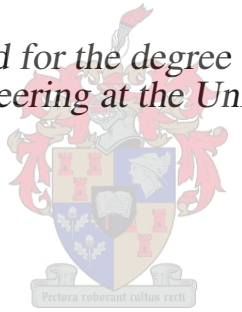
Dual Sequential Approximation Methods in Structural Optimisation

by

Derren Wesley Wood

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



Promotor:
Prof. Albert A. Groenwold

March 2012

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.

Signature: Date:฿

Abstract

This dissertation addresses a number of topics that arise from the use of a dual method of sequential approximate optimisation (SAO) to solve structural optimisation problems. Said approach is widely used because it allows relatively large problems to be solved efficiently by minimising the number of expensive structural analyses required. Some extensions to traditional implementations are suggested that can serve to increase the efficacy of such algorithms. The work presented herein is concerned primarily with three topics: the use of nonconvex functions in the definition of SAO subproblems, the global convergence of the method, and the application of the dual SAO approach to large-scale problems. Additionally, a chapter is presented that focuses on the interpretation of Sigmund's mesh independence sensitivity filter in topology optimisation.

It is standard practice to formulate the approximate subproblems as strictly convex, since strict convexity is a sufficient condition to ensure that the solution of the dual problem corresponds with the unique stationary point of the primal. The incorporation of nonconvex functions in the definition of the subproblems is rarely attempted. However, many problems exhibit nonconvex behaviour that is easily represented by simple nonconvex functions. It is demonstrated herein that, under certain conditions, such functions can be fruitfully incorporated into the definition of the approximate subproblems without destroying the correspondence or uniqueness of the primal and dual solutions.

Global convergence of dual SAO algorithms is examined within the context of the CCSA method, which relies on the use and manipulation of conservative convex and separable approximations. This method currently requires that a given problem and each of its subproblems be relaxed to ensure that the sequence of iterates that is produced remains feasible. A novel method, called the bounded dual, is presented as an alternative to relaxation. Infeasibility is catered for in the solution of the dual, and no relaxation-like modification is required. It is shown that when infeasibility is encountered, maximising the dual subproblem is equivalent to minimising a penalised linear combination of its constraint infeasibilities. Upon iteration, a restorative series of iterates is produced that gains feasibility, after which convergence to a feasible local minimum is assured.

Two instances of the dual SAO solution of large-scale problems are addressed herein. The first is a discrete problem regarding the selection of the point-wise optimal fibre orientation in the two-dimensional minimum compliance design for fibre-reinforced composite plates. It is solved by means of the discrete dual approach, and the formulation employed gives rise to a partially separable dual problem. The second instance involves the solution of planar material distribution problems subject to local stress constraints. These are solved in a continuous sense using a sparse solver. The complexity and dimensionality of the dual is controlled by employing a constraint selection strategy in tandem with a mechanism by which inconsequential elements of the Jacobian

of the active constraints are omitted. In this way, both the size of the dual and the amount of information that needs to be stored in order to define the dual are reduced.

Opsomming

Hierdie proefskrif spreek ’n aantal onderwerpe aan wat spruit uit die gebruik van ’n duale metode van sekwensiële benaderde optimering (SBO; sequential approximate optimisation (SAO)) om strukturele optimeringsprobleme op te los. Hierdie benadering word breedvoerig gebruik omdat dit die moontlikheid skep dat relatief groot probleme doeltreffend opgelos kan word deur die aantal duur strukturele analises wat vereis word, te minimeer. Sommige uitbreidings op tradisionele implementerings word voorgestel wat kan dien om die doeltreffendheid van sulke algoritmes te verhoog. Die werk wat hierin aangebied word, het hoofsaaklik betrekking op drie onderwerpe: die gebruik van nie-konvekse funksies in die definiëring van SBO-subprobleme, die globale konvergensie van die metode, en die toepassing van die duale SBO-benadering op grootskaalse probleme. Daarbenewens word ’n hoofstuk aangebied wat fokus op die interpretasie van Sigmund se maasonafhanklike sensitiwiteitsfilter (mesh independence sensitivity filter) in topologie-optimering.

Dit is standaard praktyk om die benaderde subprobleme as streng konveks te formuleer, aangesien streng konveksiteit ’n voldoende voorwaarde is om te verseker dat die oplossing van die duale probleem ooreenstem met die unieke stasionêre punt van die primaal. Die insluiting van nie-konvekse funksies in die definisie van die subprobleme word selde gepoog. Baie probleme toon egter nie-konvekse gedrag wat maklik deur eenvoudige nie-konvekse funksies voorgestel kan word. In hierdie werk word daar gedemonstreer dat sulke funksies onder sekere voorwaardes met vrug in die definisie van die benaderde subprobleme inkorporeer kan word sonder om die korrespondensie of uniekheid van die primale en duale oplossings te vernietig.

Globale konvergensie van duale SBO-algoritmes word ondersoek binne die konteks van die CCSA-metode, wat afhanklik is van die gebruik en manipulering van konserwatiewe konvekse en skeibare benaderings. Hierdie metode vereis tans dat ’n gegewe probleem en elk van sy subprobleme verslap word om te verseker dat die sekwensie van iterasies wat geproduseer word, toelaatbaar bly. ’n Nuwe metode, wat die begrensde duaal genoem word, word aangebied as ’n alternatief tot verslapping. Daar word vir ontoelaatbaarheid voorsiening gemaak in die oplossing van die duaal, en geen verslappings-tipe wysiging word benodig nie. Daar word gewys dat wanneer ontoelaatbaarheid teëngekom word, maksimering van die duaal-subprobleem ekwivalent is aan minimering van sy begrensingsontoelaatbaarhede (constraint infeasibilities). Met iterasie word ’n herstellende reeks iterasies geproduseer wat toelaatbaarheid bereik, waarna konvergensie tot ’n plaaslike KKT-punt verseker word.

Twee gevalle van die duale SBO-oplossing van grootskaalse probleme word hierin aangespreek. Die eerste geval is ’n diskrete probleem betreffende die seleksie van die puntsgewyse optimale veseloriëntasie in die tweedimensionele minimum meegeefbaarheidsontwerp vir veselversterkte saamgestelde plate. Dit word opgelos deur middel van die diskrete duale benadering, en die for-

mulering wat gebruik word, gee aanleiding tot ’n gedeeltelik skeibare duale probleem. Die tweede geval behels die oplossing van in-vlak materiaalverspredingsprobleme onderworpe aan plaaslike spanningsbegrensings. Hulle word in ’n kontinue sin opgelos met die gebruik van ’n yl oplosser. Die kompleksiteit en dimensionaliteit van die duaal word beheer deur gebruik te maak van ’n strategie om begrensings te selekteer tesame met ’n meganisme waardeur onbelangrike elemente van die Jacobiaan van die aktiewe begrensings uitgelaat word. Op hierdie wyse word beide die grootte van die duaal en die hoeveelheid inligting wat gestoor moet word om die duaal te definieer, verminder.

Acknowledgements

First and foremost, I would like to thank my wife, my partner on life's journey and my best friend. Without Gretel's influence I would very likely never have embarked on doctoral studies, and my growth as an individual over the last years has been in large measure due to her.

The one person to whom I owe the deepest debt of gratitude is Professor Albert Groenwold, who has been my mentor during these doctoral studies. He was also the supervisor of my final year project and my promoter for my Masters thesis. Ever the source of interesting conversation, academic guidance, financial support and, not least, friendship, Albert has been a formative influence in my life for over a decade and, certainly, the foremost influence in my academic career.

Special thanks are extended to Professor Anton Basson, who made it possible for me to focus on compiling the dissertation at a time when other responsibilities were keeping me from doing so. Furthermore, I am obliged to Marisa Honey for her thorough proofreading of the finished document. Financial assistance from the NRF is also gratefully acknowledged.

Finally, I would like to voice my profound appreciation of my mother, my father and Leo, my stepfather, who have always provided a loving support structure and the encouragement for me to pursue my goals. This gratitude is also extended to Gretel's parents, Tania and Justo, who are now an integral part of that supportive base, and to Koko and Rocco for their unconditional love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Structural optimisation is an area in which the physical design of a structure can be derived algorithmically in a computational, automated fashion, with minimal human creative input. Depending on the type of structural optimisation problem considered, this can mean that decisions about the size, shape, orientation and/or connectivity of structural elements – or more generally the physical distribution of material(s) within a given design domain – are determined as the result of a systematic optimisation procedure. Structural optimisation can be used as an important design tool because it has the potential to deliver structurally near-optimal initial designs for designers to embroider upon. In this way, the process of arriving at effective designs for complex problems can be formalised and made more efficient.

It is usually not possible to compute the optimal configuration for a structure directly from knowledge of its boundary conditions and the (guessed) initial state alone. The system responses are invariably dependent on changes in the system's state variables in a nonlinear fashion. Structural optima are arrived at iteratively through a controlled search, governed by one or other optimisation procedure. Each iteration entails a re-design and a subsequent re-analysis of the structure to determine the new structural responses. Hence, structural optimisation requires the coupling of an optimisation algorithm with a structural analysis package. The structural analysis is a computationally expensive procedure, entailing, for instance, a finite element analysis to determine both the system responses and the sensitivities of these responses to changes in the design variables. As the computational requirements grow superlinearly with the size or refinement of the analysis model, analysis of large-scale models can take a considerable amount of time. In addition, the optimisation component requires an iterative procedure in which possibly several hundred re-analyses may be required to locate a system optimum. Hence, the optimisation procedures that are favoured in the field of structural optimisation are those that limit or minimise the required number of re-analyses; otherwise the process of optimisation becomes unfeasibly time consuming or the structural model must remain inadequately unrefined.

For example, two widely studied structural optimisation problems – the minimum compliance and minimum weight material distribution problems – are inherently large in scale, there being at least one variable per finite element in the discretised form of each, and both may potentially have a large number of constraints. Morever, the two problems are either difficult, constrained integer programming problems or, in the more usual relaxed and penalised formulations, nonconvex, con-

strained and multimodal. They are thus challenging problems from the point of view of numerical optimisation, and it is important, if the field of structural optimisation is to find greater application in industry, to identify or develop algorithms that can solve these types of problems efficiently.

The size of the problems that may be solved is limited both by the computational storage requirements demanded by the analysis model, and also by the necessity to store whatever information is needed by the optimisation procedure. For large-scale problems, the latter can be substantial. Therefore, the optimisation procedures historically preferred in structural optimisation are those that in some way balance the conflicting imperatives of minimising the required number of structural analyses and, at the same time, minimising the computational storage requirements in order that larger structures may be addressed (or conversely that sufficiently refined structural analysis models may be used). Currently, the dominant methodology involves the use of sequential approximate optimisation (SAO). The idea underlying SAO is simply that it may be more efficient to solve a series of explicit approximations to a problem, rather than solving the problem itself directly, especially when each evaluation of the objective function and/or constraints in the problem requires that a structural analysis be carried out.

The optimisation approach that has been utilised in the work presented in this document for the solution of popular structural optimisation problems is that of sequential approximate optimisation using explicit separable approximations and employing a dual solver to solve each subproblem. The approach has its genesis principally in the work that Fleury [1] presented in the late 1970s, which ultimately led to the development of efficient methods of sequential convex programming (SCP) for structural optimisation, utilising strictly convex and separable subproblems. The dual solvers suggested by Fleury depend upon the key conceptualisation of a Lagrangian dual due to Falk [2], which was presented even earlier, in 1967.

In the work considered here, we build on Fleury's approach of using strictly convex and separable approximations in combination with using a dual solution strategy based on the Falk dual, though we don't necessarily employ the same approximation strategies to construct the subproblems. Several optimisation algorithms for structural optimisation are based on this framework, popular examples being the method of moving asymptotes (MMA) of Svanberg [3], and convex linearisation (CONLIN) of Fleury and Braibant [4, 5]. These methods have in common that the SAO subproblems generated during their application are explicitly formulated to be strictly convex, the reason being that strictly convex programming problems have unique solutions, and it has been proved that the dual method can be used to locate such minima. Be this as it may, problems in structural optimisation are often nonconvex. What is more, the problems themselves sometimes suggest simple nonconvex forms for the approximating functions, which more accurately track the local behaviour of the problem, and the question arises whether such approximations can be incorporated into the dual SAO approach without destroying the utility of the dual solution strategy.

Compliance minimisation with the inclusion of volumetric penalisation is such an example, in which the volumetric constraint gives rise to a nonconvex feasible region and is easily represented as a concave function. The standard weight minimisation problem is another example, for which the feasible region is generally nonconvex due to the inclusion of stress or displacement constraints. First-order reciprocal approximations of these constraints naturally acquire this nonconvexity. In both cases it is standard practice to construct strictly convex approximations of the nonconvex behaviour, and to use these in the definition of the subsequent subproblems. Here it is investigated

under what conditions the nonconvex forms may be used instead in the construction of the approximate subproblems, and the abovementioned problems are used as explicit examples in which the nonconvex behaviour is specifically retained.

No matter how the subproblems are defined, the point in the design space at which a particular subproblem is minimised becomes the point at which the following subproblem is defined, and thus also the initial point in the search for its minimum. In this way a sequence of iterates is produced, each member of which corresponds to the solution of a particular subproblem. This sequence may be made to converge to a local optimum of the problem by employing one or other method for encouraging global convergence within the optimisation algorithm.

Global convergence is a second aspect of the dual SAO approach that is investigated in this document. One method that can be used to drive global convergence is the use of conservative convex and separable approximations (CCSA) in the construction of successive subproblems, as suggested and developed by Svanberg [6]. Since, conventionally, convex and separable functions are used anyway, it is relatively straightforward to incorporate conservatism as an additional prerequisite in choosing the approximating functions during each iteration. However, conservatism requires that each iterate is feasible, and so it is necessary to solve a relaxed version of a given problem and its approximate subproblems. The term 'relaxation' here refers to a modification of the original problem that ensures feasibility; Svanberg has shown that, under certain conditions, the solutions to a relaxed problem correspond to the solutions to the original problem.

Relaxation unavoidably introduces additional complexity into the optimisation procedure. A novel alternative to relaxation is discussed in which it is argued that global convergence may instead be driven inherently by the solution of the dual subproblems when CCSA approximations are used. Infeasibility is catered for by maximising the dual subproblems subject to a sufficiently large upper bound restriction on the dual variables. For infeasible subproblems, the dual strategy acts as a linear penalty formulation that minimises a linear combination of the constraint violations, and this drives successive iterates towards the feasible region. Once feasibility is achieved, the CCSA approach itself ensures global convergence without requiring relaxation.

The dual method is recognised as being advantageous for use primarily for problems in which the number of constraints is less (and usually significantly less) than the number of design variables. The reason for this is that the dual is defined in the space of the Lagrange multipliers, there being one associated with each constraint. If there are fewer constraints than primal variables, then the dual problem has a lower dimensionality than the primal problem. It is, moreover, concave and only simply constrained, so maximising the dual is usually numerically easier than minimising the primal. However, if the number of constraints approaches the number of primal variables and if the problem has a large number of variables, the advantages of using the dual methodology are eroded.

A third focus of the work presented here is the application of the dual to the solution of problems that have both a large number of variables and a large number of constraints. For such large-scale problems, even though the dual itself is very large, it retains its advantageous concave and simply constrained structure. Two of the forthcoming chapters are devoted to the application of the dual SAO approach in circumstances such as this.

Outline

The body of this dissertation (namely Chapters 3 to 9) is essentially a reproduction of a series of self-contained papers intended for submission and peer review; some have indeed seen publication. They have been slightly modified here from their original forms so as to avoid excessive repetition of the underlying theory that links them, although some repetition unavoidably remains in order to preserve the stand-alone character of each chapter. Hence, each chapter constituting the body of the dissertation has its own abstract, introduction, discussion, presentation of results and conclusion, and each concerns itself in a detailed way with one of the themes delineated above (all but Chapter 3 that is, which explores a topic particular to topology optimisation). Being articles, each of the chapters has collaborators originally recorded as co-authors. Said collaboration is now noted in a short prologue at the beginning of each chapter that additionally provides the original paper's title, as well as its publication or submission details if applicable. The layout of this dissertation is as follows:

Chapter 2 serves to introduce some of the theory underlying the work presented in subsequent chapters. It gives a brief overview of SAO, duality and the material distribution method, which underlies the minimum weight and minimum compliance problems that are used as example problems in the remaining chapters.

In Chapter 3, sensitivity filtering in topology optimisation is discussed. Superficially, this topic is not directly connected with the application of the dual SAO method in structural optimisation. However, whenever the minimum compliance problem has been addressed in this work, this particular form of filter has been utilised in its solution (which is fairly standard practice). There is some debate in the topology optimisation community on how the use of the filter should be interpreted, because it effectively modifies the problem formulation. The use of dual SAO as a solution strategy actually motivates an interpretation of the filter, and this has led to the arguments presented in Chapter 3.

Chapter 4 describes the application of the dual method to a large-scale problem concerned with deducing the optimal fibre orientation at each point in a composite plate. The problem is formulated and solved as a discrete problem, through the application of Fleury's discrete dual method, whereas the problems considered in all the other chapters are solved in a continuous sense. Though the number of constraints is greater than the number of design variables, for the considered problem the dual gains a special separable structure, which enables it to be maximised relatively efficiently.

Chapters 5, 6 and 7 explore the use of nonconvex approximations in the dual SAO approach. These chapters all draw on observations presented in Chapter 2, which describe under what conditions the dual of nonconvex problems may be consistently defined. In Chapter 5, the inclusion of a power-law volumetric penalisation in the minimum compliance problem is described. When the concave constraint is retained in the definition of the subproblems, the dual subproblems must be derived from nonconvex primal subproblems, which is unusual. It is often assumed that strict convexity of the primal subproblems is a prerequisite for a consistent dual formulation, but this is not so.

In Chapter 5 it is argued that the type of nonconvex problems that arise as approximate subproblems in the consideration of volumetric penalisation are amenable to solution via the standard Falk dual. Furthermore, it is pointed out that incorporating the nonconvex behaviour of the problem into the construction of the subproblems can lead to a more efficient solution procedure, relative to that

which results from constructing strictly convex approximations to the nonconvex functions.

In Chapter 6, this line of reasoning is continued and the use of nonconvex approximations is discussed in the context of weight minimisation. Taking a cue from the development of CONLIN, which is a so-called 'method of mixed variables', other methods of mixed variables are derived that are based on the use of the separable exponential approximation, including its higher-order terms. The suggested methods incorporate nonconvexity and can be used as general methods of function approximation in a dual SAO approach. The weight minimisation problem is used as an example.

The conditions that allow for the use of nonconvex functions in the dual SAO approach originate from Falk's original definition of the dual problem. They do not explicitly require that the nonconvex problems can be transformed into convex ones. However, it is true that the nonconvex approximate subproblems discussed in Chapters 5 and 6 can all in fact be transformed into strictly convex forms, which motivates an investigation of whether the existence of a convex transform is related to the conditions expressed in Chapter 2. This relationship is delineated in Chapter 7, although the inquiry is confined to separable problems.

The theme of global convergence is taken up in Chapter 8. Chapter 8 introduces the possibility of omitting relaxation in a globally convergent dual SAO approach based on the use of the CCSA approximations. This is accomplished by simply introducing a sufficiently large upper bound on the dual variables, which is respected when the dual is maximised. A proof of global convergence for this scheme is proffered.

Applying the dual SAO approach to large-scale structural problems is a topic that is returned to in Chapter 9. Weight minimisation and minimum compliance problems are solved subject to the addition of local stress constraints. These problems have as many constraints as design variables and the work presented illustrates the utility of the dual approach even for problems such as these. Different convex approximation schemes are compared and various ideas for minimising the necessitated computational storage requirements are discussed, as is an alternative method of stress relaxation.

Finally, a summary of the conclusions drawn throughout the report is presented in Chapter 10, and some thoughts and recommendations for future work are expressed.

Chapter 2

Structural optimisation, SAO and duality

Optimisation problems in structural design are informally categorised as falling into one of three types, namely shape, sizing and topology problems. In a shape optimisation problem, a structure is defined by the spatial domain that it occupies, and the perimeter of the domain corresponds to the physical surface of the structure. The purpose of the optimisation is to search for the optimal structural shape, for a given problem formulation, by varying the domain boundaries that are parameterised by control variables in some way. Figure 2.1 presents a diagrammatic representation of a planar shape optimisation problem. In it, the design domain is defined by a number of control points joined by straight lines (although, more generally, some form of spline may be used). In this case, the vertical position of the control points may be adjusted by the optimisation algorithm. During a numerical analysis of the design, which is normally accomplished using the finite element method, the domain is discretised. Since the domain itself is varied during shape optimisation, the implementational problems that must be overcome in shape optimisation are typically associated with mesh distortion and the remeshing of the structure.

In sizing optimisation, the design variables are physical properties of pre-existing design elements. As such, the procedure requires that an initial ground structure be defined, its elements being subsequently modified by the optimisation algorithm. An example of this is optimal truss design (depicted in Figure 2.2), in which the configuration of the truss elements is defined a priori and remains unchanged over the course of the optimisation. The positions of the supports, truss nodes and applied loads are all pre-defined, and together with the truss connectivity define the ground structure. The physical cross-sectional dimensions of the individual truss elements are frequently the design variables in such a problem. Sizing design is also applied to problems in which the design elements are not necessarily physically discrete. In two-dimensional continuum structures, for instance, the thickness of the structure may be considered as spatially variable. However, in this type of analysis the design domain is two-dimensional, and the thickness enters the analysis only as a set of parameters in the constitutive description of the structure. Varying these parameters does not change the domain in which the structure is defined or its connectivity (modelled by the connectivity of the finite elements in the FEM mesh).

Topology optimisation, on the other hand, is concerned with the geometric features of the design domain and with how these affect the structural responses. The domain itself is again defined a priori. In topology optimisation of truss structures, the connectivity of the truss elements can

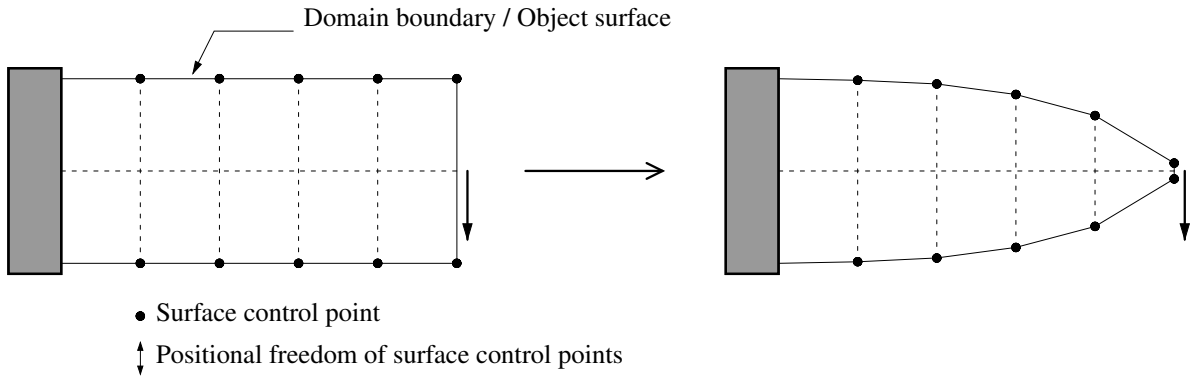


Figure 2.1: Variation of the boundary of the design domain, and thus the structural shape, in a planar shape optimisation problem.

be modified between a defined set of joints and supports, which together with the applied forces define the ground structure. The set ground structure limits the possible truss configurations that can be accommodated, and the purpose of the optimisation procedure would be to identify the optimal truss connectivity (that is, to identify which joints and/or supports are connected by truss elements). This is a discrete problem if the cross-sections of the elements are not variable. In the consideration of continuum structures, the distribution of material within the design domain is variable. The goal of the optimisation is to decide on the physical placement and nature of features such as holes in the design domain, or even of differing materials. This type of problem still requires the definition of an unchanging ground structure - the domain to be considered along with the supports and forces. In the strict sense, the topology problem is combinatorial, which is to say that, at a given point (or connection) in the design domain, the structure should be in one of only a few possible states.

However, the lines differentiating the three traditional branches of structural optimisation are blurred. A truss sizing procedure in which the dimensions of the truss elements can be reduced to zero may equally well be termed a topology problem, because the connectivity of the domain is modified thereby. In the same vein, a topology procedure that generates a solid-void design of a structure within a given domain may just as well be called generalised shape optimisation, and has been [7], since optimal structural configurations are generated with minimal restrictions on the types of shapes produced.

2.1 The material distribution method

According to Bendsøe and Sigmund [8], the prevalent approach currently used in determining optimum lay-outs for continuum structures is the material distribution method. Whereas the above discussion divides the features of structural optimisation problems into three perhaps overly narrow and artificially segregated classes, the ‘lay-out’ of a structure is described as being a more general concept that combines features of all three. As such, the material distribution method is described as being cabable of addressing all three aspects of structural optimisation simultaneously[1].

[1]It should be noted that the term ‘lay-out’ is not necessarily used the same way in [7] and in [8].

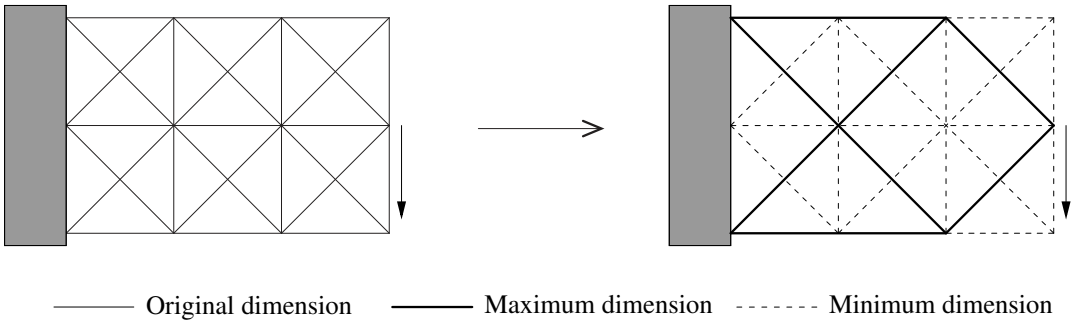


Figure 2.2: An illustration of the type of modification a truss structure may undergo during sizing optimisation.

Given a particular structural objective f_0, as well as j constraints f_j on the design, the material distribution method is aimed at identifying the optimal distribution of material $x(\boldsymbol{r})$ within a known, pre-defined design domain Ω, where the structural supports and applied loads are also defined[2]. Hence, the objective function has to be phrased as a function of the material distribution throughout the design domain. In the current document, two popular material distribution problems are considered, namely the minimum weight problem and the (classical) minimum compliance problem, both of which have the following general form, in which the field $x(\boldsymbol{r})$ denotes the presence or absence of material at a point $\boldsymbol{r}$ in the design space:

$$\begin{array}{rll} \min\limits_{x} & f_0(x) & \\ \text{subject to} & f_j(x) \leq 0 & j = 1, 2, \cdots, m, \\ \text{and with} & x(\boldsymbol{r}) \in [0,1] & \forall \quad \boldsymbol{r} \in \Omega\,. \end{array} \tag{2.1}$$

These problems are not only used as challenging test problems for the optimisation procedures employed, they have also motivated some of the ideas that have been integrated into the optimiser and that are the focus of this document. It should be noted that the label 'topology optimisation' is commonly used to refer to the optimisation of general material distribution problems, using the material distribution method, and sometimes specifically to the minimum compliance problem. In the remainder of this document, the former usage is used regularly, and I have endeavoured to expunge occurrences of the latter.

For our purposes, namely to discuss the efficient optimisation procedure we use to solve the problems, it is sufficient to depart from statements of the problems discretised in terms of the finite element method. However, since some of the complications involved in topology design are inherent in the underlying continuum problem, it is instructive first to consider an example of the continuum description. The 'classical' minimum compliance problem is used as an example, and is presented as described by Bendsøe and Sigmund [8].

[2]The script $\boldsymbol{r}$ is used to denote spatial position, since the more usual script $\boldsymbol{x}$ is used in this document to denote the vector of variables in an optimisation problem. For the problems that are considered here, the design variables are not spatial coordinates. Instead, $\boldsymbol{x}$ denotes the scaling of the material properties associated with elements in a finite element mesh. The normal-type x is here used to represent the scalar material distribution function, whose discretised form is denoted by the bold-type $\boldsymbol{x}$.

2.1.1 An example of a continuum formulation (compliance)

In the minimum compliance topology optimisation problem, the optimal spatial material distribution within the design domain is sought that minimises the structural compliance subject to an explicit constraint on the allowable material distribution. The variational form for minimising compliance is given in [8] as

$$\begin{aligned} \min_{u \in U, C} \quad & l(\boldsymbol{u}) \\ \text{subject to} \quad & a_C(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}) \qquad \forall \quad \boldsymbol{v} \in U \\ \text{and with} \quad & C \in C_{ad}\,. \end{aligned} \tag{2.2}$$

The compliance $l(\boldsymbol{u})$ is given as

$$l(\boldsymbol{u}) = \int_{\Omega} \boldsymbol{f}\boldsymbol{u}\,\mathrm{d}\Omega + \int_{\Gamma_T} \boldsymbol{t}\boldsymbol{u}\,\mathrm{d}s\,,$$

in which $\boldsymbol{f}$ represents the body forces and $\boldsymbol{t}$ denotes the tractions applied to portions of the boundary Γ_T of the design domain Ω. The equilibrium displacement field $\boldsymbol{u}$ satisfies the equilibrium equations, in which

$$a_C(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega} C_{ijkl}(\boldsymbol{r})\,\varepsilon_{ij}(\boldsymbol{u})\,\varepsilon_{kl}(\boldsymbol{v})\,\mathrm{d}\Omega$$

is the internal virtual work for an arbitrary virtual displacement $\boldsymbol{v}$ (provided $\boldsymbol{v}$ is a member of the set of kinematically allowable displacements U). Additionally, ε denotes the linearised strain field

$$\varepsilon_{ij}(\boldsymbol{u}) = \frac{1}{2}\left(\frac{\partial \boldsymbol{u}_i}{\partial r_j} + \frac{\partial \boldsymbol{u}_j}{\partial r_i}\right).$$

The dependence of the structural compliance on the material distribution enters the problem via the constitutive tensor C_{ijkl}, which is a function of the spatial position $\boldsymbol{r}$. At any point in the domain, the possible material properties are limited by the admissible set C_{ad}, to which $C_{ijkl}(\boldsymbol{r})$ must belong. The examples of the minimum compliance problem presented in this document are all formulated in terms of isotropic material descriptions. The desired optimal topologies are solid-void designs, meaning that material should ideally be either present or absent at any given point in the domain, with no other possible states besides the binary $[0, 1]$. The binary material distribution function can be denoted

$$x(\boldsymbol{r}) \in [0, 1] \quad \forall\, \boldsymbol{r} \in \Omega\,,$$

and the material compliance tensor, in turn, can be viewed as a function of x. If material is present at some point in the domain, it has the compliance tensor of a solid isotropic material $C(x(\boldsymbol{r})) = C(1) = C_0$ at that point. On the other hand, if there is no material present at a point in the domain, the material description for that point is conceptually $C(x(\boldsymbol{r})) = C(0) = 0$ (the computational solution of the problem makes it difficult to meet this stipulation, but it can be approximated closely).

As an aside, it should be noted that in most numerical solution procedures for material distribution problems the binary discreteness requirements on x are relaxed, so that $x(\boldsymbol{r})$ may assume any real value between 0 and 1. When this is done in the context of isotropic problems, the material

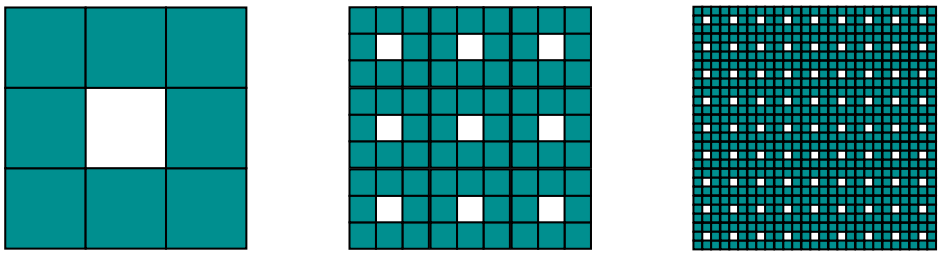


Figure 2.3: Qualitatively different structures produced by decreasing the length scale associated with the main structural feature (the holes). Each structure, however, has the same volume. Given a set of boundary conditions and loads applied to the unchanging structural domain, the structural compliance generally improves as the scale of the perforations decreases.

properties at a point in the design domain scale in a continuous way with x. In particular, when the domain is discretised by means of the finite element method, both the volume and mass of material within element i scale with x_i, the material occupancy of element i. Perhaps for this reason x is almost universally referred to as 'density', and the same terminology has been adopted in this document. However, the reader should bear in mind that x is not related to the physical mass density[3] ρ, except insofar as it (in effect) scales ρ, as it does the other material properties.

In the classical isotropic minimum compliance problem, a limit is traditionally placed on the total volume of solid material in the domain by introducing the following single constraint, in which $\bar{\nu}$ is the stipulated maximum allowable volume:

$$\int_{\Omega} x\left(\boldsymbol{r}\right) \mathrm{d}\Omega \;\leq\; \bar{\nu}. \tag{2.3}$$

The continuum problem apparently lacks solutions. The reason for this complication is frequently explained by at first considering a domain with a given distribution of solid material and holes, such as is illustrated in Figure 2.3. It is then noted that, if the holes are made smaller and more numerous so that the total volume of solid or void material within the domain does not change relative to the original structure, the resulting material distribution tends to improve in terms of its structural compliance. This process of successive refinement can be continued *ad infinitum*, producing an ever more perforated material microstructure.

The non-existence problem can be answered by the use of the homogenisation approach to topology design, in which material that possesses a microstructure can be introduced into the continuum formulation. One type of microstructure that is often used is designed from a composite combination of only the original isotropic material and void, in a way that is parameterisable by control variables. It is spatially periodic and its aggregate material properties can be calculated as a function of said control variables. This approach also provides for a physical interpretation of non-binary values of $x\left(\boldsymbol{r}\right)$ in the design domain, since the ratio of solid to void material is variable in the microstructure. Hence, both the density[4] of the material at a point in the domain and its other material properties are dependent on the microstructure at that point, and the parameters that

[3] Unless the homogenisation approach to topology optimisation is used. None of the work presented in this dissertation utilises homogenisation

[4] The term 'density' can here mean either the macroscopic mass density ρ or the material occupancy $x\left(\boldsymbol{r}\right)$, since in the homogenisation approach the two concepts are closely linked.

define it become the variables in the optimisation process. Using the homogenisation approach, other types of microstructure are also possible[5].

The materials with microstructure introduced in the homogenisation approach are anisotropic, so the approach allows for the introduction of composite material characteristics into the topology problem. Furthermore, the approach has spurred theoretical investigations in materials science and the design of material microstructures. However, the topology examples considered in this document are concerned with the more traditional topology problem in which the optimal distribution of isotropic solid material and void is sought. In this context, the problem of non-existence of solutions is seemingly related to the fact that the continuum structure has no minimum-length scale - i.e. there is no lower bound on the characteristic size of structural features.

Whereas the homogenisation approach relies on an extension of the design space to address the problem (allowing anisotropic materials), other approaches are availabe that involve a restriction of the design space instead, and these can be employed in the design of isotropic structures. 'Restriction' involves the addition of other constraints to the formulation that in one way or another introduce a finite limit on the minimim length scale for the structure, which in turn ensures that the restricted version of the problem has solutions.

It should be noted that when the topology problem is discretised to facilitate numerical analysis, usually with the finite element method, a minimum-length scale is automatically introduced into the domain in the form of the discretising mesh. Therefore, in the discretised problem, the existence of solutions is not strictly an issue, since the mesh can only represent a finite number of different $[0, 1]$ designs. However, the problem manifests itself in the tendency for different mesh discretisations to produce qualitatively different optimal topologies for the same problem. Increasing the mesh discretisation reduces the minimum-length scale and allows finer grained alternation of solid-void regions, thinner structural members and more intricate designs. Figure 2.4 presents an example of such behaviour[6].

Since optimal topologies should be useful in guiding the design of physical, manufacturable structures, this mesh dependence is considered unsatisfactory. By refining the mesh in an analysis, one would ideally like to arrive at a finer grained model of the same structure, rather than a different structure entirely. Moreover, from the point of view of the potential manufacturability of the designs, it would be useful to have some control over the minimum size of structural features. In the discrete setting, the restriction methods can provide the mechanism for achieving mesh independence and feature size control.

Some popular examples of restriction methods are perimeter control, local gradient control, density filtering and sensitivity filtering. For the continuum compliance problem, the first three methods have been proved to resolve the existence problem. Interested readers may refer to [10] for an overview of these methods, as well as their origination. Briefly, perimeter control places an upper bound on the perimeter of the design, which is, loosely, "the sum of the circumferences of all holes and outer boundaries," [10]. In this way a single extra global constraint is implemented. Local gradient control, on the other hand, places point-wise constraints on the magnitudes of the

[5]Refer to [7] for a brief overview and contextualisation, and [8] for a detailed discussion of the homogenisation approach.

[6]The results depicted were generated using Sigmund's freely availabe 99-line Matlab topology code [9], with a filter radius of 1.5 elements for each mesh discretisation.

(a) Mesh discretisation: 60×10

(b) Mesh discretisation: 120×20

(c) Mesh discretisation: 240×40

Figure 2.4: The practical result of the non-existence problem: the dependence of the solution on mesh discretisation (minimum compliance for the MBB beam).

derivatives of the material distribution function,

$$\check{c} \leq \frac{\partial x\left(\boldsymbol{r}\right)}{\partial r_i} \leq \hat{c}\,,$$

where $\check{c}$ and $\hat{c}$ are lower and upper bounds respectively. In numerical implementations, local gradient control requires the introduction of two additional constraints per element, makings its performance computationally expensive, especially for large problems (refined meshes).

Neither density filtering nor sensitivity filtering necessitate the inclusion of extra constraints. They are based instead on methods borrowed from image processing. The basic idea, according to Bourdin [11], is "to replace a (possibly) non-regular function by its regularisation obtained by the convolution with a smooth function." In density filtering, the entity that is filtered is the material distribution function (the density field). A new density field $x'\left(\boldsymbol{r}\right)$ is defined in which the material occupancy at each point is derived as a kind of 'weighted average' of the original field x, accomplished by means of the convolution operator

$$x'\left(\boldsymbol{r}\right) = F\left(\boldsymbol{r}\right) * x\left(\boldsymbol{r}\right) = \int_{\mathbb{R}^d} F\left(\boldsymbol{r} - \boldsymbol{r}'\right) x\left(\boldsymbol{r}'\right) \mathrm{d}\boldsymbol{r}',$$

where F is a smooth differentiable 'filter' function defined over $\mathbb{R}^d$, the physical dimension d being either 2 or 3 (Bourdin considers planar problems specifically). The choice of F differentiates one density filter from another. The form of F is always chosen so that it has its maximim value at $\boldsymbol{r}' = \boldsymbol{r}$, and then decays monotonically as $\boldsymbol{r}'$ diverges from $\boldsymbol{r}$. The normal distribution function is one such example. Bourdin's theoretical analysis has the convolution operator acting over all $\mathbb{R}^2$, which in turn requires that the density field be defined on $\mathbb{R}^2$, outside of Ω as well. In numerical

implementations a consistent method should be followed to ensure that the filter operation does not produce spurious results due simply to the presence of the boundaries on the design domain. A few suggestions are given in [11].

In sensitivity filtering it is the derivatives of the objective function, with respect to changes in the density field $x(\boldsymbol{r})$, that are filtered. This is reputedly the most popular restriction method in use, being very easy to implement. Although density filtering is similarly straightforward to implement, sensitivity filtering is apparently preferred by many in the optimisation community because it does not directly modify the designs themselves. It should be noted, however, that there is no proof as yet that the use of sensitivity filtering corresponds to a continuum compliance problem for which solutions exist.

The first three restriction methods mentioned above are all practically applied as operators or constraints in the numerical solution of the discretised version of the compliance problem. However, it is recognised that each of these methods defines a corresponding restricted continuum problem. Unlike the original continuum compliance problem (2.2), it has been shown that these restricted problems possess solutions [11, 12, 13]. Hence, far from being interpreted as mere operators, the methods are part of the definition of the problems. As such, it is no longer (2.2) that is solved, but rather a related problem defined by the incorporation of the given restriction method in the continuum setting. It is these related problems that are discretised by means of the finite element method, and it is necessary that the solutions for the discrete compliance problems produced thereby converge to the solutions for the associated restricted continuum problems in the limit of mesh refinement.

This is not the case for sensitivity filtering. No existence proof has been produced for this type of filter, so is not clear as yet whether a separate restricted continuum problem exists that is defined by the incorporation of sensitivity filtering. Also, assuming that one does exist, it is not known whether this problem possesses solutions to which the discrete solutions should converge. Consequently, the filter is generally seen as a heuristic that can be used to develop pleasing designs, but there is doubt as to how these solutions should be interpreted. Nevertheless, exhaustive experience with its implementation by those in the topology optimisation community have shown the filters in this class to be successful in generating mesh-independent designs. Due to its popularity and ease of use, we have used sensitivity filtering for all the examples presented in this dissertation (wherever filtering was necessary), utilising Sigmund's mesh independency filter [14, 15], the most popular form of sensitivity filtering. A more thorough discussion of this filter is presented in Chapter 3.

2.1.2 The discretised minimum compliance problem

The continuum topology problem (2.2) can be disctretised using the finite element method. In the discretised model, the elasticity (stiffness) matrix for element i is given as

$$\boldsymbol{C}_i(x_i) = x_i \, \boldsymbol{C}_0 \, . \tag{2.4}$$

Here, $x_i \in [0, 1]$ is an element of the binary-valued discretised density field $\boldsymbol{x}$, $\boldsymbol{C}_0$ is the plane stress elasticity matrix of the solid isotropic material, and $\boldsymbol{C}_i(x_i)$ is the elasticity matrix for element i. Subscript i indicates elemental quantities and operators and there are n finite elements in the mesh.

The principle of stationary potential energy may be used to demonstrate that the finite element stiffness matrices are expressed as

$$\boldsymbol{K}_i = \int_{\nu_i} \boldsymbol{B}_i^T \boldsymbol{C}_i(x_i) \boldsymbol{B}_i \, d\nu_i \,,$$

where the $\boldsymbol{B}_i$ represents the elemental strain-displacement operator and ν_i is the volume of a single element (we assume a regular mesh). If we denote $\boldsymbol{w}$ as the vector of applied nodal loads and $\boldsymbol{q}$ as the vector of nodal displacements, the compliance of the structure is obtained as

$$f_0(\boldsymbol{x}) = \boldsymbol{q}^T \boldsymbol{w} = \boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} = \sum_{i=1}^{n} x_i \, \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \,. \tag{2.5}$$

Furthermore, the volume constraint (2.3) can be expressed as

$$f_1(\boldsymbol{x}) = \frac{1}{\nu_0} \sum_{i=1}^{n} \nu_i x_i - \bar{\nu} \leq 0 \,, \tag{2.6}$$

if ν_0 is understood to be the volume of the design domain. Hence, the classical compliance problem with its single volume constraint, and in which it is assumed that the loads $\boldsymbol{w}$ are design independent, is expressed in a general way as an integer programming problem as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_1(\boldsymbol{x}) \leq 0, \\ & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{w}, \\ & x_i \in [0,1] \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{2.7}$$

Note, however, that a non-zero lower bound $\check{x}$ on the x_i is actually required to prevent complications arising from numerical ill-conditioning. The discrete problem has solutions by virtue of its discretisation, but to prevent mesh dependence a restriction method needs to be incorporated. However, confining our attention to (2.7) we note that this discrete programming problem is NP-complete and is very difficult to solve as a discrete problem, particularly since practical examples have high dimensionality. Thus, it is often replaced by a relaxed continuous problem in which the elemental densities are allowed to take on intermediate values

$$0 < \check{x} \leq x_i \leq \hat{x} \,,$$

in which $\hat{x}$ represents the allowable upper bound on the x_i, namely $x_i = 1$. The relaxed problem is amenable to solution using standard optimisation strategies for continuous nonlinear programming[7]. This relaxed continuous modification of (2.7) is obviously no longer representative of the original solid-void compliance problem (2.2). If the design domain is planar, it instead corresponds to a continuum formulation known as the variable thickness sheet problem, in which the field $x\,(\boldsymbol{r})$

[7] It is of course possible to attempt to solve the original discrete problem directly, using standard methods of integer programming, but such methods are not very efficient for problems of this size. However, see Fleury [16], Beckers [17] and Chapter 4 for a way of tackling the discrete problem that is based on the dual method and avoids using integer programming.

represents the (normalised) real-valued point-wise thickness of a planar structure. As it happens, this problem has a unique solution. The optimal material distribution is characterised by much grey material of intermediate thickness between 0 and 1.

In the solution strategy for (2.2), the relaxation is really only employed as a facility to enable the use of methods of continuous programming. A method is therefore employed to encourage the generation of $[0, 1]$ solutions for the relaxed continuous problem, so that the solution set of this now updated problem approximates the solution set of the original continuum compliance problem (or actually whatever restricted version thereof is considered). The most popular method for doing so is the so-called 'simple isotropic material with penalisation' approach, or SIMP, suggested independently by Bendsøe [18] and Rozvany and Zhou [19], which imposes a penalisation on intermediate densities by replacing the elemental material description (2.4) with

$$\boldsymbol{C}_i(x_i) = x_i^p \, \boldsymbol{C}_0 \,, \quad p > 1. \tag{2.8}$$

The penalisation does not affect the stiffness of elements that have densities of 0 or 1, but the stiffness of an element with intermediate density is rendered disproportionately low (i.e. less than a linearly scaled stiffness). Such an element is thus described as "uneconomical" in the classical compliance problem [10]. That the solutions to the SIMP-penalised relaxed continuous problem converge to the solutions to the original restricted continuum problem as the penalisation is increased has been shown by Petersson for the perimeter constraint restriction [20].

The form of the minimum compliance problem considered in this dissertation

We are now finally in a position to state the form of the minimum compliance problem that is frequently used as an example problem in testing some of the methods devised for sequential approximate optimisation described in the forthcoming chapters. The relaxed continuous form of the minimum compliance problem that is most amenable to numerical solution is

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_1(\boldsymbol{x}) \leq 0, \\ & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{w}, \\ & 0 < \check{x} \leq x_i \leq \hat{x} \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{2.9}$$

The discreteness requirements present in (2.7) are relaxed, and it is now implicitly assumed that problem (2.9) is combined with some (heuristic) method to arrive at an (approximate) discrete solution. We have invariably used the SIMP penalisation strategy, or a derivative thereof, to try to encourage convergence towards solid-void solutions. Therefore, given (2.5) and (2.8), the penalised objective function $f_0(\boldsymbol{x})$ in (2.9) becomes

$$f_0(\boldsymbol{x}) = \boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} = \sum_{i=1}^{n} x_i^p \, \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \,, \tag{2.10}$$

the subscript i denoting elemental quantities. If the applied loads $\boldsymbol{w}$ are taken to be independent of the design $\boldsymbol{x}$, then with minimal manipulation the gradients of f_0 can be shown to be

$$\frac{\partial f_0}{\partial x_i} = -p x_i^{p-1} \, \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \,. \tag{2.11}$$

Hence, the sensitivities of the compliance objective may be evaluated directly using information that is already available from the finite element solution for the structural displacements (and which is necessary for evaluating the objective function anyway). This is advantageous, since the gradients are typically necessary for the construction of the approximate subproblems in sequential approximate optimisation schemes – certainly for the algorithms highlighted in this document – and little additional work is required to derive them for the objective function or for the volume constraint in the compliance problem[8].

It is obviously possible to have multiple constraints f_j in (2.9), most commonly for the purposes of restricting the design space, for instance, or for representing allowable limits on stresses and/or displacements, or for incorporating manufacturing considerations. However, the compliance problem is frequently solved with only a single constraint, given by (2.6), that limits the maximum allowable volume of the structure. In this case it is common to use a filter as a restriction method. In the compliance problems that are discussed in the forthcomming chapters (with the exception of Chapter 9) we use Sigmund's mesh independence filter exclusively. In Chapter 9 the compliance problems are solved without using a restriction method.

2.1.3 The minimum weight problem

The second important structural optimisation problem considered here is the weight minimisation problem. As with the compliance problem, the only form of the problem considered is that in which the design domain is planar and continuous. The minimum weight problem is also phrased in terms of the material distribution $x(\boldsymbol{r})$ as in (2.1), the objective function being the weight[9] of the structure, given by

$$f_0(x) = \int_\Omega \rho(\boldsymbol{r})\, x(\boldsymbol{r})\, d\Omega\,.$$

The mass density $\rho(\boldsymbol{r})$, as with the other material properties, can conceptually vary as a function of position, but we confine our attention to problems in which the distribution of a single material with a uniform mass density is optimised.

Conventionally, the minimum weight topology is sought, subject to constraints on the allowable displacements and/or stresses within the structure. In the case of displacements, it is frequently the case that only a single constraint is considered (that limits the displacement of the point at which a load is applied, for example). Stress constraints, on the other hand, are by their nature local, point-wise restrictions. So, for example, a limit is placed on the maximum value that the von Mises stress, or another stress-related failure measure, can attain anywhere in the structure. For planar structures, the von Mises stress is defined as

$$\sqrt{\sigma_x^2 - \sigma_x\sigma_y + \sigma_y^2 + 3\tau_{xy}^2} \leq \sigma_{max}. \tag{2.12}$$

In keeping with (2.1) it is required that solid-void material distributions are identified as solutions, but a relaxed continuous form of the discretised problem is again considered, so that methods of continuous programming can be utilised in the optimisation. Given this relaxation, the discretised

[8] Stress and displacement constraints, on the other hand, require a bit more work (as will be seen in Chapter 9).

[9] Actually, the mass.

objective function is clearly linear in the design variables x_i, whereas it turns out that both nodal displacement constraints and elemental stress constraints are reciprocal in the x_i (as will be discussed in Section 2.2.2). Hence, the general (relaxed) form of the weight minimisation problem considered herein is given by

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) = \sum_{i=1}^{n} \rho_i \nu_i x_i \\ \text{subject to} \quad & f_j(\boldsymbol{x}) = c_{0j} + \sum_{i=1}^{n} \frac{c_{ij}}{x_i} \leq 0 \qquad j = 1, 2, \cdots, m, \\ & 0 < \check{x} \leq x_i \leq \hat{x} \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{2.13}$$

in which the ν_i are the elemental volumes. The SIMP method is again employed to drive the solutions to solid-void designs, so the material description is given by (2.8) in the FEM. In this case, the penalisation does not affect the objective function in the optimisation problem. Instead, the impetus for obtaining solid-void designs is provided by the way that penalisation affects constraint satisfaction. For instance, consider the elemental stress vector for an element i, which scales according to

$$\sigma_i = x_i \boldsymbol{C}_i \epsilon_i$$

in the relaxed but unpenalised formulation, and according to

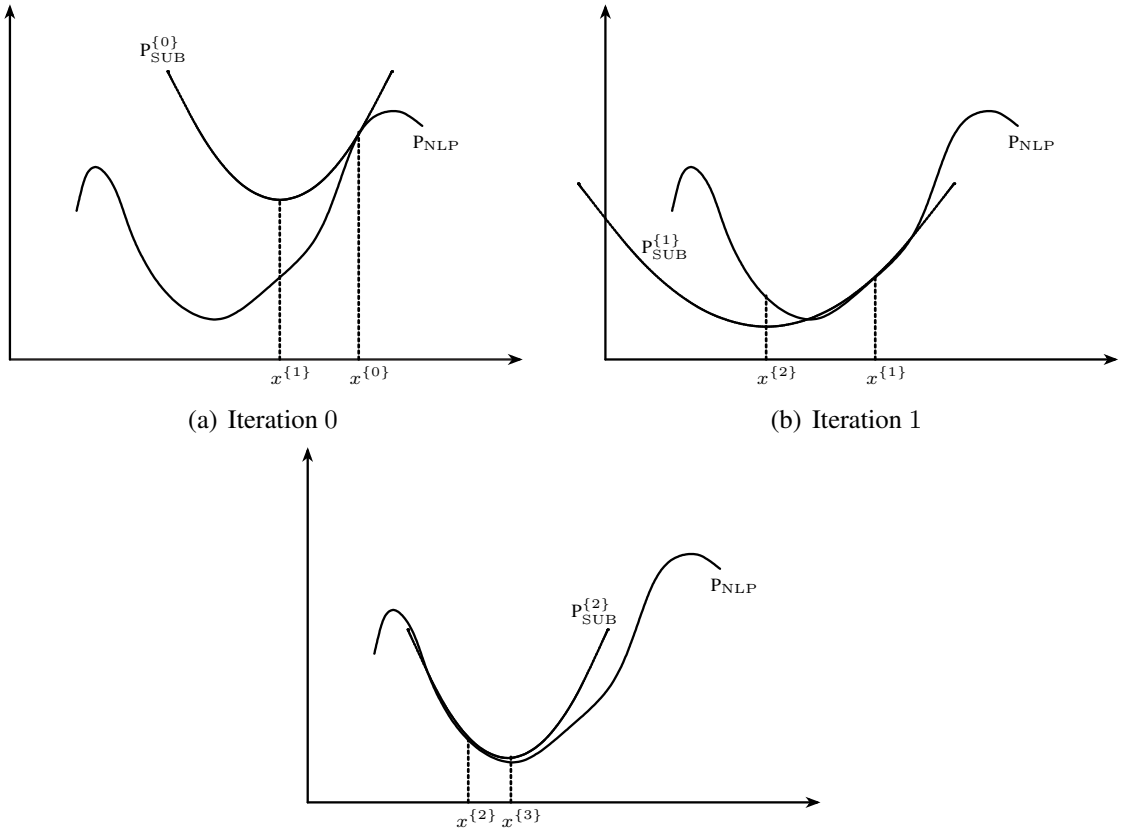
$$\sigma_i = x_i^p \boldsymbol{C}_i \epsilon_i$$

when penalised. Since the components of σ_i all scale the same way, stress measures such as the von Mises stress also scale according to either x_i or x_i^p (depending on whether a penalised or unpenalised formulation is used). If a constraint based on such a stress measure is active for this element, then clearly (for a given elemental strain vector ϵ_i) the value of x_i would have to be higher in the latter case than in the former, because $x_i^p < x_i$. Conceptually then, in a fully stressed design, all the elements for which the stress constraint is active would be driven towards $x_i = 1$. The x_i would be minimised for the other elements due to the action of the objective function.

2.2 Sequential approximate optimisation (SAO)

The conceptual framework for SAO is represented diagrammatically in Figure 2.5. In such a procedure, an explicit surrogate optimisation problem (from here on termed $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$) is derived that approximates the local behaviour of the actual problem (henceforth denoted $\mathrm{P}_{\mathrm{NLP}}$) near to a given point $\boldsymbol{x}^{\{k\}}$ in the design space, using information from $\mathrm{P}_{\mathrm{NLP}}$ evaluated at that point. This surrogate problem, known as the approximate subproblem, is solved using a standard mathematical programming (MP) procedure, rather than solving $\mathrm{P}_{\mathrm{NLP}}$. Since $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$ is constructed from elementary (usually convex) functions, it is much easier and much more efficient to evaluate and to minimise, iteratively if necessary, than $\mathrm{P}_{\mathrm{NLP}}$.

The point in the design space that denotes the solution (the optimum) to the approximate subproblem, namely $\boldsymbol{x}^{\{k+1\}}$, provides an approximation to the optimum of the original problem. Conceptually, a re-analysis can be carried out at this point and the information derived thereby can be



(a) Iteration 0

(b) Iteration 1

(c) Iteration 2

Figure 2.5: A sequence of SAO solutions to an unconstrained problem.

used to construct another approximate subproblem $P_{SUB}^{\{k+1\}}$. This process is iterated, the superscript k representing the iteration number, and thus produces a series of points – each representing the solution to a subproblem – that (ideally) converges to a local optimum of the original problem P_{NLP}. With the imposition of various restrictions that facilitate stable convergence characteristics, said convergence can be shown theoretically to occur. Hence, in this manner, the bulk of the numerical calculations required during an iterative optimisation procedure are carried out during the evaluation of the elementary functions comprising the approximate subproblems, and the number of expensive re-analyses required for the evaluation of P_{NLP} is kept to a minimum.

For example, probably the earliest and most straightforward example of an SAO algorithm is sequential linear programming (SLP), in which the approximate subproblems are constructed as linear programming (LP) problems. This is done by taking first-order Taylor series expansions of the objective and constraint functions that comprise the original problem at the current point $\boldsymbol{x}^{\{k\}}$ (the current approximate solution) in the design space:

$$\tilde{f}_j(\boldsymbol{x}) = f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n} \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \left(x_i - x_i^{\{k\}}\right) \qquad j = 0, 1, 2, \cdots, m. \tag{2.14}$$

Here, the index j denotes the particular function considered. By convention $j = 0$ denotes the objective function, while $j = 1, 2, \cdots, m$ denotes the associated constraint, of which there are m

in total. The notation

$$\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \equiv \frac{\partial f_j}{\partial x_i}\left(\boldsymbol{x}^{\{k\}}\right)$$

signifies a constant that is determined by evaluating the partial derivative at the point $\boldsymbol{x}^{\{k\}}$. In SLP, the relevant information required from the re-analysis, that is necessary for the construction of the subproblems, consists of the function values and gradients of the structural objective and constraints at the point of approximation $\boldsymbol{x}^{\{k\}}$. Hence, if the original problem is represented by the following nonlinear programming problem[10] $\mathrm{P_{NLP}}$

$$\begin{array}{rll} \min\limits_{\boldsymbol{x}} & f_0(\boldsymbol{x}) & \\ \text{subject to} & f_j(\boldsymbol{x}) \leq 0 & j = 1, 2, \cdots, m, \\ & 0 < \check{x} \leq x_i \leq \hat{x} & i = 1, 2, \cdots, n, \end{array} \tag{2.15}$$

then an SLP subproblem derived at the point $\boldsymbol{x}^{\{k\}}$ would be the linear programming problem $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$

$$\begin{array}{rll} \min\limits_{\boldsymbol{x}} & \tilde{f}_0(\boldsymbol{x}) = f_0\left(\boldsymbol{x}^{\{k\}}\right) + \sum\limits_{i=1}^{n}\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}}\left(x_i - x_i^{\{k\}}\right) & \\ \text{subject to} & \tilde{f}_j(\boldsymbol{x}) = f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum\limits_{i=1}^{n}\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\left(x_i - x_i^{\{k\}}\right) \leq 0 & j = 1, 2, \cdots, m, \\ & 0 < \check{x} \leq x_i \leq \hat{x} & i = 1, 2, \cdots, n. \end{array}$$

Note that, although the objective f_0 and constraints f_j that represent the original structural behaviour are functions of the design variables, it is not usually possible to express them as simple algebraic statements. However, the functions that comprise the approximate subproblem ($\tilde{f}_0$ and the $\tilde{f}_j$) can be expressed as simple algebraic statements, in which the function values and sensitivities of the original problem, evaluated at $\boldsymbol{x}^{\{k\}}$, simply appear as constants.

The surrogate linear programming problem thus derived has function values and gradients that can be evaluated easily and efficiently at any point $\boldsymbol{x}$. Its optimum can be found using one of the many standard efficient solution algorithms for linear programming (such as the SIMPLEX method or one of the interior point methods developed for LP). This is one of the advantages of SAO: having derived a subproblem of a standard form it is often possible to utilise an existing tried-and-tested optimisation algorithm for its solution.

Convergence of the SAO procedure to a local optimum of $\mathrm{P_{NLP}}$ has been proved for many SAO algorithms (see for instance [21] for the proof of convergence for an SLP algorithm equipped with a so-called NLP filter). Generally speaking, the two ingredients that are often relied on to produce convergence are firstly that the subproblems be accurate to first order and secondly that the SAO routine includes a mechanism for encouraging global convergence (such as the NLP filter). First-order accuracy means that the gradients of the objective and constraint functions in the subproblem should match the sensitivities of the objective and the associated constraints in the original problem. This ensures that if the necessary conditions for a local optimum[11] of the

[10] For the sake of notational simplicity only one type of constraint is represented here, although both equality and inequality constraints may generally be present.

[11] The familiar KKT conditions. See [22], for example.

original problem are satisfied at a point $\boldsymbol{x}^*$, then the subproblem defined at $\boldsymbol{x}^*$ will also satisfy the necessary conditions at $\boldsymbol{x}^*$. In turn, this implies that the SAO procedure will recognise (that is, terminate at) the local optimum $\boldsymbol{x}^*$. It also implies that, if a direction of descent is identified for the subproblem $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$, it will also be a descent direction for the original problem $\mathrm{P}_{\mathrm{NLP}}$ (at least locally at the point of approximation)[12].

For constrained optimisation, the idea of 'descent' must be qualified, since minimisation of the objective function value and reduction of the (possible) constraint violation are both facets of the optimisation process. For example, a particular design update may decrease the objective function value but, by so doing, increase the measure of constraint violation. Alternatively, if the current design is infeasible, a reduction in the constraint violation may require that the objective function value increases. As the constraints must be strictly satisfied at the solution of an MP problem, the minimisation of the constraint violation takes precedence over the minimisation of the objective function. The classical way of combining these two imperatives in an optimisation procedure is through the use of penalty methods, which have a long history in mathematical programming. In a penalty method, an unconstrained problem is defined by adding penalised functions of the infeasibilities to the original objective function. Although many different penalty formulations exist, the idea is essentially that

$$f_0^{pen}\left(\boldsymbol{x}\right) = f_0\left(\boldsymbol{x}\right) + \sum_{j=1}^{m} \lambda_j \left|f_j\left(\boldsymbol{x}\right)\right|_+ ,$$

in which

$$\left|f_j\left(\boldsymbol{x}\right)\right|_+ = \begin{cases} f_j\left(\boldsymbol{x}\right) & \text{if } f_j\left(\boldsymbol{x}\right) > 0\,, \\ 0 & \text{otherwise.} \end{cases}$$

A direction of descent for the problem would then be one along which the value of this penalised objective function decreases. Convergence to feasible solutions is encouraged by increasing the penalties associated with the constraint violations, thereby accentuating the importance of the violated constraints over the objective function and the feasible constraints.

In Chapter 8 we argue that the dual method inherently contains such a penalisation scheme (of the form given above). We show that the dual variables (the Lagrange multipliers) associated with the initially infeasible constraints can be seen as penalty parameters that scale the importance of the constraint functions relative to the objective function. This behaviour is useful when the dual method is used to solve SAO subproblems. If an SAO subproblem $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$ is constructed at a point that is infeasible for the original problem $\mathrm{P}_{\mathrm{NLP}}$, it can sometimes occur that $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$ has no feasible solution. The dual method then has the ability to locate points of minimum infeasibility if the subproblems are convex, and it can be shown that the sequence of SAO iterates so produced will restore feasibility. As a consequence, the dual SAO scheme described in Chapter 8 is able to cope with infeasible starting points.

The second ingredient for convergence, the use of a global convergence procedure, represents an important 'restriction' on the native behaviour of SAO algorithms, without which convergence to local optima usually cannot be demonstrated. In an SAO algorithm, if the solution to subproblem $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$ defined at $\boldsymbol{x}^{\{k\}}$ is unconditionally accepted as the point at which the following subproblem

[12] Barring pathological occurrences, such as the Maratos effect.

$\mathrm{P}_{\mathrm{SUB}}^{\{k+1\}}$ is defined, namely $\boldsymbol{x}^{\{k+1\}}$, then the sequence of solutions $\boldsymbol{x}^{\{k+1\}}$, $k = 1, 2, \cdots$ is not necessarily guaranteed to converge at all. Indeed, the sequence may oscillate indefinitely or even diverge. Global convergence mechanisms curb such behaviour by controlling the step that can be taken from $\boldsymbol{x}^{\{k\}}$, and/or by ensuring that the sequence produced has monotone descent characteristics. The global convergence mechanisms include the linesearch routines (carried out, perhaps, on a penalty function, such as the merit functions advocated for sequential quadratic programming [23]), trust regions [24], filtering (in the sense of Fletcher and Leyfer [25]), and the use of conservative convex and separable approximations [6].

The various sequential approximate optimisation algorithms differ primarily in how the subproblems are defined – that is, the specific function approximations that are chosen to construct the subproblems – and in what method is chosen to solve the subproblems. There are pros and cons associated with the different choices that may be made as regards these two. We have already introduced sequential linear programming, and it was noted that an advantage of this approach is the availability of various trusted and efficient algorithms for solving the subproblems. Another advantage is that, since only function values and gradient information are required to construct the subproblems, computer storage requirements are far less than for algorithms that require curvature information to be stored as well, and this means that the solution of comparatively larger problems can be attempted. However, for the same reason, convergence can be expected to be poorer (in terms of the number of iterations required) than for the algorithms that take advantage of second-order information.

Probably the most successful algorithm that uses curvature information is sequential quadratic programming (SQP). In SQP, Newton's method is applied to the system composed of the following two KKT conditions

$$\begin{bmatrix} \nabla_x \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) \\ f_j\left(\boldsymbol{x}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

in which $\mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right)$ denotes the Lagrangian of the problem (which will be introduced in Section 2.3). The solution to the resulting system of linear equations (known as the KKT system) yields an approximation to a saddle point of the Lagrangian, i.e. a KKT point of the problem. Although the algorithm is not derived explicitly as an SAO approach, it is equivalent to solving the following quadratic subproblem during each iteration k of the algorithm:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\boldsymbol{d}^T \left[W\left(\boldsymbol{x}^{\{k\}}, \boldsymbol{\lambda}^{\{k\}}\right)\right] \boldsymbol{d} + \left[\nabla f_0\left(\boldsymbol{x}^{\{k\}}\right)\right]^T \boldsymbol{d} \\ \text{subject to} \quad & \left[\nabla f_j\left(\boldsymbol{x}^{\{k\}}\right)\right]^T \boldsymbol{d} + f_j\left(\boldsymbol{x}^{\{k\}}\right) = 0 \qquad j = 1, 2, \cdots, m_a\,. \end{aligned}$$

Here, m_a represents a set of active constraints, treated as equalities, which must be identified during each iteration. The solution to the KKT system locates the minimum of the above approximate quadratic objective function on the null space of the chosen active constraints. The matrix W in the objective function is the Hessian of the Lagrangian $\nabla_{xx}\mathcal{L}$, evaluated at the current approximation to the saddle point $\left(\boldsymbol{x}^{\{k\}}, \boldsymbol{\lambda}^{\{k\}}\right)$; the subproblem is here written in terms of $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}^{\{k\}}$. To encourage convergence, a line search is often carried out on a penalty merit function in the direction of the approximated optimum to identify the next iterate. For details, the reader is referred to [23].

SQP is highly regarded because it has excellent convergence properties, which it inherits from Newton's method. The rate of convergence local to a KKT point is theoretically quadratic, provided that the active constraint set can always be identified consistently and that the Hessian of

the Lagrangian remains positive definite. Indeed, SQP is probably considered the state of the art for moderately sized problems. For large-scale problems, however, the algorithm suffers from the necessity of having to evaluate and store the fully populated Hessian of the Lagrangian, in addition to the gradient vectors of the objective function and constraints that are identified as active in a given iteration. Even if the Hessian can be approximated from first-order information using a quasi-Newton method such as BFGS (see for instance [26]), which alleviates the necessity for evaluating the curvature terms, the Hessian still needs to be stored and the resulting linear system has to be manipulated. Much research is currently being devoted to finding more efficient methods of deriving and solving the linear system in SQP (for instance by using efficient sparse solvers).

2.2.1 The dual SAO approach for structural optimisation

According to Fleury [1], there had been two dominant approaches to the solution of structural optimisation problems[13]. The first was the use of optimality criterion (OC) methods, wherein designs are improved or updated using rules derived from statements of the optimality criteria for a problem. In a very general sense, these are statements that are thought to be valid at the optimum of a problem, and are not valid elsewhere. The form of a given optimality criterion often suggests a scheme (which is usually heuristic) by which a non-optimal design may be improved, such that the scheme will produce no design changes for optimal designs (for which the OC is satisfied). The second approach entails the use of the more rigorous, but often less efficient, methods of mathematical programming.

Haftka and Gürdal [27] point out that the OC methods were generally not viewed favourably outside of the structural optimisation community. One reason is that some of these methods lack mathematical rigour, relying on ad hoc updating schemes and/or intuitive optimality criteria. The other principal criticism frequently levelled at the OC approaches is that, even if the OC are rigorous, they are often used in problems for which they are not strictly valid, having been derived for other problems. Be this as it may, many of these methods yield very efficient algorithms whose facility can be demonstrated practically. In comparison, the methods of mathematical programming are of course recognised as having a strong theoretical foundation. The main impediment to their use in structural optimisation is that they are frequently computationally inefficient (relative to OC approaches), a drawback that becomes more acute the larger the considered problems become.

Hence, Fleury intimates that in the 1970s there were two communities working on converging lines of research regarding the optimisation procedures used in structural optimisation. On the one hand was the comunity of OC practitioners, who were seeking more generally applicable methods derived from rigorous optimality criteria, using physically justifiable and interpretable update schemes. On the other hand was the school of researchers using mathematical programming algorithms, whose goal was to develop more efficient algorithms using the precepts of MP.

The two fields (or more specifically, certain techniques therein) were formally unified by Fleury in 1978. Fleury showed that a general approach emerging in the OC school at that time could be interpreted as a method of MP. Concomitantly, the favoured MP approach of the time could similarly be seen as an OC method. Under certain conditions, an exact equivalence could be demonstrated.

[13] Fleury considered specifically the weight minimisation problem.

The convergence of the two fields appears to have been the result of the widespread use of the fact that there is a reciprocal-like dependence of many structural responses on the variables in structural optimisation problems. This dependence was utilised in (that is: explicitly built into) both the construction of OC updates and the generation of efficient MP algorithms. A recognition of the close relationship between the two approaches stemmed from another pivotal ingredient. This was the use of the dual approach, both as a method for solving MP problems and as a means to analyse or interpret the OC approaches.

Fleury proceeded to derive a generalised method for the solution of the structural weight minimisation problem based on the linearisation of the problem at a point in the design space. This "generalised OC approach" [28] involved the use of the dual method to produce the design update from the linearised subproblem, with this procedure being repeated iteratively. The linearisation was accomplished either directly in terms of the design variables (a first-order Taylor series expansion) or in terms of the reciprocals of the design variables. When interpreted from an MP point of view, it is clear that a series of approximate subproblems are derived and solved using the dual method to find the stationary point of the subproblems, which, because they are derived from first-order approximations, are separable in the primal design variables. Separability is of chief importance in making the dual solution method viable, and the dual method advocated by Fleury (in [28]) was that introduced by Falk. This last is no less an important introduction, since without Falk's version of the dual, the dual problem quickly becomes prohibitively large when problems with many variables are considered due to the existence of the side (bound) constraints on the primal variables.

Fleury subsequently limited the subproblems to particular strictly convex forms by introducing a consistent way of deciding on the form of the separable approximations used to describe the problem. The method decides between the linear and reciprocal forms to model the dependencies of each of the functions comprising the problem on each of the design variables. The method is consequently termed a 'method of mixed variables', and the resultant algorithm became known as CONLIN, for convex linearisation [4, 5].

2.2.2 A brief description of OC methods

Given the historical significance of OC methods, as well as their continued use, it is instructive to elaborate on their connection to dual MP methods before describing the MP method used herein. What follows in this section is précised from Fleury [1] and from Haftka and Gürdal [27].

According to [27], most OC methods typically utilise a rigorously derived optimality criterion based on the Karush-Kuhn-Tucker (KKT) conditions, in combination with a heuristic rule for updating the design variables. If there are m_a active constraints at the optimum, then the OC is typically the condition that

$$\frac{\partial f_0}{\partial x_i} - \sum_{j=1}^{m_a} \lambda_j \frac{\partial f_j}{\partial x_i} = 0\,. \tag{2.16}$$

In describing a general OC approach, Fleury uses structural weight minimisation as an example and departs from an OC method that uses the concept of virtual strain energy. The problem he describes is the minimisation of structural weight subject to constraints on the allowable displacements of certain points in the structure, where the structure is discretised and analysed using the

finite element method. The objective function for the optimisation problem is thus

$$W = \sum_{i=1}^{n} \rho_i \nu_i x_i \,,$$

in which the volume of element i is given by the product $\nu_i x_i$, and the mass density is ρ_i. For truss problems, the ν_i can be interpreted as truss member lengths and the x_i represent cross-sectional areas, whereas for planar structures the ν_i represent elemental volumes and the x_i would then represent the presence or absence of material within the element (when solid-void solutions are sought). If the x_i are allowed to attain real values between 0 and 1, the corresponding problem can be interpreted as weight minimisation of a variable thickness sheet, in which case the x_i are seen as normalised thicknesses and the ν_i as elemental areas.

It is here tacitly assumed that the global stiffness matrix is a linear function of the design variables

$$K = \sum_{i=1}^{n} K_i x_i \,, \tag{2.17}$$

in which the K_i represent the individual element stiffness matrices. This assumption is often valid for discretised structural sizing problems. After Barnett [29] and Berke [30], using the principle of virtual strain energy the prescribed constraints on the structural displacements may be written as

$$u = \boldsymbol{q}^t \tilde{\boldsymbol{g}} = \boldsymbol{q}^t K \tilde{\boldsymbol{q}} \,. \tag{2.18}$$

Here, $\tilde{\boldsymbol{g}}$ denotes a virtual load applied at the node to which the displacement constraint applies (initially only a single constraint f_1 is considered), and $\tilde{\boldsymbol{q}}$ is the associated structural displacement vector. Due to (2.17), in the structural analysis the structural responses are implicitely a function of the design variables x_i. The displacement constraints (2.18) can be written explicitly as functions of the design variables as

$$u = \boldsymbol{q}^t K \tilde{\boldsymbol{q}} = \sum_{i=1}^{n} \frac{c_i}{x_i} \,. \tag{2.19}$$

This is an exact representation of a nodal displacement response for statically determinate structures, for which assumption (2.17) holds, the displacements being inversely proportional to the design variables. For such structures, the coefficients c_i are constant and can be written as [1, 27]

$$c_i = \left(\boldsymbol{q}_i^t K_i \tilde{\boldsymbol{q}}_i \right) x_i \,. \tag{2.20}$$

This is no longer the case for statically indeterminate structures, but (2.19) then represents a good first-order approximation (or linearisation) of the response. If the constraint is assumed active, such that

$$f_1(u) = u - \bar{u} = 0 \,, \tag{2.21}$$

with $\bar{u}$ some prescribed limit on the nodal displacement and u according to (2.19), then the application of the KKT condition (2.16) yields

$$\lambda = \frac{\partial f_0}{\partial x_i} \Big/ \frac{\partial f_1}{\partial x_i} \qquad \forall \quad i$$

for the single Lagrange multiplier. This expression can be interpreted as stating that, at the optimum, all design variables are equally cost effective at producing a change in the constraint value (the numerator being the 'cost' associated with effecting a change in the constraint value by changing x_i)[14]. An OC update scheme derived from this, and discussed by Fleury [1], Haftka and Gürdal [27] and others, is

$$x_i^{\{k+1\}} = x_i^{\{k\}} \left[\lambda \frac{c_i}{\rho_i \nu_i x_i^2} \right]^{0.5}, \tag{2.22}$$

in which the superscript k denotes the iteration number. The value of the multiplier λ at the optimum is, of course, not known a priori, but it can be estimated by requiring that the constraint, given by (2.21) and (2.19), is active at the optimum. This implies that (2.19) and (2.20) are either accurate (and therefore valid at the optimum), or are at least good local approximations.

In [1], Fleury showed that stress constraints can be handled in much the same way as displacement constraints, using the virtual work method and yielding expressions of the form

$$\sigma = \sum_{i=1}^{n} \frac{d_i}{x_i}$$

for the prescribed elemental stresses. Furthermore, he broadened the above discussion to include multiple constraints as well as inequality constraints. When multiple constraints of the form (2.19) are considered, the KKT optimality criterion generalises to

$$x_i^2 = \frac{1}{\rho_i \nu_i} \sum_{j=1}^{m} \lambda_j c_{ij} \tag{2.23}$$

(see [27]), from which the following update rule is derived (although other updates based on (2.23) are also used):

$$x_i^{\{k+1\}} = x_i^{\{k\}} \left[\frac{1}{\rho_i \nu_i \left(x_i^{\{k\}} \right)^2} \sum_{j=1}^{m} \lambda_j c_{ij} \right]^{0.5}. \tag{2.24}$$

For multiple inequality constraints the difficulty lies in finding the values of the m Lagrange multipliers at the optimum. Condition (2.23) is valid if all the constraints are active at the optimum (at least in the statically indeterminate case). Constraints that are inactive at the optimum can either be excluded from condition (2.23), or else their associated Lagrange multipliers can be assigned the values $\lambda_j = 0$, which amounts to the same thing. This last follows from the KKT conditions for inequality constrained problems and, incidentally, results naturally when the MP-plus-dual approach is used to solve the problem.

Since the set of active constraints is not known a priori, an iterative method is required to identify the active constraint set $\mathcal{A}$ and then to solve the system of equations resulting from setting

$$f_j \left(\boldsymbol{x}^{\{k+1\}} \right) = 0 \qquad j \in \mathcal{A},$$

[14] This is discussed specifically in the context of OC methods in [27], and more generally in terms of Lagrange multiplier theory in [22].

which yields a linear system of equations to be solved for the λ_j, $j \in \mathcal{A}$, with $\boldsymbol{x}^{\{k+1\}}$ given in terms of the Lagrange multipliers by (2.24). The task of determining the active constraint set is a difficult problem in itself, being combinatorial in nature, and the effort required to do so scales very badly as the number of constraints increases. Note that it is a linearisation of the f_j that is used in the above condition $f_j = 0$, which again furnishes only an approximate solution in the case of statically indeterminate structures, and so this process must be repeated iteratively to converge on a solution to the problem. One important question that arises immediately, therefore, is whether this process can be expected to converge at all.

In [1], Fleury proceeds to demonstrate that an efficient method of solving the problem can be derived from the application of the mathematical programming approach by solving a sequence of linearised subproblems derived from the original problem. In these subproblems, the objective function is expressed as a linear function of the design variables, as it is in the original problem. Due to the recognition of the form of the structural responses embodied by (2.19), the (displacement and/or stress) constraints are expressed explicitly as first-order Taylor series expansions in terms of the reciprocals of the design variables, namely

$$\tilde{f}_j\left(\boldsymbol{x}\right) = f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{x_i^{\{k\}}}{x_i}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}. \tag{2.25}$$

This being the case, the evaluation of the KKT conditions at the optimum of the linearised approximate subproblem yield exactly the optimality conditions (2.23). Hence, the general OC approach described above is interpereted from an MP perspective as furnishing the solutions to a series of linearised subproblems constructed from first-order Taylor series expansions of the actual problem. These are defined successively at the 'current' working point in the design space $\boldsymbol{x}^{\{k\}}$, in terms of either the design variables directly or the reciprocals thereof. The advantage of this interpretation is that the convergence properties of the OC method as presented here are understood, being the convergence properties of the associated method of MP. Standard methods for encouraging global convergence, such as conservatism [6], then gain relevance.

