University of Bath

UNIVERSITY OF BATH

PHD

Topology Optimization of Linear Elastic Structures

Browne, Philip

Award date:
2013

Awarding institution:
University of Bath

Link to publication

Topology Optimization of Linear Elastic Structures

submitted by

Philip Anthony Browne

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

May 2013

COPYRIGHT

Signature of Author . . . .

Philip Anthony Browne

Summary

Topology optimization is a tool for finding a domain in which material is placed that optimizes a certain objective function subject to constraints. This thesis considers topology optimization for structural mechanics problems, where the underlying PDE is derived from linear elasticity.

There are two main approaches for solving topology optimization: Solid Isotropic Material with Penalisation (SIMP) and Evolutionary Structural Optimization (ESO). SIMP is a continuous relaxation of the problem solved using a mathematical programming technique and so inherits the convergence properties of the optimization method. By contrast, ESO is based on engineering heuristics and has no proof of optimality.

This thesis considers the formulation of the SIMP method as a mathematical optimization problem. Including the linear elasticity state equations is considered and found to be substantially less reliable and less efficient than excluding them from the formulation and solving the state equations separately. The convergence of the SIMP method under a regularising filter is investigated and shown to impede convergence. A robust criterion to stop filtering is proposed and demonstrated to work well in high-resolution problems ($\mathcal{O}(10^6)$).

The ESO method is investigated to fully explain its non-monotonic convergence behaviour. Through a series of analytic examples, the steps taken by the ESO algorithm are shown to differ arbitrarily from a linear approximation. It is this difference between the linear approximation and the actual value taken which causes ESO to occasionally take non-descent steps. A mesh refinement technique has been introduced with the sole intention of reducing the ESO step size and thereby ensuring descent of the algorithm. This is shown to work on numerous examples.

Extending the classical topology optimization problem to included a global buckling constraint is considered. This poses multiple computational challenges, including the introduction of numerically driven spurious localised buckling modes and ill-defined gradients in the case of non-simple eigenvalues. To counter such issues that arise in a continuous relaxation approach, a method for solving the problem that enforces the binary constraints is proposed. The method is designed specifically to reduce the number of derivative calculations made, which is by far the most computationally expensive step in optimization involving buckling. This method is tested on multiple problems and shown to work on problems of size $\mathcal{O}(10^5)$.

Acknowledgements

Firstly I should thank Chris Budd for supervising me through this work, Nick Gould and Jennifer Scott from RAL for kindly sponsoring the CASE award and helping enormously with their technical knowledge, and Alicia Kim for developing my engineering abilities.

There are many other people in the numerical analysis group in RAL I would like to thank for their input, namely Jonathan Hogg, Daniel Robinson, Iain Duff and John Reid. I am indebted to all the staff in the Maths Department, particularly Melina Freitag, Euan Spence, Rob Scheichl and Alastair Spence for their insightful discussions.

This thesis would have taken significantly longer without Pete Dunning and Chris Brompton, who helped enormously with debugging and the use of their example codes. The time spent working on this thesis was made much more pleasant thanks to my office mates over the years, particularly James Lloyd, Chris Guiver, Caz Ashurst, Jane Temple and Adam Boden for their ability to distract and sporcle.

I would like to show my gratitude to my housemates throughout the time of my studies, Sean Buckeridge and Dom Parsons, for their unwavering humour and for sharing a contempt of Sunday morning bells. Rob Ellchuk and all those friends from the track have been vital for keeping me healthy and I thank them for giving me an excuse never to stay in the office past 17:30.

I would like to thank my family for supporting me through my time as a student, especially when I embarked on my PhD instead of getting a real job. Most importantly I should thank Vicki Cronin for tolerating me and constantly providing me with the best possible diversions from the monotony of writing; your love and encouragement has made this thesis possible.

Finally I should thank the reader for their interest in this thesis and I hope that, at the very least, the figures in this work will encourage reading to the very last page.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

BESO	Bidirectional Evolutionary Structural Optimization
BLAS	Basic Linear Algebra Subroutine
DAG	Directed Acyclic Graph
EQP	Equality Constrained Quadratic Program
ESO	Evolutionary Structural Optimization
FEA	Finite Element Analysis
FEM	Finite Element Method
FMO	Free Material Optimization
IQP	Inequality Constrained Quadratic Program
KKT	Karush Kuhn Tucker
LICQ	Linear Independence Constraint Qualification
LPP	Linear Programming Problem
MBB	Messerschmitt-Bölkow-Blohm
MEMS	Micro Electro Mechanical Systems
MFCQ	Mangasarian-Fromowitz constraint qualification
MMA	Method of Moving Asymptotes

NAND	Nested Analysis and Design
NURBS	Non-Uniform Rational B-Splines
OC	Optimality Criteria
PCG	Preconditioned Conjugate Gradient
PDE	Partial Differential Equation
QP	Quadratic Program
RAL	Rutherford Appleton Laboratory
SAND	Simultaneous Analysis and Design
SDP	Semidefinite Programming
SIMP	Solid Isotropic Material with Penalisation
SLP	Sequential Linear Programming
SNOPT	Sparse Nonlinear Optimizer
SPD	Symmetric Positive Definite
SQP	Sequential Quadratic Programming
VTS	Variable Thickness Sheet

1

Introduction

1.1 Motivation of the thesis

Topology optimization aims to answer the question, what is the best domain in which to distribute material in order to optimise a given objective function subject to some constraints?

Topology optimization is an incredibly powerful tool in many areas of design such as optics, electronics and structural mechanics. The field emerged from structural design and so topology optimization applied in this context is also known as *structural optimization.*

Applying topology optimization to structural design typically involves considering quantities such as weight, stresses, stiffness, displacements, buckling loads and resonant frequencies, with some measure of these defining the objective function and others constraining the system. For other applications aerodynamic performance, optical performance or conductance may be of interest, in which case the underlying state equations are very different to those considered in the structural case.

In structural design, topology optimization can be regarded as an extension of methods for size optimization and shape optimization. Size optimization considers a structure which can be decomposed into a finite number of members. Each member is then parametrised so that, for example, the thickness of the member is the only variable defining the member. Size optimization then seeks to find the optimal values of the parameters defining the members.

Shape optimization is an extension of size optimization in that it allows extra freedoms in the configuration of the structure such as the location of connections between members. The designs allowed are restricted to a fixed topology and thus can be written using a limited number of optimization variables.

Topology optimization extends size and shape optimization further and gives no
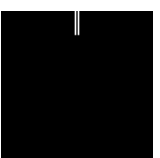
restrictions to the structure that is to be optimized. It simply seeks to find the optimal domain of the governing equations contained within some design domain.

Definition 1.1. *The* design domain *is a 2-dimensional area or 3-dimensional volume in which the optimal domain can be contained.*

To solve a topology optimization problem the design domain is discretised and the presence of material in any of the resulting divisions denotes each individual optimization variable. The goal is then to state which of the discretised portions of the design domain should contain material and which should not contain material. With the objective function denoted by ϕ and constraints on the system denoted ψ, then the topology optimization problem can be written

$$\min_{x} \quad \phi(x) \tag{1.1a}$$

$$\text{subject to} \quad \psi(x) \leq 0 \tag{1.1b}$$

$$\text{and} \quad x_i \in \{0, 1\} \tag{1.1c}$$

where $x_i = 0$ represents no material in element i of the design domain and $x_i = 1$ represents the presence of material in element i of the design domain.
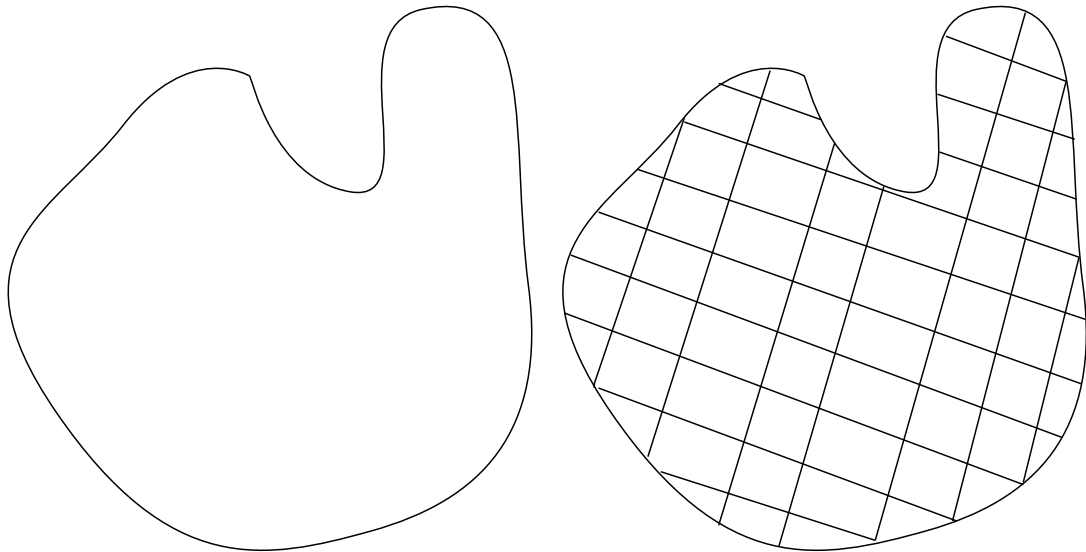
(a) Example of a 2D design domain of a topology optimization problem

(b) Example of the discretisation of a design domain of a topology optimization problem

Figure 1-1: Design domain and discretisation of a 2D topology optimization problem. The design domain is the area or volume contained within the given boundary in which material is allowed to be placed. This region is then discretised into smaller divisions within which we associate the presence of material with an optimization variable.

Figure 1-2: Design domain of the short cantilevered beam showing the applied load f and the fixed boundary conditions

This thesis is concerned with investigating the techniques and issues that arise when topology optimization is applied to structural design. In a illustrative example of this, (1.1) could have the form

$$\min_{x} \quad f^{T}u(x) \tag{1.2a}$$

$$\text{subject to} \quad \sum_{i} x_{i} - V \leq 0 \tag{1.2b}$$

$$K(x)u(x) = f \tag{1.2c}$$

$$\text{and} \quad x_{i} \in \{0,1\} \tag{1.2d}$$

where $K(x)u(x) = f$ is the finite-element formulation of the equations of linear elasticity, relating the stiffness matrix $K(x)$ and the displacements $u(x)$ resulting from an applied load f. Here the objective is minimising the *compliance* of the structure (equivalently maximising its stiffness) subject to an upper bound V on the volume of the structure.

Compliance measures the external work done on the structure. It is the sum of all the displacements at the points where the load is applied, weighted by the magnitude of the loading. Hence minimising this quantity minimises the deflection of the structure due to an applied load and thus maximises the stiffness of the structure.

