leq 1$,

[5]The randomly initialized problems are completely independent, and may therefore be computed in a reasonable amount of time on a parallel computation platform. Our algorithm, for example, can run 10 batches of 10

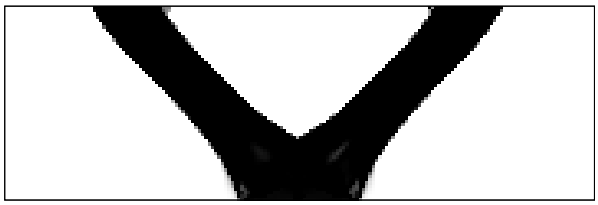


(a) $f_0(\boldsymbol{\rho})$ = .236, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.3E-04, $n_0(\boldsymbol{\rho})$ = 2030, $n_1(\boldsymbol{\rho})$ = 522, $\Phi(\boldsymbol{\rho})$ = .95.

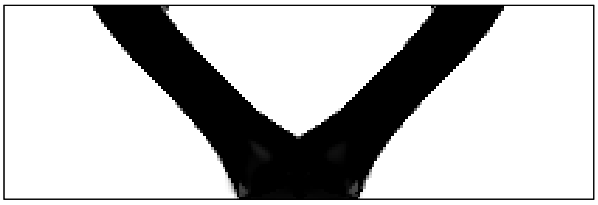
(b) $f_0(\boldsymbol{\rho})$ = .246, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.4E-03, $n_0(\boldsymbol{\rho})$ = 3572, $n_1(\boldsymbol{\rho})$ = 1048, $\Phi(\boldsymbol{\rho})$ = .96.



(c) $f_0(\boldsymbol{\rho})$ = .252, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.2E-03, $n_0(\boldsymbol{\rho})$ = 5531, $n_1(\boldsymbol{\rho})$ = 1718, $\Phi(\boldsymbol{\rho})$ = .97.



(d) $f_0(\boldsymbol{\rho})$ = .252, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 3.7E-03, $n_0(\boldsymbol{\rho})$ = 7995, $n_1(\boldsymbol{\rho})$ = 2466, $\Phi(\boldsymbol{\rho})$ = .97.



(e) $f_0(\boldsymbol{\rho})$ = .252, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 3.0E-03, $n_0(\boldsymbol{\rho})$ = 10919, $n_1(\boldsymbol{\rho})$ = 3333, $\Phi(\boldsymbol{\rho})$ = .97.

(f) $f_0(\boldsymbol{\rho})$ = .255, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.3E-03, $n_0(\boldsymbol{\rho})$ = 14213, $n_1(\boldsymbol{\rho})$ = 4509, $\Phi(\boldsymbol{\rho})$ = .98.

Figure 5.7: Large-scale two-bar truss design without stress relaxation, μ continuation.

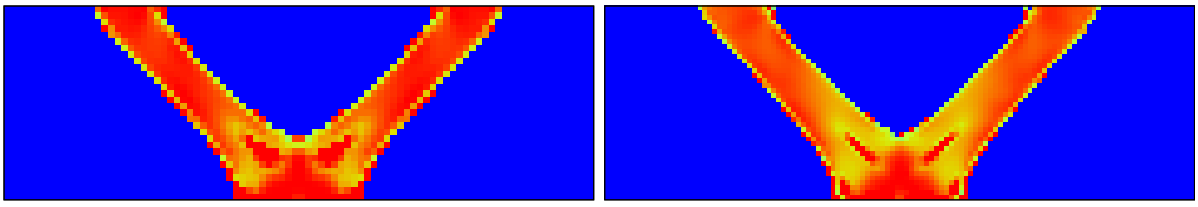
$i = 1, 2, \ldots, e$. Both cases are repeated with a less restrictive slope parameter $\mu = 21$. The multistart statistics are summarised in Table 5.2, accompanied by the fittest solution found, in each case, in Figure 5.10. The number of infeasible terminations are denoted by ζ, P is the total number of subproblems $P[k]$ the computational platform was tasked with, and T_P is the average time required per subproblem, in seconds.

The average time per subproblem T_P is about double that of the equivalently sized two-bar truss problem. This is probably due the increased difficulty of the subproblems, in part inherent to the MBB beam problem, but also because feasibility is more difficult to maintain

problems in parallel, which, in this case, requires about 50 hours of wall-time.

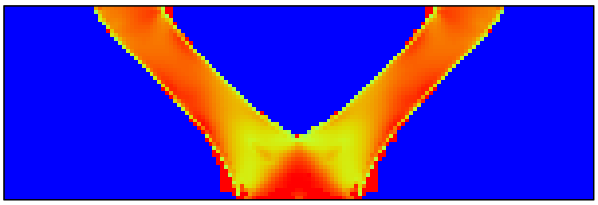
Table 5.2: Multistart statistics; 100 runs each.

Fig.	μ		ζ	P	T_P
5.10a	18	$0.75 \leq \rho_i^0 \leq 1$	4	15357	75
5.10b	21	$0.75 \leq \rho_i^0 \leq 1$	5	16304	82
5.10c	18	$0.5 \leq \rho_i^0 \leq 1$	28	17014	78
5.10d	21	$0.5 \leq \rho_i^0 \leq 1$	27	17116	90

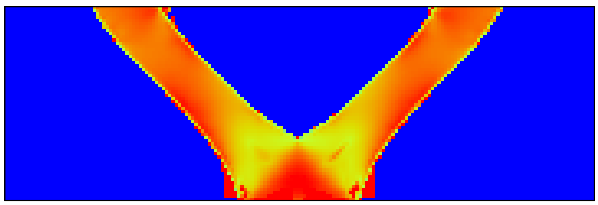


(a) $\max \sigma_i(\boldsymbol{u}) = 363.02$, $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 20.00$, $m^\sigma = 2132$. $m^\sigma_+ = 102$.

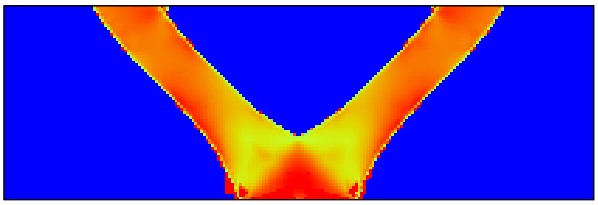
(b) $\max \sigma_i(\boldsymbol{u}) = 373.43$, $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 20.00$, $m^\sigma = 3684$, $m^\sigma_+ = 112$.



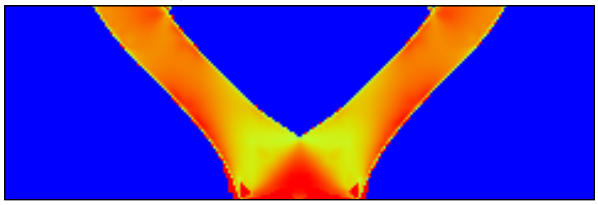
(c) $\max \sigma_i(\boldsymbol{u}) = 474.94$, $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 20.06$, $m^\sigma = 5681$, $m^\sigma_+ = 150$.



(d) $\max \sigma_i(\boldsymbol{u}) = 662.55$, $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 20.08$, $m^\sigma = 8166$, $m^\sigma_+ = 171$.



(e) $\max \sigma_i(\boldsymbol{u}) = 828.14$, $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 20.01$, $m^\sigma = 11117$, $m^\sigma_+ = 198$.



(f) $\max \sigma_i(\boldsymbol{u}) = 1518.08$, $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 20.09$, $m^\sigma = 14461$, $m^\sigma_+ = 248$.

Figure 5.8: Large-scale two-bar truss design: von Mises stress $\sigma_i(\boldsymbol{u})$ in nonzero elements $\rho_i > 0$, $i = 1, 2, \ldots, e$, μ continuation

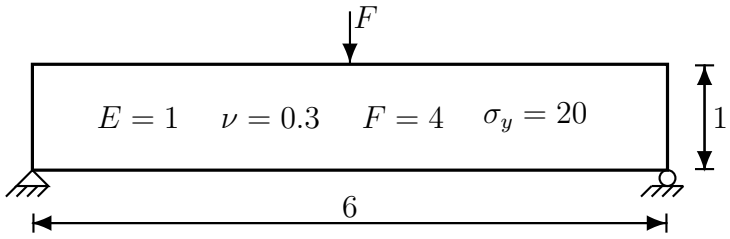
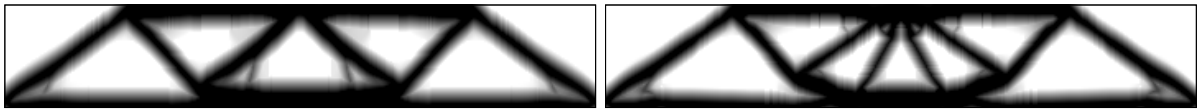


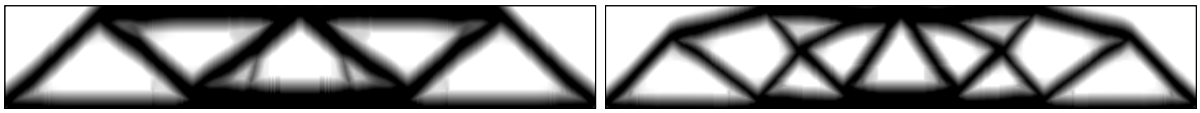
Figure 5.9: The MBB beam problem (stress-constrained minimum weight; load and roller support distributed over $1/5$th of the domain).

from the random starting positions. The number of infeasible terminations ζ are almost 1 in 3 for random initializations between $0.5 \leq \rho_i \leq 1$, $i = 1, 2, \ldots, e$. (For $0 \leq \rho_i \leq 1$, not shown here, its about 1 in 2.) In the cases with the less restrictive slope parameter $\mu = 21$, the batch with the more expansive random initializations (*i.e.* $0.5 \leq \rho^0_i \leq 1$, $i = 1, 2, \ldots, e$) arrived at a fitter solution—Figure 5.10b *vs.* Figure 5.10d—with a lower objective function value $f_0(\boldsymbol{\rho})$ and an improved solid-void fraction $\Phi(\boldsymbol{\rho})$. So, there seems to be a trade-off: an expanded scope of random initializations may have a better chance of arriving at the global minimum,



(a) $f_0(\boldsymbol{\rho})$ = .461, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 3.0E-03, $n_0(\boldsymbol{\rho})$ = 3440, $n_1(\boldsymbol{\rho})$ = 2349, $\Phi(\boldsymbol{\rho})$ = .54.

(b) $f_0(\boldsymbol{\rho})$ = .447, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.3E-03, $n_0(\boldsymbol{\rho})$ = 3693, $n_1(\boldsymbol{\rho})$ = 2194, $\Phi(\boldsymbol{\rho})$ = .55.



(c) $f_0(\boldsymbol{\rho})$ = .486, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2.0E-03, $n_0(\boldsymbol{\rho})$ = 3234, $n_1(\boldsymbol{\rho})$ = 2641, $\Phi(\boldsymbol{\rho})$ = .54.

(d) $f_0(\boldsymbol{\rho})$ = .459, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.7E-03, $n_0(\boldsymbol{\rho})$ = 3096, $n_1(\boldsymbol{\rho})$ = 1929, $\Phi(\boldsymbol{\rho})$ = .47.

Figure 5.10: MBB: material distributions without stress relaxation, $\mu = 18$ and $\mu = 21$.

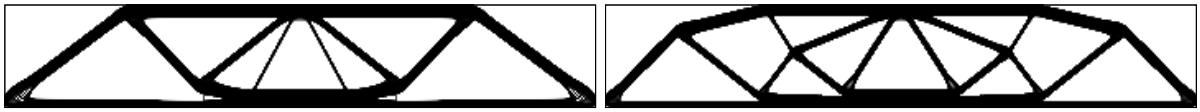
but the amount of computation time wasted due to infeasible terminations is increased. That is a major disadvantage of the SAND setting, an infeasible termination contains little-no usable information since the equilibrium condition is not satisfied.

We hope to, in the future, implement more sophisticated restoration procedures and investigate more thoroughly the properties of the multistart strategy. For now, these results demonstrate that a rudimentary trust-region algorithm can cope (to some extent) with the local stress- and slope-constrained problem P_S, without resorting to relaxation or perturbation techniques. The multistart strategy may therefore be an useful framework in which to test various restoration procedures, constraint formulations, approximation strategies *et cetera*. That is, one may be able to evaluate a modification or a perturbation of an algorithmic property or parameter, or a modification to the problem formulation itself, by comparing the number of infeasible terminations, the frequency with which the best known solution is found (something we do not attempt here), and the total computational effort required to run a batch of problems. This may also form part of a rigorous NAND-SAND benchmark study wherein the same multistart strategy is applied to the NAND problems too, which should reveal the relative multimodality of the two formulations—it is expected that the NAND case will be less severe.

The slope parameter continuation strategy, as used above in consideration of the two-bar truss problems, is repeated here. The topologies depicted in Figures 5.10b and 5.10d are used as starting positions. The results are given in Figures 5.11 and 5.12. Interestingly, the solution with the lower objective function value $f_0(\boldsymbol{\rho})$ has far fewer nonvoid-active stress constraints m_+^σ, but a much larger number or 'standard' active stress constraints m^σ. Moreover, we can report that the average von Mises stress $\sigma_i(\boldsymbol{u})$ in nonvoid elements $\rho_i > 0$ is higher in the topology with the lower objective value: 12.90 *vs.* 12.42. This seems to indicate that the topology uses less material to distribute stress more evenly—material is thus distributed in a superior way.

5.5.3 L-shape beam design

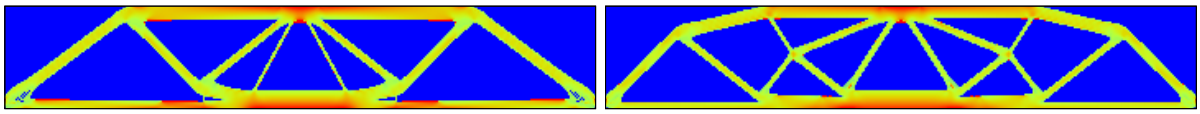
The L-shaped beam (or 'L-bracket') is a standard benchmark in topology optimization. The design domain is depicted in Figure 5.13. The issue is, as noted by Duysinx and Bendsøe [109], 'not so much the optimization part but more the numerical problem of capturing the theoretically infinite stress at the inner corner', and the solution is 'strongly dependent on the quality of the



(a) $f_0(\boldsymbol{\rho})$ = .324, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4.4E-05, $n_0(\boldsymbol{\rho})$ = 7215, $n_1(\boldsymbol{\rho})$ = 3310, $\Phi(\boldsymbol{\rho})$ = .97.

(b) $f_0(\boldsymbol{\rho})$ = .329, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1.5E-03, $n_0(\boldsymbol{\rho})$ = 7191, $n_1(\boldsymbol{\rho})$ = 3312, $\Phi(\boldsymbol{\rho})$ = .97.

Figure 5.11: MBB: material distributions without stress relaxation, μ continuation.



(a) $\max \sigma_i(\boldsymbol{u})$ = 675.95, $\max_{i \in I_+} \sigma_i(\boldsymbol{u})$ = 20.00, m^σ = 7291, m^σ_+ = 119.