More importantly perhaps is that standard methods for solving the MP subproblems acquire significance in the OC framework. In particular, the dual method offers an efficient alternative for solving the subproblems. As such, the dual method represents a consistent approach for calculating the values of the Lagrange multipliers (the dual variables) at the optimum. Moreover, the method inherently provides a means of distinguishing the active from the inactive constraints when the defined constraints are inequalities (which is usually the case).

One small complication that should be noted in the derivation of (2.23), as it is important for some of the work presented in this document, is the assumption of convexity. Equation (2.23) is derived from the familiar KKT condition

$$\min_{\boldsymbol{x}} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) = 0\,.$$

Therefore there is an inherent assumption that the Lagrangian of the problem possesses a turning point (in fact a unique turning point, if the intention is to use a dual solver) with respect to $\boldsymbol{x}$ for any $\boldsymbol{\lambda}$, and that this turning point corresponds to a minimum and not a maximum. This is not necessarily the case for the weight minimisation problem itself, nor generally for other structural optimisation problems. Therefore, in an MP approach, the subproblems $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$ are almost always

derived from strictly convex approximations, whether or not the original problem $\mathrm{P_{NLP}}$ is locally convex.

The combination of using a strictly convex programming approach in parallel with a dual solver is now a well established methodology for solving structural optimisation problems, particularly when problems with a large number of variables and a small to moderate number of constraints are considered. Methods such as Fleury's CONLIN and Svanberg's method of moving asymptotes (MMA) [3] are recognised as being both robust and computationally efficient. Indeed, and of particular importance for the work presented herein, algorithms of this type appear to be the standard in the topology optimisation and sizing community.

Finally it should be noted that equation (2.22) derives from the approximate form of the weight minimisation problem, that is: a problem with a linear objective and a reciprocal-like constraint. A generalisation of (2.22) for problems with different forms is

$$x_i^{\{k+1\}} = x_i^{\{k\}} \left[\lambda \left(\pm \frac{\partial g}{\partial x_i} \Big/ \frac{\partial f}{\partial x_i} \right) \right]^{\frac{1}{\eta}}, \tag{2.26}$$

in which the term in brackets is positive. If the Lagrangian function of the linearised or approximated problem possesses a unique minimum with respect to $\boldsymbol{x}$ for any choice of $\boldsymbol{\lambda}$, the bracketed term will be positive[15]. In the OC paradigm, η acts as a parameter that controls the size of the design changes $\boldsymbol{x}^{\{k\}} \rightarrow \boldsymbol{x}^{\{k+1\}}$ from one iteration to the next [27].

2.2.3 Examples of SAO algorithms used in structural optimisation

The SAO algorithms commonly used to solve topology optimisation problems have evolved to be suited to large structural problems. It has become standard practice to use only first-order approximations as the explicit functions that are used to construct the subproblems. However, these approximations are selected to be good local approximations for the structural responses, the local characteristics of which are frequently known. Specifically, it is standard to utilise the first-order Taylor expansion in terms of the reciprocals of the design variables (or variations thereof), discussed in Section 2.2.2.

As has been pointed out, the use of first-order approximations enables larger problems to be tackled by limiting the necessitated storage, as well as limiting the amount of information that needs to be evaluated from the original problem during the definition of the subproblems. Another important point is that the use of first-order approximations results in separable subproblems, which is a crucial characteristic if the dual method is to be used for the solution of the subproblems. Finally, these functions can be used to generate strictly convex subproblems, which guarantees either a unique solution to each subproblem or a unique point of minimum infeasibility if a subproblem happens to be infeasible (that is: if it lacks a feasible region).

The algorithms discussed in this section then use the dual method for solving the subproblems. The dual problem will be discussed in Section 2.3. For now it is sufficient to say that, under certain conditions, such as continuity and convexity of the primal subproblem, the dual subproblem is

[15] This is seen as a crucial requirement if the dual method is to be employed in an SAO strategy, which shows again the link between the OC and dual SAO approaches.

a concave function whose stationary point (its maximum) is equivalent to the KKT point (the solution) of the primal subproblem. However, the dual is often easier to solve than the primal subproblem, for the following reasons: Whether the primal subproblem is convex or not, if the dual of the subproblem can be defined uniquely it will be a concave function. Moreover, the dual subproblem has only simple bound constraints on the dual variables, which are easier to deal with than the general constraints applied to the primal subproblem. The dual subproblem often also is smaller than the primal subproblem; its dimensionality is equal to the number of constraints in the primal subproblem, which is usually less than the number of primal variables. Given the concave, simply bounded form, many standard optimisation algorithms exist that can be used for its maximisation. If the gradients of the dual are required by such an algorithm, they are easily evaluated because they correspond to the function values of the associated constraints in the primal subproblem.

The main complication in the application of the dual method is the conversion of the primal subproblem to the dual subproblem, which requires additional computations during the optimisation process. As will be discussed in Section 2.3, the primal and the dual are related by a series of equations that facilitate the calculation of the values of the primal variables corresponding to specific coordinates in the domain of the dual. The efficient evaluation of these primal-dual relationships demands that the primal subproblem be separable. In the case of the algorithms discussed below, the primal-dual relationships produced have algebraic expressions that can be hard-coded.

It should be noted finally that, although first-order primal approximations are standard, they are not a necessity, either for efficient SAO algorithms for structural optimisation or for the use of dual solvers. There are several methods available that incorporate limited information about the curvatures of the original problem into the primal subproblem and that at the same time preserve the separability and convexity of the primal subproblem, and yield easily solvable primal-dual relationships (see, for instance, the SAOi algorithm of Groenwold and Etman [31]). When this is done, however, the curvature information is limited, at most, to the diagonal elements of the Hessians of the functions describing the original problem, or approximations thereof. Otherwise the additional computational and storage requirements may again become prohibitive.

CONLIN

In convex linearisation [4, 5] the approximate subproblems are generated at the point $\boldsymbol{x}^{\{k\}}$ by applying the following approximation to each function f_j, $j = 0, 1, 2, \cdots, m$, of the optimisation problem $\mathrm{P_{NLP}}$:

$$\tilde{f}_j = f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{\{+\}_j}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \sum_{\{-\}_j}\left(x_i - x_i^{\{k\}}\right)\left(\frac{x_i^{\{k\}}}{x_i}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}. \quad (2.27)$$

The notation $\{+\}_j$ represents the set of all indices i for which the partial derivative of the function f_j with respect to x_i is positive, from which the definition of $\{-\}_j$ follows accordingly,

$$\{+\}_j = \left\{ i : \frac{\partial f_j}{\partial x_i} \geq 0,\ j = 0, 1, 2, \cdots, m \right\},$$
$$\{-\}_j = \left\{ i : \frac{\partial f_j}{\partial x_i} < 0,\ j = 0, 1, 2, \cdots, m \right\}.$$

Hence, a direct linearisation is carried out for the sensitivities belonging to the set $\{+\}_j$, whereas a reciprocal linearisation is carried out on the function dependencies that fall into $\{-\}_j$. The approximation (2.27) is therefore termed a mixed linearisation; the subproblems constructed from (2.27) are convex and separable.

MMA

The method of moving asymptotes, due to Svanberg [3, 32], is another very popular optimisation algorithm used in structural optimisation, particularly within the topology optimisation community. In MMA, each function f_j, $j = 0, 1, 2, \cdots, m$, in $\mathrm{P_{NLP}}$ is approximated as

$$\tilde{f}_j = f_j\left(\boldsymbol{x}^{\{k\}}\right) - \sum_{i=1}^{n} \left(\frac{p_{ij}^{\{k\}}}{U_i^{\{k\}} - x_i^{\{k\}}} - \frac{q_{ij}^{\{k\}}}{x_i^{\{k\}} - L_i^{\{k\}}} \right) + \sum_{i=1}^{n} \left(\frac{p_{ij}^{\{k\}}}{U_i^{\{k\}} - x_i} - \frac{q_{ij}^{\{k\}}}{x_i - L_i^{\{k\}}} \right). \quad (2.28)$$

This is an extension of the reciprocal approximation, and also results in strictly convex approximate subproblems. The constants $U_i^{\{k\}}$ and $L_i^{\{k\}}$, calculated at each iteration k, are coordinates at which the approximation asymptotes to infinity. These asymptotes function as a built-in step-size control, the optimum of the subproblem being located definitely within the box defined by the $U_i^{\{k\}}$ and $L_i^{\{k\}}$. Part and parcel of the algorithm is a routine for calculating or adjusting the location of the asymptotes from iteration to iteration. MMA thus comes equipped with a built-in mechanism for encouraging global convergence. For details, the reader is referred to [3]. In equation (2.28), the constants $p_{ij}^{\{k\}}$ and $q_{ij}^{\{k\}}$ are chosen as follows:

$$p_{ij}^{\{k\}} = \begin{cases} \left(U_i^{\{k\}} - x_i^{\{k\}}\right)^2 \left(\dfrac{\partial f_j}{\partial x_i}\right)^{\{k\}} & \text{if } \left(\dfrac{\partial f_j}{\partial x_i}\right)^{\{k\}} > 0, \\ 0 & \text{if } \left(\dfrac{\partial f_j}{\partial x_i}\right)^{\{k\}} \leq 0, \end{cases}$$

$$\text{and} \quad q_{ij}^{\{k\}} = \begin{cases} 0 & \text{if } \left(\dfrac{\partial f_j}{\partial x_i}\right)^{\{k\}} \geq 0, \\ \left(x_i^{\{k\}} - L_i^{\{k\}}\right)^2 \left(\dfrac{\partial f_j}{\partial x_i}\right)^{\{k\}} & \text{if } \left(\dfrac{\partial f_j}{\partial x_i}\right)^{\{k\}} < 0. \end{cases}$$

SAOi

The SAOi algorithm, developed by Groenwold and Etman [31], is a sequential approximate optimisation algorithm primarily intended for the solution of simulation-based inequality constrained

nonlinear optimisation problems[16]. It is based on the use of convex and separable quadratic approximating functions

$$\tilde{f}(\boldsymbol{x}) = f\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n} c_i^{\{k\}}\left(x_i - x_i^{\{k\}}\right)^2, \tag{2.29}$$

in which the curvatures can be tailored by manipulating the constants $c_i^{\{k\}}$. The algorithm exploits some of the advantages of quadratic approximations, but does so without storing exact second-order information (refer to [33] for details). Instead, approximate diagonal second-order information is constructed and stored. By judiciously choosing the curvatures $c_i^{\{k\}}$, it becomes possible to accurately and efficiently optimise problems that exhibit strong monotonicities, like those present in structural optimisation, using quadratic functions. This is achieved by selecting the $c_i^{\{k\}}$ so that the resulting function is a quadratic approximation to either the reciprocal or exponential functions about the point $\boldsymbol{x}^{\{k\}}$. These inverse functions are themselves monotonic approximations at $\boldsymbol{x}^{\{k\}}$ of the nonlinear functional dependencies exhibited by the optimisation problem being solved.

Examples of standard approximations present in the algorithm are the quadratic approximation to the reciprocal approximation, the quadratic approximation to the CONLIN approximation of Fleury and Braibant, and the quadratic approximation to the MMA approximation of Svanberg. Philosophically, using these approximated approximations is very different to using the CONLIN or MMA algorithms themselves. Two distinctly different approximate subproblems may be formulated: a separable quadratic programming problem with quadratic constraints, solved using a dual statement, and a Lagrangian diagonal quadratic program (QP), solved using a QP solver. The former is attractive when the design variables outnumber the constraints by far, and vice versa.

The algorithm is aimed in particular at large-scale optimisation. Thus, the gradients of the constraints may be stored in sparse form and the algorithm comes equipped with solvers that take advantage of the sparsity of the system of equations that is manipulated during the solving of the subproblems. SAOi is used to solve the large-scale stress-constrained material distribution problems discussed in Chapter 9. A more concrete explanation of how the constants $c_i^{\{k\}}$ are selected is also presented there.

2.3 General overview of duality

As we have seen, sequential approximate optimisation methods seek to find a solution to a given (generally nonlinear) programming problem $\mathrm{P_{NLP}}$ by solving a sequence of approximate subproblems, which are easily represented and easily evaluated. In the case of the methods of approximation inherent in the three algorithms discussed above, the subproblems themselves are also nonlinear programming problems, although of a particularly advantageous type, being constructed as strictly convex, separable and continuous. Consequently, they have at most a unique solution, which can be found using calculus-based methods that take advantage of their continuity. The subproblems may thus be solved using any applicable method of constrained nonlinear programming. As has been discussed, however, the dual method of solution is often favoured in structural optimisation, for the reasons highlighted in Section 2.2.3.

[16]Simulation-based problems are those that entail computationally demanding numerical simulations or modelling.

The terms 'dual', 'dual problem' and 'dual method' have a wide variety of meanings in mathematics and even in mathematical programming, the notion of a 'dual' being variously defined in different fields. However, the examples of dual problems used in SAO stem largely from the notion of Lagrangian duality, which itself is born of Lagrange multiplier theory[17].

Lagrange multiplier theory, which is formulated for equality constrained problems, asserts that the following conditions hold at all extrema $\boldsymbol{x}^*$ of an objective function f_0 on the subspace defined by the equality constraints $f_j = 0$, $j = 1, 2, \cdots, m$, provided that the constraints satisfy a constraint qualification at $\boldsymbol{x}^*$:

$$\left(\frac{\partial f_0\left(\boldsymbol{x}^*\right)}{\partial x_i}\right) + \sum_{j=1}^{m} \lambda_j \left(\frac{\partial f_j\left(\boldsymbol{x}^*\right)}{\partial x_i}\right) = 0 \qquad i = 1, 2, \cdots, n, \tag{2.30}$$

$$f_j\left(\boldsymbol{x}^{\{*\}}\right) = 0 \qquad j = 1, 2, \cdots, m. \tag{2.31}$$

The method of Lagrange multipliers converts an optimisation problem into the problem of solving a system of equations, which are linear in the λ_j, but generally nonlinear, and non-separable, in the x_i. It can be shown (see for instance Hadley [22]) that $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$, where $\boldsymbol{x}^*$ represents an extremum of the f_0 on f_j and $\boldsymbol{\lambda}^*$ denotes the associated Lagrange multipliers, is a solution of the above system of equations, provided that an $m \times m$ non-singular submatrix can be selected from the Jacobian of the constraints

$$\mathcal{J} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \dfrac{\partial f_m}{\partial x_2} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}^*,$$

where it is assumed that $n \geq m$ and $\mathcal{J}$ is evaluated at $\boldsymbol{x}^*$. If this condition is satisfied, then there is a unique vector of multipliers $\boldsymbol{\lambda}^*$ associated with $\boldsymbol{x}^*$ that together with $\boldsymbol{x}^*$ satisfies (2.30). Conditions (2.30) and (2.31) may be arrived at succinctly by defining a Lagrangian function that combines the objective and constraint functions into a single structure

$$\mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) = f_0\left(\boldsymbol{x}\right) + \sum_{j=1}^{m} \lambda_j f_j\left(\boldsymbol{x}\right). \tag{2.32}$$

Then, equations (2.30) are obtained by demanding that the solutions satisfy

$$\frac{\partial \mathcal{L}}{\partial x_i} = 0$$

[17] It should be said that some important duals, like the linear programming dual, were not originally developed from Lagrange multiplier theory, but can nevertheless be shown to derive from it.

and

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = 0\,,$$

which generates equations (2.31). These conditions are necessary, though not sufficient, to define the extrema of f_0 on f_j. Hence, in order to locate a global optimum for an equality constrained programming problem, all the solutions to the above system of equations need to be identified and then compared to determine the optimum. There is no algorithm for doing so generally, so in and of itself the method does not necessarily simplify the process of finding a solution to a problem, unless said problem happens to have additional structure that can be exploited.

The well-known Karush Kuhn Tucker conditions are an extension of the above conditions to problems that may also have inequality constraints. For a general nonlinear programming problem defined by

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_j(\boldsymbol{x}) = 0 \qquad j = 1, 2, \cdots, m_e \\ \text{and} \quad & f_j(\boldsymbol{x}) \leq 0 \qquad j = m_e + 1, m_e + 2, \cdots, m, \end{aligned}$$

the KKT conditions, the first-order necessary conditions that an optimum $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ satisfies, are stated as

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)}{\partial x_i} &= 0 \qquad \forall \quad i, \\ \frac{\partial \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)}{\partial \lambda_j} &= 0 \qquad \text{for} \quad j = 1, 2, \cdots, m_e, \\ \frac{\partial \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)}{\partial \lambda_j} &\leq 0 \qquad \text{for} \quad j = m_e + 1, m_e + 2, \cdots, m, \\ \lambda_j \cdot \frac{\partial \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)}{\partial \lambda_j} &= 0 \qquad \text{for} \quad j = m_e + 1, m_e + 2, \cdots, m, \\ \lambda_j &\geq 0 \qquad \text{for} \quad j = m_e + 1, m_e + 2, \cdots, m. \end{aligned} \tag{2.33}$$

Hadley [22] also provides a geometric interpretation of KKT points. KKT points are very often identified with saddle points on the Lagrangian surface, at which (for minima)

$$\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}) \leq \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \leq \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*)\,. \tag{2.34}$$

This equation is valid in the immediate vicinity of $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$, i.e. local to the KKT point, and on the part of the Lagrangian restricted by $\lambda_j \geq 0,\ j = m_e + 1, m_e + 2, \cdots, m$. Within this region the value of the Lagrangian function at the saddle point can be obtained as

$$\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \tag{2.35}$$

and, given the KKT conditions above, is equivalent to the objective function value at the local optimum, i.e.

$$f_0(\boldsymbol{x}^*) = \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)\ .$$

Central to Lagrange multiplier theory and the KKT conditions is that a unique relationship exists between the primal variables and the Lagrange multipliers at a KKT point. This unique correspondence also extends to the domain local to a KKT point. From (2.35) it may be ascertained that, for any $\boldsymbol{\lambda}^\dagger$ close to $\boldsymbol{\lambda}^*$, there is an $\boldsymbol{x}^\dagger$ close[18] to $\boldsymbol{x}^*$ that satisfies

$$\boldsymbol{x}^\dagger = \arg\min_{\boldsymbol{x}} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}^\dagger\right) .$$

The function of $\boldsymbol{\lambda}$ obtained by carrying out the minimisation in (2.35) is the dual function associated with the saddle point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$, namely

$$\gamma\left(\boldsymbol{\lambda}\right) = \min_{\boldsymbol{x}} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right), \tag{2.36}$$

which is 'dual' to f_0 in the sense that, for any feasible point $\left(\boldsymbol{x}^\dagger, \boldsymbol{\lambda}^\dagger\right)$,

$$\gamma\left(\boldsymbol{\lambda}^\dagger\right) \leq f_0\left(\boldsymbol{x}^\dagger\right). \tag{2.37}$$

Furthermore, given the KKT conditions, it is expected that the minimiser $\boldsymbol{x}^\dagger$ will satisfy the condition

$$\nabla_{\boldsymbol{x}} \mathcal{L}\left(\boldsymbol{x}^\dagger, \boldsymbol{\lambda}^\dagger\right) = 0 . \tag{2.38}$$

That $\boldsymbol{x}^\dagger$ is a unique minimiser of the Lagrangian with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^\dagger$ is certainly not a global characteristic of the Lagrangian in general. As with Lagrange multiplier theory, neither the KKT conditions nor the notion of a dual automatically gives rise to an algorithm for actually finding the KKT points for an arbitrary nonlinear programming problem. However, if the problem is strictly convex, then its Lagrangian is strictly convex in $\boldsymbol{x}$. In this case the problem has a unique KKT point (which corresponds to the global minimum), the Lagrangian surface has a unique saddle point so that (2.34) is valid globally, the dual (2.36) is defined uniquely and there is no duality gap. This last means that, at the solution $\left(\boldsymbol{\lambda}^\dagger, \boldsymbol{x}^\dagger\right)$, equation (2.37) is satisfied as an equality. Thus, when duality is used in sequential convex programming (SCP), in which strictly convex subproblems are defined during every iteration k, the subproblems may be solved by first defining the dual of the subproblem and then maximising the dual. For SCP, Wolfe [34] defined the dual by generating the primal-dual relationships through the application of (2.38).

Limitations

Even for strictly convex problems, the use of Lagrange multiplier theory and the definition of the dual that is commonly used in SCP have many limitations. Chief amongst these is that, if the problem possesses bound constraints on its primal variables (as is the case in both the structural optimisation problems of weight minimisation and minimum compliance), then each of these bound constraints is associated with a Lagrange multiplier in the definition of the Lagrangian, just as any other constraint is. Since each primal variable contributes at least one dual variable, the dual problem is at least as large as the primal problem in this case, and often considerably larger. This undermines the utility of the dual method.

[18] This notion of closeness has a rigorous definition; e.g. see Hadley [22].

Secondly, even for strictly convex problems, equation (2.38) may not have solutions $\boldsymbol{x}^\dagger$ for all $\boldsymbol{\lambda}^\dagger$. Consider, for instance, if the Lagrangian is strictly reciprocal in $\boldsymbol{x}$. A reciprocal function does not possess a stationary point at finite $\boldsymbol{x}$. Also, the approximate subproblems may not be defined for all $\boldsymbol{x}$. The reciprocal function is undefined at $\boldsymbol{x} = \mathbf{0}$, for instance. In fact, if the Lagrangian has a reciprocal form, then it is strictly concave on $\boldsymbol{x} < \mathbf{0}$, so without a means of limiting the range of validity within which the problem is to be considered, useful subproblems from which a dual can be derived consistently cannot be derived from functions like the reciprocal approximation.

Hadley, in his consideration of KKT points as saddle points of the Lagrangian in [22], already addresses these concerns. He incorporates into his definition of a saddle point a means of limiting the domain $\boldsymbol{x}$ over which the Lagrangian is considered. In his case, $\boldsymbol{x} \geq \mathbf{0}$. Instead of incorporating this restriction on $\boldsymbol{x}$ into the definition of the Lagrangian as additional constraints, Hadley demonstrates that these restrictions may instead be incorporated into the first-order optimality conditions defining the saddle point (now on the more restricted domain). If the Lagrangian is strictly convex over this domain, Hadley indicates that the saddle point defined hereby is unique.

Falk [2], in his definition of a dual method for nonlinear programming, presents a much more general analysis that addresses these same limitations. Instead of considering the restricted domain $\boldsymbol{x} \geq \mathbf{0}$, he considers a general closed and compact domain $\mathcal{C}$. His analysis considers the general nonlinear nonconvex programming problem, though subject only to inequality constraints[19]. For these problems, the dual as defined by Falk is not necessarily unique, the relationships between $\boldsymbol{\lambda}^\dagger$ and $\boldsymbol{x}^\dagger$ being point-to-set relationships in the general case. However, Falk shows that in the case of strictly convex programming problems, his dual is again uniquely defined. It is this specialisation that is used in the formation of dual subproblems for structural optimisation in the algorithms discussed in Section 2.2.3. The subproblems defined by the approximations used in those algorithms are strictly convex, and Falk's formulation allows the domain $\mathcal{C}$ to be identified with the bound constraints. This ensures firstly that the bounds do not increase the dimensionality of the dual, and secondly that $\mathcal{C}$ can be chosen so that the subproblems are always properly defined within $\mathcal{C}$.

We note that when the restriction on the domain given by $\mathcal{C}$ is introduced, equation (2.38) is not valid in general for defining the minimiser of the Lagrangian with respect to $\boldsymbol{x}$ within $\mathcal{C}$. The Lagrangian cannot be assumed to possess a turning point within $\mathcal{C}$ for all $\boldsymbol{\lambda}^\dagger$. Therefore, the primal-dual relationships are instead defined by condition (2.36). Since $\mathcal{C}$ is closed, the Lagrangian of a strictly convex problem always has a unique minimum on $\mathcal{C}$, for any feasible $\boldsymbol{\lambda}^\dagger$, although it may not meet the definition of a stationary point.

Lastly, although Falk highlighted the applicability of his dual for strictly convex problems, it is not necessarily the case that a problem must be strictly convex in order for a unique Falk dual to be defined. It is standard practice to generate strictly convex subproblems for structural optimisation problems and then to solve these using Falk's definition of the dual. The work presented in Chapters 5 and 6, however, considers instances in which nonconvex subproblems arise from the minimum compliance and minimum weight problems respectively. We show that the dual method may still be used to solve these problems, because Falk's dual is still uniquely defined for them. A brief description of Falk's dual is provided below as it appears in [2], which the reader is urged to refer to.

[19] Falk's work is apparently extensible to include equality constraint. In the current document, however, only inequality constrained problems are addressed, so equality constraints are omitted in what follows.

2.3.1 The Falk Dual

Falk [2] considers a mathematical programming problem with the form[20]

$$\begin{aligned} &\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) \\ &\text{subject to } f_j(\boldsymbol{x}) \geq 0 && j = 1, 2, \cdots, m, \\ &\qquad x_i \in \mathcal{C} && i = 1, 2, \cdots, n, \\ &\text{where} \\ &\qquad \mathcal{C} \subset \mathcal{R}^n, \\ &\qquad f_0 : \mathcal{R}^n \to \mathcal{R}^1, \\ &\qquad f_j : \mathcal{R}^n \to \mathcal{R}^1, && j = 1, 2, \cdots, m. \end{aligned} \tag{2.39}$$

A Lagrangian is defined over the space $\mathcal{R}^n \times \mathcal{R}^m$ as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f_0(\boldsymbol{x}) - \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}) \,. \tag{2.40}$$

Falk defines an auxiliary function γ by

$$\begin{aligned} \gamma(\boldsymbol{\lambda}) &= \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{C} \\ \text{and} \quad & \boldsymbol{\lambda} \geq 0 \,, \end{aligned} \tag{2.41}$$

and $\mathcal{D}[\gamma]$ – the domain of γ – is given by all $\boldsymbol{\lambda}$ for which $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ possesses a finite minimum with respect to $\boldsymbol{x}$, where $\boldsymbol{x} \in \mathcal{C}$. For a given $\boldsymbol{\lambda}$, the minimiser $\boldsymbol{x}$ need not in general be unique. The points $\boldsymbol{x}$ at which $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is minimised for given $\boldsymbol{\lambda}$ form the set $\mathcal{X}(\boldsymbol{\lambda})$. The auxiliary function (2.41) is dual to the primal problem (2.39) in the sense that $\gamma(\boldsymbol{\lambda}) \leq f_0(\boldsymbol{x})$ for all feasible points $\boldsymbol{\lambda}$ and $\boldsymbol{x}$ (by Theorem 4 of [2]).

The optimum of the dual is not guaranteed, in general, to match the optimum of the primal problem. However, for strictly convex problems (i.e. problems for which f_0 is strictly convex and all f_j are concave, given the above definitions of the programming problem and Lagrangian), Falk showed that their solution could be achieved using the dual because the following set of properties can be proved[21]:

- Theorem 7: The domain over which the dual is defined ($\mathcal{D}[\gamma]$) is an open set relative to the interior of the positive orthant in the space of the Lagrange multipliers $(\mathcal{R}^m)^+$.
- Theorem 8: The domain $\mathcal{D}[\gamma]$ is convex (which makes γ concave by Theorem 1).
- Theorem 9: $\mathcal{X}$ is a continuous function on $\mathcal{D}[\gamma]$.

[20] Note that Falk uses the positive-null ($\geq$) form to represent the constraints. In presenting his work here, we have followed suit to maintain consistency with his exposition.

[21] The theorem numbers are those listed in Falk [2].

- Theorem 10: γ is differentiable throughout the interior of $\mathcal{D}[\gamma]$ and the right-hand partial derivatives $\partial\gamma/\partial\lambda_j^+$ exist at $\lambda_j = 0$ for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^\circ$ if $\boldsymbol{\lambda}^\circ \in \mathcal{D}[\gamma]$ and $\lambda_j^\circ = 0$.
- Theorem 11: If γ is maximised over $\mathcal{D}[\gamma]$ at $\boldsymbol{\lambda}^*$, then $\boldsymbol{x}^* = \mathcal{X}(\boldsymbol{\lambda}^*)$ is the solution to (2.39) and $\gamma(\boldsymbol{\lambda}^*) = f_0(\boldsymbol{x}^*)$.

2.3.2 Nonconvexity and the dual

Although strict convexity of the primal problem is assumed in the formulation of the proofs of the above Theorems 7 through 11, we maintain that the proofs themselves are applicable to a broader class of problems. We have used this observation as the basis of two papers concerning the use of nonconvex function approximations in sequential approximate optimisation infrastructures [35, 36], which are presented in Chapters 5 and 6. In particular, we have argued that certain nonconvex forms that can arise in the consideration of the weight minimisation problem and the minimum compliance problem are still consistent with the proofs of the above theorems. Hence, we have demonstrated that these nonconvex approximate subproblems can be solved uniquely using Falk's dual formulation.

In [2], the proofs of Theorems 7 through 11 are presented specifically for strictly convex programming problems. However, the proofs themselves depend primarily on the following attributes of a problem for their validity:

Attribute 1: *The Lagrangian $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^\dagger)$ has a unique minimim in terms of $\boldsymbol{x}$ over the set $\mathcal{C}$, for any arbitrarily chosen $\boldsymbol{\lambda}^\dagger$ in $\mathcal{D}[\gamma]$.*

Attribute 2: *$\mathcal{D}[\gamma]$ is convex.*

Attribute 3: *All f_j are continuous, $j = 0, 1, 2, \cdots, m$.*

Strictly convex continuous programming problems obviously possess these attributes, and this result has encouraged the successful development of sequential approximate optimisation algorithms based on the iterative solution of strictly convex subproblems using Falk's dual approach. Note, however, that there are nonconvex problems that also possess the above attributes. We have made the assertion that these continuous nonconvex programming problems, for which Attributes 1 through 3 hold, are also amenable to solution via the same dual approach, and can therefore also be used as approximate subproblems in an SAO infrastructure, particularly for structural optimisation.

In the field of structural optimisation, a problem's objective function and constraints are most often continuous functions or are approximated as such. Moreover, the domain $\mathcal{C}$ is commonly defined only by the upper and lower bound constraints on the design variables. In the case of structural optimisation then, the problems are simplified by the fact that Attribute 3 holds and that $\mathcal{C}$ is compact. Under these circumstances, the observation can be made immediately (see [2]) that $\mathcal{D}[\gamma]$ corresponds to $(\mathcal{R}^m)^+$. Therefore, $\mathcal{D}[\gamma]$ is automatically convex and we need only concern ourselves with whether or not Attribute 1 holds.

2.3.3 Separability

The requirement that the Lagrangian always has a unique minimum with respect to the set of primal (design) variables $\boldsymbol{x}$ for any positive $\boldsymbol{\lambda}$ is very restrictive and very difficult to verify in general. However, if the SAO subproblems are defined in terms of separable functions, the requirement that Attribute 1 holds can be checked more easily. For separable functions, the Lagrangian can be expressed as a sum of n terms,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathcal{L}_1(x_1, \boldsymbol{\lambda}) + \mathcal{L}_2(x_2, \boldsymbol{\lambda}) + \ldots + \mathcal{L}_i(x_i, \boldsymbol{\lambda}) + \ldots + \mathcal{L}_n(x_n, \boldsymbol{\lambda}), \tag{2.42}$$

each being a function of only one primal variable x_i. If the domain $\mathcal{C}$ represents only the bound constraints on $\boldsymbol{x}$, then it can be defined separably as well:

$$\mathcal{C} = \{\boldsymbol{x} \mid \quad \check{x} \leq x_i \leq \hat{x} \quad \forall \quad i\}\,. \tag{2.43}$$

Minimising $\mathcal{L}$ with respect to the n design variables reduces to performing n one-dimensional minimisations

$$\left.\begin{array}{rl} \min\limits_{\boldsymbol{x}} & \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{C} \end{array}\right\} = \begin{array}{l} \displaystyle\sum_{i=1}^{n} \left(\min_{x_i} \mathcal{L}_i(x_i, \boldsymbol{\lambda}) \right) \\ \text{subject to } \check{x} \leq x_i \leq \hat{x}\,. \end{array} \tag{2.44}$$

If all n minima exist in $\mathcal{C}$ and are both finite and unique for every conceivable vector of positive multipliers $\boldsymbol{\lambda}$, then Falk provides us with the assurance that the dual can be defined uniquely and, moreover, that it is concave and continuous.

2.4 Closure

The material presented in this chapter has served to briefly introduce three topics: topology optimisation, sequential approximate optimisation and duality. It is the *combination* of these three, namely the application of the dual within an SAO infrastructure applied to topology problems, that has given rise, rather organically, to the work presented in the forthcomming chapters. The work presented neither assumes nor requires an expert knowledge of the material distribution problem, since the thesis is concerned primarily with the exploration of some facets of dual-based SAO, with material distribution problems providing challenging and important examples to which it can be applied. Having said this, it is the material distribution problems themselves that have suggested which facets of dual SAO might be investigated fruitfully. Thus, research into the discrete dual (Chapter 4), dual separability (also Chapter 4), the use of nonconvex approximation functions (Chapters 5, 6 and 7) and the potential of solving large problems (Chapters 4 and 9) was driven by the requirements of particular topology problems. Inevitably some feedback has occurred, and certain more application-specific topics are also addressed, such as sensitivity filtering in Chapter 3 and stress relaxation in Chapter 9.

Chapter 3

Sensitivity filtering in topology optimisation

The exposition in this chapter is a crystalisation of a number of ideas that have germinated from a collaboration with Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa, and Dr L.F.P. Etman of the Department of Mechanical Engineering at the Eindhoven University of Technology, Eindhoven, the Netherlands. It is intended for (possible) submission with these co-authors, and has thus been prepared in the format of an article.

3.1 Abstract

Ever since its introduction into topology optimisation, the so-called 'mesh independence filter' of Sigmund has been considered a heuristic tampering of the objective function sensitivities to achieve designs that are not only mesh independent, but also free from checkerboarding. Mesh dependence in particular stems from the fact that the underlying continuum topology problem lacks solutions, unless its solution space is restricted in some way. The filter was introduced as such a restriction method, though in the past it has been criticised as lacking mathematical justification, and there is as yet no proof that the use of the filter solves the existence problem. There is therefore some uncertainty about how the use of the filter should be interpreted, although there is a perception that by using the filter one actually solves a different problem closely related to the originally stated topology problem. Despite the uncertain basis for the filter, it has nevertheless seen widespread use in the topology optimisation community because, in practice, it does produce largely mesh-independent and checkerboard-free designs, and that very efficiently. Years of collective experience therefore testify to its utility.

In this chapter we revisit the mesh independence filter of Sigmund. Instead of being purely heuristic, we argue that, in the context of sequential approximate optimisation, the filtered sensitivities can be interpreted as defining the exact gradients of a modified approximate primal subproblem. These subproblems are not only separable in the design variables, but also (conditionally) strictly convex. Hence, the subproblems that the filter gives rise to possess unique solutions. In this sense, the filter need not be considered mathematically unsound.

While we provide an interpretation of the definition of the filter, we do not provide an explanation

of its action. That is, why using subproblems of the form defined by the filter would reduce mesh dependence and checkerboarding is a question that we cannot satisfactorily answer. However, we believe that the interpretation of the filter given herein in the context of SAO will be a fruitful starting point for explaining its efficacy in future, and we offer a few initial thoughts on the subject. In contrast, we argue that the interpretation of the filter as being associated with a different underlying problem entirely is invalid.

Lastly, viewing the filter as giving rise to subproblems in the SAO paradigm provides the basis for an analysis of more specific questions regarding the nature of the filtered optimisation problems. For example: can algorithms using the filter converge? If so, what are the characteristics of a point to which convergence occurs? Are such points solutions to the originally stated problem? And so on. We offer a few initial thoughts on these matters as well. Our hope is that, through consideration of questions like these in the context of SAO, a proof of existence for the solutions of the filtered topology problem may be devised in the future.

3.2 Introduction

In topology optimisation we seek the distribution of material within a pre-defined spacial domain such that said distribution is optimal for a given structural objective function subject to any number of linear and/or nonlinear inequality constraints. This solid-void optimisation problem is very difficult from a mathematical point of view: the continuum problem suffers from multimodality and non-existence of the solution. When the field describing the material distribution is discretised using the finite element method, the associated optimisation problem is inherently NP-complete and of (very) high dimensionality, while the non-existence problem manifests itself by making the solutions qualitatively dependent on the mesh discretisation used. Additionally, the optimisation problem may exhibit an artificial numerical stiffening phenomenon known as checkerboarding[1].

Notable effort has previously been directed towards showing that a solution to the (continuum) topology optimisation problem exists if certain methods are employed that either extend or restrict the design space (refer to Section 2.1). Probably the most widely used restriction method is due to Sigmund [14, 15], who proposed a filtering method to overcome non-existence of the solution and checkerboarding. Borrowed from digital image processing, and known as his so-called ‘mesh independence filter’, this filter is considered heuristic. Even so, it is extremely popular in topology optimisation, the reasons being that it is easy to implement, produces very little extra computational burden, and works very well, being effective at decreasing mesh dependency and the appearance of checkerboarding. In fact, it seems to satisfy most of the desirable characteristics of numerical filters for topology optimisation (according to Sigmund [37], it is desired that these methods do not introduce additional constraints, that they are effective, simple, computationally efficient, easily implemented, and robust).

The main objection to the filter is that it is not considered mathematically sound. The reason for this is that the sensitivities of the problem are tampered with, such that the information used in the solution procedure no longer corresponds to the problem that is supposed to be solved. This,

[1]The latter applies when fully integrated low-order quadrilateral finite element discretisations are used, in combination with elemental design variables (which is standard practice).

in turn, raises the question of what problem is actually being solved when the filter is used. In other words, assuming that the filter affects only the objective function and that the constraints are not modified[2], what objective function actually possesses the filtered sensitivities as its true gradients? What is the (spatially discretised relaxed continuous) optimisation problem to which it belongs? And then, what is the continuum form of the optimisation problem from which this derives? Ultimately, what is the variational problem that gives rise to the filtered objective and how is it related to compliance?

Regarding his filter, Sigmund [37] has quite recently remarked that "as the sensitivities are modified heuristically, it is probably impossible to figure out what objective function is actually being minimised, but generally, it may be stated that the filtered sensitivities correspond to the sensitivities of a smoothed version of the original objective function." In fact, filtering of the sensitivities is considered "dangerous when linesearch techniques are used". The development of density filtering techniques (e.g. see Bruns and Tortorelli [38] and Bruns [39]), which *per se* introduce a grey transition region between black and white material, was largely motivated by the desire to present a mathematically sound filtering technique as an alternative to sensitivity filtering.

As the quote above implies, it is not clear just how the filter achieves its objectives of mesh independence and the suppression of checkerboarding. It is evident that mesh independence is effected because a minimum-length scale, which is associated with the filter radius and is thus largely independent of the mesh refinement, is introduced into the problem. However, a thorough explanation of mesh independence is actually required to demonstrate that the filter solves the existence problem in the original continuum form of the topology problem. While we are not in a position to do this, an attempt is nevertheless made to describe and elucidate the working of the filter, purely because most of the minimum compliance topology results produced for this document have relied upon the filter for the regularisation of the problems.

The prevalent opinion is that the use of the filter to regularise a given topology problem actually causes a different problem to be solved, though one that is closely related to the originally stated topology problem. The nature of this relationship is as yet unclear. In describing the way the filter works in the current chapter, it is this interpretation, and the perception of the filter as an unsubstantiated heuristic, that will be addressed.

We herein assume that the filter is used in sequential approximate optimisation algorithms in which dual principles are used to solve the surrogate subproblems. This seems reasonable: since the dimensionality of the topology problem is very high and few constraints are present (if local stress and/or displacement constraints are absent), primal methods are hardly, if ever, used. Methods popular in topology optimisation are dual sequential approximate optimisation (SAO) methods, of which the method of moving asymptotes (MMA) proposed by Svanberg [3, 32] is probably the best known and most frequently used, and optimality criterion (OC) methods, which have been shown to be closely related to dual SAO procedures. We demonstrate that, in the context of SAO, the filtered sensitivities define the exact gradients of a modified approximate primal subproblem.

The crux of the work presented in this chapter is, however, an investigation into whether the filter can be interpreted as giving rise to a different objective function. We argue that this is extremely unlikely in general and we present various numerical examples for which this interpretation cannot

[2]We here consider the classical minimum compliance problem, in which only the sensitivities of the compliance objective are filtered.

be considered valid.

The chapter is constructed as follows: in Section 3.3, the standard minimum compliance topology optimisation problem is discussed briefly. This is followed in Section 3.4 by notes on both the OC updates and the dual SAO algorithms used to approximately solve the problem. In Section 3.5 the mesh independence filter of Sigmund is introduced, and we elaborate on how the filter may be interpreted in SAO algorithms. In Section 3.6 we consider the question of whether an alternative 'smoothed' objective function exists from which the filtered sensitivities derive. Specific, illustrative numerical examples are presented in Section 3.7, and our observations are summed up by our concluding remarks in Section 3.8.

3.3 Minimum compliance topology optimisation

The topology optimisation problem, perhaps more properly referred to as the material distribution problem, has been discussed in Section 2.1, in which both the minimum compliance and minimum weight problems were introduced. In discussing the use of the filter, however, it will be assumed that the objective function on which the filter operates is compliance. In Chapter 9, where spatially continuous weight minimisation and minimum compliance problems are solved (as opposed to a discrete truss-type problem), no filter is used. On the other hand, the filter is incorporated into the minimum compliance problems presented in the rest of the document.

For convenience, the relaxed continuous form of the compliance problem with a single volume constraint (2.9) is repeated here.

Relaxed continuous compliance problem P_C

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f_0(\boldsymbol{x}) = \boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} = \sum_{i=1}^{n} (x_i)^p \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \\ \text{subject to } & f_1(\boldsymbol{x}) = \frac{1}{\nu_0} \sum_{i=1}^{n} \nu_i x_i - \bar{\nu} \leq 0 \,, \\ & \boldsymbol{K}\boldsymbol{q} = \boldsymbol{r} \,, \\ & 0 \leq \check{x} \leq x_i \leq \hat{x} = 1 \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{3.1}$$

Note that although $[0, 1]$ solutions are sought, the above definition reflects the fact that it is necessary to introduce a non-zero lower bound $\check{x}$ on the design variables in order to avoid numerical ill-conditioning in the solution of the finite element equations.

3.4 The common OC design update for topology optimisation

We depart from a generally applicable optimality criterion statement used to update the topology design from $\boldsymbol{x}^{\{k\}}$ to $\boldsymbol{x}^{\{k+1\}}$ (e.g. see [40, 41]):

$$x_i^{\{k+1\}}(\boldsymbol{\lambda}) = \begin{cases} x_i^{\{k\}} \beta_i^{\eta}(\boldsymbol{\lambda}) & \text{if} \quad \check{x} < x_i^{\{k\}} \beta_i^{\eta}(\boldsymbol{\lambda}) < \hat{x}, \\ \check{x} & \text{if} \quad x_i^{\{k\}} \beta_i^{\eta}(\boldsymbol{\lambda}) \leq \check{x}, \\ \hat{x} & \text{if} \quad x_i^{\{k\}} \beta_i^{\eta}(\boldsymbol{\lambda}) \geq \hat{x}. \end{cases} \tag{3.2}$$

Here, $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$ represents the n primal design variables, while $\check{x}$ and $\hat{x}$ denote, respectively, the lower and upper bounds on x_i (which are the same for all x_i). The vector $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_m]^T$ represents the m dual variables in the general case. Superscript $k \geq 0$ represents the iteration number in the optimisation procedure. The β_i are found from the optimality conditions (2.16), as well as from a consideration of the form of the structural responses.

For the minimum compliance objective subject to a single linear constraint on the material resource, we have

$$\beta_i(\lambda) = -\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} \Big/ \lambda \left(\frac{\partial f_1}{\partial x_i}\right)^{\{k\}}, \tag{3.3}$$

for all elements $i = 1, 2, \cdots, n$. The update (3.2) has the same form as the OC update (2.26) described in Section 2.2.2. However, the sensitivities of the constraint appear in the denominator of (3.3), whereas they occur in the numerator of (2.26). The form discussed in Section 2.2.2 derives from the weight minimisation problem, in which the reciprocal behaviour of the structural responses affect the constraint functions, while the objective function is linear. Here, however, it is the objective that exhibits the reciprocal-like form, and the volume constraint is linear in the design variables. The optimality conditions for $x_i^{\{k+1\}}$ derived from the stationary condition of the Lagrangian (2.16) for the compliance problem therefore express the relationship between the gradients of f_0 and f_1 inversely with respect to the weight minimisation problem.

In (3.2), η is a heuristic numerical damping factor first introduced by Bendsøe [42] for the topology optimisation problem. Its function was discussed in Section 2.2.2; for the compliance problem a value of $\eta = 0.5$ is typically used.

Previously, Groenwold and Etman [43] have shown that (3.2) can be derived from a sequential approximate optimisation algorithm based on duality, just as Fleury noted the equivalence of SAO and OC in the definition of (2.22). The update (3.2) corresponds exactly to the SAO updating scheme one obtains if the primal objective function is approximated by

$$\begin{aligned}\tilde{f}_0^{\{k\}}(\boldsymbol{x}) &= f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(y_i - y_i^{\{k\}}\right) \left(\frac{\partial f_0}{\partial y_i}\right)^{\{k\}} \\ &= f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(x_i^q - (x_i^q)^{\{k\}}\right) \left(\frac{x_i^{1-q}}{q}\right)^{\{k\}} \left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}},\end{aligned} \tag{3.4}$$

and the primal approximate constraint is a linear function in terms of x_i, given by the expansion (2.14). The objective approximation (3.4) is a linear (first-order) truncated Taylor series expansion in terms of the *exponential* intervening variables $y_i = x_i^q$, first suggested by Fadel *et al.* [44]. The condition $q < 0$ is adequate to ensure that the approximate primal problem is strictly convex. With the objective and constraint functions approximated in this way, the Lagrangian of the subproblem is separable in the x_i. Applying the stationary conditions (2.16) to the Lagrangian yields

$$x_i = \left[\left(x_i^k\right)^{1-q} \beta_i\right]^{\frac{1}{1-q}} \qquad \forall \quad i = 1, 2, \cdots, n,$$

with the β_i given by (3.3). This is equivalent to (3.2) with

$$\eta = \frac{1}{1-q},$$

and with the bounds on x_i respected. Of particular interest is the case when $\eta = 0.5$, which results in

$$\tilde{f}_0^{\{k\}}(\boldsymbol{x}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(x_i - x_i^{\{k\}}\right) \left(\frac{x_i^{\{k\}}}{x_i}\right) \left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}}, \tag{3.5}$$

a linear Taylor series expansion in terms of the familiar *reciprocal* intermediate variables so popular in structural optimisation. The function (3.5) is obtained by setting $q = -1$ in (3.4). Incidentally, both the heuristic OC method proposed by Bendsøe and the MMA algorithm utilise reciprocal intermediate variables.

To derive (3.2) from (3.4) in a general SAO setting, it is merely required that the Falk dual [2] exists. In turn, the Falk dual may be shown to exist for an arbitrary (approximate) primal subproblem that is strictly convex and separable, on condition that the design variables represent a closed and bounded set (this being the case in the topology optimisation problem). For details, the reader is referred to References [2, 28, 41]. Accordingly, (3.2) may be understood to be a very general statement in topology optimisation. Even the popular method of moving asymptotes (MMA) may be generalised to a form similar to (3.2).

It is in the context of the OC update (3.2) that Sigmund's filter is normally considered. The filter modifies the design updates by changing the sensitivities of the objective function that enter into (3.3). The question arises: what exactly does it mean when the gradients of the stated objective are not used in the design update, being instead replaced by the filtered sensitivities? Recognising the equivalence between OC and SAO allows us to suggest an interpretation.

3.5 Sigmund's mesh independence filter

We now turn our attention to Sigmund's very well-known sensitivity filter [14, 15]. For an arbitrary objective function f_0, it is expressed as

$$\left(\widehat{\frac{\partial f_0}{\partial x_i}}\right)^{\{k\}} = \frac{\displaystyle\sum_{j=1}^{n} w_{ij} x_j^{\{k\}} \left(\frac{\partial f_0}{\partial x_j}\right)^{\{k\}}}{x_i^{\{k\}} \displaystyle\sum_{j=1}^{n} w_{ij}}, \quad i = 1, 2, \cdots, n. \tag{3.6}$$

Apparently, the sensitivities of the objective function are modified, and the elemental sensitivities

$$\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}}$$

are replaced by the 'filtered sensitivities'

$$\left(\widehat{\frac{\partial f_0}{\partial x_i}}\right)^{\{k\}},$$

which have become a function of the sensitivities and densities of a subset of the total number of elements n (most $w_{ij} = 0$). It is this very replacement of the sensitivities that is seen as the reason why the mesh independence filter is suspicious from a mathematical point of view, since the filtered sensitivities are then used in the update scheme for the β_i in (3.2). A satisfying explanation of what it means physically to insert the filtered sensitivities into (3.2) has been lacking

Of course, the effect of this insertion is recognised as advantageous. Firstly, checkerboarding is suppressed; at the solution, the design variables x_i are 'smoothed' in some sense over the subset of neighbouring elements defined by the convolution operator w_{ij}, being zero for elements 'far away' but non-zero for a number of elements in the 'close vicinity' of element i, with (typically) $w_{ij} = 1$ for $j = i$ and $0 < w_{ij} < 1$ otherwise. However, this 'smoothing' is not accomplished in a direct way, as with density filtering [38]. Instead, modifying the gradients of the objective somehow naturally gives rise to a sequence of design updates that gravitate away from solutions that exhibit checkerboarding. Secondly, the same process also produces mesh independence.

3.5.1 Interpreting Sigmund's mesh independence filter

The filter is seen as heuristic because it seems to lack a formal mathematical rationale for both its particular form and its function. If the solution of the topology problem is approached from the point of view of the OC methods, it is indeed difficult to find a formal interpretation of the filter. However, when the solution of the problem is viewed from the (equivalent) perspective of dual-SAO, the mechanism of the filter naturally acquires a more significant interpretation. It is straightforward to show that the use of the filter modifies the form of the approximate subproblems that are used in the optimisation procedure.

The problem of interpreting the mesh independence filter of Sigmund becomes tractable if the update scheme is understood to be the result of a Falk-like dual formulation. For the compliance problem considered, this perspective leads us to conclude that the update scheme containing the filtered sensitivities must derive from the primal approximation

$$\tilde{f}_0^{\{k\}}(\boldsymbol{x}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(x_i^p - (x_i^{\{k\}})^p\right) \left(\frac{x_i^{1-p}}{p}\right)^{\{k\}} \left(\widehat{\frac{\partial f_0}{\partial x_i}}\right)^{\{k\}}, \tag{3.7}$$

in exactly the same way that the combination of (3.2) and (3.3) derives from (3.4). The filtered sensitivities are constants evaluated at $\boldsymbol{x}^{\{k\}}$. They are strictly negative because the gradients of the partial derivatives of the compliance objective are all negative. Hence an easily identifiable and strictly convex primal approximate objective, defined by (3.7), is minimised whenever the filtered sensitivities are used. When constructing the Falk dual, (3.7) poses no problems whatsoever, since the modified sensitivities do not depend on the elements of $\boldsymbol{x}$ in the Lagrangian of the approximate subproblem $\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda})$.

At the point $\boldsymbol{x}^{\{k\}}$, primal approximations (3.4) and (3.7) have identical function values. However, their gradients at this point differ. The effect of the filtered sensitivities is that the SAO subproblems constructed using the filtered sensitivities are different from those that would have been constructed using the actual gradients, though each is still strictly convex, possessing a unique solution. Thus, the use of these subproblems produces a different sequence of SAO iterates $\boldsymbol{x}^{\{k+1\}}$ than would have been obtained without the filter. In this interpretation it is still the original compliance problem that

is solved, and the filter determines the form of the subproblems used in the SAO. It is the choice of the particular form of the SAO subproblems (defined by the filter) that effects mesh independence and overcomes checkerboarding.

The interpretation of Sigmund's mesh independence filter proffered here therefore changes the question of explaining the functioning of the filter from "what problem is actually being solved?" to "why should the subproblems defined by the filter be effective?" Why and how such subproblems would produce a sequence of solutions that converge (hopefully) on a design that is checkerboard-free, in a qualitatively mesh-independent way, remains to be explained. Hence, the fundamental questions regarding the filter remain unanswered. However, it is hoped that interpreting the form of the filter in terms of sequential approximate optimisation at least provides a firm basis from which an investigation of its function may be advanced.

Of course, (3.7) is no longer a genuine Taylor series expansion. The approximate subproblem derived thereby is not first-order accurate (something that is usually demanded if a convergence proof for an SAO algorithm is to be advanced), so the question can be asked whether using such an approximation is a particularly sensible choice from the perspective of SAO[3]. Nevertheless, the meaning of the filtered sensitivities, at least, is clear. Again, of particular interest is the case when $\eta = 0.5$, which simply corresponds to

$$\tilde{f}_0^{\{k\}}(\boldsymbol{x}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(x_i - x_i^{\{k\}}\right) \left(\frac{x_i^{\{k\}}}{x_i}\right) \left(\widehat{\frac{\partial f_0}{\partial x_i}}\right)^{\{k\}} . \tag{3.8}$$

Finally, it is instructive and semantically correct to refer to Sigmund's original method for filtering as 'filtering through the construction of a modified approximate primal subproblem', or possibly *'approximation-based filtering'* for short. Since the filter defines a particular form for the SAO subproblems, there is no reason to suspect that the filter itself is in any sense fundamental. This is to say that other subproblem forms, defined by other sensitivity filters, may equally accomplish mesh independence and the suppression of checkerboarding. Indeed, as a form of approximation-based filtering, the filter of Sigmund exhibits similarities with the so-called grey-scale filter previously developed by Groenwold and Etman [45]. We point out that the insights developed from an analysis of Sigmund's filter, in the context of SAO, may be used to develop alternative approximation-based filtering methods to (3.7).

3.5.2 A two-dimensional graphic example

Consider the two-dimensional programming problem

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f_0(x_1, x_2) = \frac{a_1}{x_1^2} + \frac{a_2}{x_2^2} \\ \text{subject to } & f_1(x_1, x_2) = x_1 + x_2 - 0.8 \leq 0\,, \\ & x_1, x_2 > 0\,, \end{aligned} \tag{3.9}$$

[3]Experience of course suggests that the modified primal approximation that stems from the filter is indeed sensible. The optimality of the solution, from the point of view of the original unfiltered compliance problem, is another matter altogether.

where f_0 is a monotonically decreasing inverse quadratic function. This problem is depicted in Figure 3.1(a), for $a_1 = 3$ and $a_2 = 1$.

Two linear inverse approximations to the original function are graphed in Figures 3.1(b) and 3.1(c). Both approximations use reciprocal intermediate variables. Figure 3.1(b) was constructed using (3.5), i.e. the original unfiltered sensitivities were used, whereas Figure 3.1(c) was constructed using (3.8), i.e. the filtered sensitivities were used. We have used the convolution operator

$$\begin{aligned} w_{1j} &= [1.0\ 0.4]\,, \\ w_{2j} &= [0.4\ 1.0]\,, \end{aligned}$$

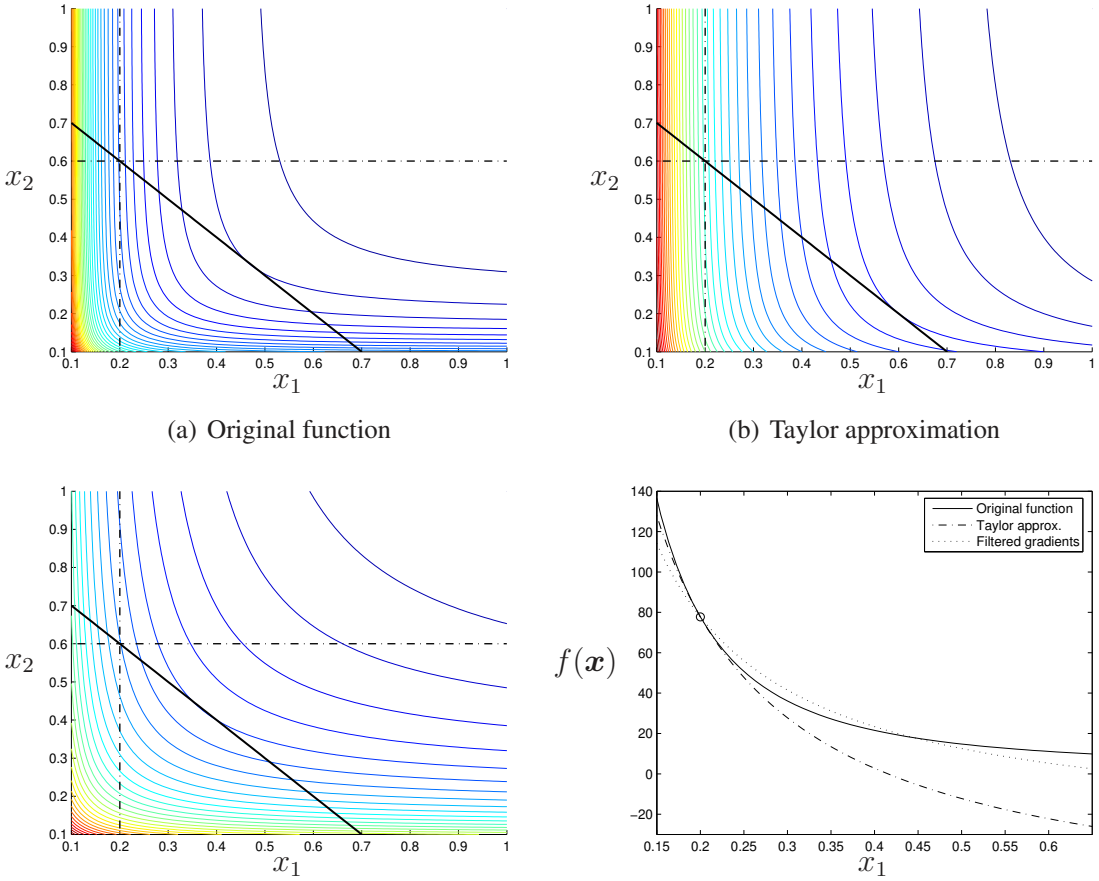
and both approximations are constructed around the point $\boldsymbol{x}^{\{k\}} = [0.2\ 0.6]$. Figure 3.1(d) depicts a comparison of each function on the subspace $(x_1, (x_2 = 0.6))$. The filtered approximation is quite different to the Taylor approximation and its gradient does not match the true gradient at the point of approximation. Naturally, the minima of the two different approximate subproblems with respect to the linear constraint f_1 are found at different positions. The approximate primal objective function defined by employing the filter has the same monotonically decreasing form as the unfiltered Taylor approximation, but the modified gradients change the position at which the approximate optimum is located.

If the approximate subproblem derived from the Taylor expansion were to be constructed at the optimum $\boldsymbol{x}^*$ of (3.9), it would have the same optimum as (3.9), namely the point of approximation $\boldsymbol{x}^*$. The same is not true of the filtered approximation. The filtered approximation constructed at $\boldsymbol{x}^*$ would have a minimum elsewhere, this being a consequence of the fact that the filtered approximation is not first-order accurate. Hence, when the filter is used in SAO (or in an OC algorithm for that matter), the solution that the optimisation process identifies will characteristically *not* be a KKT point of the stated problem. This last presumes that convergence to a solution will occur at all, which is unclear generally (for the 2D problem above, however, convergence can be demonstrated numerically).

3.6 The existence of a smoothed problem

One advantage of the interpretation of the filter given in Section 3.5.1, as part of the generation of SAO subproblems, is that, in this view, the originally stated topology problem is the problem that is addressed, and not some other related problem. It is still interesting to ponder whether this alternative view is, in fact, possible. Hence, we here consider the question of whether, for a given material distribution problem P_D, it is possible to view the filtered gradients as being the actual gradients of another (true) objective function $f_t(\boldsymbol{x})$, which is actually being minimised when the filter is employed. Note: we ask only whether a different objective function can exist for the relaxed continuous form of the optimisation problem. However, if the answer is negative, it is difficult to see how the existence of a different continuum[4] form can be espoused. This question may be viewed in two ways:

[4]Spatially continuous.



(a) Original function

(b) Taylor approximation

(c) Filter approximation

(d) Function approximations at $x_2 = 0.6$

Figure 3.1: The effect of filtering and approximation on the minimum of a simple 2D function.

Case 1: The true objective function f_t has the same objective function values as the evaluated objective function $f_0(\boldsymbol{x})$ at all point $\boldsymbol{x}^{\{k\}}, k = 1, 2, \cdots$ in the sequence of SAO iterates.

Case 2: The true objective function is entirely different from the stated compliance objective in P_D, in which case it is never actually evaluated throughout the optimisation process.

In both cases, the sensitivities of the true objective function are derived from the evaluation of P_D. They are constructed from the sensitivities of $f_0(\boldsymbol{x})$ via the application of (3.6). If we expect Case 1 to be valid, then we require that two functions, f_0 and f_t, exist that have the same function values everywhere, but different gradient fields. In other words, two identical functions must have different gradients. This is so because the points at which the objective is evaluated are essentially arbitrary. Case 1 is therefore obviously not possible because of the uniqueness of partial derivatives. That is to say: a continuous and differentiable function has a unique gradient field because its partial derivatives are unique.

Case 2 is more interesting. In case 2, the only knowledge that is gleaned about the supposed function f_t is its gradient field, which is written (and evaluated) in terms of the sensitivities of

f_0. In theory it is possible to recover the function f_t from its gradient, which is a vector field. The idea can be expressed using, for example, the fundamental theorem of line integrals (see for instance [46]), which derives from the fundamental theorem of calculus, and holds that

$$\int_S \boldsymbol{F} \cdot ds = f(\boldsymbol{x}_1) - f(\boldsymbol{x}_0). \tag{3.10}$$

Here S is a simple parameterised curve joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_0$, and ds is an infinitesimal line element on S. The gradient field of a given function f is represented by $\boldsymbol{F}$. It is a standard result from the calculus of vector-valued functions that, if $\boldsymbol{F}$ is the gradient of a scalar potential function f, i.e. $\boldsymbol{F} = \nabla f$, then equation (3.10) holds and the right-hand side of (3.10) is independent of the path S chosen between $\boldsymbol{x}_1$ and $\boldsymbol{x}_0$[5]. In other words, $\boldsymbol{F}$ is conservative. Thus, given a starting point $\boldsymbol{x}_0$ and an associated function value $f_t(\boldsymbol{x}_0)$, the 'true' objective function $f_t(\boldsymbol{x})$ can be recovered from the vector field $\boldsymbol{F}_t$ (given by the filtered sensitivities) as

$$f_t(\boldsymbol{x}) = \int_S \boldsymbol{F}_t \cdot ds + f_t(\boldsymbol{x}_0), \tag{3.11}$$

where $\boldsymbol{x}$ is now arbitrary and the curve S is chosen appropriately. If $f_t(\boldsymbol{x}_0)$ is not known, it can be chosen arbitrarily, in which case $f_t(\boldsymbol{x})$ is recovered relative to $f_t(\boldsymbol{x}_0)$. In two and three dimensions it is possible to test directly whether the vector field $\boldsymbol{F}_t$ is in fact the gradient of a scalar function f_t. Another standard theorem from calculus holds that if $\boldsymbol{F} = \nabla f$, then it must be true that

$$\operatorname{curl}(\nabla f) = \boldsymbol{0}. \tag{3.12}$$

Hence, if we have a vector field $\boldsymbol{F}_t$ in three dimensions, we can test whether an associated scalar potential function f_t exists by testing whether $\operatorname{curl}(\boldsymbol{F}_t) = \boldsymbol{0}$, because only rotation-free vector fields derive from scalar potential functions. Again, refer to [46], for example.

For a three-variable problem, the condition that $\operatorname{curl}(\boldsymbol{F}) = \boldsymbol{0}$ is equivalent to the conditions

$$\begin{aligned}
\frac{\partial^2 f}{\partial x_1 \partial x_2} - \frac{\partial^2 f}{\partial x_2 \partial x_1} &= 0, \\
\frac{\partial^2 f}{\partial x_1 \partial x_3} - \frac{\partial^2 f}{\partial x_3 \partial x_1} &= 0, \\
\frac{\partial^2 f}{\partial x_2 \partial x_3} - \frac{\partial^2 f}{\partial x_3 \partial x_2} &= 0.
\end{aligned} \tag{3.13}$$

Now, suppose the filter is applied to an arbitrary three-variable conservative vector field $\boldsymbol{F}_0$ that derives from a scalar objective f_0, so that

$$\boldsymbol{F}_0 = \nabla f_0 = \begin{bmatrix} \dfrac{\partial f_0}{\partial x_1} \\ \dfrac{\partial f_0}{\partial x_2} \\ \dfrac{\partial f_0}{\partial x_3} \end{bmatrix}$$

[5] Provided that the domain on which $\boldsymbol{F}$ is defined is simply connected, that $\boldsymbol{F}(\boldsymbol{x})$ is continuous and that the curve joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_0$ is simple.

and f_0 satisfies (3.13). By differentiating the filter equation (3.6), the off-diagonal elements ($i \neq j$) of the Hessian can be written

$$\left(\frac{\partial^2 f_t}{\partial x_j \partial x_i}\right) = \frac{\sum_{k=1}^{n} w_{ik}\left(\delta_{kj}\frac{\partial f_0}{\partial x_k} + x_k\left(\frac{\partial^2 f_0}{\partial x_j \partial x_k}\right)\right)}{x_i \sum_{k=1}^{n} w_{ik}},$$

so that for the three-variable problem the mixed partials are expressed as

$$\left(\frac{\partial^2 f_t}{\partial x_j \partial x_i}\right) = \frac{1}{x_i\left(w_{i1}+w_{i2}+w_{i3}\right)}\left(w_{ij}\frac{\partial f_0}{\partial x_j} + w_{i1}x_1\frac{\partial^2 f_0}{\partial x_j \partial x_1} + w_{i2}x_2\frac{\partial^2 f_0}{\partial x_j \partial x_2} + w_{i3}x_3\frac{\partial^2 f_0}{\partial x_j \partial x_3}\right).$$

Provided that the convolution operator is constant, so that its form does not change as a function of the spatial position $\boldsymbol{r}$ and it is independent of the design variables $\boldsymbol{x}$, then

$$W = \sum_{k=1}^{n} w_{ik} \quad \text{and} \quad w_{ii} = w \quad \forall \quad i = 1, 2, \cdots, n. \tag{3.14}$$

Choosing $j = 1$ and $i = 2$, we find that

$$\begin{aligned}\frac{\partial^2 f_t}{\partial x_1 \partial x_2} - \frac{\partial^2 f_t}{\partial x_2 \partial x_1} =& \frac{1}{W}\left[\left(\frac{w_{21}}{x_2}\right)\frac{\partial f_0}{x_1} - \left(\frac{w_{12}}{x_1}\right)\frac{\partial f_0}{x_2}\right] + \\ & \frac{1}{W}\left[\left(\frac{w_{21}x_1}{x_2}\right)\frac{\partial f_0}{\partial x_1^2} - \left(\frac{w_{12}x_2}{x_1}\right)\frac{\partial f_0}{\partial x_2^2}\right] + \\ & \frac{1}{W}\left[\left(w_{22}\right)\frac{\partial f_0}{\partial x_1 \partial x_2} - \left(w_{11}\right)\frac{\partial f_0}{\partial x_2 \partial x_1}\right] + \\ & \frac{1}{W}\left[\left(\frac{w_{23}x_3}{x_2}\right)\frac{\partial f_0}{\partial x_1 \partial x_3} - \left(\frac{w_{13}x_3}{x_1}\right)\frac{\partial f_0}{\partial x_2 \partial x_3}\right].\end{aligned} \tag{3.15}$$

Similar equations of course result for the other two differences of mixed partials, $(13-31)$ and $(23-32)$. In (3.15), the third term on the right-hand side disappears by virtue of the fact that the unfiltered field $\boldsymbol{F}_0$ satisfies the conditions in (3.13), and we have assumed conditions (3.14). However, it is unlikely that the remaining terms on the right-hand side of (3.15) sum to zero for all allowable values of the variables x_1, x_2 and x_3. This being the case, it would seem that the vector field generated upon application of the filter is unlikely to be conservative, and the filtered sensitivities are therefore unlikely to be associated with a different scalar function f_t. Certainly, the filter does not automatically produce a conservative vector field from any conservative field $\boldsymbol{F}$.

The above does not suggest that the filter *cannot* be associated with a scalar function. It may well be possible that certain combinations of function and convolution operator exist that will produce a vector field representing the gradients of another scalar potential function. But the form of (3.15) is reason enough to suspect that the generation of a conservative vector field is not the *de facto* result of applying the filter, and that, for any given scalar objective, the converse is more likely to be true. In particular, we are therefore motivated to postulate that the filtered compliance sensitivities are not associated with a different objective function at all.

Compliance objective

For the three variable compliance objective, the above can be verified directly. To simplify matters, we will assume a simplified convolution operator that meets the criterion (3.14), namely that

$$w_{ij} = 1 \qquad \forall \quad i = 1, 2, 3, \quad j = 1, 2, 3. \tag{3.16}$$

As discussed in Section 2.1.2, the unfiltered sensitivities of the SIMP-penalised compliance objective function are given by

$$\frac{\partial f_0}{\partial x_i} = -p x_i^{p-1} \, \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \, .$$

This form, in which $\boldsymbol{K}_i$ is the elemental stiffness matrix of element i, and $\boldsymbol{q}_i$ is the vector of nodal displacements for element i, is useful for the numerical calculation of the sensitivities. However, it may also be written as

$$\frac{\partial f_0}{\partial x_i} = -p x_i^{p-1} \, \boldsymbol{q}^T \boldsymbol{K}_i \boldsymbol{q} \, , \tag{3.17}$$

where $\boldsymbol{q}$ is the complete displacement vector for the mesh. The matrix $\boldsymbol{K}_i$ again denotes the elemental stiffness matrix for element i, though now it should be understood to be represented as a global matrix of size $[n_{dof} \times n_{dof}]$, in which only the degrees of freedom associated with element i are potentially non-zero (n_{dof} being the total number of degrees of freedom for the mesh). From

$$\frac{\partial}{\partial x_j} \left(\boldsymbol{K} \boldsymbol{q} \right) = \frac{\partial \boldsymbol{w}}{\partial x_j} \, ,$$

for design-independent loads we obtain

$$\frac{\partial \boldsymbol{q}}{\partial x_j} = -p x_j^{p-1} \boldsymbol{K}^{-1} \left[\boldsymbol{K}_j \boldsymbol{q} \right] \, . \tag{3.18}$$

Using (3.17) and (3.18), the second-order partial derivatives of the penalised compliance objective can be expressed as

$$\frac{\partial f_0}{\partial x_j \partial x_i} = 2p^2 \left(x_i^{p-1} \right) \left(x_j^{p-1} \right) \left[\boldsymbol{K}_i \boldsymbol{q} \right]^T \boldsymbol{K}^{-1} \left[\boldsymbol{K}_j \boldsymbol{q} \right] - \delta_{ij} p \left(p - 1 \right) \left(x_i^{p-2} \right) \left[\boldsymbol{q}^T \boldsymbol{K}_i \boldsymbol{q} \right] . \tag{3.19}$$

Note that since $\boldsymbol{K}^{-1}$ is symmetric, for the off-diagonal terms $i \neq j$,

$$\frac{\partial f_0}{\partial x_j \partial x_i} = \frac{\partial f_0}{\partial x_i \partial x_j} \tag{3.20}$$

as expected. Therefore, for the three-dimensional filtered sensitivities, in which the convolution operator has been simplified according to (3.16) and applied to the sensitivities of the compliance objective, the difference of mixed partials $12 - 21$ in equation (3.15) is given by

$$\begin{aligned} &\frac{\partial^2 f_t}{\partial x_1 \partial x_2} - \frac{\partial^2 f_t}{\partial x_2 \partial x_1} = \\ &\frac{p^2}{3x_1} \left(x_2^{p-1} \right) \left(\boldsymbol{q}^T \boldsymbol{K}_2 \boldsymbol{q} - 2x_2^p \left[\boldsymbol{K}_2 \boldsymbol{q} \right]^T \boldsymbol{K}^{-1} \left[\boldsymbol{K}_2 \boldsymbol{q} \right] - 2x_3^p \left[\boldsymbol{K}_3 \boldsymbol{q} \right]^T \boldsymbol{K}^{-1} \left[\boldsymbol{K}_2 \boldsymbol{q} \right] \right) - \\ &\frac{p^2}{3x_2} \left(x_1^{p-1} \right) \left(\boldsymbol{q}^T \boldsymbol{K}_1 \boldsymbol{q} - 2x_1^p \left[\boldsymbol{K}_1 \boldsymbol{q} \right]^T \boldsymbol{K}^{-1} \left[\boldsymbol{K}_1 \boldsymbol{q} \right] - 2x_3^p \left[\boldsymbol{K}_3 \boldsymbol{q} \right]^T \boldsymbol{K}^{-1} \left[\boldsymbol{K}_1 \boldsymbol{q} \right] \right) . \end{aligned} \tag{3.21}$$

Coordinates			Results		
x_1	x_2	x_3	$\left\lvert \frac{\partial^2 f_t}{\partial x_1 \partial x_2} - \frac{\partial^2 f_t}{\partial x_2 \partial x_1} \right\rvert$	$\left\lvert \frac{\partial^2 f_t}{\partial x_1 \partial x_3} - \frac{\partial^2 f_t}{\partial x_3 \partial x_1} \right\rvert$	$\left\lvert \frac{\partial^2 f_t}{\partial x_2 \partial x_3} - \frac{\partial^2 f_t}{\partial x_3 \partial x_2} \right\rvert$
0.146	0.176	0.058	1.572×10^{6}	3.480×10^{6}	6.844×10^{6}
0.231	0.201	0.136	1.163×10^{5}	2.161×10^{5}	5.088×10^{4}
0.027	0.002	0.021	1.443×10^{14}	1.142×10^{10}	1.857×10^{14}

Table 3.1: Differences in mixed partials at three pseudo-randomly chosen (feasible) points for the three-variable MBB beam.

Equation (3.21) can be evaluated at various points $\boldsymbol{x}$ for any particular structure discretised by only three elements. It is only strictly necessary to identify a single point for which (3.21) or either of the other two differences of mixed partials is non-zero to demonstrate that the filtered sensitivities do not represent a conservative vector field[6]. Table 3.1 contains the results of the evaluation of the differences in mixed partials for the filtered problem at three random feasible points for the MBB beam problem, in which the half-beam is discretised using only three elements (Figure 3.2). Sigmund's 99-line topology code [9] was used for this purpose. Clearly, these terms are non-zero (while the corresponding terms are verifiably zero for the unfiltered problem), implying that the vector field defined by the filtered sensitivities is not conservative.

For other, more representative compliance problems, which have a greater number of variables and employ a more standard convolution operator, the conservativeness of the filtered gradients is not tested as easily. The test involving the curl operator (3.12) is valid in two and three dimensions, but not in higher dimensions, being defined in terms of the cross product. An equivalent notion for higher dimensions is difficult to come by, and even more difficult to understand (for this author at least). Therefore, for larger problems we propose to test whether the filtered compliance gradient field is conservative (or not) by numerically checking the path independence of (3.11). We do so in Section 3.7.

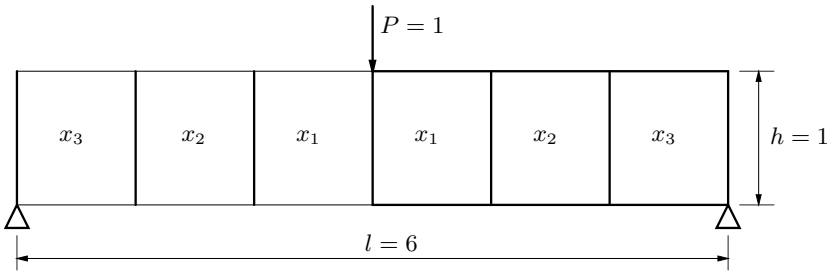


Figure 3.2: The MBB beam (unit thickness; plane stress; $E = 1$, $\nu = 0.3$).

[6]The mixed partial derivatives could of course be approximated using finite differences instead.

3.7 Numerical examples

In this section we numerically catalogue the propensity for the filter to produce non-conservative vector fields from the unfiltered conservative gradient fields of a few additional problems. The first problem considered is another analytical example, which we use to generate graphical results that illustrate the method used to establish whether or not the considered field is conservative. We also use this example to express a few thoughts regarding the convergence of SAO procedures using the filter. The other problems considered are larger and more representative minimum compliance topology problems. That is: they have more than three variables (though they are still very small) and use a standard discrete convolution operator.

Since the compliance problems have more than three variables, we lack a straightforward test for the conservativeness of their gradient fields. We therefore propose to use equation (3.11) to numerically calculate the function values from the gradient field along two separate piecewise linear curves, or routes, both originating at the same point p_0 and terminating at the same point p_2 in the design space. Figure 3.3 illustrates the process. If the filtered gradient field $\boldsymbol{F}_t$ is conservative, and is thus associated with a scalar objective function, then the function value arrived at for p_2 using route 1 should be identical to the function value determined using route 2, so that

$$f_t^{R1}\left(\boldsymbol{x}_{p_2}\right) - f_t^{R2}\left(\boldsymbol{x}_{p_2}\right) = 0\,. \tag{3.22}$$

Each route is constructed from two line segments defined by the random selection of two additional points, p_1 for route 1 and p_1' for route 2. An outline of the procedure used is as follows:

1. Calculate $\triangle \boldsymbol{s} = \beta\left(\boldsymbol{x}_e - \boldsymbol{x}_b\right)$ for the current line segment, where $\boldsymbol{x}_b$ and $\boldsymbol{x}_e$ are respectively

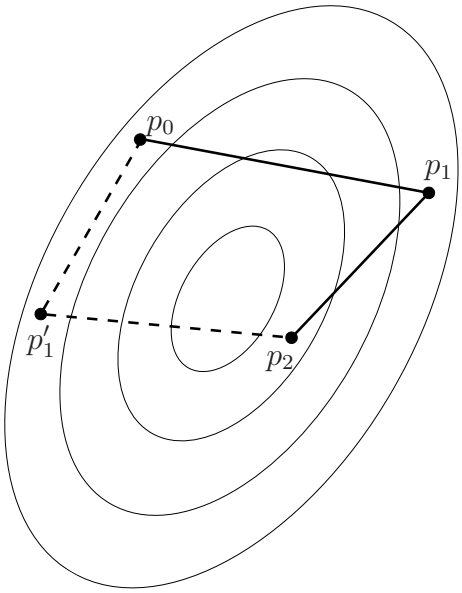


Figure 3.3: Two piecewise linear curves joining two points p_0 and p_2.

the beginning and end points of the line segment and β is the desired step size.

2. Given the current point $\boldsymbol{x}^{\{k\}}$ and its function value $f_t\left(\boldsymbol{x}^{\{k\}}\right)$, evaluate $\boldsymbol{F}_t\left(\boldsymbol{x}^{\{k\}}\right)$.