There are two distinct approaches to solving this optimization problem: a continuous relaxation of the binary constraint (1.1c) which is referred to as Solid Isotropic Material with Penalisation (SIMP) and a method based on engineering heuristics referred to as Evolutionary Structural Optimization (ESO). The SIMP approach uses
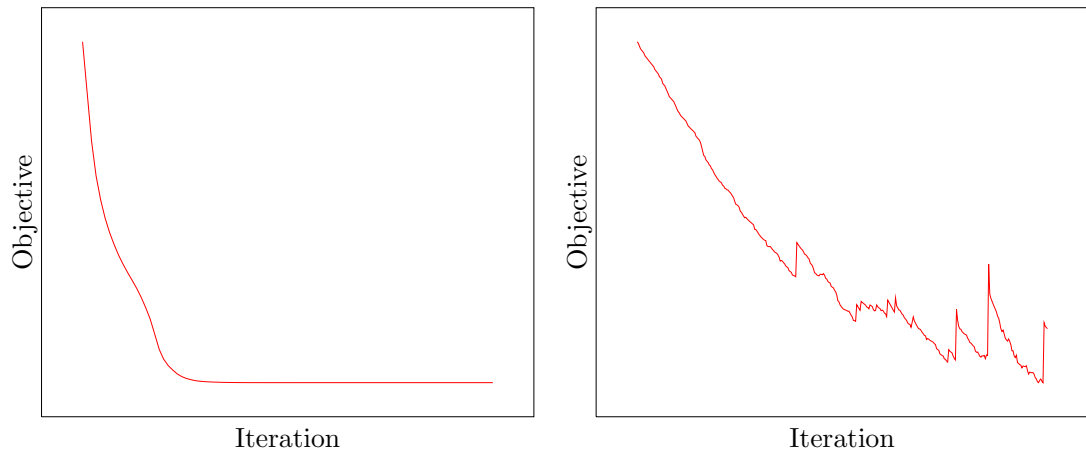
(a) Typical convergence of SIMP approach to topology optimization

(b) Typical convergence of ESO approach to topology optimization

Figure 1-3: Convergence behaviour of different approaches to topology optimization

a mathematical programming technique and so inherits the convergence properties of the optimization method used, whereas the ESO method does not have such qualities. Figure 1-2 shows a test example known as the short cantilevered beam. Figure 1-3 shows the objective function history of applying both the SIMP approach and the ESO method to the short cantilevered beam. It can be seen that the SIMP approach has monotonic convergence whereas the ESO method takes many non-descent steps. Rozvany 2008 [139] wrote a highly critical article in which the lack of mathematical theory for ESO led him to favour such methods as SIMP for topology optimization. This motivates this thesis to bring together all the existing theory for the SIMP approach and to further develop the theory of ESO.

1.2 Aims of the thesis

This thesis aims to give a formal mathematical justification to the choice of approaches used to solve topology optimization problems applied to structural design. Previous work has concentrated on comparisons between different approaches and selecting an appropriate method for a given problem. Different approaches for topology optimization will be considered in isolation and questions pertaining to them will be answered, as opposed to proposing an alternative solution method. This new in-depth knowledge of the approaches can then be used to inform the choice of approach taken to solve a structural optimization problem.

1.3 Achievements of the thesis

1. This thesis thoroughly investigates the convergence behaviour of the ESO method for topology optimization. A discrete heuristic method, ESO is seen to take non-descent steps and these have been explained by observing the nonlinear behaviour of the linear elasticity equations with respect to varying the domain of the PDE. Furthermore, this behaviour has been eradicated by introducing a simple adaptive mesh refinement scheme to allow smaller changes in the structure to be made. This is covered in Chapter 7.

2. Including the solution of the state equations in the formulation of a topology optimization problem using the SIMP approach has been implemented in multiple optimization software packages. In all cases the same difficulty in finding feasible solutions was found, and this motivated the proof that certain constraint qualifications do not hold in this formulation. This result then gives a solid justification for why removing these variables from the optimization formulation has gained prevalence over solving the same problem with them included. Poor convergence is observed when filtering is applied to regularise the problem. A robust criterion to stop filtering is proposed to recover convergence. This forms the basis of Chapter 5.

3. This thesis then considers extending the classical structural optimization problem to include a buckling constraint. This extra constraint significantly increases the difficulty of the problem when the optimization variables are relaxed to vary continuously. Spurious localised buckling modes are observed in this approach and a formal justification for a technique to eradicate them is given. This eradication technique then leads to the calculation of critical loads that are inconsistent with the underlying state equations. To avoid these issues, and to have a computationally efficient solution method for such problems, a new method designed specifically for this problem is introduced which has been published in Browne et al. 2012 [27]. This is shown in Chapter 6.

4. In the process of bringing together the theory of linear elasticity which is applicable to topology optimization, a gap has been found in the literature (Karal and Karp 1962 [84]) of the categorisation of singularities which occur at a re-entrant corner. Knowledge of these singularities is essential when analysing topology optimization methods as some authors believe them to be a source of numerical error. The classification of the singularity which occurs at a re-entrant corner is formalised in Chapter 3.

5. It has been shown that for the linear elasticity systems considered in topology optimization, direct linear algebra methods remain very effective on problems with matrices of size $\mathcal{O}(10^6)$.

These achievements have immediate importance in the engineering application of topology optimization. The most efficient and robust formulation of the general topology optimization problem as a mathematical programming problem has been stated. However, the traditional engineering approach to topology optimization is the ESO method which relied on heuristics for its justification. A simple modification to the ESO algorithm, motivated by a new understanding of its nonmonotonic convergence behaviour, then resulted in monotonic convergence to an approximate stationary point, hence verifying ESO as an optimization algorithm. This is a very important result for the community of researchers working on ESO, as previously their method had little mathematical justification.

1.4   Structure and content of the thesis

For a comprehensive view of the field of topology optimization it is necessary to bring together three key areas of science; namely, elasticity theory, engineering and optimization theory. This thesis begins by covering these areas before moving on to showing the original new work in the subsequent chapters. Hence the thesis is organised as follows.

Chapter 2 contains a comprehensive literature review of the field of structural optimization. Truss topology optimization, optimization of composites and topological derivatives are detailed in the early sections, though are not investigated in this thesis. The technique of homogenisation for structural optimization is detailed in Section 2.5 which leads into the review of the SIMP method in Section 2.6. The SAND approach to formulating the optimization problem is reviewed in Section 2.7. ESO and its successor BESO are reviewed in Section 2.8 followed by a review of the work that has been done on buckling optimization in Section 2.9. Finally in Chapter 2, this thesis examines the literature on chequerboard patterns emerging in topology optimization, symmetry properties of optimal solutions and linear algebra matters.

Chapter 3 contains the derivation and analysis of the state equations that are used to compute the response of a structure to an applied load. Starting with Newton's laws of motion, in Section 3.1 the Lamé equation is derived which is the underlying PDE to be solved. The process of discretising this PDE in a finite element context is presented for linear elasticity in Section 3.2. In Section 3.3, the conditioning of the finite element stiffness matrices are considered. The stress stiffness matrix is derived in Section 3.4, which is used to compute the linear buckling load of a structure. Section 3.5 describes

the linear algebra technique employed to find the buckling load of a structure. Finally, corner singularities inherent in the underlying equations are discussed in Section 3.6 by first considering Poisson's equation and then looking at the elasticity case.

In Chapter 4 mathematical optimization methods are surveyed, beginning with general definitions in Section 4.1. The simplex method for linear programming is discussed in Sections 4.2 and 4.3. Integer programming methods are covered in Sections 4.4 to 4.6. Nonlinear continuous programming methods are explored in Sections 4.7 to 4.11.

Chapter 5 is concerned with the formulation of structural optimization as an mathematical programming problem that can be solved efficiently using the methods of Chapter 4. Sections 5.1 and 5.2 formulate the problem in the SIMP approach. Section 5.3 discusses appropriate optimization methods to solve the mathematical programming problem. Section 5.4 investigates the possibility of including the state equations directly in the optimization formulation. Section 5.5 introduces filters in order to regularise the problem and make it well posed. Finally Section 5.6 shows the latest results in solving this particular structural optimization problem.

In Chapter 6 adding a buckling constraint to the standard structural optimization problem is considered. This adds a great deal of complexity and introduces a number of issues that do not arise in the more basic problem considered in Chapter 5. Section 6.1 introduces the buckling constraint and shows how a direct bound on the buckling constraint becomes non-differentiable when there is a coalescing of eigenvalues. Section 6.2 discusses the issues arising with spurious buckling modes. The problem is reformulated in Sections 6.3 to 6.4 and an analytic formula for the derivative of the stress stiffness matrix is presented. In Section 6.5 we then introduce a new method in order to efficiently compute a solution to an optimization problem involving a buckling constraint.

Chapter 7 is concerned with the convergence of the ESO algorithm and contains substantial new results on the topic. Section 7.1 commences the chapter by introducing the algorithm. This is followed by a typical example of the convergence behaviour of the algorithm. The choice of strain energy density as the sensitivity is demonstrated in Section 7.3. Sections 7.4 and 7.5 find analytic examples of nonlinear and linear behaviour of the linear elasticity equations respectively. A motivating example in the continuum setting is presented in Section 7.6 that shows the nonlinear behaviour of the algorithm and inspires the modified ESO algorithm which is given in Section 7.7. This modified algorithm is then applied to the tie beam problem in Section 7.8 in order to show its effectiveness.

Finally, Chapter 8 concludes the thesis by recounting the achievements and limita-

tions of the work. Ideas for future work are set out as possible topics for investigation.

2

Literature review

In this chapter the history of structural optimization will be reviewed. Starting from its beginnings with analytic optima of simple structures going through to the computational methods used to optimize complex structures, this chapter will detail the methods used and the difficulties associated with each. The theory and applications of SIMP and ESO will be detailed, followed by listing some properties of the solutions to structural optimization problems, such as ill-posedness and symmetries.

2.1 The foundations of structural optimization

Structural optimization can easily be traced back to 1904 when Michell derived formulae for structures with minimum weight given stress constraints on various design domains [112]. Save and Prager 1985 [147] proved that the resulting structures (known at the time as *Michell structures*) had the minimum compliance for a structure of the corresponding volume and hence were global optimum of minimisation of compliance subject to volume problems.

Long before this, one-dimensional problems were considered by Euler and Lagrange in the 1700s. They were interested in problems to design columns [49] or bars for which the optimal cross-sectional area needed to be determined. Euler also considered the problem of finding the best shape for gear teeth [50]. Typically an analytic solution to a structural optimization problem may only be found for very specific design domains and loading conditions such as those considered by Michell. Automating the solution of the state equations using the finite element method with computers allowed significant advances in the field of structural optimization (see for example, Schmit and Fox [148]).

In 1988, Bendsøe and Kikuchi [21] used a homogenisation method which allowed them to create microstructure in the material. This resulted in a composite-type structure where material in each element was composed of both solid material and voids.

This was the first foray into a continuous relaxation of the problem and will be discussed in Section 2.5.

In both the Solid Isotropic Material with Penalisation (SIMP) and Evolutionary Structural Optimization (ESO) approaches (which will be introduced in sections 2.6 and 2.8 respectively), the topology of the structure is typically represented by values of material in an element of a finite-element mesh. Other representations of the structure are possible, for example, using non-uniform rational B-splines (NURBS) to represent the boundary of the material. The control points of the NURBS can then be moved in order to find an optimal structure. For an example of B-spline use in shape optimization see Herskovits et al. [68]. This approach is not considered in this thesis but is covered in detail in the thesis of Edwards [47].

Level-sets are another possible way to represent the topology of a structure. In this approach an implicit functional is positive where there is material and negative where there is no material in the design domain. Thus the level-set is the set of points for which this functional is zero and represents the boundary of the structure. The implicit function can be modified in order to find an optimal structure. Xia et al. [183] used a level-set approach to maximise the fundamental frequency of a continuum structure. In 2010, Challis [35] produced an educational article that was a short MATLAB code for topology optimization using a level-set approach. There are many issues still to be answered regarding the use of level-sets for topology optimization such as schemes for hole insertion [32] and the optimal methods of structural analysis using level-sets [179]. This approach is not considered in this thesis but is covered in detail in the thesis of Dunning [46]. Instead we focus on the analysis of the two leading methods for topology optimization, namely the element based approaches, SIMP and ESO.