(b) $\max \sigma_i(\boldsymbol{u})$ = 1841.23, $\max_{i \in I_+} \sigma_i(\boldsymbol{u})$ = 20.15, m^σ = 3742, m^σ_+ = 646.

Figure 5.12: MBB: von Mises stress $\sigma_i(\boldsymbol{u})$ in nonzero elements $\rho_i > 0$, $i = 1, 2, \ldots, e$, μ continuation.

analysis'. This then seems to be an especially interesting problem in the SAND setting, for, in SAND, the 'analysis' part is united with the 'optimization' part, and the equilibrium condition may only be satisfied at convergence.

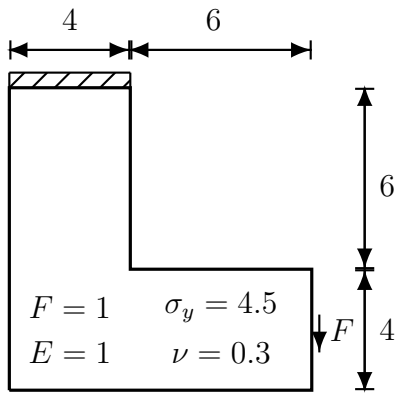
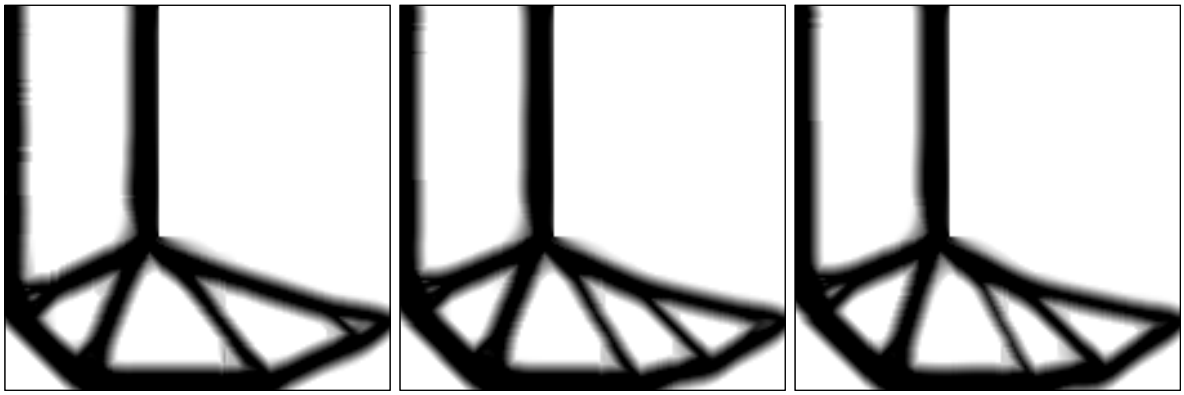


Figure 5.13: The L-shape beam problem (load distributed over $1/25$th of the domain).

The design domain is discretized with 14400 FE's[6]. All algorithmic settings are retained from before. The slope parameter μ is set to 27. One hundred problem instances are initialized with random material density distributions $0.75 \leq \rho_i^0 \leq 1$, $i = 1, 2, \ldots, e$. Six infeasible terminations are encountered. The three fittest solutions that were found are depicted in Figure 5.14. The results correspond well with the topological designs reported in Reference [43].

As mentioned before, the prevalence of 'grey' material is problematic. The result of slope constraint continuation is depicted in Figure 5.15. In Figure 5.16 the entire stress map—void and nonvoid elements—is depicted for each design; accompanied by the stress maps of nonzero

[6]The encompassing square domain is actually discretized with 22500 FE's, but those FE's in the upper-right 'void' region are restricted to 0 on both the lower bound $\rho_{i,\min} = 0$ and the upper bound $\rho_{i,\max} = 0$. Displacement boundary conditions are enforced in the same way.



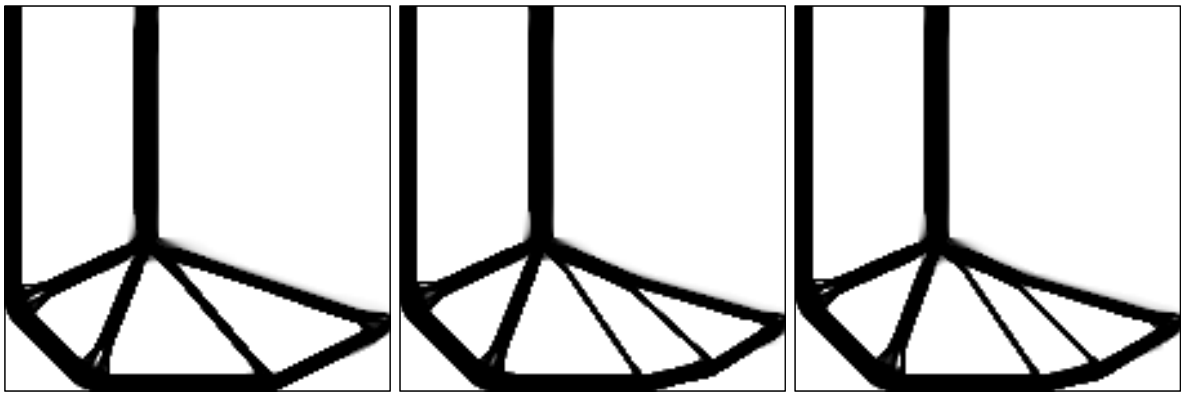
(a) $f_0(\boldsymbol{\rho})$ = .385, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 1E-3, $n_0(\boldsymbol{\rho})$ = 6818, $n_1(\boldsymbol{\rho})$ = 3320, $\Phi(\boldsymbol{\rho})$ = .70.

(b) $f_0(\boldsymbol{\rho})$ = .390, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 7E-4, $n_0(\boldsymbol{\rho})$ = 6670, $n_1(\boldsymbol{\rho})$ = 3262, $\Phi(\boldsymbol{\rho})$ = .69.

(c) $f_0(\boldsymbol{\rho})$ = .390, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 4E-5, $n_0(\boldsymbol{\rho})$ = 6684, $n_1(\boldsymbol{\rho})$ = 3265, $\Phi(\boldsymbol{\rho})$ = .69.

Figure 5.14: L-shape beam: material distributions without stress relaxation, $\mu = 27$.

elements $\rho_i > 0, i = 1, 2, \ldots, e$ in Figure 5.17. One can see that the highly stressed inner corner is problematic in terms of the complete removal of some low density, fully stressed elements in the vicinity of the inner corner, but also along the main diagonal member. This is somewhat reminiscent of the original stress singularity problem. Nevertheless, all three solutions are feasible (and may be improved with a tighter convergence tolerance), and although the von Mises stresses in void elements $\rho_i = 0$ far exceed the stress limit of $\bar{\sigma} = 4.5$—see Figure 5.16—-all the stresses in nonvoid elements $\rho_i > 0$ are strictly limited to the critical value—*cf.* Figure 5.17.

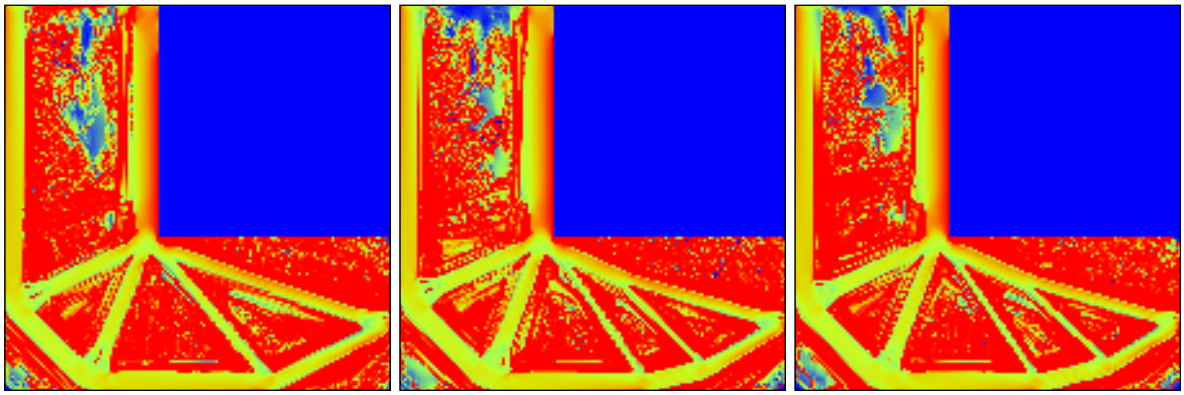


(a) $f_0(\boldsymbol{\rho})$ = .303, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 2E-4, $n_0(\boldsymbol{\rho})$ = 9663, $n_1(\boldsymbol{\rho})$ = 4045, $\Phi(\boldsymbol{\rho})$ = .95.

(b) $f_0(\boldsymbol{\rho})$ = .304, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 3E-4, $n_0(\boldsymbol{\rho})$ = 9698, $n_1(\boldsymbol{\rho})$ = 4033, $\Phi(\boldsymbol{\rho})$ = .95.

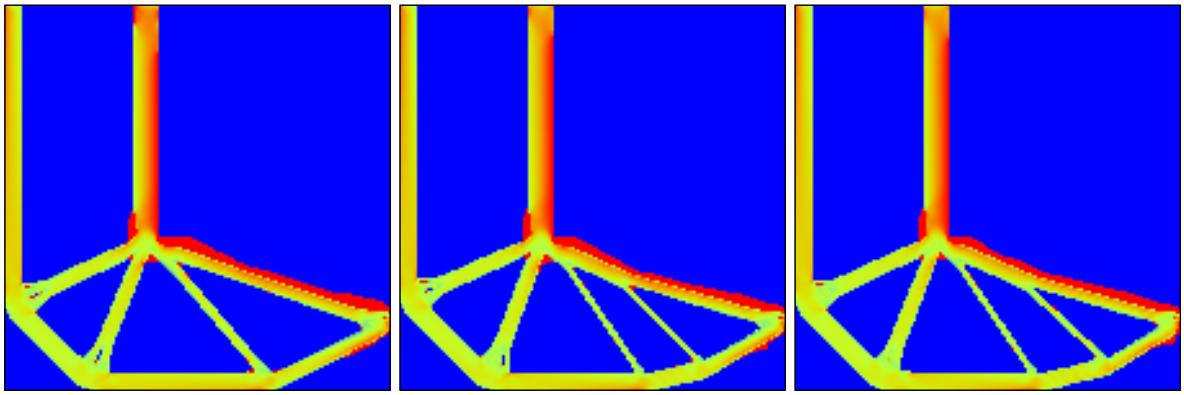
(c) $f_0(\boldsymbol{\rho})$ = .303, $\Theta(\boldsymbol{u}, \boldsymbol{\rho})$ = 8E-4, $n_0(\boldsymbol{\rho})$ = 9705, $n_1(\boldsymbol{\rho})$ = 4022, $\Phi(\boldsymbol{\rho})$ = .95.

Figure 5.15: L-shape beam: material distributions without stress relaxation, μ continuation.



(a) $\max \sigma_i(\boldsymbol{u}) = 2774.51$, $m^\sigma = 10164$. (b) $\max \sigma_i(\boldsymbol{u}) = 2714.27$, $m^\sigma = 10149$. (c) $\max \sigma_i(\boldsymbol{u}) = 2713.34$, $m^\sigma = 10130$.

Figure 5.16: L-shape beam: von Mises stress $\sigma_i(\boldsymbol{u})$ in each and every element $i = 1, 2, \ldots e$, μ continuation.



(a) $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 4.50$, $m^\sigma_+ = 501$. (b) $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 4.50$, $m^\sigma_+ = 444$. (c) $\max_{i \in I_+} \sigma_i(\boldsymbol{u}) = 4.50$, $m^\sigma_+ = 432$.

Figure 5.17: L-shape beam: von Mises stress $\sigma_i(\boldsymbol{u})$ in nonzero elements $\rho_i > 0$, $i = 1, 2, \ldots, e$, μ continuation.

5.6 Concluding remarks

We have demonstrated that solution methods and problem formulations based on an alternative SAND formulation of the topology design problem may be worth further investigation. Herein a local stress- and slope-constrained version of the SAND topology design problem is solved with a standard SAO solution procedure taken from a conventional structural optimization methodology. Beyond known disadvantages of the SAND setting—*e.g.* very large optimization problems, CQ, feasibility and convergence issues—it is demonstrated that singular optima may be converged to without having to resort to relaxation or perturbation techniques. Moreover, due to the sparsity of the problem the required computational effort is fairly modest. A rudimentary trust-region algorithm can cope with feasibility issues. And finally, the *ad-hoc* structural analysis which forms part of the SAND optimization problem seems to be able to capture the structural responses which characterise standard benchmark problems.

Wang and Arora [91] make the pertinent point that ‘SAND represents a fundamental shift in the way analysis and design problems are currently treated’. There is thus fantastic scope for future work on tailored solution methods in the SAND setting, alternative SAND problem formulations, the incorporation of more sophisticated finite element formulations, mesh refinement, and rigorous benchmarking in order to enable a fair comparison with conventional NAND solution procedures.

Chapter 6

SEASAND: A direct approach to structural topology design

This chapter is a reproduction of a paper entitled 'SEASAND: a direct approach to structural topology design' [129]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical and Mechatronic Engineering at the University of Stellenbosch, South Africa.

6.1 Abstract

We delineate the union of conventional sequential approximate optimization (SAO) with the 'direct' or 'simultaneous analysis and design' (SAND) formulation of the local stress-constrained topology design problem. The finite element equilibrium equations are retained as a set of nonlinear equality constraints and the state variables—*i.e.*, displacements—form part of the overall set of primal optimization variables. Therefore, the typically complex and expensive sensitivity analyses associated with state-based constraints, simplify to the calculation of simple partial derivatives. Due to the presence of nonlinear equality constraints, the SAO procedure (exclusively geared, traditionally, for inequality constrained problems) is extended to a general, nonlinear and nonconvex, mathematical programming framework. It is demonstrated that 'sequential approximate simultaneous analysis and design' (SEASAND) may offer very large-scale structural optimization capabilities.

6.2 Introduction

From the outset, we consider an important and difficult problem in structural optimization; the 'simultaneous analysis and design' (SAND, or 'direct')[1] formulation of the minimum weight,

[1] See for example the contributions by Arora and Wang [49, 91] and Rojas-Labanda and Stolpe [44, 130].