3. Calculate $f_t\left(\boldsymbol{x}^{\{k+1\}}\right) = \left[\boldsymbol{F}_t\left(\boldsymbol{x}^{\{k\}}\right)\right] \cdot \triangle \boldsymbol{s}$ and determine the following point on the line segment $\boldsymbol{x}^{\{k+1\}} = \boldsymbol{x}^{\{k\}} + \triangle \boldsymbol{s}$.

4. Repeat (2) and (3) until the end of the line segment is reached.

Numerical error, which is dependent on both the step length and the magnitude of the elements of the gradient field, is inherent in the process just described. However, it is again only necessary to generate one result for which (3.22) does not hold, and for which we are confident that the result is not caused by the error, to conclude that the field is not conservative.

In an effort to define the scale of the error produced by the numerical line integrals, the same process that is carried out on $\boldsymbol{F}_t$ is also carried out on the unfiltered gradient field $\boldsymbol{F}_0$, which is already known to be conservative because it derives from a scalar objective function f_0. Since $f_t\left(\boldsymbol{x}_{p_2}\right)$ is calculated relative to $f_t\left(\boldsymbol{x}_{p_0}\right)$, the error produced by integrating $\boldsymbol{F}_0$ along a specific route R_i, namely $\left[f_0\left(\boldsymbol{x}_{p_2}\right) - f_0^{R_i}\left(\boldsymbol{x}_{p_2}\right)\right]$, is normalised with respect to $\left[f_0^{R_i}\left(\boldsymbol{x}_{p_2}\right) - f_0\left(\boldsymbol{x}_{p_0}\right)\right]$, the superscript R_i indicating that the function value was obtained by numerical integration along route i. Then, the error that we expect to be produced by numerically integrating $\boldsymbol{F}_t$ along the same path R_i is estimated as

$$E^{R_i} = \left| \left[f_t^{R_i}\left(\boldsymbol{x}_{p_2}\right) - f_0\left(\boldsymbol{x}_{p_0}\right)\right] \frac{\left[f_0\left(\boldsymbol{x}_{p_2}\right) - f_0^{R_i}\left(\boldsymbol{x}_{p_2}\right)\right]}{\left[f_0^{R_i}\left(\boldsymbol{x}_{p_2}\right) - f_0\left(\boldsymbol{x}_{p_0}\right)\right]} \right|, \tag{3.23}$$

in which $f_0\left(\boldsymbol{x}_{p_0}\right)$ and $f_0\left(\boldsymbol{x}_{p_2}\right)$ are the actual function values at the beginning and end of route i, determined by evaluating the function directly. Additionally, to minimise the error further, all points are chosen close together (within a unit radius of one another).

3.7.1 A 3D convex and separable example

Consider the following strictly convex and separable programming problem, similar to the graphic example (3.9) given in Section 3.5.2:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) = \frac{3}{x_1^2} + \frac{1}{x_2^2} + \frac{2}{x_3^2} \\ \text{subject to} \quad & f_1(\boldsymbol{x}) = x_1 + x_2 + x_3 - 0.8 \leq 0\,, \\ & x_1, x_2, x_3 > 0\,. \end{aligned} \tag{3.24}$$

Conservatism of the filtered vector field is investigated as described above. The results of four numerical experiments are given. Table 3.2 lists the coordinates of the four (pseudo-randomly selected) points used in each of the tests, as well as the actual function values $f_0\left(\boldsymbol{x}_{p_0}\right)$ and $f_0\left(\boldsymbol{x}_{p_2}\right)$

and the function values at p_2 derived by numerical integration of the unfiltered gradients $f_0^{R_1}(\boldsymbol{x}_{p_2})$ and $f_0^{R2}(\boldsymbol{x}_{p_2})$. The convolution operator for these tests is set at $w_{ij} = 1 \,\forall\, i, j$.

Table 3.3 summarises the results obtained by integrating the filtered gradient field. The differences in function values obtained at p_2 are compared with the estimated expected error. We may thereby judge whether the differences in function values obtained are the result of error, or the result of the field being non-conservative. We expect that, if the filtered field is conservative, the difference in function values should be of the same order as the sum of the expected errors for each integration route. The step size used for this example is $\beta = 5 \times 10^{-5}$.

The results of four tests are shown. The results are typical in that a difference in function values at p_2 is evident, indicating that the results obtained are path dependent. Only those results have been shown for which the integration error produced by integrating the unfiltered field is low. The integration paths for which this is not the case are likely to include regions in which the gradient is very steep (this can be expected since the function has asymptotes), in which case the numerical procedure employed will be deficient without resorting to smaller step sizes. The four tests presented clearly exhibit function differences that are orders of magnitude greater than the estimated numerical error, so we conclude that the filtered sensitivities do not correspond to any scalar objective function.

Figure 3.4 (page 60) displays the results graphically. It shows the functions $f_t(\boldsymbol{x})$ obtained by integrating the filtered sensitivities $\boldsymbol{F}_t \cdot d\boldsymbol{s}$ along the piecewise linear routes defined in Table 3.2. For comparison, the function derived from carrying out the same procedure on the unfiltered sensitivities $\boldsymbol{F}_0$ is also shown (i.e. we integrate the directional derivatives of f_0 along S). It should be noted that the graphs obtained by integrating $\boldsymbol{F}_0 \cdot d\boldsymbol{s}$ are indistinguishable from the true function values f_0 along the specified routes on the scales at which the graphs are plotted. The points of discontinuity in the curves correspond to the points p_1 and p_1' in each route at which there is an abrupt change in direction.

Of course, the $\operatorname{curl}(\nabla f) = \mathbf{0}$ argument could have been used to prove the filtered sensitivities are non-conservative for this problem, it being only three-dimensional, but the example serves to illustrate the numerical process applied to the higher dimensional compliance problems below. Also, and more importantly perhaps, is that this problem allows us to test the possibility of convergence for optimisation algorithms using the filtered sensitivities rather than the original ones.

A word on convergence

The fact that the filter creates subproblems that are not first-order accurate raises the question of whether convergence can be expected to occur at all when the filter is used in an OC or SAO framework. The widespread use of the filter is itself probably reason enough to presume that the filter does not disturb convergence. However, that convergence will occur has not been established generally; a formal proof that the filter does not upset the ability of SAO to converge, despite the lack of first-order accuracy, would place it on a surer footing theoretically. We cannot proffer such a proof. However, we here discuss certain observations based on the three-variable separable and strictly convex problem introduced above.

As with the compliance problem, this three-variable problem has partial derivatives that are everywhere negative. Provided that all the x_i and all the w_{ij} are positive, the filter preserves this

Coordinates				Function values			
Point	x_1	x_2	x_3	$f_0(\boldsymbol{x}_{p_0})$	$f_0(\boldsymbol{x}_{p_2})$	$f_0^{R_1}(\boldsymbol{x}_{p_2})$	$f_0^{R2}(\boldsymbol{x}_{p_2})$
				Test 1			
p_0	0.2080	0.1044	0.0908	4.036×10^2	4.281×10^2	4.277×10^2	4.279×10^2
p_1	0.2520	0.1480	0.0637				
p_1'	0.1797	0.1824	0.2540				
p_2	0.1035	0.1407	0.1430				
				Test 2			
p_0	0.203	0.140	0.206	7.394×10^3	2.642×10^3	2.634×10^3	2.632×10^3
p_1	0.029	0.029	0.075				
p_1'	0.026	0.243	0.258				
p_2	0.163	0.020	0.150				
				Test 3			
p_0	0.207	0.050	0.132	5.811×10^2	1.510×10^3	1.509×10^3	1.506×10^3
p_1	0.277	0.150	0.207				
p_1'	0.163	0.130	0.031				
p_2	0.089	0.076	0.046				
				Test 4			
p_0	0.182	0.189	0.299	1.413×10^2	3.111×10^2	3.110×10^2	3.105×10^2
p_1	0.140	0.100	0.152				
p_1'	0.064	0.242	0.339				
p_2	0.121	0.138	0.193				

Table 3.2: Coordinates defining the line segments used in examining the conservatism of the filtered gradient field of problem (3.24), together with function values at the initial and terminal points of the integration paths.

characteristic: all the elements of the filtered gradient field will be everywhere negative. Therefore, any sequence of descent steps produced by a descent algorithm is bound to intersect the linear constraint (descent being defined in this case as proceeding in the direction negative to the gradient).

Ordinarily (that is, when minimising a scalar function f) one would observe that, provided the function actually has a finite absolute minimum on a closed and continuous feasible region (i.e. it does not asymptote to negative infinity), the sequence of descent steps produced is bounded below. Convergence can then be adduced, although the point to which convergence occurs can lie in a subspace on which the function is constant, or the function may be multimodal, so convergence to a *particular* point cannot be assured. When examining the convergence characteristics for an SAO sequence, one would normally take as starting assumptions the existence of KKT points, as

Test	$f_t^{R_1}(\boldsymbol{x}_{p_2})$	E^{R_1}	$f_t^{R2}(\boldsymbol{x}_{p_2})$	E^{R_2}	$\left\|E^{R_i}+E^{R_i}\right\|$	$\left\|f_t^{R_1}(\boldsymbol{x}_{p_2})-f_t^{R2}(\boldsymbol{x}_{p_2})\right\|$
1	3.707×10^{2}	0.539	4.215×10^{2}	0.157	0.696	5.08×10^{1}
2	1.121×10^{4}	6.147	0.364×10^{4}	7.426	13.573	7.57×10^{3}
3	1.413×10^{3}	0.997	1.975×10^{3}	4.783	5.780	5.62×10^{2}
4	2.878×10^{2}	0.077	4.315×10^{2}	1.089	1.166	1.44×10^{2}

Table 3.3: A comparison of the differences in function values obtained for p_2 by numerical integration along two different paths with the expected error involved in the integration. If F_t is conservative, the difference in the function values should be of the same order as the cumulative error.

well as the idea that the function can be seen as locally convex in some small region surrounding each KKT point. Whatever the specific nature of the assumptions, it is the properties of the scalar objective function that are used in arguments to assert convergence. The familiar KKT conditions, which characterise the optimal solutions, are also phrased in terms of the partial derivatives of the objective function. However, when the filter is used it is a little difficult to build a similar argument asserting convergence if no such scalar function exists. Therefore, an analysis of convergence must be based on this filtered gradient field itself.

Consider, for instance, the curves obtained by integrating the filtered gradients F_t along linear routes, depicted in Figure 3.4. Although these curves do not correspond to sections of any function, one notes that the portions that correspond to particular line segments in each route are strictly convex. The original function f_0 is, of course, convex and it appears that the filter has preserved some measure of convexity. For convex functions, we know that their Hessians are positive definite, which is to say that

$$\boldsymbol{x}^T\left[\nabla^2 f\right]\boldsymbol{x}=\boldsymbol{x}^T\left[\nabla F\right]\boldsymbol{x}>0 \qquad \forall \quad \boldsymbol{x}\neq\mathbf{0}\,. \tag{3.25}$$

Although we cannot ascribe positive definiteness to any function corresponding to the filtered gradient field for this three variable problem (since no such function exists), it is easy to verify that in this case $\boldsymbol{x}^T\left[\nabla F_t\right]\boldsymbol{x}>0$, since all the components of $[\nabla F_t]$ are positive. It seems justifiable, by an extension of the familiar properties of convex functions, to suspect that any vector fields that satisfies (3.25) will have a unique terminal point for any sequence of descent steps (projected onto active constraints if necessary) on any convex domain.

Indeed, one is able to test numerically that a projected gradient descent algorithm (that takes steps in the direction of the maximal descent vector projected onto the subspace of active constraints) will converge to the same point for the filtered version of (3.24) for a given set of weights w_{ij} that define the convolution operator. Table 3.4 lists the terminal point $\boldsymbol{x}^*$ to which the algorithm converges for three symmetric choices of w_{ij}. A maximum step size of 0.001 (before projection) is used.

The process used here to minimise the function is similar to the process of numerically producing trajectories through phase space for nonlinear systems, given the differential equations describing the system [47]. It is known that, depending on the properties of the system in question, such trajectories can converge to a point within the space. However, trajectories can also converge to a limiting closed cyclic trajectory, and systems may even have strange attractors (about which the

Example	w_{11}	w_{12}	w_{13}	w_{22}	w_{23}	w_{33}	x_1^*	x_2^*	x_3^*
1	1.0	1.0	1.0	1.0	1.0	1.0	0.2667	0.2667	0.2667
2	1.0	0.5	1.0	1.0	0.5	1.0	0.2767	0.2775	0.2458
3	1.0	0.5	0.0	1.0	0.5	1.0	0.2715	0.2898	0.2388

Table 3.4: Convergence points for a descent algorithm applied to the 3D convex test problem. The point of convergence depends on the definition of the convolution operator w_{ij}. Here, three symmetric operators ($w_{ij} = w_{ji}$) have been used. The solution to the unfiltered problem is $\boldsymbol{x}^* = (0.3116, 0.2161, 0.2723)$.

trajectories are non-repetitive but also non-terminating. Of course, trajectories may also diverge. Superficially, it appears to us that proving convergence for the optimisation process applied to the filtered problem would require showing that the filtered gradient field only possesses point attractors, at least on the closed feasible region defined by the problem's constraints.

If convergence is achieved then the solution obtained, which we denote $\boldsymbol{x}^{\{k*\}}$, satisfies the KKT conditions for a stationary point of the approximate subproblem defined at $\boldsymbol{x}^{\{k*\}}$, provided the familiar constraint qualification is satisfied. Therefore, at the subproblem level, the stationary condition for the Lagrangian reads

$$\left(\frac{\partial \tilde{f}_0}{\partial x_i}\right)^{\{k*\}} + \lambda \left(\frac{\partial \tilde{f}_1}{\partial x_i}\right)^{\{k*\}} = 0\,, \tag{3.26}$$

from which the β_i in (3.3) are derived. The bound constraints on $\boldsymbol{x}$ are handled separately in the update (3.1), which is consistent with the use of the Falk dual (see Section 2.3.1) rather than a standard Lagrangian dual in which the bounds would have to be included in the definition of the Lagrangian.

As discussed, the point of convergence $\boldsymbol{x}^{\{k*\}}$ (if it exists) generally will not be a KKT point of the original, unfiltered problem P_C. Since a filtered objective function f_t does not exist, condition (3.26) really doesn't form part of the KKT conditions for any associated problem besides the terminal SAO subproblem. However, at the problem level (as opposed to the subproblem level) we may write the stationary conditions (3.26) in terms of the filtered gradient field as

$$(F_t)_i^{\{k*\}} + \lambda \left(\frac{\partial f_1}{\partial x_i}\right)^{\{k*\}} = 0\,, \tag{3.27}$$

in which

$$(F_t)_i^{\{k*\}} = \left(\widehat{\frac{\partial f_0}{\partial x_i}}\right)^{\{k*\}}.$$

Lastly, we must emphasise the fact that the remarks presented here regarding convergence pertain to the solution of the relaxed continuous minimum compliance problem. However, in addressing the topology problem for a given spatial discretisation (i.e. mesh refinement), it is really the discrete programming problem (2.7) that we are trying to solve. When viewed strictly in a combinatorial sense, (2.7) does not possess gradients anyway, and neither are the KKT conditions relevant for

characterising its optima. So, while convergence is obviously an important aspect of the optimisation, the observation that the filter disturbs convergence to a KKT point of the relaxed continuous problem is probably subordinate to the question of whether the filter encourages mesh-independent solid-void solutions to be found. The results depicted in Figure 5.5 are a good example. They are minimum compliance results for the MBB beam structure, generated using a relaxed continuous formulation in which the filter is employed. The results exhibit mesh independence and exceptionally high black-and-white fractions. Termination of the search algorithm occurred on a minimum tolerance imposed on the magnitude of the design changes $\|\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}\| \leq \epsilon_x$.

3.7.2 Larger MBB beam problems

The method described above, for assessing whether or not a given gradient field is conservative, is now applied to the filtered sensitivities of the compliance objective for larger MBB problems, being more representative of the problems that are typically solved using the sensitivity filter. Although the analysed problems are still quite small, the mesh discretisation is sufficient to allow the use of the standard convolution operator, with the filter radius being set at $r = 1.5$ elements in the topology code (refer to [9]).

The results are depicted in Table 3.5 for two mesh discretisations. Three results are displayed for each discretisation. Column 3 in the table records the maximum error produced (for the two routes) by applying the numerical integration to the conservative unfiltered gradients. Column 4 shows the sum of the expected errors, to be compared with the actual difference in function values obtained by integrating the filtered sensitivities along the two different paths.

The differences obtained are an order of magnitude greater than the expected numerical error involved in the integration. Therefore we conclude that the gradient field defined by the filtered sensitivities of the compliance objective is not a conservative vector field, and therefore that there is no scalar function to which it corresponds. If no spatially discretised relaxed continuous function exists that is associated with the filtered sensitivities, it is difficult to see how a different continuum problem could exist.

Finally, it is in order to relay a thought regarding checkerboarding in the context of the filter interpreted as a generator of SAO subproblems. Checkerboarding is known to be a spurious anomaly

Test	Mesh	$\max_{i=1,2} \left\lvert f_0^{R_i}(\boldsymbol{x}_{p_2}) - f_0(\boldsymbol{x}_{p_2}) \right\rvert$	$\left\lvert E^{R_1} + E^{R_2} \right\rvert$	$\left\lvert f_t^{R_1}(\boldsymbol{x}_{p_2}) - f_t^{R2}(\boldsymbol{x}_{p_2}) \right\rvert$
1	9×3	0.714	1.510	5.88×10^1
2	9×3	0.747	1.482	4.05×10^1
3	9×3	5.123	10.970	2.55×10^2
4	15×5	0.828	0.481	1.85×10^2
5	15×5	0.243	1.126	4.91×10^1
6	15×5	6.924	13.072	8.51×10^2

Table 3.5: Expected errors and differences in function values obtained by numerical integration along two separate integration paths joining p_0 and p_2 for the MBB compliance problem.

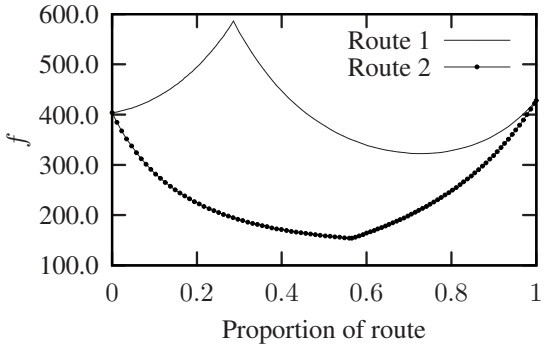
of the numerical analysis of the structure when displacement-based Q4 elements are used in the FEM, which results in checkerboarded designs being unphysically over-stiff. From the point of view of the optimiser, however, this anomaly is part of the definition of the objective, and external to the optimiser. If no additional constraints are present in the problem that constrain out checkerboarded designs, then the solutions to the optimisation problem that correspond to the over-stiff checkerboarded designs are superior designs from the point of view of the optimiser, which seeks solutions of minimal compliance. The point here is that an optimiser based on SAO using first-order accurate approximations *should* converge to the stationary points that represent the stiff checkerboarded solutions. This implies that one cannot use any first-order accurate filter-based technique for constructing SAO subproblems if the intention is to avoid checkerboarded designs when Q4 elements have been used. Interestingly, it appears to be the loss of first-order accuracy that allows checkerboarded designs to be avoided. But then, of course, the solutions obtained are in general not stationary points of the originally stated (relaxed continuous) compliance problem either.

3.8 Conclusion

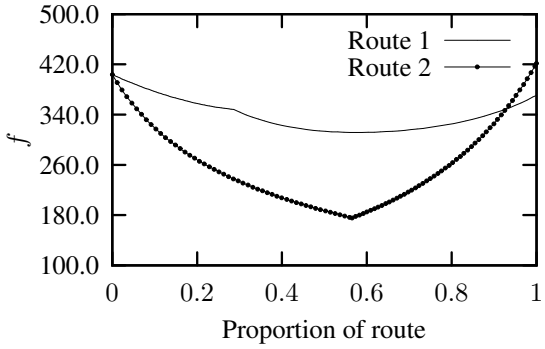
We have studied the mesh independence filter of Sigmund in the context of sequential approximate optimisation algorithms based on dual principles. We have shown that the filtered sensitivities define the exact gradients of a modified approximate primal subproblem, and therefore that a concrete interpretation of the filter exists in the context of SAO. These subproblems are not only separable in the design variables, but also (conditionally) strictly convex. According to this interpretation, the problem that is solved when the filter is applied is still the originally stated topology problem. We have also argued that the accepted contrary view, that the filter gives rise to another objective function entirely, is not valid.

We have thus provided an explanation of the form of the filter based on the concepts of SAO, but many interesting questions remain unanswered. Firstly, we do not know why the use of these particular approximations would ensure either mesh independence or checkerboard-free designs for topology problems. However, we hope that this novel interpretation of the filter may be used in future as the basis on which to explore these questions, perhaps more fruitfully than in the past. Secondly, the subproblems defined by the filter are not accurate to first order, and this raises the question of whether convergence is assured when SAO algorithms that incorporate the filter are utilised. Also, if convergence occurs, how is the solution obtained to be interpreted relative to the stated objective function? There is certainly ample scope for further research along this line of reasoning.

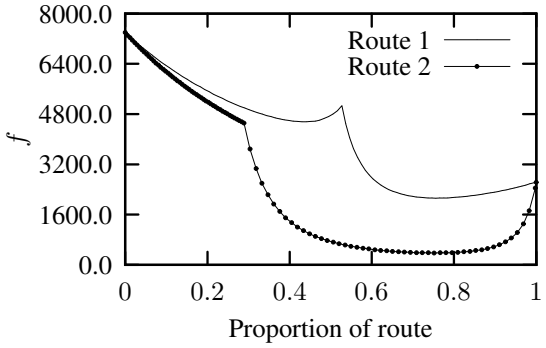
Lastly, we consider it instructive and correct to refer to Sigmund's original method for filtering as 'filtering through modified approximate primal subproblems', or 'approximation-based filtering' for short. The insights developed herein may be used to propose alternative forms of approximation-based filtering.



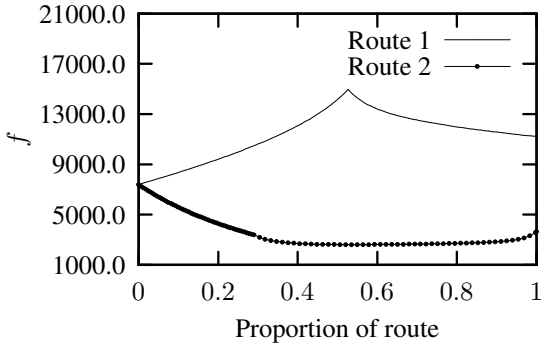
(a) Example 1: Unfiltered



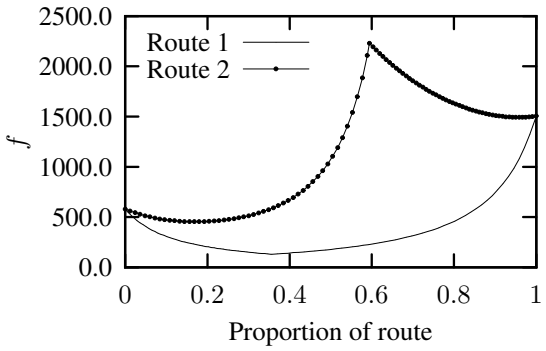
(b) Example 1: Filtered



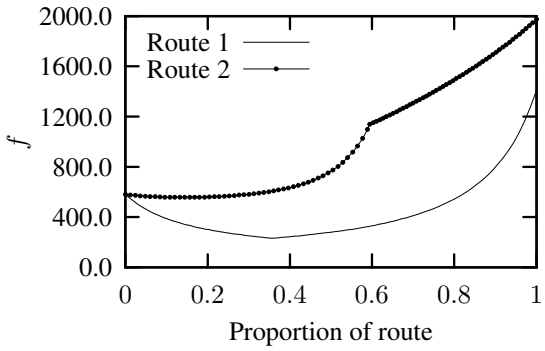
(c) Example 2: Unfiltered



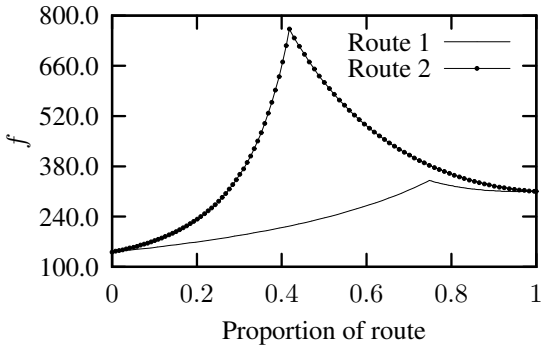
(d) Example 2: Filtered



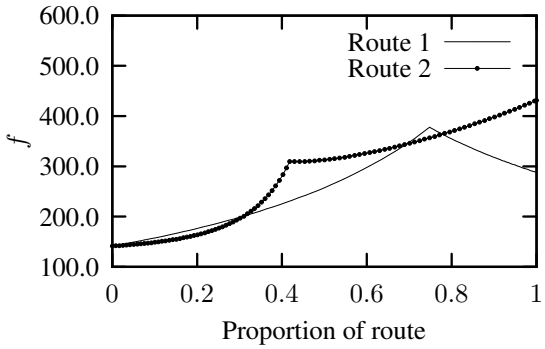
(e) Example 3: Unfiltered



(f) Example 3: Filtered



(g) Example 4: Unfiltered



(h) Example 4: Filtered

Figure 3.4: Linear sections through the unfiltered objective function and the filtered ‘function’ constructed by numerically integrating the directional derivatives.

Chapter 4

A discrete topology problem

This chapter is based in part on a paper presented at a conference on fibre-reinforced composites, hosted by the South African Council for Scientific and Industrial Research (CSIR) from the 9th to the 12th of December 2007, in Port Elizabeth, South Africa. The original conference paper is titled "Optimisation of constrained mixed discrete continuous composite problems via a dual method of sequential approximate optimisation" [48]. It is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa.

4.1 Abstract

In this chapter, a dual method for the optimisation of discrete and mixed discrete-continuous constrained problems that appear in the analysis of fibre-reinforced composite structures is presented. Developed initially by Schmit and Fleury, the method is an extension of a popular dual approach used in the sequential approximate optimisation of continuous problems to (mixed) discrete problems. As such, the primal problem is substituted by a suitable convex and separable approximate subproblem during each iteration of the algorithm. A continuous but piecewise linear Falk dual is defined and solved subject only to non-negativity constraints on the dual variables, yielding an approximation to the (mixed) discrete primal optimum. Both the advantages and shortcomings of the method are illustrated. Its utility for the optimisation of constrained discrete problems is demonstrated by a novel application of the method to a problem concerning the combined selection of the optimum point-wise fibre direction and solid-void material distribution in the minimum compliance design of planar composite structures.

4.2 Introduction

The use of fibre-reinforced composite (FRC) materials has become important in structural design, particularly in industries that place a premium on developing high-strength, low-weight structures, such as the aerospace industry. In addition to the advantages gained by their high strength-to-weight ratios, the orthotropic nature of FRCs also affords designers great freedom to tailor com-

posite designs to suit the prevailing structural loads. However, the drawback of such flexibility is the increased complexity of analysis, design and optimisation of FRC structures over structures composed of isotropic material. Venkataraman and Haftka [49] provide an overview of the design of composite panels from the standpoint of complexity, and it is noted that high-fidelity analysis of large, complex composite structures is currently nigh impossible due to the computational burden that this entails.

On the other hand, the optimal design of single components or composite laminates has become an active and fruitful area of research. Most commonly, problems concerning optimal ply orientation, thickness and stacking sequence have been addressed for the purposes of buckling, vibration or failure analysis [49, 50]. In these analyses, a given ply's fibre direction(s) and thickness are assumed to be spatially constant throughout the laminate. It has become interesting, however, to consider the design of laminates in which the fibre angle can change as a function of position, particularly now that tow-placement machines have made these so-called 'variable stiffness' laminates manufacturable [51]. The problem of optimal spatially-varying fibre orientation has been addresses by Landriani and Rovati [52], amongst others, and the design of laminates in which both the fibre direction and the laminate thickness are allowed to vary has been considered, for instance, by Pederson [53]. A natural extension of this line of research is the combination of optimal topology and optimal fibre angle design for a laminate. It is toward this question that the work presented in the current chapter is directed. Hansel and Becker [54] and Duvaut *et al.* [55] both present work in which fibre orientation is optimised, in combination with density in the former case and fibre volume fraction in the latter. Both present results that are reminiscent of optimal topologies, though neither algorithm is based on a traditional penalisation-based solid-void topology formulation. Setoodeh *et al.* [56], by contrast, have presented a method that uses the solid isotropic material with penalisation (SIMP) approach to topology optimisation, suggested independently by Bendsøe [18] and Rozvany and Zhou [19], to generate designs in which the optimal topology and optimal local fibre orientation is determined concurrently for the minimum compliance design of FRC plates. Interestingly, they accomplish this through the use of cellular automata.

In this chapter a method is presented for generating such designs that is based largely on the work of Stegmann and Lund [57]. They solve the problem by using the SIMP approach, and by applying a technique that they call 'discrete material optimisation' (DMO), which was first introduced by Sigmund *et al.* [58] as 'multiphase topology optimisation'. DMO allows the optimiser (which in their case was the method of moving asymptotes [3]) to search for the optimal set of material properties from a set of candidate materials. The selection of element-wise fibre directions is accomplished by evaluating the stiffness matrix for an FRC composite at multiple discrete angles, and then defining these as the candidate materials. The DMO formulation described in [57] does not allow for the generation of true solid-void topologies. Instead, an isotropic material with a low stiffness may be included as one of the candidate materials, and the optimiser is free to approximate voids in the domain through the selection of this material. Stegmann and Lund solve the problem in the continuous sense (that is, they solve the relaxed continuous problem) and rely on the SIMP method of penalisation, as well as clever interpolation schemes for the elemental material properties, to generate solutions in which each element is representative of one of the candidate materials.

The approach that we follow is an adaptation of DMO. As with DMO, we consider the material properties of an element to be a linear combination of several discrete candidate materials.

However, we introduce inequality constraints on the density of each element, which allows the optimiser to generate a void element by driving the contribution of all the candidate materials to zero. The resulting problem can be considered large-scale, since there are as many constraints as elements in the finite element mesh and the number of primal variables is a multiple of the number of elements. In order to ensure that at most one candidate material is selected per element, the problem is formulated on the binary discrete set and solved using the discrete dual approach introduced by Schmit and Fleury [16]. As far as we know, this is both a novel adaptation of DMO as well as a novel application of the discrete dual, which has previously been applied to the minimum compliance topology design of isotropic structures by Beckers [17].

Relative to DMO, the complexity and size of the optimisation problem is increased by the addition of elemental constraints, which is obviously not desirable. However, the application of DMO within the framework of dual SAO for this problem gives rise to an advantageous structure for the dual subproblems. Alhough it is standard practise, and even necessary, to formulate the primal subproblems as separable when dual solvers are used, it is unusual to encounter a dual which itself has a separable structure in the space of the dual variables. The FRC topology problem presented here gives rise to just this situation. Even though the dual has large dimensionality, we take advantage of its separability to solve the dual problem efficiently using only a linesearch technique.

4.3 A dual method of sequential approximate optimisation

We begin by considering a general nonlinear programming problem (NLP), which may be stated as

$$\begin{aligned} &\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) \\ \text{subject to } &f_j(\boldsymbol{x}) \leq 0 \qquad j = 1, 2, \cdots, m, \\ &\check{x} \leq x_i \leq \hat{x} \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{4.1}$$

where f_0 represents the n-dimensional function to be minimised and the $f_j,\ j = 1, 2, \cdots, m$, denote the j constraint functions. Each primal variable x_i is considered to be bound constrained between allowable upper and lower bounds ($\check{x}$ and $\hat{x}$ respectively). This form is typical of the relaxed continuous forms of the structural optimisation problems described in Section 2.1. A wide variety of methods exist for solving (4.1). We consider the class that fall under the label of sequential approximate optimisation (SAO), a short description of which was given in Section 2.2.

Due to the great expense of solving large nonlinear constrained structural optimisation problems directly, it has become standard practice to instead derive explicit approximations to the original problem and to optimise these approximate subproblems instead. Since the approximations are only valid locally, it is necessary to iteratively solve the original problem by considering a sequence of approximate subproblems – hence the name SAO. Under certain additional restrictions, e.g. conservatism, separability and convexity, it can be proved that this process converges to a stationary point of the original (relaxed continuous) problem [6]. Obviously, it is desirable to use relatively good quality approximations to the original problem, such as the ones described in Section 2.2.3 for structural optimisation.

Since the constrained subproblems dealt with very often have a large number of primal variables and only a small number of constraints, it is frequently advantageous to solve them using a dual method. The dual method essentially converts a constrained problem into a simple non-negatively constrained problem in the space of the Lagrange multipliers $\boldsymbol{\lambda}$, whose dimensionality equals the number of primal constraints. We apply the dual method proposed by Falk [2], introduced in Section 2.3.1, in which the upper and lower bound constraints on the primal variables do not have to be explicitly considered in the definition of the Lagrangian, which is given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f_0(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}) \tag{4.2}$$

when the constraints are defined in the negative-null sense, as in (4.1). If f_0 is strictly convex and all the f_j are convex, then the global minimiser will uniquely satisfy the KKT conditions [22] for a saddle point of the Lagrangian. Moreover, Falk shows that the dual, defined by

$$\gamma(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \{\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) : \check{x} \leq x_i \leq \hat{x}\}\,, \tag{4.3}$$

is concave under these conditions, and that the (unique) maximum of the dual with respect to $\boldsymbol{\lambda}$, subject only to $\lambda_j \geq 0$, corresponds to the minimum of the original subproblem defined for (4.1), given the relationship between primal and dual variables (4.3). In general, equation (4.3) is difficult to apply, but in the special case that f_0 and all f_j are separable functions, the Lagrangian itself becomes a separable function and (4.3) reduces to a set of one-dimensional minimisations, each in terms of a single primal variable x_i. Separability, then, makes the dual method viable. The approximations described in Section 2.2.3 are all separable and convex, making the dual method applicable whenever said approximations are utilised. It is frequently possible to accomplish the minimisations in (4.3) analytically. The resulting relationships are substituted into (4.2), yielding $\gamma(\boldsymbol{\lambda})$ explicitly. When this is not the case, the one-dimensional minimisations must be performed numerically, degrading the efficiency of the dual method. Finally, one further advantage of the dual approach is the ease of calculating the gradients of the dual function with respect to the dual variables. They are simply given by the values of the associated constraints, found via the primal-dual relationship (4.3), at any given point $\boldsymbol{\lambda}$ in the dual space.

4.3.1 A dual method for mixed discrete-continuous problems

The structural optimisation problems described in Section 2.1 are really discrete in nature. When the underlying continuum problems are discretised (spatially) by means of the finite element method, they take the following general form,

$$\begin{aligned} &\min_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) \\ \text{subject to} \;\; &f_j(\boldsymbol{x}) \leq 0 \qquad j = 1, 2, \cdots, m, \\ &x_i \in [0, 1] \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{4.4}$$

in which the variables are limited to the binary values 0 and 1, signifying the absence or presence of material at a point in the design space[1]. Although the solution of (4.1) has often been considered

[1] In practical implementations it is necessary to replace the set $x_i \in [0, 1]$ with $x_i \in [\check{x}, 1]$, where $\check{x}$ has a strictly positive value close to zero, so as to avoid ill-conditioning problems in the structural analysis.

when $\boldsymbol{x}$ is defined on a discrete set instead of a real interval, relatively little attention has been given to the application of the above dual method in this case. One method of applying the dual is simply to numerically carry out the minimisations in (4.3) over the allowable discrete set using some discrete search method such as Branch and Bound, Genetic Algorithms, rounding of the continuous optimum etc. (see Salajegheh [59] for example). This results in a large number of numerical minimisations if the number of primal variables is high. Due to the cumbersome nature of the integer programming methods relative to methods of continuous programming when applied to such large problems, in structural optimisation it is usually preferred to solve a relaxed continuous version of (4.4) in which the x_i with values intermediate between 0 and 1 are penalised in some way.

However, an alternative (mixed) discrete approach was developed independently by Schmit and Fleury [16] and Sepúlveda and Cassis [60], in which a set of discrete mappings are derived from (4.2) and (4.3) for the variables defined on a discrete set. For the discrete variables, these mappings are used in the definition of the dual, which is accomplished directly, without the need for numerical minimisation of (4.3) over the allowable discrete set. The minimisation in (4.3) – or the sometimes equivalent stationary conditions – are still applied to define the primal-dual mappings for the continuous variables. Thus, each point in the dual space maps to a primal coordinate. However, the discrete mapping is not everywhere unique. A brief description of the details (based on [16]) follows below. Without loss of generality, we describe only the primal-dual relationships for the discrete variables.

Consider a discrete problem that has been approximated by convex and separable functions, as described above. This yields a separable Lagrangian that is convex with respect to each x_i and varies linearly with respect to $\boldsymbol{\lambda}$ (see Section 2.3.3). Figure 4.1(a) depicts a contour plot of one separable part of such a Lagrangian, i.e. $\mathcal{L}_i\left(x_i, \boldsymbol{\lambda}\right)$, and for simplicity we represent a one-dimensional problem with only one constraint, which is to say that $\mathcal{L}_i\left(x_i, \boldsymbol{\lambda}\right) = \mathcal{L}\left(x, \lambda\right)$. The Lagrangian is shown as a continuous function, but, since the problem is discrete, we will assume for the purposes of this description that $\mathcal{L}\left(x, \lambda\right)$ is only strictly defined at integer values of x. On the other hand, λ is not limited to discrete values.

Line A in Figure 4.1(a) is the continuous solution of (4.3) for this problem and defines the continuous dual function. Lines B represent the integer solution to (4.3), defining the discrete dual in this case. It is evident that there are values for λ – for example λ^* in the figure – for which there are two consecutive integer values of x that satisfy (4.3) for the same λ. $\mathcal{L}\left(x, \lambda^*\right)$ is shown in Figure 4.1(b). At other values of λ the Lagrangian has a unique integer minimum with respect to x. $\mathcal{L}\left(x, \lambda = 4\right)$ is graphed in Figure 4.1(c), for example. Figure 4.1(d) depicts both the continuous dual and the discrete dual, which is the union of all lines B (Figure 4.1(a)) in this case. It is piecewise linear and the vertices are points that map to two distinct primal points. Hence, the dual becomes a piecewise continuous function composed of surfaces with constant gradient that join at points at which the gradients are discontinuous – owing to a jump in the primal integer minimiser.

The tenets exemplified above extend easily to problems possessing a greater number of dual variables and discrete sets other than integer. Essentially, the relations

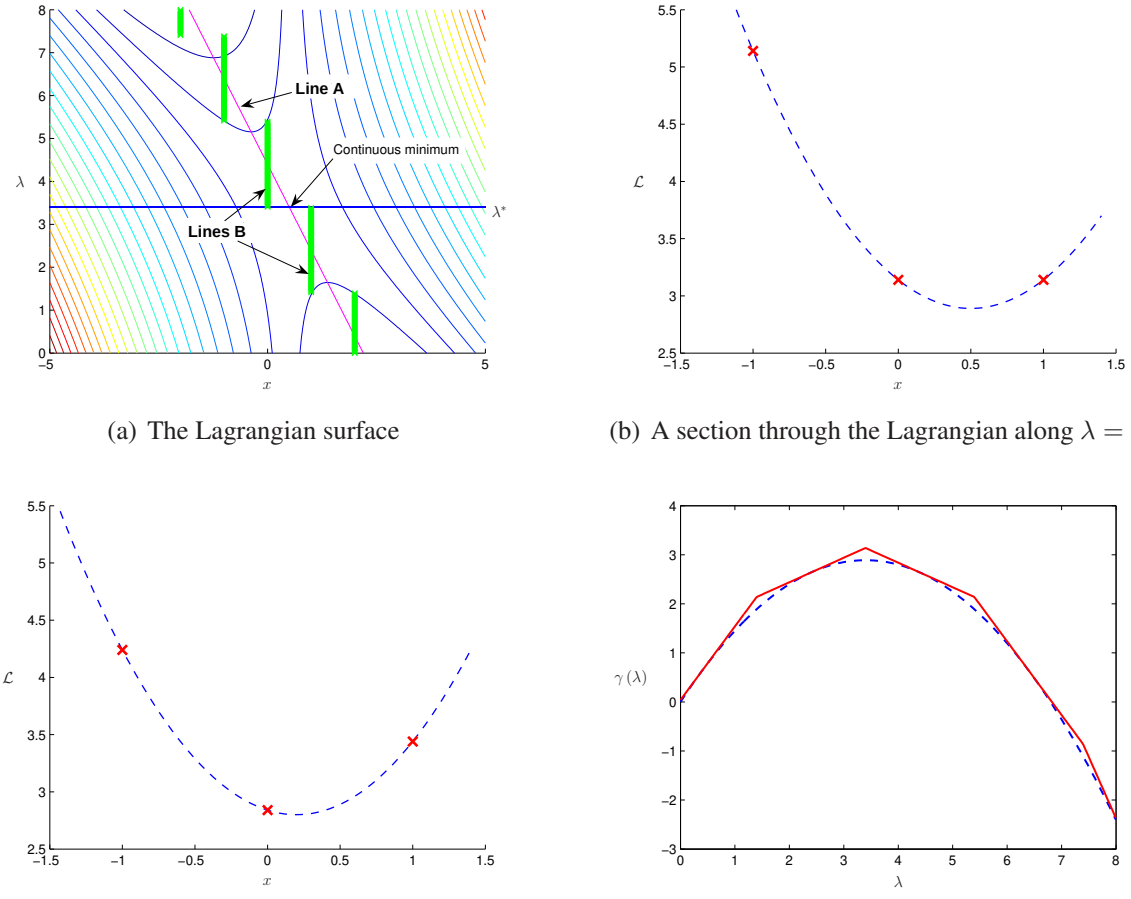
$$\mathcal{L}_i\left(x_i^A, \boldsymbol{\lambda}\right) = \mathcal{L}\left(x_i^B, \lambda\right) \qquad i = 1, 2, \cdots, n, \tag{4.5}$$

with A and B denoting consecutive indices from the allowable (ordered) discrete set, define hypersurfaces in the dual space at which the discrete minimiser of $\mathcal{L}_i\left(x_i, \boldsymbol{\lambda}\right)$ jumps from one value x_i^A to

an adjacent value x_i^B. A given dual coordinate maps to a primal discrete coordinate via a mapping derived from (4.5). The details of the mapping obviously depend on the particular approximations used to construct the Lagrangian; some examples are given in Section 4.3.2. An example of what the dual surface may look like for a fully discrete problem with two constraints is shown in Figure 4.2. The surface is concave and consists of multiple intersecting linear hyperplanes. The edges along which the hyperplanes join constitute the domains in the dual space along which one of the primal variables jumps from one allowable discrete value to another in the primal-dual relationships. On the edges themselves, the discrete primal minimiser defined by the primal-dual mapping is not unique.

The above discussion considers a purely discrete problem. If a problem is continuous in some variables $\boldsymbol{x}_c$ then, due to separability, the associated parts of the Lagrangian $\mathcal{L}_c(\boldsymbol{x}_c, \boldsymbol{\lambda})$ are simply minimised in a continuous sense when defining the dual (4.3). In this case the dual will no longer be piecewise linear in form, as in Figure 4.2.

There are two major difficulties that must be overcome in relation to the implementation of the



(a) The Lagrangian surface

(b) A section through the Lagrangian along $\lambda = \lambda^*$

(c) A section through the Lagrangian along $\lambda = 4.0$

(d) The discrete dual vs. the continuous dual

Figure 4.1: Construction of the discrete and continuous duals for a one-dimensional example with one constraint.

discrete dual method. The first is the application of a move limit suited to discrete problems. The global convergence characteristics of an algorithm utilising a discrete dual solver can be very unstable if nothing is done to limit the scale of the changes between successive designs. For continuous problems it is relatively straightforward to limit the maximum allowable step in the design domain that may be taken between one design and the next, but in the case of discrete problems an analogous strategy is difficult to implement, particularly for zero-one problems. Beckers [17] suggests two approaches based on the introduction of continuation strategies, which allow the problem definition to be modified gradually while preserving its discrete nature. The second complication concerns the difficulty inherent in maximising a piecewise linear surface. An ascent algorithm adapted for doing this is also presented by Schmit and Fleury in [16].

4.3.2 Specific examples of the discrete primal-dual mapping

The form of the discrete primal-dual mapping in an SAO subproblem depends on the form of the approximation functions used to define the approximate subproblems. We present here two specific examples. The first is the case discussed by Schmit and Fleury [16], who construct subproblems in which the $\tilde{f}_0$ have reciprocal forms and the $\tilde{f}_j$ are linear functions (the tildes indicate that the functions $\tilde{f}$ belong to the approximate subproblems). The second example has a quadratic objective and linear constraints.

Reciprocal objective and linear constraints

Consider the subproblems obtained when the objective function f_0 in (4.4) is approximated using the separable reciprocal approximation (2.25), and the constraints f_j are represented by the Taylor series expansion to first order (2.14). In this case, the primal approximate subproblems have the

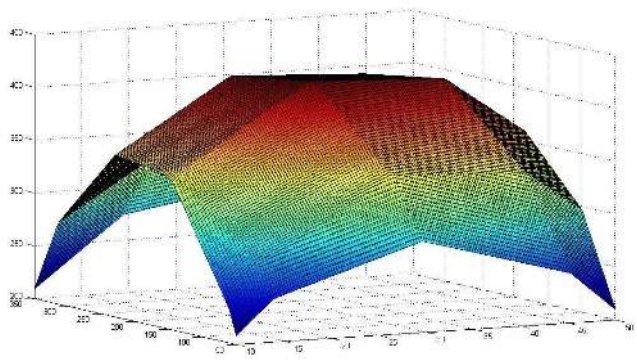


Figure 4.2: Example of a discrete dual: the surface is generated for a discrete problem with two constraints.

form

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & \tilde{f}_0(\boldsymbol{x}) = f_0\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{x_i^{\{k\}}}{x_i}\right)\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} \\
\text{subject to} \quad & \tilde{f}_j(\boldsymbol{x}) = f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \leq 0 \qquad j = 1, 2, \cdots, m, \\
& x_i \in \mathcal{D} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, 2, \cdots, n,
\end{aligned}$$

in which the set $\mathcal{D}$ represents an ordered set of discrete values. Terms bearing the superscript $\{k\}$ denote constants evaluated at the point in the design space $\boldsymbol{x}^{\{k\}}$ at which the subproblem is defined. This form of subproblem will be used to represent the variable stiffness laminate problem discussed below. The separable parts of the Lagrangian function for this problem become

$$\mathcal{L}_i(x_i, \boldsymbol{\lambda}) = \left(x_i - x_i^{\{k\}}\right)\left(\frac{x_i^{\{k\}}}{x_i}\right)\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \sum_{j=1}^{m}\lambda_j\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}.$$

Applying (4.5), with x_i^A and x_i^{A+1} denoting two consecutive allowable values of x_i from the set $\mathcal{D}$, we have

$$x_i^A \cdot x_i^{A+1} = -\left(x_i^{\{k\}}\right)^2\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} \Bigg/ \sum_{j=1}^{m}\lambda_j\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}. \tag{4.6}$$

This equation defines the subspace at which the discrete primal minimiser of $\mathcal{L}_i(x_i, \boldsymbol{\lambda})$ transitions from x_i^A to x_i^{A+1}. On this subspace, both x_i^A and x_i^{A+1} are primal minimisers of $\mathcal{L}_i(x_i, \boldsymbol{\lambda})$, which is the situation depicted in (4.1(b)). Thus, all points on the dual that lie between the surfaces defined by x_i^A and x_i^{A+1} on the one side and x_i^{A-1} and x_i^A on the other, map to the primal coordinate x_i^A; i.e. a particular point in the dual corresponds to (maps to) x_i^A if

$$x_i^{A-1} \cdot x_i^A \leq \left[-\left(x_i^{\{k\}}\right)^2\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} \Bigg/ \sum_{j=1}^{m}\lambda_j\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right] \leq x_i^A \cdot x_i^{A+1}. \tag{4.7}$$

Quadratic objective and linear constraints

Consider a second-order approximation in which all the off diagonal curvatures $c_{ij},\ i \neq j$, are set as zero:

$$\tilde{f}(\boldsymbol{x}) = f\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f_i}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n}c_{ii}\left(x_i - x_i^{\{k\}}\right)^2.$$

There are various ways that the curvatures c_{ii} may be defined; the reader is referred to [33, 61] for specific examples. For our purposes here, the particular definition of the c_{ii} is not important. It is sufficient to stipulate that $c_{ii} > 0\ \forall\ i$, so that the resulting function approximation is strictly convex. A primal approximate subproblem, constructed using the above separable quadratic approximation

for the objective function in $\mathrm{P_{NLP}}$ and linear approximations for all the constraints, is

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & \tilde{f}_0(\boldsymbol{x}) = f\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f_i}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n} c_{ii}\left(x_i - x_i^{\{k\}}\right)^2 \\
\text{subject to} \quad & \tilde{f}_j(\boldsymbol{x}) = f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \leq 0 \qquad j = 1, 2, \cdots, m, \\
& x_i \in \mathcal{D} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, 2, \cdots, n.
\end{aligned}$$

Applying (4.5) as before, the equations for the surfaces in the dual space on which the primal coordinates transition are obtained as

$$\left(x_i^A + x_i^{A+1}\right) = \left(\frac{2}{c_{ii}}\right)\left[c_{ii}x_i^{\{k\}} - \left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} - \sum_{j=1}^{m}\lambda_j\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right].$$

Therefore, all points in the dual that satisfy the following condition map to x_i^A in the primal space:

$$\left(x_i^{A-1} + x_i^A\right) \leq \left(\frac{2}{c_{ii}}\right)\left[c_{ii}x_i^{\{k\}} - \left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} - \sum_{j=1}^{m}\lambda_j\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right] \leq \left(x_i^A + x_i^{A+1}\right).$$

4.4 A closer look at the discrete dual approach

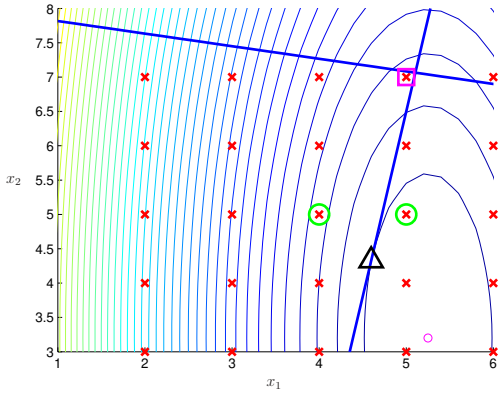
From Figure 4.1 we see that, for any $\boldsymbol{\lambda}$, the discrete minimiser of $\mathcal{L}_i(x_i, \boldsymbol{\lambda})$ will be one of the discrete points immediately adjacent to the relaxed continuous minimum. This is simply a consequence of the fact that $\mathcal{L}_i(x_i, \boldsymbol{\lambda})$ is convex with respect to x_i and is perforce constructed that way. We expect, then, that when the dual is maximised to solve the primal minimisation problem, the resulting discrete primal minimiser will also be one of the discrete points immediately surrounding the relaxed continuous minimum. We now discuss the consequences of this by referring to two convex two-dimensionality problems.

4.4.1 Two small example problems

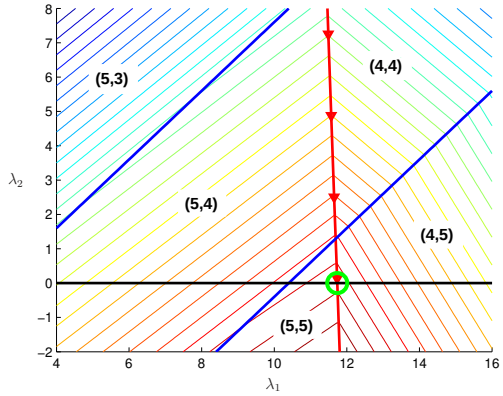
Figure 4.3 shows contour plots of a pair of two-dimensional constrained problems together with the parts of their dual surfaces containing the dual maxima. Each problem has two constraints, so the duals are also two-dimensional. Figures 4.3(a) and 4.3(b) concern a separable quadratic function subject to linear constraints. Figures 4.3(c) and 4.3(d) depict a separable reciprocal function with linear constraints. In the figures, the primal continuous constrained minima are indicated by triangles, whereas the discrete optima are indicated by squares. The feasible regions should be obvious given the position of the discrete minima. The maxima of the duals, together with the primal points to which they map, are indicated by circles.

4.4.2 Pros and cons

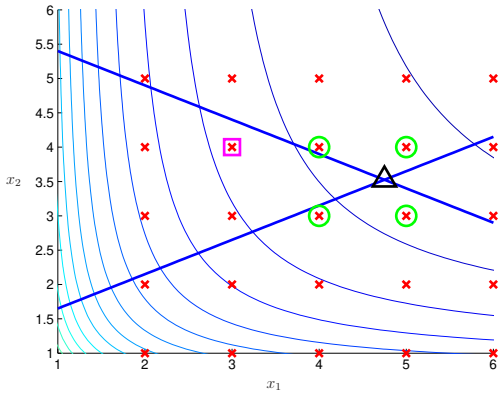
The dual maxima do not map to the discrete primal optima for either of the two problems presented above. Instead, the optimal primal points indicated by the duals are in both cases found to be



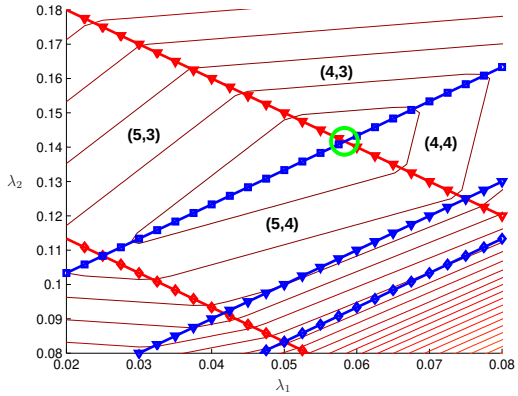
(a) Primal problem: quadratic objective, linear constraints



(b) Discrete dual: quadratic objective, linear constraints



(c) Primal problem: reciprocal objective, linear constraints



(d) Discrete dual: reciprocal objective, linear constraints

Figure 4.3: Contour plots of the primal problems and the associated discrete duals for two small 2D example problems.

immediately adjacent to the continuous optimum. Even then one is not guaranteed to find the most optimal of these points surrounding the continuous solution. In Figure 4.3(a), point $(4, 4)$ has a lower function value than either of the points indicated by the dual. Furthermore, the solutions found by the method can violate the constraints – all four of the solutions found in Figure 4.3(c) violate one of the constraints. Evidently, then, one would be ill-advised to utilise the discrete dual approach presented above to find the discrete minima for small or moderately sized problems, or in situations where constraint violations are strictly not permissible. Given this last, what then are the advantages of the method?

The primary advantage of the method is simply its efficiency. Many problems in structural optimisation are simply too large (in terms of the number of variables considered) for the traditional discrete search algorithms to be comfortably applicable. These problems are typically multimodal, even in the continuous sense, so no method short of complete enumeration guarantees that the global optimiser can be found at all. Under these conditions, the overriding concern is the use of

a technique that can arrive at high-quality solutions efficiently. Additionally, to evaluate a given design usually entails running a lengthy analysis and so a method is preferable that minimises the number of analyses required. The dual SAO method achieves this, since it is based on the construction of explicit approximations. Lastly, given the examples presented above, one might argue that an enumerative search of the points surrounding a continuous local optimum appears to represent a better strategy than the discrete dual approach. However, for large problems, even this limited enumeration is a daunting task and would require an unfeasible number of analyses.

If the number of constraints is small compared to the number of primal variables, then the number of primal points corresponding to the dual maximum is considerably less than the number of points surrounding the continuous optimum. The $[0, 1]$ topology problem (discussed below) is an extreme example of this. Each and every possible discrete solution is located at the vertex of a hypercube that surrounds the continuous optimum. Yet, for isotropic materials and one constraint, the dual approach usually returns a choice between only two primal points for the discrete solution.

4.5 Minimum compliance design: isotropic material

Topology optimisation is concerned with determining the distribution of material within a given design domain such that, ultimately, the domain will be composed of solid and void regions and the emergent structure, defined by the union of solid regions, will be optimal according to some pre-defined measure. One such measure typically used (and, indeed, the measure used herein) is that of minimal structural compliance.

4.5.1 The classical minimum compliance topology problem

The minimum compliance problem was introduced in Section 2.1.1. In the current chapter we are concerned with the spatially discretised form of the problem, represented in (2.7), in which we admit only a single constraint[2] on the allowable structural volume. The problem is restated here for convenience: we assume that the design domain is discretised by the finite element method, in which case the classical minimum compliance problem can be stated as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_1(\boldsymbol{x}) \leq 0\,, \\ & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{w}\,, \\ & x_i \in [0,1] \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{4.8}$$

In the above, f_0 denotes the objective function, which here corresponds to the structural compliance and depends on the material distribution vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$ defined over the binary $[0, 1]$ set. The symbol $\boldsymbol{K}$ represents the global assembled finite element stiffness matrix, $\boldsymbol{q}$ is the global displacement vector and $\boldsymbol{w}$ the vector of applied loads, which is assumed to be design independent.

[2]What follows is equally valid for multiple constraints.

The constraint function f_1 denotes the limit on the volume of the design, namely

$$f_1(\boldsymbol{x}) = \frac{1}{\nu_0} \sum_{i=1}^{n} \nu_i x_i - \bar{\nu} \leq 0\,, \tag{4.9}$$

in which ν_0 is the total volume of the design domain, $\bar{\nu}$ is the limiting value and ν_i is the volume of element i. In topology optimisation it is usual to consider a relaxed version of problem (4.8), in which the constraints, the objective and their first derivatives are all defined at real values of x_i between zero and one. Hence, a continuous problem is solved iteratively and it is the purpose of the so-called SIMP specialisation to drive the solution towards a solid-void design.

4.5.2 SIMP

When problem (4.8) is relaxed, the variables that, in the discrete case, describe the presence or absence of material at a point in the design space are instead interpreted as material 'densities', which serve to scale the properties of the solid isotropic material. The SIMP method is used to penalise the material of intermediate density in an attempt to generate solid-void designs as solutions to the relaxed problem. In the SIMP approach, the material properties are scaled according to

$$C_i = (x_i)^p\, C_0\,, \qquad p > 1\,, \tag{4.10}$$

where C_0 is the elasticity tensor describing the actual material. The volume of material in the design domain is not affected by the penalisation, unless volumetric penalisation is also employed (see Chapter 5).

Recalling the discussion in Section 2.1.1, the stiffness matrix $\boldsymbol{K}$, and therefore the compliance objective in (4.8), are functions of these element densities x_i. If problem (4.8) is solved purely in a zero-one sense, the space defined by an element will either be void or solid, and the resulting topology has an unambiguous interpretation. However, the relaxation of (4.10) means that elements may have fictitious intermediate densities, which interpolate for fictitious material properties because the penalisation is usually unable to get rid of all intermediate-density material. In this case it becomes more difficult to interpret the results obtained, and particularly to compare different results that are not completely solid-void (this last will be touched upon in Chapter 9). There are, therefore, advantages to solving the problem in a purely discrete sense.

4.5.3 Discrete solution

Fleury's method for solving the topology problem in a discrete sense was applied to the minimum compliance problem with some success by Beckers [17]. The problem formulation presented in [17] differs slightly from (4.8) in that an additional perimeter control constraint is included, this being one of the methods alluded to in Section 2.1.1 that can be used to combat mesh dependency. The solution procedure discussed in [17] does not use Sigmund's filter described in Chapter 3, nor is the SIMP procedure required.

In keeping with Schmit and Fleury [16], we solve (4.8) by an SAO strategy in which the approximate subproblems are constructed as described in the first part of Section 4.3.2. The compliance

objective is approximated using a separable reciprocal function, while the volume constraint is represented as a first-order Taylor series expansion. We have used Sigmund's mesh independence filter [14] instead of perimeter control to ensure mesh independence, as it simplifies the solution strategy, particularly for the FRC problem. However, it must be said that a comparison of our results for the isotropic case with those of Beckers indicates that perimeter control may contribute much to stabilising the optimisation.

Since the problem is solved in a discrete sense it is seemingly unnecessary to apply SIMP penalisation. Indeed, Beckers does not utilise SIMP; however, she does note the need for a move limit suited to binary variables, and introduces certain continuation strategies that serve to encourage global convergence. It has also been our experience that direct solution of (4.8) on the $[0, 1]$ set yields unsatisfactory results. We have found it necessary to employ both a continuation strategy and SIMP penalisation in our attempts to solve the discrete composite problem, and we discuss both of these here with reference to the isotropic case.

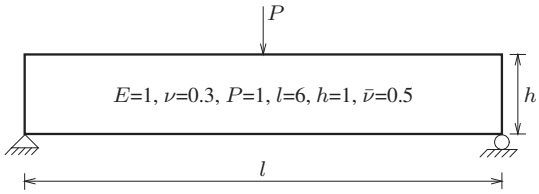
SIMP penalisation and the continuation strategy

A continuation strategy similar to one of the options suggested in [17], in which the problem remains strictly binary, was attempted initially. The problem is first solved using two allowable values $[x_{low}, x_{high}]$ for x_i that are chosen close to the allowable volume fraction $\bar{\nu}$, which defines the volume constraint f_1. As the iterations progress, these values are moved steadily apart, so that $x_{low} \rightarrow 0$ and $x_{high} \rightarrow 1$. These values assume the bound values $[\check{x}, 1]$ after a predetermined number of iterations, which means that the convergence time of the algorithm is fixed a priori.

We have not been able to generate satisfactory results using this procedure. Figure 4.4(b) is indicative of the type of designs obtained in searching for optimal (minimum compliance) topologies for the MBB beam structure, diagrammed in Figure 4.4(a). Apart from being a relatively inferior local minimum, the design is 'messy'. *Despite the use of a filter*, it contains small holes or channels (smaller than the filter radius), as well as semi-isolated solid elements connected to the bulk structure on only one edge. Additionally, this result was generated using a standard SIMP penalisation of $p = 3$. Although this strategy produces checkerboard-free results with $p < 3$, they are worse than the one depicted in Figure 4.4(b). Finally, this purely binary continuation strategy proves to be completely inadequate for the FRC problem.

A similar continuation strategy is suggested in [17], except that four discrete values are allowed for x_i. The values x_{low} and x_{high} are permitted, while $x = 0$ and $x = 1$ are also retained in all the iterations. Otherwise, the continuation proceeds as described above. This strategy represents a kind of relaxation of the binary prescription during the optimisation process, though the binary set is strictly restored in the final iteration of the algorithm. We have had more success using this idea, except that, instead of initially allowing only four discrete values for x_i, we allow a larger number. All the results that follow have been generated using an initial discretisation of 26 values in S_x, which denotes a discrete set of values between 0 and 1. Allowing a larger number of discrete values seems to be more of a boon when applied to the FRC problem than in the isotropic case.

The continuation strategy we use relies on the mapping (4.7). However, in an effort to avoid the use of lookup tables, we define (4.6) as a primal-dual mapping between $\boldsymbol{\lambda}$ and the set of discrete values between 0 and 1 through the intermediary of a set of integers S_z. The number of discrete



(a) Ground structure for the MBB beam

(b) Mesh: 150×50 elements, $f_0^* = 246.45$, $p = 3$, $t_{cpu} = 944$ sec.

Figure 4.4: Ground structure and 'optimal' discrete topology for the isotropic MBB beam. The optimal design is found using a continuation strategy based on a binary mapping, with convergence occurring after 105 iterations.

values in S_x and S_z is N, those in S_z ranging from 0 to $N-1$. We relate the values in S_x to those in S_z by

$$x_i^A = \frac{z_i^A}{N-1}\,. \tag{4.11}$$

From (4.6), if z_i^A and z_i^{A+1} are consecutive integers in S_z, then a reliable mapping can be defined for x_i^A by first determining z_i^A as

$$z_i^A = \text{ceil}\left\{\text{root}_+\left[\left(\frac{r}{N-1}\right)^2 + \frac{r}{N-1} - y = 0\right]\right\},$$

in which y is the right-hand side of the equality (4.6). The notation root_+ simply denotes the positive root of the quadratic function in brackets, while the ceil operator is a standard MATLAB[3] operator that rounds up its argument to the next highest integer. Having determined z_i^A, the relationship (4.11) is used to calculate the corresponding x_i^A. Of course, the only reason the integer intermediaries are necessary is because of the ceil operator; what we have in effect is an update equation for the discrete problem.

Now, the continuation is accomplished by mapping the constant set S_x to a variable set $S'_{x'}$, whose elements also lie distributed between 0 and 1, through the use of the sigmoidal function

$$x'_i = \frac{1}{1 + e^{-b(x_i - 0.5)}}\,. \tag{4.12}$$

The shape of the sigmoid is altered by changing the parameter b. Figure 4.5 depicts how the distribution of the elements in $S'_{x'}$ changes as the sigmoid is made steeper by increasing b. We

[3]We use Sigmund's 99-line MATLAB topology code [9] as a backbone for the algorithm we employ, though it is greatly modified for the FRC problem.

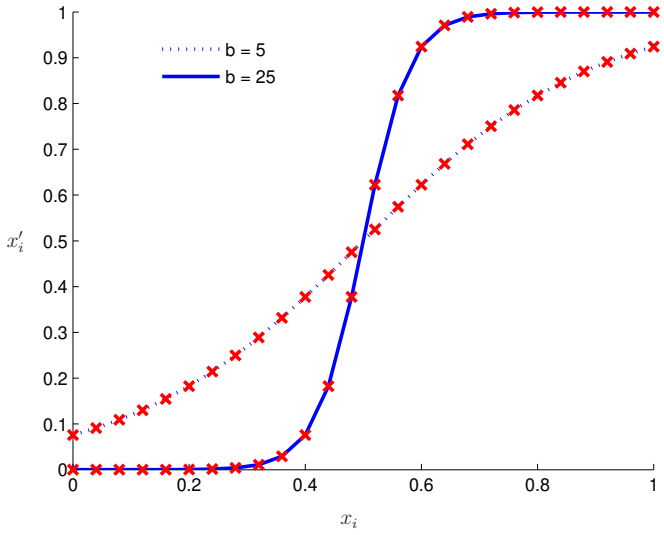


Figure 4.5: The sigmoidal function accomplishes the mapping of the evenly distributed discrete points x_i to points x_i', whose distribution is biased more towards $[0, 1]$.

solve the compliance problem (4.8) on $S'_{x'}$. Additionally, we define an upper bound $\hat{x}$ close to 1 so that all $x_i' > \hat{x}$ are made 1. Similarly, all values of $x_i' < \check{x}$ are made $\check{x}$. Algorithmically, therefore, when determining the discrete mapping, only the portion of the sigmoid between $\check{x}$ and $\hat{x}$ is considered. In this case the update equation becomes

$$z_i^A = \text{ceil}\left\{\left[(1-N)\ln(r)\right]/b\right\}, \tag{4.13}$$

where

$$r = \text{root}_+\left[C_2(r)^2 + C_1(r) + C_0 = 0\right]. \tag{4.14}$$

The coefficients in (4.14) are expressed in terms of the transformation (4.12) as

$$\begin{aligned} C_2 &= e^b \cdot e^{-b/(N-1)}, \\ C_1 &= e^{0.5b}\left[1 + e^{-b/(N-1)}\right], \\ \text{and } C_0 &= 1 - 1/y\,. \end{aligned}$$

Figure 4.6 shows some of the optimal topologies gained when using the continuation strategy just described, in combination with different values for the SIMP penalty parameter p. Figures 4.6(a) and 4.6(b) are examples in which the half-beam mesh discretisation is 90×30 elements and the filter radius for Sigmund's mesh independence filter is 2.1. Figure 4.6(c), on the other hand, is generated using a mesh of 150×50 elements and a filter radius of 2.5 elements. In all cases, the same continuation is used and termination occurs at a purely binary solution after 100 iterations. The objective function value and penalty parameter for each design are stated in the figure, as well as the cpu time required. For each design, the limiting volume fraction was set at $\bar{\nu} = 0.5$. Evidently, good solutions can be obtained with values of $p < 3$, although we still fail to generate satisfying results with $p = 1$ (i.e. without SIMP penalisation). For the FRC problem described below, however, higher values of the penalty were necessary, and we use $p = 3$ throughout.

(a) $f_0^* = 280.13$, $p = 1$, $t_{cpu} = 199$ sec.

(b) $f_0^* = 186.95$, $p = 2$, $t_{cpu} = 188$ sec.

(c) Mesh: 150×50 elements, $f_0^* = 188.13$, $p = 3$, $t_{cpu} = 982$ sec.

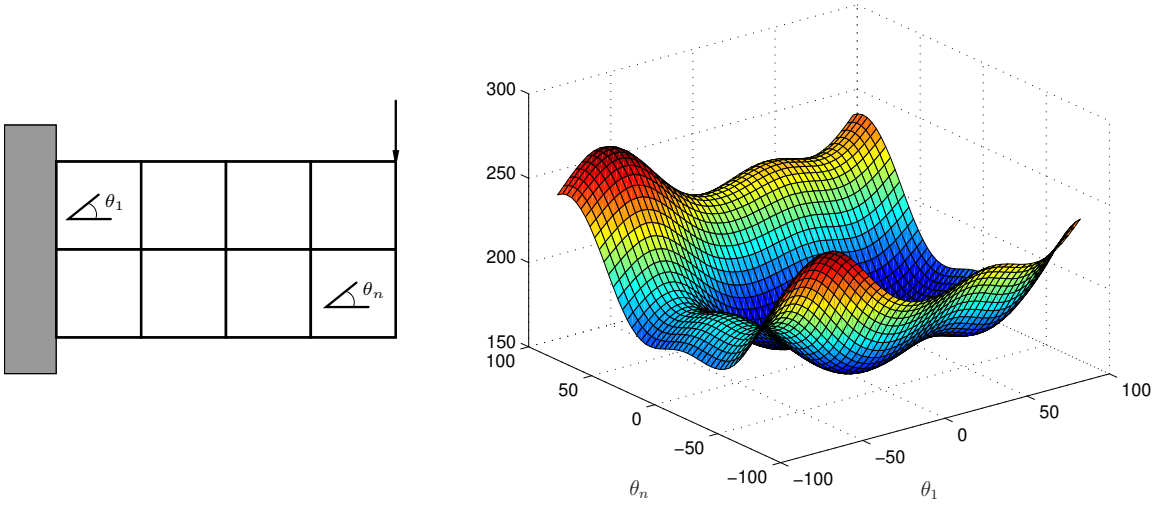
Figure 4.6: Optimal $[0, 1]$ topologies for the isotropic MBB beam using a continuation strategy based on a relaxed discrete set distributed according to a variable sigmoidal mapping.

4.6 Compliance and fibre angle optimisation: FRC laminates

In a continuous sense, the problem of finding the optimal spatially varying fibre orientation for minimal compliance of an FRC laminate is heavily multimodal, even when spatially discretised using FEM. Consider, for instance, the discretised cantilevered structure shown in Figure 4.7(a), in which the fibre direction in each element is allowed to vary continuously between $+90$ and -90 degrees. Figure 4.7(b) plots the variation in compliance for the structure as the fibre orientations θ_1 and θ_n for the elements labelled in Figure 4.7(a) are varied. As can be seen from the figure, the relationship between compliance and fibre direction has an underlying sinusoidal character.

The problem we wish to consider here is the concurrent optimisation of topology and fibre orientation. That is: the density of each element in the mesh is subject to change, and so is the fibre orientation in each element. As in the examples depicted above for isotropic structures, the material contribution of each element is required over the binary $[0, 1]$ set. However, for the material that is present in the design, the optimal fibre orientation is also desired. Both the fibre direction and the element density (via the SIMP formulation) are inherent in the material properties of each element. Thus, the two types of primal variables (x_i and θ_i) are intrinsically coupled, and true optima cannot be found by first solving the isotropic material distribution problem and then solving the orthotropic fibre orientation problem for the material that remains.

The traditional approach to topology optimisation is to consider the relaxed continuous form of a problem, and then to effect $[0, 1]$ material distributions using penalisation. When considering the relaxed continuous form of the FRC problem, the primary complication lies in the formulation of a material description that incorporates both the effect of element density as well as fibre direction, and simultaneously encourages the element densities to discrete values via penalisation. A technique known as discrete material optimisation (DMO) has been introduced that accomplishes just this. Previous implementations of DMO have focused on solving the relaxed continuous material distribution problem, and have used materials with low stiffness to approximate voids in the design domain. We here adapt DMO to solve the discrete problem in a way that allows true voids to be generated (at the expense of incorporating many additional constraints).



(a) Discretised cantilever structure (b) Variation of compliance as a function of the fibre directions in two elements

Figure 4.7: An example of the variation in compliance with fibre orientation for a discretised FRC structure in which the elemental fibre directions can vary continuously.

4.6.1 Discrete material optimisation

Stegmann and Lund [57] have introduced the possibility of concurrently optimising for fibre direction in a topology infrastructure that seeks an optimal material distribution within a design domain. They accomplish this by considering the material properties assigned to a particular element to be a weighted combination of a fixed number of candidate materials. In the context of orthotropic materials, this means that the space defining the allowable elemental fibre orientations is discretised. The elasticity tensor of each candidate material is derived from the FRC material and calculated using a different fibre orientation. The material constituting a given element is made up of a limited fixed number of these candidate materials, as is depicted in Figure 4.8. Each candidate material has a pre-defined fibre angle and its own weight factor, which in what follows can be interpreted as a material density. The task of the optimiser is to find a zero-one solution for the weights, which amounts to selecting the single optimal material (i.e. fibre direction) for the element. If one of the weights is driven to unity, then the weights associated with all the other candidate materials within the element must be driven to zero. Our adaptation of DMO allows all the weights associated with a given element to be driven to zero, which results in a void and thereby facilitates solid-void topology optimisation.

We will discuss the above more explicitly in the following section. For now, note that there are many ways that this “weighted combination” of materials can be defined. Also, if the SIMP approach is used to encourage solid-void solutions, the complication arises of how best to penalise the element density (see [57] for further details).

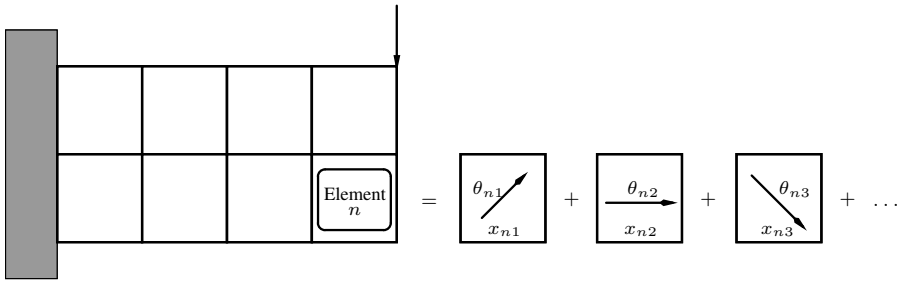


Figure 4.8: The DMO formulation of elemental material properties as a function of many candidate materials.

4.6.2 Our method for discrete topology and fibre angle design

We use the simplest relationship suggested in [57] to define the material characteristics associated with an element i in the finite element mesh:

$$C_i = \sum_{j=1}^{n_\theta} (x_{ij})^p \, C_0 \, (\theta_j) \, . \tag{4.15}$$

The elasticity tensor of a given orthotropic material is calculated at n_θ different (user-defined) angles. C_i is a penalised weighted sum of the resulting set of material characteristics, where each weight x_{ij} is penalised individually. The weights can still be considered as densities (or, more properly, material occupancies) in the sense that the volume of material in the design domain is now given by

$$V = \sum_{i=1}^{n} \sum_{j=1}^{n_\theta} x_{ij} \, . \tag{4.16}$$

Naturally, it makes no sense to have element densities of greater than one, so the set of weights pertaining to a single element must also satisfy

$$V_i = \sum_{j=1}^{n_\theta} x_{ij} \leq 1 \, . \tag{4.17}$$

Stegmann and Lund rejected (4.15) on the grounds that it fails to allow a sufficiently discrete selection of the optimal candidate material within an element when the problem is optimised in the continuous sense and the SIMP approach is used. The difficulty stems from the fact that they prefer not to take (4.17) into account explicitly, since this greatly increases the complexity of the problem, contributing an additional n constraints. They instead find other methods of implicitly satisfying (4.17), albeit as equality constraints.

We solve the topology problem (4.8) discretised by a finite element mesh containing n elements and we use (4.15) as our material description. We explicitly retain all n constraints (4.17) and the problem is additionally subject to a global constraint on the total volume (4.16) of the design. There are, therefore, $n \times n_\theta$ design variables and $(n + 1)$ constraints. We find the optimum iteratively using the discrete dual SAO approach, in which the objective function f_0 is approximated by (2.25) and each constraint f_j is approximated by a linear truncated Taylor series expansion (equation (2.14) up to the linear term).

This last is one of the advantages of using the DMO approach for the FRC problem. Although the dependency of the compliance on fibre orientation is sinusoidal in nature for the continuous problem, in the DMO formulation the effect of fibre orientation is reflected instead in terms of the candidate material densities. The variation in compliance as a function of these densities is locally reciprocal in nature, so the same function approximations can be used as for the isotropic problem. Moreover, since the additional elemental constraints are also linear, the same primal-dual relationships can be used, as well as the same continuation strategy. Indeed, we use essentially the same optimisation infrastructure to solve the FRC problem as we used to solve the discrete isotropic problem described in Section 4.5, except that the dual maximisation scheme is modified to account for the greater dimensionality of the dual and to take advantage of its special structure.