2.2   Truss topology optimization

A truss structure is formed from a number of straight bars that are joined only at their ends. In order to optimize a truss, a *ground structure* of all allowable bars is described. The goal of truss optimization is to determine which of these bars should be included in the final design and the optimal thickness of each bar. A typical example is shown in Figure 2-1.

Optimality criteria (OC) has been a technique widely applied to truss optimization problems. In the OC approach, the KKT conditions (see Definition 4.5) are written down for the given problem and an iterative scheme adopted to try and converge to meet these conditions. Khot et al. 1976 [87] used OC to design a reinforced truss structure for minimum weight subject to stability constraints. The same technique was

(a) A typical truss problem. Bars are allowed only between nodes.

(b) Example solution of an optimized truss.
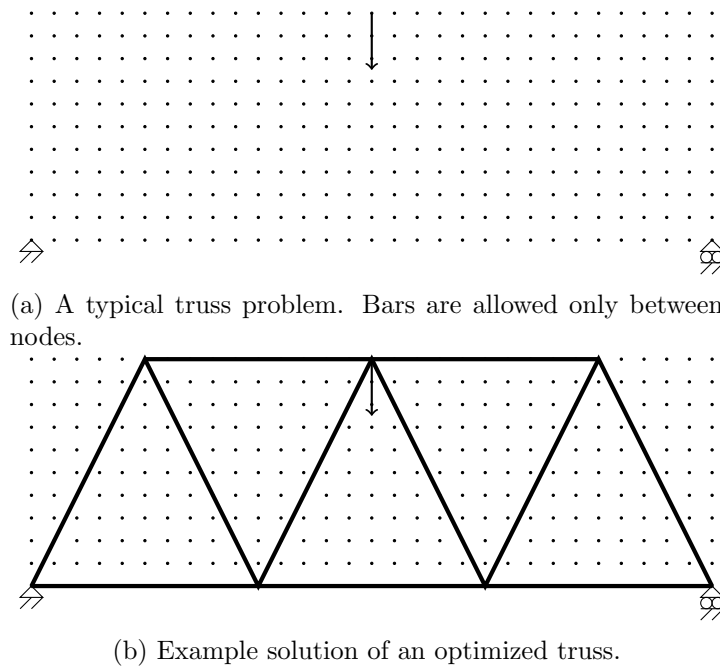
Figure 2-1: An example of a possible truss optimization problem and its solution.

applied to the design of structures from material that exhibit nonlinear behaviour [86].

Ringertz 1985 [133] worked on topology optimization of trusses for minimisation of weight subject to stress and displacement constraints. Firstly an optimal topology was found via linear programming then the sizes of the bars were optimized via nonlinear programming.

Branch-and-bound methods have been used in truss optimization to find global minimisers of weight subject to stress and displacement constraints [134, 142]. Ringertz 1988 [135] compared methods for solving discrete truss topology optimization problems. He compared branch-and-bound methods, dual methods and a continuous problem with rounding and found that the problem size was highly limiting for the discrete methods. Achtziger and Stolpe 2007 [6] used a branch-and-bound method to find the globally optimal solution to truss topology optimization problems.

Achtziger and Stolpe 2008 [7] give the theoretical basis for the relaxed subproblem in a branch-and-bound approach. They followed this with a paper [8] discussing the implementation and numerical results of truss topology optimization. Yonekura and Kanno 2010 [185] used a branch-and-bound algorithm to find the global minimiser of a truss topology problem that was written in a semidefinite formulation.

Buckling has also been considered in truss optimization. There are two types of buckling which can be considered in truss optimization: local and global buckling.

Local buckling is where each member bar is considered individually and there is a critical buckling load for every bar in the system. Global buckling is where the system is considered as a whole and there are more than one possible deformative modes for the system (see Chapter 6). Local buckling poses significantly fewer computational difficulties than global buckling.

Many other formulations of truss topology optimization problems have been posed. For instance, Beckers and Fleury 1997 [17] used a primal-dual approach to minimisation of compliance subject to volume for truss topology problems. Achtziger 2007 [3] considered truss topology optimization where both the location of connections and the cross sectional area of the bars were design variables. Kanno and Guo 2010 studied truss topology optimization with stress constraints in a mixed integer programming manner [82]. The largest example they computed (and found the global solution) has 29 design variables.

This thesis is concerned with topology optimization of continuum structures which poses more computational challenges than truss topology optimization.

2.3   Optimization of composites

Optimization of composite materials is an active research area with many open questions. A composite material consists of multiple layers (or *plys*) of anisotropic material, and the goal of the optimization of the composite is to find the optimal orientation of the alignment of each ply of anisotropic material. These optimization problems typically have reasonably small dimension (fewer than 20 variables) but are subject to many manufacturing constraints. This leads to feasible regions which are nonconvex and possibly disconnected.

For example, Starnes and Haftka 1979 [157] looked at composite panels and optimized them for maximum buckling load subject to strength and displacement constraints. Tenek and Hagiwara 1994 [173] used homogenisation techniques (see section 2.5) to maximise the fundamental eigenfrequency of both isotropic and composite plates. To perform the optimization they used SLP methods.

Setoodeh et al. [149] and Lindgaard and Lund 2010 [101, 102] optimize the layout of fibre angles in a composite material in order to maximise the buckling load of the material. Karakaya and Soykasap 2011 [83] used a genetic algorithm and simulated annealing to optimize composite plates.

This thesis shall not look at optimization of composite panels, but instead will be concerned with topology optimization problems where the material is isotropic.

2.4   Topological derivatives

The *topological derivative* is a measure of how a functional changes when an infinitesimally small spherical hole is introduced into the structure. In 1999, Sokolowski and Żochowski [154] worked on the topological derivative in shape optimization. They formally defined the topological derivative T at a point ξ for an arbitrary functional $J \in \Omega$ as

$$T(\xi) := \lim_{h \to 0} \frac{J(\Omega \backslash B(\xi, h)) - J(\Omega)}{|B(\xi, h)|}$$

where $B(\xi, h)$ is the ball of radius h centred at ξ. In 2001, Garreau et al. [55] gave the specific formulations for the topological derivative of planar linear elasticity equations.

Suresh 2010 [164] wrote an educational article on Pareto-optimal tracing in topology optimization. They produced an educational MATLAB code that made use of topological-sensitivity (or topological derivative). Amstutz 2011 [13] used the topological derivative approach to write a topology optimization problem with cone constraints. They presented results for minimisation of weight subject to compliance and harmonic eigenvalue constraints.

This thesis shall not consider using the topological derivative. To do so would require the use of a structural representation other than an element based approach, which is how we have chosen to implement our methods.

2.5   Homogenisation

Bendsøe and Kikuchi 1988 [20] were the first to apply a homogenisation method to structural optimization. Here a small cell structure was designed using a fixed grid finite element representation and then homogenisation was used to calculate the effective properties of a material composed of the individual cells. Suzuki and Kikuchi 1991 [165] applied the homogenisation method of Bendsøe and Kikuchi [20] to extra problems in order to validate it.

Tenek and Hagiwara 1994 [173] used homogenisation techniques to maximise the fundamental eigenfrequency of both isotropic and composite plates and used SLP to perform the optimization. In a famous industrial example of topology optimization, Larsen et al. 1997 [98] designed compliant mechanisms and the microstructure of a material with negative Poisson's ratio.

Maar and Schulz 2000 [104] applied multigrid methods within a homogenisation setting for structural optimization. More recently homogenisation approaches have fallen out of favour, giving way to the SIMP approach for topology optimization.

2.6 Solid Isotropic Material with Penalisation (SIMP)

Until 1989 only integer values were used as the design variables for structural optimization. In his paper of that year, Bendsøe proposed a method to vary the design variables continuously which resulted in a non-discrete solution [21]. In order to obtain a non-discrete solution that approximated a discrete solution the underlying mathematical model used to perform the analysis of the structure was changed to give less influence to intermediate values of the variables. This type of scheme was later named Solid Isotropic Material with Penalisation (SIMP) [140].

Buhl et al. 2000 [31] used the SIMP approach along with the Method of Moving Asymptotes (MMA) [168] to minimise various objective functions of geometrically non-linear structures subject to volume constraints. In 2001, Rietz showed how the penalty function in the SIMP method was sufficient to give discrete solutions under some conditions [132]. In 2001, Stolpe and Svanberg [160] discussed using a continuation method to incrementally increase the penalty parameter in the SIMP method. They concluded that this avoids many local minima which may be attained when using a constant value of the penalty parameter but at the expense of increased computational cost. They also found specific examples where the solution will contain intermediate densities regardless of the size of the penalty parameter.

In 2001, Sigmund published a freely available code for topology optimization written as a short MATLAB code [150]. The code was based on the SIMP formulation and used a Nested Analysis and Design (NAND) based approach to update the structure using an iterative method to converge to the given optimality criteria (OC) for the minimisation of compliance subject to a volume constraint problem.

Rozvany 2001 [138] presented a semi-historical article about the SIMP method and its advantages over other approaches for topology optimization. Bendsøe and Sigmund 2003 [19] produced the monograph on topology optimization in which the SIMP approach was the main technique considered.

Martinez 2005 [108] showed that in the SIMP approach, solutions to this problem exist under given assumptions about the penalisation function. An example of an industrial application of the SIMP method was given in Sardan et al. 2008 [146] where they presented optimization of Micro Electro Mechanical Systems (MEMS) grippers for application in the manufacturing of carbon nanotubes.

Niu et al. 2011 [119] looked at applying both external forces and non-zero displacements to the structure. Here the stiffness of the structure is measured by a function that differs from compliance so extra techniques are required to deal with this situation. This is a prime example of the power and flexibility of the SIMP method. Formulated

in this manner, the topology optimization problem can be tackled using generic mathematical programming software that has the ability to address such constraints.

2.7 Simultaneous Analysis and Design (SAND)

Haftka 1985 [66] wrote a paper called Simultaneous Analysis and Design (SAND). In it he describes SAND as the formulation of the optimization problem with the solution of the state equations also as optimization variables. This increases the dimension of the optimization problem which has to be solved and therefore potentially the computational difficulty of the problem. It also increases the potential search space and therefore may, on occasion, find a solution with an improved objective function or find a solution in a fewer number of steps.

Orozco and Ghattas 1992 [123] wrote about trying to use sparsity to help in a SAND approach to structural optimization. They found that the SAND approach bettered the NAND approach whenever the sparsity of the Jacobian was utilised in the SAND approach. Kirsch and Rozvany 1994 [89] discuss the SAND method and its advantages and disadvantages.

Sankaranarayanan et al. 1994 [145] used a SAND approach to truss topology optimization using an Augmented Lagrangian method. They had difficulties with efficiency though found that in some cases very good solutions were attained.

In 1997, Orozco [122] used a SAND approach to solve structural optimization problems with non-linear material. Hoppe and Petrova 2004 [71] used a Primal-dual Newton interior point method to solve shape and topology optimization problems in a SAND based approach.

More recently, Bruggi and Venini 2008 [28] considered stress-constrained topology optimization with stresses in the optimization formulation. Canelas et al. 2008 [34] used the SAND approach and boundary element methods for shape optimization. The reasons why the SAND approach is not widely used have not been documented, which is noteworthy as it has the potential to produce improved local optima. The SAND approach will be investigated in Chapter 5.

2.8 Evolutionary Structural Optimization (ESO)

Evolutionary Structural Optimization (ESO) is a different approach to finding solutions to structural optimization problems. It was originally developed by Xie and Stephen 1993 [184]. The basic premise of ESO is to systematically remove material that appears to be the least important to the structure.