SIMP[2], local stress-constrained[3] and slope-constrained[4] topology optimization problem[5] $\mathcal{P}_S$:

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{\rho}} \quad & \sum_{i=1}^{e} \frac{\rho_i}{e} \\ \text{subject to} \quad & [\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} - \boldsymbol{r} = \boldsymbol{0} \\ & \boldsymbol{\rho} \diamond \left[\frac{\boldsymbol{\sigma}(\boldsymbol{u})}{\bar{\sigma}} - \boldsymbol{1} \right] \leq 0 \\ & \boldsymbol{0} \leq \boldsymbol{\rho} \leq \boldsymbol{1} \, . \\ & \rho_{i(k)} - \rho_{j(k)} \leq \mu\pi \, , \qquad k = 1, 2, \ldots, b \\ & \rho_{j(k)} - \rho_{i(k)} \leq \mu\pi \, , \qquad k = 1, 2, \ldots, b \end{aligned} \tag{6.1}$$

The design domain is discretized with e finite elements (FE's), with b inter element boundaries. Each FE is assigned a material density variable ρ_i and a stress measure $\sigma_i(\boldsymbol{u})$, with $i = 1, 2, \ldots, e$. The $\rho_{i(k)}$ and $\rho_{j(k)}$ represent the elements which share inter element boundary number k, for $k = 1, 2, \ldots, b$. The vector of material density variables is denoted by $\boldsymbol{\rho} \in \mathcal{R}^e$, and the stress vector by $\boldsymbol{\sigma}(\boldsymbol{u}) \in \mathcal{R}^e$. The 'diamond' symbol ($\diamond$) indicates element-wise multiplication: $\boldsymbol{\rho} \diamond \boldsymbol{\sigma}(\boldsymbol{u}) \in \mathcal{R}^e$. The vector of nodal displacements is $\boldsymbol{u} \in \mathcal{R}^d$, with d the number of degrees of freedom. The globally assembled SIMP modified stiffness matrix $[\boldsymbol{K}(\boldsymbol{\rho})] = \sum_{i=1}^{e} (\rho_i)^p [\boldsymbol{K}_i]$ has dimension $d \times d$, with p the SIMP parameter. The vector of primal optimization variables is denoted by

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\rho} \end{bmatrix} \in \mathcal{R}^d \, . \tag{6.2}$$

Problem $\mathcal{P}_S$ has a total of $m = d + e + 2b$ constraints, of which d are nonlinear equality constraints. The SIMP modified FE discretized equations of equilibrium

$$[\boldsymbol{K}(\boldsymbol{\rho})]\boldsymbol{u} = \boldsymbol{r} \in \mathcal{R}^d \tag{6.3}$$

form the latter—that is, the 'structural analysis' component in conventional formulations and methods. The vector of nodal forces is denoted by $\boldsymbol{r} \in \mathcal{R}^d$.

On the one had, by specifying the slope parameters $\mu\pi$, slope constraints control the minimum length scale of the design[6]. The minimum length scale is a macro-scale property, relevant to reductive manufacturability considerations—*e.g.* casting, moulding and machining. Local stress constraints, on the other hand—which limit elemental stresses $\sigma_i(\boldsymbol{u})$ to a predefined

[2]The 'solid isotropic material with penalization' technique, independently proposed by Bendsøe [54] and Rozvany and Zhou [55].

[3]First studied in the continuum (or 'bi-dimensional') setting by Duysinx and Bendsøe [109]. See Reference [88] and the references therein for details specific to the SAND variant of the problem with 'vanishing' stress constraints.

[4]A design-set restriction technique proposed by Petersson [60]. See Reference [59] for details particular to the SAND setting.

[5]Singular local minima are feasible and available—material density variables are limited on the lower bound at *zero* (**0**) exactly—and may be converged to using standard gradient-based optimization methods, without having to resort to relaxation or perturbation techniques [88]—as first demonstrated by Stolpe and Svanberg [116].

[6]In practice, this should probably be accompanied with a global optimization procedure [59].

bound $\bar{\sigma}$—model and control localized material failure—or 'strength failure' [131]—which is, in general, a micro-scale property [109]. Considering strength failure early in the design phase is an important topic in its own right [104], but particularly so in view of the relationship between topology optimization and additive manufacturing processes. From a mathematical programming point of view, figuring out an efficient and robust algorithm for problems with basic local stress constraints is the first step in modelling and controlling for the residual stresses and strains associated with metal additive manufacturing processes—in the light of a general notion of *state-based* constraints, it might be possible to easily incorporate the associated thermal phenomena (see for example Megahed *et al.* [132]) as well.

More than two decades ago, Haftka, Gürdal and Kamat suggested that the 'high computational cost associated with the analysis of many complex real-life problems', 'motivates a focus on optimization techniques that call for minimal interference with the structural analysis package', and, 'require only a small number of structural analysis runs' [3]. The issue is that, in conventional 'nested analysis and design' (NAND)—the 'reduced' formulation—each active state-based constraint implicates an additional structural analysis run in order to calculate the associated first-order sensitivity derivatives [88]. In general, the imposition of design-dependent local state-based constraints may require as many additional structural analysis runs as there are active elements in the FE mesh [40]. One way to circumvent the computational burden is by collecting all the constraints in a single aggregated (global) constraint, reducing the size of the optimization problem, and the cost of the adjoint analysis[7] [122, 133]. However, sequential approximate optimization (SAO) techniques may exhibit unstable numerical behaviour in the presence of global constraint formulations, typically based on highly nonlinear (and parameter specific) P-norm or Kreisselmeier-Steinhauser (KS) functions [2, 133].

In SAND—the 'direct' formulation—the structural analysis phase is *consumed* by the optimization algorithm—or indeed, *vice versa*. The FE equilibrium equations (6.3) are retained as a set of nonlinear equality constraints, forming part of an unified mathematical program. Therefore, structural analysis occurs sequentially, and the equilibrium conditions are only satisfied at convergence of the optimization algorithm—see for example Reference [33]. This is somewhat unconventional, and SAO procedures (specifically in structural optimization) are traditionally not geared—or presented as such—for problems with nonlinear equality constraints. The eminent 'method of moving asymptotes' (MMA) [36], for example, and particularly its convergent counterpart (GC-MMA) [67], are restricted to inequality constrained problems. Technological and behavioural constraints—the constraints imposed over and above the equilibrium constraints—are naturally formulated as one-sided limits on physical quantities; with the equilibrium constraints effectively removed from the optimization problem in the conventional NAND approach—hence 'reduced'—only the inequality constraints remain.

Convergent sequential optimization methods do however exist for general mathematical programming—*i.e.*, problems with nonlinear inequality and equality constraints. The convergent trust-region algorithm due to Fletcher *et al.* [66] is one example. (Line searches may also be used, but not preferred [85].) In SAO, convergent (*i.e.*, terminating) optimization methods

[7] In terms of explicit *local* constraints, Bruggi and Duysinx [43] note that, in spite of formulations ostensibly devised to reduce the number of active stress constraints—see also the 'Damage approach' [126]—the 'level of reduction might not be be enough to provide an efficient methodology for such kind of problems (mainly due to the cost of the adjoint analysis)'.

may be cast in a 'conditional acceptance of iterates' framework [74]. Moreover, the mechanism of "conservatism"—the cornerstone of the convergent MMA [67]—may be demonstrable in the presence of nonlinear equality constraints [75].

SAO methods are distinguished by the use of reciprocal-like intervening variables in order to construct accurate and efficient—strictly *convex* and *separable*—approximate subproblems—*e.g.* the 'Convex Linearization method' (CONLIN) [34, 35], the MMA [36], SCP-IP [45] and SAO*i* [41][8]. In general however, SAO may be described succinctly in a sequential diagonal quadratic programming (QP) framework [65], based on incomplete Taylor series expansions [63] and so-called 'approximated approximations' [64].

In the conventional NAND setting, reciprocal intervening variables (and the associated approximations) capture the implicit functional relationship between density ('design') variables $\boldsymbol{\rho}$ and displacement ('state') variables $\boldsymbol{u}$, imposed by the structural analysis step prior to each design iteration—*i.e.*, the calculation of $\boldsymbol{u}$ for a given $\boldsymbol{\rho}$ via (6.3). Moreover, curvature approximations based on intervening variables circumvent the additional structural analyses that are required if the subproblems are constructed with exact second-order information. In SAND, the abovementioned functional relationship does not exist—design $\boldsymbol{\rho}$ and state variables $\boldsymbol{u}$ are independent optimization variables—and thus it may be said that intervening variables are redundant[9]. Moreover, exact second-order sensitivity information is readily available in the form of simple partial derivatives [44].

Sequential quadractic programming (SQP) methods are closely related to SAO methods, but differ in that the full Hessian matrix—*i.e.*, not diagonal and not separable—is used in the approximate subproblems. In SAND in particular, due to the cheap availability of second-order information, QP subproblems with full Hessian matrices seem intuitively attractive—see the sophisticated second-order SQP method proposed by Rojas-Labanda and Stolpe [130] (and the associated benchmarking study [44]). In both cases the authors apply the method to NAND and SAND formulations of simply constrained topology design problems[10]: the use of 'exact' second-order information is shown to improve convergence properties.

SQP methods may however be at a disadvantage in view of large-scale (SAND) structural optimization[11]: first, the computational requirements associated with the full Hessian matrix—*i.e.,* calculation and storage—may be prohibitive in general; and second, ensuring positive-definiteness of a nondiagonal Hessian matrix—*i.e.*, not separable—requires nontrivial computational procedures (which are, in general, rather expensive). In SAO, positive-definiteness of the Hessian matrix (and the convexity of the primal approximate subproblem) is enforced trivially [71], and more importantly, the consequential sparsity of the diagonal QP subproblem is maintained in the Karush-Kuhn-Tucker (KKT) linear system of equations—the equations on which most of the computational effort is spent [45, 130]. Indeed, the use of 'exact' second-

[8]The term 'SAO' usually implies the use of dual methods for solving the approximate subproblems—specifically the dual of Falk [38]—but nowadays Lagrange-Newton diagonal quadratic subproblems in combination with primal-dual interior-point subsolvers is considered superior [33, 45, 46]

[9]In classical NAND topology design the QP 'approximated approximations' approach—based on *reciprocal* intervening variables—is equivalent to traditional Optimality Criteria (OC) methods [48]. In Reference [33] this equivalence is extended to the SAND setting—using *direct* variables.

[10]That is, subject to either a volume constraint, or a global compliance constraint—not, however, *local*, *design-dependent*, *state-based* constraints.

[11]*i.e.*, with local, design dependent, state-based constraints.

order information may result in improved convergence properties, but with convergence and termination enforced anyway in a 'conditional acceptance of iterates' framework [74], the speed with which the subproblems can be constructed and solved might as well take precedent.

In closing this introduction, we wish to highlight a recent contribution by Rong *et al.* [131]. The authors derive a sophisticated active constraint technique based on quadratic approximations and the MMA—in a NAND setting. Reportedly, 'there are 21 empirical [*heuristic*] parameters to be determined'. The numerical experiments are in the order of 5000 elements, and the the paper contains about 50 equations (some of which are, it is fair to say, quite complicated). 'Sequential approximate simultaneous analysis and design' (SEASAND)—as delineated in Section 6.3—may point the way to a simpler yet more efficient alternative. Indeed, the numerical experiments herein—Section 6.4—contains one of the largest local stress-constrained problems presented in the open literature to date.

6.3 SEASAND

6.3.1 Function approximation

Given a function $f(\boldsymbol{x})$, a diagonal quadratic approximation [65] based on an incomplete Taylor series expansion [63], constructed at a fixed point $\boldsymbol{x}^k$, may be written as

$$q^k(\boldsymbol{s}^k) = f(\boldsymbol{x}^k) + \boldsymbol{s}^{k,T}\boldsymbol{\nabla} f(\boldsymbol{x}^k) + \frac{1}{2}\boldsymbol{s}^T[\boldsymbol{C}^k]\boldsymbol{s}\ , \tag{6.4}$$

with $\boldsymbol{s}^k = (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$ the relative step, and $\boldsymbol{C}^k$ an appropriate approximate *diagonal* Hessian matrix. Herein vectors are column vectors (the elements are arranged row-wise) by default—*e.g.,* the gradient $\boldsymbol{\nabla} f(\boldsymbol{x}^k) \in \mathcal{R}^{n\times 1}$, abbreviated to $\boldsymbol{\nabla} f^k$ from here-on in.

The diagonal Hessian matrix $\boldsymbol{C}^k$ is denoted by

$$\boldsymbol{C}^k = \boldsymbol{I}\boldsymbol{c}^k \in \mathcal{R}^{n\times n}\ , \tag{6.5}$$

with $\boldsymbol{c}^k$ the diagonal entries—or 'curvatures'—stored in a vector of dimension $\mathcal{R}^n$. At this point, any number of approximations may be devised [63, 64]. In the past we have, for example, used analytic ('exact') diagonal partial derivatives with respect to Problem $\mathcal{P}_{\mathrm{S}}$, see Reference [33] (there is, of course, no reason for analytic diagonal information to be any good). In SEASAND, per default, a problem independent approximation is employed: the *spherical* quadratic approximation

$$\boldsymbol{c}^k = c^k\mathbf{1}\ , \tag{6.6}$$

as per Snyman and Hay [134]. The *spherical* quadratic approximation is said to do 'very well when applied to extremely ill-conditioned problems' [134]. Equation (6.6) implies the calculation (and storage) of a single term c^k, obtained by interpolating between the known function values at the current point (k) and the previous point $(k-1)$:

$$c^k = \frac{2\left[f(\boldsymbol{x}^{k-1}) - f(\boldsymbol{x}^k) - \boldsymbol{s}^{k-1,T}\boldsymbol{\nabla} f^k\right]}{\boldsymbol{s}^{k-1,T}\boldsymbol{s}^{k-1}}\ . \tag{6.7}$$

6.3.2 Primal approximate subproblem

In terms of a general nonlinear mathematical program $\mathcal{P}_{\text{NLP}}$, written as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_j(\boldsymbol{x}) \, [=, \leq] \, 0, \quad j = 1, 2, \ldots, m \\ & \check{\boldsymbol{x}} \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}} \, , \end{aligned} \tag{6.8}$$

wherein each f_j is either an equality or inequality constraint—*i.e.*, $[=, \leq]$—the *spherical* quadratic approximation (6.7) is used to approximate the objective f_0 and every constraint function f_j, $j = 1, 2, \ldots, m$, similar to the Dynamic-Q optimization method [135]. The primal approximate subproblem $P_{\text{P}}[k]$, at the current point $\boldsymbol{x}^k$, is written as

$$\begin{aligned} \min_{\boldsymbol{s}^{\{k\}}} \quad & q_0^k(\boldsymbol{s}^k) \\ \text{subject to} \quad & q_j^k(\boldsymbol{s}^k) \, [=, \leq] \, 0, \quad j = 1, 2, \ldots, m \\ & \boldsymbol{x}^{k+1} = (\boldsymbol{s}^k + \boldsymbol{x}^k) \in \Delta^k \, . \end{aligned} \tag{6.9}$$

The objective approximation is

$$q_0^k(\boldsymbol{s}^k) = f_0(\boldsymbol{x}^k) + \boldsymbol{\nabla} f_0^{k,T} \boldsymbol{s}^k + \frac{1}{2} c_0^k \boldsymbol{s}^{k,T} \boldsymbol{s}^k \tag{6.10}$$

and the constraint approximations are

$$q_j^k(\boldsymbol{s}^k) = f_j(\boldsymbol{x}^k) + \boldsymbol{\nabla} f_j^{k,T} \boldsymbol{s}^k + \frac{1}{2} c_j^k \boldsymbol{s}^{k,T} \boldsymbol{s}^k \tag{6.11}$$

for $j = 1, 2 \ldots, m$.