4.6.3 Maximising the dual

When considered as a discrete DMO formulation, the combined minimum compliance and optimal local fibre orientation problem for an FRC laminate is expressed as

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \\
\text{subject to} \quad & f_i(\boldsymbol{x}) = \sum_{j=1}^{n_\theta} x_{ij} \leq 1 \qquad\qquad i = 1, 2, \cdots, n, \\
& f_{n+1}(\boldsymbol{x}) = \frac{1}{\nu_0} \sum_{i=1}^{n} \left(\nu_i \sum_{j=1}^{n_\theta} x_{ij} \right) \leq \bar{\nu}\,, \\
& x_{ij} \in [0, 1] \qquad\qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, n_\theta,
\end{aligned} \tag{4.18}$$

where the $\boldsymbol{q}_i$ are the elemental nodal displacement vectors, ν_i denotes the volume of element i, ν_0 the volume of the design domain and $\bar{\nu}$ the desired limit placed on the volume of the optimal design. There are n elements in the finite element mesh and n_θ candidate materials per element, the material properties for each being defined at a different fibre angle θ. The elemental stiffness matrices are given by

$$\boldsymbol{K}_i = \int_{\nu_i} \boldsymbol{B}_i^T \boldsymbol{C}_i \boldsymbol{B}_i \, d\nu_i$$

in terms of the elemental strain-displacement operators, and the elemental elasticity matrix is calculated in accordance with (4.15). When constructing the subproblems we apply the reciprocal approximation to the objective function and a linear Taylor expansion to all the constraints (refer

to Section 4.3.2). Thus, the Lagrangian for any of the subproblems has the form

$$\begin{aligned}\tilde{\mathcal{L}}\left(\boldsymbol{x},\boldsymbol{\lambda}\right) = f_0\left(\boldsymbol{x}^{\{k\}}\right) &+ \sum_{i=1}^{n}\left\{\sum_{j=1}^{n_\theta}\left(x_{ij}-x_{ij}^{\{k\}}\right)\left(\frac{x_{ij}^{\{k\}}}{x_{ij}}\right)\left(\frac{\partial f_0}{\partial x_{ij}}\right)^{\{k\}}\right\} \\ &+ \sum_{i=1}^{n}\lambda_i\left[f_i\left(\boldsymbol{x}^{\{k\}}\right)+\sum_{j=1}^{n_\theta}\left(x_{ij}-x_{ij}^{\{k\}}\right)\right] \\ &+ \lambda_{n+1}\left[f_{n+1}\left(\boldsymbol{x}^{\{k\}}\right)+\sum_{i=1}^{n}\left\{\alpha_i\sum_{j=1}^{n_\theta}\left(x_{ij}-x_{ij}^{\{k\}}\right)\right\}\right],\end{aligned} \tag{4.19}$$

in which the following identities hold:

$$\left(\frac{\partial f_i}{\partial x_{ij}}\right)^{\{k\}} = 1 \qquad \forall \qquad i = 1,2,\cdots,n,$$

$$\text{and} \quad \left(\frac{\partial f_{n+1}}{\partial x_{ij}}\right)^{\{k\}} = \alpha_i = \frac{\nu_i}{\nu_0}.$$

The Lagrangian is separable in the primal variables x_{ij} and so, following the discussion presented in Section 4.3.2, the primal-dual mappings are determined using

$$x_{ij}^{A}\cdot x_{ij}^{A+1} = -\left(x_{ij}^{\{k\}}\right)^2\left(\frac{\partial f_0}{\partial x_{ij}}\right)^{\{k\}} \Big/ \left(\lambda_i+\alpha_i\lambda_{n+1}\right). \tag{4.20}$$

The primal-dual relationship for a given primal variable x_{ij}, which represents the density of a single candidate material j within element i, is a function of only the dual variables associated with constraint i (limiting the density of the i^{th} element) and the multiplier associated with the global volume constraint λ_{n+1}. This allows us to write the Lagrangian for a subproblem in a 'partially separable' form as the sum of $i+1$ terms. Each of the first n terms is associated solely with one element in the finite element mesh, whereas the final term is a function purely of λ_{n+1} and does not involve the primal variables

$$\begin{aligned}\tilde{\mathcal{L}} = \sum_{i=1}^{n}\Bigg\{\lambda_i f_i\left(\boldsymbol{x}^{\{k\}}\right) &+ \sum_{j=1}^{n_\theta}\left[\left(x_{ij}-x_{ij}^{\{k\}}\right)\left(\frac{x_{ij}^{\{k\}}}{x_{ij}}\right)\left(\frac{\partial f_0}{\partial x_{ij}}\right)^{\{k\}}\left(\lambda_i+\alpha_i\lambda_{n+1}\right)\left(x_{ij}-x_{ij}^{\{k\}}\right)\right]\Bigg\} \\ &+ \left[f_0\left(\boldsymbol{x}^{\{k\}}\right)+\lambda_{n+1}f_{n+1}\left(\boldsymbol{x}^{\{k\}}\right)\right].\end{aligned}$$

The primal-dual relationships expressed in (4.20) are used as the basis for the sigmoidal mapping in the continuation strategy discussed in Section 4.5.3. When the mapping is applied to accomplish the minimisation of the Lagrangian required in the definition of the dual (4.3), the dual function inherits the same separable form as $\tilde{\mathcal{L}}$, namely

$$\begin{aligned}\tilde{\gamma}\left(\boldsymbol{\lambda}\right) = \sum_{i=1}^{n}\Bigg\{&\lambda_i f_i\left(\boldsymbol{x}^{\{k\}}\right)+ \\ &\sum_{j=1}^{n_\theta}\left[\left(x_{ij}^{\dagger}-x_{ij}^{\{k\}}\right)\left(\frac{x_{ij}^{\{k\}}}{x_{ij}^{\dagger}}\right)\left(\frac{\partial f_0}{\partial x_{ij}}\right)^{\{k\}}\left(\lambda_i+\alpha_i\lambda_{n+1}\right)\left(x_{ij}^{\dagger}-x_{ij}^{\{k\}}\right)\right]\Bigg\}+ \\ &\left[f_0\left(\boldsymbol{x}^{\{k\}}\right)+\lambda_{n+1}f_{n+1}\left(\boldsymbol{x}^{\{k\}}\right)\right],\end{aligned}$$

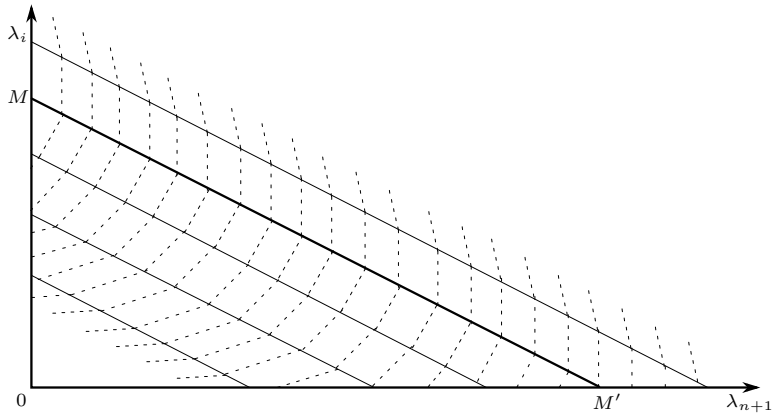


Figure 4.9: The structure of the sub-duals in the discrete combined FRC topology and fibre orientation problem.

where $x_{ij}^{\dagger}$ here represents the dependence of the primal variables on the dual coordinates:

$$x_{ij}^{\dagger} = x_{ij}\left(\lambda_j, \lambda_{n+1}\right).$$

The dual is only weakly coupled, and its maximisation may be accomplished using a series of one-dimensional search procedures. We apply a linesearch maximisation scheme to the dual in the coupling variable λ_{n+1}. For each particular value for λ_{n+1}, the maxima of the i parts of the dual with respect to their independent variables λ_i may be calculated independently[4], also using only a linesearch strategy.

Each of the i parts of the dual has the form depicted in Figure 4.9. Since the considered FRC problem is fully discrete in the primal variables, each sub-dual is a surface composed of linear planes that intersect on lines marking the discrete transition of one of the primal variables. The equations of these lines in the $(\lambda_i, \lambda_{n+1})$ space are given, via a manipulation of (4.20), as

$$\lambda_i = -\alpha_i \lambda_{n+1} - \frac{\left(x_{ij}^{\{k\}}\right)^2}{x_{ij}^{A} \cdot x_{ij}^{A+1}} \left(\frac{\partial f_0}{\partial x_{ij}}\right)^{\{k\}}, \tag{4.21}$$

in which the last term is simply a constant. The lines are parallel and intersect both the λ_i and λ_{n+1} axes. This means that between $\lambda_{n+1} = 0$ and $\lambda_{n+1} = M'$ (refer to Figure 4.9), the maximum of the dual with respect to λ_i lies on the same line, or ridge. We use a gradient-only linesearch strategy to maximise the dual on each of the $n+1$ directions. For a given direction i the linesearch strategy begins by locating two points λ_i^A and λ_i^B in sub-dual i at which the partial derivatives of the dual $\partial\gamma/\partial\lambda_i$ have opposite signs. The gradients thus calculated are used to construct linear approximations of the dual in direction i, and the subsequent point λ_i^C at which the sub-dual is evaluated is determined by the intersection of these linear functions. The point λ_i^C replaces either λ_i^A or λ_i^B in the following iteration of the linesearch, depending on the sign of the partial derivative evaluated there. Since the dual is concave on all i, if $\partial\gamma/\partial\lambda_i \leq 0$ at $\lambda_i = 0$, then $\lambda_i = 0$ represents the maximum of sub-dual i in λ_i and, provided that the primal subproblem is feasible, there will

[4]Maximisation of the separate dual parts may be carried out in parallel, if one has the facilities to do so.

always be a point at which $\partial\gamma/\partial\lambda_i \leq 0$ with λ_i large enough. Given that the dual is both concave and piecewise linear, this represents a straightforward strategy to locate the apex at which the dual is maximised.

4.7 Numerical results

We model structures utilising a shear-weak material description (C_0 is based on the material characteristics given in [56]). In this case we expect the uniaxial fibre directions in an optimal design to be aligned with the local major principal stress direction. In the case where the topology optimiser yields a design composed of struts, we expect the fibre directions to be aligned with the axes of the struts.

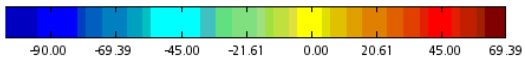
In Figure 4.10 we present results for a Michell truss with a centre load, which may be compared to the results presented in [56]. In Figure 4.11 we present results for a cantilever beam subjected to a distributed load, which may be compared to results presented in [53] and [57]. In the figures, the white elements are void, and the remaining elements are solid with a particular fibre angle as indicated by the colour keys (Figures 4.10(b) and 4.11(b)). The solution algorithm was implemented in MATLAB and run on an ACER 1.73 GHz laptop. The solution time for the problem depicted in Figure 4.10(d) was 2.25 hours (91 iterations), whereas the problem shown in Figure 4.11(c) required 28 minutes (90 iterations).

Lastly, we present results for the MBB beam structure depicted in Figure 4.4(a). We take advantage of symmetry and model only the right-hand half of the design space, so the fibre angles denoted by the colour bars in Figure 4.12 indicate the optimal fibre directions determined for the right-hand half of the beams. The left-hand side is simply a mirror image of the right. The colour key in Figure 4.12(d) is also the key associated with the solutions depicted in Figure 4.13.

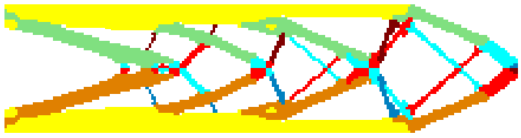
4.8 Conclusions

A novel approach to performing coupled optimal topology and optimum fibre orientation design of planar fibre reinforced composite components was presented. The discrete material optimisation technique, detailed by Stegmann and Lund, is used to formulate the problem as an optimisation problem in which only material densities are varied. Angular fibre orientation does not enter into the problem explicitly. Instead, a number of isotropic candidate materials are defined, the elasticity matrix for each corresponding to the elasticity matrix of the FRC material evaluated at one of a discrete set of allowable fibre angles. The elasticity matrix for an element in the finite element mesh is determined as a weighted combination of these candidate materials. The allowable values for the weights are limited to the discrete $[0, 1]$ set, and elemental constraints are used to ensure that an element in the optimal design is either void or composed of only a single candidate material. The discrete dual sequential approximate optimisation algorithm introduced by Schmit and Fleury is applied to obtain the solid-void designs in which the solid material has a fibre direction that varies spatially. The algorithm utilises a dual solver; for the problem discussed, the dual subproblems are piecewise linear and have a larger dimensionality than the primal subproblems. However, because they have a separable structure, the dual subproblems can nevertheless be solved efficiently.

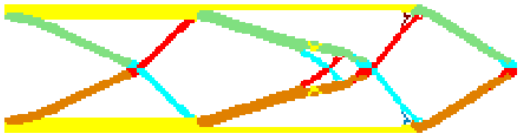
(a) Volume and loading



(b) Colour key for fibre directions

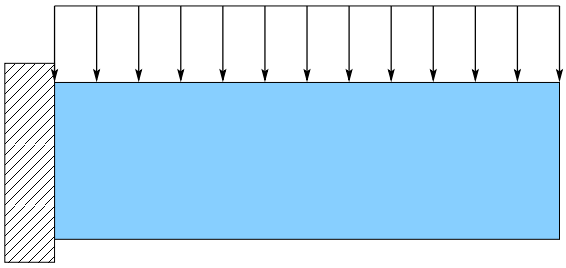


(c) Volume constraint: $\bar{\nu} = 50\%$; mesh discretisation: 200×50 elements

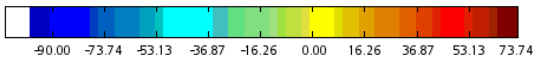


(d) Volume constraint: $\bar{\nu} = 30\%$; mesh discretisation: 240×60 elements

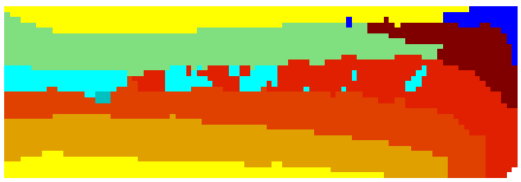
Figure 4.10: Results obtained for the combined optimisation of topology and fibre orientation for the Michell truss test problem.



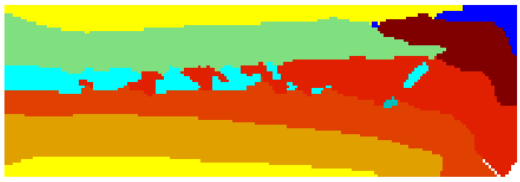
(a) Volume and loading



(b) Colour key for fibre directions



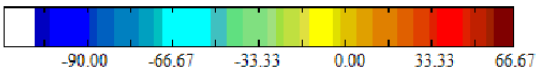
(c) Volume constraint: $\bar{\nu} = 100\%$; mesh discretisation: 96×32 elements



(d) Volume constraint: $\bar{\nu} = 100\%$; mesh discretisation: 180×60 elements

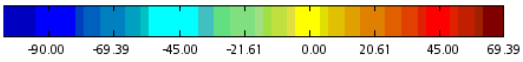
Figure 4.11: Results obtained for the combined optimisation of topology and fibre orientation for the cantilever beam test problem.

(a) Fibre angle discretisation: $\theta_D = 6$



(b) Colour key for fibre directions

(c) Fibre angle discretisation: $\theta_D = 8$



(d) Colour key for fibre directions

Figure 4.12: Results obtained for the combined optimisation of topology and fibre orientation for the MBB beam with a half-beam mesh discretisation of 60×20 elements, and a maximum volume constraint of $\bar{\nu} = 50\%$.

(a) Volume constraint: $\bar{\nu} = 40\%$

(b) Volume constraint: $\bar{\nu} = 50\%$

Figure 4.13: Results obtained for the combined optimisation of topology and fibre orientation for the MBB beam with a half-beam mesh discretisation of 150×50 elements, and a fibre angle discretisation of $\theta_D = 8$.

Chapter 5

Compliance minimisation subject to a concave volume constraint

The work presented here originates from a paper titled "On concave constraint functions and duality in predominantly black-and-white topology optimisation" [35]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa.

5.1 Abstract

We study the 'classical' discrete, solid-void or black-and-white topology optimisation problem, in which minimum compliance is sought subject to constraints on the available material resource. We assume that this problem is solved using methods that relax the discreteness requirements during intermediate steps, and that the associated programming problems are solved using sequential approximate optimisation (SAO) algorithms based on duality. More specifically, we assume that the advantages of the well-known Falk dual are exploited. Such algorithms represent the state of the art in (large-scale) topology optimisation when multiple constraints are present, an important example being the method of moving asymptotes (MMA).

We depart by noting that the aforementioned SAO algorithms are invariably formulated using strictly convex subproblems. We then numerically illustrate that strictly concave constraint functions, like those present in *volumetric penalisation*, as recently proposed by Bruns and co-workers, may increase the difficulty of the minimum compliance problem when strictly convex approximations are used in the SAO algorithm. In turn, volumetric penalisation methods are of notable importance, since they seem to hold much promise for generating predominantly solid-void or discrete designs.

We then argue that the nonconvex problems we study may in some instances be solved efficiently using dual SAO methods based on *nonconvex* (strictly concave) approximations that exhibit monotonicity with respect to the design variables. Indeed, for the minimum compliance problem resulting from SIMP-like volumetric penalisation, we show explicitly that convex approximations are not necessary. Even though the volumetric penalisation constraint is strictly concave, the max-

imum of the resulting dual subproblem still corresponds to the optimum of the original primal approximate subproblem.

5.2 Introduction

Topology optimisation seeks to introduce topological features into a structure, such that the distribution of material is optimal in some sense, subject to any number of linear and/or nonlinear inequality constraints. From an algorithmic point of view, this discrete programming problem is very difficult. Not only are the design variables discrete and the design region possibly disjointed, but, more often than not, the dimensionality of the problem is (very) high. In recent years, this problem has nevertheless been solved regularly in an approximate sense. Broadly speaking, this is mainly due to two important 'ingredients'.

The *first* ingredient is the very popular solid isotropic material with penalisation (SIMP) method. Independently proposed by Bendsøe [18] and Rozvany and Zhou [19], this method to some extent overcomes certain of the difficulties associated with discrete design variables. In the SIMP method, an approximate, relaxed, continuous programming problem is solved. A penalised material model is used in the definition of the objective function; this effects partially solid-void or black-and-white designs through the penalisation of intermediate densities, e.g. see Bendsøe [42] for details.

The *second* ingredient is the use of sequential approximate optimisation (SAO) algorithms based on dual principles (which include the sometimes equivalent [43] OC methods). These SAO algorithms are often based on *strictly convex* and *separable* primal approximate subproblems, which may be transformed into highly efficient dual subproblems when the number of constraints m is far less than the number of design variables n. Examples of successful convex dual SAO algorithms are, amongst others, the well-known CONLIN algorithm [4] and its generalisation, the method of moving asymptotes (MMA) [3, 32].

We should at this point elaborate on the discreteness requirement of the topology optimisation problem, since this has important implications for the first ingredient: whereas SIMP-like penalisation is quite efficient in generating solid-void or black-and-white designs, this efficiency may decrease notably when filtering methods are introduced into the problem formulation. In turn, filtering methods are used in topology optimisation for good reason, since they overcome the dependence of the solution on mesh discretisation. Undoubtedly, the most popular filtering methods use filtering of the design *sensitivities*, proposed by Sigmund [14, 15].

Recently, Sigmund [37] has forcefully made the point that the discreteness requirement should be taken very seriously, using a nano-optical device as an example – the effectiveness of the device is degraded and notably changed in the presence of grey (that is, intermediate-density) material, to the extent that even post-processing methods become troublesome. In many practical situations we have become accustomed to accepting designs for which the black-and-white fraction is only in the region of 60%. Hence, methods that are able to generate predominantly black-and-white results should be considered to be of fundamental importance.

In an attempt to generate predominantly black-and-white results, Bruns [39] has recently proposed the introduction of a penalty into the volume constraint, rather than penalisation of the objective function, thereby building on the work of Zhou and Rozvany [62], Guedes and Taylor [63], and

Rietz [64]. He denoted this volumetric penalisation method, in which intermediate-density material is volumetrically unattractive, the SINH method (pronounced 'cinch'), where the name reflects the use of the hyperbolic sine function in the constraint penalisation. In combination with penalisation of the volumetric constraint function, Bruns also employed filtering of the element densities, rather than of the design sensitivities. However, note that these aspects are not dependent on each other; it is perfectly possible to combine the use of volumetric penalisation via the hyperbolic sine function (or any other penalty method for that matter) with sensitivity filtering, and density filtering may of course be combined with classical SIMP-like penalisation. The approach proposed by Bruns does result in predominantly black-and-white solutions, and it has the added advantage that the resulting optimisation problem is regularised and consistent.

Bruns then solves his SINH problem using the aforementioned method of moving asymptotes (MMA) proposed by Svanberg, which has been very widely used in topology optimisation. In MMA, the convex approximate subproblems are based on linear first-order Taylor series expansions formulated in terms of reciprocal-like intervening (intermediate) variables. However, as said, volumetric penalisation results in a (strictly) *concave* constraint function. This, in turn, implies that volumetric penalisation may be expected to complicate the optimisation process *per se*, since strictly convex subproblems are used to approximate a nonconvex problem. We will numerically demonstrate this herein. From an algorithmic and computational point of view, concave constraint functions may indeed complicate the optimisation process if problem solution is effected by algorithms constructed using convex arguments.

However, an important advantage of convexifying the subproblems in the first place is that they are easily amenable to solution via dual methods (as is done in MMA). In turn, the most popular of these – and probably the most effective by far if the number of design variables n is far greater than the number of constraints m – is the dual defined by Falk [2]. In Falk's definition of the dual, discussed in Section 2.3.1, the upper and lower bound constraints on the design variables do not explicitly have to be included as constraints in the definition of the Lagrangian. For convex programming problems, Falk demonstrated that maximisation of his dual corresponds to minimisation of the original primal problem [2]. It is now widely recognised that the use of *strictly convex* approximate subproblems is a *sufficient* condition to ensure that the primal and dual solutions are identical. However, perhaps because of the ubiquity of the algorithms that depend on convexification, it is not as often recognised that strict convexity is *not* a *necessary* condition. Certainly, dual algorithms based on separable nonconvex approximations are not widely used, if at all, but much of what was developed by Falk [2] holds for less restrictive classes of problems.

We do not intend to provide additional proofs that the Falk dual is useful for other classes of problems; this will require the development of a proof for many a different problem. Instead, we draw on the argument put forward in Section 2.3.2, that the proofs presented by Falk for convex programs hold whenever a problem satisfies certain attributes, and that a problem need not be convex to satisfy these conditions. We here show specifically that these attributes are fulfilled by a particular form of nonconvex mathematical programming problem that is useful in minimum compliance topology optimisation when volumetric penalisation is considered. To construct the problem we use a SINH-like method, but we effect penalisation of the volume constraint using a more traditional power law approximation, such as is normally used in SIMP. Approximation of the constraint function is then straightforward. For the sake of brevity and simplicity, we retain

filtering of the design sensitivities[1]; this may result in less grey material.

The development of this chapter is as follows: In Section 5.3 we summarise the minimum compliance problem; this includes a reflection on the SIMP method and volumetric penalisation methods. A brief note regarding the approximations used to develop the SAO subproblems for this study is presented in Section 5.4. In Section 5.5 we show explicitly that the nonconvex problem resulting from the application of SIMP-like volumetric penalisation can be solved using the Falk dual. Thereafter, we briefly discuss the computational implications of volumetric penalisation in Section 5.6, and we present results generated by convex and nonconvex dual algorithms alike. Finally, in Section 5.7 we offer some conclusions and recommendations for future work.

5.3 The classical minimum compliance problem

The classical minimum compliance topology optimisation problem for linear elastostatic structures was introduced in Sections 2.1.1 and 2.1.2. In keeping with the description advanced there, it is explicitly assumed that the structural design domain is discretised using the very popular finite element method (FEM), and that only one constraint is present, which represents a prescribed limit on the structural volume[2]. To facilitate numerical solution, the discretised form of the problem (2.7) is replaced by the relaxed continuous problem (2.9), which is re-expressed here for convenience:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_1(\boldsymbol{x}) \geq 0\,, \\ & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{w}\,, \\ & 0 < \check{x} \leq x_i \leq 1 \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{5.1}$$

where the lower bound $\check{x} > 0$ is introduced for the sake of numerical stability (it prevents disjointed regions, etc.). Note that, for the sake of continuity with the work of Falk presented in Section 2.3.1, we have here resorted to the positive-null form. Since we restrict ourselves to linear elastic materials, the constitutive relationship used in the finite element discretisation is adequately described by

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\epsilon}\,,$$

where $\boldsymbol{\sigma}$, $\boldsymbol{C}$ and $\boldsymbol{\epsilon}$ are the stress, elasticity and strain tensors respectively. After Bruns, we now introduce the notion of the first density measure $\mu_{1_i}(x_i)$ for element i, which can be interpreted as 'scaling' the material properties between 0 or void, and 1 or solid; it is introduced into the problem formulation via the elasticity tensor $\boldsymbol{C}_i$, using

$$\bar{\boldsymbol{C}}_i(x_i) = \mu_{1_i}(x_i)\boldsymbol{C}_0\,.$$

Here, $\boldsymbol{C}_0$ is the elasticity tensor of the solid material and $\boldsymbol{C}_i(x_i)$ is the effective elasticity tensor. We assume that $\mu_{1_i}(x_i)$ depends on element i only, for reasons that will become clear shortly.

[1] It may (correctly) be argued that these differences neglect important advantages of the SINH method. However, this is irrelevant in any discussion of the lack of convexity of volumetric penalisation.

[2] Though it is important to note that multiple constraints pose no problem whatsoever in algorithms developed using dual principles.

The compliance $f_0(\boldsymbol{x})$ is obtained in terms of the first density measure as

$$f_0(\boldsymbol{x}) = \boldsymbol{q}^T \boldsymbol{r} = \boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} = \sum_{i=1}^{n} \mu_{1_i}(x_i) \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \,, \tag{5.2}$$

in which the $\boldsymbol{K}_i$ are elemental stiffness matrices defined by (2.5) and $\boldsymbol{q}_i$ is the vector of nodal displacements. The subscript i indicates elemental quantities and operators, and there are n finite elements in the mesh. For an elemental volume of ν_i, the effective elemental material volume can be represented as

$$\nu_i^e = \nu_i \mu_{2_i}(x_i) \,, \tag{5.3}$$

with $\mu_{2_i}(x_i)$ the second density measure. We then formulate the volume constraint

$$f_1(\boldsymbol{x}) = \bar{\nu} - \frac{\nu(\boldsymbol{x})}{\nu_0} = \bar{\nu} - \frac{1}{\nu_0} \sum_{i=1}^{n} \nu_i \mu_{2_i}(x_i) \geq 0 \,, \tag{5.4}$$

where $\nu(\boldsymbol{x})$ represents the material or final structural volume, ν_0 the total volume of the design domain Ω, and $0 < \bar{\nu} < 1$ a prescribed limit on the final volume fraction allowed.

5.3.1 The SIMP method

In the classical SIMP method, we have

$$\begin{aligned} \mu_{1_i}(x_i) &= x_i^p \,, \\ \mu_{2_i}(x_i) &= x_i \,, \end{aligned} \tag{5.5}$$

where $p \geq 1$ is the penalty parameter that, in the case of the inequality, drives the solution towards the bounds $\check{x}$ and 1, e.g. see [42]. Finally, since the SIMP method relies on penalisation of the *first* density measure, we will temporarily denote the SIMP method the SIMP$^{(1)}$ method, for 'solid isotropic material with penalisation of the *first* density measure'.

5.3.2 Volumetric penalisation

Bruns' SINH method

As an alternative to the SIMP$^{(1)}$ method, in which the first density measure $\mu_{1_i}(x_i)$ is penalised, Bruns [39] and others have recently proposed to rather penalise the second density measures $\mu_{2_i}(x_i)$. Using the hyperbolic sine function rather than a power law, this is expressed as

$$\begin{aligned} \mu_{1_i}(x_i) &= x_i \,, \\ \mu_{2_i}(x_i) &= 1 - \frac{\sinh(d(1-\rho))}{\sinh(d)} \,, \end{aligned} \tag{5.6}$$

with ρ a generalisation of the design variables (required when the design is filtered, rather than the sensitivities). The first density measure $\mu_{1_i}(x_i)$ is not penalised. $d \geq 1$ is the penalty parameter;

in the case of the inequality, intermediate-density material becomes volumetrically inefficient, e.g. see Bruns [39].

Advantages of Bruns' approach, which used the SINH method in combination with density filtering, are that the optimisation problem is consistently defined, the topology description is unambiguous, and the method leads to predominantly solid-void (black-and-white) designs. The consistency and unambiguity result from filtering the design rather than the sensitivities (which is pretty standard in SIMP$^{(1)}$). Apparently, the predominantly black-and-white designs are to be attributed to penalisation of the second density measure in the SINH method, and not to the problem being consistent or to the use of the hyperbolic sine function.

According to Bruns, a drawback of his implementation is that the designs are 'somewhat less distinct or more diffuse' than in the SIMP$^{(1)}$ method, since the design *itself* is defined via a filtered density design field. This shortcoming, resulting from density filtering, may largely be overcome by a hybrid formulation, being a combination of the SIMP$^{(1)}$ and SINH methods, in that both the first and second density measures are penalised, viz.

$$\begin{aligned} \mu_{1_i}(x_i) &= \frac{\sinh(p\beta)}{\sinh(p)}\,, \\ \mu_{2_i}(x_i) &= 1 - \frac{\sinh(d(1-\beta))}{\sinh(d)}\,, \end{aligned} \tag{5.7}$$

with $p \geq 1$ and $d \geq 1$. Bruns reports that the hybrid SINH method may be sensitive to the relative values of the first and second density measures [39]. Also, the upper bound on the volume is not satisfied, in particular during intermediate iterations, if the solution algorithm utilises convex approximations to the constraint. This is not, however, considered problematic, since the final designs are predominantly black and white, while the prescribed volume is normally an approximate goal only (but the implications for additional arbitrary nonconvex constraints are clear).

Finally, the hyperbolic sine function used in the foregoing has some advantages over the more traditional power law (e.g. the derivatives do not vanish as the design variables $x_i \to 0$, which may be advantageous in some applications).

SIMP-like volumetric penalisation

As argued in the foregoing, the hyperbolic sine function does not seem fundamental to the development of the SINH method; many a penalty method can conceptually be used in combination with volumetric penalisation (i.e. penalisation of the second density measure). Neither is filtering of the *densities* essential in volumetric penalisation methods. For the sake of simplicity we therefore rather use the traditional power law in this study, and we employ the well-established approach of filtering the design sensitivities. Although this filtering method is not without its problems, filtering of the design itself seems unattractive in that it is not customary in structural topology optimisation. In addition, filtering of the design *per se* results in 'diffuse' solutions[3].

[3]This is a dilemma of significant proportions. Some argue that regularisation of the problem is of significant importance. In our opinion, this is indeed the case if problems with continuous design variables are studied (e.g. the so-called variable-sheet problem). However, if the optimal design variables are required to be discrete, it is in our opinion also important to realise that the relaxed, continuous problem is merely a surrogate problem for the intractable discrete

Hence, instead of (5.6), we use

$$\begin{aligned}\mu_{1_i}(x_i) &= x_i\,,\\ \mu_{2_i}(x_i) &= x_i^d\,,\end{aligned} \tag{5.8}$$

with $0 < d \leq 1$. Accordingly, we will denote this method by SIMP$^{(2)}$, for 'solid isotropic material with penalisation of the *second* density measure. If we retain SIMP$^{(1)}$ penalisation of the first density measure (which is not a requirement), we replace (5.7) by

$$\begin{aligned}\mu_{1_i}(x_i) &= x_i^p\,,\\ \mu_{2_i}(x_i) &= x_i^d\,,\end{aligned} \tag{5.9}$$

with $p \geq 1$ and $0 < d \leq 1$. For obvious reasons, we will denote this hybrid method by SIMP$^{(1,2)}$. On purpose, we let μ_{1_i} depend on x_i only. This lowers the complexity of the resultant optimisation problem, and is possible when the sensitivities of the objective are filtered, rather than the design itself, as proposed by Bruns and Tortorelli [38], and Bruns [39].

Finally, note that volumetric penalisation results in separable, strictly concave constraint functions that exhibit strict *monotonicities* with respect to the design variables x_i.

5.4 Approximate subproblems

The approximate subproblems used herein are based on first-order Taylor series expansions of the objective and constraint functions. The familiar direct linear expansion valid for the k^{th} iteration of the SAO algorithm is

$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f}{\partial x_i}\right)^{\{k\}}, \tag{5.10}$$

where the quantities bearing the superscript k are constants evaluated at the optimal solution of the previous subproblem defined for iteration $k-1$. The direct linear expansion (5.10) is conventionally used to approximate the linear (unpenalised) volume constraint usually present in the minimum compliance problem. On the other hand, since the sensitivities of the minimum compliance objective function, given by

$$\frac{\partial f}{\partial x_i} = -\frac{\partial}{\partial x_i}\left(\mu_{1_i}(x_i)\right)\boldsymbol{q}_i^T\boldsymbol{K}_i\boldsymbol{q}_i\,, \tag{5.11}$$

are always negative, the objective function can be approximated using a linear expansion in terms of either reciprocal or exponential intervening variables with negative exponents. When volumetric penalisation is employed, the concave constraint can be approximated in terms of exponential intervening variables with positive exponents, though in this case the constraint need not be approximated at all, as it can be used directly.

programming problem. Hence, the quality of the final discrete solution is of some importance, while regularisation should mostly be of interest from the point of view that solving the subproblems should be problem free. We hope to elaborate on this elsewhere.

5.4.1 Reciprocal intervening variables

We write (5.10) in terms of the variables y_i, $i = 1, 2, \cdots, n$, whereafter we substitute the reciprocal intervening variables

$$y_i = \frac{1}{x_i},\ i = 1, 2, \cdots n.$$

In terms of the original variables x_i, the approximation is given as

$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(x_i - x_i^{\{k\}}\right) \left(\frac{x_i^{\{k\}}}{x_i}\right) \left(\frac{\partial f}{\partial x_i}\right)^{\{k\}}, \tag{5.12}$$

since the intervening variables y_i, $i = 1, 2, \cdots, n$ are functions of a single design variable x_i only. The convexity of (5.12) depends on the sign of the partial derivatives $(\partial f/\partial x_i)^{\{k\}}$. When these derivatives are negative (as they are for the minimum compliance topology optimisation problem), we obtain a *strictly* convex approximation.

5.4.2 Exponential intervening variables

If instead we substitute the exponential intervening variables

$$y_i = x_i^{r_i},\ i = 1, 2, \cdots n,$$

a so-called exponential approximation results [44]:

$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left[\left(\frac{x_i}{x_i^{\{k\}}}\right)^{r_i^{\{k\}}} - 1\right] \left(\frac{x_i^{\{k\}}}{r_i^{\{k\}}}\right) \left(\frac{\partial f}{\partial x_i}\right)^{\{k\}}. \tag{5.13}$$

The convexity of (5.13) depends on the values of the $r_i^{\{k\}}$ and the signs of $(\partial f/\partial x_i)^{\{k\}}$. If the $r_i^{\{k\}}$ are all negative, the requirements for convexity of (5.13) are similar to the requirements for (5.12). Finally: for $r_i^{\{k\}} = -1$, we recover a reciprocal approximation in term i, whereas for $r_i^{\{k\}} = 1$ we recover a direct linear approximation in term i.

It is in order to note that the standard OC method in minimum compliance topology optimisation is equivalent to the use of the exponential approximation for the objective function [43]. However, the exact effect of sensitivity filtering on the compliance objective function is not clear. Indeed, it is fair to say that development of the primal objective function in the presense of sensitivity filtering is considered an open research issue by many. Nevertheless, the use of sensitivity filtering with approximations that employ intervening variables is commonplace, and we will not concern ourselves with theoretical deficiencies of sensitivity filtering here. (An important example of an algorithm that uses reciprocal-like intervening variables is MMA, which is often used in combination with sensitivity filtering.)

Furthermore: while the filtered compliance objective function suffers from theoretical difficulties, the primal and dual *subproblems* at least are problem free (from the point of view that the dual can be developed, and that they are equivalent). And we reiterate that filtering of the densities

should possibly also be considered to suffer from theoretical difficulties: the optimisation problem is posed as a discrete problem, but density filtering *per se* prevents discrete variables between solid and void regions, albeit that the relaxed (continuous) optimisation problem is consistent. (Given the possibilities of post-processing methods for some problems, it may be too restrictive to formulate all topology optimisation problem as discrete in the first place, but that is again a question we will not concern ourselves with here.)

5.5 Analysis of the nonconvex problem

5.5.1 Purely nonconvex constraints

When addressing the minimum compliance problem we will use the penalised volumetric constraint (5.4) – with the second density measure defined by (5.8) – directly, which is equivalent to employing an exponential approximation (5.13) with positive exponents. The objective function will be approximated either as (5.12) or as (5.13) with negative exponents. In either case, the resulting mathematical programming problem represented by the approximate subproblem has the following explicit form[4]:

$$
\begin{aligned}
&\min_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i^{r_i} \\
&\text{subject to} \;\; f_j(\boldsymbol{x}) = c_{0j} + \sum_{i=1}^{n} c_{ij} x_i^{q_{ij}} \geq 0 \qquad j = 1, 2, \cdots, m, \\
&\qquad\qquad a_i > 0 \qquad\qquad i = 1, 2, \cdots, n, \\
&\qquad\qquad \alpha \leq r_i < 0 \qquad\qquad i = 1, 2, \cdots, n, \\
&\qquad\qquad c_{ij} < 0 \qquad\qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m, \\
&\qquad\qquad 0 < q_{ij} \leq 1 \qquad\qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m, \\
&\qquad\qquad 0 < \check{x}_i \leq x_i \leq \hat{x}_i \qquad\qquad i = 1, 2, \cdots, n.
\end{aligned}
\tag{5.14}
$$

The upper bound on all the primal variables is $\hat{x} = 1$, and the curvatures of the approximate objective function are limited by setting α at some small negative number ($\alpha = -4$, for instance). In (5.14) we have not restricted ourselves to the consideration of only a single constraint, but we assume that, if there are multiple constraints, all the constraints have an exponential form with $0 < q_{ij} \leq 1$. We further assume that at least one $0 < q_{ij} < 1$, otherwise the problem becomes convex and what follows is rendered uninteresting.

The objective is a strictly decreasing reciprocal function (i.e. strictly convex and monotonic), but in this case the constraints are also strictly decreasing reciprocal functions (i.e. monotonic and convex), which means that the feasible region is nonconvex, bearing in mind that we represent the constraints using the positive-null form in this chapter. However, maximising the associated Falk dual corresponds to minimising (5.14) in the primal form.

[4] In the remainder of this chapter we do not explicitly indicate that we now solve an approximate substitute subproblem, but this is clear from the context.

For (5.14), each separable term in the Lagrangian – see (2.42) – has the following general form:

$$\mathcal{L}_i = a_i x_i^{r_i} + \sum_{j=1}^{m} \lambda_j b_{ij} x_i^{q_{ij}}, \tag{5.15}$$

with

$$\begin{aligned} &b_{ij} = -c_{ij} > 0 && i = 1, 2, \cdots, n, \\ &\alpha \leq r_i < 0 && i = 1, 2, \cdots, n, \\ &0 < q_{ij} \leq 1 && i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m. \end{aligned}$$

From the arguments presented in Section 2.3.2, we only need to show that each $\mathcal{L}_i$ has a unique minimum with respect to x_i, for any $\boldsymbol{\lambda}$, to show that Falk's proofs apply to (5.14). This being the case, we know immediately that solving the dual corresponds to solving (5.14), and that both have unique optima. Hence, we consider (5.15) closely.

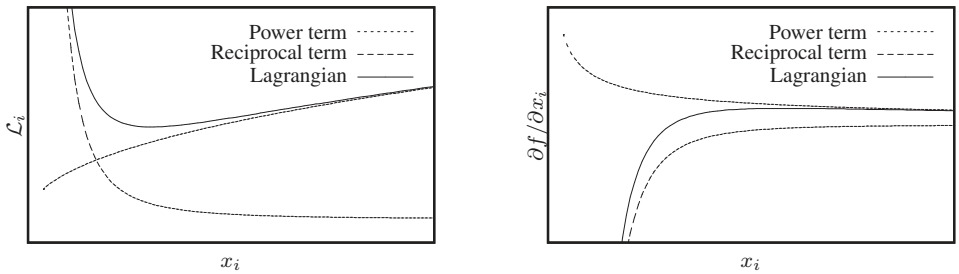
If all q_{ij} are different, there is generally no further simplification of (5.15) that sheds any more light on it. However, we note that

$$\lim_{x \to 0} \mathcal{L}_i = \lim_{x \to 0} a_i x_i^{r_i} = +\infty, \tag{5.16}$$

and

$$\lim_{x \to +\infty} \mathcal{L}_i = \lim_{x \to +\infty} x_i^{\overline{q}} = +\infty, \qquad 0 < \overline{q} \leq 1, \tag{5.17}$$

whenever at least one λ_j is greater than 0. Therefore, in this case there must exist at least one stationary point at finite x_i that represents a minimum. For the case where all λ_j are zero, $\mathcal{L}_i$ obviously reduces to a decreasing monotonic function, and its minimum over $\mathcal{C}$ will be at $x_i = \hat{x}_i$. Figure 5.1(a) illustrates the case where at least one $\lambda_j > 0$. The reciprocal term dominates for small x_i, and $\mathcal{L}_i$ is therefore convex where x_i is sufficiently small. For large x_i, on the other hand, the power terms dominate, and these are concave. $\mathcal{L}_i$, then, is a nonconvex function, but we wish to show that it has a unique stationary point so that, by the limit argument presented above, this stationary point must be a minimum. To this end, we start by observing that at any stationary point



(a) Terms in the Lagrangian (b) Gradients of the terms

Figure 5.1: The form of the one-dimensional functions in the Lagrangian, and their gradients, for problem (5.14).

it is necessary that

$$-\frac{\partial}{\partial x_i}\left(a_i x_i^{r_i}\right) = \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{m} \lambda_j b_{ij} x_i^{q_{ij}}\right),$$

$$\text{or} \qquad -a_i r_i x_i^{(r_i-1)} = \sum_{j=1}^{m} \lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}. \tag{5.18}$$

The term on the left of the equality in (5.18) is a monotonically decreasing function that is always positive for positive x_i. Figure 5.1(b) contains the negative of this curve. The term on the right of the equality is the sum of positive, monotonically decreasing functions, and is thus itself a monotonically decreasing function that is always positive for positive x_i (again, refer to Figure 5.1(b) for an example). However, since the q_{ij} are different in general, it is difficult to simplify this sum. Now consider the following observations for the case when the q_{ij} are strictly less than 1:

Observation 1: *The curve defined by any one term of the sum in (5.18), i.e.* $\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}$, *intersects the curve defined by* $-a_i r_i x_i^{(r_i-1)}$ *exactly once.*

To show this, it is only necessary to note that the equation

$$-a_i r_i x_i^{(r_i-1)} = \lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}$$

has the unique real solution

$$x_i^* = \left(\frac{-a_i r_i}{\lambda_j b_{ij} q_{ij}}\right)^{\frac{1}{q_{ij}-r_i}},$$

bearing in mind that r_i is negative.

Observation 2: $\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)} < -a_i r_i x_i^{(r_i-1)} \quad \forall \quad x_i < x_i^*$ *and*
$\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)} > -a_i r_i x_i^{(r_i-1)} \quad \forall \quad x_i > x_i^*$.

This is easily seen if one writes

$$x_i = x_i^* \epsilon\,.$$

Then

$$\frac{\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}}{-a_i r_i x_i^{(r_i-1)}} = \epsilon^{(q_{ij}-r_i)} \qquad \text{and} \qquad q_{ij} - r_i > 0\,.$$

Therefore,

$$x_i < x_i^* \quad \text{and} \quad \frac{\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}}{-a_i r_i x_i^{(r_i-1)}} < 1 \quad \text{when} \quad \epsilon < 1, \quad \text{and}$$

$$x_i > x_i^* \quad \text{and} \quad \frac{\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}}{-a_i r_i x_i^{(r_i-1)}} > 1 \quad \text{when} \quad \epsilon > 1.$$

Since both $\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}$ and $-a_i r_i x_i^{(r_i-1)}$ are positive numbers (for positive x_i), Observation 2 is verified.

Observation 3: *The curve defined by the gradient of any one term of the sum in (5.18), i.e.* $\frac{\partial}{\partial x_i}\left(\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-r_i)}\right)$*, intersects the curve defined by* $\frac{\partial}{\partial x_i}\left(-a_i r_i x_i^{(r_i-1)}\right)$ *exactly once.*

Once again we simply note that the equation

$$-a_i r_i \left(r_i - 1\right) x_i^{(r_i-2)} = \lambda_j b_{ij} q_{ij} \left(q_{ij} - 1\right) x_i^{(q_{ij}-2)}$$

has the unique real solution

$$x_i^{\diamond} = \left(\frac{-a_i r_i \left(r_i - 1\right)}{\lambda_j b_{ij} q_{ij} \left(q_{ij} - 1\right)}\right)^{\frac{1}{q_{ij}-r_i}},$$

bearing in mind that r_i and $(q_{ij} - 1)$ are both negative.

Observation 4: $x_i^* < x_i^{\diamond}$.

$$x_i^{\diamond} = x_i^* \left(\frac{r_i - 1}{q_{ij} - 1}\right)^{\frac{1}{q_{ij}-r_i}} \qquad \text{and} \qquad \frac{r_i - 1}{q_{ij} - 1} > 1.$$

Observation 5: $\frac{\partial}{\partial x_i}\left(\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}\right) > \frac{\partial}{\partial x_i}\left(-a_i r_i x_i^{(r_i-1)}\right) \quad \forall \quad x_i < x_i^{\diamond}$ *and*
$\frac{\partial}{\partial x_i}\left(\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}\right) < \frac{\partial}{\partial x_i}\left(-a_i r_i x_i^{(r_i-1)}\right) \quad \forall \quad x_i > x_i^{\diamond}$.

This last can again be shown by using the ϵ-argument given under Observation 2, replacing x_i^* with $x_i^{\diamond}$, and additionally taking note of the fact that $\frac{\partial}{\partial x_i}\left(\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)}\right)$ and $\frac{\partial}{\partial x_i}\left(-a_i r_i x_i^{(r_i-1)}\right)$ are both negative numbers for positive x_i.

Armed with the above observations we now proffer the following argument to indicate the uniqueness of the stationary point for $\mathcal{L}_i$. Since a stationary point occurs wherever (5.18) is satisfied, we concern ourselves with the curves that represent the gradients of the functions in $\mathcal{L}_i$, depicted in Figure 5.1(b). For the sake of simplicity, we denote the gradient of the reciprocal term $(-a_i r_i x_i^{(r_i-1)})$ as A_i, the gradient of each power term in the sum $(\lambda_j b_{ij} q_{ij} x_i^{(q_{ij}-1)})$ as g_{ij} and the gradient of the sum total as B_i. Each g_{ij} intersects A_i only once (Observation 1). Hence, there are at most m points at which A_i is intersected by a curve representing a term in the sum. We call the least of these x_i^* values $x_i^{*\dagger}$ and the term associated with it $g_{ij}^{\dagger}$. Then we deduce that all $g_{ij} < A_i$ for $x_i < x_i^{*\dagger}$ (by Observation 2). Also, at each one of the intersection points $x_i^{*k}, k \in m$ (k is here used as an index, not an exponent), the gradient of the associated curve g_{ij}^k is shallower than the gradient of A_i and remains so over the region $x_i < x_i^{*k}$ (Observations 3, 4 and 5). This, therefore, is true of all g_{ij} for $x_i < x_i^{*\dagger}$. So, the quantity

$$\delta_i = A_i - B_i = A_i - \sum_{j=1}^{m} g_{ij} = A_i - \sum_{j=1}^{m} \left[A_i - \left(A_i - g_{ij}\right)\right] \tag{5.19}$$

increases as x_i moves from $x_i^{*\dagger}$ towards zero, because all the terms $(A_i - g_{ij})$ each increase individually. If $q_{ij} = 1$, then Observations 3, 4 and 5 are void. However, in this case g_{ij} is simply a positive constant, and it immediately follows that its representative curve intersects A_i only once at a x_i^*, that

$$g_{ij} < A_i \quad \forall \quad x_i < x_i^* \qquad \text{and} \qquad g_{ij} > A_i \quad \forall \quad x_i > x_i^*, \tag{5.20}$$

and that $(A_i - g_{ij})$ must therefore still increase as x_i moves from $x_i^{*\dagger}$ towards zero. Now, at $x_i^{*\dagger}$, $B_i > g_{ij}^{\dagger}$ because B_i is the sum of $g_{ij}^{\dagger}$ and other g_{ij} that also have positive values. Given the above, if B_i intersects A_i at some point x_i^{min} (necessarily less than $x_i^{*\dagger}$), then B_i must remain less than A_i for all $x_i < x_i^{min}$, i.e. there can at most be one stationary point in the region $0 < x_i \leq x_i^{*\dagger}$. Moreover, by Observation 2, there can be no stationary points in the region $x_i^{*\dagger} \leq x_i \leq \infty$, because $B_i > A_i$ there. Ergo, (5.15) has at most a unique stationary point and, by the limit argument, this stationary point must exist and must represent a minimum.

We have shown that Attribute 1 is always met by (5.14) and the Falk dual is thus properly defined. Actually locating the minimum of $\mathcal{L}_i$ would in general require a numerical line search if there is more than one constraint present and the q_{ij} are different. If, however, there is only one constraint $(q_{ij} \to q_i)$, as is the case in the topology problem considered, the minima of $\mathcal{L}_i,\ i = 1, 2, \cdots, n$, can be found analytically and these minima are given by the statement

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i(\boldsymbol{\lambda}) & \text{if} \quad \check{x} < \beta_i(\boldsymbol{\lambda}) < \hat{x}, \\ \check{x} & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \leq \check{x}, \\ \hat{x} & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \geq \hat{x}, \end{cases} \tag{5.21}$$

with

$$\beta_i = \left(\frac{-a_i r_i}{\lambda b_i q_i} \right)^{\frac{1}{q_i - r_i}}. \tag{5.22}$$

For the problem described in Section 5.3 with a single volume constraint given directly by (5.4) and incorporating the SIMP$^{(2)}$ volumetric penalisation given by (5.8), all $q_i = d$. The objective function is approximated by (5.13) (since (5.13) includes (5.12) as a special case), so in iteration k the β_i become

$$\beta_i = \left(\frac{-\left(x_i^{\{k\}}\right)^{\left(1 - r_i^{\{k\}}\right)} \left(\dfrac{\partial f_0}{\partial x_i} \right)^{\{k\}}}{\lambda \left(\dfrac{1}{\nu_0} \right) d} \right)^{\frac{1}{d - r_i^{\{k\}}}}. \tag{5.23}$$

Equation (5.23) is valid whether or not SIMP$^{(1)}$ penalisation is carried out on the first density measure in the objective (5.2).

5.5.2 The addition of convex monotonic constraints

In (5.19), the quantity δ_i corresponds to the negative of the gradient of the Lagrangian term $\mathcal{L}_i$. We have noted that δ_i increases as x_i moves from $x_i^{*\dagger}$ towards zero, which is to say that the gradient of the Lagrangian term becomes increasingly negative as $x_i \to 0$ with $x < x_i^{*\dagger}$. We have also indicated that the minimum of $\mathcal{L}_i$, which we denote x_i^{min}, lies in this region. From this we infer

that $\mathcal{L}_i$ is convex over the region $x < x_i^{min}$. Now, assume that an additional set of constraints of the form

$$f_l^{add}(\boldsymbol{x}) = c_{0l}^{add} + \sum_{i=1}^{n} c_{il}^{add} x_i^{t_{il}} \geq 0 \qquad l = (m+1), (m+2), \cdots, s$$

is added to problem (5.14), where we again require that $c_{il}^{add} < 0$ for all i and l. Here we let $t_{il} > 1 \; \forall \; i, l$, so the additional constraints are separable, monotonically increasing power functions. The individual Lagrangian terms now acquire additional convex terms

$$\begin{aligned}\mathcal{L}_i^{new} =& a_i x_i^{r_i} + \sum_{j=1}^{m} \lambda_j b_{ij} x_i^{q_{ij}} + \sum_{l=m+1}^{s} \lambda_l b_{il}^{add} x_i^{t_{il}} \\ =& \mathcal{L}_i^{nc} + \sum_{l=m+1}^{s} \lambda_l b_{il}^{add} x_i^{t_{il}},\end{aligned}$$

in which $b_{il}^{add} = -c_{il}^{add} > 0$ and $\mathcal{L}_i^{nc}$ denotes the nonconvex Lagrangian of (5.15). Regarding the existence of a unique stationary point of $\mathcal{L}_i^{new}$, the addition of the convex terms does nothing to change the limit argument proffered in Section 5.5.1, so we know that a minimum of $\mathcal{L}_i^{new}$ must exist. Also, since the gradients of $\mathcal{L}_i^{nc}$ and the additional terms are all positive in the region $x_i > x_i^{min}$, the minimum must be in the region $x_i \leq x_i^{min}$. However, it is evident that $\mathcal{L}_i^{nc}$ and the additional terms are convex over this latter region, so the minimum of $\mathcal{L}_i^{new}$ must also be unique. This means that it is possible to use Falk's dual formulation to solve the following broader version of (5.14):

$$\begin{aligned}&\min_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i^{r_i} \\ &\text{subject to } f_j(\boldsymbol{x}) = c_{0j} + \sum_{i=1}^{n} c_{ij} x_i^{q_{ij}} \geq 0 \qquad j = 1, 2, \cdots, m, \\ &\qquad a_i > 0 \qquad i = 1, 2, \cdots, n, \\ &\qquad \alpha \leq r_i < 0 \qquad i = 1, 2, \cdots, n, \\ &\qquad c_{ij} < 0 \qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m, \\ &\qquad 0 > q_{ij} \qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m, \\ &\qquad 0 < \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n,\end{aligned} \tag{5.24}$$

in which the q_{ij} are only required to be non-negative. Once again, in solving the dual for the general case of arbitrary constraints, the determination of the primal-dual relationship (2.41) will require n numerical line searches at any given $\boldsymbol{\lambda}$. In terms of the minimum compliance problem with SIMP$^{(2)}$ volumetric penalisation considered in this chapter, the above indicates that the problem can be solved directly using the dual formulation when the constraints are all of decreasing exponential form with positive exponents.

5.6 Computational implementations of volumetric penalisation

5.6.1 On constraint violation

Let us first reflect on the observation that volumetric penalisation implies that the upper bound on the volume of the material in the design space $\bar{\nu}$ is not adhered to, in particular during intermediate steps of the optimisation process. This is not generally considered to be problematic, since any design that satisfies the penalised volume constraint will have a volume that is less than or equal to $\bar{\nu}$, due to the concavity of the penalised constraint. This discrepancy between the specified volume limit and the volume of the design found by the optimiser is a natural result of employing a penalised density measure in the volume constraint, while the physical volume of the design must still be understood to be a linear function of the unpenalised density, viz.,

$$V_l = \sum_{i=1}^{n} \nu_i x_i \,. \tag{5.25}$$

The volume of the design (as per (5.25)) and the volume calculated by using the second density measure

$$V_p = \sum_{i=1}^{n} \nu_i \mu_{2_i}(x_i) \,, \tag{5.26}$$

with $\mu_{2_i}(x_i)$ given as in (5.8), are identical only at $[0, 1]$ solutions. For numerical reasons, a hard lower limit on the densities $\check{x}$ is always necessitated, so true $[0, 1]$ solutions are not achievable but, in theory at least, V_p can be brought arbitrarily close to V_l at black-and-white designs by letting $\check{x} \to 0$.

5.6.2 On concavity

We now investigate the numerical solution of the (relaxed) topology problem (5.1), using minimum compliance as the objective (5.2), which may incorporate SIMP$^{(1)}$ penalisation, and a single penalised volumetric constraint (5.4) where SIMP$^{(2)}$ volumetric penalisation (5.9) is used. We apply two different optimisation algorithms. Firstly, we apply the standard MMA algorithm, which constructs strictly convex approximations to the nonconvex constraint. Secondly, we represent the constraint exactly and solve the dual by means of (5.23), and we refer to the resulting algorithm as the 'nonconvex algorithm'. Both of these algorithms solve the approximate subproblems in the space of the dual variables.

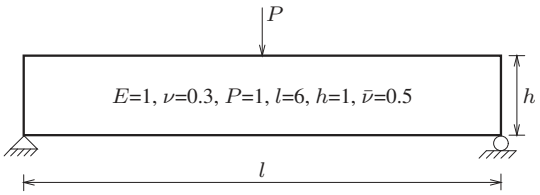


Figure 5.2: The MBB beam (unit thickness; plane stress).

The particular problem considered is the well-known MBB beam depicted in Figure 5.2, and the problem settings are as follows: we use a 150×50 mesh and a linear mesh independence filter with radius $r_{\text{mesh}} = 4.0$ (see Sigmund [14, 15]). The minimum allowable density $\check{x}$ is a function of the SIMP$^{(1)}$ penalty p and of machine precision. We use the following combinations of p and $\check{x}$: $(p = 1, \check{x} = 10^{-10})$ and $(p = 3, \check{x} = 10^{-3})$. We use standard four-node displacement-based isoparametric finite elements with bilinear interpolation (often known as Q4 elements). To ensure feasible starting designs, we initiate the optimisation process at the point in the design space given by

$$x_i = \bar{\nu}^{\frac{1}{d}} \qquad \forall \qquad i = 1, 2, \cdots, n. \tag{5.27}$$

Finally, we introduce $\phi_{B\&W}$, which represents the elemental 'black-and-white fraction', viz. the sum of the combined number of elements on the lower and upper bounds $n_{[0]} + n_{[1]}$, divided by the total number of elements n, viz.,

$$\phi_{B\&W} = \frac{n_{[0]} + n_{[1]}}{n}\,. \tag{5.28}$$

Results are presented for each of the algorithms using $\bar{\nu} = 0.5$ and the two penalty pairs $(p = 1, d = 0.35)$, which is an instance of SIMP$^{(2)}$, and $(p = 3, d = 0.35)$, an instance of SIMP$^{(1,2)}$.

The topology after 100 iterations is shown together with the associated objective function value (f_0^{100}) and black-and-white fraction ($\phi_{B\&W}$), whenever the latter are meaningful. In these figures, the plotted grey-scale values of the grid elements correspond to their direct, unpenalised design densities x_i. Plots of the initial convergence histories of the objective function and the (feasible) constraint values are also proffered.

Strictly convex constraint approximation

MMA has become the algorithm of choice for solving the topology problem, particularly when multiple constraints are applied. The MMA approximations are strictly convex, and the 'moving asymptotes' function as built-in move limits. Consequently, it is customary to run MMA without applying additional (external) move limits. We have, however, found it necessary to introduce such a move limit. Also, it is necessary to set the penalties $c_i \geq 10000$ to generate feasible solutions (see the MMA literature).

The results generated by the MMA algorithm for the minimum compliance problem with a single concave constraint are presented in Figure 5.3. For both sets of penalties, we present results for two values of the applied external move limit (δ_∞), namely $\delta_\infty = 1$ (which, given the bounds on x_i, amounts to applying no external move limit whatsoever) and $\delta_\infty = 0.2$.

For $p = 1$ and $d = 0.35$, MMA oscillates severely (Figure 5.3(c)), though it is evident that reducing the move limit damps the amplitude of the oscillation somewhat. Hence, the topology image presented has been chosen to correspond to the analysis in which $\delta_\infty = 0.2$. Given the scale of the oscillations, the presented topology at iteration 100 is rendered fairly meaningless. It is given for the purposes of maintaining consistency with the results presented in the remainder of the chapter. Equally, there is little use in stating the optimal objective function value at 100 iterations and its associated black-and-white fraction. In instances such as this, we instead present the minimum objective value found during the whole analysis (f_0^*), and the corresponding black-and-white fraction ($\phi_{B\&W}^*$).

Lastly, for $p = 3$ and $d = 0.35$, large-scale oscillations again appear in the analysis (Figure 5.3(d)). Curiously, the amplitude of the oscillations seems unaffected by the reduction of the move limit from $\delta_\infty = 1$ to $\delta_\infty = 0.2$ in this case. Here again, we state f_0^* and $\phi_{B\&W}^*$.

The convergence behaviour of MMA on the minimum compliance problem using SIMP$^{(1)}$ without volumetric penalisation is known to be very good. The foregoing results show that the presence of a concave constraint complicates the optimisation problem, to the extent that the performance of algorithms that rely on strictly convex approximations may be markedly degraded.

Nonconvex algorithm: exact representation of the constraint

The results generated using the nonconvex algorithm with the abovementioned penalty parameters are presented in Figure 5.4. No plots are given depicting the constraint value, since said measure is always of the order of 10^{-11}. It is determined mainly by the tolerance imposed on the dual maximisation scheme (i.e. for all intents and purposes, the constraint is always active, $\bar{\nu}$ being satisfied exactly).

We here present results only for a reciprocal approximation to the objective function (5.12), and we have used a move limit of $\delta_\infty = 0.4$. Since the constraint is represented exactly, the move limit represents the only control over the global search characteristics of the algorithm, so one would expect some sensitivity to δ_∞.

For $p = 3$ and $d = 0.35$, two oscillations occur during the first six iterations, though they are not visible on the graph in Figure 5.4(c). Reducing the move limit eliminates these oscillations, though they are not important in terms of convergence anyway. It is evident from the results that the use of a nonconvex approximation has produced a stable algorithm.

It is also possible to find results with improved black-and-white fractions by using an exponential approximation (5.13) with $r = -0.5$. However, this leads to an increased sensitivity to δ_∞ for large p ($p = 3$, for example). We have, however, used the exponential approximation in combination with a continuation strategy.

5.6.3 Preliminary comments on continuation methods

In the foregoing, we have used fixed values for the penalties p and d. This is certainly not recommendable in general; it is preferable to increase p (and decrease d) iteratively via some continuation method [65]. This stabilises the global search by controlling the rate of convergence and may reduce the likelihood of convergence to a local minimum (although convergence to the global optimum can never be demonstrated, since problem (5.1) is intractable). However, it is not our intention to exhaustively consider optimal continuation methods. Rather, we are interested in the *fundamental form of the subproblems that arise in the search for predominantly black-and-white designs.* Thus, we only present results for a single continuation strategy.

We keep $p = 1$ and $d = 1$ for the initial 15 iterations. Then p is slowly increased (by multiplication by $\alpha_1 = 1.02$) and d is decreased (by devision by $\alpha_2 = 1.01$) per iteration. An upper limit on p of $p = 3$ is set, as well as a lower limit on d of $d = 0.35$. The results are presented in Figure 5.5 for three finite element mesh discretisations. We use $r = -0.5$, $\delta_\infty = 0.4$ and $\breve{x} = 10^{-3}$. In

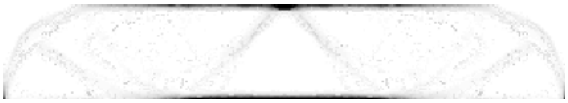
each case we allow the program to run to termination. The topologies are therefore well nigh zero-one solutions. In addition, the results were obtained without any significant oscillatory behaviour whatsoever; again, compare with Figure 5.3(c) for a typical convergence history obtained with MMA. The topologies, objective function values and black-and-white fractions are reported at termination; the superscripts represent the number of iterations that were required; they range between 120 and 180.

5.7 Conclusions and recommendations

We have studied the minimum compliance topology optimisation problem with SIMP-like volumetric penalisation, in which minimum compliance is sought subject to a single concave constraint on volume. We have shown numerically that the presence of the concave constraint may increase the difficulty of the problem dramatically if one employs a method based on strictly convex approximation. This is evidenced by the results obtained by the standard MMA algorithm, which exhibits large-scale oscillatory behaviour unless (and sometimes even though) an additional external move limit is applied.

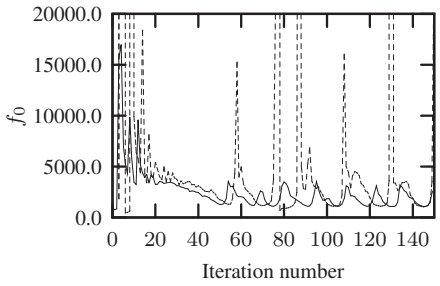
Regardless of the problems posed by concavity to (dual) algorithms based on convex primal approximations, we have shown that it is sometimes possible to solve nonconvex problems directly using a dual method. Accordingly, we have developed a nonconvex dual method that accommodates the concave constraint function involved in volumetric penalisation directly, without resorting to convex approximation. This is possible since strict convexity of the approximate subproblems is sufficient, but not necessary, to ensure that the solutions of the primal and dual problems are identical. We present numerical results that show that the developed nonconvex algorithm is indeed practicable, as the solutions obtained thereby are of a high quality for the considered problem.

Finally, our endeavours herein were merely aimed at drawing some attention to the idea that nonconvex forms may be amenable to solution via the Falk dual. We are not necessarily advocating the use of the SINH method (although use of volumetric penalisation and/or the SINH method may indeed constitute fruitful optimisation strategies). The ability of volumetric penalty methods to assist in generating predominantly solid-void discrete solutions in particular is considered to be of much importance.

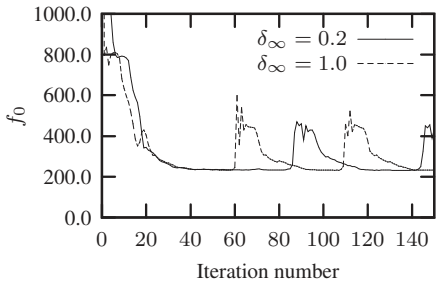


(a) $p = 1$, $\delta_\infty = 0.2$: $f_0^* = 752.78$, $\phi_{B\&W}^* = 0.003$

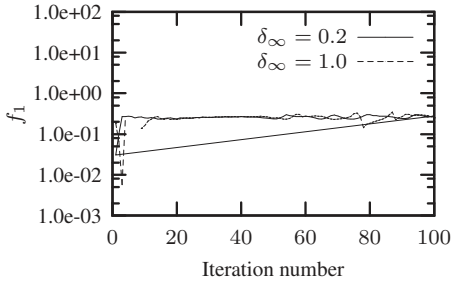
(b) $p = 3$, $\delta_\infty = 0.2$: $f_0^* = 231.65$, $\phi_{B\&W}^* = 0.806$



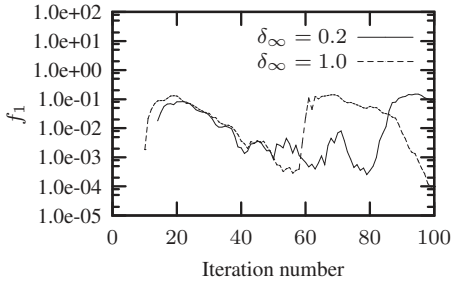
(c) Objective function history, $p = 1$



(d) Objective function history, $p = 3$



(e) Penalised constraint history, $p = 1$

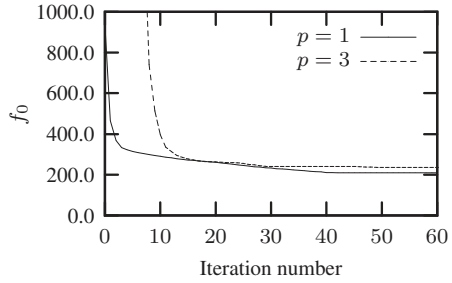


(f) Penalised constraint history, $p = 3$

Figure 5.3: The MBB beam. Optimal topologies and convergence histories obtained with MMA using SIMP$^{(1)}$ material penalisation ($p = 1$) and ($p = 3$), as well as SIMP$^{(2)}$ volumetric penalisation ($d = 0.35$), and two different move limit strategies δ_∞.

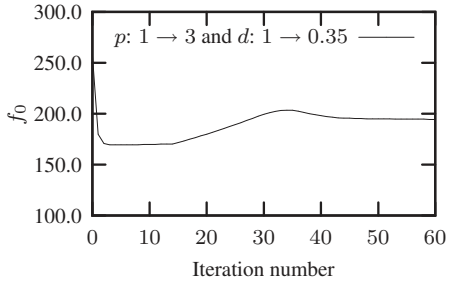
(a) $p = 1$, $f_0^{100} = 208.56$, $\phi_{B\&W}^{100} = 0.598$

(b) $p = 3$, $f_0^{100} = 234.55$, $\phi_{B\&W}^{100} = 0.797$



(c) Objective function histories

Figure 5.4: The MBB beam. Optimal topologies and convergence histories obtained with the nonconvex algorithm using SIMP$^{(1)}$ material penalisation ($p = 1$) and ($p = 3$), as well as SIMP$^{(2)}$ volumetric penalisation ($d = 0.35$). The move limit is set to $\delta_\infty = 0.4$.



(a) Objective function history for the half-beam mesh discretisation of 150×50

(b) Mesh discretisation for the half-beam: 75×25, $f_0^{128} = 187.51$, $\phi_{B\&W}^{128} = 0.9995$

(c) Mesh discretisation for the half-beam: 150×50, $f_0^{177} = 187.72$, $\phi_{B\&W}^{177} = 0.9999$

(d) Half-beam mesh discretisation: 225×75, $f_0^{162} = 188.24$, $\phi_{B\&W}^{162} = 0.9999$

Figure 5.5: The MBB beam. Optimal topologies and convergence histories obtained with the nonconvex algorithm using $r = -0.5$ in the approximation of the objective function and a continuation strategy on the penalty parameters p and d. Optimal topologies are given for three mesh discretisations. $\delta_\infty = 0.4$.

Chapter 6

Nonconvex forms in weight minimisation

The current chapter is a reproduction of a paper titled "Nonconvex dual forms based on exponential intervening variables, with application to weight minimisation" [36]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa.

6.1 Abstract

We study the weight minimisation problem in a dual setting. We propose new dual formulations for nonlinear multipoint approximations with diagonal approximate Hessian matrices, which derive from separable series expansions in terms of exponential intervening variables. These generally nonconvex approximations are formulated in terms of intervening variables with negative exponents, and are therefore applicable to the solution of the weight minimisation problem in a sequential approximate optimisation framework.

Problems in structural optimisation are traditionally solved using sequential approximate optimisation algorithms, like the method of moving asymptotes, which require the approximate subproblems to be strictly convex. Hence, during solution, the nonconvex problems are approximated using convex functions, and this process may in general be inefficient. We argue, based on Falk's definition of the dual, that it is possible to base the dual formulation on nonconvex approximations. To this end we reintroduce a nonconvex approach to the weight minimisation problem originally due to Fleury, and we explore certain convex and nonconvex forms for subproblems derived from the exponential approximations by the application of various methods of mixed variables. We show in each case that the dual is well defined for the form concerned, which may consequently be of use to future code developers.

6.2 Introduction

In recent years, sequential approximate optimisation (SAO) has firmly been established as the optimisation methodology of choice for simulation-based optimisation problems. A notable example

of such an algorithm is the well-known method of moving asymptotes (MMA) [3, 32], which is almost exclusively used in topology optimisation when multiple constraints are present.

As a consequence of the expense associated with the evaluation and storage of second-order information, most SAO methods aimed at simulation-based optimisation problems use only first-order sensitivity information. Frequently, these methods then exploit the advantages of so-called intervening variables, which can introduce some application-specific nonlinearities into the approximation functions used. In structural optimisation, for example, reciprocal intervening variables are very popular; among others, they have been included in the well-known CONLIN algorithm of Fleury and Braibant [4], whereas MMA uses reciprocal-like approximations with adjustable asymptotes, which make the form of the approximations variable. While exponential intervening variables are not quite as popular, they can potentially yield approximations of increased accuracy, an example being the first-order exponential approximation proposed by Fadel *et al.* [44].

If second-order information is included in an SAO algorithm, it is normally restricted to the diagonal terms of the Hessian or higher-order matrices, so that the approximations obtained are separable functions. Examples include the reputedly highly accurate TANA-2 and TANA-3 approximations proposed by Grandhi and his co-workers [66, 67, 68].

Separability of the approximations is often considered important, since solution of the resulting separable approximate subproblems may sometimes be easily effected using highly efficient dual formulations. These dual methods are particularly efficient when the number of constraints is (far) less than the number of design variables and when the primal-dual relationships can be determined analytically. Both the CONLIN and MMA algorithms employ a dual approach in solving their subproblems.

The most popular of the dual methods used in conjunction with SAO for continuous simulation-based optimisation problems is the dual as defined by Falk [2]. With this definition, the upper and lower bound constraints on the design variables do not have to be included explicitly as constraints in the definition of the Lagrangian. Falk proved that strict convexity of the approximate objective function, together with concavity of the approximate constraint functions (Falk defined the optimisation problem in the positive-null sense), are sufficient conditions to guarantee that the resulting dual function is concave and that its maximum corresponds to the minimum of the primal approximate subproblem. Convexity is of course also a sufficient requirement to guarantee the existence of a unique KKT point for the primal approximate subproblem. (Naturally, we assume herein that the primal problem is feasible.)

Most, if not all, general-purpose SAO codes exploit convexity as a rule, which is to say that convex functions (in the above sense) are used to construct the approximate subproblems, even if the problem itself is locally nonconvex. This approach is judicious, of course, if no problem-specific information is known a priori. Be that as it may, we wish to point out that, for certain popular structural optimisation problems, it can be advantageous to use the more naturally arising nonconvex approximations in the construction of the approximate subproblems. This is true, for example, of the minimum compliance topology optimisation problem if volumetric penalisation is used, and of the weight minimisation problem. The minimum compliance problem was the subject of Chapter 5, and in the current chapter we address the weight minimisation problem.

The solution of the weight minimisation problem via a dual method with the possible utilisation of certain first-order nonconvex approximations was presented previously by Fleury [28]. The

justification given for allowing these nonconvex approximations was that the resulting subproblem is transformable into a convex subproblem. The argument is valid in this case, but does not easily translate into a general rule, since its validity is dependent on the types of transformations that are allowed. This question was not formally explored in Reference [28].

We argue that the type of nonconvex subproblem arrived at by Fleury is catered for directly in the proof that Falk presented for convex problems. This is to say that Falk's proof holds without modification in this case, even though the subproblem is nonconvex. This obviates any discussion of convex transformability for the problem. We have discussed under which conditions nonconvex problems can be solved directly using the Falk dual in Section 2.3.2, and we here use Fleury's original nonconvex approach to the weight minimisation problem as a demonstrative example.

Fleury's approach utilised approximations based on first-order Taylor series expansions, both in terms of direct (design) variables and in terms of reciprocal intermediate variables. Both of these approximations are special cases of the separable expansion in terms of *exponential* intermediate variables [61] that we consider in this chapter. Previously, Groenwold *et al.* have presented an incomplete series expansion (ISE) as a basis for function approximation [61, 69]. The exponential function considered herein is one such expansion; it is expressed in terms of the 'main' or diagonal terms of second, third and even higher orders, but excludes 'interaction' or off-diagonal terms. That is, the function excludes all terms resulting from mixed partial derivatives.

Following on from our treatment of Fleury's approach, we investigate how approximations that derive from the general (higher-order) separable exponential expansion with negative exponents can be used in a dual SAO framework, as this may be pertinent to the weight minimisation problem and of interest to code developers. The exponential expansion includes nonconvex forms, and we discuss when such forms can be used in conjunction with the Falk dual. Two frequently encountered problems in structural optimisation, namely the weight minimisation problem with sizing design variables and the minimum compliance topology optimisation problem, represent degenerate cases of the formulations we present.

The chapter proceeds as follows: In Section 6.3, Fleury's (first-order nonconvex) treatment of the classical weight minimisation problem is described. We use the tenets born of Section 2.3.2 to demonstrate that the dual is properly defined for this problem, even though it is nonconvex. In Section 6.4 we introduce the separable expansion in terms of exponential intervening variables, and we explore whether a derivative form with negative exponents can be used to approximate functions in a dual SAO setting. Having examined the structure of the approximation, in Section 6.5 we go on to suggest three general methods of mixed variables based on this function that additionally incorporate other functions, which also derive from the exponential expansion. One of these methods produces strictly convex subproblems, two retain the higher-order terms. Section 6.6 describes the construction of the dual approximate subproblem once the primal approximate subproblem has been defined in terms of these approximations. Two first-order examples are also presented. In Section 6.7 we present a telling numerical example for a simple implementation. Finally, Section 6.8 reiterates the main points made in the chapter and presents our conclusions.

6.3 The weight minimisation problem

Fleury discussed the classical structural weight minimisation problem at length in Reference [28], in which a dual method for solving this problem was also introduced. The general form of the SAO subproblems given in Reference [28] is

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f_0(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i \\ \text{subject to} \; & f_j(\boldsymbol{x}) = c_{0j} + \sum_{i=1}^{n} \frac{c_{ij}}{x_i} \leq 0 && j = 1, 2, \cdots, m, \\ & a_i > 0 && i = 1, 2, \cdots, n, \\ & 0 < \check{x}_i \leq x_i \leq \hat{x}_i, && i = 1, 2, \cdots, n, \end{aligned} \tag{6.1}$$

which is also an exact representation of the problem for a statically determinate structure subject to static stress and displacement constraints only. The subproblem is formulated as a first-order Taylor series expansion about a given point in the domain using separable approximations. The Taylor approximation of the objective function is given in terms of direct variables, whereas the constraints are represented in terms of reciprocal intervening variables.

Problem (6.1) is convex only if all $c_{ij} \geq 0$. However, the signs of c_{ij} reflect the signs of the constraint gradients at a given point, and these can be either positive or negative. Hence problem (6.1) must in general be considered to be nonconvex. Since the adoption of general purpose algorithms like CONLIN and MMA, it has become standard practice to solve the weight minimisation problem (and other structural optimisation problems) using dual methods based on convex approximations. We wish to show that this convexification is not a necessary aspect of solution via the dual method, and it was not considered necessary in Fleury's original treatment of (6.1). Moreover, we argue that it is not even necessary if Falk's proof for convex problems is espoused, because, given the discussion in Section 2.3.2, the proof holds for certain nonconvex cases as well.

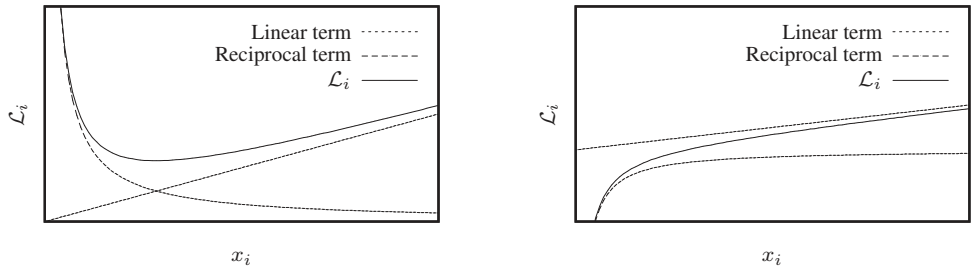
Fleury pointed out that, when expressed in terms of the reciprocals of the design variables, (6.1) becomes strictly convex regardless of the signs of c_{ij}. Then, for this transformed problem, Falk's proof for convex problems obviously applies, in which case the Falk dual can be used to solve the problem and, moreover, it possesses a unique KKT point. Expressing (6.1) in terms of the reciprocals of the design variables really implies a coordinate transformation of the form $x_i \rightarrow 1/x_i'$. Whether or not the properties of the transformed problem can be cited as being directly indicative of the properties of the untransformed problem depends, in general, on the nature of the transformation employed. In Chapter 7 the issue of convex transformability is examined more closely. For the purposes of the current chapter, it is simply noted that Falk's proof applies directly to the untransformed problem (refer to Section 2.3.2), so that evocation of a transformation is unnecessary, rendering questions regarding its validity irrelevant.

6.3.1 A discussion of Fleury's subproblem

For problem (6.1), each separable term in the Lagrangian has the form

$$\begin{aligned}\mathcal{L}_i &= a_i x_i + \left(\frac{1}{x_i}\right)\left(\sum_{j=1}^{m} \lambda_j c_{ij}\right) \\ &= a_i x_i + B_i \left(\frac{1}{x_i}\right), \end{aligned} \tag{6.2}$$

where a_i is always positive and non-zero and $B_i = \sum_{j=1}^{m} \lambda_j c_{ij}$ can be either positive, negative or zero, depending on the constants c_{ij} and the values of the Lagrange multipliers. In accordance with the argument presented in Section 2.3.2, in order that the Falk dual can be used it is necessary that each $\mathcal{L}_i$ possesses a unique minimum in $\mathcal{C}$ with respect to x_i for every $\boldsymbol{\lambda}$. Figure 6.1 shows the general forms of $\mathcal{L}_i$ for positive and negative B_i respectively.