Querin et al. 2000 [128, 129] introduced an additive ESO algorithm which was named Bi-directional Evolutionary Structural Optimization (BESO). This follows the same basic premise as ESO but can reintroduce material into the structure.

Zhou and Rozvany 2001 [187] proposed their "tie-beam" and showed how ESO, when applied to this problem, produces a highly non-optimal solution. Huang and Xie 2007 [73] developed the filter that is used in BESO and also introduced the idea of using historical based sensitivity to improve the convergence, albeit without mathematical justification. In this, to improve the nonmonotonic behaviour of the objective function, they define the sensitivity of an element as a weighted average of the sensitivity of the element over previous iterations.

Burry et al. 2005 [33] wrote about architectural examples in which ESO and BESO have been applied. The centrepiece of this work was to show that the design of a façade of the Sagrada Família in Barcelona is structurally optimum.

Huang and Xie 2008 [72] published an article on how all boundary conditions need to remain in order for ESO/BESO methods not to find highly nonoptimal solutions. Again, no mathematical justification for this was given. Rozvany 2008 [139] wrote a highly critical article in which the lack of mathematical theory for ESO led him to favour such methods as SIMP for topology optimization.

Zuo et al. 2009 [76] combined BESO with a genetic algorithm but did not say how many individuals they kept in their population at each step. They did find that only a small number of iterations was required to find an optimum that was better than the local optimum found for the same problems by the SIMP approach. Huang and Xie 2010 [74] talk about recent advances to ESO/BESO and show numerical examples of where it is effective. In that year they also produced a book on evolutionary structural optimization [75].

There has been very little written about the convergence of ESO. Tanskanen 2002 [172] published a paper comparing ESO with the simplex method. He found that the step taken by ESO is equivalent to taking an optimal simplex step. However he did not address the nonmonotonic behaviour of the convergence of ESO. Chapter 7 of this thesis will examine the question of why ESO has nonmonotonic convergence.

2.9 Buckling optimization

For a given load, a structure may have many possible deformation shapes. When this occurs and the structure deforms into a different one of these shapes from its current configuration, the structure is said to buckle. A formal definition of this is given in Chapter 6.

Giles and Thompson 1973 [77] considered the implications of structural optimization on the nonlinear behaviour of structures. They noted that a "process of optimization leads almost inevitably to designs which exhibit the notorious failure characteristics often associated with the buckling of thin elastic shells". Thus removing material deemed unnecessary based on a given get of loading and boundary conditions may make the structure subject to failure or collapse under differing loads.

This has led to engineers wanting to impose extra constraints on the optimization problem in order to find optimal structures which are not unstable. This constraint is an eigenvalue constraint which is similar in mathematical structure to a constraint on the harmonic (or resonant) modes of the structure.

Haftka and Gürdal 1991 [67] published their book on elements of structural optimization. They prescribe the derivative of an eigenvalue constraint for the case in which the eigenvalue is simple. However they completely neglect to give an expression for the derivative of the stress stiffness matrix, which is presented in Section 6.4 of this thesis.

In truss optimization, Gu et al. 2000 [63] considered optimization of trusses with buckling objectives subject to weight constraints and vice versa. Pedersen and Nielsen 2003 [126] looked at truss optimization with stress and local buckling constraints and performed the optimization with Sequential Linear Programming (SLP) methods. Guo et al. 2005 [64] considered truss topology optimization to minimise the weight of a structure whilst maintaining stress and local buckling constraints.

Neves et al. 1995 [117] maximise the minimum buckling load of a continuum structure subject to a volume constraint in an optimal reinforcement sense. They do find spurious buckling modes in which the buckling of the structure is confined to the regions which are supposed to represent voids (see section 6.2). Their solution to eradicate such modes was to set the stress contributions of low density elements in the stress stiffness matrix to zero.

Pedersen 2000 [125] considered using the SIMP approach to maximise the minimum harmonic eigenvalue of a structure. He applied this method to the design of MEMS. Spurious localised modes were observed and were eradicated using a similar technique to Neves et al. 1995 [117]. Ben-Tal et al. 2000 [18] considered truss topology design with a global buckling constraint and solved this problem using SDP. In the same year Cheng at al. 2000 [36] performed maximisation of the critical load of a structure subject to a volume constraint using OC.

Kočvara 2002 also considered truss topology design with a global buckling constraint [90]. Within this paper there is a clear description of the difference between the global buckling of the structure and local Euler buckling of each bar. There is also an excellent

description of how the semidefinite approach is equivalent to bounding the smallest positive eigenvalue. They used an interior point technique to solve the problem when written as a semidefinite programming problem.

Also Neves et al. 2002 [118] considered the problem of minimising a linear combination of the homogenised elastic properties of the structure subject to volume and bucking constraints applied to periodic microstructures. They do not use the SIMP method to penalise intermediate densities but instead add a penalty term to the objective function considered. They also make some assumptions that all the eigenvalues of the buckling problem are positive which significantly simplifies the calculations. They note that "the appearance of low-density regions may result in non-physical localized modes in the low-density regions, which are an artefact of the inclusion of these low-density regions that represent void material in the analysis". Their strategy of eradicating these spurious buckling modes by setting the stress in low density regions to an insignificant value (10^{-15}) is an approximation of setting the stress to zero, but is necessitated by their assumption that all the buckling modes are positive.

A SIMP approach to buckling optimization on a continuum structure was used by Rahmatalla and Swan [130] in 2003. They assumed that the eigenvalues were all simple, or that symmetry could be removed from the problem so that they did not occur. Kočvara and Stingl 2004 [92] utilise an Augmented Lagrangian formulation of the SDP formulation and solve this within the code PENNON. They solve some problems of buckling and vibration constrained optimization but only in a Variable Thickness Sheet (VTS) setting.

Maeda et al. 2006 [105] developed a method for maximising the harmonic frequency of a continuum structure. Jensen and Pedersen 2006 [78] optimized topologies to get the largest separation of harmonic eigenvalues around a specific frequency.

Achtziger and Kočvara 2007 [4] consider the maximisation of the fundamental (harmonic) eigenvalue in truss topology optimization. They do not include penalisation in their approach and so have a convex problem which they solve with SDP methods. In the same year they also considered using SDP methods to solve truss topology optimization problems involving buckling [5].

Bruyneel et al. 2008 [29] discussed convergence properties of buckling optimization. They talk about the need for considering multiple eigenvalues as (in continuous optimization) mode switching can occur and so a buckling constraint can easily be violated by an eigenvalue that was not being considered.

Zhan et al. 2009 [186] considered a SIMP approach to maximise the minimal harmonic frequency of a continuum structure. They used an SLP method, similar to Stingl et al. 2009 [159] who used an FMO approach along with SDP to optimize

structures with constraints on the fundamental eigenfrequency.

Bogani et al. 2009 [23] have applied an adapted version of their semidefinite codes to find VTS solutions to buckling problems. This made use of a reformulation of a semidefinite constraint using the indefinite Cholesky factorisation of the matrix, and solving a resulting nonlinear programming problem with an adapted version of MMA. With these techniques they were able to solve a non-discrete problem with 5000 variables in about 35 minutes on a standard PC. The approach was based on an observation by Fletcher in 1985 [53] who noted a formulation of a semidefinite matrix constraint that consists of bounding the values of the inertia of the matrix involved and can be computed by looking at the values of the diagonal factors in an LDL^T factorisation.

Du and Olhoff 2005 [44] presented methods for dealing with multiple eigenfrequencies. Lee 2007 [99] also introduced a method for calculating a derivative of a nonsimple eigenvalue. The derivative of a nonsimple eigenvalue is not well defined, as is shown in Section 6.1. Therefore, if we try and apply a derivative based optimization method that is not designed specifically to deal with this eventuality, we will be providing the optimization method with the wrong values of the derivative. Thus the method may fail to converge or indeed it may return a highly nonoptimal solution.

Other approaches to buckling optimization have included Sadiku 2008 [141] who used variational principals to compute the optimal cross-sectional area for columns of given height and volume in order to maximise the buckling load. Mijailović 2010 [113] minimised the weight of a braced column subject to both global and local buckling constraints as well as deformation constraints. Nagy et al 2011 [115] used a NURBS representation of a structure and optimized them to maximise the fundamental frequency of an arch. Buckling constraints have been included in composite optimization, for example in 1997, Mateus et al. [109] investigated the buckling sensitivities of composite structures.

The inclusion of buckling into a structural optimization problem was very well summed up by Bruyneel et al. 2008 [29], "it must be noted that buckling optimisation is a very difficult problem".

Chapter 6 of this thesis will consider optimization with global buckling constraints.

2.10 Chequerboarding

A problem of minimization of compliance subject to volume is known to be ill-posed (see for example Ambrosio and Buttazzo 1993 [9] and Kohn and Strang 1986 [93, 94, 95]). That is, improved structures can be found by taking increasingly smaller microstructure. Therefore the problem as stated in general has no solution. In a
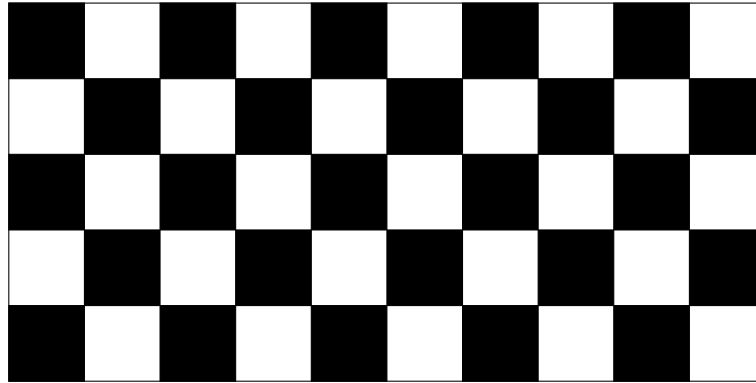
Figure 2-2: Chequerboard pattern of alternating solid and void regions
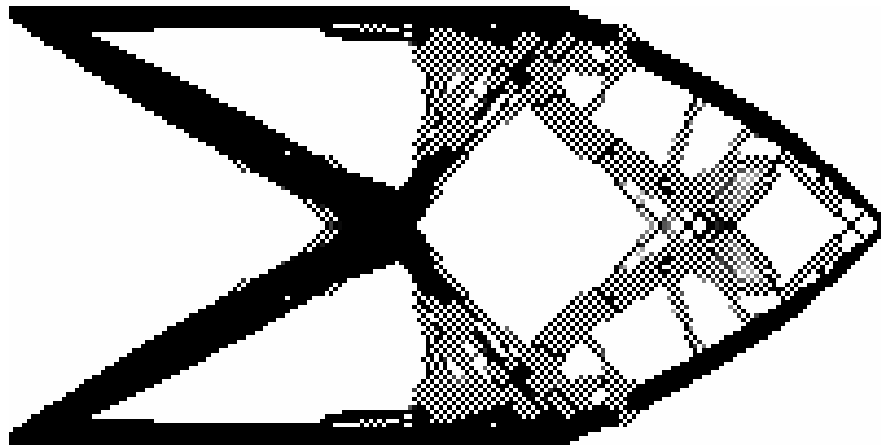
Figure 2-3: Chequerboard pattern appearing in the solution to a cantilevered beam problem.

numerical calculation the solutions of the problem would therefore be dependent on the size of the mesh that is employed. Microstructure is found commonly in nature: materials such as bone and wood have have multiple length scales associated with them, with different organisations of material at the various scales [96].

In an element-based topology optimization approach there may exist solutions that are not desired by engineers. These solutions typically exhibit *chequerboard patterns* as shown in Figure 2-2. In an actual example of minimising the compliance of a cantilevered beam this may manifest itself as in Figure 2-3.