It is customary to control the size of the generated step $\|\boldsymbol{s}^k\|$ with a 'trust-region'—also known as 'restricted step methods' [89]—delimiting a 'trusted' neighbourhood Δ^k around the current iteration point $\boldsymbol{x}^k$. This may take the form of a proportional move-limit $0 < \delta^k < \epsilon_\delta \leq 1$, such that

$$\Delta^k = \{\boldsymbol{x}^{k+1} \in \mathcal{C} \mid \boldsymbol{s}^k \leq \delta^k (\hat{\boldsymbol{x}} - \check{\boldsymbol{x}})\} \, , \tag{6.12}$$

and $\mathcal{C}$ the domain of the global (actual) problem $\mathcal{P}_{\text{NLP}}$:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathcal{R}^n \mid \check{\boldsymbol{x}} \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}}\} \, , \tag{6.13}$$

written in terms of the lower $\check{\boldsymbol{x}}$ and upper bounds $\hat{\boldsymbol{x}}$ on each design variable x_i, for $i = 1, 2, \ldots, n$, respectively. Numerical experience suggests that, to reduce the frequency with which infeasible iterates occur, the proportional move-limit δ^k should be limited to an upper bound $\delta^k \leq \epsilon_\delta$ less than 1 (*e.g.* 0.2).

In general, due to the presence of nonlinear equality constraints—which typically describe nonconvex feasible regions [136]—subproblem $\mathcal{P}_{\text{P}}[k]$ is nonconvex. Moreoever, feasibility can not guaranteed in a straightforward manner—compared with, for example, Svanberg's [67] artificial variable approach for inequality constrained problems. In Dynamic-Q a penalty method is used to determine the 'best possible compromised solution' of $\mathcal{P}_{\text{P}}[k]$ if no feasible solution exists [135]. The version of the algorithm presented herein does not employ a relaxation strategy or sophisticated restoration procedure—both of which should be considered in future work.

6.3.3 Quadratic programming and Newton's Method

Linearising the constraint functions in subproblem $\mathcal{P}_{\mathrm{P}}[k]$ yields a simple QP subproblem, $\mathcal{P}_{\mathrm{Q}}[k]$, which may be written as

$$\begin{aligned} \min_{\boldsymbol{s}^k} \quad & \tilde{q}_0^k(\boldsymbol{s}^k) \\ \text{subject to} \quad & \tilde{q}_j^k(\boldsymbol{s}^k)\ [=, \leq]\ 0, \quad j = 1, 2, \ldots, m \\ & \boldsymbol{x}^{k+1} = (\boldsymbol{s}^k + \boldsymbol{x}^k) \in \Delta^k\ , \end{aligned} \tag{6.14}$$

wherein the approximate constraint functions have reduced to linear constraints

$$\tilde{q}_j^k(\boldsymbol{s}^k) = f_j(\boldsymbol{x}^k) + \boldsymbol{\nabla} f_j^{k,T} \boldsymbol{s}^k\ , \tag{6.15}$$

and a single curvature term c^k is retained in the approximate objective

$$\tilde{q}_0^k(\boldsymbol{s}^k) = f_0(\boldsymbol{x}^k) + \boldsymbol{\nabla} f_0^{k,T} \boldsymbol{s}^k + \frac{1}{2} c^k \boldsymbol{s}^{k,T} \boldsymbol{s}^k\ . \tag{6.16}$$

Strict convexity $c^k > 0$ is readily enforced (see below), and feasibility—associated with a specialized version of 'linear independence constraints qualification' (LICQ)—may hold in many practical situations [33, 88, 117].

Given that constraint functions in $\mathcal{P}_{\mathrm{Q}}[k]$ are linear, how can the curvature information in subproblem $\mathcal{P}_{\mathrm{P}}[k]$ be related to the single curvature approximation c^k in $\mathcal{P}_{\mathrm{Q}}[k]$? On the one hand, consider an approximate Lagrangian $\tilde{\mathcal{L}}^k(\boldsymbol{s}^k, \boldsymbol{\lambda}^{k+1})$, corresponding to subproblem $\mathcal{P}_{\mathrm{Q}}[k]$:

$$\begin{aligned} \tilde{\mathcal{L}}^k(\boldsymbol{s}^k, \boldsymbol{\lambda}^{k+1}) = & \boldsymbol{\nabla} f_0^{k,T} \boldsymbol{s}^k + \frac{1}{2} c^k \boldsymbol{s}^{k,T} \boldsymbol{s}^k \\ & + \boldsymbol{\lambda}^{k+1} \left[\boldsymbol{f}^k + \boldsymbol{J}^k \boldsymbol{s}^k \right]\ , \end{aligned} \tag{6.17}$$

wherein the Jacobian matrix $\boldsymbol{J}^k$ is composed of transposed gradients

$$\boldsymbol{J}^k = \begin{bmatrix} \boldsymbol{\nabla} f_1^{k,T} \\ \boldsymbol{\nabla} f_2^{k,T} \\ \vdots \end{bmatrix} \in \mathcal{R}^{m \times n}\ . \tag{6.18}$$

The vector of Lagrange multipliers associated with the minimizer $\boldsymbol{x}^{k+1}$ (expressed in terms of the step $\boldsymbol{s}^k$), is denoted by $\boldsymbol{\lambda}^{k+1} \in \mathcal{R}^m$. For the sake of brevity—here we are concerned with *curvature* information—the problem is simplified to an unbounded and equality-constrained version—whether a variable bound or inequality constraint is active/inactive is dealt with automatically by many modern QP routines, see for example References [33, 88].

The Kuhn-Tucker stationary conditions of the Lagrangian $\tilde{\mathcal{L}}^k(\boldsymbol{s}^k, \boldsymbol{\lambda}^{k+1})$ with respect to the primal variables $\boldsymbol{s}^k$ and dual variables $\boldsymbol{\lambda}^{k+1}$, may be expressed as

$$\begin{aligned} \boldsymbol{\nabla}_{\boldsymbol{x}} \tilde{\mathcal{L}}^k(\boldsymbol{s}^k, \boldsymbol{\lambda}^{k+1}) = \boldsymbol{\nabla} f_0(\boldsymbol{s}^k) + c^k \boldsymbol{s}^k + \boldsymbol{J}^{k,T} \boldsymbol{\lambda}^{k+1} = \boldsymbol{0} \\ \boldsymbol{\nabla}_{\boldsymbol{\lambda}} \tilde{\mathcal{L}}^k(\boldsymbol{s}^k, \boldsymbol{\lambda}^{k+1}) = \boldsymbol{f}^k + \boldsymbol{J}^k \boldsymbol{s}^k = \boldsymbol{0}\ . \end{aligned} \tag{6.19}$$

That is, the stationary point $(\boldsymbol{s}^k, \boldsymbol{\lambda}^{k+1})$ of (6.17) is a solution to the linear system

$$\begin{bmatrix} [c\boldsymbol{I}] & [\boldsymbol{J}]^T \\ [\boldsymbol{J}] & [0] \end{bmatrix}^k \begin{bmatrix} \boldsymbol{s}^k \\ \boldsymbol{\lambda}^{k+1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\nabla} f_0 \\ \boldsymbol{f} \end{bmatrix}^k . \tag{6.20}$$

On the other hand, consider the Lagrangian $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ of the actual (global) nonlinear problem $\mathcal{P}_{\text{NLP}}$—approximated by $\mathcal{P}_{\text{P}}[k]$, and in turn, $\mathcal{P}_{\text{Q}}[k]$—expressed in similar terms

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f_0(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}) , \tag{6.21}$$

and the associated Kuhn-Tucker stationary conditions

$$\begin{aligned} \boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{\nabla} f_0(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{\lambda} = \boldsymbol{0} \\ \boldsymbol{\nabla}_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} . \end{aligned} \tag{6.22}$$

The equations in (6.22) are a set of $n + m$ nonlinear equations

$$\boldsymbol{F}(\boldsymbol{Y}) = \begin{bmatrix} \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \\ \boldsymbol{f}(\boldsymbol{x}) \end{bmatrix} = \boldsymbol{0} \tag{6.23}$$

in $n + m$ unknowns

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} . \tag{6.24}$$

According to Newton's method[12], the vector of unknowns $\boldsymbol{Y}$ may be determined in an iterative fashion

$$\boldsymbol{Y}^{k+1} = \boldsymbol{Y}^k + \boldsymbol{\Delta Y} , \tag{6.25}$$

wherein $\boldsymbol{\Delta Y}$ is a solution to the linear system

$$[\boldsymbol{\nabla F}]^k \Delta \boldsymbol{Y} = -\boldsymbol{F}(\boldsymbol{Y}^k) , \tag{6.26}$$

alternatively written as

$$\begin{bmatrix} [\boldsymbol{\nabla}^2_{\boldsymbol{xx}} \mathcal{L}] & [\boldsymbol{J}]^T \\ [\boldsymbol{J}] & [0] \end{bmatrix}^k \begin{bmatrix} \boldsymbol{s}^k \\ \boldsymbol{\lambda}^{k+1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\nabla} f_0 \\ \boldsymbol{f} \end{bmatrix}^k . \tag{6.27}$$

Except for the full Hessian matrix of the Lagrangian $\boldsymbol{\nabla}^2_{\boldsymbol{xx}} \mathcal{L}^k$ *vs.* the spherical diagonal Hessian matrix $c^k \boldsymbol{I}$, the linear system in (6.27) is exactly the linear system in (6.20). This suggests that c^k should, in some way, mimic or capture the curvature of the Lagrangian

$$c^k = c_0^k + \sum_{j=1}^{m} \lambda_j^k c_j^k , \tag{6.28}$$

wherein $\boldsymbol{\lambda}^k$ is the vector of Lagrange multipliers retained from the stationary point $(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$ of the previous convex (and feasible) subproblem $\mathcal{P}_{\text{Q}}[k-1]$. The multipliers at the solution of the current subproblem $\mathcal{P}_{\text{Q}}[k]$, $\boldsymbol{\lambda}^{k+1}$, are of course, unknown.

[12] We follow the development by Rao [137].

The equivalence between, on the one hand, Netwon's Method applied to the stationary conditions of the Lagrangian, and, on the other, a QP-based SAO method for constrained nonlinear optimization, is consistent with the superiority of the '*nonconvex*' approximation strategy over the '*convex*' variant—see Reference [71]. In short, the '*nonconvex*' approximation strategy is defined as

$$c^k \leftarrow \max(c^k, \epsilon_c) \,, \tag{6.29}$$

with ϵ_c typically in order of 10^{-6}. Therefore, although $\mathcal{P}_{\mathrm{Q}}[k]$ is strictly convex $c^k > 0$, some 'nonconvex' curvature information $\lambda_j^k c_j^k < 0$ may be 'retained' in the curvature summation (6.28). Compare this with the the '*convex*' approximation strategy, which dictates that convexity be enforced on the approximate objective function—$c_0^k \geq \epsilon_c$—and on each and every constraint contribution—$\max(\lambda_j c_j^k, 0)$, for $j = 1, 2, \ldots, k$—in order for the curvature summation (6.28) to turn out strictly positive[13].

By invoking the equivalence between the approximate curvature of the Lagrangian c^k and the 'numerical damping' term often found in optimality criteria (OC) methods [33, 41]—which may be extendible to the notion of 'conservatism' [67]—an argument can be made for an 'absolutely nonconvex' approximation strategy

$$c^k \leftarrow \max(|c^k|, \epsilon_c) \,. \tag{6.30}$$

Clearly, the approximation in (6.30) should 'damp' any oscillatory behaviour in regions where the approximate Lagrangian has 'high curvature', regardless of sign.

In Section 6.4 numerical experiments are used to test the *absolutely nonconvex* variant (6.30) against the original *nonconvex* approximation strategy (6.29)—and by implication, the asserted equivalence between approximate curvature c^k and 'numerical damping'. Next, the computational character of the SEASAND method is portrayed from a theoretical point of view.

6.3.4 Sequential approximate analysis and design

In keeping with classical SAO—based on the dual of Falk [33, 38]—the equations in (6.20) may be separated into the primal stationary conditions

$$c^k \boldsymbol{I} \boldsymbol{s}^k + \boldsymbol{J}^{k,T} \boldsymbol{\lambda}^{k+1} = -\boldsymbol{\nabla} f_0^k \in \mathcal{R}^n \tag{6.31}$$

and the dual stationary conditions

$$\boldsymbol{J}^k \boldsymbol{s}^k = -\boldsymbol{f} \in \mathcal{R}^m \,. \tag{6.32}$$

Equation (6.31) yields a relation between primal and dual variables

$$\boldsymbol{s}^k = -\frac{1}{c^k} \left[\boldsymbol{J}^{k,T} \boldsymbol{\lambda}^{k+1} + \boldsymbol{\nabla} f_0^k \right] \in \mathcal{R}^n \,, \tag{6.33}$$

which, substituted into (6.32) and rearranged as

$$\boldsymbol{J}^k \boldsymbol{J}^{k,T} \boldsymbol{\lambda}^{k+1} = c^k [\boldsymbol{f}^k - \boldsymbol{J}^k \boldsymbol{\nabla} f_0^k] \in \mathcal{R}^m \,, \tag{6.34}$$

[13] As is customary in SAO literature, the analysis in Reference [71] is restricted to problems with inequality constraints, viewed in standard negative-null form—the associated Lagrange multipliers are limited to $\lambda_j \geq 0$, for $j = 1, 2, \ldots, m$—hence, c_j^k retains its sign in $c_j^k \lambda_j$.

illustrates that the calculation of a solution to the linear system (6.20) hinges on the 'computational kernel' $\boldsymbol{J}^k\boldsymbol{J}^{k,T}$—as per Fleury [40]. The dual computational kernel $\boldsymbol{J}^k\boldsymbol{J}^{k,T}$ is derived for the purposes of illustration: modern primal-dual methods have increased flexibility in solving the primal-dual system of equations (6.20), with increased potential to maintain its sparsity [45].