(a) Positive B_i (b) Negative B_i

Figure 6.1: The form of the one-dimensional separable terms in the Lagrangean for problem (6.1).

If B_i is positive, then

$$\lim_{x \to 0} \mathcal{L}_i = +\infty \quad \text{and} \quad \lim_{x \to +\infty} \mathcal{L}_i = +\infty, \tag{6.3}$$

the linear term dominating for large values of x_i and the reciprocal term dominating for small values of x_i. The minimum is unique and sits either at the stationary point or at one of the bound values on x_i. To see this, recognise that the stationary condition

$$\frac{\partial \mathcal{L}_i}{\partial x_i} = 0 \tag{6.4}$$

yields

$$x_i(\boldsymbol{\lambda}) = \left(\frac{B_i}{a_i}\right)^{1/2}, \tag{6.5}$$

which has only one positive real solution when B_i is positive. Indeed, for B_i positive, the $\mathcal{L}_i$ are convex, and this observation would simplify the analysis for this specific problem. We resist relying on convexity here, since we will unfortunately not be able to do so in the remainder of this study, when neither monotonicity nor convexity will always hold. Haftka and Gürdal [27] note that

(6.5) is only valid if B_i is positive, since it has no real-valued solution when B_i is negative. This does not mean, however, that the dual cannot be used. When B_i is negative,

$$\lim_{x\to 0} \mathcal{L}_i = -\infty \quad \text{and} \quad \lim_{x\to +\infty} \mathcal{L}_i = +\infty. \tag{6.6}$$

The lack of a real-valued solution to (6.5) indicates that $\mathcal{L}_i$ must be monotonic. It is, in fact, monotonically increasing, as exemplified in Figure 6.1(b). When $B_i = 0$, (6.2) indicates that $\mathcal{L}_i$ will be an increasing linear function with gradient a_i. In both instances, $\mathcal{L}_i$ has a finite unique minimum in $\mathcal{C}$ at $x_i = \check{x}_i$. The primal-dual relationship still exists uniquely and the dual is still defined. Therefore, the following relationship between the primal and dual variables given by Fleury in his 'generalised optimality criteria approach' to the weight minimisation problem [28] is always valid, and derives rigorously from the application of the Falk dual methodology to the (in general) nonconvex problem (6.1):

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i^{1/2}(\boldsymbol{\lambda}) & \text{if} \quad \check{x}_i^2 < \beta_i(\boldsymbol{\lambda}) < \hat{x}_i^2, \\ \check{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \leq \check{x}_i^2, \\ \hat{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \geq \hat{x}_i^2, \end{cases} \tag{6.7}$$

where

$$\beta_i(\boldsymbol{\lambda}) = \left(\frac{B_i}{a_i}\right). \tag{6.8}$$

We have shown here that (6.1) may be solved directly using the Falk dual, even though it is a nonconvex problem. The proof of this last does not rely on the existence of a transformation that makes (6.1) convex, but is instead contained within Falk's original proofs for convex problems, which apply to some more general problems.

6.4 Higher-order separable approximations based on exponential intervening variables

We have previously proposed a family of approximating functions derived from truncated Taylor series expansions in which only the terms on the diagonals of the Hessian and higher-order matrices are retained. This family of approximations was named the incomplete series expansion (ISE) approximations [61]. Since all the coupling off-diagonal terms are dropped, these approximations are separable and have the additional advantage of minimising the number of parameters that need to be stored. Many popular approximations used in SAO frameworks can be thought of as deriving from the ISE as special cases. Foremost among these are the reciprocal and exponential approximations, which are commonly used only to first order. One reason for this is that, if they are retained, the higher-order terms are not convex. In this section we examine the possibility of retaining the higher-order terms.

Before discussing the use of the higher-order approximations, we consider it important to make three points clearly. Firstly, we discuss here the consequences of using the nonconvex terms in a dual approach to SAO, using, specifically, Falk's notion of the dual. We do not intend to imply that these higher-order terms automatically represent improved approximations over the usual

first-order approximations (though they may well, since additional information about the original functions is exploited in formulating the higher-order[1] terms). Nor will we analyse under which circumstances and for which problems the use of the higher-order terms is effective. We only wish to point out whether or not the resulting approximations satisfy the prerequisites for the Falk dual if they are used.

Secondly, we will limit our discussion to SAO subproblems in which all approximations used derive in some way from a general higher-order separable series expansion in terms of exponential intermediate variables (see Section 6.4.1). This expansion can be reduced to strictly convex, strictly concave, linear and generally nonconvex forms, depending on how the parameters are chosen.

Lastly: we restrict our attention to problems whose bound domain $\mathcal{C}$ lies completely within the positive orthant $x_i > 0 \; \forall \; i$. The approximating functions are generally only properly defined over this space, and may contain asymptotes at $x_i = 0$. (A given problem can of course be moved into this space by defining a coordinate translation.) Structural optimisation problems are generally defined only over this space anyway, since the variables in such a problem are normally physical dimensions or material properties, which are non-negative. In the remainder of this chapter, the general nonlinear programming problem considered will be referred to as $\mathrm{P_{NLP}}$. It is assumed to be consistent with (2.15), and is represented in the negative-null form.

6.4.1 Expansion in terms of exponential intervening variables

When approximating $\mathrm{P_{NLP}}$ as a separable expansion in terms of exponential intervening variables, we replace the functions $f_\alpha(\boldsymbol{x})$ by the expressions $\tilde{f}_{\mathrm{E}\alpha}(\boldsymbol{x})$ to form the primal approximate subproblem [61], with

$$\tilde{f}_{\mathrm{E}\alpha}(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left[x_i^{a_{i\alpha}} - \left(x_i^{\{k\}} \right)^{a_{i\alpha}} \right] \left[\frac{\left(x_i^{\{k\}} \right)^{(1-a_{i\alpha})}}{a_{i\alpha}} \right] \left(\frac{\partial f_\alpha}{\partial x_i} \right)^{\{k\}} + \sum_{p=2}^{\bar{p}} \sum_{i=1}^{n} \frac{c_{ip\alpha}}{p!} \left| x_i^{a_{i\alpha}} - \left(x_i^{\{k\}} \right)^{a_{i\alpha}} \right|^p , \tag{6.9}$$

or equivalently,

$$\tilde{f}_{\mathrm{E}\alpha}(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left[\left(\frac{x_i}{x_i^{\{k\}}} \right)^{a_{i\alpha}} - 1 \right] \left(\frac{x_i^{\{k\}}}{a_{i\alpha}} \right) \left(\frac{\partial f_\alpha}{\partial x_i} \right)^{\{k\}} + \sum_{p=2}^{\bar{p}} \sum_{i=1}^{n} \frac{c_{ip\alpha}}{p!} \left| x_i^{a_{i\alpha}} - \left(x_i^{\{k\}} \right)^{a_{i\alpha}} \right|^p . \tag{6.10}$$

[1] It is in order here to mention that higher-order terms may complicate the primal-dual relationships to the extent that simple analytical relationships between the primal and dual variables cannot be formulated. Indeed, the primal-dual relationships may require the solution of one-dimensional minimisations, e.g. see Reference [41]. However, Duysinx [70] reports that the computational effort associated with this may be very reasonable (in particular if a substantial increase in accuracy is indeed realised due to the additional higher-order terms). An alternative computational implementation is to use the quadratic approximations to the approximations with the higher-order terms, e.g. see Reference [71]. This results in a new and simple form of the dual, which does not depend on the specific approximations used. We indeed hope to investigate these approaches in the future.

Notationally, $\alpha = 0$ indicates the approximate objective function, whereas $1 \leq \alpha \leq m$ denotes the corresponding approximate inequality constraint. We have introduced $\bar{p}$ to indicate that the series used contains only a finite number of terms. For the sake of notational simplicity, it is understood that

$$\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} = \frac{\partial f_\alpha}{\partial x_i}(\boldsymbol{x}^{\{k\}}),$$

being the partial derivative of f_α with respect to x_i at the point $\boldsymbol{x}^{\{k\}}$. The convexity of (6.9) depends on the values of the $a_{i\alpha}$ and the $c_{ip\alpha}$, as well as on the signs of the $\partial f_\alpha/\partial x_i$. If the $a_{i\alpha}$ are negative, the second term on the right-hand side of (6.9) is strictly convex for all $\partial f_\alpha/\partial x_i$ negative. Since the third term is nonconvex over the interval $x_i > x_i^{\{k\}}$, we are guaranteed to obtain a *strictly* convex (or strictly concave) approximation only if

$$c_{ip\alpha} = 0 \quad \forall \quad i \text{ and } p. \tag{6.11}$$

If the $a_{i\alpha} > 1$, the first-order terms are strictly convex for $\partial f_\alpha/\partial x_i > 0$, although the higher-order terms are still nonconvex. The expression in terms of exponential intervening variables (6.9) represents a variety of specific approximations that can be obtained by specifying or limiting the parameters $a_{i\alpha}$ and $c_{ip\alpha}$. For instance, by setting $a_{i\alpha} = 1 \; \forall \; i$, we recover the direct approximation [61]

$$\tilde{f}_{\mathrm{D}\alpha}(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}}(x_i - x_i^{\{k\}}) + \sum_{p=2}^{\bar{p}}\frac{1}{p!}\sum_{i=1}^{n} c_{ip\alpha}\left|x_i - x_i^{\{k\}}\right|^p, \tag{6.12}$$

and by setting $a_{i\alpha} = -1 \; \forall \; i$ we recover the reciprocal approximation [61]

$$\tilde{f}_{\mathrm{R}\alpha} = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left(x_i - x_i^{\{k\}}\right)\left(\frac{x_i^{\{k\}}}{x_i}\right)\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} + \sum_{p=2}^{\bar{p}}\sum_{i=1}^{n}\frac{c_{ip\alpha}}{p!}\left|\frac{1}{x_i} - \frac{1}{x_i^{\{k\}}}\right|^p. \tag{6.13}$$

6.4.2 Analysis of a higher-order nonconvex form

We are interested in ascertaining whether or not the higher-order functions listed above can be used in a general dual approach to SAO. The question, then, is whether a general method can be defined for the solution of $\mathrm{P}_{\mathrm{NLP}}$ that utilises these forms. With reference to the attributes listed in Section 2.3.2, we note that these functions are all continuous and differentiable everywhere to first order at least, despite the existence of absolute value operators. Also, we assume that the set $\mathcal{C}$ has the simple structure discussed in Section 2.3.2 for the types of problems that may be considered. In other words, we take it as said that Attributes 2 and 3 hold when we apply these approximations to a problem of interest. In the current section, we examine under which circumstances Attribute 1 also holds.

To this end, we first examine the basic form that the separable parts of the Lagrangian $\mathcal{L}_i$ are likely to take when approximations that derive from (6.9) are used. Hence, we consider a general

function l_f that contains the following terms:

$$l_f(x) = \sum_{\alpha=0}^{m} \lambda_\alpha \left[x^{a_\alpha} - \left(x^{\{k\}}\right)^{a_\alpha} \right] \left[\frac{\left(x^{\{k\}}\right)^{(1-a_\alpha)}}{a_\alpha} \right] \left(\frac{\partial f_\alpha}{\partial x} \right)^{\{k\}} + \sum_{\alpha=0}^{m} \sum_{p=2}^{\bar{p}} \lambda_\alpha \frac{c_{p\alpha}}{p!} \left| x^{a_\alpha} - \left(x^{\{k\}}\right)^{a_\alpha} \right|^p , \tag{6.14}$$

where we have dropped the subscript i. Here, λ_α for $\alpha = 1, 2, \cdots, m$ are the Lagrange multipliers associated with the j constraints. They are always positive constants. Since $\alpha = 0$ denotes the objective function, $\lambda_0 = 1$.

When $c_{p\alpha} < 0$, the associated term in the Lagrangian is a strictly concave and increasing function over the interval $x \in (0, x^{\{k\}})$ and monotonically decreasing over $(x^{\{k\}}, \infty)$. Also, a first-order term is concave and monotonically decreasing whenever a_α is positive and $(\partial f_\alpha/\partial x)^{\{k\}}$ is negative. Since, for the moment, we want $(\partial f_\alpha/\partial x)^{\{k\}}$ to be able to take on either a positive or a negative sign, to ensure that l_f has a unique minimum in general it is necessary to require that

$$c_{p\alpha} \geq 0 \quad \forall \quad p\,,$$

and that

$$a_\alpha < 0 \quad \forall \quad \alpha\,.$$

Equation (6.14) stems from the use of the general exponential expression (6.9). As it is given, the powers a_α may have different values for every α. This being the case, it is quite easy to find examples for which Attribute 1 does not hold for l_f in general (even if only the first-order terms are present). Hence, another stipulation that must be made immediately is that

$$a_\alpha = a \quad \forall \quad \alpha\,.$$

With these preliminary considerations taken into account, and defining the constants

$$A = \sum_{\alpha=0}^{m} \lambda_\alpha \left[\frac{\left(x^{\{k\}}\right)^{(1-a)}}{a} \right] \left(\frac{\partial f_\alpha}{\partial x} \right)^{\{k\}}$$

and

$$b_p = \sum_{\alpha=0}^{m} \lambda_\alpha \frac{c_{p\alpha}}{p!}$$

at the point $\boldsymbol{x}^{\{k\}}$, we are left with a function of the form

$$l_f = A \left[x^a - \left(x^{\{k\}}\right)^a \right] + \sum_{p=2}^{\bar{p}} b_p \left| x^a - \left(x^{\{k\}}\right)^a \right|^p . \tag{6.15}$$

To simplify the discussion that follows we choose to write

$$l_{fA} = A \left[x^a - \left(x^{\{k\}}\right)^a \right]$$

and

$$l_{fB} = \sum_{p=2}^{\bar{p}} b_p \left| x^a - \left(x^{\{k\}}\right)^a \right|^p .$$

Since $a < 0$ and $b_p \geq 0$, each term in the sum l_{fB} has the general form depicted in Figure 6.2(b), regardless of the exponent p. We demonstrate below that these functions are convex and decreasing over $x \in (0, x^{\{k\}})$, and nonconvex but monotonically increasing on the interval $x \in (x^{\{k\}}, \infty)$. They possess unique minima, which are located at $x = x^{\{k\}}$. As $x \to +\infty$, these functions tend towards $b_p \left(x^{\{k\}}\right)^{ap}$ asymptotically. Of course, the sum itself has the same general characteristics as its constituent functions.

As exemplified in Figure 6.2(a), the first term in (6.15), namely l_{fA}, is either convex and monotonically decreasing or concave and monotonically increasing, depending on the sign of A. Regardless of the sign of A, or of the values that A or the various b_p might take, l_f is a function that possesses a unique minimum over $x \in [\check{x}_i, \hat{x}_i]$. This is easy to see in the case that $A = 0$ or all $b_p = 0$. However, to verify the uniqueness of the minimum if neither of these eventualities transpires, note firstly that the gradient of l_f is

$$\frac{\partial (l_f)}{\partial x} = a x^{a-1} \left[A + \sum_{p=2}^{\bar{p}} p b_p \left[x^a - \left(x^{\{k\}}\right)^a \right] \left| x^a - \left(x^{\{k\}}\right)^a \right|^{p-2} \right], \tag{6.16}$$

or equivalently

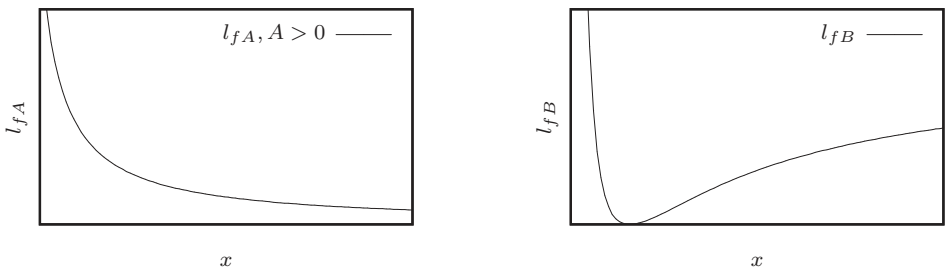
$$\frac{\partial (l_f)}{\partial x} = a x^{a-1} \left[A + \sum_{p=2}^{\bar{p}} p b_p s(x) \left| x^a - \left(x^{\{k\}}\right)^a \right|^{p-1} \right], \tag{6.17}$$

in which $s(x)$ is an operator that assumes the sign of $\left[x^a - \left(x^{\{k\}}\right)^a\right]$. Again, for the sake of clarity we write

$$D_B = \sum_{p=2}^{\bar{p}} p b_p s(x) \left| x^a - \left(x^{\{k\}}\right)^a \right|^{p-1}$$

and

$$D_T = \left[A + \sum_{p=2}^{\bar{p}} p b_p s(x) \left| x^a - \left(x^{\{k\}}\right)^a \right|^{p-1} \right].$$



(a) First-order term l_{fA} (b) Higher-order terms l_{fB}

Figure 6.2: The general form of l_{fA} and l_{fB} with $a < 0$.

Properties of D_B

Clearly, $x^a > \left(x^{\{k\}}\right)^a$ when $x < x^{\{k\}}$ and the difference between the two $\left[x^a - \left(x^{\{k\}}\right)^a\right]$ decreases monotonically to zero as x approaches $x^{\{k\}}$ from below. From this, it is evident that all the terms of which D_B is comprised have a strictly positive value on the interval $x \in (0, x^{\{k\}})$. D_B inherits this property and also decreases monotonically to zero as x approaches $x^{\{k\}}$ from below. Because x^{a-1} is also a decreasing function with a positive value, the multiple of the two, namely

$$x^{a-1} D_B = x^{a-1} \sum_{p=2}^{\bar{p}} p b_p s(x) \left| x^a - \left(x^{\{k\}}\right)^a \right|^{p-1}, \tag{6.18}$$

can only be a function of the same type. When the negative factor a is taken into account, we may conclude that the function represented by l_{fB} is convex over $x \in (0, x^{\{k\}})$, since its gradient is strictly negative and monotonically increasing over this interval.

Considering the interval $x \in (x^{\{k\}}, \infty)$, it is sufficient to point out that (6.18) is always negative when $x > x^{\{k\}}$, making l_{fB} a monotonically increasing function on $(x^{\{k\}}, \infty)$. Also, realise that, since D_B has essentially the same structure as l_{fB} over this interval, except that $s(x)$ is negative here, D_B must be a negative-valued and monotonically decreasing function in this region. D_B decreases to some limiting value asymptotically.

Now we examine l_f for two cases characterised by the sign of A.

For $A > 0$

In this case, l_{fA} corresponds to a decreasing reciprocal function of order $|a|$, and is strictly convex. Given the convexity of l_{fB} on the interval $x \in (0, x^{\{k\}})$, l_f must itself be convex there. Given the facts that $A > 0$, $D_B = 0$ at $x = x^{\{k\}}$, and that D_B is monotonically decreasing on $(x^{\{k\}}, \infty)$, we conclude that D_T is also monotonically decreasing on $(x^{\{k\}}, \infty)$. It has a positive value at $x = x^{\{k\}}$ and can pass through zero at most once if A is greater than the absolute value of the limit to which D_B converges. We call this point at which D_T equals zero x^*, if it exists, and we know that $x^* > x^{\{k\}}$.

Now, on the interval $(x^{\{k\}}, x^*)$, both D_T and x^{a-1} are positive-valued decreasing functions, which implies once again that l_f is convex in this region. For $x > x^*$, D_T is strictly negative, meaning that the gradient of l_f is strictly positive, implying that l_f is monotonically increasing on (x^*, ∞).

In summary, for $A > 0$: because l_f is convex and decreasing over $x < x^*$, and monotonically increasing over $x > x^*$, l_f can have only one minimum. This minimum is located at x^* if it exists. Moreover, these facts imply that l_f has a unique minimum over any convex and closed bounded interval. If x^* does not exist, l_f is convex and monotonically decreasing over the whole real line, in which case the minimum of l_f over a bounded interval will be located at the upper bound on the interval $\hat{x}_i$.

For $A < 0$

Both l_{fA} and l_{fB} are monotonically increasing functions on $x > x^{\{k\}}$ in this case and, consequently, so is l_f. Therefore, we focus on the interval $(0, x^{\{k\}})$ in which l_{fA} is monotonically

increasing and strictly concave, whereas l_{fB} is monotonically decreasing and convex.

It is evident that D_B is here a positive-valued monotonically decreasing function, so we can conclude that there is once again a unique point x^* that makes $D_T = 0$, except that this time $x^* < x^{\{k\}}$. By following a similar rationale as in the case for $A > 0$, we again are led to conclude that l_f is convex and decreasing over $x < x^*$ and monotonically increasing over $x > x^*$. For $A < 0$, the point x^* is bound to exist and defines the unbounded unique minimum of l_f.

6.5 Methods of mixed variables

The methods presented here are based on the inverse exponential form discussed above, which is pertinent to the weight minimisation problem. That is to say, we make the assumption that these methods must be able to incorporate general inverse exponential forms (in which the exponents are negative). Therefore, we take l_f as a basis and we investigate which other forms can be added in such a way as to guarantee that the Lagrangian functions $\mathcal{L}_i$ still have unique minima. These methods are meant to be general. That is, each of them incorporates a range of specific approximations, which are obtained by restricting or specifying parameter values.

6.5.1 Incorporating additional functions into l_f

We have shown that functions of the form given in (6.15) have a unique minimum regardless of the sign of A, provided that $a < 0$ and all $b_p > 0$. This is to say that a Lagrangian, separably composed of terms of the form given in (6.14), has a unique minimum on any interval of the form $x_i \in [\check{x}_i, \hat{x}_i]$ regardless of the signs of the partial derivatives, provided that all $a_{i\alpha}$ are identical and negative for a given i and that all $c_{ip\alpha} \geq 0$.

The Lagrangian associated with a problem $\mathrm{P_{NLP}}$ would take this form if the objective and constraint functions were all approximated as functions consistent with the following expression:

$$\tilde{f}_{\mathrm{E}\alpha}(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left[x_i^{a_i} - \left(x_i^{\{k\}}\right)^{a_i}\right] \left[\frac{\left(x_i^{\{k\}}\right)^{(1-a_i)}}{a_i}\right] \left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} + \sum_{p=2}^{\bar{p}} \sum_{i=1}^{n} \frac{c_{ip\alpha}}{p!} \left|x_i^{a_i} - \left(x_i^{\{k\}}\right)^{a_i}\right|^p . \tag{6.19}$$

Equation (6.19) is a restricted version of the general expression in terms of exponential intermediate variables (6.9). However, it can be thought of more properly as a generalisation of the reciprocal approximation (6.13) to other fixed negative exponents.

Equation (6.15) is convex at first and strictly increasing thereafter. The addition of any other term to l_f that is convex over the same interval as l_f is convex, and also strictly increasing thereafter, would not change the basic structure of (6.15). The resulting function would still have a unique minimum. Remember that if $l_{fB} = 0$, $l_f = l_{fA}$ could be concave but increasing everywhere or convex but decreasing everywhere, so appropriate choices of functions are those that are both

convex and strictly increasing over all $x > 0$. Therefore, for instance, one could add to (6.15) terms of the form

$$l_{fC} = d_C \left[x^q - \left(x^{\{k\}} \right)^q \right]$$

in which $d_C > 0$ and $q \geq 1$. Such terms come from exponential approximations truncated to first order

$$\tilde{f}_{\mathrm{E}\beta}(\boldsymbol{x}) = f_\beta(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left[x_i^{q_{i\beta}} - \left(x_i^{\{k\}} \right)^{q_{i\beta}} \right] \left[\frac{\left(x_i^{\{k\}} \right)^{(1-q_{i\beta})}}{q_{i\beta}} \right] \left(\frac{\partial f_\beta}{\partial x_i} \right)^{\{k\}}, \tag{6.20}$$

for which the following restriction holds:

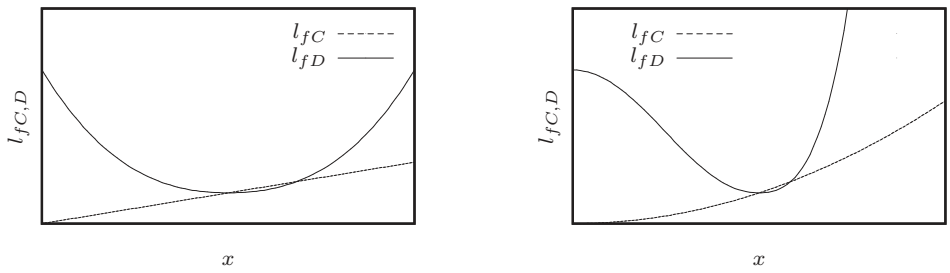
$$\frac{\partial f_\beta}{\partial x_i} > 0 \,.$$

Lastly, if one were to insist that $(\partial f_\alpha / \partial x_i) < 0$ were to apply strictly to (6.19), in which $a_i < 0$, then terms of the form

$$l_{fD} = d_D \left| x^q - \left(x^{\{k\}} \right)^q \right|^p$$

could also be added to (6.15), provided that $d_D > 0$, $q \geq 1$ and $p \geq 2$. With these restrictions, functions of the form l_{fD} are nonconvex but monotonically decreasing over $\left(0, x^{\{k\}}\right)$, unless $q = 1$, in which case they are convex and decreasing over $\left(0, x^{\{k\}}\right)$ (see Figure 6.3). In either case they are convex and increasing over $\left(x^{\{k\}}, \infty\right)$. The observation that they can be added to l_f as additional terms stems from:

- For l_f, $x^* > x^{\{k\}}$ in this case.
- Both l_f and l_{fD} are monotonically decreasing and positive-valued over $\left(0, x^{\{k\}}\right)$.
- Both l_f and l_{fD} are convex over $\left(x^{\{k\}}, x^*\right)$.
- Both l_f and l_{fD} are monotonically increasing and positive-valued over (x^*, ∞).



(a) When $q = 1$ (b) When $q > 1$

Figure 6.3: The general form of l_{fC} and l_{fD} with $q \geq 1$.

Functions of the form l_{fD} come from the higher-order terms in a general exponential expansion in which the exponents are greater than unity. This would appear to imply that, when $(\partial f_\alpha/\partial x_i) < 0$ in (6.19), the full exponential form

$$\tilde{f}_{\mathrm{E}\beta}(\boldsymbol{x}) = f_\beta(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left[x_i^{q_{i\beta}} - \left(x_i^{\{k\}}\right)^{q_{i\beta}}\right]\left[\frac{\left(x_i^{\{k\}}\right)^{(1-q_{i\beta})}}{q_{i\beta}}\right]\left(\frac{\partial f_\beta}{\partial x_i}\right)^{\{k\}} + \sum_{p=2}^{\bar{p}}\sum_{i=1}^{n}\frac{c_{ip\beta}}{p!}\left|x_i^{q_{i\beta}} - \left(x_i^{\{k\}}\right)^{q_{i\beta}}\right|^p, \tag{6.21}$$

with $q_{i\beta} \geq 1$, all $c_{ip\beta} \geq 0$ and all $(\partial f_\beta/\partial x_i) > 0$, could also be utilised for the approximation of some functions f_β in $\mathrm{P_{NLP}}$. Unfortunately, this is not the case. Although the use of either of the forms l_{fC} and l_{fD} presents no problems individually, it is possible to find cases of Lagrangian functions $\mathcal{L}_i$, derived from (6.21), which do not have unique minima. If we wish to use such approximations, we are forced to impose

$$q_{i\beta} = q_i \quad \forall \quad \beta\,.$$

Then the approximation becomes

$$\tilde{f}_{\mathrm{E}\beta}(\boldsymbol{x}) = f_\beta(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left[x_i^{q_i} - \left(x_i^{\{k\}}\right)^{q_i}\right]\left[\frac{\left(x_i^{\{k\}}\right)^{(1-q_i)}}{q_i}\right]\left(\frac{\partial f_\beta}{\partial x_i}\right)^{\{k\}} + \sum_{p=2}^{\bar{p}}\sum_{i=1}^{n}\frac{c_{ip\beta}}{p!}\left|x_i^{q_i} - \left(x_i^{\{k\}}\right)^{q_i}\right|^p, \tag{6.22}$$

which is again a restricted version of the general exponential expression (6.9), and can be thought of as a generalisation of the direct approximation (6.12) to other positive exponents. In (6.22), all $(\partial f_\beta/\partial x_i) > 0$, all $c_{ip\beta} \geq 0$ and all $q_i \geq 1$, and (6.22) can be used together with (6.19) whenever all $(\partial f_\alpha/\partial x_i) \leq 0$ in (6.19).

We will not demonstrate explicitly that (6.22) always results in Lagrangian functions that have unique minima. To do so would entail defining a function l_{f2} from the sum of forms l_{fC} and l_{fD}. An analysis of l_{f2} would be similar to the one given in Section 6.4.2, for l_f, except that, instead of $a < 0$, we have $q \geq 1$ and the coefficient A would always be non-negative. Suffice it to say that, if both l_{fC} and l_{fD} are present in l_{f2}, then l_{f2} would have the following characteristics:

- A point $x^\dagger$ may exist at which $(\partial l_{f2}/\partial x) = 0$.
- $x^\dagger < x^{\{k\}}$.
- l_{f2} is monotonically decreasing over $\left(0, x^\dagger\right)$.
- l_{f2} is convex and strictly increasing over $\left(x^\dagger, \infty\right)$.
- If $x^\dagger$ does not exist, then l_{f2} is convex and strictly increasing everywhere.

- Given the above, $(l_f + l_{f2})$ also has a unique minimum on any convex bounded interval.

These considerations motivate four possible courses of action for applying approximations derived from the exponential expression (6.9) in Section 6.4 within a general SAO framework. The first is simply to use the generalised reciprocal approximation with higher-order terms (6.19) to approximate all the functions f_α in $\mathrm{P_{NLP}}$, irrespective of the signs of $(\partial f_\alpha/\partial x_i)$. In this case, the exponents a_i must be chosen a priori as negative numbers, and the same a_i must be used for every $\tilde{f}_\alpha$. The other three approaches are methods of mixed variables.

With each of the three methods listed below we develop the general form of the approximating function specific to that method. It is crucial to understand that these methods require that the same function approximation is applied to *every* function in a given problem. This ensures that the Lagrangian functions associated with the approximate subproblem will have unique minima. The approximations are quite general, however, so considerable scope is present for tailoring the functions by restricting or specifying the various parameters. This will become clearer in Section 6.6.

6.5.2 An almost convex method of mixed variables

The second option is the most obvious, and requires that sets $\mathcal{S}_\alpha^a$ and $\mathcal{S}_\alpha^q$ are determined solely by the sign of the partial gradients of the functions in $\mathrm{P_{NLP}}$ at $\boldsymbol{x}^{\{k\}}$. Ergo, for each of the functions f_α, the sets are defined according to

$$\begin{aligned}\mathcal{S}_\alpha^a =& \{i : \frac{\partial f_\alpha}{\partial x_i} \leq 0,\ i = 1, 2, \cdots, n\},\\ \mathcal{S}_\alpha^q =& \{i : \frac{\partial f_\alpha}{\partial x_i} > 0,\ i = 1, 2, \cdots, n\}.\end{aligned}$$

With the sets so defined, we can apply an approximation that is itself a combination of the two generalised approximations (6.19) with $a_{i\alpha} = a_i < 0$, and (6.22) with $q_{i\alpha} = q_i \geq 1$:

$$\begin{aligned}\tilde{f}_{\mathrm{AM}\alpha} = f_\alpha(\boldsymbol{x}^{\{k\}}) &+ \sum_{i\in\mathcal{S}_\alpha^q}\left[x_i^{q_i} - \left(x_i^{\{k\}}\right)^{q_i}\right]\left[\frac{\left(x_i^{\{k\}}\right)^{(1-q_i)}}{q_i}\right]\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}}\\ &+ \sum_{i\in\mathcal{S}_\alpha^q}\sum_{p=2}^{\bar{p}}\frac{c_{ip\alpha}}{p!}\left|x_i^{q_i} - \left(x_i^{\{k\}}\right)^{q_i}\right|^p\\ &+ \sum_{i\in\mathcal{S}_\alpha^a}\left[x_i^{a_i} - \left(x_i^{\{k\}}\right)^{a_i}\right]\left[\frac{\left(x_i^{\{k\}}\right)^{(1-a_i)}}{a_i}\right]\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}}\\ &+ \sum_{i\in\mathcal{S}_\alpha^a}\sum_{p=2}^{\bar{p}}\frac{c_{ip\alpha}}{p!}\left|x_i^{a_i} - \left(x_i^{\{k\}}\right)^{a_i}\right|^p.\end{aligned} \tag{6.23}$$

We call this an 'almost convex' approximation because the only nonconvex terms are the higher-order terms. If the positive-valued parameters $c_{ip\alpha}$ are all small, then the deviation from convexity of the associated Lagrangian functions is likely to be correspondingly small or even non-existent within the allowable bounds.

6.5.3 A partial method of mixed variables

The third possible course to follow assumes that something more is known about the forms of the functions in $\mathrm{P_{NLP}}$. This knowledge is again used to split the terms in the series expansion of each of the functions f_α into two sets, $\mathcal{S}_\alpha^a$ and $\mathcal{S}_\alpha^q$. Once more, for all terms contained in $\mathcal{S}_\alpha^q$ it must be true that

$$\frac{\partial f_\alpha}{\partial x_i} \; > \; 0\,.$$

However, we now allow $(\partial f_\alpha / \partial x_i)$ to be positive or negative for the terms contained in $\mathcal{S}_\alpha^a$. Then, it is possible to apply the following general approximation

$$\begin{aligned}\tilde{f}_{\mathrm{PM}\alpha} = f_\alpha(\boldsymbol{x}^{\{k\}}) &+ \sum_{i\in\mathcal{S}_\alpha^q}\left[x_i^{q_{i\alpha}} - \left(x_i^{\{k\}}\right)^{q_{i\alpha}}\right]\left[\frac{\left(x_i^{\{k\}}\right)^{(1-q_{i\alpha})}}{q_{i\alpha}}\right]\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} \\ &+ \sum_{i\in\mathcal{S}_\alpha^a}\left[x_i^{a_i} - \left(x_i^{\{k\}}\right)^{a_i}\right]\left[\frac{\left(x_i^{\{k\}}\right)^{(1-a_i)}}{a_i}\right]\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} \\ &+ \sum_{i\in\mathcal{S}_\alpha^a}\sum_{p=2}^{\bar{p}}\frac{c_{ip\alpha}}{p!}\left|x_i^{a_i} - \left(x_i^{\{k\}}\right)^{a_i}\right|^p,\end{aligned} \tag{6.24}$$

which is composed of the generalised reciprocal approximation (6.19), complete with its restrictions on $a_{i\alpha}$, and the truncated first-order exponential approximation (6.20), in which all $q_{i\alpha} \geq 1$. Here $q_{i\alpha}$ may take different values for different α and i. Naturally, if the $\mathcal{S}_\alpha^q$ are empty for all α, the partial method in effect reduces to our first approximation strategy: the application of (6.19) for the approximation of all functions in $\mathrm{P_{NLP}}$.

We call this a 'partial method' of mixed variables because the components of the functions do not have to be partitioned solely according to the signs of their partial gradients. It may be possible to exploit additional information about the functions in applying the set-partitioning strategy. We have already seen an example of the application of this method. Fleury's original approach to the weight minimisation problem (6.1), described in Section 6.3, is an example of this method in which all $a_i = -1$, all $q_{i\alpha} = 1$ and all $c_{ip\alpha} = 0$. There, all of the components of the objective function f_0 were placed in set $\mathcal{S}_0^q$, whereas the components of the constraint functions were all placed in sets $\mathcal{S}_\alpha^a$, and such a partitioning strategy is but a special case of (6.24).

An extension of the weight minimisation problem was presented in Reference [4] as a motivation for the introduction of CONLIN. This extension involves adding an additional set of linear constraints to (6.1). The solution approach detailed in Section 6.3 is inadequate for this new problem, because the new constraints can appear as negative linear terms in the Lagrangian functions (6.2), which generally destroys the uniqueness of their minima. The above partial method of mixed variables (6.24) can be used to solve this extended weight minimisation problem if all new linear components with negative gradients join a set $\mathcal{S}_\alpha^a$, while all new linear terms with positive gradients join a set $\mathcal{S}_\alpha^q$. Otherwise, the original set-partitioning strategy for (6.1) still applies.

6.5.4 A strictly convex method of mixed variables

Lastly, one can define a method of mixed variables using the functions discussed in which the Lagrangian functions are always strictly convex. This comes about when sets $\mathcal{S}_\alpha^a$ and $\mathcal{S}_\alpha^q$ are defined as

$$\mathcal{S}_\alpha^a = \{i : \frac{\partial f_\alpha}{\partial x_i} \leq 0,\ i = 1, 2, \cdots, n\},$$
$$\mathcal{S}_\alpha^q = \{i : \frac{\partial f_\alpha}{\partial x_i} > 0,\ i = 1, 2, \cdots, n\},$$

and an exponential approximation is used to first order that incorporates both inverse terms $a_{i\alpha} < 0$ and power terms $q_{i\alpha} \geq 1$. The resulting approximation is

$$\tilde{f}_{\text{SM}\alpha} = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i \in \mathcal{S}_\alpha^q} \left[x_i^{q_{i\alpha}} - \left(x_i^{\{k\}} \right)^{q_{i\alpha}} \right] \left[\frac{\left(x_i^{\{k\}} \right)^{(1-q_{i\alpha})}}{q_{i\alpha}} \right] \left(\frac{\partial f_\alpha}{\partial x_i} \right)^{\{k\}}$$
$$+ \sum_{i \in \mathcal{S}_\alpha^a} \left[x_i^{a_{i\alpha}} - \left(x_i^{\{k\}} \right)^{a_{i\alpha}} \right] \left[\frac{\left(x_i^{\{k\}} \right)^{(1-a_{i\alpha})}}{a_{i\alpha}} \right] \left(\frac{\partial f_\alpha}{\partial x_i} \right)^{\{k\}}. \tag{6.25}$$

At first glance, equation (6.25) looks like a special case of the almost convex method (6.23) in which all $c_{ip\alpha} = 0$. However, the omission of the higher-order terms allows us to drop the restrictions on the exponents, which are part and parcel of (6.23). In this case, both $a_{i\alpha}$ and $q_{i\alpha}$ may take different values for different α and i, since all the functions involved are strictly convex.

6.6 Duality

We have presented a number of strategies for the approximation of P_{NLP} (2.15) at a point $\boldsymbol{x}^{\{k\}}$, in such a way as to ensure that Falk's dual method can be used to solve the resulting primal approximate subproblem $P_P[k]$, provided that its feasible region is non-empty. This itself implies that $P_P[k]$ has a unique minimum (see Falk [2]), even though, in our case, it is not necessarily convex. By construction, the form of $P_P[k]$ is consistent with (2.39), which is here re-presented in the negative-null form.

Primal approximate subproblem $P_P[k]$

$$\begin{aligned} \min\ & \tilde{f}_0(\boldsymbol{x}) \\ \text{subject to}\ & \tilde{f}_j(\boldsymbol{x}) \leq 0 \qquad j = 1, 2, \cdots, m, \\ & \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{6.26}$$

In the notation $P_P[k]$, k denotes the iteration index ($k = 1, 2, 3, \cdots$). Consequently, $\boldsymbol{x}^{\{k\}}$ is the optimum of problem $P_P[k-1]$, at which the new subproblem $P_P[k]$ is defined. We now describe explicitly how to go about defining and solving the dual approximate subproblem $P_D[k]$

for a few particular instances of the application of the methods of mixed variables outlined in Section 6.5. Following the material presented in Section 2.3.1, we start by constructing the Lagrangian $\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda})$ in terms of our function approximations, given as

$$\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda}) = \tilde{f}_0^{\{k\}}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \tilde{f}_j^{\{k\}}(\boldsymbol{x}) \,, \tag{6.27}$$

where the λ_j, $j = 1, 2, \cdots, m$ represent the Lagrangian multipliers; λ_j may be understood to be indicative of the sensitivity of $\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda})$ to constraint j. From Falk [2], as well as our observations in Section 2.3.2 and Reference [35], if Attributes 1 through 3 hold for primal approximate subproblem (6.26), then the stationary saddle point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ of $\mathcal{L}^{\{k\}}$ defines the global minimiser $\boldsymbol{x}^*$ of $P_P[k]$. The definition of the saddle point (i.e. the KKT conditions satisfied by $\boldsymbol{x}^*$) also needs to take the bound constraints into account, because they are not included in the definition of $\mathcal{L}^{\{k\}}$. For such a treatment of the KKT conditions, refer to Hadley [22], where lower bounds $x_i = 0$ are considered. The saddle point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ is given by

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x}} \{\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda}) : \check{x}_i \leq x_i \leq \hat{x}_i\} = \max_{\boldsymbol{\lambda}} \gamma(\boldsymbol{\lambda}) \,, \tag{6.28}$$

where the bound constraints represent a *closed* and *bounded* set. The function $\gamma(\boldsymbol{\lambda})$ defines the Falk dual [2, 28]. A crucial requirement for the construction of *efficient* dual formulations is that the primal approximate subproblem is formulated in terms of *separable* approximations. In this case, the primal-dual relationships

$$\boldsymbol{x}(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{x}} \{\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda}) : \check{x}_i \leq x_i \leq \hat{x}_i\} \tag{6.29}$$

are determined by a set of n one-dimensional minimisations. We obtain $\gamma(\boldsymbol{\lambda})$ in terms of the approximation functions as

$$\begin{aligned} \gamma(\boldsymbol{\lambda}) =& \mathcal{L}^{\{k\}}\left(\boldsymbol{x}\left(\boldsymbol{\lambda}\right), \boldsymbol{\lambda}\right) \\ =& \min_{\boldsymbol{x}} \left\{ \left[\tilde{f}_0(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \tilde{f}_j(\boldsymbol{x}) \right] : \check{x}_i \leq x_i \leq \hat{x}_i,\ i = 1, 2, \cdots, n \right\}. \end{aligned} \tag{6.30}$$

With the assumption of separability, it is always possible to express the $x_i\left(\boldsymbol{\lambda}\right)$ that minimise (6.30) independently, in the form

$$x_i = x_i(\boldsymbol{\lambda}) : \check{x}_i \leq x_i \leq \hat{x}_i, \qquad i = 1, 2, \cdots, n. \tag{6.31}$$

The saddle point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ is then found by maximising the dual using (6.31), so the dual approximate subproblem becomes

Dual approximate subproblem $P_D[k]$

$$\begin{aligned} &\max_{\boldsymbol{\lambda}} \{\gamma(\boldsymbol{\lambda}) = \tilde{f}_0(\boldsymbol{x}(\boldsymbol{\lambda})) + \sum_{j=1}^{m} \lambda_j \tilde{f}_j(\boldsymbol{x}(\boldsymbol{\lambda}))\}, \\ &\text{subject to } \lambda_j \geq 0, \qquad j = 1, 2, \cdots, m, \end{aligned} \tag{6.32}$$

with the $\tilde{f}_\alpha(\boldsymbol{x}(\boldsymbol{\lambda}))$, $\alpha = 0, 1, \cdots, m$ represented by any suitable combination of approximations, which in this chapter are assumed to derive from (6.9). If negative (inverse) exponential forms are present, the word 'suitable' implies that said combination is generally represented by one of the methods discussed in Section 6.5. However, as we have mentioned, other combinations are no doubt possible for particular cases in which the parameters are tailored specifically.

This simply constrained problem requires the determination of the m unknowns λ_j only, subject to m non-negativity constraints on the λ_j. Recall that the primal approximate subproblem (6.26) has n unknowns, m constraints, and $2n$ side constraints. Hence, the solution of the dual approximate subproblem (6.32) is far more efficient than the solution of $P_P[k]$ if $m \ll n$. In structural optimisation, a well-known example in which the dual method is efficiently applied is the (classical) minimum compliance optimisation problem. For many other (structural) optimisation problems, the number of effective constraints may be reduced using a constraint deletion strategy, which has no effect on the final outcome whatsoever. Furthermore, even for $m \approx n$, the dual approach may still be expected to be efficient when compared to primal methods, because the dual is 'essentially unconstrained'. Finally, and obviously, if a given subproblem is unconstrained (in that no approximate constraints are active), dual problem (6.32) still holds.

We will remark on suitable solvers for dual problem (6.32) in Section 6.6.3. First, though, the x_i that minimise (6.30), as given in (6.31), are derived for a few simple, illustrative cases based on the weight minimisation problem with sizing design variables.

6.6.1 Weight minimisation with sizing design variables

Weight minimisation problems with sizing design variables are often formulated with linear objective functions, subject to nonlinear stress and displacement constraint functions; the objective and constraint functions exhibit monotonicity with respect to the design variables, and are either exactly or approximately known. In exploiting this knowledge, we would like to use the linear approximation with direct (design) variables to describe the objective function, since this is exact, and (other) exponential intervening variables for the constraints.

In fact, it may be counterproductive to approximate a linear objective by a nonlinear function, in that the complexity of the approximate subproblem becomes unnecessarily high. Consider, for example, a linear (univariate) objective function in x, approximated by a reciprocal intervening variable $y = 1/x$. As $x \to 0$, the approximation becomes increasingly inaccurate. The reader is referred to the argument put forward by Groenwold *et al.* 'there is no free lunch in function approximation,' briefly outlined in Reference [61].

Weight minimisation using a conservative mixed approximation

Along the lines of our arguments above, we use (6.12) for the objective function. Note that, for the weight minimisation problem, the order of the approximation $\bar{p}$ in (6.12) is irrelevant, as $c_{0i}^{\{k\}}$ will all be zero if the conditions we have previously proposed in Reference [61] are used to determine the $c_{0i}^{\{k\}}$ in (6.12). We use reciprocal intermediate variables (i.e. the exponential form with $a_{i\alpha} = -1$) to represent the components of the constraints that have negative partial gradients, while we express those that have positive partial gradients as linear functions of the direct variables.

For the sake of brevity, we only present the simplest possible dual formulation here, being of order 1 for the constraint approximations, which means that the method of approximation just described is equivalent to the application of CONLIN. The resulting dual approximate subproblem (6.32) has the form

$$\begin{aligned}\gamma(\boldsymbol{\lambda}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \frac{\partial f_0^{\{k\}}}{\partial x_i}(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}) \\ + \sum_{j=1}^{m} \lambda_j \left\{ f_j(\boldsymbol{x}^{\{k\}}) + \sum_{i \in \mathcal{S}_j^a} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right) \left(\frac{x_i^{\{k\}}}{x_i(\boldsymbol{\lambda})}\right) \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \right. \\ \left. + \sum_{i \in \mathcal{S}_j^q} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right) \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \right\}.\end{aligned} \tag{6.33}$$

This dual problem results, as a particular restriction, from the application of either (6.23), (6.24) or indeed of (6.25) to the weight minimisation problem. All three can produce (6.33) by enforcing various restrictions. Although not explicitly indicated, all of the components of the objective function are, of course, in set $\mathcal{S}_0^q$. As defined here, the approximate subproblem is convex with respect to all x_i. We apply the stationary conditions

$$\frac{\partial}{\partial x_i}\mathcal{L}^{\{k\}}(\boldsymbol{x}, \boldsymbol{\lambda}) = 0 \tag{6.34}$$

and define

$$\beta_i(\boldsymbol{\lambda}) = -\left(\frac{\partial f_0^{\{k\}}}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \left.\frac{\partial f_j^{\{k\}}}{\partial x_i}\right|_{i \in \mathcal{S}_j^q}\right)^{-1} \left(\sum_{j=1}^{m} \lambda_j (x_i^{\{k\}})^2 \left.\frac{\partial f_j^{\{k\}}}{\partial x_i}\right|_{i \in \mathcal{S}_j^a}\right), \tag{6.35}$$

where, for the sake of notational simplicity, it is understood that

$$\left.\frac{\partial f_j^{\{k\}}}{\partial x_i}\right|_{i \in \theta} = \begin{cases} \dfrac{\partial f_j^{\{k\}}}{\partial x_i} & \text{if } i \in \theta, \\ 0 & \text{if } i \notin \theta. \end{cases} \tag{6.36}$$

Then we obtain

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i^{1/2}(\boldsymbol{\lambda}) & \text{if} \quad \check{x}_i^2 < \beta_i(\boldsymbol{\lambda}) < \hat{x}_i^2\,, \\ \check{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \le \check{x}_i^2\,, \\ \hat{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \ge \hat{x}_i^2\,, \end{cases} \tag{6.37}$$

for $i = 1, 2, \cdots, n$. These are the analytical expressions for evaluating the $x_i\,(\boldsymbol{\lambda})$ in (6.29), and are all that is required for solving dual problem (6.32) when using the above approximation functions. Incidentally, in the weight minimisation problem,

$$\frac{\partial f_0^{\{k\}}}{\partial x_i} > 0 \quad \forall \quad i\,.$$

Hence the denominator in (6.35) always exists, since

$$\sum_{j=1}^{m} \lambda_j \left. \frac{\partial f_j^{\{k\}}}{\partial x_i} \right|_{i \in \mathcal{S}_j^q} \geq 0 \, .$$

Alternatively, using a two-point mixed exponential approximation with negative exponents (still of order 1), rather than reciprocal intermediate variables, we obtain a maximisation problem in which

$$\begin{aligned} \gamma(\boldsymbol{\lambda}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \frac{\partial f_0^{\{k\}}}{\partial x_i}(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}) \\ + \sum_{j=1}^{m} \lambda_j \left\{ f_j(\boldsymbol{x}^{\{k\}}) + \sum_{i \in \mathcal{S}_j^a} \left[\left(\frac{x_i(\boldsymbol{\lambda})}{x_i^{\{k\}}} \right)^{a_{ij}^{\{k\}}} - 1 \right] \left(\frac{x_i^{\{k\}}}{a_{ij}^{\{k\}}} \right) \left(\frac{\partial f_j}{\partial x_i} \right)^{\{k\}} \right. \\ \left. + \sum_{i \in \mathcal{S}_j^q} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}} \right) \left(\frac{\partial f_j}{\partial x_i} \right)^{\{k\}} \right\} . \end{aligned} \tag{6.38}$$

This is seen as a special case of the mixed variable method (6.25). We have introduced the superscript $\{k\}$ in $a_{ij}^{\{k\}}$ to indicate that $a_{ij}^{\{k\}}$ is determined at the inception of iteration k. This time, application of the stationary conditions (6.34) results in

$$\frac{\partial f_0^{\{k\}}}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \left. \frac{\partial f_j^{\{k\}}}{\partial x_i} \right|_{i \in \mathcal{S}_j^q} + \sum_{j=1}^{m} \lambda_j x_i^{(a_{ij}^{\{k\}} - 1)} \left(x_i^{\{k\}} \right)^{1 - a_{ij}^{\{k\}}} \left. \frac{\partial f_j^{\{k\}}}{\partial x_i} \right|_{i \in \mathcal{S}_j^a} = 0 \, .$$

Hence, we have nonlinear expressions in x_i of the form

$$b_i(\boldsymbol{\lambda}) + \sum_{j=1}^{m} c_{ij}(\boldsymbol{\lambda}) x_i^{d_{ij}} = 0 \tag{6.39}$$

for $i = 1, 2, \cdots, n$, which are best solved numerically for $x_i(\boldsymbol{\lambda})$. However, (6.39) is easily solved analytically if we require that

$$a_{ij}^{\{k\}} = a_i^{\{k\}} \quad \forall \quad j \, ,$$

in which case (6.38) can once again be thought of as arising from the application of (6.23) or (6.24). By a similar argument as for the reciprocal intermediate variables, the primal subproblem is convex with all $a_{ij}^{\{k\}} < 0$. (In a practical computer implementation, we set $-3 \leq a_{ij}^{\{k\}} \leq \epsilon_e < 0$, where -3 is selected rather arbitrarily; it merely serves to prevent very high negative exponents, while most exponents are expected to be in the vicinity of -1.)

Allowing concave approximations in the weight minimisation subproblems

We here explicate the approach to the weight minimisation problem described in Section 6.3.1. The direct approximation (6.12) is used to first order for the objective function, and (6.13) is used

to first order for the constraint approximations, irrespective of the sign of their partial gradients. We have already remarked in Section 6.5 that this approach is consistent with the application of the 'partial' method of mixed variables (6.24). The dual approximate subproblem now has the form

$$\begin{aligned}\gamma(\boldsymbol{\lambda}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \frac{\partial f_0^{\{k\}}}{\partial x_i}(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}) \\ + \sum_{j=1}^{m} \lambda_j \left(f_j(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right) \left(\frac{x_i^{\{k\}}}{x_i(\boldsymbol{\lambda})}\right) \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} \right).\end{aligned} \tag{6.40}$$

Applying (6.34), and defining

$$\beta_i(\boldsymbol{\lambda}) = -\left(\frac{\partial f_0^{\{k\}}}{\partial x_i}\right)^{-1} \left(\sum_{j=1}^{m} \lambda_j (x_i^{\{k\}})^2 \frac{\partial f_j^{\{k\}}}{\partial x_i}\right), \tag{6.41}$$

we obtain

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i^{1/2}(\boldsymbol{\lambda}) & \text{if} \quad \check{x}_i^2 < \beta_i(\boldsymbol{\lambda}) < \hat{x}_i^2\,, \\ \check{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \leq \check{x}_i^2\,, \\ \hat{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \geq \hat{x}_i^2\,, \end{cases} \tag{6.42}$$

for $i = 1, 2, \cdots, n$ in (6.29). Once again,

$$\frac{\partial f_0^{\{k\}}}{\partial x_i} > 0 \quad \forall \quad i\,,$$

so the denominator in (6.41) always exists. However, the numerator in (6.41) may be positive or negative and the turning point of $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ is given by the square root of $\beta_i(\boldsymbol{\lambda})$. If the numerator is negative, then there is a positive real solution for $x_i^* = \beta_i^{1/2}$. In this case, $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ is composed of a reciprocal (always convex and decreasing) part and a linearly increasing part. $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ cannot continually decrease because of the presence of the linear term, so a turning point must exist at finite x_i. It might be that the turning point lies outside the defined bounds, but this eventuality is catered for in (6.41) when deriving the minimum.

If the numerator is positive, however, there is no real solution for $\beta_i^{1/2}$, which would appear to imply that $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ has no turning point on $x_i > 0$. This is absolutely correct and stems from the inclusion of concave (general) reciprocal functions. In this context, Fleury [28] and Fleury and du Veubeke [72] have demonstrated that the weight minimisation problem with sizing design variables is strictly convex when the primal approximate subproblem is *recast* in terms of the reciprocals of the design variables, which would relieve the problem. Although this is true, such a recasting or transformation is unnecessary, as is an appeal to the recasting argument (as we have already remarked upon in Section 6.3.1). This is because, although $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ does not have a turning point, it still has a unique minimum between the imposed bounds, which means that the dual is always properly defined (refer to Sections 2.3.1 and 2.3.2).

When the numerator in (6.41) is positive, $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ is composed of a concave increasing part and a linear increasing part. It is, therefore, monotonically increasing over all $x_i > 0$. Hence, there is

no turning point. But there is a finite minimum, which obviously occurs at the lower bound on x_i. This minimum cannot be located by the condition

$$\frac{\partial}{\partial x_i}\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda}) = 0\,,$$

which is invalid in this case, but the mere fact that there is no real solution for the resulting expression $\beta_i^{1/2}$ is sufficient to indicate that the minimum is on the lower bound, given the structure of the Lagrangian. In this case, $\beta_i(\boldsymbol{\lambda}) < 0$, and so (6.42) is still valid for determining the minimum with respect to x_i.

There are no problems with (6.42) in terms of the existence of the solutions $x_i\,(\boldsymbol{\lambda})$. The squared form of the conditional parts that derive from the dual can always be evaluated. There is no reason to resort to the conservative convex approximation proposed by Starnes and Haftka [73], since imaginary numbers cannot result. Equation (6.42) is always logically consistent with the structure of the Lagrangian and correctly yields the minimum. Now, if one chooses to use a two-point exponential approximation with negative exponents (still of order 1) to approximate the constraints in this case, one obtains

$$\begin{aligned}\gamma(\boldsymbol{\lambda}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\frac{\partial f_0^{\{k\}}}{\partial x_i}(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}) \\ + \sum_{j=1}^{m}\lambda_j\left(f_j(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left[\left(\frac{x_i}{x_i^{\{k\}}}\right)^{a_{ij}^{\{k\}}} - 1\right]\left(\frac{x_i^{\{k\}}}{a_{ij}^{\{k\}}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right).\end{aligned} \tag{6.43}$$

This structure does *not* in general possess unique minima with respect to x_i for any given $\boldsymbol{\lambda}$. In fact, applying (6.34) and solving the resulting equations may even yield maxima for $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ instead of minima. Hence, $x_i\,(\boldsymbol{\lambda})$ may be either non-unique or flatly wrong, and the dual $\gamma(\boldsymbol{\lambda})$ is thus improperly defined.

As we noted in Section 6.4.2, if the partial derivatives $\partial f_j/\partial x_i$ are allowed to take on any sign, the exponents a_{ij} cannot in general be calculated independently. However, with the stipulation that $a_{ij} = a_i\ \forall\ j$, we may apply (6.19) to first order for the constraint approximation, and the resulting dual function

$$\begin{aligned}\gamma(\boldsymbol{\lambda}) = f_0(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\frac{\partial f_0^{\{k\}}}{\partial x_i}(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}) \\ + \sum_{j=1}^{m}\lambda_j\left(f_j(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n}\left[\left(\frac{x_i}{x_i^{\{k\}}}\right)^{a_i^{\{k\}}} - 1\right]\left(\frac{x_i^{\{k\}}}{a_i^{\{k\}}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right)\end{aligned} \tag{6.44}$$

is uniquely defined and is again consistent with the partial method of mixed variables (6.24). The primal-dual relationships are now given by

$$\beta_i(\boldsymbol{\lambda}) = -\left(\frac{\partial f_0^{\{k\}}}{\partial x_i}\right)^{-1}\left(\sum_{j=1}^{m}\lambda_j(x_i^{\{k\}})^{r_i}\frac{\partial f_j^{\{k\}}}{\partial x_i}\right) \tag{6.45}$$

and

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i^{1/r_i}(\boldsymbol{\lambda}) & \text{if} \quad \check{x}_i^{r_i} < \beta_i(\boldsymbol{\lambda}) < \hat{x}_i^{r_i}, \\ \check{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \leq \check{x}_i^{r_i}, \\ \hat{x}_i & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \geq \hat{x}_i^{r_i}, \end{cases} \tag{6.46}$$

in which $r_i = 1 - a_i^{\{k\}}$. This is simply a case in which fixed negative values other than $a_i = -1$ are used in defining (6.40). In compliance optimisation, compliant mechanism design is a very well-known example of such a strategy.

6.6.2 A general routine for the solution of $\mathrm{P_{NLP}}$

The two examples presented above are both first-order approximation strategies that have been applied in the past to the weight minimisation problem. The subproblems created thereby have unique minima and, moreover, can be solved using Falk's dual method. However, the approximations used in the examples are just special cases of the application of one or other of the methods described in Section 6.5, of which three allow for the use of higher-order terms.

We will not explicitly present an example involving the use of these terms; instead, we here run through the method involved in defining and solving the approximate subproblem, with or without higher-order terms.

Step 1: Choose an approximation (which may represent a method of mixed variables) relevant to the given problem $\mathrm{P_{NLP}}$.

Step2: Define the primal approximate subproblem at $\boldsymbol{x}^{\{k\}}$; apply the approximation consistent with the chosen method (or various special cases thereof) to all the functions in $\mathrm{P_{NLP}}$.

Step 3: Define the Lagrangian. Actually, it is only necessary to note the general form of $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$ explicitly, since $\mathcal{L}^{\{k\}}$ is separable and all $\mathcal{L}_i^{\{k\}}$ have the same general structure.

Step 4: Using $\mathcal{L}_i^{\{k\}}(x_i, \boldsymbol{\lambda})$, derive the primal-dual relationship (6.29). With this, the dual approximate subproblem is effectively defined.

Step 5: Maximise the dual (see Section 6.6.3).

The main difficulty lies in Step 4, which requires some elaboration. In the examples presented in Section 6.6.1, the primal-dual relationship was defined by applying the stationary conditions (6.34) to $\mathcal{L}_i^{\{k\}}$ and solving the resulting equation, which yielded (6.35) or (6.41). We also noted that the conditions (6.34) are not always valid. They are only valid if $\mathcal{L}_i^{\{k\}}$ possesses a turning point somewhere on the positive real line, but, when noting the special structure of $\mathcal{L}_i^{\{k\}}$, criteria of the form (6.42) were obtained.

This will be the case generally. We have indicated in Section 6.5 that $\mathcal{L}_i^{\{k\}}$ will always have a unique minimum on any given interval $\check{x}_i \leq x_i \leq \hat{x}_i$. If $\mathcal{L}_i^{\{k\}}$ has no turning point within these bounds, then it is monotonic over the interval. Hence, it is a good idea to check the sign of $\partial \mathcal{L}_i^{\{k\}} / \partial x_i$ at $\check{x}_i$ and $\hat{x}_i$. If

$$\left. \frac{\partial \mathcal{L}_i^{\{k\}}}{\partial x_i} \right|_{\check{x}_i} \geq 0 \,,$$

then $x_i(\boldsymbol{\lambda}) = \check{x}_i$. Alternatively, if

$$\left.\frac{\partial \mathcal{L}_i^{\{k\}}}{\partial x_i}\right|_{\hat{x}_i} \leq 0\,,$$

then $x_i(\boldsymbol{\lambda}) = \hat{x}_i$. If neither of these holds we would expect the minimum to be a turning point in $\check{x}_i \leq x_i \leq \hat{x}_i$, which would be located by applying the stationary conditions. When deriving and evaluating the gradients $\partial \mathcal{L}_i^{\{k\}}/\partial x_i$, note that the absolute value operators in the higher-order terms of the approximations have to be taken into account. These result in the unity-valued sign operators $s(x_i)$, as in (6.17).

Lastly, the application of the stationary conditions is likely to yield a function that is not reducible to an analytical expressions for $x_i(\boldsymbol{\lambda})$, especially if the higher-order terms are retained. However, these functions must have unique solutions and are always one-dimensional (since we have demanded primal separability), but the solutions will generally have to be found numerically.

6.6.3 Solving the dual approximate subproblem

Dual approximate subproblem (6.32) may be solved efficiently using first-order or second-order methods. First-order gradient-based methods are very simple; steepest descent, or preferably conjugate gradient solvers in the Fletcher-Reeves tradition, are suitable and obvious possibilities. Second-order Newton methods may be very efficient, see for example Huang and Arora [28]. For the 'standard' minimum compliance topology optimisation problem, being expressed in terms of a single linear (volume) constraint, one may even use an efficient linesearch method to solve dual approximate subproblem (6.32), if so desired. The solvers used need only be modified to take the simple non-negativity conditions on the Lagrangian multipliers $\lambda_j \geq 0$, $j = 1, 2, \cdots, m$ into account.

Our current implementation uses a limited memory BFGS variable metric solver [74, 75], which is able to take the simple non-negativity constraints into consideration. For the limited memory BFGS solver, we only require the gradients of $\gamma(\boldsymbol{\lambda})$ with respect to the λ_j. These are obtained as

$$\frac{\partial \gamma(\boldsymbol{\lambda})}{\partial \lambda_j} = \gamma'(\boldsymbol{\lambda}) = \tilde{f}_j(\boldsymbol{x}(\boldsymbol{\lambda})) \qquad j = 1, 2, \cdots, m. \tag{6.47}$$

Note that the $\tilde{f}_j(\boldsymbol{x}(\boldsymbol{\lambda}))$ in (6.47) would already be calculated anyway upon evaluating the dual. It should be noted that discontinuity planes exist in the second derivatives of $\gamma(\boldsymbol{\lambda})$ [2, 28]. These discontinuity planes arise from the modified definition domains of dual approximate subproblem (6.32), due to the bounds $\check{x}_i$ and $\hat{x}_i$ in (6.31):

$$x_i(\boldsymbol{\lambda}) \in (\check{x}_i, \hat{x}_i) \qquad i = 1, 2, \cdots, n.$$

When any of the potential $2n$ discontinuity planes is crossed, the distribution of free and fixed variables in the primal problem may be modified (free variables being those with inactive bounds). In turn, this may result in the appearance of angular points in $\gamma'(\boldsymbol{\lambda})$, and discontinuities in $\gamma''(\boldsymbol{\lambda})$, being the second derivatives of $\gamma(\boldsymbol{\lambda})$. This may also have implications for any linesearch procedure used in the solvers.

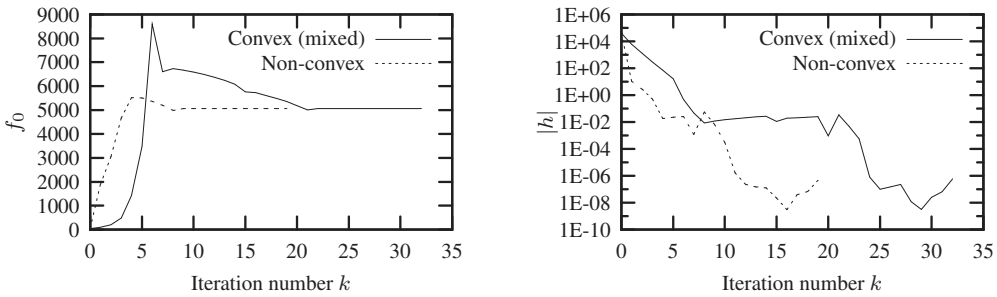
For reasons that we do not fully understand, the limited memory BFGS solver we have used [74, 75] seems to have few, if any, problems with the second-order discontinuities present in the dual. What is more, second-order methods are often used in algorithms based on the Falk dual. An important example is the well-known MMA algorithm of Svanberg. However, the dual may of course be solved perfectly well using first-order methods, e.g. conjugate gradient methods; to machine precision, this results in exactly the same primal iteration path. We have opted for the second-order method simply because there seems to be a computational advantage on the subproblem level (in terms of the required effort).

For the weight minimisation problem with sizing design variables, illustrative examples of the discontinuity planes and definition domains may be found in the paper of Fleury [28].

6.7 A numerical example

As an example of the potential implications of some of our aforementioned developments, we consider the very well-known nonconvex 10-bar truss problem with displacement constraints, previously studied by so many authors. For the sake of brevity we do not reiterate the problem formulation here. Instead, the reader is referred to Haftka and Gürdal [27], Section 6.7, Case B. The optimal solution is $f_0^* = 5060.854$ (the units used in the example are imperial). The only change we make to the data presented by Haftka and Gürdal is that we depart with all the variables on the lower bound $\check{x}_i = 0.1$, since this amplifies the phenomena we wish to illustrate.

Numerical results are presented in Figure 6.4, which depicts the objective function value f_0 versus the iteration number k, as well as the largest constraint value, defined as $h = \max(f_j)$, $j = 1, 2, , \cdots, m$. In the figure, 'nonconvex' implies direct application of (6.1), whereas 'convex' implies CONLIN, or equivalently (6.25) with all $q_{i\alpha} = 1$ and all $a_{i\alpha} = -1$. Clearly, the convex method of mixed variables impairs convergence. (This is not necessarily always the case. It is possible to generate results for which the convex method of mixed variables actually yields faster convergence, but our experiments suggest that this is marginal.)



(a) Objective function value (b) Largest constraint value

Figure 6.4: The effect of enforcing convexity for the nonconvex 10-bar truss problem with displacement constraints.

6.8 Conclusions

We have discussed the use of inverse (negative exponential) approximations in a dual approach to sequential approximate optimisation. The approximations derive from a separable series expansion in terms of exponential intervening variables that contains higher-order 'main' or diagonal terms, but omits terms associated with mixed partial derivatives.

Since the exponential expansion is generally nonconvex, we have discussed under what conditions nonconvex approximations can be used along with the dual method of solution introduced by Falk. These conditions, together with an analysis of the functional forms that derive from the exponential expansion, suggest four general approximation strategies that can accommodate negative exponentials. Three of these represent methods of mixed variables, and three retain the higher-order terms, and are thus generally nonconvex. Despite this, we have shown that the resulting subproblems are amenable to solution via the Falk dual without necessitating a convex transformation. That is: despite being nonconvex, the subproblems have unique solutions provided that they are primal feasible. As such, we have not tried to investigate the *conditions* that ensure that $\mathcal{L}_i$ has a unique minimum. Such conditions, if they exist, are likely to be quite involved. For example, the easily assessed stipulations that $\mathcal{L}_i$ must be globally monotonic or globally convex are often too strict. Many of the statements we have studied are neither, but nevertheless admit unique minima.

We have used the weight minimisation problem as an example and have reintroduced a nonconvex approach due to Fleury (and predating CONLIN) that is consistent with one of the methods suggested.

We conclude that it is indeed possible to use higher-order nonconvex exponential approximations for SAO and to retain a unique KKT point for the subproblems, provided that the various parameters that define the approximations are chosen or limited judiciously. This should be of interest to code developers. However, whether or not the nonconvex higher-order approximations can be used to improve algorithmic performance for a given problem is a matter that must be evaluated numerically, and which we hope to pursue in the future.

Chapter 7

Convex transformability and the Falk dual

The exposition below originates from a paper titled "On a link between convex transformability and the solution of nonconvex problems via the dual of Falk" [76]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa.

7.1 Abstract

In structural optimisation, most successful sequential approximate optimisation (SAO) algorithms solve a sequence of *strictly convex* subproblems using the dual of Falk. Previously, we have shown that, under certain conditions, a *nonconvex* nonlinear (sub)problem may also be solved using the Falk dual. In particular, we have demonstrated this for two nonconvex examples of approximate subproblems that arise in important structural optimisation problems. The first is used in the SAO solution of the weight minimisation problem, while the minimum compliance problem that results from volumetric penalisation gives rise to the other. In both cases, the nonconvex subproblems arise naturally in the consideration of the physical problems, so it seems counterproductive to discard them in favour of using standard, but less well-suited, strictly convex approximations. Although we have not required that strictly convex transformations exist for these subproblems in order that they may be solved via a dual approach, we note that both of these examples can indeed be transformed into strictly convex forms. In this chapter we explore the link between convex transformability and the salient criteria that make nonconvex problems amenable to solution via the Falk dual, and we assess the effect of the transformation on the dual problem. However, we consider only a restricted class of problems, namely separable problems that are at least C^1 continuous, and a restricted class of transformations: those in which the functions that represent the mapping are each continuous, monotonic and univariate.

7.2 Introduction

Today, sequential approximate optimisation (SAO) is recognised as an efficient technique for the solution of nonlinear structural optimisation problems. In the development of SAO methods, it has

become almost standard practice to utilise strictly convex function approximations to define the approximate subproblems, which are then optimised as surrogates for the physical problem, almost invariably using a dual approach. This is typically true even if the physical problem is known to be locally nonconvex. The reason for using convex approximations stems largely from the fact that the minima of feasible strictly convex programs are bound to be unique (a fact that leads to necessary and sufficient conditions on global optima, and facilitates the analysis of algorithmic convergence).

Many well-known SAO methods used for structural optimisation solve their strictly convex subproblems using a dual method due originally to Falk [2]. This is done because significant gains can be achieved in algorithm efficiency when the dual method is implemented. The dual problem has a simple structure: it is concave and the only constraints present are non-negativity constraints on the dual variables. The dual of Falk does not require dual variables for the primal bounds (albeit at the cost of introducing discontinuities in the second derivatives of the dual function [28]). Also, it is often the case that the dimensionality of the dual is less than that of the primal, since the number of (active) primal constraints is often (far) less than the number of primal variables. Examples of popular SAO algorithms that use the dual of Falk are the method of moving asymptotes (MMA) of Svanberg [3], and the convex linearisation algorithm (CONLIN) of Fleury and Braibant [4]. Hence, the use of the Falk dual to solve strictly convex subproblems has become fairly standard.

However, we have previously indicated that it is also possible to use nonconvex approximate subproblems in combination with the Falk dual [35, 36]. This is not unprecedented: Fleury already presented an example of this in 1979 [28] in his study of the nonconvex weight minimisation problem. However, this idea appears to be applied rarely. In Reference [28], Fleury justified the use of a nonconvex subproblem by arguing that the subproblem could be transformed into a strictly convex form, which has a unique KKT point. In Chapter 6 (and [36]), in which we address the same problem, we argue instead that the theorems given by Falk, which prove that his dual approach can be used to solve strictly convex programming problems, utilise certain attributes of strictly convex problems and that these attributes are also exhibited by Fleury's nonconvex subproblem, and others. Therefore, we maintain that any nonconvex programming problems that possess these attributes can also be uniquely solved using Falk's dual approach.

This does not invalidate the transformation rationale, but it must be recognised that the transformation argument needs to be qualified. In other words, given a particular nonconvex problem, the rationale is only true for bijective transformations, which are themselves only a subset of all possible transformations that will yield convex problems. Transformations are 'valid' if they yield a one-to-one correspondence between the transformed and untransformed problems.

Nevertheless, the transformation originally applied by Fleury in [28] was of course bijective, and bijective transformations also exist for the problem discussed in Chapter 5 (based on [35]), which addresses the nonconvex minimum compliance topology optimisation problem that results from volumetric penalisation (aimed at generating predominantly solid-void designs). It would appear, then, that a link may exist between the ability to find a strictly convex transformation for a nonconvex problem, and the ability to solve the problem directly using the Falk dual. Here, we investigate this link for the continuous, separable subproblems that are prevalent in structural optimisation, but we limit our investigation to cases in which the 'bijective' transformations are defined by univariate functions that are at least C^1 continuous.

The arguments presented in this chapter follow from Falk's definition of the dual of a nonlinear programming problem introduced in Section 2.3.1, as well as our assertion, put forward in Section 2.3.2, that the proof that Falk presented for strictly convex problems also holds for certain nonconvex forms. First, a summary of the assumpions that are used in the subsequent analysis is given. Said assumptions only serve to reiterate the particular form of the subproblems considered here. In Section 7.4 we go on to investigate how the possibility of finding a strictly convex transform for such a subproblem relates to the possibility of solving it directly using the Falk dual.

7.3 Summary of assumptions

In light of the discussion presented thus far, it should be noted that the remainder of this chapter deals specifically with the following general form for a programming problem, consistent with (2.39), which is assumed to define an *approximate subproblem* in an SAO algorithm:

$$\begin{aligned} &\min_{\boldsymbol{x}} \ \tilde{f}_0(\boldsymbol{x}) \\ &\text{subject to} \ \ \tilde{f}_j(\boldsymbol{x}) \geq 0 \qquad j = 1, 2, \cdots, m, \\ &\qquad\qquad \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{7.1}$$

The $\check{x}_i$ and $\hat{x}_i$ denote the lower and upper bounds respectively on the variable x_i, and the tildes on $\tilde{f}_0$ and $\tilde{f}_j$ denote that they are approximation functions defined at a particular point $\boldsymbol{x}^{\{k\}}$ in the design space. They are constructed to represent the real objective function f_0 and the m constraint functions f_j around $\boldsymbol{x}^{\{k\}}$.

Typically, $\tilde{f}_0$ and all $\tilde{f}_j$ are continuous and separable functions chosen so that the approximate subproblem is strictly convex. This is the case in the popular SAO algorithms MMA and CONLIN, for example. As such, a strictly convex approximation $\tilde{f}_0$ is chosen for the objective, whereas concave approximations $\tilde{f}_j$ are selected for the constraints[1]. The set $\mathcal{C}$ in (2.39) consists here of only the upper and lower bounds on x_i.

We herein consider problems of the form (7.1), in which $\tilde{f}_0$ and $\tilde{f}_j$ are all separable functions that are at least C^1 continuous. However, we do not enforce the typical convexity assumptions on $\tilde{f}_0$ and $\tilde{f}_j$. Instead, we assume that they can all be chosen as separable nonconvex functions, but in such a way that the resulting programming problem (7.1) possesses the three attributes discussed in Section 2.3.2. We have argued that continuous problems for which Attributes 1 through 3 hold can be solved by a dual method utilising the Falk dual, and that the proof of this is Falk's proof for convex problems.

Nonconvex examples to which the above applies are the subject of Chapters 5 and 6. In Chapter 5, the following form of subproblem was discussed, which serves to approximate the minimum compliance problem with volumetric penalisation:

[1]Please note that we here use the positive-null representation of an inequality constrained programming problem, and its associated Lagrangian, to be consistent with Falk [2].

Primal approximate subproblem $P_T^{\{k\}}$

$$\begin{aligned}
&\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i^{r_i} \\
&\text{subject to } f_j(\boldsymbol{x}) = c_{0j} + \sum_{i=1}^{n} c_{ij} x_i^{q_{ij}} \geq 0 \qquad j = 1, 2, \cdots, m, \\
&\qquad 0 < \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n, \\
&\qquad a_i > 0 \qquad i = 1, 2, \cdots, n, \\
&\qquad \alpha \leq r_i < 0 \qquad i = 1, 2, \cdots, n, \\
&\qquad c_{ij} < 0 \qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m, \\
&\qquad 0 < q_{ij} \leq 1 \qquad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m.
\end{aligned} \tag{7.2}$$

The form of subproblem discussed in Chapter 6 that can be used in the weight minimisation problem is given as:
Primal approximate subproblem $P_W^{\{k\}}$

$$\begin{aligned}
&\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i \\
&\text{subject to } f_j(\boldsymbol{x}) = c_{0j} + \sum_{i=1}^{n} \frac{c_{ij}}{x_i} \geq 0 \qquad j = 1, 2, \cdots, m, \\
&\qquad 0 < \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n, \\
&\qquad a_i > 0 \qquad i = 1, 2, \cdots, n.
\end{aligned} \tag{7.3}$$

7.4 Attribute 1 and convex transformability

Since Attribute 1 (see Section 2.3.2) holds for both subproblems (7.2) and (7.3), they can be solved without reference or recourse to any transforms that may make the subproblems convex. However, it is also true that both $P_T^{\{k\}}$ and $P_W^{\{k\}}$ can be transformed into strictly convex problems via separable *one-to-one* transformations. $P_T^{\{k\}}$ becomes strictly convex under the application of

$$x_i' = x_i^{p_i}, \tag{7.4}$$

where

$$p_i = \min_j q_{ij} \quad j = 1, 2, \cdots, m,$$

while $P_W^{\{k\}}$ can be transformed by

$$x_i' = \frac{1}{x_i}, \tag{7.5}$$

as discussed in [28], though it should be noted that the range of validity of the transformations is the positive orthant $\boldsymbol{x} > \mathbf{0}$. The above raises the question of whether or not a link exists between

Attribute 1 and the existence of a bijective convex transformation. We examine this question here, but we consider only the restricted case of univariate transformations, where each coordinate in the transformed space can be written as a function of a single coordinate in the untransformed space. With this restriction, separability is preserved under transformation. If this is not the case, the mapping is likely to be very difficult to define in practice, and the transformed problem may not be as easily solved via the dual approach.

To investigate the connection between Attribute 1 and convex transformability, we start with a strictly convex problem. It is well established that the optimum of a strictly convex problem satisfies the necessary and sufficient KKT conditions for a saddle point on the Lagrangian. We then apply a bijective transformation to the problem and examine what conditions the transform of the optimum satisfies.

First, however, it will be necessary to clarify what the KKT conditions look like for a Lagrangian consistent with the definition of the Falk dual, because the bound constraints (the set $\mathcal{C}$) are not used in the derivation of the Lagrangian.