Diaz and Sigmund 1995 [42] discuss how chequerboard patterns have artificially high stiffness for their relative density. These patterns were also observed by, amongst others, Jog and Haber 1996 [79] in topology optimization problems. Sigmund and Petersson 1998 [151] surveyed the methods for dealing with chequerboard patterns appearing in

topology optimization. The two most popular methods are *filtering techniques* and imposing a constraint on the perimeter of the structure.

Rahmatalla and Swan 2004 [131] implemented topology optimization with higher order elements and showed that this eradicated chequerboard patterns. However, this did not lead to mesh independent designs and they still had to include a perimeter constraint. If the underlying mesh has no corner contacts (such as a hexagonal mesh) then these issues do not arise. This has been observed by Talischi et al. 2008 [171, 170]. However, automatic mesh generation techniques in general do not exclude corner contacts between elements so it is necessary to employ techniques to eradicate chequerboard patterns from any mesh.

To overcome the illposedness of the problem a lower length scale is imposed on the problem, but care has to be taken so that the optimization strategy is not too dependent on the regularisation strategy. This will be considered in detail in Section 5.5.

2.11 Symmetry properties of optimal structures

It might be expected that the solution to a topology optimization problem with symmetric design domain, boundary conditions and loading would be symmetric. There has recently been a lot of interest in this problem. Stolpe 2010 showed that the optimal solutions to topology optimization problems in general are not unique and that discrete problems possibly have inactive volume or compliance constraints [161]. He showed how optimal solutions to the considered problems in general are not symmetric even if the design domain, the external loads and the boundary conditions are symmetric around an axis.

This article prompted a series of responses, notably Rozvany 2011 [137, 136] and Watada et al. 2011 [178] looking at the nonuniqueness and nonsymmetry of solutions to symmetric minimisation of compliance problems. Cheng and Liu 2011 [37] discussed the symmetry of solutions of frame topology optimization with harmonic eigenvalue constraints and found that optimal solutions were nonsymmetric. This has implications for the buckling problem considered in Chapter 6 as we cannot either remove symmetries from the design domain nor expect symmetric solutions.

2.12 Linear algebra

Linear algebra always forms a large part of optimization. For example, even a simple Newton method requires the solution of a linear system involving the Jacobian ma-

trix. In structural optimization the solves involving the elasticity stiffness matrix are typically one of the most computationally intensive parts of the algorithms. If fact, Borrvall and Petersson [24] reported that up to 97% of the computational time is spent on the linear solve.

There are two broad categories of linear solver: *iterative* and *direct* methods. Iterative methods begin with an initial guess of the solution and apply a sequence of update steps to hopefully converge to the solution of the original equation. A convergence criterion is specified and the method continues until this is satisfied up to a certain tolerance. Examples of such methods include Jacobi iteration, Krylov subspace methods such as the preconditioned conjugate gradient (PCG) method and multigrid methods.

Direct solvers on the other hand decompose a matrix into a form which is then easy to invert using forward and backward substitution. These factors are computed in a finite number of arithmetic computations. For any method of solving a linear system to be effective, the *sparsity* of the matrix must be utilised. This is generally very easy to achieve with an iterative method such as a Krylov subspace technique as these rely on matrix-vector multiplication to find a solution. In a direct method the use of sparsity is much more complex [45].

When performing a Cholesky decomposition of a matrix A:

$$A = LDL^T \tag{2.1}$$

where L is a lower triangular matrix and D is a diagonal matrix, the efficiency of the process will depend greatly on the degree of sparsity of the matrix L. Pivot ordering strategies are used in order to improve the degree of *fill-in* that occurs.

Typically the convergence of an iterative method will depend on the condition number of the matrix in question (see Section 3.3). In contrast, the efficiency of a direct method is generally independent of the condition number of the matrix. To try and overcome this deficiency of iterative methods, *preconditioning* is applied to the matrix in order to try and give the resulting matrix a significantly lower condition number. Many of these techniques have been applied to topology optimization.

There are many issues around using multigrid methods to solve the linear-elasticity equations that occur in structural optimization. For instance the domain of the problem may be highly complex and the material in each element may vary in a SIMP approach. Stevenson 1993 [158] looks at multigrid methods for solving equations on domains with re-entrant corners and discusses the nontrivial issues of convergence. Karer and Kraus 2010 used algebraic multigrid (AMG) for solving finite element elasticity equations with non-constant Young's modulus [85].

Dreyer et al. 2000 [43] used multigrid and SQP for turbine blade profile optimization as well as simple cantilevered beam test topology optimization problems. For other examples of multigrid use, see Griebel et al. 2003 [62] or Buckeridge 2010 [30].

Borrvall and Petersson 2001 [24] considered 3D topology optimization on a distributed machine using PCG and domain decomposition to solve the elasticity equations. They used a simple diagonal preconditioner. They solved problems with a maximum of $144,000$ elements. Wang et al. 2007 [177] used a NAND approach to large-scale topology optimization. They used a preconditioned Krylov subspace method with subspace recycling in order to reduce the computational cost of each linear solve. Amir et al. 2009 [10] looked at a NAND approach to topology optimization. They discussed the need to accurately solve the state equations and found that an approximate solve is acceptable when the error is taken into account in the sensitivity analysis. This resulted in a saving of computation time.

Amir and Sigmund 2011 [11] discussed the latest challenges in reducing the computational complexity of topology optimization. They discussed the need for better preconditioners and appropriate stopping criteria for iterative solution of linear systems. Amir et al. 2010 [12] looked at efficient use of iterative solvers for a NAND approach to topology optimization. They use a preconditioned conjugate gradient (PCG) method for solving the linear system but precondition using an incomplete Cholesky factorisation.

El maliki et al. 2010 [48] compared general iterative solvers for 3D linear elasticity problems. They found that for linear elements, a direct solver (MUMPS) is generally more efficient than an iterative scheme provided that memory does not become an issue. This is the same result as Edwards 2008 [47] found when performing a comparison of solvers for the systems in topology optimization.

Venkataraman and Haftka 2004 [176] considered how Moore's Law has influenced structural optimization. There is always the possibility of using more computing power to solve a problem, but it is important to know how best to solve the problem given the available resources. In this thesis we consider computing on a standalone workstation, that is a machine with shared memory such as a desktop PC or a laptop computer.

Unless otherwise stated, the linear solver that will be used throughout this thesis will be HSL_MA87 [69, 70], a DAG (Directed Acyclic Graph) based direct solver from the HSL [1] mathematical software library. This is the successor to HSL_MA57, a multifrontal solver from HSL. HSL_MA87 is designed to make use of multiple processing cores accessing shared memory, and so attains a good degree of parallelism.

In the work carried out for this thesis, it has been found that a direct solver is still very efficient at solving linear algebra problems resulting from the linear elasticity

equations underlying the considered topology optimization problems when the matrices in question are up to size $\mathcal{O}(10^6)$. Beyond this problem size, memory issues come into play and an *in-core* direct factorisation will fail on a machine without more than than 4GB RAM. Indeed, problems with matrices of size $\mathcal{O}(10^7)$ have been solved efficiently on a server with larger amounts of shared memory.

2.13   Summary

There are a number of key issues in the field of structural optimization that this literature review has highlighted. Firstly, the issue of why a SAND approach to structural optimization has fallen out of favour compared to a NAND approach has not been thoroughly investigated. This thesis will consider this question in Chapter 5.

In Chapter 6 this thesis shall study the introduction of a buckling constraint into the optimization problem. The issues surrounding the use of existing methods for this problem will be highlighted and ultimately this will lead to the development of a new algorithm to give solutions to this problem.

The lack of mathematical justification for the ESO method for structural optimization will be addressed in Chapter 7. We shall try and provide some more theoretical basis for the optimization path which ESO takes and hope to explain the non-monotonic convergence behaviour which is typically exhibited by ESO.

Linear elasticity and finite elements

This chapter contains the derivation and analysis of the state equations that are used to compute the response of a structure to an applied load. Starting with Newton's laws of motion, in Section 3.1 the Lamé equation is derived which is the underlying PDE to be solved. The process of discretising this PDE in a finite element context is presented for linear elasticity in Section 3.2, i.e. the structure undergoes small displacements and the material obeys a linear stress–strain relationship. In Section 3.3, the conditioning of the finite element stiffness matrices are considered. The stress stiffness matrix is derived in Section 3.4, which is used to compute the linear buckling load of a structure. Section 3.5 describes the linear algebra technique employed to find the buckling load of a structure. Finally, corner singularities inherent in the underlying equations are discussed in Section 3.6 by first considering Laplace's equation and then looking at the elasticity case.

3.1 Linear elasticity

In this section, the Lamé equation is derived from Newton's laws of motion.

Definition 3.1. *A surface traction* $\mathbf{t}(\mathbf{e}_j)$ *is defined as follows*

$$\mathbf{t}(\mathbf{e}_j) := \sigma_{ij}\mathbf{e}_i$$

where σ_{ij}, $i, j = 1, 2, 3$ *are stresses.*

As a preliminary, consider a tetrahedron (Figure 3-1) whose *skewed* face has external normal $\mathbf{n}$.

Let $\mathrm{d}S_1, \mathrm{d}S_2$ and $\mathrm{d}S_3$ be surface elements perpendicular to x_1, x_2 and x_3 respectively. $\mathrm{d}S_n$ is the surface element perpendicular to the skewed face. $\mathbf{e}_1$ represents the
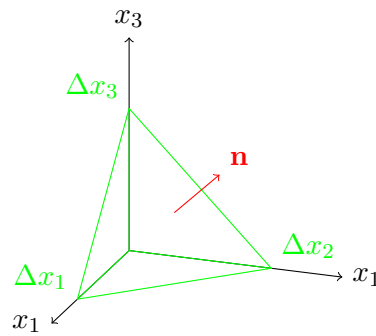
Figure 3-1: Tetrahedron relating tractions and stresses

vector $[1, 0, 0]^T$ etc.

The forces on the faces perpendicular to the axes are given by

$$f_i = \mathbf{t}(-\mathbf{e}_i)\mathrm{d}S_i \qquad i = 1, 2, 3$$

and for the skewed face are given by

$$f_n = \mathbf{t}(\mathbf{e}_n)\mathrm{d}S_n$$

Newton's second law gives

$$\mathbf{t}(-\mathbf{e}_1)\mathrm{d}S_1 + \mathbf{t}(-\mathbf{e}_2)\mathrm{d}S_2 + \mathbf{t}(-\mathbf{e}_3)\mathrm{d}S_3 + \mathbf{t}(\mathbf{e}_n)\mathrm{d}S_n = m\ddot{\mathbf{x}} \qquad (3.1)$$

where m is a mass and $\ddot{\mathbf{x}}$ an acceleration. Consider now the following integral:

$$\int_V \frac{\partial}{\partial x_i} 1 \ \mathrm{d}x_1 \ \mathrm{d}x_2 \ \mathrm{d}x_3 = 0 \qquad \text{as the integrand } = 0$$

but applying the divergence theorem to the left hand side yields the following

$$\int_V \frac{\partial}{\partial x_i} 1 \ \mathrm{d}x_1 \ \mathrm{d}x_2 \ \mathrm{d}x_3 = \int_{\partial V} N_i \ \mathrm{d}S \qquad \text{where } N_i \text{ is external normal to } \partial V$$
$$= -\mathrm{d}S_i + n_i \mathrm{d}S_n$$

Hence

$$\mathrm{d}S_i = n_i \mathrm{d}S_n \qquad i = 1, 2, 3$$

and substituting this into (3.1) gives

$$(-\mathbf{t}(\mathbf{e}_j)n_j + \mathbf{t}(\mathbf{n}))\mathrm{d}S_n = m\ddot{\mathbf{x}} \qquad (3.2)$$
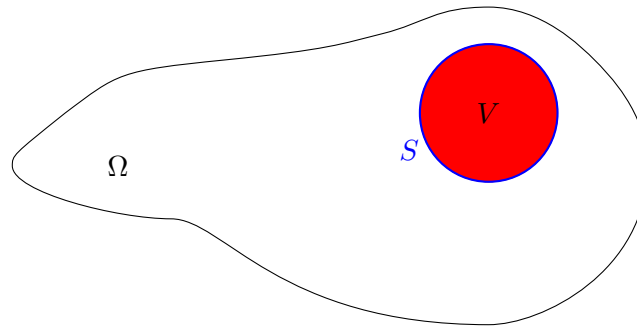
Figure 3-2: A continuum body Ω containing an arbitrary volume V

noting the use of Einstein notation where the sum is denoted by the repeated index.