In order to further investigate the computational kernel $\boldsymbol{J}^k\boldsymbol{J}^{k,T}$ particular to Problem $\mathcal{P}_{\mathrm{S}}$, consider the slope constraints relaxed $\mu\pi > 1$ (and thus inactive); the problem has $n = d + e$ primal variables and an equal number of constraints (dual variables) $m = d + e$. (Using the dual of Falk [33, 38], the variable bounds may be neglected in this way.) Conveniently, given a generic iteration point $\boldsymbol{x}^k$, the Jacobian associated with $\mathcal{P}_{\mathrm{Q}}[k]$ is a square matrix $\boldsymbol{J}^k \in \mathcal{R}^{n\times n}$ (or $\mathcal{R}^{m\times m}$). Therefore, the computational kernel, which may be thought of as the 'inversion' of $\boldsymbol{J}^k\boldsymbol{J}^{k,T}$, can be simplified to

$$\left[\boldsymbol{J}^k\boldsymbol{J}^{k,T}\right]^{-1} = \left[\boldsymbol{J}^k\right]^{-1}\left[\boldsymbol{J}^{k,T}\right]^{-1} , \tag{6.35}$$

with the Jacobian matrix $\boldsymbol{J}^k$, particular to Problem $\mathcal{P}_{\mathrm{S}}$ with $\mu\pi > 1$, written as

$$\left[\boldsymbol{J}^k\right] = \begin{bmatrix} [\boldsymbol{K}(\boldsymbol{\rho}^k)] & [\boldsymbol{\nabla}_{\boldsymbol{\rho}}\boldsymbol{K}(\boldsymbol{\rho}^k)\boldsymbol{u}^k] \\ \frac{1}{\bar{\sigma}}[\boldsymbol{\nabla}_{\boldsymbol{u}}\boldsymbol{\rho}^k \diamond \boldsymbol{\sigma}(\boldsymbol{u}^k)] & [\frac{1}{\bar{\sigma}}\boldsymbol{I}(\boldsymbol{\sigma}(\boldsymbol{u}^k) - \bar{\sigma}\mathbf{1})] \end{bmatrix} . \tag{6.36}$$

The density gradients $\boldsymbol{\nabla}_{\boldsymbol{\rho}}$ of the equilibrium constraints is denoted by

$$\boldsymbol{\nabla}_{\boldsymbol{\rho}}\boldsymbol{K}(\boldsymbol{\rho}^k)\boldsymbol{u}^k \in \mathcal{R}^{d\times e} , \tag{6.37}$$

the displacement (state) gradients $\boldsymbol{\nabla}_{\boldsymbol{u}}$ of the local stress constraints is denoted by

$$\frac{1}{\bar{\sigma}}[\boldsymbol{\nabla}_{\boldsymbol{u}}\boldsymbol{\rho}^k \diamond \boldsymbol{\sigma}(\boldsymbol{u}^k)] \in \mathcal{R}^{e\times d} , \tag{6.38}$$

and the diagonal matrix composed of the 'nonvanishing'[14] part of the stress constraint vector, is written as

$$\frac{1}{\bar{\sigma}}\boldsymbol{I}(\boldsymbol{\sigma}(\boldsymbol{u}^k) - \bar{\sigma}\mathbf{1}) \in \mathcal{R}^{e\times e} . \tag{6.39}$$

If material density variables are limited to nonzero values $\rho_i^k > 0$, for $i = 1, 2, \ldots, e$, the globally assembled stiffness matrix $\boldsymbol{K}(\boldsymbol{\rho})$ is nonsingular. In addition, if each and every nonvanishing stress constraint is active, then the nonvashining stress constraint vector $\frac{1}{\bar{\sigma}}(\boldsymbol{\sigma}(\boldsymbol{u}^k) - \bar{\sigma}\mathbf{1})$ is all zeros: $\mathbf{0} \in \mathcal{R}^e$. Using block-inversion (in terms of a 'partitioned' matrix) [138], it can be demonstrated that the 'inversion' of $[\boldsymbol{J}^k]$—by which the calculation of the solution to the linear system in (6.34) is implied—may be reduced to the 'inversion' of the stiffness matrix

$$[\boldsymbol{K}(\boldsymbol{\rho}^k)] \in \mathcal{R}^{d\times d} , \tag{6.40}$$

and, following that, the 'inversion' of the matrix product

$$[\boldsymbol{\nabla}_{\boldsymbol{u}}\boldsymbol{\rho}^k \diamond \boldsymbol{\sigma}(\boldsymbol{u}^k)][\boldsymbol{K}(\boldsymbol{\rho}^k)]^{-1}[\boldsymbol{\nabla}_{\boldsymbol{\rho}}\boldsymbol{K}(\boldsymbol{\rho}^k)\boldsymbol{u}^k] \in \mathcal{R}^{e\times e} . \tag{6.41}$$

For the sake of argument, assume that the linear system solver tasked with the 'structural analysis'—equivalent to, conceptually speaking, the inversion of (6.40)—requires $\mathcal{O}(d^\eta)$ operations, with η a constant number $\eta > 1$. Assume that the aforementioned solver can be used

[14] See Reference [117] and Reference [88] for details on vanishing local stress constraints.

to 'invert' (6.41) as well. The number of degrees of freedom d is related by a constant factor γ to the number of elements e. Thus, the overall computational effort required to solve $\mathcal{P}_{\mathrm{Q}}[k]$ may be written as

$$\mathcal{O}((e\gamma)^{\eta}) + \mathcal{O}(e^{\eta}) \tag{6.42}$$

which, in terms of asymptotic complexity, simplifies to

$$\mathcal{O}(e^{\eta}) \, . \tag{6.43}$$

That is, using SEASAND, the QP routine—tasked with solving $\mathcal{P}_{\mathrm{Q}}[k]$—conducts the equivalent of a single structural analysis—requiring $\mathcal{O}(e^{\eta})$ operations—per iteration k. Conventional NAND algorithms, which require multiple structural analyses prior to every design iteration—for problems with local state-based constraints—can not scale better than $\mathcal{O}(e^{\eta+1})$ [88]. Assume, for example, the availability of an ideal preconditioning routine and sparse linear system solver with computational scaling properties close to $\mathcal{O}(e)$. Consequently (using this hypothetical linear system solver), an SAO-NAND method can at best scale to $\mathcal{O}(e^2)$, but the equivalent SAO-SAND (SEASAND) method could—hypothetically speaking—scale to $\mathcal{O}(e)$.

Finally, infeasible instances of $\mathcal{P}_{\mathrm{Q}}[k]$ may indeed occur [88]—due to, for example, regions of very low density material (but not yet void) and an insufficiently large bound on $\boldsymbol{u}$ (or a too restrictive move-limit δ^k). In the following subsection the SEASAND algorithm is outlined, which, in basic form, employs a simple backtracking procedure to circumvent, in part, the issue of infeasible subproblems.

6.3.5 Convergence and termination

In order to enforce convergence (*i.e.,* termination) the trust-region Δ^k is combined with a Pareto front (or 'filter') of the objective function value f_0 and the maximum constraint violation, denoted by Θ. A pseudo-code representation of SEASAND is given in Algorithm (6).

The filter $\mathcal{F}^k$ consists of the list of pairs from at most k previously visited points

$$\mathcal{F}^k \subset \{(\Theta^h, f_0^h) : \; h = 1, 2, \ldots, k\} \, . \tag{6.44}$$

Using a 'slanting envelope test' [66, 85] (*Step 5*), if either

$$\Theta^{k+1} \leq \epsilon_{\mathcal{F}} \Theta^h \quad \text{or} \quad f_0^{k+1} + (1 - \epsilon_{\mathcal{F}})\Theta^{k+1} \leq f_0^h \tag{6.45}$$

for all $h = 1, 2, \ldots, k$, the pair $(\Theta^{k+1}, f_0^{k+1})$ is deemed acceptable, and may pass through the filter. The envelope parameter is constant and restricted to $0 < \epsilon_{\mathcal{F}} < 1$. Following that (in *Step 5*), the new iterate $(\Theta^{k+1}, f_0^{k+1})$ has to pass one of two conditions: the first is reminiscent of the 'conservatism' of the objective function approximation $\tilde{q}_0^k$ relative to the actual value f_0^{k+1} at the new point

$$\frac{\Delta f_0^k}{\Delta \tilde{q}_0^k} := \frac{f_0^k - f_0^{k+1}}{\tilde{q}_0^k(\mathbf{0}) - \tilde{q}_0^k(\boldsymbol{s}^k)} > \epsilon_{\Delta_1} \, , \tag{6.46}$$

and the second, a feasible-descent-like condition

$$\tilde{q}_0^k(\mathbf{0}) - \tilde{q}_0^k(\boldsymbol{s}^k) := \Delta \tilde{q}_0^k \leq \epsilon_{\Delta_2} (\Theta^k)^{\epsilon_{\Delta_3}} \, , \tag{6.47}$$

Algorithm 6 (SEASAND)

Step 0:

Set problem parameters
$\check{\boldsymbol{x}} \in \mathcal{C}$, $\hat{\boldsymbol{x}} \in \mathcal{C}$, $\boldsymbol{x_0} \in \mathcal{C}$, $\boldsymbol{x}^g \leftarrow \boldsymbol{x}^0$

Set algorithmic parameters
ϵ_δ, ϵ_c, $\epsilon_{\mathcal{F}}$, ϵ_{Δ_1}, ϵ_{Δ_2}, ϵ_{Δ_3}, Δ_{min}, $k_{\max}$

for $k = 0 : k_{\max}$ **do**

Step 1:

Construct $\mathcal{P}_{\text{Q}}[k]$, attempt to solve $\mathcal{P}_{\text{Q}}[k]$

Step 2:

If $\mathcal{P}_{\text{Q}}[k]$ is infeasible, add (Θ^k, f_0^k) to $\mathcal{F}^k$, $\delta^k \leftarrow 1$, $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^g$

If $k = 0$, terminate.

If $\boldsymbol{x}^k = \boldsymbol{x}^{k-1}$, terminate.

Else return to 1

Step 3:

If $\mathcal{P}_{\text{Q}}[k]$ is feasible, set $\boldsymbol{\lambda}^{k+1}$ and $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \Delta \boldsymbol{s}^k$

Step 4:

Construct $\mathcal{P}_{\text{Q}}[k+1]$

Step 5:

Test filter and descent

if $\Theta^{k+1} \leq \epsilon_{\mathcal{F}} \Theta^h$ or $f_0^{k+1} + (1 - \epsilon_{\mathcal{F}})\Theta^{k+1} \leq f_0^h$, for $h = 1, 2, \ldots, k$ **then**

if $\frac{\Delta f_0^k}{\Delta \tilde{q}_0^k} > \epsilon_{\Delta_1}$ or $\Delta \tilde{q}_0^k < \epsilon_{\Delta_2} (\Theta^k)^{\epsilon_{\Delta_3}}$ **then**

Continue

else

$\delta^k \leftarrow \delta^k / 2$, return to 1

end if

else

$\delta^k \leftarrow \delta^k / 2$, return to 1

end if

Step 6:

Update the trust-region and filter

$\delta^{k+1} \leftarrow \min(2\delta^k, \epsilon_\delta)$, $\boldsymbol{x}^g \leftarrow \boldsymbol{x}^k$

if $\Delta \tilde{q}_0^k < \epsilon_{\Delta_2} (\Theta^k)^{\epsilon_{\Delta_3}}$ **then**

add $(\Theta^{k+1}, f_0^{k+1})$ to $\mathcal{F}^{k+1}$

end if

Continue

Step 7:

Check for termination

if $\|\Delta \boldsymbol{s}^k\| < \Delta_{\text{min}}$ **then**

Terminate.

end if

end for

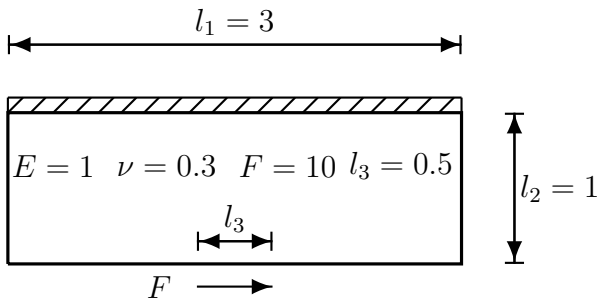


Figure 6.1: The two-bar truss problem.

with constants $0 < \epsilon_{\Delta_1} < 1, 0 < \epsilon_{\Delta_2} < 1$ and $\epsilon_{\Delta_3} > 1$. The new iterate $(\Theta^{k+1}, f_0^{k+1})$ is included in the filter list $\mathcal{F}^{k+1}$ and the move-limit δ^{k+1} may be expanded if the feasible-descent-like condition (6.47) is satisfied (*Step 6*).

In the case of an infeasible subproblem $\mathcal{P}_{\mathrm{Q}}[k]$, the backtracking procedure (*Step 2*) is activated: the filter pair (Θ^k, f_0^k) at the infeasible point k is included in the filter $\mathcal{F}^k$ (forcing, hopefully, a feasible iterate in the next attempt), the move-limit is expanded to $\delta^k = 1$ (hence the entire domain $\mathcal{C}$ may be traversed), and the algorithm is restored to the last accepted iterate $\boldsymbol{x}^g$.

The algorithm will either terminate because the backtracking procedure fails to produce a feasible subproblem $\mathcal{P}_{\mathrm{Q}}[k]$—*i.e.*, the rudimentary 'restoration procedure' fails—or converge to a stationary (KKT) point of problem $\mathcal{P}_{\mathrm{NLP}}$, or accumulate at a feasible point which fails to satisfy CQ [66].

The algorithmic parameters are set at the following default values: $\epsilon_\delta = 0.2$, $\epsilon_c = 1 \times 10^{-6}$, $\epsilon_{\mathcal{F}} = 1 \times 10^{-7}$, $\epsilon_{\Delta_1} = 0.01$, $\epsilon_{\Delta_2} = 0.001$ and $\epsilon_{\Delta_3} = 2$. In addition, a convergence tolerance $\Delta_{\min} > 0$ and a maximum number of iterations $k_{\max} > 0$ are specified by the user.

6.4 Numerical experiments

Consider the standard 'two-bar truss' design domain—see for example Reference [109]—depicted in Figure 6.1. We intend to investigate the computational properties of the SEASAND algorithm, and particularly the computational requirements associated with the SAND-based QP subproblem $\mathcal{P}_{\mathrm{Q}}[k]$, hence a simple ground-structure will suffice. The interested reader is referred to the aforementioned literature for additional details on the ground-structure (and local stress-constrained topology design in general). The von Mises stress measure is used for each element $\sigma_i(\boldsymbol{u})$, for $i = 1, 2, \ldots, n$, the stress-limit is $\bar{\sigma} = 20$, the slope parameter is set to $\mu = 25$, and the mesh-size parameter is $\pi = \frac{1}{e_2}$.

It is fair to say that slope constraints are employed specifically in order to suppress the (typically severe) multimodality of the problem—*i.e.*, the issue of many local minima—increasing the likelihood that the algorithm will consistently convergence to the same solution given different settings and starting positions, thereby permitting a fair comparison. To this end, considering more complex design domains, random multistart strategies—see for example References [33, 59, 88]—and Bayesian stopping criteria [61], should probably be resorted to.

Table 6.1: Large-scale two-bar truss design, nonconvex spherical quadratic approximation; computational requirements.

e	n	m	Fig.	$k^\star$	P	n_+^ρ	m_+^σ	T
270×90=24,300	171,542	268,022	6.2a	204	235	6,581	4,195	21,761
300×100=30,000	211,602	330,802	6.2b	222	298	7,965	4,870	44,327
330×110=36,300	255,862	400,182	6.2c	256	420	9,034	5,935	68,739
360×120=43,200	304,322	476,162	6.2d	219	259	11,498	7,808	61,532
390×130=50,700	356,982	558,742	6.2e	267	348	12,832	8,271	104,026
420×140=58,800	413,842	647,922	6.2f	319	522	14,785	9,512	285,612
450×150=67,500	474,902	743,702	6.2g	136†	692	31,782	18,891	564,959
480×160=76,800	540,162	846,082	6.2h	329	388	19,058	12,606	187,074
510×170=86,700	609,622	955,062	6.2i	283	347	21,725	16,847	185,125
540×180=97,200	683,282	1,070,642	6.2j	349	519	24,081	15,848	493,642
570×190=108,300	761,142	1,192,822	6.2k	360	429	26,895	16,871	468,794

The design domain is discretized with $e = e_1 \times e_2$ (square) Q8 FE's. In Table 6.1 the associated number of primal variables n and the total number of constraints m are given for each e in a range of mesh discretizations—from $24,300$ to $108,300$. The number of degrees of freedom—*i.e.*, the number of displacement variables and equilibrium constraints—is $d = n - e$, and the number of slope constraints (twice the number of interelement boundaries b) is $2b = m - e - d$.