7.4.1 A note on the KKT conditions

In this section we present the necessary KKT conditions for the bound constrained problem defined in (7.1), bearing in mind that, in keeping with Falk, the bound constraints on the variables x_i are not explicitly included as constraints in the definition of the Lagrangian in (2.40). We do so by first presenting Hadley's treatment of the KKT conditions for a problem with non-negativity constraints on the primal variables [22]. These constraints similarly are not included in the definition of the Lagrangian of the problem. We then simply extend Hadley's result to account for bound constraints, rather than non-negativity constraints. For the sake of completeness, we also point out a possible degeneracy that may arise in the definition of the saddle point due to the existence of the bounds on x_i. We illustrate this by using the non-negativity constraints in Hadley's problem.

In his treatment of the KKT conditions in [22], Hadley derives the necessary conditions for a saddle point on the Lagrangian of a general nonlinear programming problem subject to non-negativity constraints on its primal variables, i.e.:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \tilde{f}_0(\boldsymbol{x}) \\ \text{subject to } & \tilde{f}_j(\boldsymbol{x}) \geq 0 \qquad j = 1, 2, \cdots, m, \\ & x_i \geq 0 \qquad\quad i = 1, 2, \cdots, n. \end{aligned} \tag{7.6}$$

In the analysis it is assumed that the objective and constraint functions are all at least continuous to first order. Similarly to Falk, Hadley uses f_0 and the f_j, but not the lower bounds on the x_i, to define the Lagrangian of the problem. According to Hadley, for a Lagrangian so defined, a saddle point is a point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ that satisfies the following conditions:

$$\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}) \leq \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \leq \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*), \qquad x_i^* \geq 0 \quad \forall \quad i, \qquad \lambda_j^* \geq 0 \quad \forall \quad j. \tag{7.7}$$

Such a point necessarily satisfies the conditions

$$\frac{\partial}{\partial x_i}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) \geq 0 \quad \forall \quad i \,:\, x_i^* \geq 0\,, \tag{7.8}$$

$$x_i^* \frac{\partial}{\partial x_i}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) = 0 \quad \forall \quad i\,, \tag{7.9}$$

$$\frac{\partial}{\partial \lambda_j}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) \leq 0 \quad \forall \quad j \,:\, \lambda_j^* \geq 0\,, \tag{7.10}$$

$$\lambda_j^* \frac{\partial}{\partial \lambda_j}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) = 0 \quad \forall \quad j\,, \tag{7.11}$$

and of course all $\lambda_j^* \geq 0$. The inequality in condition (7.8) occurs because the non-negativity constraints are not taken into account explicitly to form the Lagrangian. Condition (7.9) exists for the same reason. If these additional constraints are included explicitly, then the strict equality would hold in (7.8) and (7.9) would be absent entirely, but the dimensionality of the Lagrangian, as well as of the dual, would increase. The non-negativity constraints on x_i are therefore catered for not in the definition of the Lagrangian, but in the definition of the saddle point. Saddle points denote local extrema, and Hadley shows that the above conditions are sufficient to define the global minimiser in the case of strictly convex problems[2] (actually [22] discusses the maximisation of strictly concave problems).

For our purposes, we use [22] to state the necessary conditions that the optimum must satisfy for separable bound-constrained problems of the form given in (7.1). Since (7.1) describes a problem that satisfies Attribute 1, the saddle point of its Lagrangian is unique and corresponds to the optimum of the primal problem (analogously to the strictly convex case). By a simple extension of Hadley's arguments, the optimum must satisfy (7.10) and (7.11), as well as

$$\frac{\partial}{\partial x_i}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) \geq 0 \quad \forall \quad x_i^* = \check{x}_i\,, \tag{7.12}$$

$$\frac{\partial}{\partial x_i}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) \leq 0 \quad \forall \quad x_i^* = \hat{x}_i\,, \tag{7.13}$$

$$\left(x_i^* - \check{x}_i\right)\left(\hat{x}_i - x_i^*\right)\frac{\partial}{\partial x_i}\mathcal{L}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}^*\right) = 0 \quad \forall \quad i\,. \tag{7.14}$$

Equation (7.9) is no longer a valid condition, because the problem is no longer one whose bound constraints are only non-negativity constraints. It has been replaced by the condition (7.14). Now, let us examine whether or not a saddle point defined in this way is unique. Consider the following one-dimensional strictly convex example:

$$\begin{aligned} \min_x \; & f_0(x) = x^2 - 1 \\ \text{subject to } & f_1(x) = x - 1 \geq 0\,, \\ & x \geq 0\,. \end{aligned} \tag{7.15}$$

A contour plot of the associated Lagrangian is shown in Figure 7.1. The Lagrangian is strictly

[2]Provided, of course, that a linear independence constraint qualification is also satisfied at the optimum.

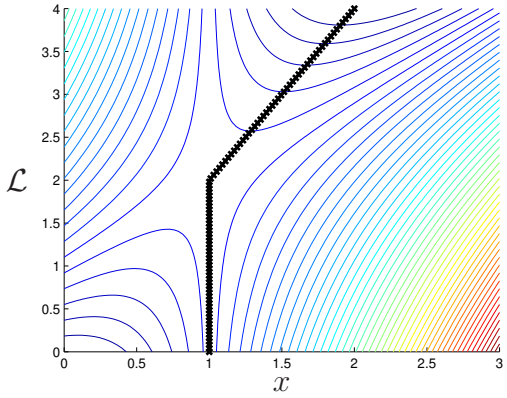


Figure 7.1: Contour plot of the Lagrangian for the one-dimensional convex problem (7.15).

convex in x, and therefore has a unique minimum with respect to x for any value of λ. The non-negativity constraint on x defines the set $\mathcal{C}$ in Falk's treatment of the dual. If this non-negativity constraint were absent, the necessary conditions for a saddle point on the Lagrangian would be satisfied at a unique point, namely $(x, \lambda) = (1, 2)$.

The bounded minimum of the Lagrangian with respect to x is depicted by the broad line in Figure 7.1. This represents the Falk dual (2.41) of problem (7.15). When the bound constraint on x is respected, the same primal coordinate $x = 1$ minimises the Lagrangian at all dual coordinates on the subspace $0 \leq \lambda \leq 2$. Hence, all points $(x = 1,\ 0 \leq \lambda \leq 2)$ satisfy conditions (7.8) through (7.11), since $\partial\mathcal{L}/\partial\lambda = 0$ on this subspace. For this particular example, then, the Falk dual is constant on $\lambda \leq 2$ (it is strictly concave and decreasing on $\lambda > 2$). The saddle is therefore a degenerate form of saddle in this case, but all points that satisfy the necessary conditions map to the same primal point x, so the conditions are still sufficient.

Falk proves that the dual is *concave*, but as the preceding example illustrates, it is not necessarily *strictly* concave. In general, though, if the problem is strictly convex, or if it fulfils Attributes 1 through 3, all points in the space for which $\partial\mathcal{L}/\partial\lambda_j = 0\ \forall\ j$ will map to the same primal point, so the primal optimum will be referenced uniquely by conditions (7.8) through (7.11), even though the dual maximum may be non-unique.

The purpose of the above discussion is to introduce the definition of the KKT point for the type of problem considered herein, namely (7.1), and to point out that such a problem has a unique optimum, due to Attribute 1. We will use the necessary conditions that define the KKT point in Section 7.4.2 to examine the effects on the dual of imposing univariate convex transformations on separable nonconvex problems. Secondly, we remark that the Falk dual, while being concave, is not necessarily strictly concave for strictly convex problems, or for those problems that satisfy Attributes 1 through 3, because degenerate saddle points may exist along subspaces in the Lagrangian due to the bound constraints on the primal variables.

7.4.2 Convex transformability: Implications for the Falk dual

In this section we wish to examine the relationship between a nonconvex problem of the form given in (7.1), which satisfies Attributes 1 through 3, and its strictly convex transform, which would also be a problem of the same form (7.1). Hence, we assume that our nonconvex problem does possess such a strictly convex transform, although we have not proved this to be true generally. Additionally, as (7.1) is separable, we here consider only the univariate transforms discussed below. We begin by discussing the transformed problem: consider a separable bound-constrained problem of the form given in (7.1). Assume that $f_0(\boldsymbol{x})$ is strictly convex and that all $f_j(\boldsymbol{x})$ are concave over the *compact* set $\mathcal{C}$ defined by the bound constraints. Under these assumptions, (7.1) represents a strictly convex problem, which for convenience we label P_{SC}. The Lagrangian for P_{SC} is

$$\mathcal{L}^{sc}(\boldsymbol{x}, \boldsymbol{\lambda}) = f_0(\boldsymbol{x}) - \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}). \tag{7.16}$$

Since P_{SC} is separable, (7.16) can be written as the sum of n terms as in (2.42), with the i^{th} term given by

$$\mathcal{L}_i^{sc}(x_i, \boldsymbol{\lambda}) = f_{i0}(x_i) - \sum_{j=1}^{m} \lambda_j f_{ij}(x_i), \tag{7.17}$$

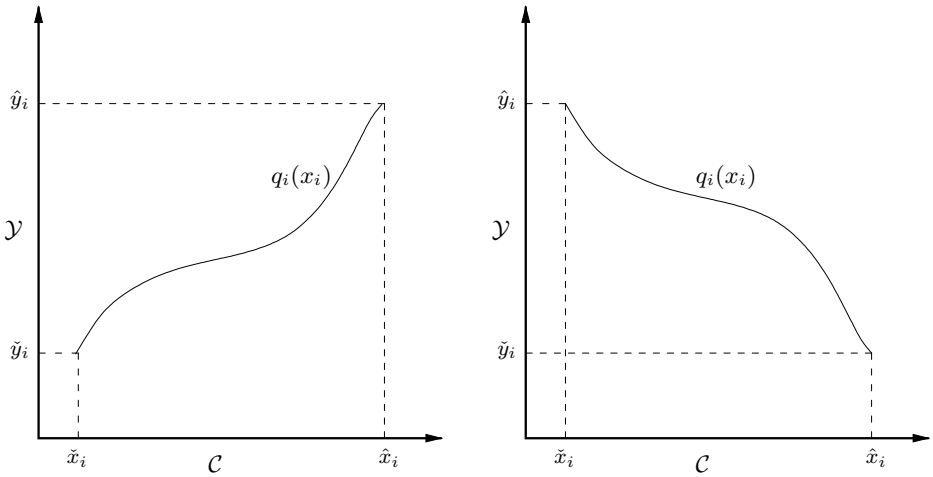
which is itself a strictly convex function for all i. Assuming that a feasible solution exists, the conditions given for a saddle point on the Lagrangian (namely (7.10) through (7.14)) are uniquely satisfied by the optimum of P_{SC}. If we consider only the univariate transformations alluded to above, then under such a transformation, each $\mathcal{L}_i$ can be written as a composite function in terms of an intermediate variable y_i

$$\begin{aligned}\mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda}) =& \mathcal{L}_i^{sc}\left(q_i^{-1}(y_i), \boldsymbol{\lambda}\right) \\ =& f_{i0}\left(q_i^{-1}(y_i)\right) - \sum_{j=1}^{m} \lambda_j \left[f_{ij}\left(q_i^{-1}(y_i)\right)\right],\end{aligned} \tag{7.18}$$

which yields the i^{th} component of the Lagrangian of our associated nonconvex problem, which we label P_{NC}. We have taken $\mathcal{L}_i^{sc}(x_i, \boldsymbol{\lambda})$ and expressed it as a function of y_i, using

$$\begin{aligned}y_i =& q_i(x_i), \\ x_i =& q_i^{-1}(y_i).\end{aligned} \tag{7.19}$$

Here, q_i^{-1} denotes an inverse (reverse) mapping, not the operation $1/q_i$. The $q_i,\ i = 1, 2, \cdots, n$ are functions that together define a mapping, or transformation, from the set $\mathcal{C} \subset \mathcal{R}^n$ to a set $\mathcal{Y} \subset \mathcal{R}^n$. We require that the functions q_i and q_i^{-1} be C^1 continuous and that the mapping corresponds to a bijection between the sets $\mathcal{C}$ and $\mathcal{Y}$. This guarantees that both q_i and q_i^{-1} are uniquely defined, so that $x_i^{\ddagger} = q_i^{-1}\left(q_i\left(x_i^{\ddagger}\right)\right)$ for any arbitrary $x_i = x_i^{\ddagger}$ in $\mathcal{C}$. This being the case, problem P_{NC} and problem P_{SC} are obviously identical, being only different representations of the same problem. We can equally write the original Lagrangian $\mathcal{L}_i^{sc}(x_i, \boldsymbol{\lambda})$ as a composite function by writing $\mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda})$



(a) $\frac{\partial q_i}{\partial x_i} > 0$ (b) $\frac{\partial q_i}{\partial x_i} < 0$

Figure 7.2: Invertable univariate transformation functions.

in terms of x_i:

$$\begin{aligned}\mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}\right) =& \mathcal{L}_i^{nc}\left(q_i\left(x_i\right), \boldsymbol{\lambda}\right) \\ =& f_{i0}\left(q_i\left(x_i\right)\right) - \sum_{j=1}^{m} \lambda_j \left[f_{ij}\left(q_i\left(x_i\right)\right)\right]. \end{aligned} \tag{7.20}$$

It is known, however, that the only one-dimensional functions that meet these requirements on q_i and q_i^{-1} are strictly monotone functions [77]. From this last we infer the following properties (refer to Figure 7.2): firstly, either

$$\frac{\partial q_i}{\partial x_i} \geq 0\,, \tag{7.21}$$

or

$$\frac{\partial q_i}{\partial x_i} \leq 0\,, \tag{7.22}$$

for all points $\check{x}_i \leq x_i \leq \hat{x}_i$. If the equalities hold, then they can only hold at a number of discrete (separated) points in the domain, and the inequalities will hold everywhere else.

The bounds on the set $\mathcal{C}$ transform to the bounds on the set $\mathcal{Y}$. In the case that (7.21) holds, the lower bound in x_i, namely $\check{x}_i$, becomes the lower bound in y_i, namely $\check{y}_i = q_i\left(\check{x}_i\right)$, and the upper bound $\hat{x}_i$ transforms to the upper bound $\hat{y}_i$. However, the reverse occurs if (7.22) holds. In this case, $\check{y}_i = q_i\left(\hat{x}_i\right)$ and $\hat{y}_i = q_i\left(\check{x}_i\right)$. Now, the optimum of P_{SC}, which we label $\boldsymbol{x}^*$, transforms uniquely to the point $\boldsymbol{y}^*$ under the above univariate transformation. Given (7.18), in $\mathcal{Y}$ we have

$$\begin{aligned}\frac{\partial \mathcal{L}_i^{nc}\left(y_i^*, \boldsymbol{\lambda}\right)}{\partial \lambda_j} =& - f_{ij}\left(q_i^{-1}\left(y_i^*\right)\right) \\ =& - f_{ij}\left(x_i^*\right) \\ =& \frac{\partial \mathcal{L}_i^{sc}\left(x_i^*, \boldsymbol{\lambda}\right)}{\partial \lambda_j} \end{aligned} \tag{7.23}$$

for all i, which implies that, if $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ satisfies the necessary conditions (7.10) and (7.11), where $\boldsymbol{\lambda}^*$ are the Lagrange multipliers $\boldsymbol{\lambda}$ associated with $\boldsymbol{x}^*$ at the saddle point of P_{SC}, then $(\boldsymbol{y}^*, \boldsymbol{\lambda}^*)$ satisfies

$$\frac{\partial}{\partial \lambda_j} \mathcal{L}^{nc}\left(\boldsymbol{y}^*, \boldsymbol{\lambda}^*\right) \leq 0 \quad \forall \quad j \; : \; \lambda_j^* \geq 0\,, \tag{7.24}$$

$$\lambda_j^* \frac{\partial}{\partial \lambda_j} \mathcal{L}^{nc}\left(\boldsymbol{y}^*, \boldsymbol{\lambda}^*\right) = 0 \quad \forall \quad j\,. \tag{7.25}$$

Using (7.20) and the chain rule we may also write

$$\begin{aligned} \frac{\partial \mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}\right)}{\partial x_i} =& \frac{\partial}{\partial x_i}\left[\mathcal{L}_i^{nc}\left(q_i\left(x_i\right), \boldsymbol{\lambda}\right)\right] \\ =& \frac{\partial \mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}\right)}{\partial y_i} \frac{\partial q_i}{\partial x_i}\,. \end{aligned} \tag{7.26}$$

We examine below the case in which neither $\partial q_i / \partial x_i$ nor $\partial q_i^{-1} / \partial y_i$ can equal zero or become infinite anywhere, i.e. the inequalities hold strictly in (7.21) and (7.22). In this case, $\partial q_i / \partial x_i$ is non-zero for all x_i and, moreover, has the same sign for all x_i in $\mathcal{C}$.

Observation 1: *For* $x_i^\dagger = \arg\min_{x_i} \mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}^\dagger\right)$, *if* $\check{x}_i < x_i^\dagger < \hat{x}_i$ *then* $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right)$ *is strictly monotone over the half-intervals* $y_i < y_i^\dagger$ *and* $y_i > y_i^\dagger$.

If $\check{x}_i < x_i^* < \hat{x}_i$, then condition (7.14) holds and $\check{y}_i < y_i^* < \hat{y}_i$. Then (7.26) implies that

$$\frac{\partial \mathcal{L}_i^{nc}\left(y_i^*, \boldsymbol{\lambda}^*\right)}{\partial y_i} = 0\,, \quad \check{y}_i < y_i^* < \hat{y}_i\,. \tag{7.27}$$

Since $\mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}^*\right)$ is strictly convex and $\partial q_i / \partial x_i$ has a constant sign, relation (7.26) implies that, if (7.27) is satisfied in $\mathcal{Y}$, then y_i^* represents the minimum of $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^*\right)$ over feasible y_i, and that $\partial \mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^*\right) / \partial y_i = 0$ only at $y_i = y_i^*$. Furthermore, the same observation demands that, for every $\boldsymbol{\lambda}^\dagger \neq \boldsymbol{\lambda}^*$ for which

$$x_i^\dagger = \check{x}_i < \arg\min_{x_i} \mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}^\dagger\right) < \hat{x}_i\,, \tag{7.28}$$

there exists a unique $y_i^\dagger = q\left(x_i^\dagger\right)$ that represents the minimum of $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right)$. Here again, clearly, $\partial \mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right) / \partial y_i = 0$ only at $y_i = y_i^\dagger$. This, in turn, leads us to conclude that $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right)$ must be strictly monotone over the half-intervals $y_i < y_i^\dagger$ and $y_i > y_i^\dagger$, where the sign of the gradient $\partial \mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right) / \partial y_i$ changes across $y_i = y_i^\dagger$ for any $\boldsymbol{\lambda}^\dagger$ (including $\boldsymbol{\lambda}^\dagger = \boldsymbol{\lambda}^*$).

□

Observation 2: *For* $x_i^\dagger = \arg\min_{x_i} L_i^{sc}\left(x_i, \boldsymbol{\lambda}^\dagger\right)$, *if* $x_i^\dagger$ *is on the bounds of* $\mathcal{C}$ *then* $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right)$ *is strictly monotone over the whole interval* $\check{y}_i \leq y_i \leq \hat{y}_i$.

If (7.12) holds and $\partial q_i / \partial x_i < 0$, or if (7.13) holds and $\partial q_i / \partial x_i > 0$, then $y_i^* = \hat{y}_i$. In this case, (7.26) implies that

$$\frac{\partial \mathcal{L}_i^{nc}\left(y_i^*, \boldsymbol{\lambda}^*\right)}{\partial y_i} \leq 0\,, \quad y_i^* = \hat{y}_i\,, \tag{7.29}$$

and that $\partial \mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda}^*)/\partial y_i < 0$ for any other feasible $(x_i, y_i = q_i(x_i), \boldsymbol{\lambda}^*)$. Also, if (7.12) holds and $\partial q_i/\partial x_i > 0$, or if (7.13) holds and $\partial q_i/\partial x_i < 0$, then $y_i^* = \check{y}_i$. It follows that

$$\frac{\partial \mathcal{L}_i^{nc}(y_i^*, \boldsymbol{\lambda}^*)}{\partial y_i} \geq 0\,, \quad y_i^* = \check{y}_i\,, \tag{7.30}$$

and $\partial \mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda}^*)/\partial y_i > 0$ for any other feasible $(x_i, y_i = q_i(x_i), \boldsymbol{\lambda}^*)$. Therefore y_i^* is again a minimum of $\mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda}^*)$ over feasible y_i. Clearly, for any other $\boldsymbol{\lambda}^\dagger$ for which

$$x_i^\dagger = \arg\min_{x_i} \mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}^\dagger\right) \tag{7.31}$$

lies on the boundary of $\mathcal{C}$, there will be a corresponding $y_i^\dagger$ that satisfies one of (7.29) or (7.30) at $\left(y_i^\dagger, \boldsymbol{\lambda}^\dagger\right)$, and $y_i^\dagger$ will be the minimum of $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right)$. In this case, $\mathcal{L}_i^{nc}\left(y_i, \boldsymbol{\lambda}^\dagger\right)$ must be strictly monotone over $\check{y}_i \leq y_i \leq \hat{y}_i$.

□

Observation 3: *A separable nonconvex problem that can be convexified by univariate transformations satisfies Attribute 1 and possesses a unique point that satisfies the necessary conditions for a saddle point of the Lagrangian.*

Given a strictly convex problem consistent with (7.1), together with a transformation of the type just discussed, the transformed problem is always one for which Attribute 1 holds. Additionally, given (7.27), (7.29) and (7.30), we can infer that

$$(y_i^* - \check{y}_i)(\hat{y}_i - y_i^*)\frac{\partial}{\partial y_i}\mathcal{L}(\boldsymbol{y}^*, \boldsymbol{\lambda}^*) = 0 \quad \forall \quad i\,. \tag{7.32}$$

Hence $\boldsymbol{y}^*$ satisfies the necessary conditions for a saddle point of the Lagrangian of the transformed problem $\mathcal{L}(\boldsymbol{y}, \boldsymbol{\lambda})$ in $\mathcal{Y}$, namely (7.24), (7.25), (7.29), (7.30) and (7.32). Notice that the above discussion can just as easily be run in reverse, in which case one would start with the observation that

$$\begin{aligned}\frac{\partial \mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda})}{\partial y_i} &= \frac{\partial}{\partial y_i}\left[\mathcal{L}_i^{sc}\left(q_i^{-1}(y_i), \boldsymbol{\lambda}\right)\right] \\ &= \frac{\partial \mathcal{L}_i^{sc}(x_i, \boldsymbol{\lambda})}{\partial x_i}\frac{\partial q_i^{-1}}{\partial y_i}\,.\end{aligned}$$

Then, by invoking the fact that Attribute 1 holds for $\mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda})$, the type of monotonicity exhibited by $\mathcal{L}_i^{nc}(y_i, \boldsymbol{\lambda})$ and the constant sign of $\partial q_i^{-1}/\partial y_i$, it is possible to show (analogously to the discussion above) that every saddle point of $\mathcal{L}(\boldsymbol{y}, \boldsymbol{\lambda})$ would correspond to a saddle point of $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$. However, since $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ represents a strictly convex problem, its saddle $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ is unique (up to the degeneracy discussed in Section 7.4.1). Therefore, the transformed problem can similarly only possess a single point, given by $(\boldsymbol{y}^*, \boldsymbol{\lambda}^*)$, that satisfies the necessary conditions for a saddle on its Lagrangian (up to the same degeneracy).

□

Observation 4: *The dual of the problem as defined by Falk is unchanged by convex transformation.*

Equations (7.28) and (7.31) are the primal-dual transformations that define the dual as given in (2.41). The dual is the same whether it is defined from the nonconvex problem or from its strictly convex transform, because

$$y_i^\dagger = q_i\left(\arg\min_{x_i} \mathcal{L}_i^{sc}\left(x_i, \boldsymbol{\lambda}^\dagger\right)\right)$$

and

$$\mathcal{L}_i^{nc}\left(y_i^\dagger, \boldsymbol{\lambda}^\dagger\right) \text{ in } \mathcal{Y} \;= \mathcal{L}_i^{sc}\left(x_i^\dagger, \boldsymbol{\lambda}^\dagger\right) \text{ in } \mathcal{C}.$$

□

Hence, every separable problem that is transformable to a strictly convex problem via the type of transformation defined above also satisfies Attribute 1, and thus is solvable using a dual method utilising the Falk dual. It is not necessary to actually apply the transformation. The untransformed (possibly nonconvex) problem has the same dual as its convex transform.

These observations allow us to further motivate the use of nonconvex approximations for building separable approximate subproblems in sequential approximate optimisation codes. It can be seen that the convexifiable problems discussed herein are essentially equivalent to their strictly convex counterparts. In some instances, such as with the example problems discussed in Chapters 5 and 6, it may be both advantageous and convenient to use nonconvex subproblems that more naturally fit the original problem, rather than either convexifying the subproblems or using other, less well-suited, strictly convex approximations.

7.5 Conclusions

We have investigated the link between two properties of continuous, separable, nonlinear and generally nonconvex programming problems. The first property is the ability of the problem to be solved via the application of Falk's dual method. The second is its ability to be transformed into a corresponding strictly convex form. We have limited the generality of this analysis, however, by considering only univariate bijective transformations.

We find that if such a nonconvex problem can be transformed into a corresponding strictly convex form via the types of transformations discussed, then it is also amenable to direct solution via Falk's dual method (i.e. without the necessity of actually transforming it) because its Lagrangian always has a unique minimum with respect to the primal variables. We have not established the converse though. This analysis does not indicate whether or not nonconvex problems exist that can be solved via the Falk dual but that cannot be transformed into a corresponding strictly convex form via the considered transformations. We also indicate that, given the types of programming problems and transformations discussed, the dual of a given problem remains unchanged upon application of a transformation.

The discussion helps to motivate and encourage the use of nonconvex approximations in sequential approximate optimisation algorithms that use Falk's dual approach in the solution of the SAO subproblems. We argue that it can sometimes be both possible and theoretically defensible to construct

separable nonconvex approximations to nonconvex problems, and to solve these subproblems in a dual setting. It may also be numerically advantageous to utilise nonconvex subproblems that suit the original problem more naturally, rather than using a standard, strictly convex approximation that may represent the original problem poorly.

Chapter 8

Bounding the dual for global convergence

The work presented in this chapter is reproduced from a paper titled "Placing upper bounds on the dual to circumvent the requirement of relaxation in globally convergent SAO implementations" [78]. The paper is co-authored by Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa, and Dr L.F.P. Etman of the Department of Mechanical Engineering at the Eindhoven University of Technology, Eindhoven, the Netherlands.

8.1 Abstract

We implement upper bounds on the popular Falk dual, and consider the use of the resulting bounded dual in globally convergent sequential approximate optimisation (SAO) procedures. We do so using conservative SAO sequences, but trust region sequences may equally well be used. We show that, in combination with conservatism, relaxation of the approximate subproblems is not required when such bounds are placed on the dual. Relaxation is commonly done to ensure that a KKT point exists for each subproblem; using a bounded dual, it is adequate to terminate each approximate subproblem at a non-stationary point. Under the assumption that the original problem possesses a KKT point and is not multimodal, the SAO sequence is guaranteed to converge, firstly, to a feasible point, and thereafter to the KKT point if the bounds on the dual variables are sufficiently large. In most cases of practical interest, upper bounds in the order of say 10^8 suffice. The bounded dual may be viewed as a simple penalty formulation to minimise the constraint infeasibility in some sense, but with the important advantage that the minimisation over the primal variables is done analytically – this retains the advantage that dual methods present when the number of design variables n is (far) greater than the number of constraints m. The proposed procedure has important implications for very large-scale optimisation, since no artificial variables are required, which may be demanding in terms of storage requirements.

8.2 Introduction

In the current chapter, we consider a general continuous nonlinear programming problem of the form stated in (2.15), and re-stated here for convenience (adopting the negative-null form for what follows):

Problem P_{NLP}

$$\begin{aligned} \min_{\boldsymbol{x}} \ & f_0(\boldsymbol{x}) \\ \text{subject to } & f_j(\boldsymbol{x}) \leq 0 \qquad j = 1, 2, \cdots, m, \\ & \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{8.1}$$

The function $f_0(\boldsymbol{x})$ is a real-valued scalar objective function, and the $f_j(\boldsymbol{x})$, $j = 1, 2, \cdots, m$ are m inequality constraint functions. The objective function $f_0(\boldsymbol{x})$ and constraint functions $f_j(\boldsymbol{x})$ all depend on the n real (design) variables $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}^T \in \mathcal{R}^n$, and the symbols $\check{x}_i$ and $\hat{x}_i$ denote, respectively, lower and upper bounds on the continuous real variable x_i. We do not here assume any special form associated with structural optimisation problems, since what follows is relevant to the solution of more general nonlinear programming problems.

The functions $f_\alpha(\boldsymbol{x})$, $\alpha = 0, 1, 2, \cdots, m$ are assumed to be (at least) once continuously differentiable. Problem P_{NLP} represents the general nonlinear (possibly multimodal) inequality constrained optimisation problem. However, it is assumed that the feasible region of Problem P_{NLP} is non-empty, and that in fact at least one KKT point exists.

If the evaluation of any of the functions f_α, $\alpha = 0, 1, 2, \cdots, m$ requires a numerical simulation, problem P_{NLP} is often solved using sequential approximate optimisation (SAO) methods. Most SAO algorithms used in structural optimisation are based on convex and separable approximation functions, which in turn makes using the Falk dual [2] attractive, since this allows for highly efficient dual forms (in particular when the number of constraints m are far less than the number of design variables n). For a discussion of the approximations often used in SAO, see Haftka and Gürdal [27] and Barthelemy and Haftka [79]; examples of SAO algorithms based on these approximations include the CONLIN algorithm developed by Fleury and Braibant [4], the method of moving asymptotes (MMA) developed by Svanberg [3, 32], generalisations of MMA by Bruyneel *et al.* [80], and SAOi developed by Groenwold and Etman [31].

It is often deemed necessary to relax[1] problem P_{NLP}, largely for the following reasons:

- Relaxation ensures the existence of a feasible solution to the problem. Specifically, the problem derived by relaxing P_{NLP} is guaranteed to have at least one optimal solution (which satisfies the KKT conditions), even if P_{NLP} itself happens to lack feasible solutions [6].
- In the same way, relaxation ensures the existence of optimal solutions for each approximate subproblem in an SAO implementation. If relaxation is not employed, it may happen (when a point of approximation is infeasible with respect to P_{NLP}) that the subproblem lacks feasible solutions, even if P_{NLP} does not.

[1]The term 'relaxation' in this chapters refers to the scalable modification of a problem to ensure that feasible solutions exist. The same term is used elsewhere in this document to denote (a) the weakening of the discreteness requirements often employed in the solution of material distribution problems and (b) the mechanism of allowing the stresses in portions of a structure to exceed the imposed constraint values.

- If the subproblem is constructed at a point that is infeasible with respect to $\mathrm{P_{NLP}}$, then a relaxation exists that makes the point of approximation feasible with respect to the associated relaxed problem.

Many forms for relaxation are possible. We will herein restrict ourselves to the form used by Svanberg [3, 6], which is given as follows:

Problem $\bar{P}_{\mathrm{NLP}}$

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \quad & f_0(\boldsymbol{x}) + \sum_{j=1}^{m} \left(c_j y_j + \frac{1}{2} d_j y_j^2 \right) \\ \text{subject to} \quad & f_j(\boldsymbol{x}) - y_j \leq 0 && j = 1, \cdots, m, \\ & y_j \geq 0 && j = 1, \cdots, m, \\ & \check{x}_i \leq x_i \leq \hat{x}_i && i = 1, \cdots, n. \end{aligned} \tag{8.2}$$

Typical settings are $c_j = 10^3$ and $d_j = 1$ ($d_j > 0$ results in a strictly convex penalty). In [6], Svanberg introduces a set of SAO methods based on conservative, convex and separable approximations (the CCSA methods). In the methodology, relaxation is used to make sure that the iterates are always feasible, and it is shown that the use of conservative approximations then leads to the robust global convergence characteristics that these methods possess. Concerning the optima of the relaxed problem, Svanberg demonstrates that, if $\boldsymbol{x}^*$ is a KKT point of problem $\mathrm{P_{NLP}}$ and the c_j are selected sufficiently large, then $(\boldsymbol{x}^*, \boldsymbol{y}^* = \boldsymbol{0})$ will be a KKT point of the relaxed problem $\bar{\mathrm{P}}_{\mathrm{NLP}}$ [3, 6].

While there are indeed various reasons why relaxation may be desirable, it is also possible to imagine situations where the contrary may be true. Reasons for not enforcing relaxation may include the increased storage requirements due to the auxiliary variables y_j, $j = 1, 2, \cdots, m$, and the sometimes unknown effect of the penalty parameters c_j, d_j on numerical performance. Hence, we herein aim to overcome the need for relaxation; we will do so in the dual setting, and we retain the use of conservatism for its global convergence characteristics.

More specifically, with reference to the reasons listed above for employing relaxation, we present an alternative approach for dealing with the second and third of these in a CCSA infrastructure. We develop a very straightforward (indeed trivial) modification to the dual originally proposed by Falk [2], which is so popular in SAO. We simply impose upper bounds on the dual variables. Although this is not unusual in and of itself, since upper bounds of some form must be imposed when numerically maximising the dual function, the novelty in our approach is to accept the bounded dual maximum as the next SAO iterate. If the dual upper bounds are selected large enough, then feasible subproblems will possess dual maxima within the bounded dual space. If, however, the subproblems are infeasible, then maximising the bounded dual corresponds to minimising the infeasibility. In this case we can show that a sequence of convex conservative subproblems will have decreasing infeasibility.

Hence, we assert that the bounded dual does not require relaxation of the subproblems, which is commonly done to ensure that a KKT point exists for each subproblem; it is adequate to terminate each subproblem at a non-stationary point if no stationary point exists. The resulting algorithm allows infeasible iterates and, more importantly, infeasible starting points; convergence to either a

local KKT point or a local point of minimum infeasibility is assured. In the case that $\mathrm{P_{NLP}}$ has a non-empty feasible domain and a unique optimum, the SAO sequence is guaranteed to converge, firstly to a feasible point, and thereafter to the KKT point, provided that the upper bounds on the dual variables are sufficiently large.

We make the assumption throughout that $\mathrm{P_{NLP}}$ has a non-empty feasible region, and we subscribe to the opinion that, if this is not the case, the problem itself should be reformulated. In the general case that $\mathrm{P_{NLP}}$ is multimodal, a globally convergent SAO algorithm using a bounded dual will converge to either a local KKT point, or to a local point of minimum infeasibility.

The chapter is arranged as follows: in Section 8.3, we discuss SAO using relaxation, followed in Section 8.4 by a discussion of SAO without relaxation, using a bounded dual. We present two numerical examples in Section 8.5, followed by concluding remarks in Section 8.6.

8.3 SAO using relaxation

Sequential approximate optimisation as a solution strategy for problem $\mathrm{P_{NLP}}$ seeks to construct successive approximate analytical subproblems $P[k],\ k = 1, 2, 3, \cdots$ at successive approximations $\boldsymbol{x}^{\{k\}}$ to the solution $\boldsymbol{x}^*$. The solution to subproblem $P[k]$ is $\boldsymbol{x}^{\{k*\}} \in \mathcal{R}^n$, to be obtained using any suitable continuous programming method. Thereafter, $\boldsymbol{x}^{\{k+1\}} = \boldsymbol{x}^{\{k*\}}$, the minimiser of subproblem $P[k]$.

In the following we will restrict ourselves to continuous SAO subproblems that are strictly convex, and that are constructed using separable approximation functions. More specifically: we will require that the approximate objective function $\tilde{f}_0$ is strictly convex, whereas the approximate constraint functions $\tilde{f}_j$ are required to be convex.

8.3.1 The approximate primal subproblem

A suitable approximate continuous subproblem for problem $\mathrm{P_{NLP}}$, constructed at $\boldsymbol{x}^{\{k\}}$, is

Primal approximate subproblem $\tilde{P}_P[k]$

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \tilde{f}_0(\boldsymbol{x}) \\ \text{subject to } & \tilde{f}_j(\boldsymbol{x}) \leq 0 \qquad j = 1, 2, \cdots, m, \\ & \check{x}_i \leq x_i \leq \hat{x}_i \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{8.3}$$

This primal approximate subproblem has n unknowns, m constraints, and $2n$ side or bound constraints; it may be solved using many a technique for constrained nonlinear programming.

8.3.2 The relaxed approximate primal subproblem

A suitable relaxed approximate continuous subproblem for problem $\bar{\mathrm{P}}_{\mathrm{NLP}}$, constructed at $\boldsymbol{x}^{\{k\}}$, is

Relaxed primal approximate subproblem $\bar{P}_P[k]$

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \quad & \bar{f}_0(\boldsymbol{x},\boldsymbol{y}) \\ \text{subject to} \quad & \bar{f}_j(\boldsymbol{x},y_j) \leq 0 \qquad j=1,2,\cdots,m, \\ & y_j \geq 0 \qquad\qquad j=1,2,\cdots,m, \\ & \check{x}_i \leq x_i \leq \hat{x}_i \qquad i=1,2,\cdots,n. \end{aligned} \tag{8.4}$$

Approximate primal subproblem $\bar{P}_P[k]$ has $n+m$ unknowns, m constraints, and $2n+m$ side or bound constraints; it is more demanding of storage requirements than approximate primal subproblem $\tilde{P}_P[k]$. Subproblems $\tilde{P}_P[k]$ and $\bar{P}_P[k]$ are related via the relationships

$$\bar{f}_0(\boldsymbol{x},\boldsymbol{y}) = \tilde{f}_0(\boldsymbol{x}) + \sum_{j=1}^{m}\left(c_j y_j + \frac{1}{2}d_j y_j^2\right)$$

and

$$\bar{f}_j(\boldsymbol{x},\boldsymbol{y}) = \tilde{f}_j(\boldsymbol{x}) - y_j, \quad j=1,2,\cdots,m.$$

8.3.3 The approximate dual subproblem

If primal approximate subproblem (8.3) is strictly convex and separable, we may invoke the efficient dual of Falk [2] and construct the following approximate dual subproblem:

Dual approximate subproblem $\tilde{P}_D[k]$

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & \tilde{\gamma}(\boldsymbol{\lambda}) = \tilde{f}_0\left(\boldsymbol{x}(\boldsymbol{\lambda})\right) + \sum_{j=1}^{m}\lambda_j \tilde{f}_j\left(\boldsymbol{x}(\boldsymbol{\lambda})\right) \\ \text{subject to} \quad & \lambda_j \geq 0 \qquad j=1,2,\cdots,m. \end{aligned} \tag{8.5}$$

This bound constrained problem requires the determination of the m unknowns λ_j only, subject to m non-negativity constraints on the λ_j. For what follows it is necessary to elaborate on the form of $\tilde{\gamma}(\boldsymbol{\lambda})$ in (8.5). We depart with the Lagrangian of the approximate subproblem during iteration k, $\tilde{\mathcal{L}}^{\{k\}}(\boldsymbol{x},\boldsymbol{\lambda})$, written as

$$\tilde{\mathcal{L}}^{\{k\}}(\boldsymbol{x},\boldsymbol{\lambda}) = \tilde{f}_0^{\{k\}}(\boldsymbol{x}) + \sum_{j=1}^{m}\lambda_j \tilde{f}_j^{\{k\}}(\boldsymbol{x}),$$

where the λ_j, $j=1,2,\cdots,m$, represent the Lagrangian multipliers. If the primal approximate subproblem is chosen to be strictly convex, which is standard practice, then $\tilde{\mathcal{L}}^{\{k\}}(\boldsymbol{x},\boldsymbol{\lambda})$ possesses a unique saddle point $(\boldsymbol{x}^{\{k*\}},\boldsymbol{\lambda}^{\{k*\}})$. Dropping the superscript $\{k\}$ for notational convenience, we note that the saddle point of the subproblem is given by

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x}}\{\tilde{\mathcal{L}}(\boldsymbol{x},\boldsymbol{\lambda}) : \check{x}_i \leq x_i \leq \hat{x}_i\} = \max_{\boldsymbol{\lambda}} \tilde{\gamma}(\boldsymbol{\lambda})$$

if the bound constraints of the primal subproblem form a closed and bounded domain in $\mathcal{R}^n$. This being the case, the function $\tilde{\gamma}(\boldsymbol{\lambda})$ is precisely the dual of Falk [2]. This dual becomes highly

efficient if the primal approximate subproblem is formulated in terms of *separable* approximations. As discussed in Section 2.3.3, minimising the Lagrangian in this case with respect to the n design variables reduces to performing n one-dimensional minimisations. Provided the minima exist, the dual is uniquely defined and the primal-dual relationships are derived from

$$x_i(\boldsymbol{\lambda}) = \arg\min_{x_i}\{\tilde{\mathcal{L}}(x_i, \boldsymbol{\lambda}) : \check{x}_i \leq x_i \leq \hat{x}_i\}\,, \tag{8.6}$$

which express the primal variables $\boldsymbol{x}$ (uniquely) as a function of the dual variables $\boldsymbol{\lambda}$. It is often necessary to employ a numerical method to solve (8.6) for the x_i, given particular $\boldsymbol{\lambda}$, even if the approximations used to construct the subproblem are strictly convex. However, for certain judicious choices of simple approximation functions (like quadratic functions, for instance), operation (8.6) results in algebraic expressions for the x_i in terms of $\boldsymbol{\lambda}$ that can be hard-coded into the dual solver. The dual function $\tilde{\gamma}(\boldsymbol{\lambda})$ is expressed as

$$\tilde{\gamma}(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}}\left[\tilde{f}_0(\boldsymbol{x}) + \sum_{j=1}^{m}\lambda_j\tilde{f}_j(\boldsymbol{x})\right] = \tilde{f}_0\left(\boldsymbol{x}\left(\boldsymbol{\lambda}\right)\right) + \sum_{j=1}^{m}\lambda_j\tilde{f}_j\left(\boldsymbol{x}\left(\boldsymbol{\lambda}\right)\right). \tag{8.7}$$

8.3.4 The relaxed approximate dual subproblem

Similar to the foregoing, if relaxed primal approximate subproblem (8.4) is strictly convex and separable, we may construct the following efficient relaxed approximate dual subproblem:

Relaxed dual approximate subproblem $\bar{P}_D[k]$

$$\begin{aligned} \max_{\boldsymbol{\lambda}}\ & \bar{\gamma}(\boldsymbol{\lambda}) = \bar{f}_0\left(\boldsymbol{x}(\boldsymbol{\lambda}), \boldsymbol{y}(\boldsymbol{\lambda})\right) + \sum_{j=1}^{m}\lambda_j\bar{f}_j\left(\boldsymbol{x}(\boldsymbol{\lambda}), y_j(\boldsymbol{\lambda})\right) \\ \text{subject to}\quad & \lambda_j \geq 0 \qquad\qquad j = 1, 2, \cdots, m. \end{aligned} \tag{8.8}$$

This bound constrained problem also requires the determination of the m unknowns λ_j only, subject to m non-negativity constraints on the λ_j. Due to the introduction of the additional variables $\boldsymbol{y}$, there can be many more primal-dual relationships for relaxed subproblem $\bar{P}_D[k]$ than for subproblem $P_D[k]$, particularly for m large.

8.3.5 Convergence of a relaxed approximate dual subproblem sequence

An arbitrary sequence of dual subproblems $\bar{P}_D[k]$ will not necessarily converge, nor terminate. However, if the sequence is cast in the framework of conservatism [6] or trust regions [24, 81], global convergence may be demonstrated under some conditions.

We will herein restrict ourselves to conservatism, since it is so simple and elegant (but not necessarily the best from a computational point of view for all possible problems); we do so in the dual context. A conservative approximation is one for which $\tilde{f}_\alpha^{\{k\}}\left(\boldsymbol{x}^{\{k*\}}\right) \geq f_\alpha\left(\boldsymbol{x}^{\{k*\}}\right)$ for all functions $\alpha = 0, 1, 2, \cdots, m$.

Proposition 1 *A relaxed SAO sequence* $(\boldsymbol{\lambda}^{\{k*\}}, \boldsymbol{x}^{\{k*\}}, \boldsymbol{y}^{\{k*\}})$, $k = 0, 1, 2, \cdots$ *resulting from a sequence of dual approximate subproblems* $\bar{P}_D[k]$ *will converge to a KKT point* $(\boldsymbol{\lambda}^*, \boldsymbol{x}^*, \boldsymbol{y}^*)$ *of relaxed problem* $\bar{P}_{\mathrm{NLP}}$ *if the primal approximate subproblems* $\bar{P}_P[k]$ *are conservative, convex and separable*[2].

Moreover, if problem P_{NLP} *has a feasible global minimiser* $\boldsymbol{x}^*$*, and the* c_j *in problem* $\bar{P}_{\mathrm{NLP}}$ *are selected sufficiently large, then there will exist a coincident solution to problem* $\bar{P}_{\mathrm{NLP}}$ *for which* $\boldsymbol{y}^* = \mathbf{0}$.

Proof: Firstly, from Theorem 11 of Falk [2], it follows that, if relaxed primal approximate subproblem (8.4) is strictly convex, then $\bar{\gamma}(\boldsymbol{\lambda}^{\{k*\}}) = \tilde{f}_0(\boldsymbol{x}^{\{k*\}}, \boldsymbol{y}^{\{k*\}}) \ \forall \ k$. Secondly, Theorem 7.1 of Svanberg [6] proves that if the approximations $\bar{f}_\alpha$, $\alpha = 0, 1, 2, \cdots, m$ are conservative, the SAO sequence will converge to a KKT point of $\bar{\mathrm{P}}_{\mathrm{NLP}}$ $(\boldsymbol{\lambda}^*, \boldsymbol{x}^*, \boldsymbol{y}^*)$. Furthermore, Svanberg also shows that, for every KKT point of $\mathrm{P}_{\mathrm{NLP}}$, there will exist a coincident KKT point of $\bar{\mathrm{P}}_{\mathrm{NLP}}$ $(\boldsymbol{\lambda}^*, \boldsymbol{x}^*, \boldsymbol{y}^*) = (\boldsymbol{\lambda}^*, \boldsymbol{x}^*, \mathbf{0})$, provided that the corresponding c_j are selected sufficiently large.

□

8.4 SAO without relaxation

It seems unnecessarily strict to require that a KKT point exists for each and every subproblem in the SAO sequence. Certainly, one could argue that it is simpler, and probably less demanding of computational resources, merely to show that a conservative subproblem sequence will eventually go to the KKT point of *some (feasible) subproblem*. If this can be shown, then, by virtue of the proof by Svanberg, convergence to a minimiser $\boldsymbol{x}^*$ will occur if the approximations reside in the CCSA class. Although the proof in [6] is phrased in terms of relaxed subproblems, it remains valid for unrelaxed problems, provided that the original problem possesses a KKT point and that the SAO is started at a feasible point. Then, each convex approximate subproblem has a unique KKT point and, due to conservatism, each iterate remains feasible with respect to the original problem $\mathrm{P}_{\mathrm{NLP}}$. In light of this, we specialise Proposition 1 as follows:

Proposition 2 *An SAO sequence* $(\boldsymbol{\lambda}^{\{k*\}}, \boldsymbol{x}^{\{k*\}})$, $k = 0, 1, 2, \cdots$ *resulting from a sequence of dual approximate subproblems* $\tilde{P}_D[k]$ *will converge to a KKT point* $(\boldsymbol{\lambda}^*, \boldsymbol{x}^*)$ *of unrelaxed problem* P_{NLP} *if the primal approximate subproblems* $\tilde{P}_P[k]$ *are conservative, convex and separable, and if the initial point in the sequence is feasible.*

□

[2] We assume throughout that, if the problem is feasible, the constraint qualification is satisfied at its solution(s).

8.4.1 The bounded approximate dual subproblem

Consider the following very simple bounded approximate dual subproblem:

Dual approximate subproblem $\hat{P}_D[k]$

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \ & \tilde{\gamma}(\boldsymbol{\lambda}) = \tilde{f}_0\left(\boldsymbol{x}(\boldsymbol{\lambda})\right) + \sum_{j=1}^{m} \lambda_j \tilde{f}_j\left(\boldsymbol{x}(\boldsymbol{\lambda})\right) \\ \text{subject to } & 0 \leq \lambda_j \leq \hat{\lambda} \qquad j = 1, 2, \cdots, m, \end{aligned} \tag{8.9}$$

with $\hat{\lambda} \to \infty$. We will interpret the operator '$\to \infty$' to mean that, although $\hat{\lambda}$ is a finite real number, its value may be chosen unrestrictedly large[3]. Bound constrained dual approximate subproblem $\hat{P}_D[k]$ merely requires the determination of the m unknowns λ_j, subject to $2m$ bound constraints. The addition of the upper bound on the λ_j does not influence the primal-dual relationships (8.6); nor are the storage requirements increased notably (the storage of a single scalar $\hat{\lambda}$ suffices).

8.4.2 Global convergence for a bounded approximate dual subproblem sequence

Proposition 2 indicates that relaxation is not required in a CCSA implementation when the original problem has feasible solutions and the initial iterate is feasible. We now consider the more general case of when the initial point is arbitrary, and potentially infeasible.

In this case, Proposition 1 argues that, when relaxation is employed, the solution found will be a KKT point of the relaxed problem $\bar{\text{P}}_{\text{NLP}}$. If the auxiliary variables are non-zero at this optimum, then the point of convergence will not correspond to a KKT point of the original unrelaxed problem P_{NLP}. This can occur when P_{NLP} possesses no feasible solutions, and/or when P_{NLP} is multimodal. In the latter case it is possible that the method can converge on a KKT point of the relaxed problem $\bar{\text{P}}_{\text{NLP}}$ that corresponds to an infeasible solution for the unrelaxed problem P_{NLP}, even though a feasible solution to P_{NLP} may exist. This is a familiar consequence of multimodality. Thus, for CCSA methods employing relaxation, there are two types of points to which convergence can occur. These are KKT points of $\bar{\text{P}}_{\text{NLP}}$ that do correspond to KKT points of P_{NLP}, and KKT points of $\bar{\text{P}}_{\text{NLP}}$ that do not correspond to KKT points of P_{NLP}.

A similar situation arises when global convergence of the CCSA methods is examined when employing the bounded dual instead of relaxation. In this case, the convergence proof below implies that the two types of points to which convergence can be proved are firstly KKT points of P_{NLP}, and secondly points at which the infeasibility of P_{NLP} is locally minimised, in the case when the set of KKT points is empty or when the problem is multimodal.

Noting the above, it is sufficient to follow the standard practice of presenting the convergence proof under the assumption that the problem has at least one KKT point to which local convergence must be demonstrated, starting from an arbitrary initial point inside its region of attraction.

[3]For many problems, the requirement that $\hat{\lambda}$ is finite is not required in the primal-dual relationships, but making this assumption here simplifies the step to the eventual computer implementation.

In the context of general (global) optimisation, however, it should be noted that $\mathrm{P_{NLP}}$ is an arbitrary nonlinear problem and, as such, may have many local minima identified by KKT points, as well as many points of minimal infeasibility, each with their own local regions of attraction. Convergence to either a local KKT point or a local point of minimal infeasibility is assured.

Proposition 3 *An SAO sequence* $(\boldsymbol{\lambda}^{\{k*\}}, \boldsymbol{x}^{\{k*\}})$, $k = 0, 1, 2, \cdots$ *resulting from a sequence of bounded dual approximate subproblems* $\hat{P}_D[k]$*, will converge to a KKT point* $(\boldsymbol{\lambda}^*, \boldsymbol{x}^*)$ *of problem* P_{NLP}*, or to a point of minimal infeasibility, if the primal approximate subproblems* $\tilde{P}_P[k]$ *are conservative, convex and separable, and if* $\hat{\lambda}$ *is sufficiently large (i.e.* $\hat{\lambda} \rightarrow \infty$*). This last implies that* $\hat{\lambda}$ *is required to be at least as large as the maximum component of* $\boldsymbol{\lambda}^*$*.*

Proof:

Consider the primal approximate subproblem $\tilde{P}_P[k]$, which is understood to have a strictly convex objective function, and convex and/or strictly convex constraints. Let us define the function $[\tilde{f}_j(\boldsymbol{x})]_+$ associated with a constraint function $\tilde{f}_j(\boldsymbol{x})$ to be

$$[\tilde{f}_j(\boldsymbol{x})]_+ = \max\{0, \tilde{f}_j(\boldsymbol{x})\}\,, \tag{8.10}$$

and let us for the time being assume that all $\tilde{f}_j(\boldsymbol{x})$ are strictly convex. The function $[\tilde{f}_j(\boldsymbol{x})]_+$ is the infeasibility associated with constraint $\tilde{f}_j(\boldsymbol{x})$ (it is non-zero only where $\tilde{f}_j(\boldsymbol{x})$ is infeasible). We indicate below that the total infeasibility $\tilde{F}^T(\boldsymbol{x}) = \sum_{j=1}^{m}[\tilde{f}_j(\boldsymbol{x})]_+$ is either minimised uniquely or, when this is not the case, that $\tilde{F}^T = 0$.

We use the term 'level surface' to denote the domain on which $f = a$ for a given function f, where a is some (real) number. It is evident that the only closed, convex level surface that $[\tilde{f}_j(\boldsymbol{x})]_+$ can possess is the domain on which $[\tilde{f}_j(\boldsymbol{x})]_+ = 0$ (even if $\tilde{f}_j(\boldsymbol{x})$ is monotonic, its domain is ultimately closed by the bound constraints on the x_i). For any other value of a, the associated level surface will not be convex. The function $[\tilde{f}_j(\boldsymbol{x})]_+$ is convex (constant) over this level surface (associated with $a = 0$), and strictly convex on any convex domain that does not intersect with this level surface. The feasible region of the subproblem is defined by the intersection of all the level surfaces $[\tilde{f}_j(\boldsymbol{x})]_+ = 0$, i.e. $\cap(\tilde{f}_j(\boldsymbol{x}) \leq 0) = \cap([\tilde{f}_j(\boldsymbol{x})]_+ = 0),\ j = 1, \cdots, m$.

Now, consider the function $\tilde{F}^T(\boldsymbol{x})$. Given the above, the only closed, convex level surface of $\tilde{F}^T$ is the feasible region $\tilde{F}^T = 0$, if it exists. $\tilde{F}^T$ is generally only convex, not strictly convex, but if the feasible region is empty, $\tilde{F}^T$ is strictly convex everywhere. $\tilde{F}^T$ is non-smooth, as its gradient is not continuous across the boundaries where the max operators take effect.

Due to the convexity of $\tilde{F}^T$, if the minimum of $\tilde{F}^T$ is not unique, it must be part of a closed convex level surface $\tilde{F}^T = a$, where a is some number that must satisfy $a \geq 0$. But, since we know that the only such level surface of $\tilde{F}^T$ is defined by $a = 0$, it follows that, if the feasible region is non-existent, $\tilde{F}^T$ must have a unique minimum. Obviously, $\tilde{F}^T$ does not have a unique minimum if the feasible region exists, unless it consists of only a single point. The function $\tilde{F}^T(\boldsymbol{x})$ is the infeasibility at $\boldsymbol{x}$ for (8.3). When the feasible region is empty, we will denote the unique minimum of $\tilde{F}^T$ by $\boldsymbol{x}^\ddagger$, while $\boldsymbol{x}^\diamond$ denotes the unique feasible minimum of the subproblem $\tilde{P}_P[k]$, when it exists.

Imagine that subproblem (8.3) is defined at point $\boldsymbol{x}^k$, and that it has no feasible solution. In this case, the dual is a concave surface that has no unbounded extremum. Therefore, a direction can be

found in the dual space, defined by (8.5), along which the dual of (8.3) increases without bound. Hence, the bounded dual (8.9) will have a bounded maximum at which at least one of the dual variables is on its upper bound. Denoting the dual coordinates of the dual maximum as $\boldsymbol{\lambda}^\dagger$, and the corresponding point in the primal space as $\boldsymbol{x}^\dagger$, we have

$$\tilde{\gamma}\left(\boldsymbol{\lambda}^\dagger\right) = \tilde{\mathcal{L}}\left(\boldsymbol{x}^\dagger\left(\boldsymbol{\lambda}^\dagger\right), \boldsymbol{\lambda}^\dagger\right) = \tilde{f}_0\left(\boldsymbol{x}^\dagger\right) + \sum_{j=1}^{m} \lambda_j^\dagger \tilde{f}_j\left(\boldsymbol{x}^\dagger\right),$$

and the following conditions hold:

$$\begin{aligned}
\lambda_j^\dagger = 0 \;\; \forall\, j \in \{j_1 : \frac{\partial \tilde{\gamma}}{\partial \lambda_{j_1}} = \tilde{f}_{j_1}\left(\boldsymbol{x}^\dagger\right) < 0\}, \\
0 \leq \lambda_j^\dagger \leq \hat{\lambda} \;\; \forall\, j \in \{j_2 : \frac{\partial \tilde{\gamma}}{\partial \lambda_{j_2}} = \tilde{f}_{j_2}\left(\boldsymbol{x}^\dagger\right) = 0\}, \\
\lambda_j^\dagger = \hat{\lambda} \;\; \forall\, j \in \{j_3 : \frac{\partial \tilde{\gamma}}{\partial \lambda_{j_3}} = \tilde{f}_{j_3}\left(\boldsymbol{x}^\dagger\right) > 0\},
\end{aligned}$$

in which j_1, j_2 and j_3 are sets of indexes. Hence, every

$$\begin{aligned}
[\tilde{f}_{j_1}\left(\boldsymbol{x}^\dagger\right)]_+ = 0 \leq [\tilde{f}_{j_1}\left(\boldsymbol{x}^k\right)]_+ \\
\text{and} \;\; [\tilde{f}_{j_2}\left(\boldsymbol{x}^\dagger\right)]_+ = 0 \leq [\tilde{f}_{j_2}\left(\boldsymbol{x}^k\right)]_+ .
\end{aligned} \tag{8.11}$$

At $\boldsymbol{\lambda}^\dagger$ we know that

$$\tilde{\mathcal{L}}\left(\boldsymbol{x}^\dagger, \boldsymbol{\lambda}^\dagger\right) \leq \tilde{\mathcal{L}}\left(\boldsymbol{x}^k, \boldsymbol{\lambda}^\dagger\right),$$

where the equality holds only if $\boldsymbol{x}^\dagger = \boldsymbol{x}^k$. Since all $\lambda_{j_1}^\dagger = 0$ and all $\lambda_{j_3}^\dagger = \hat{\lambda}$, we have

$$\begin{aligned}
\tilde{f}_0\left(\boldsymbol{x}^\dagger\right) + \sum_{j_2} \lambda_{j_2}^\dagger \tilde{f}_{j_2}\left(\boldsymbol{x}^\dagger\right) + \sum_{j_3} \hat{\lambda} \tilde{f}_{j_3}\left(\boldsymbol{x}^\dagger\right) \;\; \leq \\
\tilde{f}_0\left(\boldsymbol{x}^k\right) + \sum_{j_2} \lambda_{j_2}^\dagger \tilde{f}_{j_2}\left(\boldsymbol{x}^k\right) + \sum_{j_3} \hat{\lambda} \tilde{f}_{j_3}\left(\boldsymbol{x}^k\right).
\end{aligned} \tag{8.12}$$

Dividing through by $\hat{\lambda}$ and defining λ'_{j_2} as

$$0 \leq \left(\lambda'_{j_2} = \frac{\lambda_{j_2}^\dagger}{\hat{\lambda}}\right) \leq 1,$$

we note that $\sum_{j_3} \tilde{f}_{j_3}\left(\boldsymbol{x}^\dagger\right) = \sum_{j_3} [\tilde{f}_{j_3}\left(\boldsymbol{x}^\dagger\right)]_+$, all $\tilde{f}_{j_2}\left(\boldsymbol{x}^\dagger\right) = 0$, and $\lambda'_{j_2} \tilde{f}_{j_2}\left(\boldsymbol{x}^k\right) \leq [\tilde{f}_{j_2}\left(\boldsymbol{x}^k\right)]_+$. Therefore, equation (8.12) can be simplified to yield

$$\begin{aligned}
\frac{\tilde{f}_0\left(\boldsymbol{x}^\dagger\right)}{\hat{\lambda}} + \sum_{j_2} [\tilde{f}_{j_2}\left(\boldsymbol{x}^\dagger\right)]_+ + \sum_{j_3} [\tilde{f}_{j_3}\left(\boldsymbol{x}^\dagger\right)]_+ \;\; \leq \\
\frac{\tilde{f}_0\left(\boldsymbol{x}^k\right)}{\hat{\lambda}} + \sum_{j_2} [\tilde{f}_{j_2}\left(\boldsymbol{x}^k\right)]_+ + \sum_{j_3} [\tilde{f}_{j_3}\left(\boldsymbol{x}^k\right)]_+ .
\end{aligned}$$

Using (8.11), we conclude that

$$\begin{aligned}\frac{\tilde{f}_0\left(\boldsymbol{x}^{\dagger}\right)}{\hat{\lambda}}+\sum_{j_1}[\tilde{f}_{j_1}\left(\boldsymbol{x}^{\dagger}\right)]_+ + \sum_{j_2}[\tilde{f}_{j_2}\left(\boldsymbol{x}^{\dagger}\right)]_+ + \sum_{j_3}[\tilde{f}_{j_3}\left(\boldsymbol{x}^{\dagger}\right)]_+ \quad \leq \\ \frac{\tilde{f}_0\left(\boldsymbol{x}^{k}\right)}{\hat{\lambda}}+\sum_{j_1}[\tilde{f}_{j_1}\left(\boldsymbol{x}^{k}\right)]_+ + \sum_{j_2}[\tilde{f}_{j_2}\left(\boldsymbol{x}^{k}\right)]_+ + \sum_{j_3}[\tilde{f}_{j_3}\left(\boldsymbol{x}^{k}\right)]_+\end{aligned}$$

or, more succinctly:

$$\frac{\tilde{f}_0\left(\boldsymbol{x}^{\dagger}\right)}{\hat{\lambda}}+\sum_{j=1}^{m}[\tilde{f}_j\left(\boldsymbol{x}^{\dagger}\right)]_+ \quad \leq \quad \frac{\tilde{f}_0\left(\boldsymbol{x}^{k}\right)}{\hat{\lambda}}+\sum_{j=1}^{m}[\tilde{f}_j\left(\boldsymbol{x}^{k}\right)]_+ . \tag{8.13}$$

We have already shown that the infeasibility is minimised uniquely for a strictly convex subproblem with strictly convex constraints and no feasible solution, in which case (8.13) indicates that, as $\hat{\lambda} \rightarrow \infty$, $\boldsymbol{x}^{\dagger} \rightarrow \boldsymbol{x}^{\ddagger}$. If the subproblem has a feasible solution $\boldsymbol{x}^{\diamond}$, the primal coordinates $\boldsymbol{x}^{\dagger}$ associated with the dual maximum must tend to (or become) $\boldsymbol{x}^{\diamond}$ as $\hat{\boldsymbol{\lambda}} \rightarrow \infty$, because the dual has a definite maximum in this case. For subproblems in which the $\tilde{f}_j,\ j = 1, \cdots, m$ are convex, but not necessarily strictly convex, the infeasibility $\tilde{F}^T(\boldsymbol{x})$ might not have a unique minimum. Instead, the points at which the infeasibility is minimised in this case can generally occupy a convex set $\mathcal{X}$ in the domain of the subproblem. However, the presence of the strictly convex term $\tilde{f}_0$ in (8.13) ensures that the primal-dual relationships exist uniquely for any finite value of $\hat{\lambda}$. Furthermore, the presence of $\tilde{f}_0$ also ensures that the point $\boldsymbol{x}^{\ddagger}$, to which $\boldsymbol{x}^{\dagger}$ tends as $\hat{\lambda} \rightarrow \infty$, is again a unique point. Obviously, $\boldsymbol{x}^{\ddagger}$ will be a member of the set $\mathcal{X}$.

Now, making use of conservatism implies that a set of conservative approximations can be found for which $f_j\left(\boldsymbol{x}^{\dagger}\right) \leq \tilde{f}_j\left(\boldsymbol{x}^{\dagger}\right)\ \forall\ j$, and Svanberg has proved that conservatism can be satisfied within a finite number of inner loop iterations [6]. Renaming the $\boldsymbol{x}^{\dagger}$ at which conservatism is satisfied for outer iteration k as $\boldsymbol{x}^{\{k*\}}$, and labelling the associated dual coordinates as $\boldsymbol{\lambda}^{\{k*\}}$, we assert that the bounded dual serves to reduce the infeasibility of consecutive iterates in an infeasible conservative SAO sequence, because

$$\sum_{j=1}^{m}[f_j\left(\boldsymbol{x}^{\{k*\}}\right)]_+ \leq \sum_{j=1}^{m}[\tilde{f}_j\left(\boldsymbol{x}^{\{k*\}}\right)]_+ \leq \sum_{j=1}^{m}[\tilde{f}_j\left(\boldsymbol{x}^{k}\right)]_+ = \sum_{j=1}^{m}[f_j\left(\boldsymbol{x}^{k}\right)]_+ . \tag{8.14}$$

With the assumption that the original problem is unimodal and has a non-empty feasible region, conservatism ensures convergence firstly to a feasible point, and then, by virtue of Proposition 2, to a KKT point of $\mathrm{P_{NLP}}$.

The bounded dual may be viewed as a penalty formulation to minimise the constraint infeasibility. The definition of the dual requires that the Lagrangian is minimised with respect to the primal variables $\boldsymbol{x}$. If the subproblem has no feasible solution, then (8.13) implies that, at the bounded maximum of the dual, the equation

$$\tilde{f}_0\left(\boldsymbol{x}\right)+\hat{\lambda}\sum_{j}^{m}[\tilde{f}_j\left(\boldsymbol{x}\right)]_+$$

is minimised. This is a linear penalty function in which the total infeasibility has been penalised with the factor $\hat{\lambda}$. As $\hat{\lambda}$ becomes unrestrictedly large, the resulting minimisation locates the point of minimal infeasibility for the subproblem. If, in addition, the infeasibility of $\mathrm{P_{NLP}}$ – now possibly multimodal – is locally minimised at such a point, then it becomes a terminal point for the SAO.

□

8.4.3 Numerical considerations

In practice, we do not require $\hat{\lambda} \to \infty$. Instead, we require that $\hat{\lambda}$ is 'sufficiently large' in the spirit of inexact minimisation methods (see Bertsekas [26], for example, who discusses inexact minimisation in the context of augmented Lagrangian methods). It is, however, required that $\hat{\lambda} > \max\{\lambda_j^*\}$. The magnitudes of the λ_j^* are of course unknown, but, in practice, any large number for the $\hat{\lambda}$ suffices. We typically[4] use $\hat{\lambda} = 10^8$, but larger values presented no problems whatsoever to the bound-constrained BFGS [74, 75] solver that we often use to solve the dual subproblems.

Reasonable estimates for $\hat{\lambda}$ can sometimes even be made on the basis of knowledge of the optimisation problem at hand, e.g. see Svanberg [3, 6], who argues in favour of using similar information to estimate reasonable values for the relaxation penalties in his MMA algorithm.

We have assumed throughout that a feasible global minimiser $\boldsymbol{x}^*$ for problem $\mathrm{P_{NLP}}$ does exist. If the contrary is true, then the constraint infeasibility $\sum_{j=1}^{m}[f_j(\boldsymbol{x})]_+$ is clearly minimised in some sense. In fact, the algorithm will terminate at a point of local minimum infeasibility (in terms of a linear, unweighted sum of the infeasibilities), provided that such a point exists and if the infeasibility of the problem is locally convex around such a point. However, reformulation of problem $\mathrm{P_{NLP}}$ may then be called for, rather than accepting this point.

Finally: for problems (mostly pathalogical in nature) that are 'wildly' infeasible, it may be computationally demanding to find the point $(\boldsymbol{\lambda}^{\{k*\}}, \boldsymbol{x}^{\{k*\}})$ on the subproblem level to a reasonable accuracy, due to scaling effects. (Not that a high accuracy is required in practice under these conditions.) A computational 'shortcut' then is to simply enforce $\lambda_e^{\{k*\}} = \hat{\lambda}$, where e represents the set of constraints for which $f_j(\boldsymbol{x}^{\{k\}}) > \mu$ hold, with μ large, say 10^6, and then to eliminate these dual variables from the maximisation of $\tilde{\gamma}(\boldsymbol{\lambda})$. Fixing the eliminated variables λ_e at the upper bound $\hat{\lambda}$ implies that these dual variables still influence the primal variables, but that it is assumed that a feasible solution to these constraints cannot be found in iteration k. In other words, we assume that $\tilde{f}_e(\boldsymbol{x}^{\{k*\}}) > 0$, which seems reasonable for $\tilde{f}_e(\boldsymbol{x}^{\{k\}}) > \mu = 10^6$. By virtue of Proposition 2, setting $\lambda_e^{\{k*\}} = \hat{\lambda}$ will still drive the subproblems to feasibility.

[4] We have performed extensive numerical experimentation with an upper bound of 10^8, using many test problems popular in the SAO literature (not reported herein); this always proved adequate.

8.5 Numerical experiments

8.5.1 The approximations used in the example

The approximations used in the following examples are the simple separable spherical quadratic approximations that we have previously proposed [31] for use in convergent dual SAO algorithms. These approximations derive from an incomplete series expansion (ISE) suggested by Groenwold *et al.* as the basis for function approximation in separable SAO infrastructures [61]; they are expressed as

$$\tilde{f}_\alpha(\boldsymbol{x}) = f_\alpha(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} (x_i - x_i^{\{k\}}) + \frac{1}{2}\sum_{i=1}^{n} c_{2i_\alpha}^{\{k\}} \left(x_i - x_i^{\{k\}}\right)^2 . \tag{8.15}$$

For $\alpha = 0$, we understand that the objective function is approximated; for $1 \leq \alpha \leq m$, inequality constraint function j is approximated. It is also understood that

$$\left(\frac{\partial f_\alpha}{\partial x_i}\right)^{\{k\}} = \frac{\partial f_\alpha}{\partial x_i}(\boldsymbol{x}^{\{k\}}) ,$$

being the partial derivative of f_α with respect to x_i at the point $\boldsymbol{x}^{\{k\}}$. Approximation (8.15) is convex if $c_{2i_\alpha}^{\{k\}} \geq 0 \; \forall \; i$, while the approximation is strictly convex if the inequality holds for all i. We select $c_{2i_\alpha}^{\{k\}} \equiv c_{2_\alpha}^{\{k\}} \forall \; i$, which results in a *spherical* quadratic approximation [82], and requires the determination of the single unknown $c_{2_\alpha}^{\{k\}}$. This is the simplest instance of the ISE, and the unknown parameter $c_{2_\alpha}^{\{k\}}$ may then be obtained by enforcing the condition

$$\tilde{f}_\alpha(\boldsymbol{x}^{\{k-1\}}) = f_\alpha(\boldsymbol{x}^{\{k-1\}}) , \tag{8.16}$$

which implies that

$$c_{2_\alpha}^{\{k\}} = \frac{2[f_\alpha(\boldsymbol{x}^{\{k-1\}}) - f_\alpha(\boldsymbol{x}^{\{k\}}) - \boldsymbol{\nabla}^T f_\alpha(\boldsymbol{x}^{\{k\}})(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})]}{\|\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}\|_2^2} . \tag{8.17}$$

To obtain strictly convex dual subproblems, we enforce $c_{2i_\alpha}^{\{k\}} = \max\{\epsilon_n > 0, c_{2i_\alpha}^{\{k\}}\} \; \forall \; i$ if $\alpha = 0$, and $c_{2i_\alpha}^{\{k\}} = \max\{0, c_{2i_\alpha}^{\{k\}}\} \; \forall \; i$ if $\alpha > 0$, with ϵ_n selected rather arbitrarily as 10^{-5}. The curvatures $c_{2i_\alpha}^{\{k\}}$ are also bounded above.

8.5.2 Nonconvex example

We start with a nonconvex example problem proposed by Svanberg [6]. The problem is expressed in terms of the symmetric, fully populated $n \times n$ matrices $\boldsymbol{S}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$, with elements given by

$$s_{ij} = \frac{2 + \sin(4\pi\vartheta_{ij})}{(1 + |i-j|)\ln(n)} , \quad p_{ij} = \frac{1 + 2\vartheta_{ij}}{(1 + |i-j|)\ln(n)} , \quad q_{ij} = \frac{3 - 2\vartheta_{ij}}{(1 + |i-j|)\ln(n)} ,$$

where

$$\vartheta_{ij} = \frac{i + j - 2}{2n - 2} \in [0, 1] \quad \forall \quad i \text{ and } j,$$

and $n > 1$. The problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} \;\; f_0(\boldsymbol{x}) &= \boldsymbol{x}^T \boldsymbol{S} \boldsymbol{x} \\ \text{subject to} \;\; f_1(\boldsymbol{x}) &= \frac{n}{2} - \boldsymbol{x}^T \boldsymbol{P} \boldsymbol{x} \leq 0, \\ f_2(\boldsymbol{x}) &= \frac{n}{2} - \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} \leq 0, \\ -1 \leq x_i &\leq 1\,, \end{aligned}$$

in which the objective function $f_0(\boldsymbol{x})$ is strictly convex, but the nonlinear constraint functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are strictly concave. The strictly convex approximation strategy described in Section 8.5.1 is used to construct the approximate subproblems.

The iterations are terminated when $\|\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}\| \leq \epsilon_x = 10^{-4}$ and we do not use any move limit whatsoever. For $\hat{\lambda}$ we selected a value of $\hat{\lambda} = 10^5$, and for the relaxation penalty parameters we used $c_j = 10^3$ and $d_j = 1$. The conservative SAO algorithm used to generate the results is presented in [31]. (The specific algorithmic settings used are not very interesting for our current purposes; we merely wish to illustrate the working of the bounded dual).

The iteration paths for the bounded dual and the relaxed problems are presented in Table 8.1; for the sake of brevity and clarity, we present results for $n = 2$ only. The infeasible starting point (listed in the table) is randomly generated. In the table, $h = \max\{f_1, f_2\}$, while N_s reflects the required number of evaluations of the *subproblems* by the bound-constrained BFGS solver that we mentioned in Section 8.4.3.

To machine precision, the trajectories of the bounded dual and the relaxed formulation are identical. The (unrelaxed) primal subproblems are strictly convex and, of course, are identical for both algorithms. Their minimisers should therefore coincide. During the first iteration, however, the constructed subproblem has no feasible minimiser. The algorithm employing relaxation locates a relaxed feasible solution that is identical to the point of minimal infeasibility found by bounding the dual. This correspondence makes sense (in retrospect), given the convexity of the relaxation penalisation.

Hence, bounding the dual accomplishes exactly what relaxation does, although perhaps more simply. Note that the computational effort for minimising the subproblem is markedly less for the bounded dual during the first iteration. Tentatively, this may indicate that finding the maximum of a bounded dual is numerically easier than finding the turning point of the dual for the equivalent relaxed subproblem if the subproblem is infeasible. On the bounded dual surface, the gradients of the dual corresponding to the violated constraints are all positive, and the associated dual variables 'sit' at the upper bounds.

8.5.3 The snake problem

Next, consider the so-called 'snake problem', also proposed by Svanberg [83], in particular for "anyone who wants to test a new method for nonlinear optimisation". Let d be a given positive integer, and let δ_s be a given 'small' positive real number. For $i = 1, 2, \cdots, d$, let

$$\psi_i = \frac{(3i - 2d)\pi}{6d}\,, \quad g_i(\boldsymbol{x}) = \frac{x_i^2 + x_{d+i}^2 - 1}{\delta_s} \;\text{ and }\; h_i(\boldsymbol{x}) = \frac{x_{2d+i} - 2x_i x_{d+i}}{\delta_s}\,.$$

Then consider the following problem in the variables $\boldsymbol{x} = (x_1, \cdots, x_{3d})^T$:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & f_0(\boldsymbol{x}) = \sum_{i=1}^{d} (x_i \cos\psi_i + x_{d+i} \sin\psi_i - 0.1 x_{2d+1}) \\ \text{subject to } & \sum_{i=1}^{d} (x_i^2 + x_{2d+i}^2) \leq d\,, \\ & -2 \leq g_i(\boldsymbol{x}) + g_i(\boldsymbol{x})^7 \leq 2,\ i = 1, 2, \cdots, d\,, \\ & -2 \leq h_i(\boldsymbol{x}) + h_i(\boldsymbol{x})^7 \leq 2,\ i = 1, 2, \cdots, d\,, \\ & -2 \leq x_j \leq 2,\ j = 1, 2, \cdots, 3d\,. \end{aligned}$$

For a short discussion of the problem, see Svanberg, who considers the problem "rather difficult to solve" if the following feasible, but far from optimal, starting point $\boldsymbol{x}^{\{0\}}$ is chosen:

$$x_i^{\{0\}} = \cos(\psi_i + \frac{\pi}{12}),\ x_{d+i}^{\{0\}} = \sin(\psi_i + \frac{\pi}{12})\,,$$

and

$$x_{2d+i}^{\{0\}} = \sin(2\psi_i + \frac{\pi}{6}),\ i = 1, 2, \cdots, d\,.$$

We will present results for $d = 10$ (and hence, $n = 30$ and $m = 41$), and $\delta_s = 0.1$. However, we generate a random starting point, which is highly infeasible, and apparently far more difficult than the feasible starting point given above. For this problem we use the default value for $\hat{\lambda}$ in our code, namely $\hat{\lambda} = 10^8$, while for relaxation we once again use $c_j = 10^3$ and $d_j = 1$.

Let $h = \max\{f_j\}$, $j = 1, 2, \cdots, m$, i.e. the maximum constraint violation. Results for the bounded dual are depicted in Figure 8.1, which nicely illustrates how the infeasibility is decreased after every conservative iteration until a feasible iterate is obtained at iteration 35. Thereafter, the iterates remain feasible. Note the very large magnitude of h in the earlier iterations. Figure 8.1(b) is deserved of some elaboration: after iteration 35, h is notably less than zero for many an iteration. This is a result of the strategy used to enforce conservatism, in which the curvatures in the inner loop are simply multiplied by 2. A smaller resolution in increasing the curvatures results in iterates for which h is closer to zero.

Figure 8.1(c) illustrates the effect of the concavity of the dual on the values of the dual variables associated with violated constraints. For the sake of clarity and brevity, we have only depicted the value of the largest dual variable $\Lambda = \max\{\lambda_j\}$, $j = 1, 2, \cdots, m$. For the snake example, $\Lambda^* \approx 0.494$.

We have *not* been able to solve the snake problem with a random starting point using only relaxation in a numerically stable way. The reason, which once again is proposed tentatively, again seems to be that finding the maximum of the bounded dual is numerically easier than finding the turning point of the dual for the relaxed subproblem. Presumably, the dual subproblems become badly posed for the very high values of relaxation needed. The initial infeasibilities at the random starting point are of the order of 10^{13} for this problem, which means that (some of) the initial relaxations y_j must be of the same order. And they are squared in the objective function of the relaxed subproblem. It is widely known that, in order for convergence to be achieved using dual solvers, the dual maximum must be located accurately, and the primal-dual relationships must likewise be

determined precisely. At any rate, the bound constrained BFGS solver that we use to maximise the duals appears to prefer the bounded dual for this problem.