Now let $\Delta x_1, \Delta x_2$ and $\Delta x_3 \to 0$ then $m \to 0$ cubically but $\mathrm{d}S_n \to 0$ quadratically so the term in brackets in (3.2) is equal to zero. So

$$\mathbf{t}(\mathbf{n}) = \mathbf{t}(\mathbf{e}_j)n_j = \sigma_{ij}\mathbf{e}_i n_j$$

and as a result, for each component

$$(\mathbf{t}(\mathbf{n}))_i = \sigma_{ij}n_j$$

Now consider a continuum body Ω containing an arbitrary volume $V \in \Omega$ with boundary $S = \partial V$. Let ρ represent the density of mass at a point, $\mathbf{f}$ body forces, $\mathbf{t}$ surface tractions applied to V, $\mathbf{u}$ displacements of the body and $\ddot{\mathbf{u}}$ the accelerations of the body.

Newton's second law then gives the following equality:

$$\int_V \rho\ddot{\mathbf{u}} \ \mathrm{d}V = \int_V \rho\mathbf{f} \ \mathrm{d}V + \int_S \mathbf{t} \ \mathrm{d}S$$

Splitting (3.1) into each component gives:

$$\int_V (\rho\ddot{u}_i - \rho f) \ \mathrm{d}V = \int_S t_i \ \mathrm{d}S \qquad i = 1, 2, 3$$

$$= \int_S \sigma_{ij}n_j \ \mathrm{d}S$$

$$= \int_V \sigma_{ij,j} \ \mathrm{d}V \qquad \text{by the divergence theorem}$$

Hence

$$\int_V (\rho\ddot{u}_i - \sigma_{ij,j} - \rho f_i) \ \mathrm{d}V = 0 \qquad i = 1, 2, 3$$

Hence as V was arbitrary this leads to the *equation of motion*

$$\rho \ddot{u}_i - \sigma_{ij,j} = \rho f_i \qquad i = 1, 2, 3$$

and in the special case when the body is equilibrium, i.e. $\ddot{u} = 0$, this gives the *equation of equilibrium*

$$\sigma_{ij,j} = -\rho f_i \qquad i = 1, 2, 3$$

which can be written in vector notation to give the *Lamé* equation

$$-\nabla.(\sigma) = \rho f.$$

3.2 The finite-element discretisation of the linear elasticity equations

In this section the equilibrium equations governing linear elasticity are derived. Firstly begin by defining concepts needed for the presentation of the finite-element method.

Definition 3.2.

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

where the norm is given by

$$||f||_{L^p(\Omega)} := \left(\int_\Omega |f(x)|^p d\mathbf{x} \right)^{\frac{1}{p}}$$

Definition 3.3. *A* multi-index *is an ordered list of n non-negative integers $\alpha = \alpha_1, \ldots, \alpha_n$. The order of α is $|\alpha| := \alpha_1 + \ldots + \alpha_n$.*

Given α there exist associated polynomial functions

$$x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$$

and partial differential operators

$$(D^\alpha v) = (\frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \cdots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}}) v$$

Definition 3.4.

$$H^k(\Omega) := \{ f \in L^2(\Omega) \quad s.t. \quad ||f||_{H^k(\Omega)} < \infty \}$$
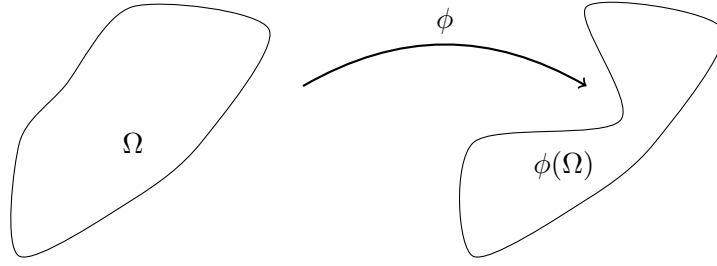
Figure 3-3: Elastic body before and after deformation

where the norm is defined by

$$||f||_{H^k(\Omega)} = \{ \sum_{0 \leq |\alpha| \leq k} ||D^\alpha f||^2_{L^2(\Omega)} \}^{\frac{1}{2}}$$

and denote

$$H_0^k(\Omega) := \{ f \in H^k(\Omega) \quad s.t. \quad \mathbf{f} = \mathbf{0} \ on \ \partial\Omega_D \}.$$

Let $\Omega \subset \mathbb{R}^3$ be an elastic body in its unstressed state. Under stress it undergoes a deformation

$$\phi : \bar{\Omega} \to \mathbb{R}^3.$$

Write $\phi = I + \mathbf{u}$ where I is the identity map and $\mathbf{u} : \bar{\Omega} \to \mathbb{R}^3$ is the displacement vector.

Definition 3.5. *The dot product of two tensors is defined by*

$$\mathcal{T} : \sigma = \sum_{i,j} \mathcal{T}_{ij} \sigma_{ij}$$

Definition 3.6. *Assuming small displacements, the strain tensor is defined by*

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

which can be equivalently written as

$$\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2}(u_{i,j} + u_{j,i})$$

Hooke's law for for the relationship between stress and strain is given by the following relation.

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl}$$

where σ_{ij} is the **stress tensor** and c_{ijkl} is referred to as the **stiffness tensor**.

Definition 3.7. *For an isotropic solid, Hooke's law defines the stress tensor as follows*

$$\sigma_{ij} = \lambda \delta_{ij} \varepsilon_{kk} + 2\mu \varepsilon_{ij}$$

where $\lambda, \mu \in \mathbb{R}$ *are known as the Lamé constants,* $\varepsilon_{kk} := \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}$ *and* δ_{ij} *is the Kronecker delta.*

This is equivalent to

$$\sigma_{ij} = \frac{E\nu}{1 - \nu^2} \delta_{ij} \varepsilon_{kk} + \frac{E}{1 + \nu} \varepsilon_{ij} \tag{3.3}$$

with E the Young's modulus of the material, and ν the Poisson's ratio of the material.

The constituent equation that the displacement $\mathbf{u}$ then satisfies is the Lamé equation

$$-\nabla \cdot (\sigma(\mathbf{u})) = f \qquad \text{on } \Omega$$

subject to

$$\mathbf{u} = \mathbf{g} \qquad \text{on } \partial\Omega_D$$

and

$$\sigma(\mathbf{u})\nu = \mathbf{t} \qquad \text{on } \partial\Omega_N \tag{3.4}$$

where $\mathbf{f}$ is the body force, $\mathbf{g}$ is the boundary displacement, $\mathbf{t}$ is the boundary traction and ν is the outward unit normal to Ω.

The derivation of the weak form of (3.2) is shown subsequently. Let $\mathbf{v} \in H_0^1(\Omega)$.

$$
\begin{aligned}
-\int_\Omega \nabla \sigma . \mathbf{v} &= -\sum_i \sum_j \int_\Omega \sigma_{,j} v_i \\
&= \sum_i \sum_j \left(\int_\Omega \sigma \frac{\partial v_i}{\partial x_j} - \int_{\partial\Omega} \sigma v_i \nu_j \right) \\
&= \int_\Omega (\sigma : \nabla \mathbf{v}) - \int_{\partial\Omega} \sigma \nu . \mathbf{v}
\end{aligned}
$$

Rearranging this gives

$$\int_\Omega (\sigma : \nabla \mathbf{v}) = \int_\Omega -\nabla \sigma . \mathbf{v} + \int_{\partial\Omega} \sigma \nu . v$$

and substituting the boundary condition (3.4) and the Lamé equation (3.2) gives

$$\int_\Omega (\sigma : \nabla \mathbf{v}) = (\mathbf{f}, \mathbf{v})_{L^2(\Omega)} + (\mathbf{t}, \mathbf{v})_{L^2(\partial\Omega_N)} \tag{3.5}$$

where

$$(\mathbf{f}, \mathbf{g})_{L^2(\Omega)} := \sum_i \int_\Omega f_i g_i$$

Note that

$$\varepsilon(\mathbf{u}) : \nabla \mathbf{v} = \frac{1}{2} \sum_{i,j} (u_{i,j} + u_{j,i}) v_{i,j} \tag{3.6}$$

Since the double sum on the RHS is over all i and j, the result is unchanged if i and j are interchanged in the summand, i.e.

$$\varepsilon(\mathbf{u}) : \nabla \mathbf{v} = \frac{1}{2} \sum_{i,j} (u_{j,i} + u_{i,j}) v_{j,i} \tag{3.7}$$

Summing both (3.6) and (3.7) gives the following

$$
\begin{aligned}
2\varepsilon(\mathbf{u}) : \nabla \mathbf{v} &= \frac{1}{2} \sum_{i,j} (u_{i,j} + u_{j,i}) v_{i,j} + \frac{1}{2} \sum_{i,j} (u_{j,i} + u_{i,j}) v_{j,i} \\
&= \frac{1}{2} \sum_{i,j} (u_{i,j} + u_{j,i}) (v_{i,j} + v_{j,i})
\end{aligned}
$$

hence

$$
\begin{aligned}
\varepsilon(\mathbf{u}) : \nabla \mathbf{v} &= \sum_{i,j} \frac{1}{2} (u_{i,j} + u_{j,i}) \frac{1}{2} (v_{i,j} + v_{j,i}) \\
&= \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})
\end{aligned}
$$

A direct calculation can show

$$(\delta_{ij} \varepsilon_{kk}(\mathbf{u})) : \nabla \mathbf{v} = \nabla.\mathbf{u} \nabla.\mathbf{v}$$

and hence using Definition 3.7, (3.5) can be written as

$$\int_\Omega (2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) + \lambda \nabla.\mathbf{u} \nabla.\mathbf{v}) = (\mathbf{f}, \mathbf{v})_{L^2(\Omega)} + (\mathbf{t}, \mathbf{v})_{L^2(\partial\Omega_N)} \tag{3.8}$$

for all $\mathbf{v} \in H_0^1(\Omega)$. Writing (3.8) in abstract form becomes

$$\mathbf{a}(\mathbf{u}, \mathbf{v}) = F(\mathbf{v}) \qquad \forall \mathbf{v} \in H_0^1(\Omega) \tag{3.9}$$

where $\mathbf{a}$ is a symmetric bilinear form.