The displacement variables $\boldsymbol{u}$ are bounded to $\pm 1 \times 10^3$. Throughout, a starting position of $\boldsymbol{\rho}^0 = \mathbf{1}$ and $\boldsymbol{u}^0 = \mathbf{0}$ is used. The default algorithmic settings surmised in Section 6.3.5—6 in total—are retained. The algorithm is deemed to have converged if the Euclidean norm of the step in primal optimization variables $\|\boldsymbol{s}^k\| \leq 1$, in which case the slope constraint parameter is increased by $1.1 \times \mu$, and the filter-list $\mathcal{F}^k$ is cleared—repeated until $\mu\pi \geq 1$, at which point the algorithm is allowed to terminate.

In Table 6.1 the results obtained using the SEASAND method with the *nonconvex* approximation strategy (6.29) is summarised. The number of iterations required for termination is $k^\star$. An iteration k may involve 'inner iterations' (see Section 6.3.5), therefore the total number of subproblems $\mathcal{P}_{\mathrm{Q}}[k]$ actually solved is given by P. At termination, the number of nonzero density variables $\rho_i > 0$ is n_+^ρ, and the total number of nonzero elements $\rho_i > 0$ stressed equal to or beyond $\sigma_i(\boldsymbol{u}) \geq 19$ is denoted by m_+^σ. The total time to convergence is given by T (wall-time, in seconds)[15]. In Figure 6.2 the corresponding topologies are depicted: the objective function value is $f_0^\star$, the maximum constraint violation is $\Theta^\star$, the number of material density variables equal to zero is n_0^ρ, the total number equal to 1 is n_1^ρ, and the solid-void fraction is $\Phi^\star$. The algorithm failed to terminate with a feasible subproblem—indicated by the 'dagger' (†) in Table 6.1—only once.

In Table 6.2 the results obtained using the *absolutely nonconvex* approximation strategy (6.30) is summarised. The associated designs are depicted in Figure 6.3. In all but one case

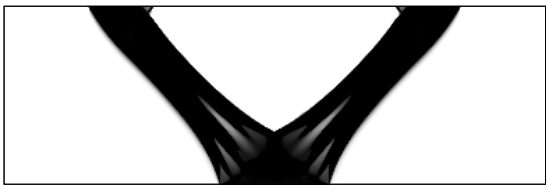
[15]The ILOG CPLEX Barrier QP optimizer [128] is employed as subsolver. The computational platform is the Rhasatsha HPC [77].

(a) $f_0^\star = .232$, $\Theta^\star = 3\text{E}{-3}$, $n_0^\rho = 17719$, $n_1^\rho = 4681$, $\Phi^\star = .89$.

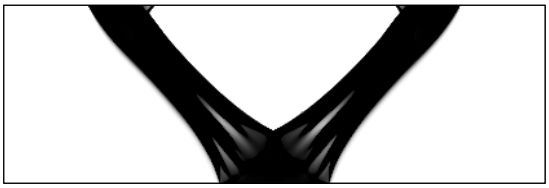
(b) $f_0^\star = .231$, $\Theta^\star = 8\text{E}{-3}$, $n_0^\rho = 22035$, $n_1^\rho = 5845$, $\Phi^\star = .91$.

(c) $f_0^\star = .225$, $\Theta^\star = 2\text{E}{-4}$, $n_0^\rho = 27266$, $n_1^\rho = 7014$, $\Phi^\star = .91$.

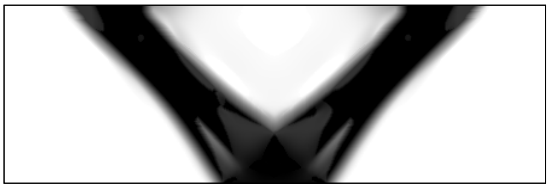
(d) $f_0^\star = .229$, $\Theta^\star = 1\text{E}{-3}$, $n_0^\rho = 31702$, $n_1^\rho = 8090$, $\Phi^\star = .89$.



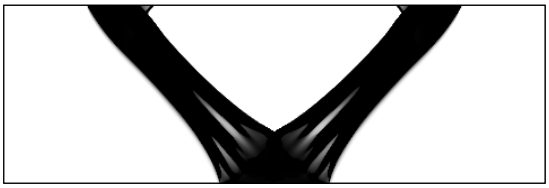
(e) $f_0^\star = .226$, $\Theta^\star = 5\text{E}{-3}$, $n_0^\rho = 37868$, $n_1^\rho = 9927$, $\Phi^\star = .91$.



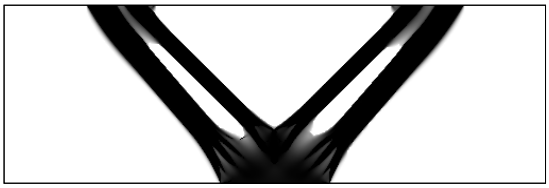
(f) $f_0^\star = .225$, $\Theta^\star = 5\text{E}{-4}$, $n_0^\rho = 44015$, $n_1^\rho = 11537$, $\Phi^\star = .91$.



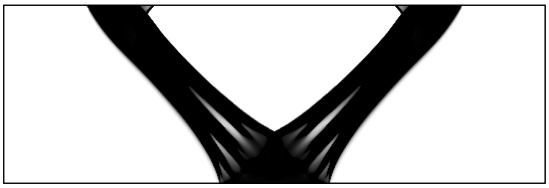
(g) $f_0^\star = .275$, $\Theta^\star = 7\text{E}{-3}$, $n_0^\rho = 35718$, $n_1^\rho = 10152$, $\Phi^\star = .66$.



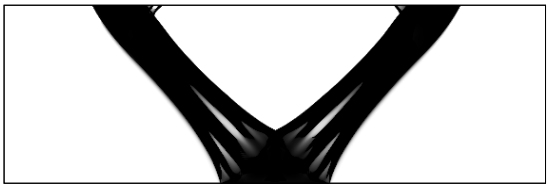
(h) $f_0^\star = .223$, $\Theta^\star = 1\text{E}{-3}$, $n_0^\rho = 57742$, $n_1^\rho = 15149$, $\Phi^\star = .91$.



(i) $f_0^\star = .223$, $\Theta^\star = 7\text{E}{-4}$, $n_0^\rho = 64975$, $n_1^\rho = 15818$, $\Phi^\star = .89$.



(j) $f_0^\star = .223$, $\Theta^\star = 3\text{E}{-3}$, $n_0^\rho = 73119$, $n_1^\rho = 19364$, $\Phi^\star = .91$.



(k) $f_0^\star = .224$, $\Theta^\star = 1\text{E}{-3}$, $n_0^\rho = 81405$, $n_1^\rho = 21729$, $\Phi^\star = .92$.

Figure 6.2: Large-scale two-bar truss designs; nonconvex spherical quadratic quadratic approximation.

the *absolutely nonconvex* variant of the algorithm required fewer iterations $k^\star$ for convergence, but, in every case, substantially fewer subproblem evaluations P are required, and hence, a substantial reduction in total computation time T. This in itself suggests the existence of 'conservatism'—or a mechanism to this end—in general nonlinear programming: the *absolutely nonconvex* approximation may result in 'conservative' iterates which pass the filter conditions (6.45), the relative-descent condition (6.46) and feasible-descent-like condition (6.47),

Table 6.2: Large-scale two-bar truss design, absolutely nonconvex spherical quadartic approximation; computational requirements

e	n	m	Fig.	$k^\star$	P	n_+^ρ	m_+^σ	T
270×90=24,300	171,542	268,022	6.3a	143	143	7,075	4,070	6,778
300×100=30,000	211,602	330,802	6.3b	168	168	7,701	4,822	15,719
330×110=36,300	255,862	400,182	6.3c	226	226	9,297	5,877	15,652
360×120=43,200	304,322	476,162	6.3d	213	213	10,942	6,800	25,508
390×130=50,700	356,982	558,742	6.3e	246	272	12,814	8,192	62,589
420×140=58,800	413,842	647,922	6.3f	276	276	14,830	9,157	62,078
450×150=67,500	474,902	743,702	6.3g	193	230	16,879	10,758	121,973
480×160=76,800	540,162	846,082	6.3h	247	289	19,109	12,001	124,347
510×170=86,700	609,622	955,062	6.3i	304	333	21,614	13,592	189,391
540×180=97,200	683,282	1,070,642	6.3j	317	334	24,179	15,512	157,465
570×190=108,300	761,142	1,192,822	6.3k	309	324	26,845	17,177	325,272
780×260=202,800	1,423,762	2,232,882	6.4a	389	411	101,645	31,106	617,985

more often.

The *absolutely nonconvex* (6.30) variant of the algorithm is used to solve one of the largest problem instances reported in stress-constrained topology design literature to date—depicted in Figure 6.4a—with more than $200,000$ material density variables (the *nonconvex* variant failed to converge). Figure 6.4b is a plot of the stress vector $\boldsymbol{\sigma}(\boldsymbol{u})$: for the sake of readability, all the von Mises stresses $\boldsymbol{\sigma}(\boldsymbol{u})$ equal to or greater than $\bar{\sigma} = 20$ are coloured red, blue regions represent elemental stresses at 0. Figure 6.4c is the same stress plot, but with the stresses $\sigma_i(\boldsymbol{u})$ related to zero elements $\rho_i = 0$ (*i.e.*, 'vanished') ignored—the maximum von Mises stress value is 20.74, and occurs in an element with density $\rho_i = 0.22$, the next highest value is 20.005.

Figure 6.5a is a plot of the total computation times from initialization to termination—included is a polynomial function fitted to the *absolutely nonconvex* data poins. Figure 6.5b is a plot of *average* computation times per subproblem evaluation, for the respective runs—again, a polynomial is fitted to the *absolutely nonconvex* data points. Because the slope constraints are more restrictive for finer mesh discretizations—the slope parameter product $(\mu)(\frac{1}{e_x})$ is smaller, initially—more continuation steps, and thus an increased number of iterations, are required for larger problem sizes. Therefore, the average time per subproblem evaluation provides a fair indication of the computational scaling properties of the SEASAND method. Crucially, average times per iteration scale practically the same as for simply-constrained problems [59], confirming the computational character, as asserted in Section 6.3.4, of the SEASAND method.

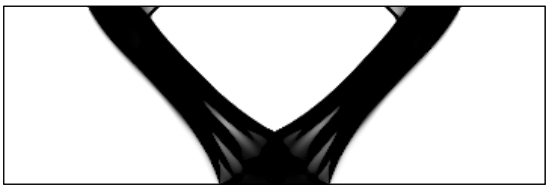
6.5 Concluding remarks

Although we have considered a fairly simple design domain—and limited empirical evidence—in the light of general theoretical considerations, the results suggest that the SEASAND method may offer very large-scale structural optimization capabilities. For more complex problems with many local minima, a global optimization procedure—like for example a multistart strategy—

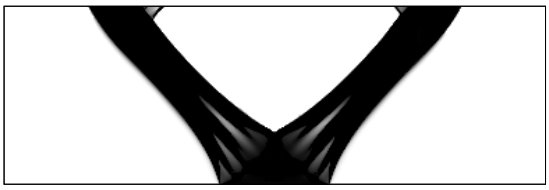
(a) $f_0^\star = .237$, $\Theta^\star = 2\text{E}{-3}$, $n_0^\rho = 17225$, $n_1^\rho = 4846$, $\Phi^\star = .89$.

(b) $f_0^\star = .227$, $\Theta^\star = 5\text{E}{-3}$, $n_0^\rho = 22299$, $n_1^\rho = 5910$, $\Phi^\star = .91$.

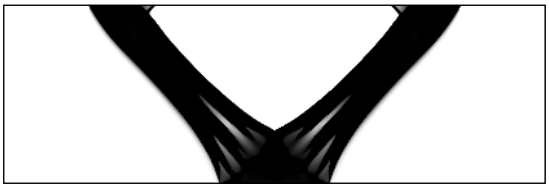
(c) $f_0^\star = .227$, $\Theta^\star = 1\text{E}{-2}$, $n_0^\rho = 27003$, $n_1^\rho = 7156$, $\Phi^\star = .91$.



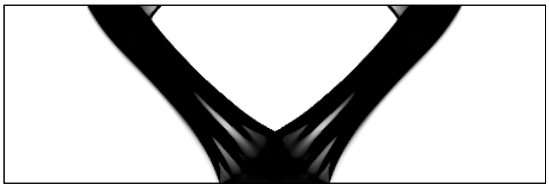
(d) $f_0^\star = .227$, $\Theta^\star = 4\text{E}{-3}$, $n_0^\rho = 32258$, $n_1^\rho = 8575$, $\Phi^\star = .91$.



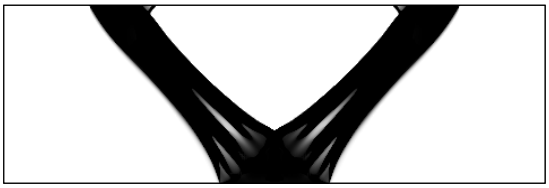
(e) $f_0^\star = .226$, $\Theta^\star = 5\text{E}{-3}$, $n_0^\rho = 37886$, $n_1^\rho = 9962$, $\Phi^\star = .91$.



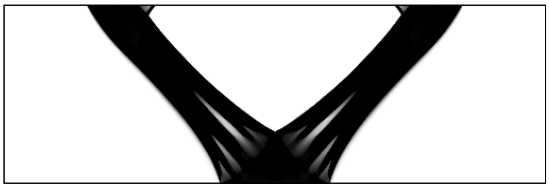
(f) $f_0^\star = .225$, $\Theta^\star = 9\text{E}{-3}$, $n_0^\rho = 43970$, $n_1^\rho = 11781$, $\Phi^\star = .92$.



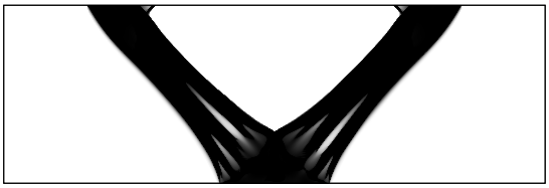
(g) $f_0^\star = .225$, $\Theta^\star = 3\text{E}{-3}$, $n_0^\rho = 50621$, $n_1^\rho = 13472$, $\Phi^\star = .92$.