The above comparison does not imply that we expect the bounded dual to always outperform relaxation, nor that we believe bounding the dual to be an inherently superior procedure to relaxation. Any such assertion is a problem-specific statement, as a no-free-lunch-like argument would of course suggest. Indeed, for some problems our preliminary experimentation suggests that it sometimes may be attractive to use both. The purpose of these numerical examples is simply to show that bounding the dual is a viable and simple alternative to implementing relaxation, and that it also may be numerically more stable in some cases.

8.6 Conclusions

We have presented a simple modification to the popular dual proposed by Falk for use in convergent SAO sequences. The modification requires only that upper bounds are placed on the dual variables. This dual does not require relaxation of the approximate subproblems to ensure that a KKT point exists for each approximate subproblem; if a subproblem has an empty feasible region, it is adequate to terminate the search for its optimum at a non-stationary point. However, the convergence of the SAO sequence is not influenced detrimentally. Indeed, the sequence is guaranteed to converge firstly to a feasible point, and thereafter to a feasible minimiser, if the bounds on the dual are sufficiently large and if a unique minimiser indeed exists. In most cases of practical interest, extensive numerical experimentation suggests that upper bounds on the dual variables in the order of 10^8 suffice.

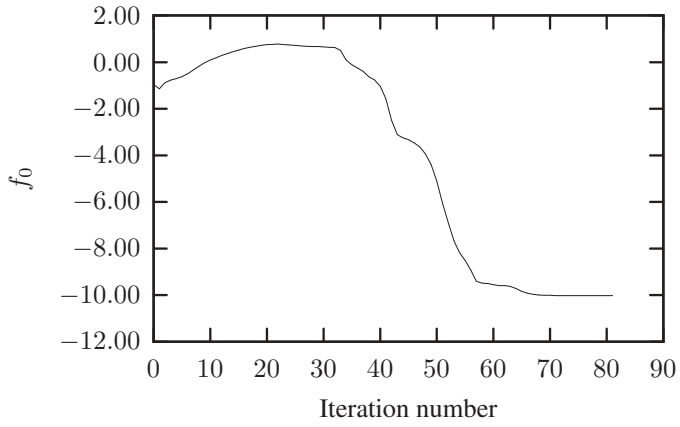
We have demonstrated that the SAO sequence converges using conservative, convex and separable approximations, but the same may also be demonstrated for dual trust-region methods, etc. In addition, like relaxation, the bounded dual may also be used in SAO implementations that have no facility to force global convergence.

The proposed bounded dual not only has important implications for large-scale optimisation, since no artificial variables are required that may be demanding of storage requirements, but possibly also for the restoration phases of incompatible subproblems in primal algorithms based on the nonlinear filtered acceptance of iterates. The bounded dual may then be viewed as a simple penalty formulation to minimise the constraint infeasibility, but with the important advantage that the minimisation over the primal variables is done analytically. Finally, the bounded dual is also extremely simple to implement.

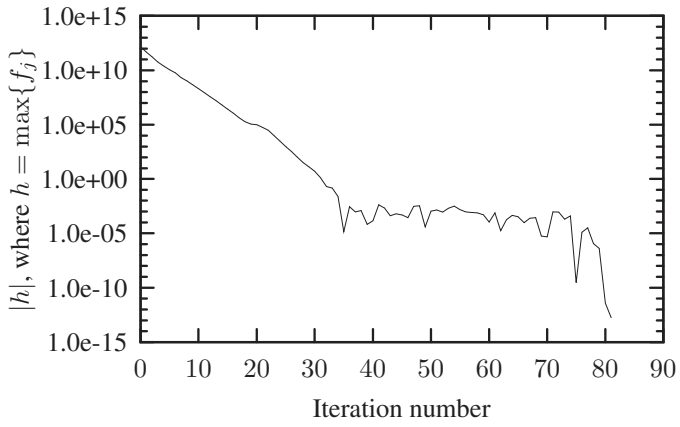
	Bounded dual							Relaxation				
k	f_0	h	x_1	x_2	λ_1	λ_2	N_s	λ_1	λ_2	y_1	y_2	N_s
0	0.2536498	8.374×10^{-01}	0.336	-0.224								
1	1.2881137	2.909×10^{-02}	0.736	-0.567	10^5	92843	28	1000.5	927.9	0.514	0.000	61
2	1.0592279	-3.840×10^{-02}	0.611	-0.599	0.519	0.421	35	0.519	0.421	0.000	0.000	28
3	1.0011105	-5.423×10^{-04}	0.589	-0.588	0.476	0.454	5	0.476	0.454	0.000	0.000	5
4	1.0000001	-2.312×10^{-07}	0.588	-0.588	0.499	0.499	5	0.499	0.499	0.000	0.000	5
5	1.0000000	-6.087×10^{-14}	0.588	-0.588	0.499	0.499	4	0.499	0.499	0.000	0.000	8

Table 8.1: The iteration paths for the nonconvex example problem. For relaxation, the columns f_0, h, x_1 and x_2 are not shown, since they are identical to those obtained with the bounded dual, except for h at the final iteration, which equals 2.948×10^{-10} in the case of relaxation. (The values in the four mentioned columns are similar to at least the number of digits shown, but mostly more. For the primal variables x_1 and x_2, for example, the first 10 significant digits are identical.)

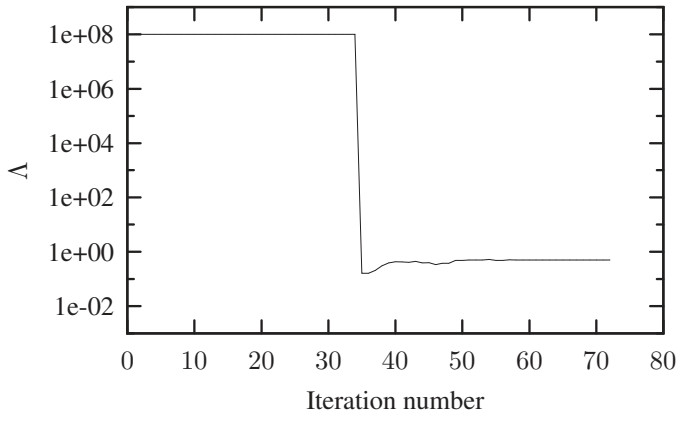
Figures



(a) Objective function value f_0



(b) Maximum constraint violation $|h|$, where $h = \max\{f_j\}$



(c) Largest dual variable $\Lambda = \max\{\lambda_j\}$

Figure 8.1: The snake problem: convergence history for $d = 10$, beginning at a highly infeasible, randomly generated starting point.

Chapter 9

Large-scale structural optimisation with stress constraints

The bulk of the work presented in this chapter is intended for submission, as an article, in collaboration with Prof. Albert A. Groenwold of the Department of Mechanical Engineering at the University of Stellenbosch, Stellenbosch, South Africa.

9.1 Abstract

This chapter is concerned with the solution of large-scale topology optimisation problems using sequential approximate optimisation in combination with a dual method for the solution of the generated approximate subproblems. Specifically, we solve standard examples of the weight minimisation problem with local stress constraints, as well as standard examples of the minimum compliance problem subject to local stress constraints and the usual constraint on the maximum allowable volume. In the context of sequential approximate optimisation using separable approximations, the procedure followed for solving the problems depends on a number of considerations. Among these are the types of approximations used to construct the subproblems, the method of constraint relaxation employed for the stress constraints, and the strategy used to determine which constraints should be included in the definition of a subproblem (the pre-selection strategy). Additionally, it is suggested that a computational advantage can be gained by limiting the number of terms used to construct the Jacobian of the subproblem, since for structural optimisation problems it is often the case that many elements in the Jacobian are orders of magnitude smaller than the most significant elements and therefore would appear to be insignificant by comparison. Intuitively it seems permissible to ignore the insubstantial elements when constructing the approximate subproblems; we investigate the effect of doing so. The aim of this chapter is to provide an indication of how these aspects affect the numerical solution of large-scale topology optimisation problems. Thus, the method applied is simply to chronicle the behaviour of the numerical solution procedure and the quality of the solutions obtained, rather than to attempt a comparative theoretical justification for the various algorithmic permutations available. In this way we hope to demonstrate the utility of the solution algorithm for large-scale problems, as well as to provide useful indications of the effect that various parameters have on the solution of the problem.

9.2 Introduction

Topology optimisation, via the material distribution method explained by Bendsøe and Sigmund [8], is able to provide an indication of the optimal distribution of material within a given domain subject to physical constraints and under the application of applied loads. However, topology optimisation inherently is a computationally expensive procedure, being a marriage of structural analysis and numerical optimisation. The structural analysis component (we herein confine ourselves to the prediction of structural responses using the finite element method) requires the numerical solution of a system of equations arrived at by discretising the physical structure to solve the equilibrium equations. In the material distribution method, it is the entire spatial domain that structural elements may occupy that must be discretised and is subject to analysis. The larger the domain size and the more refined the required scale of the structural details, the larger the analysis model becomes and the more demanding is the solution of the associated equations.

The optimisation component is itself also a numerically demanding procedure and is, moreover, iterative in nature for all but the simplest types of problems. It too becomes more demanding to solve the greater the number of design variables and the greater the number of constraints considered – both of which are usually directly related to the discretisation employed in the analysis component. Hence, we see that a barrier to the widespread adoption of these techniques in industry is the computational expense of carrying out the process, which limits the application of topology optimisation to either the design of small single components, or else to the design of larger structures using limited low-fidelity models in the analysis. For these reasons, it is necessary to employ or develop solution and optimisation procedures for topology optimisation that, as regards their required computational imperatives, are as efficient as possible.

Historically, two optimisation approaches to the solution of structural optimisation problems have been pursued for their efficiency (refer to the brief introduction given in Section 2.2). The first is the family of procedures known as the optimality criteria methods, explained, for example, in [27], and the second is the use of sequential approximate optimisation utilising separable strictly convex approximations, as used, for instance, in [28]. The two approaches were shown to be closely related, and in some instances equivalent, by Fleury [1]. Using the weight minimisation problem, Fleury showed that the iterative design updates arrived at by the favoured OC method of the time could also be derived using an SAO approach in which explicit separable and strictly convex subproblems were derived based on a linearisation of the objective and constraint functions at a point in the design space, and in which a dual method was employed in solving the subproblems.

The problem linearisation is constructed as a first-order Taylor expansion in terms of either the design variable or the reciprocals thereof. In each case, the approximations reflect the sensitivities of the many structural dependencies well. For instance, in the weight minimisation problem the objective function is linear and the stress or displacement constraints are well represented by the reciprocal linearisations. On the other hand, in the minimum compliance problem the volume constraint is linear, whereas the compliance objective is approximated well by the reciprocal functions. Both of these approximation techniques require only first-order information to be evaluated from the problem itself and then to be stored. This is another advantage of the type of SAO algorithm advocated by Fleury for structural problems, as the use of second-order information greatly increases the memory storage requirements and so places additional limits on the size of problem that can be solved.

It appears that first-order SAO algorithms using dual solvers are now recognised as the state of the art for the solution of large-scale structural optimisation problems. Algorithms like CONLIN, due to Fleury and Braibant [4], and MMA, due to Svanberg [3], having found widespread use, particularly in the topology optimisation community. Bendsøe and Sigmund, for example, advocate the use of MMA in [8] as a generally applicable procedure to solve topology optimisation problems, while Duysinx and Bendsøe use CONLIN for their solution of large-scale stress-constrained weight minimisation problems in [84].

In this chapter we thus consider the solution of large-scale topology optimisation problems using the efficient separable SAO approach, and we briefly compare different approximation schemes for the construction of the subproblems. We accept that an efficient finite element solution package is available for the analysis component of the procedure. In our work we have used the FORTRAN-based finite element code EDSAP, written by Edward Wilson of the University of California at Berkeley and made available freely to academic researchers. This package is used chiefly to develop and test different finite elements for finite element analysis, and direct access is thus provided to the FEM source code.

For constrained optimisation, the size of the dual subproblems is dependent on the number of constraints considered in the construction of the primal approximate subproblems. If a step-size limitation or trust region is employed, or if sufficiently conservative approximations are used to ensure robust global convergence characteristics, then it is only necessary to include the active or near-active constraints when constructing the approximate subproblems. Limiting the number of constraints included reduces the dimensionality of the dual subproblems, which are easier and quicker to solve than larger ones (provided the conditioning of the dual is not adversely affected by constraint selection). Naturally, the criterion whereby the constraints are considered significant in the following subproblem is a relative determination. We briefly illustrate the effect of constraint selection.

Additionally, it is proposed that omitting terms from the Jacobian of the constraints when constructing the subproblems may prove advantageous during the solution of large-scale problems, since fewer elements need to be stored to define the subproblem. The idea of Jacobian filtering has been voiced previously [85], and it is here formally incorporated into the applied SAO procedure. Depending on the strategy used to select the 'significant' elements, this may result in a substantial decrease in storage requirements, which is useful if sparse implementations of the optimisation algorithms are used, and the resulting solution strategy may be more efficient. On the other hand, by omitting terms from the Jacobian the accuracy of the approximations is decreased. We test whether these effects are noteworthy.

The inclusion of local stress constraints in topology problems produces another complication quite apart from the large size of the resulting optimisation problems. This complication is commonly labelled the 'singularity problem', and stems from the observation that the feasible region in the stress-constrained problem may contain degenerate domains in which the global optimum for the problem is frequently located. Loosely, these degenerate domains are k-dimensional hyperplanes emanating from the 'bulk' of the n-dimensional feasible region and protruding into the infeasible space as infinitely thin slivers ($k < n$). The problem has been studied in the context of truss design, for example, by Kirsch [86], and by Cheng and Jiang [87], who show that the degenerate regions are a result of discontinuities in the stress constraints. An overview of the topic is presented in [88].

Several methods have been suggested with a view to making the search for local optima more tractable under these circumstances. Methods such as the introduction of smooth envelope functions (SEF) [89] and ε-relaxation [90] entail a modification of the constraint formulation that results in a broadening of the degenerate regions, making them n-dimensional. Such methods were introduced in the context of truss optimisation; seemingly, there has been less attention paid to the solution of planar and 3D problems with similar constraints. When planar problems are considered, similar methods are used to deal with the singularity problem; Duysinx and Bendsøe [84] use ε-relaxation, for example, while Bruggi [91] and Le *et al.* [92] introduce a particular form of stress interpolation for material in the relaxed continuous form of the problem.

Since Bruggi and Venini formally relate the ε-relaxation and stress interpolation approaches in [93], and the relationship between ε-relaxation and the use of SEF is pointed out in [88], these methods are all really aspects of the same idea, which is to allow increased stresses in elements that have near zero density[1]. Since our work follows mainly on the ideas set forth in [84], we too implement ε-relaxation in a form closely related to that of Cheng and Guo [90]. However, we introduce a numerical implementation that is contrary to what is suggested in [90], and applied in [84], but which yields good numerical results.

The progression of this chapter is as follows: The formulations for both the minimum weight and minimum compliance topology problems are discussed briefly in Section 9.3, as is the SIMP method for encouraging the generation of solid-void designs. Then, in Section 9.4, we describe the primal approximate subproblems that are used in this study, and the quadratic approximation strategies that are utilised in their construction. Thereafter, the definition of the dual subproblems is described. Two specific examples are given for different approximation schemes. In Section 9.5 the formulation of the local stress constraints is reviewed. We describe the calculation of the stress sensitivities that are required in the construction of the approximate subproblems, as well as the stress relaxation strategies that are employed in the generation of numerical results. Some numerical considerations are outlined briefly in Section 9.6, before the numerical results are presented in Section 9.7. The results are presented by comparing the various approximation strategies considered, and the effect of constraint selection and filtering of the Jacobian is discussed. We also compare two different stress relaxation strategies, and we illustrate the difference between the results gained from a standard minimum compliance problem, the stress-constrained minimum compliance problem and the stress-constrained minimum weight problem. Lastly, results for large mesh refinements are given, before concluding remarks and thoughts for future research are profferred in Section 9.8.

9.3 Problem formulation

The material distribution problems discussed in this chapter are fundamentally of the form represented by equation (2.1). The particular distribution of an isotropic material is sought within a given domain[2], constrained and loaded in some way, such that one or other structural objective is minimised and one or multiple constraints on the structural responses are satisfied. Of key im-

[1] By 'density' we here mean the material occupancy of an element, namely x_i for element i. The material property ρ will be referred to as the 'mass density' where necessary.

[2] Only planar problems are considered.

portance is the stipulation that, at any point in the domain, the material in question may either be present or absent, but no other states are physically meaningful. When the design domain is discretised so that the material distribution function is represented by a vector of finite length, each element may assume only the binary values 0 or 1. As was explained in Section 2.1, the binary requirements on the variables in such problems is usually relaxed in order to facilitate the use of efficient continuous nonlinear programming algorithms to search for the optima. Variables having values intermediate between 0 and 1 are then penalised in the relaxed problem in an effort to find purely $[0, 1]$ solutions, which are then solutions to the intended discretised but unrelaxed problem. Despite penalisation, purely solid-void designs are seldom produced, which raises the following conceptual questions. Firstly, how should intermediate values of the discretised material distribution function be interpreted in the context of solid-void isotropic topology design? Secondly, how should two different solutions, neither purely binary, be compared?

The standard discretised and relaxed form of the weight minimisation problem with displacement and/or local stress constraints was introduced in Section 2.1.3 as equation (2.13). The form of the problem defined by (2.13) results from the consideration of both truss-like structures as well as spatially discretised representations of continuum (planar and 3D) structures. One important difference, however, is that truss problems are usually interpreted as sizing problems, in which case there is no underlying $[0, 1]$ problem. Since the design variables in this case represent cross-sectional areas for truss elements (usually), the variables are not penalised to produce a binary design, and any value that the variable assumes (at the optimum) in between the defined upper and lower bound constraints is physically meaningful. The truss sizing problems discussed in [86], [87] and [90], for example, are of this type.

As was discussed in Section 2.2.2, displacement and stress constraints have the same basic form: both are reciprocal in the design variables for statically indeterminate problems. Only stress constraints will be considered in the current chapter, but the formulation and method of solution discussed below are obviously valid if displacement constraints are also present. Without loss of generality, it is assumed that the structural domain is discretised by the finite element method using a regular mesh of n elements (each element being square, undistorted and identical in size), and so we here commence by stating the weight minimisation problem as follows:

Minimum weight topology problem P_W

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f_0(\boldsymbol{x}) = \sum_{i=1}^{n} \rho_i \nu_i x_i \\ \text{subject to} \; & f_j(\boldsymbol{x}) = \sigma_j^m \leq \bar{\sigma} \qquad j = 1, 2, \cdots, n, \\ & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{w}, \\ & 0 < \check{x} \leq x_i \leq \hat{x} \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{9.1}$$

The symbols $\check{x}$ and $\hat{x}$ represent, respectively, the lower and upper bounds on x_i, the density of element i. We assume that these bounds are the same for all elements. The optimal distribution of a single isotropic material is sought within the defined domain such that the mass of the structure is minimised and the defined (static) loads $\boldsymbol{w}$ are supported without risk of static failure occurring. Hence, ν_i in the objective function f_0 represents the elemental volume, and we assume a 2D design domain that has unit thickness. The symbols $\boldsymbol{K}$ and $\boldsymbol{q}$ denote the global assembled finite element stiffness matrix and the global vector of nodal displacements respectively, and the constraints f_j

represent upper bounds on a chosen stress-related failure criterion. The stress constraints are pointwise (local) in nature, so in the spatially discretised problem each element must satisfy a constraint on its internal stresses. For our purposes, a limit on the equivalent von Mises stress, calculated at the element centroids, will be used. Other choices are of course also possible, and the symbol σ_j^m thus represents the desired stress measure, whereas $\bar{\sigma}$ denotes the limiting value for said criterion.

Since we discretise the design domain using a regular mesh, each element being square and identical in size, and since we consider only the distribution of a single isotropic material with uniform mass density, we replace the objective function in (9.1) with

$$f_0(\boldsymbol{x}) = \sum_{i=1}^{n} x_i \,,$$

which serves the same purpose. The standard, or 'classical', discretised minimum compliance problem, in which the solid-void material distribution is sought that minimises the structural compliance subject to a single constraint on the allowable structural volume, was introduced in Section 2.1.2. With the addition of local stress constraints, the problem may be written as

Minimum compliance topology problem P_C

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f_0(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \\ \text{subject to} \; & f_j(\boldsymbol{x}) = \sigma_j^m \leq \bar{\sigma} \qquad\qquad j = 1, 2, \cdots, n, \\ & f_{n+1}(\boldsymbol{x}) = \frac{1}{\nu_0} \sum_{i=1}^{n} \nu_i x_i \leq \bar{\nu}, \\ & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{w}, \\ & 0 < \check{x} \leq x_i \leq \hat{x} \qquad\qquad i \; = 1, 2, \cdots, n. \end{aligned} \tag{9.2}$$

The subscripts i in the objective function indicate elemental quantities, while the ν_0 and $\bar{\nu}$ in the volume constraint are, respectively, the total volume of the design domain and a limiting value for the volume of material within the domain. From the finite element equations, for a structure in static equilibrium we have

$$\boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} = \boldsymbol{q}^T \boldsymbol{w} \,. \tag{9.3}$$

For linear elastic structures, the right-hand side of this equation corresponds to twice the work done by the applied loads $\boldsymbol{w}$ in producing deformation $\boldsymbol{q}$. The left-hand side is equivalent to twice the internal strain energy within the structure, and the equation expresses the requirement that these be balanced at static equilibrium.

The 'classical' minimum compliance problem takes no account of the strength of the material in searching for optimal topologies, so the solutions derived may not be useful for physical design because the local stresses at points in the optimal topologies may exceed the maximum stress that the material can support. One may seek a topology that will not fail by increasing the allowable structural volume, but material will not necessarily be added only in the vicinity of the highly stressed areas. By employing stress constraints, the algorithm is encouraged to distribute the allowable material in a way that reflects, first, the necessity to maintain structural integrity, with the

minimisation of compliance being subordinate to this necessity. This in itself restricts the solution space of the compliance problem.

Since the stress constraints are local in nature, one constraint is associated with each element in the finite element mesh. The optimisation problem therefore requires the consideration of an n-dimensional problem with at least n constraints. In contrast with the 'classical' minimum compliance problem, which has n primal variables but only a few constraints (a volume constraint and perhaps a perimeter constraint, for example), problems with local stress constraints scale very badly with n in terms of the effort required to solve them.

Due to the fact that there often are much fewer constraints than primal variables, these structural problems are frequently solved using a dual method. Since the dual is defined in the space of the Lagrange multipliers associated with the constraints, it is much smaller than the primal. Therefore, operating on the dual facilitates the efficient solution of what would otherwise be an extremely large and challenging problem in the primal space. This advantage is diminished when local stress constraints are present, because the dual becomes very large as well. Indeed, while it is common to see classical compliance problems solved with several thousand design variables (see for example the results in Chapter 5, which can be solved without prodigious effort), it is quite rare to see examples of similarly large stress-constrained problems. As an example, Duysinx and Bendsøe, in their influential paper on the subject [84], use CONLIN to solve 2D weight minimisation problems discretised by a mesh of 60×20 elements. Their test problems therefore have 1200 primal variables as well as 1200 stress constraints. We are unaware of any larger test problems incorporating local stress constraints having been cited in the literature to date.

Both formulations, P_W and P_C, are continuous relaxed forms (in the design variables $\boldsymbol{x}$) of what should ideally be discrete problems, since solid-void material distributions are sought. The relaxation is introduced into the material description via the elasticity matrix associated with element i, as

$$\boldsymbol{C}_i\left(x_i\right) = x_i \boldsymbol{C}_0\,. \tag{9.4}$$

Hence, the material properties for element i, embodied by the elasticity matrix $\boldsymbol{C}_i$, are scaled linearly with x_i relative to the elasticity matrix of the solid isotropic material $\boldsymbol{C}_0$. As we have mentioned, relaxation is employed so that efficient methods of continuous nonlinear programming (NLP) may be used to solve the optimisation problems, avoiding the usually more demanding methods for integer programming, but then something else must be done to encourage the generation of solid-void designs. The method used here for both P_W and P_C is a method of penalising intermediate-density material, known as SIMP (for 'solid isotropic microstructure with penalisation'), also introduced briefly in Section 2.1.2.

9.3.1 SIMP

Suggested independently by Bendsøe [18] and Rozvany and Zhou [19], the SIMP approach introduces a penalty into the linearised material description presented above, by modifying it so that

$$\boldsymbol{C}_i\left(x_i\right) = x_i^p \boldsymbol{C}_0 \qquad p > 1. \tag{9.5}$$

We will use the standard value for the penalisation, $p = 3$. This can be interpreted as a material law, describing the material properties of elements with 'densities' intermediate between 0 and

1. By introducing the penalty parameter p, the elements of $\boldsymbol{C}_i$ are decreased relative to the linear scaling law (9.4) for non-binary values of x_i, but $\boldsymbol{C}_i = \boldsymbol{C}_0$ for $x_i = 0$ and $x_i = 1$. An element with $0 < x < 1$ is "uneconomical" in the classical compliance problem because, as described in [8], "the stiffness obtained is small compared to the cost (volume) of the material". This material penalisation affects the compliance objective directly, which in the relaxed penalised case is

$$f_0(\boldsymbol{x}) = \boldsymbol{q}^T \boldsymbol{K} \boldsymbol{q} = \sum_{i=1}^{n} x_i^p \, \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \, . \tag{9.6}$$

Although superficially it looks like the compliance would decrease for values of $x_i < 1$ relative to $x_i = 1$, due to the implicit dependence of $\boldsymbol{q}$ on $\boldsymbol{x}$ through the finite element equilibrium equations, the sensitivity of the compliance objective to small changes in the design variables can be shown to be

$$\frac{\partial f_0}{\partial x_i} = -p x_i^{p-1} \, \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \, . \tag{9.7}$$

Thus, given some point $\boldsymbol{x}^{\{k\}}$in the design space, to first order the change in compliance achieved by a small increase in x_i, namely $\triangle x_i$, is

$$\triangle f_0 = \left(\frac{\partial f_0}{\partial x_i} \right)^{\{k\}} \triangle x_i \; = \; -p \left(x_i^{\{k\}} \right)^{p-1} \left(\boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i \right)^{\{k\}} \triangle x_i \, .$$

Clearly, in the relaxed but unpenalised case, $\triangle f_0$ is not explicitly dependent on $x_i^{\{k\}}$, whereas in the penalised case the decrease in $\triangle f_0$ is greater when $x_i^{\{k\}} \approx 1$ than when $x_i^{\{k\}} < 1$. This would tend to indicate that, for penalised compliance problems, the minima $\boldsymbol{x}^*$ are characterised by the prescribed volume being distributed efficiently amongst elements for which $x_i \approx 1$.

In the weight minimisation problem the penalisation does not enter into the objective function directly. The local stresses, however, are still dependent on the material penalisation, which then provides the propensity for generating $[0, 1]$ solutions. As discussed in Section 2.1.3, with

$$\sigma_{ij} = x_i^p C_0 \epsilon_{ij} \tag{9.8}$$

for a given strain ϵ_{ij}, if a stress constraint $\sigma_i^m = f\left(\sigma_{ij}\right)$ is active, the value of x_i would be higher for $p > 1$ than for $p = 1$. The minimisation of structural weight ensures that the stress constraints become active.

9.4 The dual SAO procedure

A sequential approximate optimisation procedure is employed for the iterative solution of problems P_W and P_C. During each iteration k, the original problem, being either P_W or P_C, is replaced by an explicit surrugate subproblem $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$, which is derived as an approximation to the original problem at the current iteration point $\boldsymbol{x}^{\{k\}}$. The solution to the subproblem yields the approximate $\boldsymbol{x}^{\{k+1\}}$, at which the following subproblem is constructed. Under certain conditions, such as continuity and convexity of the subproblems and the imposition of a method to ensure global convergence (like the use of CCSA approximations [6] or trust regions [24]), the sequence of iterates $\boldsymbol{x}^{\{k+1\}}$can

be shown to converge to a KKT point of the original problem as k increases (provided a sensitivity filter is not used in the problem formulation). Thus, during each iteration, the solution of the following problem is considered

Explicit approximate subproblem $\boldsymbol{P}_{\mathrm{SUB}}^{\{k\}}$

$$\begin{aligned} &\min_{\boldsymbol{x}} \ \tilde{f}_0(\boldsymbol{x}) \\ &\text{subject to} \ \ \tilde{f}_j(\boldsymbol{x}) \leq 0 \qquad\qquad j = 1, 2, \cdots, m, \\ &\qquad\qquad 0 < \check{x} \leq x_i \leq \hat{x} \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{9.9}$$

where $m = n$ when P_W is considered, whereas $m = n + 1$ when P_C is solved. The tildes over $\tilde{f}_0$ and $\tilde{f}_j$ denote function approximations. The various SAO algorithms are distinguished by the particular form of function approximation(s) chosen to construct the subproblems, as well as the method chosen to solve the subproblems.

While the approximate subproblem can be solved using any applicable method for constrained nonlinear programming, we utilise the dual solution method. In the field of structural optimisation there are various methods available that utilise a sequential approximate optimisation procedure in which the subproblems are constructed from strictly convex and separable functions, and in which a dual method of solution is used that relies on a definition of the dual problem due to Falk [2]. Examples are the method of moving asymptotes, due to Svanberg [3], and CONLIN, due to Fleury and Braibant [4]. Such methods were popularised originally, and formally linked to the widely used OC methods, by Fleury [1, 28], and subsequently also by Groenwold and Etman [43]. While Fleury specifically considered the weight minimisation problem in the cited references, Groenwold and Etman regarded the minimum compliance problem.

9.4.1 Approximate subproblem

In the consideration of P_W, the objective function may be represented exactly, namely $\tilde{f}_0 = f_0$, and in the consideration of P_C the volume constraint may be represented exactly ($\tilde{f}_{n+1} = f_{n+1}$), since both are linear in the design variables. Equivalently, they can be written as the first-order Taylor series expansion

$$\tilde{f}\left(\boldsymbol{x}\right) = f\left(\boldsymbol{x}\right) + \sum_{i=1}^{n} \left(x_i - x_i^{\{k\}}\right) \left(\frac{\partial f}{\partial x_i}\right)^{\{k\}}. \tag{9.10}$$

The particular form of function approximation favoured herein to approximate the stress constraints for both P_W and P_C, as well as the compliance objective for P_C, in the construction of the subproblems, is the quadratic approximation previously developed by Groenwold *et al.* for structural topology optimisation [33]. This is a separable quadratic approximation, in which the off-diagonal terms in the Hessian matrix are all zero; it is given as

$$\tilde{f}\left(\boldsymbol{x}\right) = f\left(\boldsymbol{x}\right) + \sum_{i=1}^{n} \left(x_i - x_i^{\{k\}}\right) \left(\frac{\partial f}{\partial x_i}\right)^{\{k\}} + \frac{1}{2} \sum_{i=1}^{n} c_i^{\{k\}} \left(x_i - x_i^{\{k\}}\right)^2. \tag{9.11}$$

The curvatures $c_i^{\{k\}}$ are chosen very carefully to ensure that the reciprocal-like behaviour of many of the dependencies in structural problems can be well represented. Thus, the constants $c_i^{\{k\}}$ are derived from a consideration of the separable first-order exponential approximation (5.13), expressed here again for convenience:

$$\tilde{f}_E(\boldsymbol{x}) = f(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left[\left(\frac{x_i}{x_i^{\{k\}}} \right)^{r_i^{\{k\}}} - 1 \right] \left(\frac{x_i^{\{k\}}}{r_i^{\{k\}}} \right) \left(\frac{\partial f}{\partial x_i} \right)^{\{k\}}. \tag{9.12}$$

The curvatures are found by enforcing the condition that (9.11) is the quadratic approximation to (9.12) at the point $\boldsymbol{x}^{\{k\}}$, the curvatures in the quadratic function being the second-order partial derivatives of the exponential function at the point $\boldsymbol{x}^{\{k\}}$, which results in

$$c_i^{\{k\}} = \frac{\partial^2 \tilde{f}_E}{\partial x_i^2} \left(\boldsymbol{x}^{\{k\}} \right) = \frac{r_i^{\{k\}} - 1}{x_i^{\{k\}}} \left(\frac{\partial f}{\partial x_i} \right)^{\{k\}}. \tag{9.13}$$

The exponents $r_i^{\{k\}}$ are calculated from historic data by enforcing

$$\nabla \tilde{f}_E \left(\boldsymbol{x}^{\{k-1\}} \right) = \nabla f \left(\boldsymbol{x}^{\{k-1\}} \right),$$

in which $f(\boldsymbol{x})$ is the true function being approximated. From this, the exponents are derived as

$$r_i^{\{k\}} = 1 + \frac{\ln \left\{ \left(\frac{\partial f}{\partial x_i} \right)^{\{k-1\}} / \left(\frac{\partial f}{\partial x_i} \right)^{\{k\}} \right\}}{\ln \left\{ x_i^{\{k-1\}} / x_i^{\{k\}} \right\}}. \tag{9.14}$$

We call the resulting approximation T2:E; it is a quadratic approximation to the exponential approximation. This function is strictly convex when $\partial f / \partial x_i < 0$ and $r_i^{\{k\}} < 1$, as is the case when the compliance objective f_0 in P_C is approximated. For the stress constraints considered in this chapter, the partial derivatives $\partial f_j / \partial x_i$ may be positive or negative. Since it is desired that the quadratic approximation (9.11) be strictly convex, we replace (9.13) by

$$c_i^{\{k\}} = \frac{\partial^2 \tilde{f}_E}{\partial x_i^2} \left(\boldsymbol{x}^{\{k\}} \right) = -\frac{r_i^{\{k\}} - 1}{x_i^{\{k\}}} \left| \frac{\partial f}{\partial x_i} \right|^{\{k\}} \tag{9.15}$$

and restrict $r_i^{\{k\}}$ by enforcing $r_i^{\{k\}} < 0$, which serves for both the compliance objective and the stress constraints. Finally, if we set $r_i^{\{k\}} = -1$ for all i, instead of applying (9.14), then as a special case we generate the quadratic approximation to the reciprocal approximation, which we denote T2:R and for which

$$c_i^{\{k\}} = \frac{2}{x_i^{\{k\}}} \left| \frac{\partial f}{\partial x_i} \right|^{\{k\}}. \tag{9.16}$$

In the same manner, strictly convex separable quadratic approximations may be derived for many of the other popular forms of function approximations used in SAO, such as CONLIN, MMA and the TANA approximations proposed by Grandhi and his collaborators [67, 68]. For further details, the reader is referred to our previous efforts [33].

In Section 9.7 a brief comparison is carried out using a weight minimisation problem with a coarse mesh to assess which of CONLIN, T2:CONLIN, T2:R, T2:E or T2:MMA can be used most efficiently to solve the problems considered. Superficially, we find little difference between them; we continue with T2:R to investigate the effect of other parameters, and then with T2:CONLIN for the solution of larger problems, as the solution using T2:CONLIN appears marginally more efficient. These approximation strategies have in common that no historic information is required for the definition of the associated subproblems, and we are also not faced with the additional complexity of adjusting the asymptotes for MMA.

9.4.2 Dual solution procedure

The use of the dual method allows the subproblems to be solved by considering instead a dual subproblem, defined in the space of the Lagrange multipliers associated with the constraint functions in the primal subproblem. If the primal subproblem is strictly convex and continuous, it can be shown that the maximum of the dual subproblem corresponds to the solution, the minimiser, of the primal. The advantage of using the dual formulation is that the dual problem has a very simple structure. In Reference [2], Falk showed that the dual problem is concave. Additionally, it is simply constrained, the only constraints being non-negativity constraints on the dual variables. The gradients of the dual, which are invariably required by the NLP technique chosen to accomplish the dual maximisation, are also straightforward to evaluate. They are simply the values of the primal constraints, evaluated at the primal coordinates corresponding to a given point in the dual space, said correspondence being dictated by the primal-dual relationships. Thus, there are several reasons why a dual solution strategy might be preferred. Most importantly, however, is that the number of constraints in the primal problem is frequently far less than the number of primal variables, so the dual typically is much smaller than the primal. In the case of the classical minimum compliance problem, for instance, which has only a single constraint on the allowable volume of the design, the dual is one-dimensional. Being concave, it is extremely straightforward to optimise and so to identify the corresponding optimum of the original primal subproblem.

When stress constraints are present in the problem formulation, as is the case for both P_W and P_C, the primary advantage afforded by the use of the dual method, namely its (typically) small dimensionality, is (partially) eradicated. One is still able to take advantage of this characteristic of the dual formulation by considering in each iteration only the active constraints, and possibly also the most critical inactive constraints, in the definition of the primal subproblem. Even in the event that the dual is of the same size as the primal, it still retains its simple structure, which in itself may be reason enough to attempt to solve the dual rather than the primal. It must be remembered that an additional computational cost is incurred in evaluating the primal-dual relationships, relative to primal solution algorithms. Also, although it is concave, the dual can be badly scaled, so despite the fact that the dual is only simply constrained its maximisation is not necessarily trivial, particularly if there are a large number of active constraints. For the stress-constrained topology problems presented herein, we retain the used of the dual method of solution.

The dual problem is derived from the Lagrangian function, which is defined as follows in terms of

the functions involved in the SAO subproblems:

$$\mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) = \tilde{f}_0\left(\boldsymbol{x}\right) + \sum_{j=1}^{m} \lambda_j \tilde{f}_j\left(\boldsymbol{x}\right).$$

The KKT point of the primal problem is identified with the saddle point of the Lagrangian function (which is unique by construction, due to the convexity of the primal subproblem, provided that the subproblem has a feasible solution) and may be found by maximising the dual function, defined according to Falk by

$$\begin{aligned} \gamma(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \\ \text{subject to} \quad \boldsymbol{x} \in \mathcal{C}\,, \\ \text{and} \quad \boldsymbol{\lambda} \geq 0\,. \end{aligned} \tag{9.17}$$

When separable approximations such as (9.10) and (9.11) are used to construct the primal subproblems, the corresponding Lagrangian function is separable in the primal variables x_i. The minimisation with respect to $\boldsymbol{x}$ in (9.17) can then be carried out as n separate minimisations with respect to x_i, which yields a set of expressions $x_i\left(\boldsymbol{\lambda}\right)$ that define the relationship between the primal and dual variables.

The set $\mathcal{C}$ typically, and certainly in our case, consists of the box constraints on the primal variables $\check{x} \leq x_i \leq \hat{x}$, which are then not included as constraint functions $\tilde{f}_j$ in the definition of the Lagrangian. Given the strict convexity of $\mathcal{L}$ with respect to the primal variables $\boldsymbol{x}$, if $\mathcal{L}\left(x_i, \boldsymbol{\lambda}\right)$ possesses a stationary point on the interval $\check{x} \leq x_i \leq \hat{x}$, then the minimum of $\mathcal{L}$ with respect to x_i, namely

$$\arg\min_{x_i} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right),$$

can be located using the stationarity condition as the solution of

$$\frac{\partial}{\partial x_i}\mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) = 0\,.$$

Otherwise, the minimiser will be located either at $\check{x}$ or $\hat{x}$. This is reflected in the conditional form of the primal-dual relationships given below, where two explicit examples are given for constructing the dual for two particular primal approximate subproblems, T2:R and T2:CONLIN. Note that we introduce the notation $\mathcal{A}$ to designate the set of active and critical constraints used to define the primal subproblem.

The dual problem for $\mathrm{P_W}$ using T2:R

When employing T2:R to build the approximate subproblems for the weight minimisation problem, the objective function reduces to (9.10), while the constraints are represented as (9.11) with the c_i given by (9.16). The dual is

$$\begin{aligned} \gamma\left(\boldsymbol{\lambda}\right) = f_0\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i\left(\boldsymbol{\lambda}\right) - x_i^{\{k\}}\right)\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \\ \sum_{j\in\mathcal{A}} \lambda_j \left(f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i\left(\boldsymbol{\lambda}\right) - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n} c_{ji}^{\{k\}}\left(x_i\left(\boldsymbol{\lambda}\right) - x_i^{\{k\}}\right)^2 \right). \end{aligned}$$

Applying the stationary condition, and with $\partial f_0^{\{k\}}/\partial x_i = 1$, we obtain

$$\beta_i(\boldsymbol{\lambda}) = x_i^{\{k\}} - \left(\sum_{j\in\mathcal{A}} \lambda_j c_{ji}^{\{k\}}\right)^{-1} \left(1 + \sum_{j\in\mathcal{A}} \lambda_j \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right), \tag{9.18}$$

so that the primal-dual relationships can be expressed as

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i(\boldsymbol{\lambda}) & \text{if} \quad \check{x} < \beta_i(\boldsymbol{\lambda}) < \hat{x}\,, \\ \check{x} & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \le \check{x}\,, \\ \hat{x} & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \ge \hat{x}\,. \end{cases} \tag{9.19}$$

The dual problem for $\mathbf{P_C}$ using T2:R

For the compliance minimisation problem, the objective function as well as the j stress constraints, $j = 1, 2, \cdots, n$, are all described by (9.11), with the c_i again given by (9.16). The volume constraint ($j = n+1$) is linear, so the curvatures in the quadratic approximation fall away and the constraint is represented by (9.10). The compliance objective has negative partial derivatives everywhere, with the result that the volume constraint is always active at the solution of every subproblem. Hence we denote by $\mathcal{A}$ the set of active and critical stress constraints, and explicitly include the volume constraint in the following equation for the dual:

$$\begin{aligned} \gamma(\boldsymbol{\lambda}) = f_0\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right) \left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n} c_{0i}^{\{k\}} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)^2 + \\ \sum_{j\in\mathcal{A}} \lambda_j \left(f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right) \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n} c_{ji}^{\{k\}} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)^2\right) + \\ \lambda_{n+1}\left(f_{n+1}\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n} \left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right) \left(\frac{\partial f_{n+1}}{\partial x_i}\right)^{\{k\}}\right). \end{aligned}$$

In this case (with $\partial f_{n+1}^{\{k\}}/\partial x_i = 1$),

$$\beta_i(\boldsymbol{\lambda}) = x_i^{\{k\}} - \left(c_{0i} + \sum_{j\in\mathcal{A}} \lambda_j c_{ji}^{\{k\}}\right)^{-1} \left(\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \sum_{j\in\mathcal{A}} \lambda_j \left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \lambda_{n+1}\right) \tag{9.20}$$

and

$$x_i(\boldsymbol{\lambda}) = \begin{cases} \beta_i(\boldsymbol{\lambda}) & \text{if} \quad \check{x} < \beta_i(\boldsymbol{\lambda}) < \hat{x}\,, \\ \check{x} & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \le \check{x}\,, \\ \hat{x} & \text{if} \quad \beta_i(\boldsymbol{\lambda}) \ge \hat{x}\,. \end{cases} \tag{9.21}$$

The dual problem for $\mathbf{P_W}$ using T2:CONLIN

Instead of directly representing the constraints by (9.11), we respect the method of mixed variables underlying the CONLIN algorithm when generating the subproblems for $\mathbf{P_W}$. For each constraint function f_j, if

$$\frac{\partial f_j}{\partial x_i} < 0\,,$$

the corresponding dependence of the approximation on x_i is given by

$$\tilde{d}_{ji}\left(x_i\right) = \left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}c_i^{\{k\}}\left(x_i - x_i^{\{k\}}\right)^2, \tag{9.22}$$

with the c_i still given by (9.16). We define the set Q_j for f_j, which contains all indices i for which the above holds. On the other hand, if

$$\frac{\partial f_j}{\partial x_i} > 0\,,$$

then the corresponding dependence of the approximation on x_i is given by the linear term

$$\tilde{d}_{ji}\left(x_i\right) = \left(x_i - x_i^{\{k\}}\right)\left(\frac{\partial f}{\partial x_i}\right)^{\{k\}}. \tag{9.23}$$

The dual of the subproblem for T2:CONLIN can therefore be expressed as

$$\gamma\left(\boldsymbol{\lambda}\right) = f_0\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i\left(\boldsymbol{\lambda}\right) - x_i^{\{k\}}\right)\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \sum_{j\in\mathcal{A}}\lambda_j\left(f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i\left(\boldsymbol{\lambda}\right) - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i\in Q_j}c_{ji}^{\{k\}}\left(x_i\left(\boldsymbol{\lambda}\right) - x_i^{\{k\}}\right)^2\right),$$

and we thus obtain the $\beta_i\left(\boldsymbol{\lambda}\right)$ for (9.18) as

$$\beta_i\left(\boldsymbol{\lambda}\right) = \frac{x_i^{\{k\}}\sum_{j\in\mathcal{A}}\lambda_j\left(c_{ji}\right)_{Q_j}^{\{k\}} - \left(1 + \sum_{j\in\mathcal{A}}\lambda_j\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}}\right)}{\sum_{j\in\mathcal{A}}\lambda_j\left(c_{ji}\right)_{Q_j}^{\{k\}}}, \tag{9.24}$$

where the term $\left(c_{ji}\right)_{Q_j}^{\{k\}}$ is interpreted as

$$\left(c_{ji}\right)_{Q_j}^{\{k\}} = \begin{cases} c_{ji} & \text{if} \quad i \in Q_j\,, \\ 0 & \text{otherwise.} \end{cases}$$

The primal variables are again determined from (9.19). It is now possible, however, that the denominator in (9.24) is zero if the set Q_j is empty for all j. In this case the Lagrangian is strictly linear in the variable x_i, with gradient g_i, and the design update $x_i\left(\boldsymbol{\lambda}\right)$ will correspond to either $\check{x}$ or $\hat{x}$, to be determined by the sign of g_i. Alternatively, it is numerically expedient simply to add quadratic terms with very small curvatures c_{ji} to all terms $\tilde{d}_{ij}$ that do not belong to Q_j in (9.23). The resulting $\beta_i\left(\boldsymbol{\lambda}\right)$ would again be given by (9.18).

The dual problem for $\mathbf{P_C}$ using T2:CONLIN

Applying the same method of mixed variables as described above to the constraints in $\mathbf{P_C}$, the expression for the dual becomes

$$\gamma(\boldsymbol{\lambda}) = f_0\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i=1}^{n} c_{0i}^{\{k\}}\left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)^2 + \sum_{j\in\mathcal{A}}\lambda_j\left(f_j\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \frac{1}{2}\sum_{i\in Q_j} c_{ji}^{\{k\}}\left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)^2\right) + \lambda_{n+1}\left(f_{n+1}\left(\boldsymbol{x}^{\{k\}}\right) + \sum_{i=1}^{n}\left(x_i(\boldsymbol{\lambda}) - x_i^{\{k\}}\right)\left(\frac{\partial f_{n+1}}{\partial x_i}\right)^{\{k\}}\right).$$

From the stationary condition,

$$\beta_i(\boldsymbol{\lambda}) = x_i^{\{k\}} - \left(c_{0i} + \sum_{j\in\mathcal{A}}\lambda_j\left(c_{ji}\right)_{Q_j}^{\{k\}}\right)^{-1}\left(\left(\frac{\partial f_0}{\partial x_i}\right)^{\{k\}} + \sum_{j\in\mathcal{A}}\lambda_j\left(\frac{\partial f_j}{\partial x_i}\right)^{\{k\}} + \lambda_{n+1}\right), \quad (9.25)$$

and $\boldsymbol{x}$ is still given by (9.21). The denominator in (9.25) cannot be zero due to the presence of the c_{0i}.

9.5 Local stress constraints

9.5.1 Constraint formulation

Consider the simple one-dimensional textbook example illustrated in Figure 9.1, in which two parallel bars of the same material are acted on by an applied load F. Each bar has variable cross-sectional area A_i, and the force internal to each bar is denoted P_i. The bars have equal deformation δ, the problem being a one-dimensional illustration. It is straightforward to show that the free-end

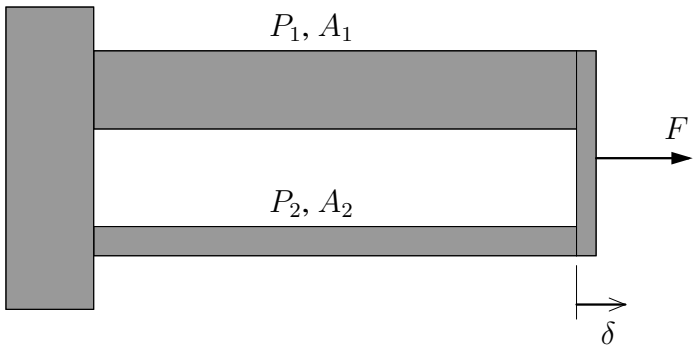


Figure 9.1: A one-dimensional example illustrating the non-zero stress in a truss element as its area tends to zero.

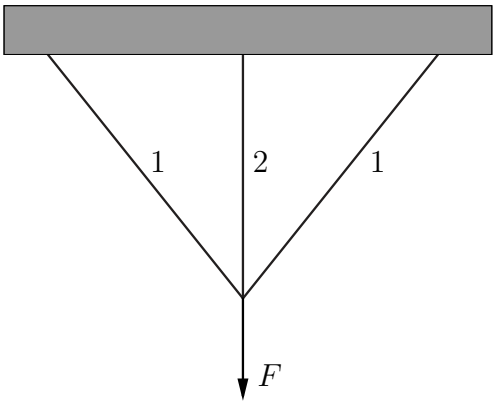


Figure 9.2: A three element truss example of stress discontinuity.

displacement is given as

$$\delta = \frac{FL}{E\,(A_1 + A_2)}$$

in terms of the variable bar areas, where L denotes the length of the bars and E represents Young's modulus for the material. In terms of the displacement, the internal loads are

$$P_i = \frac{\delta A_i E}{L}\,,$$

from which the stress in each member is calculated in terms of the applied load as

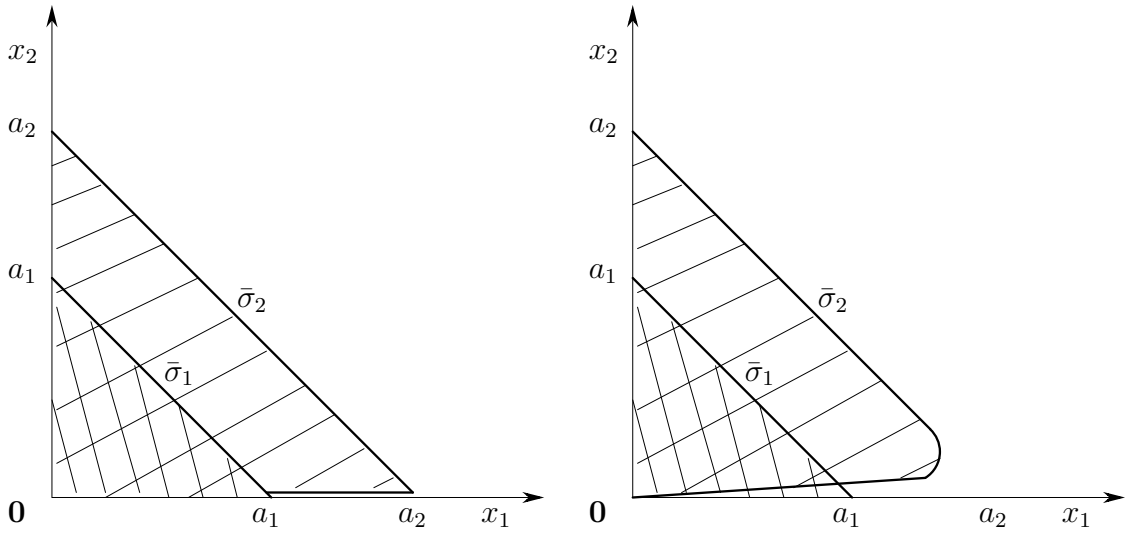
$$\sigma_i = \frac{F}{(A_1 + A_2)}\,.$$

Notice that if A_2 is kept constant and A_1 is reduced towards zero, although the internal force P_1 tends towards zero the stress in element 1 tends towards a finite value. The same behaviour is observed in more complex truss problems, as well as in continuum problems, and may prevent the removal of elements whose areas (in truss examples) or 'densities' (in discretised continuum problems) are on their lower bounds.

Take, for instance, the illustrative truss problem discussed in [90], the salient features of which are depicted in Figure 9.2. As a function of the truss cross-sections x_i, the stress constraints take the form

$$\bar{\sigma}_i = \frac{a_i}{x_1 + x_2} \leq 1\,, \tag{9.26}$$

where the a_i are constants. The feasible region is graphically represented by the un-hatched region in Figure 9.3(a). Once again it is evident that the stress in each element tends to a non-zero value as its cross-section tends to zero. Consider, for instance, the point $(a_2, 0)$ at which the stress constraint for element 2 is active, but at which element 2 has zero cross-section. The element therefore makes no contribution to the internal energy of the structure, but an algorithm would be prevented from approaching the more optimal point $(a_1, 0)$ because the stress constraint for element 2 would apparently be violated. The result is unrealistic, of course, but algorithmically these elements are difficult to identify and remove in a consistent way.



(a) Feasible region: original constraints

(b) Feasible region: constraint 2 relaxed

Figure 9.3: The feasible regions defined by the stress constraints for the three-element truss example.

With the stress constraints defined as above, the line joining $(a_1, 0)$ and $(a_2, 0)$, which represents the removal of element 2 from the structure, is excluded from the feasible region. However, the constraints may be reformulated as

$$x_i \left(a_i - x_1 - x_2\right) \leq 0 \,, \tag{9.27}$$

which provides an equivalent representation of the feasible region, except that the line joining points $(0, 0)$ and $(a_2, 0)$ is now feasible with respect to constraint 2. Similarly, the line joining points $(0, 0)$ and $(0, a_1)$ is now feasible with respect to constraint 1. Physically, these reformulated constraints represent the fact that the stress measure in a non-existent element should not contribute to the infeasibility of the design. Mathematically, such a reformulation unfortunately also introduces the point $(0, 0)$ into the feasible region if all the constraints are relaxed, which is spurious. To complicate matters, it is usually necessary to set a finite lower bound on an element cross-section or density to prevent numerical ill-conditioning in the analysis of the structure.

Algorithmically, this reformulation of the stress constraints doesn't help matters much, because, although the lines along which $x_i = 0$ are now added to the feasible regions of their respective constraints i, these lines are infinitely thin, and thus are virtually inaccessible to the optimiser. The method of ε-relaxation, introduced by Cheng and Guo [90], allows sizing algorithms to approach the singular optima such as $(a_1, 0)$ in the example above, and thus make it possible for sizing algorithms to be used in the topology optimisation of truss problems. The method works by 'relaxing' the stress constraints for structural elements close to their lower bounds, which basically allows the stresses (as calculated by (9.26)) in these elements to climb well above the limiting values set by the failure criterion. Graphically, the relaxation 'opens up' the feasible domain, as depicted in Figure 9.3(b), to allow an optimiser to approach the singular optima. The relaxed form of the constraints (9.27), with $\varepsilon > 0$, is

$$x_i \left(a_i - x_1 - x_2\right) \leq \varepsilon \,, \tag{9.28}$$

and the value of ε controls the extent to which the feasible domain is opened up, and thereby also the amount by which the stresses in the elements may exceed the limit set by the failure criterion.

The form of stress relaxation suggested by Cheng and Guo in [90], and adopted by Duysinx and Bendsøe in [84], is

$$x_i \left(\frac{\sigma_i^m}{\bar{\sigma}} - 1 \right) \leq \varepsilon \qquad \text{with} \quad \varepsilon^2 = \check{x} \leq x_i \,. \tag{9.29}$$

In the above, σ_i^m represents the stress measure calculated for element i demanded by the particular failure criterion in use, whereas $\bar{\sigma}$ denotes the limit on said stress measure. Additionally, $\check{x}$ is the lower bound on the elements of $\boldsymbol{x}$. Equation (9.29) asserts a fixed relationship between ε and $\check{x}$. Hence, if ε is changed during the optimisation process, $\check{x}$ is modified concordantly.

Cheng and Guo show that, with this relaxation, the sequence of problems defined by non-zero ε (and their KKT points) converges to the original unrelaxed problem (and its KKT points) as $\varepsilon \to 0$. While the unrelaxed problem has a degenerate feasible domain, none of the relaxed problems do. Thus, applying a continuation strategy on ε enables an optimiser to converge towards a singular optimum of the original problem, which would be inaccessible without ε-relaxation. In Reference [84], ε is reduced to a lower limit of 0.01, the corresponding value of $\check{x}$ being 1×10^{-4}. The fixed relationship between ε and $\check{x}$ in (9.29) is apparently unnecessary. For convergence it is apparently only necessary that x_i is limited by "a higher order term smaller than ε as ε tends to zero" [90]; it is permissible simply to set $\check{x}$ at a constant, $\check{x} = 1 \times 10^{-4}$ for instance.

Relaxation schemes

By reshuffling equation (9.29) we may discover the values that the stress in element i is allowed to attain.

$$\sigma_i^m \leq \bar{\sigma} \left(\frac{\varepsilon}{x_i} + 1 \right) . \tag{9.30}$$

Thus, if $\varepsilon = 0.01$ and $\check{x} = \varepsilon^2$, for an element on its lower bound the stress measure is allowed to attain a value of $101\bar{\sigma}$ before the relaxed constraint is violated. Similarly, the stress for an element on its upper bound $x_i = 1$ can reach $1.01\bar{\sigma}$. However, ε may be much larger earlier in the optimisation process. If the initial value of ε is 0.2, for instance, the stress in the solid elements may exceed the allowable stress by 20 percent. For this reason, other relaxation schemes have been proposed that do not affect the stress limit for elements on their upper bounds. An example of such a scheme, presented in [8], is

$$\frac{\sigma_i^m}{\bar{\sigma}} - \frac{\varepsilon}{x_i} + \varepsilon \leq 1 \,. \tag{9.31}$$

For the results presented in the current chapter, we utilise the following relaxation,

$$\frac{\sigma_i^m}{\bar{\sigma}} \left(1 + \theta\varepsilon\right) - \frac{\varepsilon}{x_i} \leq 1 \,, \tag{9.32}$$

which reduces to (9.29) for $\theta = 0$. In the results presented we use $\theta = 1$ exclusively, so that σ_i^m is limited by

$$\sigma_i^m \leq \frac{\bar{\sigma}}{1 + \varepsilon} \left(1 + \frac{\varepsilon}{x_i} \right) . \tag{9.33}$$

Whatever the particular form of the ε-relaxation utilised, the conventional method of applying it in topology problems is to begin by considering a problem in which all the constraints are relaxed, and then, as the optimisation progresses, to gradually 'close down' the feasible domain by reducing the value of ε via some continuation. Obviously the reason for doing so is to maintain consistency with the proof presented in [90], which intimates that the solution produced hereby corresponds to a solution of the original unrelaxed problem.

We propose an alternate method of continuation, in which we first solve the unrelaxed problem and then 'open up' the feasible region in an effort to penetrate the degenerate portions of the unrelaxed design space. This is exactly opposite to what is normally adopted. The reasoning behind such a scheme is that the initial 'opened up' problem considered in the conventional 'closing down' scheme would appear to have a greater degree of multimodality than the unrelaxed problem, the feasible region being potentially highly nonconvex. Therefore, there may be a greater propensity to converge on inferior local minima if the 'closing down' scheme is used. In using the alternative 'opening up' scheme, the object is to first encourage convergence to a good local optimum of the unrelaxed problem, and then to proceed to improve on this solution by opening up whatever originally degenerate subspace may be connected to said solution.

Naturally we can no longer claim that the set of solutions that may be approached using this scheme can approach the strict set of KKT points of the unrelaxed problem (that is, the problem without stress relaxation). One should recall, however, that these KKT points are the solutions to the relaxed (in the sense of not discretised) continuous problem. Strictly speaking, we are only interested in these solutions if they can be made to approach $[0, 1]$ solutions via penalisation. If 'opening up' allows optimal designs with higher black-and-white fractions to be found, then this in itself would make the use of the approach defendable because these solutions would better represent the desired $[0, 1]$ solutions to the underlying discrete problem.

9.5.2 Material strength

Writing equation (9.8) in terms of the elasticity matrix for the solid material $\boldsymbol{C}_0$, the nodal displacements $\boldsymbol{q}_i$ and the strain displacement operator $\boldsymbol{B}_i$ for an element i in the finite element mesh, and representing the elemental stresses vectorially, we have

$$\boldsymbol{\sigma}_i = x_i^p \boldsymbol{C}_0 \boldsymbol{B}_i \boldsymbol{q}_i \, .$$

For a given vector of nodal displacements, it is evident that the elemental stresses scale according to x_i^p for intermediate-density material when SIMP penalisation is employed. There are a variety of stress-related failure criteria defined for solid isotropic materials, but how such criteria should extend to the unphysical intermediate-density material is not well defined. Duysinx and Bendsøe [84] consider the question by viewing intermediate-density material in the context of the homogenisation approach to topology optimisation, in which porous material has a physically significant microstructure. Based on their analysis, they then suggest a material strength law for power law material descriptions like SIMP, arguing that physically relevant strength laws should mimic the microstructural considerations that they identify in their analysis of materials with anisotropic microstructure. One of these considerations is that the material law should allow the local stress measure to tend towards a finite non-zero value, even as the local material density tends towards zero.

Duysinx and Bendsøe demonstrate that the local, microstructural stress for so-called rank 2 layered materials[3], which is different from the apparent macroscopic stress experienced by the material, tends towards a non-zero value as the local macroscopic density measure tends to zero, if the macroscopic strain field remains non-zero at zero density. They contend that these microstructural considerations are also highly relevant for a description of the strength of isotropic 'porous' material if sensible numerical results are to be achieved. Duysinx and Bendsøe go on to define a local microstructural stress for the intermediate-density material, which is then limited by the material yield stress. They point out that this is equivalent to modifying the overall material strength, which limits the maximum value of the macroscopic elemental stresses according to the elemental material density x_i. We here utilise this modification of material strength.

It is suggested in [84] that the local material strength for intermediate-density material should be interpolated using the same power law that is used for the interpolation of the material elasticity. Thus

$$\bar{\sigma} = x_i^p \sigma_0 \,, \tag{9.34}$$

where σ_0 is the relevant limit for the isotropic solid material (typically the yield stress of the material, as is used in both the Tresca and von Mises failure criteria).

The stress measure σ_i^m that is used for the results presented in this chapter is the von Mises stress, calculated for a state of plane stress (2.12). The limiting value σ_0 is therefore the material yield stress, although, since the examples are only illustrative, a set of normalised material properties is used. We therefore henceforth adopt the notation σ_i^{vm} specifically for the von Mises stress in element i. For the discretised planar problems considered, element stresses may be represented vectorially as

$$\sigma_i^T = \begin{bmatrix} \sigma_x & \sigma_y & \tau_{xy} \end{bmatrix}_i \,,$$

and the von Mises stress for element i can be written in matrix notation as

$$\sigma_i^{vm} = \sqrt{\sigma_i^T \, [\mathrm{VM}] \, \sigma_i} \,,$$

where

$$[\mathrm{VM}] = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} .$$

9.5.3 Stress relaxation and scaling of the material strength

In Reference [84], Duysinx and Bendsøe suggest a power law scaling of the material strength for porous material of the form

$$\bar{\sigma} = x_i^q \sigma_0 \,, \tag{9.35}$$

and show that if q is chosen so that $q < p$, the stress constraints derived thereby are no longer discontinuous at zero density, and so no relaxation needs be employed. Moving on, however, they warn that choosing too small a value for q results in unphysical structures characterised by an overexaggerated removal of material, and they advocate that $q = p$ for coherence with their

[3] See the homogenisation literature, beginning with [8].

analysis of rank-2 homogeneous material. Using $p = q$, the stress discontinuity survives, so it is necessary to apply stress relaxation, but the material description is consistent with physics.

Drawing on the work presented in [84], Bruggi [91] has suggested that (9.35), with $q < p$, may itself be used as a method of stress relaxation, instead of ε-relaxation, in which case (9.35) is no longer strictly interpreted as a material description. In Reference [93], Bruggi and Venini go on to formally demonstrate the close relationship between ε-relaxation and the use of this alternative scaling law, denoted the qp approach

Now, let us take the view that (9.34) is the correct (that is, physically meaningful) scaling law for material with intermediate density. The relaxation (9.35) limits the allowable stress measure for element i to

$$\sigma_i^m \leq x_i^q \sigma_0 \,,$$

which may then be written in terms of (9.34) as

$$\frac{\sigma_i^m}{\sigma_0} \leq x_i^p \left[\frac{1}{x_i^{p-q}} \right] .$$

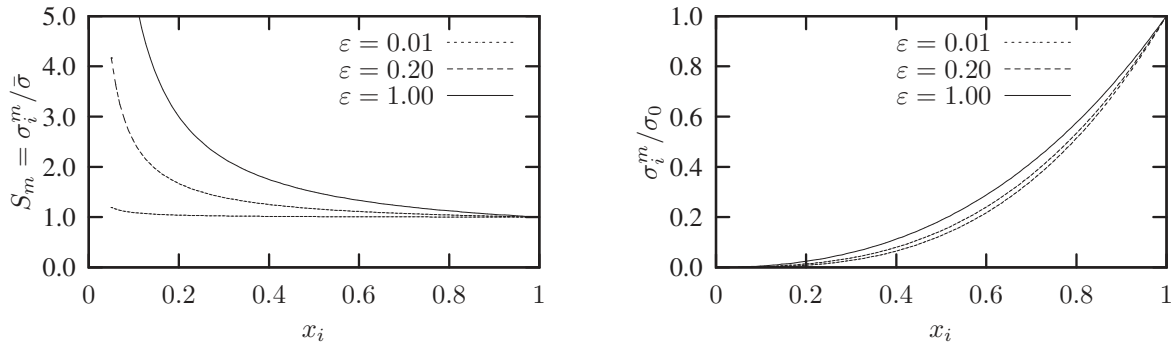
The term in square brackets $[\,\cdot\,]$ corresponds to a limiting value for what might be called a stress multiplier $S_m = \sigma_i^m / \bar{\sigma}$, which expresses the multiple by which the stress measure in an element of intermediate density can exceed the limiting value of said stress measure, given by (9.34), due to the stress relaxation. For the qp approach in which $q < p$, the stress multiplier S_m is an inverse exponential function, so that $S_m = 1$ at $x_i = 1$ and $S_m \to \infty$ as $x_i \to 0$. By substituting (9.34) into the ε-relaxed constraint (9.33), we may similarly write

$$\frac{\sigma_i^m}{\sigma_0} \leq x_i^p \left[(1+\varepsilon)^{-1} \left(1 + \frac{\varepsilon}{x_i} \right) \right] , \tag{9.36}$$

in which the stress multiplier function S_m is clearly reciprocal. As is pointed out by Rozvany in [88], the functions here referred to as the S_m recall the smooth envelope functions introduced in [89].

Figure 9.4(a) graphs the S_m from (9.36) for a few values of ε. Clearly, higher values of ε allow the stresses in elements of intermediate density to transgress the physically acceptable limiting value given by (9.34) by quite a margin. This is not an issue, of course, for elements at or near their minimum densities (near $x_i = 0$), because these elements contribute to the strain energy stored in the structure only imperceptibly, and ε-relaxation serves its purpose by creating the freedom for the densities in some elements to approach zero without violating the (now modified) stress constraints. However, the stresses in elements with not-insignificant densities are similarly allowed to be unphysically high. One assumes that these elements therefore store an unphysical and disproportionately high fraction of the strain energy in the structure. This is surely one reason why ε is reduced during the optimisation of truss structures. At the optimum, with $\varepsilon = 0.01$ (for instance), S_m is only appreciably different from unity for members whose cross-sections (or densities) are close to zero, as can be seen in Figure 9.4(a).

For the topology problems on which we focus in this chapter, we are interested in finding $[0, 1]$ solutions, or at least to get as close as we can to such solutions. In this context, it may well be possible to use larger relaxations, provided that the optimum designs found are characterised by high black-and-white fractions.



(a) Allowable stress as a multiple of $\bar{\sigma}$

(b) Scaling of material strength with ε-relaxation

Figure 9.4: The effect of ε-relaxation on the allowable stresses in material of intermediate density.

Lastly, Figure 9.4(b) graphs the right-hand side of (9.36) for different values of ε. Depicted in this way, ε-relaxation can be interpreted as a scaling of the material strength, in the manner advocated by Bruggi [91] and also by Le *et al.* [92].

9.5.4 Stress sensitivities

All of the subproblem forms mentioned in Section 9.4.1 utilise the first-order sensitivities of the objective and constraint functions in their construction. Thus it is necessary that the gradient vector of each of the constraint functions be calculated at each iteration point in the SAO sequence $\boldsymbol{x}^{\{k\}}$at which the approximate subproblems are defined. In terms of the various finite element matrices, the von Mises stress is expressed as

$$\sigma_i^{vm} = \left(x_i^{2p} \boldsymbol{q}_i^T \boldsymbol{V}_i \boldsymbol{q}_i\right)^{\frac{1}{2}}, \tag{9.37}$$

in which

$$\boldsymbol{V}_i = \boldsymbol{B}_i^T \boldsymbol{C}_0^T \left[\mathrm{VM}\right] \boldsymbol{C}_0 \boldsymbol{B}_i \,.$$

From (9.32), the constraint functions f_j are

$$f_j\left(\boldsymbol{x}\right) = \frac{\sigma_j^{vm}}{\bar{\sigma}}\left(1+\varepsilon\right) - \frac{\varepsilon}{x_j} - 1 \leq 0\,,$$

taking into account the fact that we use $\theta = 1$ throughout, and that we utilise the von Mises stress specifically. Each constraint f_j is an explicit function of x_j, but also depends implicitly on the remaining variables x_k, $k \neq j$. Taking cognisance of (9.34), the partial derivatives of the stress constraints may be expressed as

$$\frac{\partial f_j}{\partial x_i} = \frac{\partial \sigma_j^{vm}}{\partial x_i}\left(\frac{1+\varepsilon}{x_j^p \sigma_0}\right) - \delta_{ij}\left(p\frac{\sigma_j^{vm}}{x_j^{p+1}\sigma_0}\left(1+\varepsilon\right) - \frac{\varepsilon}{x_j^2}\right).$$

The partial derivatives of the von Mises stress, which is a function of the partial derivatives of the elemental stress vector, are not a standard output of the finite element code, nor are they calculable

directly from the information at hand upon solution of the finite element system, as is the case with the compliance objective and its gradients, equations (9.6) and (9.7). Furthermore, given the large number of design variables and constraints, finite difference calculations are not feasible. Two alternative, efficient methods of deriving the stress sensitivities, known as the direct and adjoint methods, are suggested in [27]. Following [84] we implement the latter, as it allows further computational advantage to be gained by incorporating an active set strategy. By differentiating (9.37) we obtain

$$\frac{\partial \sigma_j^{vm}}{\partial x_i} = \frac{1}{\sigma_j^{vm}} \left[\delta_{ij} \left(p x_j^{2p-1} \right) \boldsymbol{q}_j^T \boldsymbol{V}_j \boldsymbol{q}_j + \boldsymbol{q}_j^T \boldsymbol{V}_j \left(\frac{\partial \boldsymbol{q}_j}{\partial x_i} \right) \right] .$$

If the design loads are independent of the design variables (as we assume they are), from the finite element system

$$\frac{\partial \boldsymbol{K}}{\partial x_i} \boldsymbol{q} + \boldsymbol{K} \frac{\partial \boldsymbol{q}}{\partial x_i} = 0 .$$

Therefore, $\partial \boldsymbol{q} / \partial x_i$ is the solution $\boldsymbol{v}$ to the finite element system

$$\boldsymbol{K} \boldsymbol{v} = \boldsymbol{z} , \tag{9.38}$$

with $\boldsymbol{z}$ being a global vector of pseudo-loads defined by

$$\boldsymbol{z} = -\frac{\partial \boldsymbol{K}}{\partial x_i} \boldsymbol{q} ,$$

in which only the components corresponding to the degrees of freedom of element k are non-zero, said non-zero sub-vector being given as

$$\boldsymbol{z}_i = p x_i^{p-1} \boldsymbol{K}_i \boldsymbol{q}_i .$$

Finally, $\partial \boldsymbol{q}_j / \partial x_i$ is the sub-vector of $\boldsymbol{v}$ containing only the components corresponding to the degrees of freedom of element j. Thus, if there are n constraints it is necessary to run n additional finite element solutions (with the n pseudo-load vectors as the applied loads) in order to fully describe the SAO subproblem during only one iteration of the optimisation. This, in turn, means that the weight or compliance minimisation of even structures with relatively modest mesh refinements becomes, numerically, a daunting proposition. Hence the requirement of an active set strategy.

9.6 Numerical considerations

The optimisation code used to determine the optimal topologies for P_W and P_C is SAOi [31], a FORTRAN-based sequential approximate optimisation package developed by Groenwold and Etman[4] for the solution of large-scale nonlinear inequality constrained optimisation problems. SAOi uses approximating functions that are convex, separable and quadratic. This allows one of two solvers to be chosen: a dual solver for problems in which the primal variables outnumber the constraints, and a QP solver in cases where the reverse is true. We use the dual solver exclusively for the results presented herein.

[4] Freely available for academic use from the originator, Albert A. Groenwold, via email.

Although the approximations are quadratic, the curvatures are tailored in such a way that they approximate the local monotonic behaviour of the structural responses well [33]. The algorithm allows the user to select the types of function approximations used in the construction of the SAO subproblems from a library of available convex, separable and quadratic forms. It is equipped with a sparse implementation of the dual solver, which is used in the generation of the forthcoming results. The dual problems themselves are solved using a limited memory BFGS solver, developed by Zhu and co-workers [75], that is able to handle the non-negativity constraints on the dual variables.

To avoid checkerboarding we employ displacement-based Q8 elements in the finite element analysis of the structure, using a package called EDSAP. This program was written with an efficient means of handling memory allocation and addressing, and has been used for the development and testing of different finite element formulations. Since access is allowed to the source code, the use of this program allows us to implement the adjoint method for the calculation of the sensitivities of the stress constraints in a fairly efficient manner. EDSAP uses an active column equation solver to solve the finite element system $\boldsymbol{K}\boldsymbol{q} = \boldsymbol{w}$, in which the following three processes occur in sequence:

1. The re-ordered global stiffness matrix is factorised by LDL factorisation.
2. The load vector is modified by forward reduction.
3. The system is solved for $\boldsymbol{q}$ by back substitution.

If multiple load cases are considered, the factorisation step is carried out once, after which the solution for each load case is obtained by multiple application of the forward reduction and back substitution phases. We make use of this and the access ESDAP affords us to minimise the amount of memory that needs to be used to define and store the pseudo-load vectors and stress sensitivities, and thereby to reduce as far as possible the RAM usage and the number of disc read-writes.

As is evident from Section 9.5.4, the calculation of the sensitivities of the stress constraints via the adjoint method requires the solution of the finite element system using a pseudo-load vector, there being one pseudo-load vector corresponding to every element in the mesh for which the stress sensitivities are required. To construct the pseudo-load vector it is first necessary to obtain the nodal displacements $\boldsymbol{q}$, as well as the elemental stresses. Having first obtained the elemental stresses, we may decide whether or not the stress constraint in a given element is critical, and thus whether that constraint should be considered in the definition of the SAO subproblem for the following iteration. For each element in which the stress is deemed critical, a pseudo-load vector is generated and submitted to the FE solution subroutine, entering the solution process at the forward reduction phase. The resulting solution vector $\boldsymbol{v}$ in (9.38) can then be used to calculate the vector of stress sensitivities for element j, namely

$$\frac{\partial \sigma_j^{vm}}{\partial x_i} \qquad \forall \quad j \in \mathcal{A}, \quad i = 1, 2, \cdots, n.$$

Whether or not a particular stress constraint is considered critical is controlled by a parameter C_{lim} that is set prior to the optimisation. Only constraints for which $\sigma_j^{vm} > C_{lim}$ are considered in the definition of the subsequent subproblem, and C_{lim} is set as a small negative number to ensure

that the inactive but near-active constraints at the solution to $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$ are utilised in the definition of $\mathrm{P}_{\mathrm{SUB}}^{\{k+1\}}$.

For a stress constraint that is considered critical, all the sensitivities are provided by the adjoint method. However, not all of these need to be passed to the optimiser to take part in the definition of $\mathrm{P}_{\mathrm{SUB}}^{\{k+1\}}$. As has been suggested, from the point of view of the necessitated computational effort and computational storage requirements in the context of sparse solvers, an advantage may be gained by simply omitting the 'insignificant' partial derivatives of the constraint functions. We investigate this idea by filtering out small partial derivatives of the stress constraint functions prior to the construction of the subproblems for P_W and P_C.

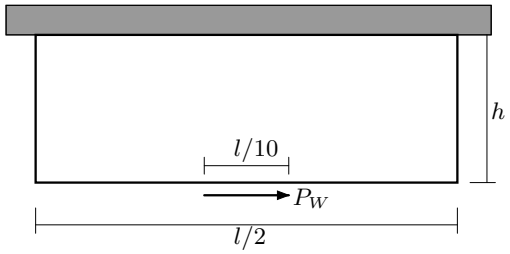
We introduce another parameter G_{lim}, again defined a priori, which acts as a lower limit on the size of the elements of the gradient vectors of the stress constraints. Thus, if $\left|\partial\sigma_j^{vm}/\partial x_i\right| < G_{lim}$ it is considered insignificant and is not passed to the optimiser. In this way, the sparsity of the Jacobian matrix of the critical stress constraints can be influenced. The reason for filtering out some of the Jacobian elements is simply to reduce the storage requirements for the algorithm in the hope of being able to solve larger problems more efficiently. Obviously, filtering out the Jacobian elements leads to the construction of subproblems that are not strictly first-order accurate. It may therefore be expected that the convergence characteristics of the SAO algorithm will be affected adversely by this strategy, certainly if the omission of Jacobian elements is applied too aggressively. We investigate whether or not a sizable computational advantage can be gained by applying this heuristic.

Lastly, it must be noted that, with the exception of two of the minimum compliance results for the MBB beam, we do not apply a mesh independence filter (nor any other restriction method) during the solution of the topology problems, as is advocated in Section 2.1.1. We prefer to use the sensitivity filter in numerical implementations but, as explained in Chapter 3, this somewhat complicates the interpretation of the results. The implementation of a restriction method *per se* adds to the computational burden of solving the topology problems (particularly the non-filter-based methods) and, as our primary goal is simply to solve large topology problems using the dual SAO method and to test the stress relaxation and Jacobian filtering strategies that we have introduced, it is unnecessary for us to enforce mesh independence in this case.

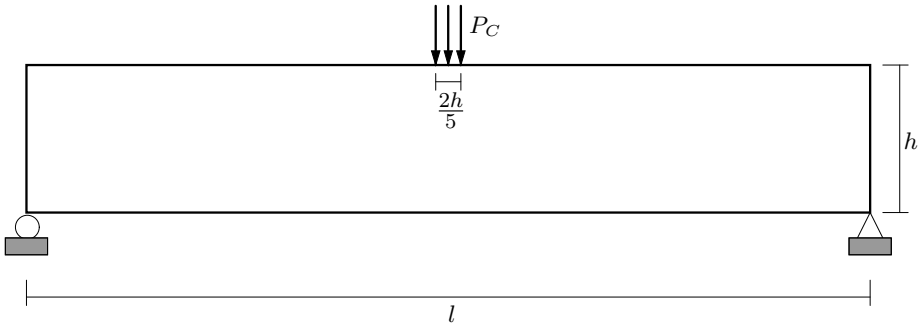
9.7 Results

We here present some of the results obtained when applying a dual SAO algorithm to solve the structural optimisation problems described in Section 9.3. For each of the problems specified, two specific ground structures will be considered, both of which are well-known ground structures that are often used in standard test problems. The first is the two-bar truss structure, illustrated in Figure 9.5(a). The second is the MBB beam structure, shown in Figure 9.5(b); symmetry is invoked in the analysis of the MBB beam so that only half the beam is modelled.