If V_h is a finite dimensional subset of $H_0^1(\Omega)$, a basis $\{\phi_i : i = 1, \ldots, N\}$ for V_h can be chosen. Thus to solve this problem it is necessary to find $u_h \in V_h$ such that

$$\mathbf{a}(\mathbf{u_h}, \mathbf{v_h}) = F(\mathbf{v_h}) \qquad \forall \mathbf{v_h} \in V_h$$

Write $u_h = \sum_{j=1}^{N} U_j \phi_j$ for some unknown coefficients U_j. Since $\mathbf{a}$ is linear this is equivalent to finding U_j such that

$$\sum_{j=1}^{N} \mathbf{a}(\phi_j, \phi_i) U_j = F(\phi_i) \qquad \forall i = 1, \ldots, N$$

Then the matrix form of (3.9) becomes

$$K\mathbf{u} = \mathbf{f} \tag{3.10}$$

where

$$K_{ij} := \mathbf{a}(\phi_j, \phi_i) \qquad \forall i, j = 1, \ldots, N$$
$$\mathbf{f}_i := F(\phi_i) \qquad \forall i = 1, \ldots, N$$

The matrix K_{ij} is known as the *stiffness matrix* and the vector $\mathbf{f}$ the applied force.

3.2.1 Coercivity of the bilinear form in linear elasticity

Definition 3.8. *The H^1 seminorm of a function $f \in H^1(\Omega)$ is defined by*

$$|f|_{H^1(\Omega)} = \left(\int_\Omega |\nabla f|^2 \right)^{\frac{1}{2}}$$

Theorem 3.9 (The Poincaré-Friedrichs Inequality)**.** *If Ω is a bounded domain then there exists a constant $C > 0$ (which depends on Ω) such that*

$$||\mathbf{u}||_{H^1(\Omega)} \leq C|\mathbf{u}|_{H^1(\Omega)} \qquad \forall \mathbf{u} \in H_0^1(\Omega)$$

Proof. The proof of this is omitted but can be found in, for example, Brenner and Scott [26] section 5.3. $\square$

Lemma 3.10. *Given an operator $[\mathbf{v}\nabla]$ defined by*

$$[\mathbf{v}\nabla]\mathbf{u} := \begin{bmatrix} \sum_i v_i \frac{\partial u_1}{\partial v_i} \\ \sum_i v_i \frac{\partial u_2}{\partial v_i} \\ \vdots \\ \sum_i v_i \frac{\partial u_n}{\partial v_i} \end{bmatrix}$$

then the following equality holds

$$2\varepsilon(\boldsymbol{v}) : \varepsilon(\boldsymbol{v}) = \nabla.([\boldsymbol{v}\nabla]\boldsymbol{v} - (\nabla.\boldsymbol{v})\boldsymbol{v}) + \nabla\boldsymbol{v} : \nabla\boldsymbol{v} + (\nabla.\boldsymbol{v})^2$$

Proof. By writing down each term, it is possible to see that equality holds. The first term is

$$2\varepsilon(\mathbf{v}) : \varepsilon(\mathbf{v}) = \tfrac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}(v_{i,j} + v_{j,i})^2 \tag{3.11}$$

$$= 2v_{1,1}^2 + 2v_{2,2}^2 + 2v_{3,3}^2 +$$
$$v_{1,2}^2 + v_{1,3}^2 + v_{2,3}^2 + v_{2,1}^2 + v_{3,1}^2 + v_{3,2}^2 +$$
$$2v_{1,2}v_{2,1} + 2v_{1,3}v_{3,1} + 2v_{2,3}v_{3,2} \tag{3.12}$$

The final term on the right hand side expands to the following

$$(\nabla.\mathbf{v})^2 = (v_{1,1} + v_{2,2} + v_{3,3})^2$$
$$= v_{1,1}^2 + v_{2,2}^2 + v_{3,3}^2 + 2v_{1,1}v_{2,2} + 2v_{1,1}v_{3,3} + 2v_{2,2}v_{3,3} \tag{3.13}$$

Similarly the middle term on the right hand side expands as follows

$$\nabla\mathbf{v} : \nabla\mathbf{v} = v_{1,1}^2 + v_{1,2}^2 + v_{1,3}^2 + v_{2,1}^2 + v_{2,2}^2 + v_{2,3}^2 + v_{3,1}^2 + v_{3,2}^2 + v_{3,3}^2 \tag{3.14}$$

For the first term on the right hand side, begin by writing down the argument inside the brackets.

$$[\mathbf{v}\nabla]\mathbf{v} - (\nabla.\mathbf{v})\mathbf{v} = \begin{bmatrix} v_1 v_{1,1} + v_2 v_{1,2} + v_3 v_{1,3} - (v_{1,1} + v_{2,2} + v_{3,3})v_1 \\ v_1 v_{2,1} + v_2 v_{2,2} + v_3 v_{2,3} - (v_{1,1} + v_{2,2} + v_{3,3})v_2 \\ v_1 v_{3,1} + v_2 v_{3,2} + v_3 v_{3,3} - (v_{1,1} + v_{2,2} + v_{3,3})v_3 \end{bmatrix}$$

Hence

$$
\begin{aligned}
\nabla.([\mathbf{v}\nabla]\mathbf{v} - (\nabla.\mathbf{v})\mathbf{v}) = {} & v_1 v_{1,11} + v_{1,1} v_{1,1} + v_2 v_{1,21} + v_{2,1} v_{1,2} + v_3 v_{1,31} + v_{3,1} v_{1,3} + \\
& - (v_{1,11} + v_{2,21} + v_{3,31}) v_1 - (v_{1,1} + v_{2,2} + v_{3,3}) v_{1,1} + \\
& v_1 v_{2,12} + v_{1,2} v_{2,1} + v_2 v_{2,22} + v_{2,2} v_{2,2} + v_3 v_{2,32} + v_{3,2} v_{2,3} + \\
& - (v_{1,12} + v_{2,22} + v_{3,32}) v_2 - (v_{1,1} + v_{2,2} + v_{3,3}) v_{2,2} + \\
& v_1 v_{3,13} + v_{1,3} v_{3,1} + v_2 v_{3,23} + v_{2,3} v_{3,2} + v_3 v_{3,33} + v_{3,3} v_{3,3} + \\
& - (v_{1,13} + v_{2,23} + v_{3,33}) v_3 - (v_{1,1} + v_{2,2} + v_{3,3}) v_{3,3} \quad (3.15)
\end{aligned}
$$

Now notice that all the terms with two derivatives in them in (3.15) cancel and what remains is

$$
\begin{aligned}
\nabla.([\mathbf{v}\nabla]\mathbf{v} - (\nabla.\mathbf{v})\mathbf{v}) = {} & 2(v_{1,2} v_{2,1} + v_{2,3} v_{3,2} + v_{1,3} v_{3,1}) - \\
& 2(v_{1,1} v_{2,2} - v_{1,1} v_{3,3} - v_{2,2} v_{3,3}) \quad (3.16)
\end{aligned}
$$

Now simply equating the terms in (3.12), (3.13), (3.14) and (3.16) gives the result. $\quad\square$

Theorem 3.11. *When $\partial\Omega_N = \emptyset$, $\mu > 0$ and $\lambda > -\mu$ then the bilinear form in (3.9) is coercive.*

Proof. Let $\mathbf{v} \in H_0^1(\Omega)$. Then

$$
\begin{aligned}
\mathbf{a}(\mathbf{v}, \mathbf{v}) & = \int_\Omega \left(2\mu\varepsilon(\mathbf{v}) : \varepsilon(\mathbf{v}) + \lambda\nabla.\mathbf{v}\nabla.\mathbf{v}\right) \\
& = \int_\Omega \mu\left(\nabla.([\mathbf{v}\nabla]\mathbf{v} - (\nabla.\mathbf{v})\mathbf{v}) + \nabla\mathbf{v} : \nabla\mathbf{v} + (\nabla.\mathbf{v})^2\right) + \lambda(\nabla.\mathbf{v})^2 \\
& = \int_\Omega \mu\nabla\mathbf{v} : \nabla\mathbf{v} + \int_\Omega (\mu + \lambda)(\nabla.\mathbf{v})^2 + \mu\int_\Omega \nabla.([\mathbf{v}\nabla]\mathbf{v} - (\nabla.\mathbf{v})\mathbf{v}) \\
& = \int_\Omega \mu\nabla\mathbf{v} : \nabla\mathbf{v} + \int_\Omega (\mu + \lambda)(\nabla.\mathbf{v})^2 + \mu\int_{\partial\Omega} ([\mathbf{v}\nabla]\mathbf{v} - (\nabla.\mathbf{v})\mathbf{v}).\mathbf{n} \\
& = \int_\Omega \mu\nabla\mathbf{v} : \nabla\mathbf{v} + \int_\Omega (\mu + \lambda)(\nabla.\mathbf{v})^2 \qquad \text{as } \mathbf{v} = 0 \text{ on } \partial\Omega
\end{aligned}
$$

where here the divergence theorem has been used. Hence

$$
\begin{aligned}
\mathbf{a}(\mathbf{v}, \mathbf{v}) & \geq \mu\int_\Omega \nabla\mathbf{v} : \nabla\mathbf{v} \qquad \text{as } \mu + \lambda > 0 \\
& = \mu|\mathbf{v}|_{H^1(\Omega)}^2 \qquad \text{by definition of } H^1(\Omega) \text{ seminorm} \\
& \geq C\mu||\mathbf{v}||_{H^1(\Omega)}^2 \qquad \text{by the Poincaré-Friedrichs Inequality}
\end{aligned}
$$

and thus $\mathbf{a}$ is coercive. $\quad\square$

For the proof of the general case where surface tractions are present, see Brenner and Scott [26] section 11.2.

Theorem 3.12. *The matrix K in (3.10) is positive definite.*

Proof. Let $\mathbf{w} \neq 0$ be an eigenvector of the matrix K in (3.10) corresponding to eigenvalue λ normalised so that $||\mathbf{w}||_1 = 1$. Then by the coercivity of $\mathbf{a}$

$$0 < \mathbf{a}(\mathbf{w}, \mathbf{w})$$

and so

$$
\begin{aligned}
\mathbf{a}(\mathbf{w}, \mathbf{w}) &= \mathbf{a}(\sum_j w_j \phi_j, \sum_i w_i \phi_i) \\
&= \sum_j w_j \mathbf{a}(\phi_j, \sum_i w_i \phi_i) \\
&= \sum_j \sum_i w_j w_i \mathbf{a}(\phi_j, \phi_i) \\
&= \mathbf{w}^\mathbf{T} K \mathbf{w} \\
&= \lambda \mathbf{w}^T \mathbf{w} = \lambda ||\mathbf{w}||_1^2 = \lambda
\end{aligned}
$$

Thus the eigenvalue λ is bounded away from zero and thus the matrix K is positive definite. $\qquad\square$

3.3   Conditioning of the stiffness matrix

Let us now consider the condition number of the stiffness matrix emanating from the SIMP method (see Section 5.2) for topology optimization as examined by Wang, Sturler and Paulino [177]. As K is SPD the condition number κ can be written as

$$\kappa(K) = \frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}$$

where $\lambda_i(K)$ are eigenvalues of the matrix K. It can then be shown that the condition number can be written in the following way.

$$\kappa(K) = \frac{\max_{||u||_2=1} ||Ku||_2}{\min_{||u||_2=1} ||Ku||_2}$$

As

$$\min_{||u||_2=1} ||Ku||_2 \leq ||Ke_l||_2 = ||c_l||_2 \leq \max_{||u||_2=1} ||Ku||_2$$
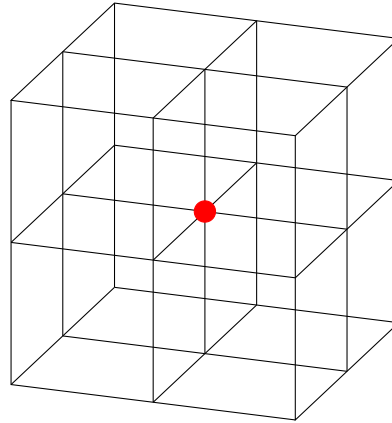
Figure 3-4: Node in the centre of elements

then

$$\max_{i,j\in 1,...,n} \frac{||c_i||_2}{||c_j||_2} \leq \kappa(K) \tag{3.17}$$

A column of the stiffness matrix may be expressed as follows:

$$c_l = \sum_{e\cap\text{D.O.F.}l} x_e^p G_e^T K_e G_e e_l$$

where K_e is the element stiffness matrix of element e and G_e is the corresponding local to global transformation matrix.