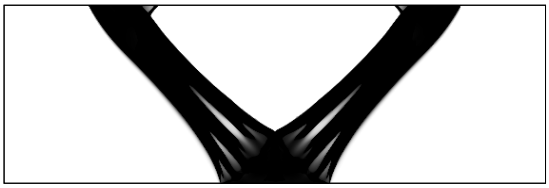
(h) $f_0^\star = .224$, $\Theta^\star = 1\text{E}{-3}$, $n_0^\rho = 57691$, $n_1^\rho = 15414$, $\Phi^\star = .92$.



(i) $f_0^\star = .224$, $\Theta^\star = 1\text{E}{-3}$, $n_0^\rho = 65086$, $n_1^\rho = 17392$, $\Phi^\star = .92$.



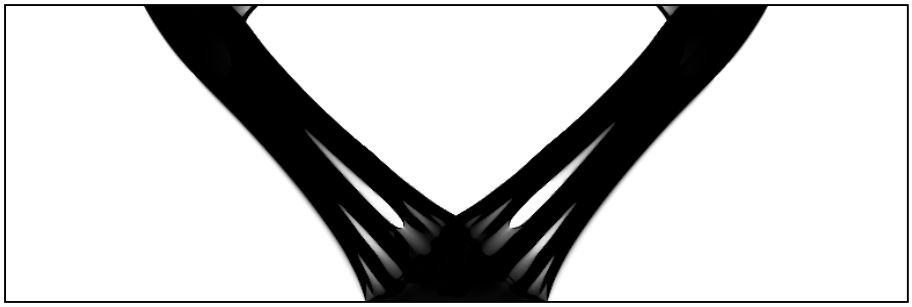
(j) $f_0^\star = .223$, $\Theta^\star = 3\text{E}{-3}$, $n_0^\rho = 73021$, $n_1^\rho = 19393$, $\Phi^\star = .91$.



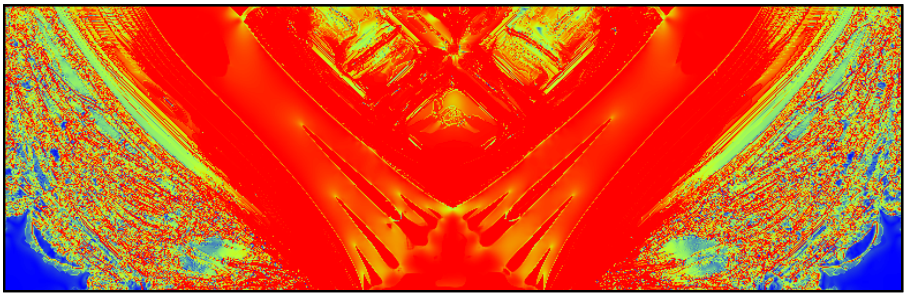
(k) $f_0^\star = .223$, $\Theta^\star = 1\text{E}{-3}$, $n_0^\rho = 81455$, $n_1^\rho = 21764$, $\Phi^\star = .92$.

Figure 6.3: Large-scale two-bar truss designs; absolutely nonconvex spherical quadratic approximation.

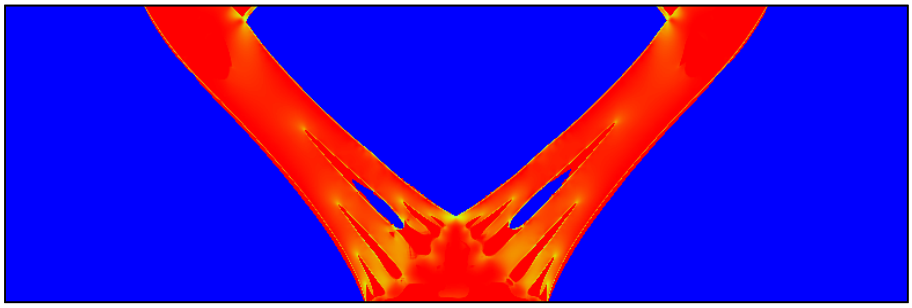
may be resorted to. To this end, the results presented herein are pertinent nevertheless, particularly due to the highly parallelizable nature of the algorithm. That is, each and every subproblem can be solved independently, hence, N randomly initialized problem instances may be solved in roughly the same amount of time that it takes to solve a single problem instance—given, of course, the availability of N sufficiently sized computational devices.



(a) $f_0^\star = .220$, $\Theta^\star = 2\text{E}{-4}$, $n_0^\rho = 153\,729$, $n_1^\rho = 41\,226$, $\Phi^\star = .92$.

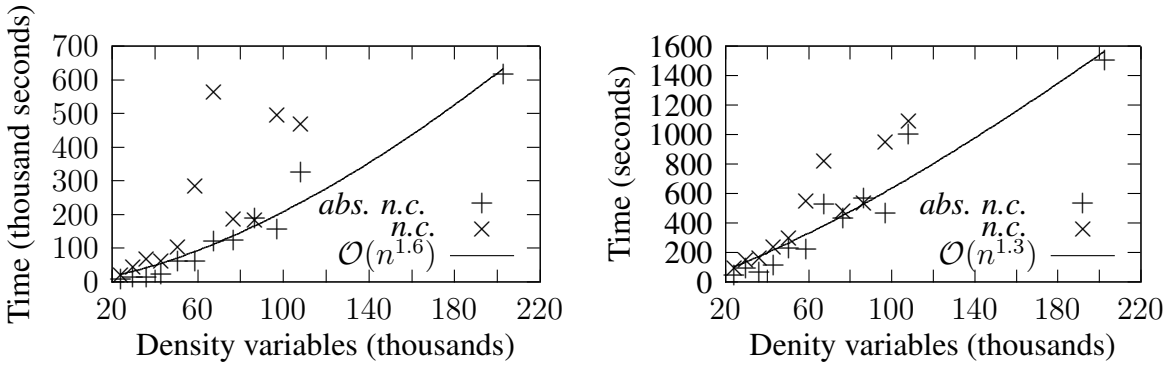


(b) $\max(\boldsymbol{\sigma}(\boldsymbol{u})) = 3200.16$.



(c) $\max(\boldsymbol{\sigma}(\boldsymbol{u})|\rho_i > 0) = 20.74$.

Figure 6.4: Very large-scale two-bar truss design; absolutely nonconvex spherical quadratic approximation.



(a) Time to convergence.

(b) Average time per subproblem evaluation.

Figure 6.5: Computation times.

Chapter 7

Closure and future research

This dissertation illustrated the potential of traditional sequential approximate optimization (SAO) methods applied to the alternative 'direct'—or 'simultaneous analysis and design' (SAND)—formulation of the structural topology optimization problem. In Chapter 2 it was demonstrated that, in classical minimum compliance design, SAO-SAND (SEASAND) methods agree with conventional SAO-NAND methods—*e.g.* the ever-popular optimality criteria (OC) approach, and the equivalent dual-SAO-NAND class of methods based on exponential intervening variables. In Chapter 3 it was demonstrated that convergence and termination can be enforced with either a restricted step method, or with a more traditional technique based on 'conservative' approximations. In Chapter 4 it is argued that design-set restriction should be imposed with point-wise bounds on the material distribution function—with so-called 'slope constraints'—while it is demonstrated that the multimodality of the problem, which is suspected of being more severe in the SAND setting, can be dealt with by a random multistart strategy. In Chapter 5 it is demonstrated that, in the SAND setting, standard gradient-based optimization methods can converge to the singular minima which characterize the local stress-constrained problem, and, because the sensitivity analyses associated with local state-based constraints reduce to simple—and sparse—partial derivatives, solutions to very large-scale problem instances could be presented in Chapter 6.

This dissertation touches on both the potential and the challenges of the envisaged class of SEASAND methods. From a mathematical programming point of view, future work could, for example, consider new and innovative subproblem formulations. This may include nonconvex formulations, second-order cone programs (SOCP), and novel second-order approximation strategies—based on, for example, random sampling, historical information, or off-diagonal second-order derivatives. Moreover, the implementation of sophisticated restoration procedures may decrease the frequency of unsuccessful (infeasible) terminations, and thereby, in turn, increase the utility of random multistart strategies—with a view to global optimization. The notion that global optimization can be achieved a random multistart strategiy in itself requires further investigation—both in terms of the initial (random) material distributions and the confidence-bound associated with the best known solution. These considerations pertaining to robust convergence and global optimization may be particularly important in view of more sophisticated and involved structural analysis—*e.g.* nonlinear—and multi-physics problem formulations. Alternative finite element formulations can also be investigated, and the

implementation of mesh-refinement procedures may reduce computation time even further.

From an engineering and manufacturing point of view, it should be investigated to what extent slope constraints—and design-set restriction in general—can be tailored to the specific requirements of a given manufacturing process—like for example the geometric requirements associated with casting. Moreover, the incorporation of transient phenomena and multi-physics formulations could be crucial to the realisation of (especially metal-based) additive manufacturing technologies. Finally, it should be determined to what extent standard structural analysis packages—*i.e.* 'simulation packages'—can be exploited in a SEASAND setting—this may, in turn, permit a fair comparison between NAND and SAND methods, in terms of practical implementation and computational effort.

References

[1] D. Munro. 2016. *A contemplation on topology optimization* [Online]. Available: http://www.digitaleng.news/de/a-contemplation-on-topology-optimization/. [2016, December 1].

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

[3] R.T. Haftka, Z. Gürdal, and M.P. Kamat. *Elements of structural optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.

[4] *Benefiting from the next production revolution*. [S.a.]. [Online]. Available: http://oecdinsights.org/2016/02/23/benefiting-from-the-next-production-revolution. [2016, August 20].

[5] T. E. Halterman. 2016. *Autodesk Within: Generative design software optimized for 3D printing* [Online]. Available: https://3dprint.com/80908/autodesk-within-software/. [2016, April 26].

[6] *Shape generator makes its debut in the latest Autodesk Inventor update*. [S.a.]. [Online]. Available: http://www.deskeng.com/virtual_desktop/?p=11056. [2015, October 29].

[7] *New Partnership Provides Advanced Virtual Testing Solutions*. [S.a.]. [Online]. http://www.prweb.com/releases/2016/01/prweb13157649.htm. [2016, April 23].

[8] *VR&D Announces GENESIS for ANSYS Mechanical users*. [S.a.]. [Online]. Available: http://www.deskeng.com/de/topology-optimization-issues. [2016, February 1].

[9] A. J. Lockwood. 2016. *Editor's Pick: solidThinking 2016* [Online]. Available: http://www.deskeng.com/de/editors-pick-solidthinking-2016. [2016, April 13].

[10] O. Sigmund. A 99 line topology optimization code written in Matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, 2001.

[11] K. Suresh. A 199-line Matlab code for Pareto-optimal tracing in topology optimization. *Structural and Multidisciplinary Optimization*, 42(5):665–679, 2010.

[12] 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.

[13] C. Talischi, G.H. Paulino, A. Pereira, and I.F.M. Menezes. Polytop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Structural and Multidisciplinary Optimization*, 45(3):329–357, 2012.

[14] K. Liu and A. Tovar. An efficient 3D topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 50(6):1175–1196, 2014.

[15] *Quick, no-cost topology optimization software released as cloud-based MCAD app*. [S.a.]. [Online]. Available: http://www.design-engineering.com/quick-no-cost-topology-optimization-software-released-as-cloud-based-mcad-app-137254/. [2015, November 5].

[16] *Wondeful widgets*. [S.a.]. [Online]. Available: http://www.economist.com/node/21662653. [2015, October 29].

[17] *Alloy Angels*. [S.a.]. [Online]. Available: http://www.economist.com/news/science-and-technology/21699432-3d-printing-produces-curious-lightweight-motorcycle-alloy-angels. [2016, June 10].

[18] *Precision casting centre and voxeljet win 'component of the year' award for topology optimised part 5X stiffer than before*. [S.a.]. [Online]. Available: http://www.3ders.org/articles/20160126-voxeljet-wins-component-of-the-year-award-for-topologically-optimised-part-5x-stiffer.html. [2016, January 27].

[19] K. Maxey. 2016. *Shape optimization leads to improved electric motor* [Online]. Available: http://www.engineering.com/PLMERP/ArticleID/11258/Shape-Optimization-Leads-to-Improved-Electric-Motor.aspx. [2016, January 10].

[20] B. Stackpole. 2014. *Ford invests heavily in lightweighting efforts* [Online]. Available: www.deskeng.com/de/ford-invests-heavily-lightweighting-efforts/. [2016, February 14].

[21] B. Stackpole. 2016. *In search of the perfect snowmobile design* [Online]. Available: http://www.deskeng.com/de/search-perfect-snowmobile-design/. [2016, April 26].

[22] *The new face of surgery*. [S.a.]. [Online]. Available: http://www.economist.com/node/16588705. [2015, October 29].

[23] *Ten questions for Alok Sutradhar*. [S.a.]. [Online]. Available: http://www.economist.com/blogs/babbage/2010/08/topological_optimisation. [2015, October 29].

[24] M. Petch. 2016. *$350k grant for 3d printing research developing topology optimization* [Online]. Available: https://3dprintingindustry.com/news/350k-grant-3d-printing-research-developing-topology-optimization-94088. [2016, August 20].

[25] B.M. Millsaps. 2016. *Swanson School of Engineering & Aerotech partner to refine metal additive manufacturing with fast computational modelling* [Online]. Available: https://3dprint.com/146259/swanson-aerotech-metal-am/. [2016, August 20].

[26] *Alcoa opens 3D printing metal powder plant*. [S.a.]. [Online]. Available: http://www.alcoa.com/global/en/news/news_detail.asp?pageID=20160706000340en&newsYear=2016. [2016, August 20].

[27] *Norsk Titanium to build world's first industrial-scale aerospace additive manufacturing plant in New York*. [S.a.]. [Online]. Available: http://www.businesswire.com/news/home/20160711005252/en/Norsk-Titanium-Build-World%E2%80%99s-Industrial-Scale-Aerospace-Additive. [2016, August 20].

[28] S. Wasserman. 2015. *3D Printing brings out the full potential of topology optimization* [Online]. Available: http://www.engineering.com/AdvancedManufacturing/ArticleID/11008/3D-Printing-Brings-Out-the-Full-Potential-of-Topology-Optimization.aspx. [2016, September 27].

[29] B. Jenkins. 2016. *System-level Optimization* [Online]. Available: http://www.deskeng.com/de/system-level-optimization. [2016, January 5].

[30] J. McNally. 2016. *Earth's most stunning natural fractal patterns* [Online]. Available: http://www.wired.com/2010/09/fractal-patterns-in-nature/. [2016, May 26].

[31] D. Deutsch. *The Beginning of Infinity: Explanations That Transform the World*. Penguin Group, London, England, 2011.

[32] D. Munro. Topology optimisation and simultaneous analysis and design: Material penalisation and local stress constraints. Master's Thesis , Department of Mechanical and Mechatronic Enigneering, Stellenbosch University, Stellenbosch, South Africa, 2014.

[33] D. Munro and A.A. Groenwold. On sequential approximate simultaneous analysis and design in classical topology optimization. *International Journal for Numerical Methods in Engineering*, 2016. Accepted and Published Online.