Both problems P_W and P_C possess multiple local minima, but we do not implement a continuation strategy on the penalty parameter p (as has been recommended, for example in [65], to stabilise the global search and avoid as far as possible convergence to inferior local minima, particularly for the compliance problem). A continuation strategy on ε is already required in the constraint



(a) The two-bar truss (unit thickness; plane stress)



(b) The MBB beam (unit thickness; plane stress)

Figure 9.5: Ground structures for the example problems ($P_W = 6N$, $P_C = 1N$, $l = 6$m, $h = 1$m, $E = 1N/\text{m}^2$, $\nu = 0.3$).

relaxation strategy, and we wish to limit the number of permutations that arise in trying to identify an acceptable set of parameters for the optimisation. In producing the results below, we have focused instead on the following:

- Testing the concept of 'opening up' the design domain, which is an alternative continuation strategy for the relaxation of the stress constraints.
- Testing whether or not a good percentage of the partial derivatives of the stress constraints can be omitted when defining the subproblem, without upsetting the ability of the algorithm to converge to a local optimum.
- Using these last two ideas in the solution of larger problems ('large' being relative, of course).

As such, we first briefly present some results from weight minimisation of the two-bar truss, using a coarse mesh discretisation, to motivate our choice of parameter settings and approximation strategies. Thereafter, results are presented for weight minimisation of the MBB structure and the minimum compliance design of both structures (at various mesh refinements) using the set of parameter values standardised upon. The mesh discretisation for a particular problem will be stated in terms of a mesh multiplier m; the height of the given ground structure is divided into $5m$ elements, and the length into $15m$. The total number of elements in the mesh is therefore $75m^2$.

The various optimisation runs have been carried out on different computers, principally because the larger meshes require machines with greater capacity to carry out the optimisation in reasonable time, and larger storage capacity to store the information used to define the subproblems (despite

the fact that only first-order information is utilised). The specific computer used to generate a given set of results will be identified by the label M_i (machine i). Details for the machines are as follows:

Computer $\mathbf{M_1}$	
Processor:	Intel(R) Core(TM) 2 CPU 6700 @ 2.66GHz (using only one of the cores)
Memory:	Total memory (RAM), 3.8 GiB
Operating system:	Linux 2.6.34-12-default x86-64 on openSUSE 11.3 (x86-64)
Computer $\mathbf{M_2}$	
Processor:	Intel(R) Xeon(TM) 8 core CPU 3.73GHz (using only one of the cores)
Memory:	Total memory (RAM), 31.5 GiB
Operating system:	Linux 2.6.34-12-desktop x86-64, openSUSE 11.3 (x86-64)

9.7.1 The selection of standard settings

Weight minimisation of the two-bar structure is used initially to define the program parameters to be used in the generation of the remainder of the results presented. For these tests, the design domain is discretised with $m = 4$ as the mesh multiplier.

The expected optimal design consists of two bars (truss members) forming a V whose vertex supports the applied shear load. It is easy enough to get an idea of the minimum required area of the structural members necessary to support the applied load if one considers a simplified symmetric structure consisting of two uniaxial truss members supporting a point load. Using this one-dimensional simplification and considering only constraint satisfaction and not weight reduction, one finds that the minimum bar areas occur when the legs of the V are oriented at ± 45 degrees to the vertical. We use the areas calculated in this way as a check on the topologies obtained, by calculating lower bounds on the widths of members allowed for feasible designs, assuming that the topologies possess two members, and assuming unit depth for the FEM mesh. Table 9.1 uses the minimum required thicknesses of the truss members to express the necessitated dimension as the number of element diagonals N_{el} required to span the width of each bar for various mesh refinements. The mesh refinement is stated in terms of the mesh multiplier m in the table and, for a given mesh refinement, l_d is the length of an element diagonal.

Due to the weight minimisation objective, we expect that the optimal topologies will have narrower V shapes (to reduce the leg lengths as far as possible) and the corresponding member widths would have to be larger than the indications given in the table. The exact optimal configuration therefore depends on the mesh discretisation, as layers of material can only be added or subtracted in discrete chunks. Figure 9.6(a) illustrates nicely the type of two-bar topology expected in the planar case. The minimum width of each member is indeed greater than three element diagonals predicted in Table 9.1.

The initial starting point for all the optimisation runs is $x_i = 0.5 \ \forall \ i = 1, 2, \cdots, n$. The two topologies depicted in Figure 9.6 are gained on $\mathbf{M_1}$ by using marginally different continuation strategies on the parameter that controls the relaxation of the stress constraints ε. Both strategies are consistent with that used in [84]. They are strategies in which the design domain is 'closed down', which is to say that ε is made smaller as the optimisation progresses, thereby closing the

m	l_d (meters)	N_{el}	m	l_d (meters)	N_{el}
1	0.2828	0.75	8	0.0354	6.00
2	0.1414	1.50	9	0.0314	6.75
3	0.0943	2.25	10	0.0283	7.50
4	0.0707	3.00	11	0.0257	8.25
5	0.0566	3.75	12	0.0236	9.00
6	0.0471	4.45	13	0.0218	9.75
7	0.0404	5.25	14	0.0202	10.50

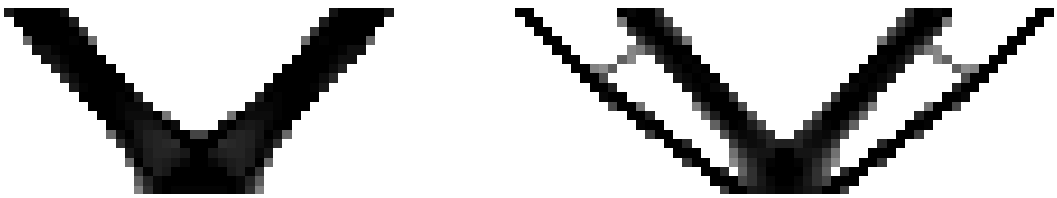
Table 9.1: Expected widths of the truss members for optimal topologies in the minimum weight two-bar truss problem.

degenerate domains caused by the stress constraints that are more accessible with ε larger. For the result in Figure 9.6(a), ε is initially set to a value of 0.1 for the first 30 iterations. Thereafter, ε is decreased as $\varepsilon = \varepsilon/1.1$ whenever the maximum constraint violation is less than 0.001. A lower bound of 1×10^{-2} is set for ε. The minimum allowable value for x_i during any iteration is linked to ε as $\check{x} = \varepsilon^2$ for all i.

The result depicted in Figure 9.6(b) is generated using an identical continuation strategy, except that the initial value of ε is 0.2, rather than 0.1. Clearly, the final topology is a different local minimum. The design is feasible, and correlates with the expected truss dimensions from Table 9.1, except that the material is distributed among four members rather than two. Both designs were generated using CONLIN. We standardise on the first strategy (that used to generate 9.6(a)) for the remainder of the results that are generated using the 'opening up' continuation, since it mimics the settings adopted by Duysinx and Bendsøe in [84].

Different approximations

We have tried various approximation strategies to determine which is able to generate superior solutions and, perhaps more importantly, which makes for the most efficient algorithm. While comparing the approximation strategies we have used the 'closing down' continuation on stress relaxation and a dense infrastructure in all cases. Although the comparison is inevitably problem dependent, it is expected that, due to the stress constraints, the forms of P_W and P_C will be suf-



(a) $\varepsilon_{init} = 0.1$ (b) $\varepsilon_{init} = 0.2$

Figure 9.6: Local optima found using a continuation strategy on ε with different initial settings for ε.

CONLIN	Fleury's standard convex linearisation algorithm, see [4].
NONCON	A method of approximation, suggested initially by Fleury, in which the stress constraints are represented by the reciprocal approximation (the objective is kept linear). Unlike CONLIN, the reciprocal terms are allowed to have positive or negative gradients, so the resulting functions are generally nonconvex. As the resulting subproblems possess a convex transform, they can still be solved using Falk's dual method. This approximation was used in Chapter 6 for the solution of the nonconvex minimum weight problem.
T2:CONLIN	This is an implementation of CONLIN in which the reciprocal terms produced by CONLIN are replaced with the quadratic approximation to the reciprocal approximation.
T2:R	For the weight minimisation problem, the objective remains linear, while the constraints are represented as the quadratic approximation to the reciprocal approximation.
T2:E	The objective is kept linear, while the constraints are represented as the quadratic approximation to the exponential approximation, derived using historic information. The first iteration is carried out using T2:R.
T2:MMA	The objective is kept linear, while the constraints are represented as the quadratic approximation to the MMA approximation. Refer to [33] for further details.

Table 9.2: The approximation strategies that are compared for the weight minimisation of the two-bar truss.

ficiently similar so that the same approximation scheme will work well for both problems. As it happens, there is in any case very little difference between the optimal topologies and their associated function values found using the different approximation strategies. A list of the approximation strategies that were tested is given in Table 9.2, as well as a brief description of each. The results are tabulated in Table 9.3, which are all produced on $\mathbf{M_1}$.

In Table 9.3, f_0^* is the objective function value at the optimum that was found, and N_{iter} is the number of iterations k required for convergence. Results marked with an asterisk* have failed to converge due to small-scale oscillation. These were terminated artificially after 150 iterations. The process is deemed to have converged if the following criterion is satisfied:

$$\sum_{i=1}^{n} \left| x_i^{\{k\}} - x_i^{\{k-1\}} \right| \leq 1 \times 10^{-4} \, .$$

The column heading E_{sub} indicates the average number of subproblem evaluations carried out per iteration k, while $\phi_{B\&W}$ is the black-and-white fraction of the solution, calculated by

$$\phi_{B\&W} = \frac{n_{[0]} + n_{[1]}}{n} \, . \tag{9.39}$$

Here, $n_{[0]}$ is the number of elements on their lower bounds ($x_i = \check{x} = 1 \times 10^{-4}$), $n_{[1]}$ is the number of elements on their upper bounds ($x_i = \hat{x} = 1$), and n is the total number of elements in the

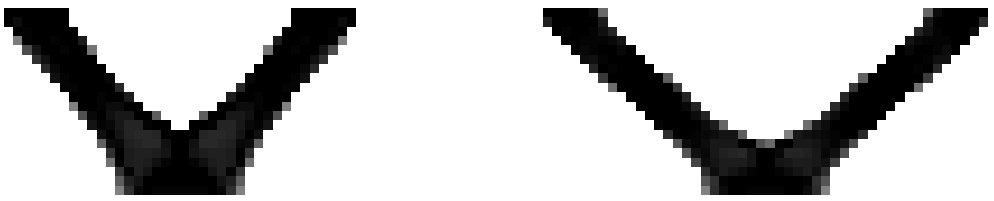
Approximation	f_0^*	N_{iter}	E_{sub}	$\phi_{B\&W}$	T_{avg}^{75}	T_{90}^{I}
CONLIN	281.50	90	1086	0.837	18.8	18.1
T2:CONLIN	281.50	90	510	0.837	4.7	6.6
T2:R	286.05	89	556	0.837	13.8	20.1
T2:E	285.41	85	732	0.838	6.7	10.2
T2:MMA	285.47	126	1897	0.843	11.8	26.1
NONCON*	281.90	150	1515	0.828	37.9	74.0

Table 9.3: Summary of results obtained for the weight minimisation of the two-bar truss using various approximations.

FE mesh. Furthermore, T_{avg}^{75} represents the average CPU time (in seconds) required per iteration, calculated for the first 75 iterations (since none of the runs have terminated by then and later iterations near termination are usually comparatively quick). The time required to complete an iteration can vary considerably due to the large variation in the effort required to solve individual subproblems. Often, one or two subproblems can require orders of magnitude more effort than the others, which skews T_{avg}^{75} somewhat. Therefore, T_{90}^{I} is also stated. Ninety percent of all the iterations individually require less than the time indicated by T_{90}^{I}.

Comparing the approximation strategies in Table 9.3, it appears that CONLIN is able to locate slightly superior solutions from the point of view of the optimal objective function values obtained, whereas T2:R is able to perform the optimisation more efficiently, requiring both fewer subproblem evaluations on average and less time than CONLIN. However, it is the combination of the two, namely T2:CONLIN, that performs best for this problem, apparently representing 'the best of both worlds', as it were.

Visually, the optimal topologies generated using these approximation strategies are all quite similar. They are all instances of the V-shaped, two-member topology expected, with their main differences being the width of the V and the black-and-white fractions obtained. Figure 9.7 depicts the range of the topologies obtained, T2:E having produced the narrowest V-shaped structure, while NONCON produced the widest (albeit that it did not finally converge). The solutions with narrow V shapes have three elements on their upper bounds, plus one additional element (usually grey) spanning the width of the bar. In the wider structure, the additional element is often absent.



(a) T2:E approximation, $f_0^* = 285.41$ (b) Nonconvex approximation, $f_0^* = 281.90$

Figure 9.7: Representative optimal topologies for the weight minimisation of the two-bar truss using various approximations.

The effect of $\mathbf{C_{lim}}$ and $\mathbf{G_{lim}}$

The effect of introducing a constraint selection strategy and a 'filtering out' strategy on the constraint sensitivities is now investigated. We use the T2:R approximation for these tests and the 'closing down' constraint relaxation strategy, with the initial value of ε being $\varepsilon = 0.1$. The results are generated on $\mathbf{M_1}$, and may be compared with the result for T2:R in Table 9.3. The first row in Table 9.4 summarises the solution obtained when only the constraints j that satisfy

$$f_j\left(\boldsymbol{x}^{\{k\}}\right) \geq -10$$

are included when defining the subproblem in each iteration k. For the constraints that are included, all of their partial derivatives are carried over into $\mathrm{P}_{\mathrm{SUB}}^{\{k\}}$. The second row corresponds to including all the constraints, but omitting any of the partial derivatives for which

$$\left|\frac{\partial f_j\left(\boldsymbol{x}^{\{k\}}\right)}{\partial x_i}\right| < 1 \times 10^{-6} \ .$$

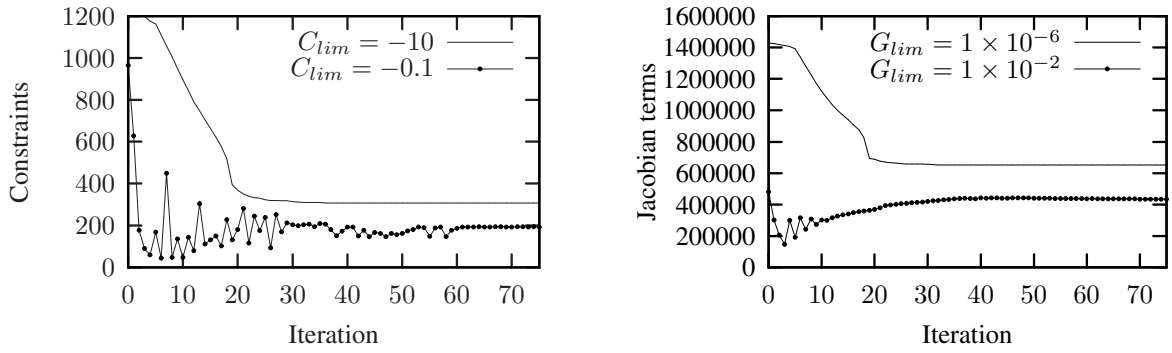
The last row represents the combined application of both the constraint selection and Jacobian filtering strategies simultaneously. By comparison with the third row in Table 9.3, the solution appears unaffected by these heuristics, as both the optimum f_0^* and the black-and-white fraction $\phi_{B\&W}$ are identical in all cases. The number of iterations required and the average number of subproblem evaluations required also do not change appreciably. However, the average CPU time required is roughly halved by each of the two schemes individually, and roughly quartered by the combined application of the two.

Figure 9.8 graphically depicts the number of constraints and Jacobian terms considered throughout the optimisation process for the parameterisation given in the first two rows of Table 9.4. With $\mathbf{C_{lim}} = -10$ or $\mathbf{G_{lim}} = 1.0 \times 10^{-6}$, the bulk of the constraints and their sensitivities are retained during the global phase of the search, while in the region of the local minimum only about one third of the constraints are selected, or approximately half of the Jacobian terms in the case of filtering the Jacobian only. There is a stable, monotonic transition between the two regimes. These heuristics lead directly to an appreciable reduction in the necessitated storage requirements, albeit that initially the implementation is effectively a dense one. Still, the reduction in CPU effort is important if larger problems are to be considered.

Figure 9.8 also graphs the behaviour of more aggressive selection and filtering schemes. Although further gains are made in terms of reducing the size of the subproblems, the behaviour of the

C_{lim}	G_{lim}	f_0^*	Iter	E_{sub}	$\phi_{B\&W}$	T_{avg}^{75}	T_{90}^{I}
-10	All	286.05	89	559	0.837	6.2	13.2
All	10^{-6}	286.05	90	576	0.837	7.2	11.7
-10	10^{-6}	286.05	89	566	0.837	2.8	8.7

Table 9.4: Solutions obtained using a selection strategy on the constraints, the partial derivatives of the constraint functions, or both simultaneously.



(a) Number of selected constraints

(b) Number of selected Jacobian terms

Figure 9.8: The effect of implementing (a) a constraint selection strategy and (b) a selection strategy on the partial derivatives of the constraint functions.

optimisation is now more oscillatory and unstable. Neither implementation using one of the more aggressive settings managed to converge to a local solution in under 150 iterations. In all of the results to follow we have used both heuristics in combination.

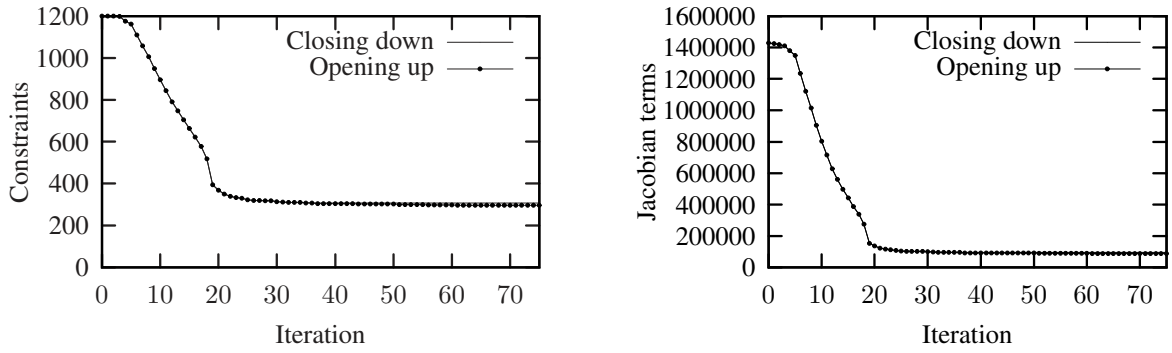
The effect of 'opening up'

Lastly, we implement the alternative continuation strategy on the constraint relaxation parameter ε, the motivation for which having been discussed in Section 9.5.1. Initially, ε is set to $\varepsilon = 0.01$. As with the 'closing down' strategy, ε is not changed during the first thirty iterations of the optimisation. Thereafter, ε is increased as $\varepsilon = 1.1\varepsilon$ whenever the maximum constraint violation is less than 0.001, and a maximum limiting value is set at $\varepsilon = 1.0$. The tests were again carried out on computer $\mathbf{M_1}$.

Table 9.5 summarises the solution obtained by applying 'opening up' in combination with two different methods of approximation. In both cases, the constraint selection and Jacobian filtering heuristics have also been applied. The results may be compared with the corresponding results from Table 9.3; 'opening up' the design domain apparently allows solutions with superior function values to be found, as well as superior black-and-white fractions. There is very little difference in the size of the generated subproblems when comparing the 'opening up' and 'closing down' schemes, as is indicated by Figures 9.9(a) and 9.9(b). The graphs are generated by comparing the optimisation runs using T2:R, and on the scale at which the graphs are drawn the differences are imperceptible. Figure 9.10 depicts the optimal solutions found in this case.

Approximation	C_{lim}	G_{lim}	f_0^*	Iter	E_{sub}	$\phi_{B\&W}$	T_{avg}^{75}	T_{90}^{I}
T2:R	-10	10^{-6}	279.17	77	423	0.865	2.5	8.6
CONLIN	-10	10^{-6}	275.40	84	683	0.857	3.0	7.1

Table 9.5: Solutions obtained when 'opening up' the design space.



(a) Number of selected constraints

(b) Number of selected constraints

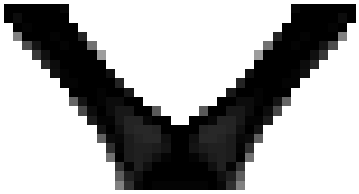
Figure 9.9: A comparison of 'closing down' versus 'opening up' the design space using T2:R.

It is very difficult to compare the quality of the solutions generated by 'opening up' and 'closing down' directly because, as was discussed in Section 9.5.3, the two different relaxations employed can be interpreted as defining two different material behaviours for intermediate-density material. So, if the strategy of 'opening up' allows intermediate-density material in the optimum design to feasibly attain higher stresses relative to the optimum found by 'closing down', then it follows that less material is required in the structure at the point of static failure. This may be all that we are seeing when we note that the function values are superior when the 'opening up' strategy is used. Hence, it is difficult to make a meaningful comparison of the optimal topologies based solely on the basis of their function values, unless they are each purely $[0, 1]$ designs.

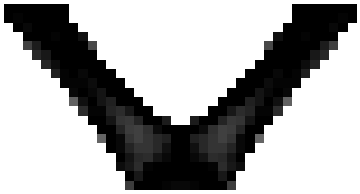
Table 9.6 shows how the strain energy in the optimal structures is divided between the solid elements $[1]$ and elements of intermediate density $[i]$ (the strain energy associated with the large number of elements on the lower bound is insignificant – as it should be). The strain energy is calculated using the left-hand side of equation (9.3), in which the elasticity matrix of the material with intermediate density is given by the SIMP scaling (9.5). Also shown are the number of elements on their lower bounds $n_{[0]}$, the number of solid elements $n_{[1]}$ and the number of elements of intermediate density $n_{[i]}$ in the design, as well as the total compliance of the structure f_c. The symbols $\%_{C[i]}$ and $\%_{C[1]}$ denote what percentage of the total compliance resides in the intermediate-density elements and the solid elements respectively. We see that the result obtained by 'opening up' the design space has fewer elements of intermediate density, and although the stresses in these elements are allowed to be higher than in their 'closed down' counterparts, they are still cumulatively responsible for a smaller portion of the strain energy in the structure[5].

Again, this comparison is by no means a rigorous justification for preferring one result over the other. The fact remains that, in both cases, the intermediate-density material plays a very large role in determining the nature of the resulting structure. In the 'closed down' result, the intermediate-density material behaves more closely in accordance to the physical law enunciated by Duysinx and Bendsøe in [84], but the structure obtained has more such material that, unlike in the case of the truss problems described earlier, is itself 'unphysical', given the strictly $[0, 1]$ nature of the underlying discrete problems for both P_W and P_C. Ultimately, we follow the example of Le *et*

[5]It must be said, however, that they have a higher average compliance per element than the intermediate-density elements in the 'closed down' solution.



(a) Solution obtained using 'closing down'



(b) Solution obtained using 'opening up'

Figure 9.10: A comparison of the optimal topologies resulting from 'closing down' versus 'opening up' the design space (using T2:R).

al. by preferring to adopt a promising numerical procedure for generating good solutions with high black-and-white fractions. The unphysical scaling of the material properties of the already unphysical intermediate-density material is, for the time being, relegated to secondary importance. Thus, we prefer to use the 'opening up' continuation in generating the rest of our results, as it shows promise in producing optima with higher black-and-white fractions.

For the sake of interest, Figure 9.11 provides a further comparison of the optimal topologies achieved using the two continuation strategies. In these pictures, with the exception of 9.11(c) and 9.11(d), the material plotted in uniform grey indicates all the elements for which x_i is above the lower bound value. The black elements in Figures 9.11(a) and 9.11(b) indicate the elements in which the relaxed stress constraint is active at the solution. From equation (9.32), with $\theta = 1$, the relaxed elemental stress constraints are given as

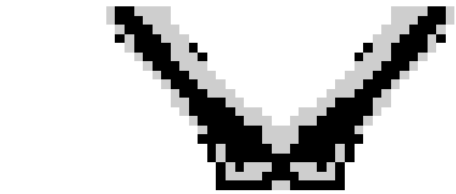
$$\sigma_i^R = \frac{\sigma_i^{vm}}{\bar{\sigma}}\,(1+\varepsilon) - \frac{\varepsilon}{x_i} - 1 \leq 0\,.$$

In creating these plots, the stress constraints have been considered active if $\sigma_i^R > -1 \times 10^{-6}$. Figures 9.11(c) and 9.11(d) depict the elements in which $S_m > 1$, see equation (9.36). The von Mises stress in these elements exceeds the 'physically relevant' allowable limiting stress for material of intermediate density suggested, by Duysinx and Bendsøe, equation (9.34). None of these elements are on their upper bounds, of course. The grey scale of the figure indicated relative values of S_m for the non-white elements. The maximum value of S_m (pure black in the figure) for the 'closed down' result is $S_m^{max} = 3.58$, while for the 'opened up' result it is $S_m^{max} = 3.35$.

Figures 9.11(e) and 9.11(f) indicate which elements are on their upper bounds in the optimal topologies. Although both topologies have black-and-white fractions above 80%, the 'black fraction' of the elements that actually makes up the structure is far lower. Given the difficulty of interpreting intermediate-density elements, as well as their effect on the optimal topologies (il-

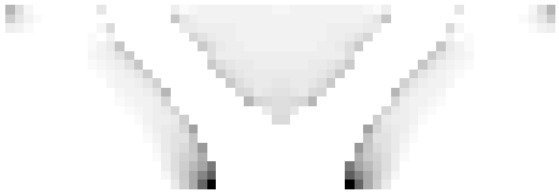
Relaxation strategy	$n_{[0]}$	$n_{[i]}$	$n_{[1]}$	$\%_{C[i]}$	$\%_{C[1]}$	f_c
Closing	892	196	112	59.8	40.2	234.37
Opening	904	162	134	50.6	49.4	240.90

Table 9.6: The distribution of strain energy between the 'solid' elements and elements of intermediate density for the solutions obtained by 'closing down' and by 'opening up'.



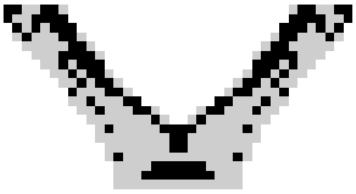
(a) Closing down: Elements with $\sigma_i^R > -1 \times 10^{-6}$

(b) Opening up: Elements with $\sigma_i^R > -1 \times 10^{-6}$

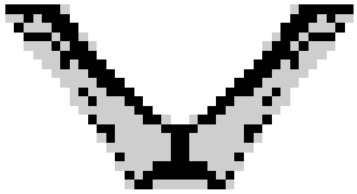


(c) Closing down: Elements for which $S_m > 1$

(d) Opening up: Elements for which $S_m > 1$



(e) Closing down: Elements on the upper bound



(f) Opening up: Elements on the upper bound

(g) Closing down: Elements for which $x_i > 0.85$

(h) Opening up: Elements for which $x_i > 0.85$

Figure 9.11: Further comparison of the optimal topologies obtained when using the 'closing down' and 'opening up' continuation strategies on the ε-relaxation.

lustrated for instance by the differences induced because of these elements when using the two dissimilar continuation strategies), the figures suggest that we still need to improve at finding $[0, 1]$ solutions. Lastly, Figures 9.11(g) and 9.11(h) record what elements of intermediate density in the optimal topologies have densities below 0.85 (the grey material in the figures).

We now proceed to catalogue the remainder of the results that we have obtained using the settings discussed. We first present solutions for weight minimisation of the two-bar truss with denser mesh discretisations, and then we illustrate the type of topologies generated when minimising the compliance for the two-bar ground structure. Lastly, we present similar results (weight minimisation and compliance minimisation) for the MBB beam structure.

To be clear, we use the SAOi algorithm in which the subproblems are defined using the T2:CONLIN approximation and the subproblems are solved using the sparse dual solver. The stress constraints are relaxed using the ε-relaxation (9.32), and we employ the 'opening up' continuation strategy on ε described here. For the two-bar truss problems we employ a constraint selection strategy

with $C_{lim} = -10$, and we filter out small terms in the Jacobian of the stress constraints using $G_{lim} = 1 \times 10^{-6}$. For the MBB beam problems, these settings are changed to $C_{lim} = -1.0$ and $G_{lim} = 1 \times 10^{-4}$; many more inactive constraints have values between -1.0 and -10 for the MBB beam than in the case of the two-bar structure.

9.7.2 Optimal designs for the two-bar truss

Minimum weight results for the two-bar truss structure are presented in Figure 9.12, beginning with a mesh multiplier of $m = 4$. The objective function value, the weight, is stated in the more convenient form of a volume fraction f_v, given by

$$f_v = \frac{1}{n} \sum_{i=1}^{n} x_i \,,$$

and the compliance of the design f_c is given for comparison with the minimum compliance results. Also indicated are the number of elements on their upper bounds $n_{[1]}$, the number of elements of intermediate density n_i, the black-and-white fraction $\phi_{B\&W}$ given by equation (9.39), the number of active constraints in the final design N_{act}, the number of iterations to termination N_{iter}, the average CPU time (in seconds) per iteration T^I_{avg} and, finally, the number of elements in the mesh. Note that the stated times correspond only to the times required by the optimizer, and exclude the times devoted to the FEM analyses.

We have not used a filter in generating the results and the optimal design clearly is mesh dependent. As the mesh discretisation increases, the optimal objective function value decreases as a result of the increased detail in the successive designs. With the exception of the $m = 6$ result, the trend is also for the compliance of the designs to decrease (slightly) with increased mesh refinement. The most refined mesh we present has 14700 elements.

Minimum compliance results for the two-bar truss are shown in Figure 9.13. The structures presented in Figures 9.13(a) and 9.13(b) were solved with a prescribed limiting volume fraction of $f_v = 0.5$ and $m = 4$. Figure 9.13(a) is the solution to the standard compliance optimisation problem without stress constraints, whereas Figure 9.13(b) depicts the solution achieved when the problem is rerun with identical settings and the addition of stress constraints. Again, neither design is solved with the aid of a filter, and consequently the black-and-white fraction is close to 1 for both. None of the stress constraints is active in Figure 9.13(b), the load being too low and the volume fraction too high to force the stresses in any of the elements to the failure stress. Figure 9.13(c) and 9.13(d) are similar results for a prescribed volume fraction of $f_v = 0.235$, chosen as such to be close to the optimal minimum weight result for $m = 4$, Figure 9.12(a). In this case, the optimiser could not find a stress-constrained solution that is feasible with respect to the volume constraint, and instead terminated at the solution depicted. The compliance value of this structure is close to that of the minimim weight result, as is the number of active constraints.

Figures 9.13(e) and 9.13(f) are generated with the allowable volume fraction set to $f_v = 0.25$, close to the optimal minimum weight result but with enough leniency to allow feasible stress-constrained minimum compliance results to be found. The addition of stress constraints to the minimum compliance problem has a marked effect on the optimal topology, even for this simple structure. The design produced is much narrower than the optimal topology without stress constraints. No doubt,

there are over-stressed elements in this design, and the response to the addition of stress constraints is to narrow the V shape, thickening and shortening the legs so as to keep the volume of the design constant, at the expense of compliance.

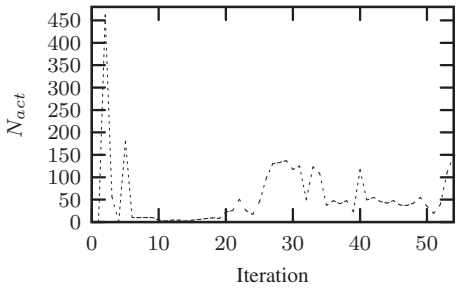
The stress-constrained minimum compliance results take markedly longer to generate than the minimum weight results. The largest we have produced is with $m = 12$, shown in Figure 9.13(g). On the whole (with the exception of the topology in Figure 9.13(d), in which the volume constraint is violated) the minimum compliance designs have higher black-and-white fractions than do the minimum weight designs.

Weight minimisation

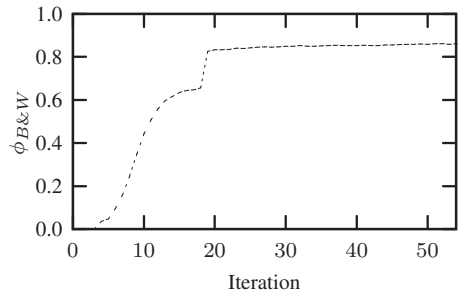
(a) Result for $m = 4$: $f_v = 0.234$, $f_c = 233.81$, $N_{act} = 144$, $n_{[1]} = 106$, $n_{[i]} = 196$, $\phi_{B\&W} = 0.837$, $T^I_{avg} = 6.7$, $N_{iter} = 26$, $n = 1200$

(b) Result for $m = 6$: $f_v = 0.225$, $f_c = 236.76$, $N_{act} = 174$, $n_{[1]} = 270$, $n_{[i]} = 372$, $\phi_{B\&W} = 0.862$, $T^I_{avg} = 10.6$, $N_{iter} = 75$, $n = 2700$

(c) Result for $m = 10$: $f_v = 0.221$, $f_c = 232.51$, $N_{act} = 373$, $n_{[1]} = 914$, $n_{[i]} = 796$, $\phi_{B\&W} = 0.894$, $T^I_{avg} = 118.8$, $N_{iter} = 88$, $n = 7500$



(d) Convergence plot $m = 6$: Number of active constraints N_{act}



(e) Convergence plot $m = 6$: Black-and-white fraction $\phi_{B\&W}$

(f) Result for $m = 14$: $f_v = 0.217$, $f_c = 232.44$, $N_{act} = 1395$, $n_{[1]} = 1751$, $n_{[i]} = 1521$, $\phi_{B\&W} = 0.897$, $T^I_{avg} = 614.6$, $N_{iter} = 240$, $n = 14700$

Figure 9.12: Optimum topologies generated by weight minimisation of the two-bar truss structure. The results for $m = 4$ and $m = 6$ were produced using computer $\mathbf{M_1}$; those for $m = 10$ and $m = 12$ were run on $\mathbf{M_2}$.

Minimum compliance

(a) No stress con. $m = 4$: $f_v = 0.50$, $f_c = 94.66$, $N_{act} = 1$, $n_{[1]} = 596$, $n_{[i]} = 4$, $\phi_{B\&W} = 0.9967$, $T^I_{avg} = 0.0$, $N_{iter} = 92$, $n = 1200$

(b) With stress con. $m = 4$: $f_v = 0.50$, $f_c = 94.35$, $N_{act} = 1$, $n_{[1]} = 598$, $n_{[i]} = 2$, $\phi_{B\&W} = 0.9983$, $T^I_{avg} = 5.21$, $N_{iter} = 28$, $n = 1200$

(c) No stress con. $m = 4$: $f_v = 0.235$, $f_c = 220.83$, $N_{act} = 1$, $n_{[1]} = 276$, $n_{[i]} = 8$, $\phi_{B\&W} = 0.993$, $T^I_{avg} = 0.0$, $N_{iter} = 50$, $n = 1200$

(d) With stress con. $m = 4$: $f_v = 0.237$, $f_c = 233.5$, $N_{act} = 147$, $n_{[1]} = 128$, $n_{[i]} = 174$, $\phi_{B\&W} = 0.855$, $T^I_{avg} = 12.30$, $N_{iter} = 102$, $n = 1200$

(e) No stress con. $m = 4$: $f_v = 0.25$, $f_c = 202.53$, $N_{act} = 1$, $n_{[1]} = 298$, $n_{[i]} = 2$, $\phi_{B\&W} = 0.9983$, $T^I_{avg} = 0.0$, $N_{iter} = 61$, $n = 1200$

(f) With stress con. $m = 4$: $f_v = 0.25$, $f_c = 212.99$, $N_{act} = 25$, $n_{[1]} = 272$, $n_{[i]} = 30$, $\phi_{B\&W} = 0.975$, $T^I_{avg} = 2.30$, $N_{iter} = 71$, $n = 1200$

(g) With stress con. $m = 12$: $f_v = 0.25$, $f_c = 202.07$, $N_{act} = 71$, $n_{[1]} = 2588$, $n_{[i]} = 120$, $\phi_{B\&W} = 0.989$, $T^I_{avg} = 3714$, $N_{iter} = 95$, $n = 10800$

Figure 9.13: Optimal topologies generated by compliance minimisation of the two-bar truss structure. The $m = 4$ results were produced using machine $\mathbf{M_1}$, while $m = 12$ was run on $\mathbf{M_2}$.

9.7.3 Optimal designs for the MBB beam

Minimum weight designs for the MBB beam are presented in Figure 9.14 for two mesh refinements, $m = 4$ and $m = 6$. The optimal topologies constitute a nested series of arches, which are more numerous and more refined in $m = 6$ than in $m = 4$, as expected when no filter is used. Also shown are plots depicting the number of active constraints N_{act} and the black-and-white fraction $\phi_{B\&W}$ as they change over the course of the optimisation for $m = 6$. Whereas the number of active constraints oscillates over the entire period of the search, convergence of $\phi_{B\&W}$ is fairly monotonic. The same can be noted for the two-bar truss results in Figures 9.12(d) and 9.12(e).

The optimisation of the MBB structure proves to be a much more difficult problem to solve than the optimal design of the two-bar truss, which can be appreciated by comparing their average CPU times per iteration T^I_{avg}. The dual subproblems that result for the MBB beam are more intractable than those that are formed in the two-bar truss optimisation, and take much longer to maximise. Our preliminary testing indicated that the difficulty in solving the dual is scale related, the dual surface being more badly scaled in the case of the MBB beam than for the two-bar problems.

Figure 9.15 contains minimum compliance results for the MBB beam, and we again first generate minimum compliance results without the application of stress constraints in Figures 9.15(a) and 9.15(b), using a volume fraction of $f_v = 0.5$. As was the case with the two-bar truss results, the inclusion of stress constraints yields entirely different topologies, as is evident in Figures 9.15(c) and 9.15(d). In particular, the algorithm appears to prefer arching the bottom of the beam when stress constraints are present, at the expense of compliance.

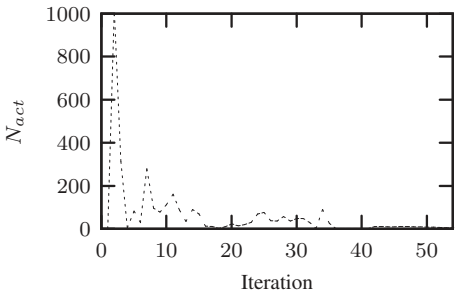
Figures 9.15(e) and 9.15(f) depict the optimal topologies generated for the stress-constrained compliance minimisation of the MBB beam when Sigmund's mesh independence filter is included as a restriction method (refer to Chapter 3) to filter the objective function. As required, the addition of the filter yields mesh-independent results. Although the compliance values are not influenced detrimentally (at least for $m = 6$), the filtered solutions have comparatively low black-and-white fractions. Note that neither of the optimisation runs in which the filter was used was able to satisfy the convergence criterion employed. Both were terminated artificially after 250 iterations. This skews T^I_{avg} somewhat, as the majority of the later iterations are run very quickly relative to the initial iterations.

Lastly, Figures 9.15(g) and 9.15(h) show minimum compliance results generated using a limiting volume fraction of $f_v = 0.36$. No filter is used, and the results may be compared with the minimum weight topologies. Strangely, the volume constraint is not active for either of the topologies depicted. Instead, local minima with volume fractions near $f_v = 0.34$ are located. Both minimum compliance results have smaller volume fractions than the corresponding minimum weight results, and the minimum weight design for $m = 6$ has a lesser compliance than the minimum compliance result for $m = 6$. This illustrates the difficult, multimodal nature of both the minimum weight and minimum compliance problems, and the improbability of finding true global minima.

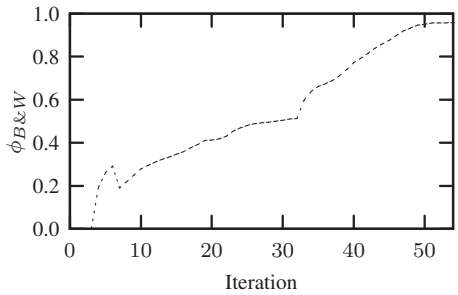
Weight minimisation

(a) Result for $m = 4$: $f_v = 0.340$, $f_c = 317.89$, $N_{act} = 54$, $n_{[1]} = 205$, $n_{[i]} = 242$, $\phi_{B\&W} = 0.798$, $T^I_{avg} = 35.5$, $N_{iter} = 88$, $n = 1200$

(b) Result for $m = 6$: $f_v = 0.353$, $f_c = 311.09$, $N_{act} = 94$, $n_{[1]} = 565$, $n_{[i]} = 482$, $\phi_{B\&W} = 0.821$, $T^I_{avg} = 229.4$, $N_{iter} = 99$, $n = 2700$



(c) Convergence plot $m = 6$: Number of active constraints N_{act}



(d) Convergence plot $m = 6$: Black and white fraction $\phi_{B\&W}$

Figure 9.14: Optimum topologies generated by weight minimisation of the MBB beam structure. Both were produced using machine $\mathbf{M_2}$.

Minimum compliance

(a) No stress con. $m = 4$: $f_v = 0.50$, $f_c = 189.69$, $N_{act} = 1$, $n_{[1]} = 543$, $n_{[i]} = 67$, $\phi_{B\&W} = 0.944$, $T^I_{avg} = 0.0$, $N_{iter} = 45$, $n = 1200$

(b) No stress con. $m = 6$: $f_v = 0.50$, $f_c = 189.05$, $N_{act} = 1$, $n_{[1]} = 1291$, $n_{[i]} = 68$, $\phi_{B\&W} = 0.975$, $T^I_{avg} = 0.0$, $N_{iter} = 33$, $n = 2700$

(c) With stress con. $m = 4$: $f_v = 0.50$, $f_c = 194.31$, $N_{act} = 3$, $n_{[1]} = 564$, $n_{[i]} = 41$, $\phi_{B\&W} = 0.966$, $T^I_{avg} = 101$, $N_{iter} = 54$, $n = 1200$

(d) With stress con. $m = 6$: $f_v = 0.50$, $f_c = 196.09$, $N_{act} = 27$, $n_{[1]} = 1299$, $n_{[i]} = 56$, $\phi_{B\&W} = 0.979$, $T^I_{avg} = 408$, $N_{iter} = 74$, $n = 2700$

(e) With filter. $m = 4$: $f_v = 0.50$, $f_c = 196.65$, $N_{act} = 7$, $n_{[1]} = 408$, $n_{[i]} = 569$, $\phi_{B\&W} = 0.526$, $T^I_{avg} = 18.95$, $N_{iter} = 250$, $n = 1200$

(f) With filter. $m = 6$: $f_v = 0.50$, $f_c = 196.49$, $N_{act} = 17$, $n_{[1]} = 935$, $n_{[i]} = 1210$, $\phi_{B\&W} = 0.552$, $T^I_{avg} = 139.98$, $N_{iter} = 250$, $n = 2700$

(g) With stress con. $m = 4$: $f_v = 0.336$, $f_c = 311.5$, $N_{act} = 101$, $n_{[1]} = 217$, $n_{[i]} = 229$, $\phi_{B\&W} = 0.809$, $T^I_{avg} = 30.55$, $N_{iter} = 161$, $n = 1200$

(h) With stress con. $m = 6$: $f_v = 0.34$, $f_c = 323.04$, $N_{act} = 118$, $n_{[1]} = 504$, $n_{[i]} = 517$, $\phi_{B\&W} = 0.809$, $T^I_{avg} = 286.89$, $N_{iter} = 114$, $n = 2700$

Figure 9.15: Optimal topologies generated by compliance minimisation of the MBB beam structure. All were run on computer $\mathbf{M_2}$.

9.8 Conclusions and recommendations

The optimisation of large-scale problems using a dual SAO approach has been studied by applying the approach to two well-known structural optimisation problems. These are the minimum weight and minimum compliance problems, and local stress constraints have been included in both. Two standard ground structures are considered: the two-bar truss and the MBB beam. The problems are large-scale in the sense that they have as many constraints as primal variables, and the sensitivities of the constraints are not straightforward to evaluate. To calculate the vector of stress sensitivities for an element requires the solution of the finite element system for the structure, with a different pseudo-load vector in place of the structural loads (we use the adjoint method to obtain the stress sensitivities). In a dense implementation it therefore would be necessary to run $n+1$ finite element analyses per iteration of the optimisation algorithm simply to define the SAO subproblem. The dual of the subproblem is also potentially very large, there being one dual variable for every constraint. The large number of constraints therefore turns the optimisation of a relatively simple structure into a cumbersome enterprise, heavily demanding of computational resources.

To solve the problems we have implemented various techniques with a view to minimising the computing resources required. Firstly, the SAO algorithm utilised, called SAOi, uses separable quadratic approximations with diagonal Hessian matrices to construct the subproblems, using only (up to) first-order information from the original problem. Since the Hessian matrices are diagonal, an n-vector is stored for the curvatures, rather than an $n \times n$ matrix. The type of quadratic approximations used in SAOi still allow the local monotonic behaviour of the structural responses to be satisfactorily represented. Secondly, we have used an efficient FORTRAN-based finite element package called EDSAP that allows access to its source code. EDSAP uses an FE solver that performs multiple forward reduction and back substitution steps when multiple load cases are present, using the same factorised global stiffness matrix resident in memory. We make use of this to reduce the required memory usage entailed in defining and storing the pseudo-load vectors required by the adjoint method.

To reduce the memory requirements further, and to reduce the size of the dual subproblems, a constraint selection strategy has been implemented so that only the gradient vectors of the active and near-active constraints are evaluated and passed to the optimiser. Additionally, it is proposed that insignificant elements of these gradient vectors can simply be omitted, so that only the 'large' partial derivatives are saved and passed to the optimiser to be used in the definition of the subproblems. We have shown that the combined application of constraint selection and constraint Jacobian filtering results in smaller dual subproblems that are solved more easily than those that result from a dense implementation. SAOi has a sparse dual solver, so full advantage can be taken of these two heuristics, and the combined application of the two resulted in a four-fold decrease in the solution time for the test problem considered.

Despite the above, the largest problem we present for weight minimisation has 14700 elements, and the largest for compliance minimisation has 10800. Both are solutions for the two-bar truss ground structure. Although these are larger than similar problems presented in the relevant literature (that we are aware of), they are not representative of the requirements of industry, particularly given the time required to solve them. The MBB beam is more difficult to solve. For both weight minimisation and compliance minimisation, the largest results we present for the MBB beam have 2700 elements.

Further gains can be made by incorporating other methods that allow for the size of potential problems to be increased. First and foremost, in our opinion it is necessary to run these types of problems on parallel computing infrastructures. Many of the processes involved in solving the problems discussed herein are parallelisable, from the large numbers of matrix and vector multiplications inherent in the dual SAO, to the definition of the pseudo-load vectors and subsequent solution of the stress sensitivities using the adjoint method.

Another line of research lies in the utilisation of element patches, which is the definition of subdomains constituting many elements, and whose important responses are a condensation of the responses of the underlying elements in some way. It is the responses of these patches that are then constrained, resulting in a reduction in the size of the optimisation problem. An extrapolation of this is the definition of global stress constraints, wherein the critical state of the structure is reduced to one global measure, instead of being reflected by many local ones.

Otherwise, and perhaps specifically for material distribution problems, there is the possibility of element deletion together with selective mesh refinement. That is, to begin by solving a problem using a coarse mesh and then to increase the mesh refinement in stages. In so doing, however, one would delete the elements constituting the holes in the structure and allow greater detail to be sought within the solid domain, starting the more refined problem from a topology derived from the result with a coarse discretisation. Ordinarily, the argument against such a strategy would be that, in deleting elements, one constrains the optimal topologies to be similar to the topology found using a coarse mesh, which might not reflect the true optimum. However, in the context of the restriction methods used to combat mesh dependence, element deletion is defensible. It is a more forceful method of ensuring that the basic structure remains the same upon mesh refinement, but it permits greater detail to be defined in the solid areas. In combination with this there is also the possibility of leaving elements of larger size within the solid areas wherever the stress is fairly uniform, and in so doing reducing the number of degrees of freedom of the resultant analysis model.

Apart from attempting to use the dual SAO method for the solution of large problems, the work presented here has also been concerned with the quality of the solutions obtained. The importance of finding solid-void solutions is reiterated, both because the spatially discretised material distribution problem is a discrete one, and because it is difficult to assess and (particularly) to compare different solutions that have material of intermediate density whose material properties are not physical. Hence, we point out that a need remains to increase the black-and-white fraction of the solutions, especially since the black-and-white fraction is often dominated by the number of elements on their lower bound densities, and that there can be roughly the same number of solid elements as elements of intermediate density, even in solutions with high black-and-white fractions.

To this end, we introduce a different method of continuation on stress relaxation, which is contrary to the conventional ‘theoretically defensible’ one. Whereas the conventional method of continuation is motivated by the need to make sure that the feasible region and KKT points of the stress-relaxed problem are ultimately the same as the relaxed continuous (but not stress-relaxed) topology problem considered, our method is motivated more by the desire to achieve designs of higher black-and-white fraction. In our opinion, it may be important to encourage convergence to the KKT points of the standard SIMP-relaxed continuous problem, but only if those KKT points

themselves coincide with (or closely approximate) solutions of the original discrete problem. Otherwise, other (perhaps heuristic) methods of finding good solutions with high black-and-white fractions cannot be ruled out as inferior. We show that the proposed new continuation on the stress relaxation does produce high-quality results, at least competitive with the standard method of relaxation.

Chapter 10

Conclusion

Problems in structural optimisation have many characteristics that make them potentially very difficult to solve. Firstly, they are simulation-based problems, meaning that the evaluation of the objective or constraint functions requires that an analysis be performed to determine the structural responses, and this can be very time consuming. Secondly, the optimisation problem may depend on a large number of variables and, furthermore, may be subject to a large number of constraints. Both of these, together with the requirements of the structural analysis, result in structural optimisation problems being heavily demanding of computational resources, both in terms of the memory required to store the description of the structure and optimisation problem, as well as the processing ability required to manipulate the equations in the structural analysis and optimisation procedures.

The optimisation problems themselves are often also inherently difficult to solve, falling under the general categorisation of mixed integer nonlinear programming (MINLP) problems. However, the underlying nature of the structural responses can often be exploited to develop efficient optimisation procedures. One class of procedures that is now widely used is dual sequential approximate optimisation (SAO), in which a series of surrogate approximate subproblems is constructed and solved to iteratively converge to the optimum of the actual system. The subproblems are solved using a dual solver. This type of procedure has proved to be comparatively efficient and has therefore seen widespread application. Briefly, the efficiency of such algorithms can be attributed to the following:

1. From the perspective of numerical optimisation, the solution of NLP problems by means of solving a sequence of surrogate approximate problems with simple structures is recognised as a very efficient procedure, provided that the global convergence characteristics can be controlled. Indeed, this procedure underlies some of the most successful optimisation algorithms available, sequential quadratic programming perhaps being the foremost example.

2. The dependence of the structural responses on the design variables is known, at least to first order, for the standard structural problems. This dependence is built into the SAO subproblems used for structural optimisation by informing the selection of the approximating functions used in the construction of the subproblems. Thus, the local responses of the system can be closely approximated.

3. The number of constraints that need be considered is often less than the number of design variables. Even in cases where the structure is subject to a large number of constraints, like local stress constraints, active set strategies can be used to limit the number of constraints that are included in the definition of the subproblems. Hence, the dimension of the dual subproblem is usually less than (and sometimes considerably less than) the dimension of the primal subproblem. Also, the dual usually has a much simpler structure than the primal, being concave and simply constrained. For these reasons, the use of a dual solver often proves to be more efficient than the use of primal solution algorithms.

4. In using dual SAO, one has some ability to limit the necessitated computational storage requirements. For instance, subproblems can be constructed from first-order separable approximations that are fairly accurate locally, instead of having to use higher-order functions with densely populated Hessians (or even higher-order curvature information). The size of the dual can be controlled using an active set strategy, and the Falk dual formulation can be used so that bound constraints are handled efficiently.

The work that has been presented in this dissertation investigates the application of the dual SAO approach to specific structural optimisation problems. Various extensions to traditional SAO implementation are suggested, which serve to increase the efficiency of the algorithm for these problems. All of the work presented assumes the use of an SAO algorithm that constructs separable primal subproblems, and a dual solver that utilises Falk's definition of the dual. Contributions have been made in areas pertaining to each of the four points listed above.

Firstly, concerning point 1, a method has been suggested that allows global convergence to be achieved through the utilisation of conservative convex and separable approximations, but without the necessity of first having to relax the subproblems in order to ensure that they are feasible. The approach, termed the 'bounded dual', utilises a trivial extension of the standard Falk dual in which the dual surface is maximised subject to the addition of sufficiently large upper bound constraints on the Lagrange multipliers. It is argued that, when a subproblem is infeasible, the bounded dual can be interpreted as a penalty formulation in which a linear combination of the constraint infeasibilities is minimised. A proof is presented that shows that the iterative use of the bounded dual results in a restorative sequence that gains feasibility, whereafter convergence to a KKT point of the problem is assured by the normal working of the CCSA scheme. Implementation of the bounded dual in numerical examples has shown it to be a viable and easily implemented alternative to relaxation in the context of a CCSA strategy.

Point 2 is augmented by the inclusion of nonconvex functions in the formulation of the subproblems for two material distribution problems. The minimum compliance problem is solved in combination with volumetric penalisation, in which the volume constraint is formulated as a power-law SIMP-like function that results in a nonconvex feasible region. It is observed that, despite being concave, the constraint can be included in the approximate subproblems without affecting the uniqueness of the primal and dual solutions, or their correspondence. Furthermore, it is argued that the direct use of the nonconvex function can result in a more efficient solution strategy, relative to the standard practice of constructing strictly convex approximations to nonconvex (or even concave) behaviour. Numerical testing has supported this conlusion. The nonconvex SIMP-like volumetric penalisation method that is implemented in the numerical test problems yields very high black-and-white fractions for the optimal designs produced when a continuation strategy is

implemented on both the curvatures of the volume constraint and the objective function.

The impetus for investigating whether nonconvex approximations can be used in the construction of the subproblems in a dual SAO method stems from the observation that the structural responses themselves sometimes suggest simple nonconvex dependencies, and we point out under what conditions the dual formulation can accommodate nonconvex forms. In these instances it may be counterproductive to ignore these in favour of popular, strictly convex functions. The weight minimisation problem is discussed as a second example, in which the first-order behaviour of the stress or displacement constraints may be concave. More generally, we present a number of methods of mixed variables for the construction of separable subproblems in dual SAO that derive from the higher-order separable exponential function, which is generally nonconvex.

Finally, the link between convex transformability for a nonconvex problem and the ability to use dual SAO to solve the problem directly (in its nonconvex form) is investigated in the context of separable problems. It is concluded that the dual of such a separable nonconvex problem is identical to the dual of its convex transform, provided it permits such a transform.

With reference to point 3, a large-scale problem is explored concerning the optimal design of orthotropic FRC plates. Solid-void minimum compliance design is combined with the simultaneous selection of the optimum point-wise fibre orientation throughout the planar structure. The problem is solved in a discrete sense, and its formulation, based on the application of the technique of discrete material optimisation, gives rise to a dual with a heavily decoupled, partially separable and piecewise-linear structure. Thus, even though the dual subproblems have a higher dimension than the primal subproblems in this case, we are able to take advantage of the peculiar structure of the dual to devise an efficient method for its maximisation. Due to its decoupled structure, maximising the dual involves the maximisation of $n + 1$ two-dimensional piecewise-linear surfaces, each of which shares the independent variable λ_{n+1} (n being the number of primal variables). Said maximisation can be accomplished efficiently using nothing more complicated than a linesearch strategy. The separable dual stems from a novel application of discrete material optimisation, and the results that are generated represent a very large application of the discrete dual in terms of the size of both the primal and dual subproblems.

Large-scale stress-constrained minimum weight and minimum compliance problems are also addressed, which pertains to point 4. These problems are defined in the continuous real space, and no special structure for the dual can be taken advantage of to make the solution strategy more efficient. Instead, the SAOi algorithm is used, which constructs separable quadratic approximations to the problem that are able to represent the local monotonic behaviour of the structural responses very well. The algorithm utilises a sparse dual solver, and the gradients of the constraints can be stored in sparse form. Therefore, we implement a novel strategy that omits inconsequential elements of the Jacobian of the constraints, and it is verified that this does not adversely affect the convergence of the algorithm to solutions of the problem, provided the Jacobian elements are not filtered too aggressively. Additionally, a constraint selection strategy is used that includes only the active and near-active constraints in the definition of the subproblems. Together with the exclusive use of separable quadratic approximations, these two algorithmic expedients result in substantial reductions in the amount of information that needs to be stored and manipulated for the definition and solution of the SAO subproblems. A non-standard method of stress relaxation is also presented, as it appears to be advantageous in generating topologies with increased black-and-white

fractions. Results are presented for the weight minimisation and compliance minimisation of both large two-bar truss and MBB beam test problems.

Lastly, although not related directly to the subject of dual SAO, a chapter is presented regarding the interpretation of Sigmund's mesh independence filter, which is widely used as a restriction method in topology optimisation. An interpretation is suggested that stems from the application of SAO to such problems in which the filter is used, and we suggest that the filter actually forms part of the definition of the approximate subproblems. The subproblems that result from the application of the filter are not first-order accurate. Moreover, it is noted that the gradient field obtained by applying the filter to the sensitivities of the actual compliance objective in a minimum compliance problem does not correspond to any scalar objective function. Some thoughts of what this might mean for the convergence of SAO algorithms that use the filter are given, as are suggestions for further research.

Suggestions for future research

The research presented in this dissertation has touched on and contributed to various topics that fall under the general heading of "the application of dual sequential approximate optimisation to structural optimisation problems". The theory underlying these procedures (at least those discussed herein, and particularly in the context of its application to structural optimisation problems) has been around for upwards of thirty years. Falk developed his dual formulation in the late 1960s. Fleury developed the idea of SAO based on convex and separable approximations that reflect the sensitivities of important structural responses in the late 1970s and early 1980s. He also addressed the weight minimisation problem using the natural nonconvex form of the displacement constraints at around the same time. Though they are continually being specialised, the procedures for locating the dual maximum, such as the method of feasible directions and the quasi-Newton methods, are essentially even older. This in itself speaks of the robust nature and efficacy of the dual SAO strategy.

However, the fact that the well-known dual SAO algorithms like CONLIN and MMA, which are continually referenced in the structural optimisation literature, have existed for some time, indicates that there is diminishingly little scope for further advantages to be gained by developing new SAO strategies. Having said this, the size and complexity of the structural optimisation problems that can be attempted in a reasonable time and with reasonable computing resources is still disappointingly small, while the use of structural optimisation in industry is probably not as widespread as it should be. Thus, in my opinion, the most pertinent research in future will likely be directed towards the computational aspects of tackling larger and more representative problems, taking increasing advantage of (for example) the promise offered by sparse implementations and (particularly) parallelisation. Along with this, further theoretical work is required to devise problem formulations that can profit from the advances made in improving the computational aspects (formulations that are more amenable to parallelisation and solvers that better utilise parallelisation, for instance).

In addition, there are certainly gains to be made by finding ways of reducing the size of the subproblems in SAO. The development of better formulations for global stress constraints and the

grouping of elements and their important responses into multiple-element patches are examples. Such effort would result in subproblems that may not be strictly first-order accurate, in which case their effect on the convergence characteristics of an algorithm would also need to be understood. In general one might ask how much information could be omitted in a system of approximation before one loses the ability to find solutions to the actual problem.

In terms of the topics addressed in this document, some are in areas where further research is demanded, while others simply represent problem-specific strategies that can be applied fruitfully in certain instances. They are each touched on below.

Sigmund's mesh independence filter is used throughout the topology optimisation community and offers a way of combating mesh dependence that is straightforward to implement. It is, however, not yet properly understood and, given its widespread use, understanding the filter remains an open and provocative research question. It is argued herein that the use of the filter on the sensitivities of the objective function does not cause a modified objective function to be solved (in compliance minimisation). It is also noted that the filter disturbs the first-order accuracy of the approximate subproblems that are generated in an SAO solution strategy, which means that the convergence characteristics of SAO algorithms using the filter are questionable. Certainly, further research into understanding the action of the filter is merited.

The discrete dual has limited application as a general method for integer programming, for the reasons given in the text. However, it is a useful method for solving binary discrete problems of large dimensionality, and it is probably the only method available for material distribution problems that guarantees purely solid-void solutions. Research is warranted primarily in three areas. Firstly, in developing convenient and efficient primal-dual mappings from a broader range of approximation functions. Secondly, in developing efficient and stable methods of controlling the global convergence of the algorithm in both the fully discrete as well as the mixed integer settings. Lastly, the application of the discrete dual to problems with multiple constraints and the development of maximisation schemes that can cope efficiently with the unique faceted structure of the dual surface, particularly if the dual has large dimensionality and is not separable.

The use of nonconvex approximations in a dual SAO procedure is very problem specific. Certainly, for some applications, nonconvexity can be exploited to reduce the number of iterations necessary to solve a problem. Better local approximations allow larger step sizes to be taken before upsetting the stability of the global convergence of the algorithm. The methods given herein for nonconvex approximation all yield subproblems that are convex transformable, so they can be used as general methods of approximation in SAO. However, in most instances the approximations give rise to primal-dual relationships that must be solved using a line search. Unless they happen to track the local behaviour of the actual problem well, there is little to be gained by incorporating them into SAO. That said, the local behaviour of some structural dependencies are known a priori, and this is no doubt true of problems in other fields. Hence, future development of nonconvex approximation strategies lies in the recognition and exploitation of the nonconvexities inherent in specific problems.

From a theoretical perspective, it remains to be shown whether convex transformability is a requirement to allow a nonconvex problem to be solved via the Falk dual. Falk's original paper on the subject does not limit the dual to strictly convex problems, but then the dual is not necessarily uniquely defined and discontinuities may enter into the formulation. We have shown that noncon-

vex problems that possess a convex transform can be solved using the dual, without transformation. We do not know whether nonconvex problems exist that are not convex transformable but that can still be solved (practicably) using the dual. Also, we have confined out attention to separable problems, and it would be interesting if the arguments were to be extended to non-separable problems. This most likely would only be of theoretical interest, however, since separability itself is crucial to the efficient practical implementation of dual solvers in SAO. Also, equality constraints are infrequently, if ever, included in the dual SAO approach in structural optimisation, and this is an area that could profitably be pursued.

The bounded dual is already in use in the SAOi optimiser, as part of the mechanism that encourages global convergence. It is extremely straightforward to implement, and seems to handle infeasibility very well. Herein we have presented an explanation of how it does so in the context of a global convergence scheme based on the use of conservatism, but it can equally well be used with other schemes, such as within a trust region infrastructure. Theoretical justification of its efficacy within other schemes is still required, and is deserved of some attention.

Bounding the dual cannot be said to work well for all problems. As with the maximisation of the dual itself, the efficacy of employing the bounded dual is probably related to the scaling of the dual, which, in terms of the efficient solution of dual subproblems, is really the area that merits further attention. Whereas relaxation affects the dual scaling, and therefore may make the dual easier to maximise in some instances, bounding the dual does not affect the shape of the dual surface. It has been our experience that there can be a significant variation in the times necessary to solve successive dual subproblems in an SAO infrastructure, and preliminary investigations (in the context of the solution of large-scale problems) have pointed to this being related to scaling. Given the importance of locating the dual maximum accurately, efficient methods of preconditioning or otherwise improving the condition of the dual are necessary, and constitute an important avenue for future enquiry.

Lastly, the work presented in this document has been concerned with the application of dual SAO techniques to large-scale problems. It is in this area that advances are required to make the use of structural optimisation more attractive and more feasible in industrial applications. The developments in computing are continually making the use of optimisation procedures more practicable, and the procedures themselves need to be designed to make optimal use of the available computing resources. Advances can be made in the increased utilisation of parallelisation as well as sparse computing methods, and the dual SAO algorithms should be developed to take better advantage of these. Additionally, avenues may be followed in the formulation of the subproblems that can be used to reduce their potential size. Formulations of global stress constraints or element patches are existing examples that merit further investigation, as are robust methods of constraint selection. Along these lines, the ideas that have been advanced herein are the formulation of partially separable duals (where possible) and filtering of the constraint sensitivities. This last may be generalised to the construction of subproblems that are not first-order accurate, but that have increased sparsity. The limits and viability of this concept still require investigation.

References

[1] C. Fleury. A unified approach to structural weight minimization. *Comp. Meth. Applied Mech. Eng.*, 20:17–38, 1979.

[2] J.E. Falk. Lagrange multipliers and nonlinear programming. *J. Math. Anal. Appls.*, 19:141–159, 1967.

[3] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *Int. J. Numer. Meth. Eng.*, 24:359–373, 1987.

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

[5] C. Fleury. Conlin: an efficient dual optimizer based on convex approximation concepts. *Struct. Optim.*, 1:81–89, 1989.

[6] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J. Optim.*, 12:555–573, 2002.

[7] G.I.N. Rozvany. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Struct. Multidisc. Optim.*, 21:90–108, 2001.

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

[9] O. Sigmund. A 99 line topology optimization code written in Matlab. *Struct. Multidisc. Opt.*, 21:120–127, 2001.

[10] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Opt.*, 16:68–75, 1998.

[11] B. Bourdin. Filters in topology optimization. *Int. J. Numer. Meth. Eng.*, 0:1–17, 2000.

[12] O. Ambrosio and G. Buttazzo. An optimal design problem with perimeter penalization. *Calc. Var.*, 1:55–69, 1993.

[13] J. Petersson and O. Sigmund. Slope constrained topology optimization. *Int. J. Numer. Meth. Eng.*, 41:1417–1434, 1998.

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

[15] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Mech. Struct. Machines*, 25:495–526, 1997.

[16] L. Schmit and C. Fleury. Discrete-continuous variable structural synthesis using dual methods. *AIAA Journal*, 18:1515–1524, 1980.

[17] M. Beckers. Topology optimization using a dual method with discrete variables. *Struct. Optim.*, 11:102–112, 1996.

[18] M.P. Bendsøe. Optimal shape design as a material distribution problem. *Struct. Optim.*, 1:193–202, 1989.

[19] G.I.N. Rozvany and M. Zhou. Applications of COC method in layout optimization. In H. Eschenauer, C. Mattheck, and N. Olhoff, editors, *Proc. Engineering Optimization in Design Processes*, pages 59–70, Berlin, 1991. Springer-Verlag.

[20] J. Petersson. Some convergence results in perimeter-controlled topology optimization. *Comp. Meth. Applied Mech. Eng.*, 171:123–140, 1999.

[21] R. Fletcher, S. Leyffer, and P.L. Toint. On the global convergence of an SLP-filter algorithm. Technical Report 00/15, Department of Mathematics, University of Namur, Namur, Belgium, 1998.

[22] G. Hadley. *Nonlinear and Dynamic Programming*. Addison-Wesley, Reading, Massachusetts, 1964.

[23] K. Schittkowski. Optimization in indistrial engineering: Sqp-methods and applications. Technical report, Radioss user meeting, Mecalog, Nice, June 2005.

[24] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-region methods*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, 2000.

[25] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91:239–269, 2002.

[26] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, Massachusetts, 1999.

[27] R.T. Haftka and Z. Gürdal. *Elements of structural optimization*, volume 11 of *Solid Mechanics and its applications*. Kluwer Academic Publishers, Dordrecht, the Netherlands, third edition, 1991.

[28] C. Fleury. Structural weight optimization by dual methods of convex programming. *Int. J. Numer. Meth. Eng.*, 14:1761–1783, 1979.

[29] R. L. Barnett. Minimum weight design of beams for deflection. *J. Eng. Mech. Div.*, ASCE 87:75–109, 1961.

[30] L. Berke. An efficient approach to the minimum weight design of deflection limited structures. Technical Report AFFDL-TM-70-4-FDTR, (USAF Tech. Mem.), 1970.

[31] A.A. Groenwold, D.W. Wood, L.F.P. Etman, and S. Tosserams. Globally convergent optimization algorithm using conservative convex separable diagonal quadratic approximations. *AIAA J.*, 47:2649–2657, 2009.

[32] K. Svanberg. A globally convergent version of MMA without linesearch. In G.I.N. Rozvany and N. Olhoff, editors, *Proc. First World Congress on Structural and Multidisciplinary Optimization*, pages 9–16, Goslar, Germany, 1995.

[33] A.A. Groenwold, L.F.P. Etman, and D.W. Wood. Approximated approximations for SAO. *Struct. Mult. Optim.*, 41:39–56, 2010.

[34] P. Wolfe. A duality theorem for nonlinear programming. *Q. Appl. Math.*, 19:239–244, 1963.

[35] D.W. Wood and A.A. Groenwold. On concave constraint functions and duality in predominantly black-and-white topology optimization. *Comp. Meth. Appl. Mech. Eng.*, 199:2224–2234, 2010.

[36] D.W. Wood and A.A. Groenwold. Non-convex dual forms based on exponential intervening variables, with application to weight minimization. *Int. J. Numer. Meth. Eng.*, 80:1544–1572, 2009.

[37] O. Sigmund. Morphology-based black and white filters for topology optimization. *Struct. Multidisc. Opt.*, 33:401–424, 2007.

[38] T.E. Bruns and D.A. Tortorelli. Topology optimization of nonlinear elastic structures and compliant mechanisms. *Comp. Meth. Appl. Mech. Eng.*, 190:3443–3459, 2001.

[39] T.E. Bruns. A reevaluation of the SIMP method with filtering and an alternative formulation for solid-void topology optimization. *Struct. Multidisc. Optim.*, 30:428–436, 2005.

[40] A.A. Groenwold and L.F.P. Etman. Duality in convex nonlinear multipoint approximations with diagonal approximate Hessian matrices deriving from incomplete series expansions. In *Proc. 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, U.S.A., September 2006. Paper AIAA-2006-7090.

[41] A.A. Groenwold and L.F.P. Etman. Sequential approximate optimization using dual subproblems based on incomplete series expansions. *Struct. Multidisc. Opt.*, 36:547–570, 2008.

[42] M.P. Bendsøe. *Optimization of structural topology, shape and material*. Springer, Berlin, 1995.

[43] A.A. Groenwold and L.F.P. Etman. On the equivalence of optimality criterion methods and sequential approximate optimization in the classical topology layout problem. *Int. J. Numer. Meth. Eng.*, 73:297–316, 2008.

[44] G.M. Fadel, M.F. Riley, and J.M. Barthelemy. Two point exponential approximation method for structural optimization. *Struct. Optim.*, 2:117–124, 1990.

[45] A.A. Groenwold and L.F.P. Etman. A simple heuristic for gray-scale suppression in optimality criterion-based topology optimization. *Struct. Multidisc. Opt.*, 39:217–225, 2009.

[46] R. Ellis and D. Gulick. *Calculus with analytical geometry*. Harcourt Brace and Company, Orlando, Florida, fifth edition, 1994.

[47] P. O'Neil. *Advanced engineering mathematics*. PWS Publishing Company, Boston, MA, fourth edition, 1995.

[48] D.W. Wood and A.A. Groenwold. Optimization of constrained mixed discrete continuous composite problems via a dual method of sequential approximate optimization. In *CSIR Fibre Reinforced Composites Conference*, pages 117–131, Port Elizabeth, South Africa, December 2007. Paper 83.

[49] S. Venkataraman and R.T. Haftka. Optimization of composite panels – a review. In *Proc. of the American Society of Composites, 14th Annual Technical Conference*, pages 479–448, Dayton, Ohio, 1999.

[50] A.A. Groenwold and R.T. Haftka. Optimization with non-homogeneous failure criteria like Tsai-Wu for composite laminates. *Struct. Multidisc. Opt.*, 32:183–190, 2006.

[51] D.O. Evans, M.M. Vaniglia, and P.C. Hopkins. Fiber placement process study. In *Proc. 34th International SAMPE Symposium and Exhibition*, pages 1822–1833, Reno, 1989.

[52] G.S. Landriani and M. Rovati. Optimal design for two-dimensional structures made of composite materials. *Trans. ASME*, 113:88, 1991.

[53] P. Pedersen. On thickness and orientational design with orthotropic materials. *Struct. Optim.*, 3:69, 1991.

[54] W. Hansel and W. Becker. Layerwise adaptive topology optimization of laminate structures. *Eng. Comput.*, 16:841, 1999.

[55] G. Duvaut, G. Terrel, F. Léné, and V.E. Verijenko. Optimization of fiber reinforced composites. *Composite Structures*, 48:83, 2000.

[56] S. Setoodeh, M.M. Abdalla, and Z. Gürdal. Combined topology and fiber path design of composite layers using cellular automata. *Struct. Multidisc. Opt.*, 30:413, 2005.

[57] J. Stegmann and E. Lund. Discrete material optimization of general composite shell structures. *Int. J. Numer. Meth. Eng.*, 62:2009, 2005.

[58] O. Sigmund and S. Torquato. Design of materials with extreme thermal expansion using a three-phase topology optimization method. *Journal of the Mechanics and Physics of Solids*, 48:461, 2000.

[59] E. Salajegheh. Discrete variable optimization of plate structures using dual methods. *Comput. & Struct.*, 58:1131, 1996.

[60] A. Sepúlveda and J. Cassis. An efficient algorithm for the optimum design of trusses with discrete variables. *Int. J. Num. Meth. Eng.*, 23:1111–1130, 1986.

[61] A.A. Groenwold, L.F.P. Etman, J.A. Snyman, and J.E. Rooda. Incomplete series expansion for function approximation. *Struct. Multidisc. Opt.*, 34:21–40, 2007.

[62] M. Zhou and G.I.N. Rozvany. The COC method. Part II. Topological, geometrical and generalized shape optimization. *Comp. Meth. Appl. Mech. Eng.*, 40:1–26, 1991.

[63] J.M. Guedes and J.E. Taylor. On the prediction of material properties and topology for optimal continuum structures. *Struct. Optim.*, 14:193–199, 1997.

[64] A. Rietz. Sufficiency of a finite exponent in SIMP (power law) methods. *Struct. Multidisc. Optim.*, 21:159–163, 2003.

[65] G.I.N. Rozvany, M. Zhou, and O. Sigmund. Optimization of topology. In H. Adeli, editor, *Advances in design optimization*. Chapman & Hall, London, U.K., 1994.

[66] L. Wang and R.V. Grandhi. Efficient safety index calculations for structural reliability analysis. *Comp. Struct.*, 52:103–111, 1994.

[67] L. Wang and R.V. Grandhi. Improved two-point function approximation for design optimization. *AIAA J.*, 33:1720–1727, 1995.

[68] S. Xu and R.V. Grandhi. Effective two-point function approximation for design optimization. *AIAA J.*, 36:2269–2275, 1998.

[69] A.A. Groenwold, L.F.P. Etman, J.A. Snyman, and J.E. Rooda. Incomplete series expansion for function approximation. In *Proc. Sixth World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, May 2005.

[70] P. Duysinx. Solution of topology optimization problems with sequential convex programming. Technical report, LTAS - Automotive Engineering, Institute of Mechanics and Civil Engineering, University of Liège, May 2008.

[71] A.A. Groenwold and L.F.P. Etman. A quadratic approximation for structural topology optimization. *Int. J. Numer. Meth. Eng.*, 82:505–524, 2010.

[72] C. Fleury and B. Fraeijs du Veubeke. Structural optimization. In *Lecture Notes in Computer Sciences*, volume 27, pages 314–326, Berlin, 1975. Springer-Verlag.

[73] J.H. Starnes Jr. and R.T. Haftka. Preliminary design of composite wings for buckling, stress and displacement constraints. *J. Aircraft*, 16:564–570, 1979.

[74] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: FORTRAN subroutines for large scale bound constrained optimization. Technical Report NAM-11, Northwestern University, EECS Department, 1994.

[75] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scient. Comput.*, 16:1190–1208, 1995.

[76] D.W. Wood and A.A. Groenwold. On convex transformability and the solution of nonconvex problems via the dual of Falk. *Struct. Mult. Optim.*, 2011. Submitted (SMO-11-0155).

[77] D. J. Estep. *Practical Analysis in One Variable*. Springer-Verlag, Berlin, 2002.

[78] D.W. Wood, A.A. Groenwold, and L.F.P. Etman. Bounding the dual of Falk to circumvent the requirement of relaxation in globally convergent SAO algorithms. *Optim. Eng.*, 2011. Submitted.

[79] J.F.M. Barthelemy and R.T. Haftka. Approximation concepts for optimum structural design - a review. *Struct. Opt.*, 5:129–144, 1993.

[80] M. Bruyneel, P. Duysinx, and C. Fleury. A family of MMA approximations for structural optimization. *Struct. Multidisc. Optim.*, 24:263–276, 2002.

[81] N.M. Alexandrov, J.E. Dennis, R.M. Lewis, and V. Torczon. A trust region framework for managing the use of approximation models in optimization. *Struct. Optim.*, 15:16–23, 1998.

[82] J.A. Snyman and A.M. Hay. The Dynamic-Q optimization method: an alternative to SQP? *Comput. Math. Appl.*, 44:1589–1598, 2002.

[83] K. Svanberg. On a globally convergent version of MMA. In *Proc. Seventh World Congress on Structural and Multidisciplinary Optimization*, Seoul, Korea, May 2007. Paper no. A0052.

[84] P. Duysinx and M. P. Bendsøe. Topology optimization of continuum structures with local stress constraints. *Int. J. Numer. Meth. Eng.*, 43:1453–1478, 1998.

[85] C. Fleury. Personal communications with A. A. Groenwold, June 2009.

[86] U. Kirsch. On singular topologies in optimum structural design. *Struct. Optim.*, 2:133–142, 1990.

[87] G. Cheng and Z. Jiang. Study on topology optimization with stress constraints. *Engng. Optim.*, 20:129–148, 1992.

[88] G. I. N. Rozvany. On design-dependent constraints and singular topologies. *Struct. Multidisc. Opt.*, 21:164–172, 2001.

[89] G.I.N. Rozvany and J. Sobieszczanski-Sobieski. New optimality criteria methods: forcing uniqueness of the adjoint strains by corner-rounding at constraint intersections. *Struct. Optim.*, 4:244–246, 1992.

[90] G. D. Cheng and X. Guo. ε-relaxed approach in structural topology optimization. *Struct. Optim.*, 13:258–266, 1997.

[91] M. Bruggi. On an alternative approach to stress constraints relaxation in topology optimization. *Struct. Multidisc. Opt.*, 36:125–141, 2008.

[92] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli. Stress-based topology optimization for continua. *Struct. Multidisc. Opt.*, 41:605–620, 2010.

[93] M. Bruggi and P. Venini. A mixed fem approach to stress-constrained topology optimization. *Int. J. Numer. Meth. Eng.*, 73:1693–1714, 2007.