[34] C. Fleury and V. Braibant. Structural optimization: a new dual method using mixed variables. *International Journal for Numerical Methods in Engineering*, 23(3):409–428, 1986.

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

[36] 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.

[37] C. Fleury. Structural weight optimization by dual methods of convex programming. *International Journal for Numerical Methods in Engineering*, 14(12):1761–1783, 1979.

[38] J.E. Falk. Lagrange multipliers and nonlinear programming. *Journal of Mathematical Analysis and Applications*, 19(1):141–159, 1967.

[39] D.W. Wood and A.A. Groenwold. Non-convex dual forms based on exponential intervening variables, with application to weight minimization. *International Journal for Numerical Methods in Engineering*, 80(12):1544–1572, 2009.

[40] C. Fleury. Structural optimisation methods for large scale problems: Computational time issues. In *8th World Congress on Structural and Multidisciplinary Optimisation*, Lisbon, Portugal, June 2009. WCSMO.

[41] A.A. Groenwold and L.F.P. Etman. SAO*i*: an algorithm for very large scale optimal design. In *Proc. Ninth World Congress on Structural and Multidisciplinary Optimization*, Shizuoka, Japan, June 2011. Paper 035.

[42] D.W. Wood. *Dual sequential approximation methods in structural optimization*. PhD thesis, Stellenbosch University, 2012.

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

[44] S. Rojas-Labanda and M. Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization*, pages 1–21, 2015.

[45] C. Zillober. SCPIP — an efficient software tool for the solution of structural optimization problems. *Structural and Multidisciplinary Optimization*, 24(5):362–371, 2002.

[46] L.F.P. 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.

[47] M.P. Bendsøe. *Optimization of structural topology, shape, and material*. Springer, Berlin, Germany, 1995.

[48] A.A. Groenwold and L.F.P. Etman. On the equivalence of optimality criterion and sequential approximate optimization methods in the classical topology layout problem. *International Journal for Numerical Methods in Engineering*, 73(3):297–316, 2008.

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

[50] A. Evgrafov. State space Newtons method for topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 278:272–290, 2014.

[51] R.T. Haftka. Simultaneous analysis and design. *American Institute of Aeronautics and Astronautics Journal*, 23(7):1099–1103, 1985.

[52] M.P. Bendsøe, A. Ben-Tal, and R.T. Haftka. New displacement-based methods for optimal truss topology design. In *32nd Structures, Structural Dynamics, and Materials Conference*, Baltimore, MD, U.S.A., April 1991. American Institute of Aeronautics and Astronautics.

[53] S. Sankaranarayanan, R.T. Haftka, and R.K. Kapania. Truss topology optimization with simultaneous analysis and design. *American Institute of Aeronautics and Astronautics Journal*, 32(2):420–424, 1994.

[54] M.P. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1(4):193–202, 1989.

[55] G.I.N. Rozvany and M. Zhou. Applications of the COC algorithm in layout optimization. In *Engineering optimization in design processes*, pages 59–70. Springer, 1991.

[56] M. Stolpe. On some fundamental properties of structural topology optimization problems. *Structural and Multidisciplinary Optimization*, 41(5):661–670, 2010.

[57] 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(1):68–75, 1998.

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

[59] D. Munro and A.A. Groenwold. On design-set restriction in SAND topology optimization. *Structural and Multidisciplinary Optimization*, 2016. Submitted.

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

[61] H.P.J. Bolton, A.A. Groenwold, and J.A. Snyman. The application of a unified bayesian stopping criterion in competing parallel algorithms for global optimization. *Computers & Mathematics with Applications*, 48(3):549–560, 2004.

[62] G.M. Fadel, M.F. Riley, and J.M. Barthelemy. Two point exponential approximation method for structural optimization. *Structural Optimization*, 2(2):117–124, 1990.

[63] A.A. Groenwold, L.F.P. Etman, J.A. Snyman, and J.E. Rooda. Incomplete series expansion for function approximation. *Structural and Multidisciplinary Optimization*, 34(1):21–40, 2007.

[64] A.A. Groenwold, L.F.P. Etman, and D.W.Wood. Approximated approximations for SAO. *Structural and Multidisciplinary Optimization*, 41:39–56, 2010.

[65] A.A. Groenwold and L.F.P. Etman. A quadratic approximation for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 82(4):505–524, 2010.

[66] R. Fletcher, N.I.M. Gould, S. Leyffer, P.L. Toint, and A. Wächter. Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. *Society for Industrial and Applied Mathematics Journal on Optimization*, 13(3):635–659, 2002.

[67] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *Society for Industrial and Applied Mathematics Journal on Optimization*, 12(2):555–573, 2002.

[68] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1):193–228, 1998.

[69] J.V. Burke and S-P. Han. A robust sequential quadratic programming method. *Mathematical Programming*, 43(1-3):277–303, 1989.

[70] 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.

[71] A.A. Groenwold. Positive definite separable quadratic programs for non-convex problems. *Structural and Multidisciplinary Optimization*, 46(6):795–802, 2012.

[72] W. Murray. Newton-type methods. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[73] A. Evgrafov. On the reduced hessian of the compliance. *Structural and Multidisciplinary Optimization*, 50(6):1197–1199, 2014.

[74] A.A. Groenwold and L.F.P. Etman. On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations. *Structural and Multidisciplinary Optimization*, 42(2):165–178, 2010.

[75] D. Munro and A. A. Groenwold. On filtered "conservatism" in direct topology design. In *The Sixth International Conference on Structural Engineering, Mechanics and Computation*, September 2016. University of Cape Town.

[76] Gurobi Optimizer Reference Manual. [S.A.]. [Online]. Available: http://www.gurobi.com. [2016, February 24].

[77] University of Stellenbosch's Rhasatsha HPC. [S.a.]. [Online]. Available: http://www.sun.ac.za/hpc. [2016, February 24].

[78] S. Ulbrich. On the superlinear local convergence of a filter-sqp method. *Mathematical Programming*, 100(1):217–245, 2004.

[79] C. Zillober. A combined convex approximationinterior point approach for large scale nonlinear programming. *Optimization and Engineering*, 2(1):51–73, 2001.

[80] A.A. Groenwold, D.W. Wood, L.F. P. Etman, and S. Tosserams. Globally convergent optimization algorithm using conservative convex separable diagonal quadratic approximations. *American Institute of Aeronautics and Astronautics Journal*, 47(11):2649–2657, 2009.

[81] J.F.M. Barthelemy and R.T. Haftka. Approximation concepts for optimum structural design—a review. *Structural and Multidisciplinary Optimization*, 5(3):129–144, 1993.

[82] W. Achtziger, M. Bendsøe, A. Ben-Tal, and J. Zowe. Equivalent displacement based formulations for maximum strength truss topology design. *IMPACT of Computing in Science and Engineering*, 4(4):315–345, 1992.

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

[84] C. Zillober. Global convergence of a nonlinear programming method using convex approximations. *Numerical Algorithms*, 27(3):265–289, 2001.

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

[86] Y. Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151(1):249–281, 2015.

[87] R. Fletcher, S. Leyffer, P. Toint, et al. A brief history of filter methods. *Society for Industrial and Applied Mathematics Activity group on Optimization*, 18(ANL/MCS/JA-58300), 2007.

[88] D. Munro and A.A. Groenwold. Local stress-constrained and slope-constrained sand topology optimisation. *International Journal for Numerical Methods in Engineering*, 2016. Accepted and Published Online.

[89] R. Fletcher. Restricted step methods. *Practical Methods of Optimization, Second Edition*, pages 95–109, 2000.

[90] R.T. Haftka and M.P. Kamat. Simultaneous nonlinear structural analysis and design. *Computational Mechanics*, 4(6):409–416, 1989.

[91] Q. Wang and J.S. Arora. Optimization of large-scale truss structures using sparse SAND formulations. *International journal for numerical methods in engineering*, 69(2):390–407, 2007.

[92] 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.

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

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

[95] T. E. Bruns and D. A. Tortorelli. An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 57(10):1413–1430, 2003.

[96] T. E. Bruns. Zero density lower bounds in topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 196(13):566 – 578, 2006.

[97] J. Petersson. Some convergence results in perimeter-controlled topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 171(12):123 – 140, 1999.

[98] P. Duysinx. Layout optimization: A mathematical programming approach. Technical report, Université de Liège, Département d'aérospatiale et mécanique, Ingénierie des véhicules terrestres, 1997.

[99] O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007.

[100] O. Sigmund and K. Maute. Sensitivity filtering from a continuum mechanics perspective. *Structural and Multidisciplinary Optimization*, 46(4):471–475, 2012.

[101] M. Zhou, R. Fleury, Y. Shyy, H. Thomas, and J.M. Brennan. Progress in topology optimization with manufacturing constraints. In *Proceedings of the 9th AIAA MDO conference AIAA-2002-4901*, 2002.

[102] T. Borrvall. Topology optimization of elastic continua using restriction. *Archives of Computational Methods in Engineering*, 8(4):351–385, 2001.

[103] M. Stolpe and K. Svanberg. On the trajectories of penalization methods for topology optimization. *Structural and Multidisciplinary Optimization*, 21(2):128–139, 2001.

[104] P. Duysinx, L. Van Miegroet, E. Lemaire, O. Brüls, and M. Bruyneel. Topology and generalized shape optimization: Why stress constraints are so important? *International Journal for Simulation and Multidisciplinary Design Optimization*, 2(4):253–258, 2008.

[105] J.T. Pereira, E.A. Fancello, and C.S. Barcellos. Topology optimization of continuum structures with material failure constraints. *Structural and Multidisciplinary Optimization*, 26(1-2):50–66, 2004.

[106] M. Stolpe and K. Svanberg. Modelling topology optimization problems as linear mixed 01 programs. *International Journal for Numerical Methods in Engineering*, 57(5):723–739, 2003.

[107] M. Stolpe. Global optimization of minimum weight truss topology problems with stress, displacement, and local buckling constraints using branch-and-bound. *International Journal for Numerical Methods in Engineering*, 61(8):1270–1309, 2004.

[108] M. Stolpe and T. Stidsen. A hierarchical method for discrete structural topology design problems with local stress and displacement constraints. *International journal for numerical methods in engineering*, 69(5):1060–1084, 2007.

[109] P. Duysinx and M.P. Bendsøe. Topology optimization of continuum structures with local stress constraints. *International Journal for Numerical Methods in Engineering*, 43(8):1453–1478, 1998.

[110] G. Sved and Z. Ginos. Structural optimization under multiple loading. *International Journal of Mechanical Sciences*, 10(10):803 – 805, 1968.

[111] U. Kirsch. On singular topologies in optimum structural design. *Structural Optimization*, 2(3):133–142, 1990.

[112] G. Cheng and Z. Jiang. Study on topology optimization with stress constraints. *Engineering Optimization*, 20(2):129–148, 1992.

[113] G. Cheng and X. Guo. ε-relaxed approach in structural topology optimization. *Structural Optimization*, 13(4):258–266, 1997.

[114] G. Cheng and X. Guo. A note on a jellyfish-like feasible domain in structural topology optimization. *Engineering Optimization*, 31(1):1–24, 1998.

[115] M. Bruggi. On an alternative approach to stress constraints relaxation in topology optimization. *Structural and Multidisciplinary Optimization*, 36(2):125–141, 2008.

[116] M. Stolpe and K. Svanberg. A note on stress-constrained truss topology optimization. *Structural and Multidisciplinary Optimization*, 25(1):62–64, 2003.

[117] W. Achtziger and C. Kanzow. Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications. *Mathematical Programming*, 114(1):69–99, 2008.

[118] A.F. Izmailov and M.V. Solodov. Mathematical programs with vanishing constraints: optimality conditions, sensitivity, and a relaxation method. *Journal of Optimization Theory and Applications*, 142(3):501–532, 2009.

[119] W. Achtziger, C. Kanzow, and T. Hoheisel. On a relaxation method for mathematical programs with vanishing constraints. *GAMM-Mitteilungen*, 35(2):110–130, 2012.

[120] J. París, F. Navarrina, I. Colominas, and M. Casteleiro. Topology optimization of continuum structures with local and global stress constraints. *Structural and Multidisciplinary Optimization*, 39(4):419–437, 2009.

[121] J. París, F. Navarrina, I. Colominas, and M. Casteleiro. Block aggregation of stress constraints in topology optimization of structures. *Advances in Engineering Software*, 41(3):433–441, 2010.

[122] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41(4):605–620, 2010.

[123] M. Bruggi and P. Venini. A mixed FEM approach to stress-constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 73(12):1693–1714, 2008.

[124] J. Paris, I. Colominas, F. Navarrina, and M. Casteleiro. Parallel computing in topology optimization of structures with stress constraints. *Computers & Structures*, 125:62–73, 2013.

[125] H. Emmendoerfer and E. A. Fancello. A level set approach for topology optimization with local stress constraints. *International Journal for Numerical Methods in Engineering*, 99(2):129–156, 2014.

[126] A. Verbart, M. Langelaar, and F. van Keulen. Damage approach: A new method for topology optimization with local stress constraints. *Structural and Multidisciplinary Optimization*, 53(5):1081–1098, 2016.

[127] M. Stolpe and K. Svanberg. On the trajectories of the epsilon-relaxation approach for stress-constrained truss topology optimization. *Structural and Multidisciplinary Optimization*, 21(2):140–151, 2001.

[128] IBM ILOG CPLEX User Manual. [S.A.]. [Online]. Available: http://www.ibm.com. [2016, February 24].

[129] D. Munro and A.A. Groenwold. SEASAND: A direct approach to structural toplogy design. *Structural and Multidisciplinary Optimization*, 2016. Submitted.

[130] S. Rojas-Labanda and M. Stolpe. An efficient second-order SQP method for structural topology optimization. *Structural and Multidisciplinary Optimization*, pages 1–19, 2015.

[131] J.H. Rong, T.T. Xiao, L.H. Yu, X.P. Rong, and Y.J. Xie. Continuum structural topological optimizations with stress constraints based on an active constraint technique. *International Journal for Numerical Methods in Engineering*, 108(4):326–360, 2016.

[132] M. Megahed, H. Mindt, N. N'Dri, H. Duan, and O. Desmaison. Metal additive-manufacturing process and residual stress modeling. *Integrating Materials and Manufacturing Innovation*, 5(1):1–33, 2016.

[133] A. Verbart, M. Langelaar, and F. van Keulen. A unified aggregation and relaxation approach for stress-constrained topology optimization. *Structural and Multidisciplinary Optimization*, 2016.

[134] J.A. Snyman and A.M. Hay. The spherical quadratic steepest descent (SQSD) method for unconstrained minimization with no explicit line searches. *Computers & Mathematics with Applications*, 42(1):169 – 178, 2001.

[135] J.A. Snyman and A.M. Hay. The Dynamic-Q optimization method: An alternative to SQP? *Computers and Mathematics with Applications*, 44(12):1589 – 1598, 2002.

[136] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[137] S.S. Rao. *Engineering optimization: Theory and practice*. John Wiley & Sons, 2009.

[138] D.S. Bernstein. *Matrix mathematics: Theory, facts, and formulas*. Princeton University Press, 2009.