Consider a node that is in the centre of all void elements and one which is in the centre of all solid elements.

If l_1 denotes a node in the centre of solid elements, and l_2 denotes a node in the centre of void elements this gives the following formula for the corresponding columns.

$$c_{l_1} = \sum_e G_e^T K_e G_e e_{l_1}$$

$$c_{l_2} = x_{\min}^p \sum_e G_e^T K_e G_e e_{l_2}$$

Hence from (3.17) a lower bound on the condition number is attained.

$$\kappa(K) \geq \frac{||c_{l_1}||_2}{||c_{l_2}||_2} = \frac{1}{x_{\min}^p}$$

With the typical values $x_{\min} = 10^{-3}$ and $p = 3$ this gives $\kappa(K) \geq 10^9$. This analysis is valid for both 2D and 3D structures and it should be noted that it is conservative. It does not take into account any geometry of the problem which, as is well known, can
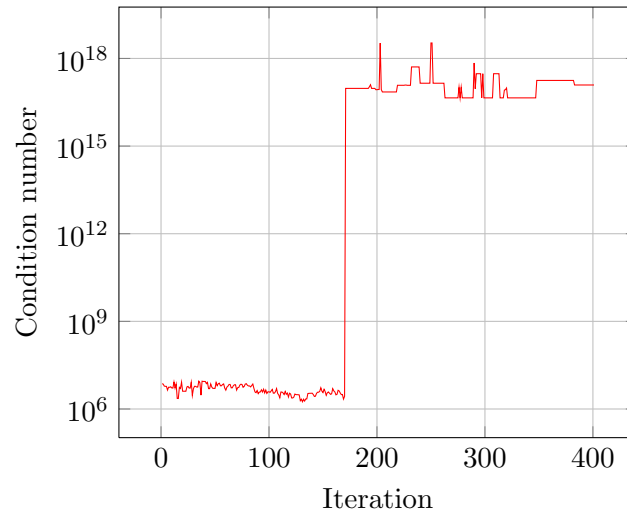
Figure 3-5: Condition number of the stiffness matrix generated by the ESO method applied to the short cantilevered beam. The condition number has been estimated by the linear solver HSL_MA57.

itself lead to highly ill-conditioned stiffness matrices.

In the ESO approach to topology optimization where there is no variation in the density of elements, only the geometry of the underlying structure varies. Figure 3-5 shows an estimate of the condition number of the matrix which is generated by the ESO method when applied to the short cantilevered beam problem, as given by the linear solver HSL_MA57. It is clear from this figure that the matrices are extremely ill-conditioned. In fact after around 170 iterations the linear solver switches to an indefinite mode as the matrices become closer to singular.

Within this section it has been shown that the matrices which arise in topology optimization are highly ill-conditioned. Indeed, with matrices as ill-conditioned as those shown in Figure 3-5 this may hint at the possibility that the structure is almost disconnected and a careful look at subsequent analyses may be warranted.

3.4   Derivation of stress stiffness matrices

In this section it is shown how the stability analysis is derived, and in doing so an explicit expression for the stress stiffness matrix is found. This is an elaboration of the derivation given by Cook [39] and a specific example of the more general case given by Oden [121].

Let u, v and w be the displacements in the x, y and z directions respectively.

The notation $\cdot_{,x}$, $\cdot_{,y}$ and $\cdot_{,z}$ mean partial differentiation with respect to x, y and z respectively.

Definition 3.13. *Green-Lagrange strain is defined as follows*

$$\varepsilon_x = u_{,x} + \frac{1}{2}(u_{,x}^2 + v_{,x}^2 + w_{,x}^2) \tag{3.18}$$

$$\varepsilon_y = v_{,y} + \frac{1}{2}(u_{,y}^2 + v_{,y}^2 + w_{,y}^2) \tag{3.19}$$

$$\varepsilon_z = w_{,z} + \frac{1}{2}(u_{,z}^2 + v_{,z}^2 + w_{,z}^2) \tag{3.20}$$

$$\varepsilon_{xy} = u_{,y} + v_{,x} + (u_{,x}u_{,y} + v_{,x}v_{,y} + w_{,x}w_{,y}) \tag{3.21}$$

$$\varepsilon_{yz} = v_{,z} + w_{,y} + (u_{,y}u_{,z} + v_{,y}v_{,z} + w_{,y}w_{,z}) \tag{3.22}$$

$$\varepsilon_{xz} = w_{,x} + u_{,z} + (u_{,x}u_{,z} + v_{,x}v_{,z} + w_{,x}w_{,z}) \tag{3.23}$$

and write

$$\varepsilon = \begin{bmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z & \varepsilon_{xy} & \varepsilon_{yz} & \varepsilon_{xz} \end{bmatrix}^T$$

$$\varsigma = \begin{bmatrix} \sigma_x & \sigma_y & \sigma_z & \sigma_{xy} & \sigma_{yz} & \sigma_{xz} \end{bmatrix}^T$$

Assuming that the initial stresses $\varsigma = \begin{bmatrix} \sigma_x & \sigma_y & \sigma_z & \sigma_{xy} & \sigma_{yz} & \sigma_{xy} \end{bmatrix}^T$ remain constant as strains ε occur. The work done by the structure is then

$$U = \int_V \varepsilon^T \varsigma \, dV \tag{3.24}$$

Consider the product $\varepsilon^T \varsigma$. Using the definitions of Green-Lagrange strain (3.18) -

(3.23) it can be written as follows.

$$\varepsilon^T \varsigma = \varepsilon_x \sigma_x + \varepsilon_y \sigma_y + \varepsilon_z \sigma_z + \varepsilon_{xy} \sigma_{xy} + \varepsilon_{yz} \sigma_{yz} + \varepsilon_{xz} \sigma_{xz} \tag{3.25}$$

$$= u_{,x} \sigma_x + v_{,y} \sigma_y + w_{,z} \sigma_z +$$

$$(u_{,y} + v_{,x}) \sigma_{xy} + (v_{,z} + w_{,y}) \sigma_{yz} + (w_{,x} + u_{,z}) \sigma_{xz} +$$

$$\frac{1}{2}(u_{,x}^2 + v_{,x}^2 + w_{,x}^2)\sigma_x + \frac{1}{2}(u_{,y}^2 + v_{,y}^2 + w_{,y}^2)\sigma_y +$$

$$\frac{1}{2}(u_{,z}^2 + v_{,z}^2 + w_{,z}^2)\sigma_z + (u_{,x}u_{,y} + v_{,x}v_{,y} + w_{,x}w_{,y})\sigma_{xy} +$$

$$(u_{,y}u_{,z} + v_{,y}v_{,z} + w_{,y}w_{,z})\sigma_{yz} + (u_{,x}u_{,z} + v_{,x}v_{,z} + w_{,x}w_{,z})\sigma_{xz} \tag{3.26}$$

$$= \bar{\varepsilon}^T \varsigma +$$

$$\frac{1}{2}(u_{,x}^2 + v_{,x}^2 + w_{,x}^2)\sigma_x + \frac{1}{2}(u_{,y}^2 + v_{,y}^2 + w_{,y}^2)\sigma_y +$$

$$\frac{1}{2}(u_{,z}^2 + v_{,z}^2 + w_{,z}^2)\sigma_z + (u_{,x}u_{,y} + v_{,x}v_{,y} + w_{,x}w_{,y})\sigma_{xy} +$$

$$(u_{,y}u_{,z} + v_{,y}v_{,z} + w_{,y}w_{,z})\sigma_{yz} + (u_{,x}u_{,z} + v_{,x}v_{,z} + w_{,x}w_{,z})\sigma_{xz} \tag{3.27}$$

where $\bar{\varepsilon}^T \varsigma$ is an equivalent, vectorised formulation of $\sigma : \nabla \mathbf{v}$ which appears in equation (3.5).

Define the vector

$$\mathbf{d} = \begin{bmatrix} u_{,x} & u_{,y} & u_{,z} & v_{,x} & v_{,y} & v_{,z} & w_{,x} & w_{,y} & w_{,z} \end{bmatrix}^T$$

then multiplying out the matrix vector products shows

$$\varepsilon^T \varsigma = \bar{\varepsilon}^T \varsigma + \frac{1}{2}\mathbf{d}^T \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & \sigma \end{bmatrix} \mathbf{d}$$

Substituting this into equation (3.24) gives

$$U = \int_V \bar{\varepsilon}^T \varsigma \mathrm{d}V + \frac{1}{2}\int_V \mathbf{d}^T \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & \sigma \end{bmatrix} \mathbf{d}\mathrm{d}V \tag{3.28}$$

If $\mathbf{v}$ are the nodal degrees of freedom then $\mathbf{d}$ and $\mathbf{v}$ are related via the equation

$$\mathbf{d} = \mathbf{G}\mathbf{v}$$

where $\mathbf{G}$ is a matrix containing derivatives of the basis functions. Substituting this

into (3.28) gives

$$U = \frac{1}{2}\mathbf{v}^T K \mathbf{v} + \frac{1}{2}\mathbf{v}^T K_\sigma \mathbf{v}$$
$$= \frac{1}{2}\mathbf{v}^T (K + K_\sigma)\mathbf{v}$$

where

$$K_\sigma = \int_\Omega \mathbf{G}^T \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & \sigma \end{bmatrix} \mathbf{G} dV \tag{3.29}$$

It should be noted that the stress stiffness matrix K_σ is not necessarily definite, i.e. in certain circumstances it is possible to find vectors x_+ and x_- such that

$$x_+^T K_\sigma x_+ > 0 \quad \text{and} \quad x_-^T K_\sigma x_- < 0.$$

The problem which is needed to be solved, as described by Bathe [16] is as follows: Find the smallest positive λ such that

$$\det(K + \lambda K_\sigma) = 0 \tag{3.30}$$

If $\lambda \leq 1$ then the system will be unstable. The *critical load* of the structure is λ times the applied load. Note that this is a symmetric generalised eigenvalue problem (as both K and K_σ are symmetric) and as such $\lambda \in \mathbb{R}$. However as K_σ is not guaranteed positive semidefinite there may exist $\lambda < 0$. Calculation of the smallest positive eigenvalue and corresponding eigenvector is non-trivial, and indeed finding an efficient method for this is the subject of Section 3.5.

3.5   Calculation of the critical load

Calculating the smallest positive eigenvalue of the system (3.30) is not trivial, and indeed can take up a lot of computational time. A very efficient method for calculating the smallest eigenvalue in modulus is *inverse iteration* (see for example Golub and Van Loan [58]). However, as K_σ is not necessarily positive definite [39] this would not be guaranteed to find a positive eigenvalue. Also, it may be necessary (for reasons which will be set out in Chapter 6) to know a number of the smallest positive eigenvalues and their associated eigenvectors.

It is possible to make a spectral transformation to take the eigenvalues of interest to one end of the spectrum.
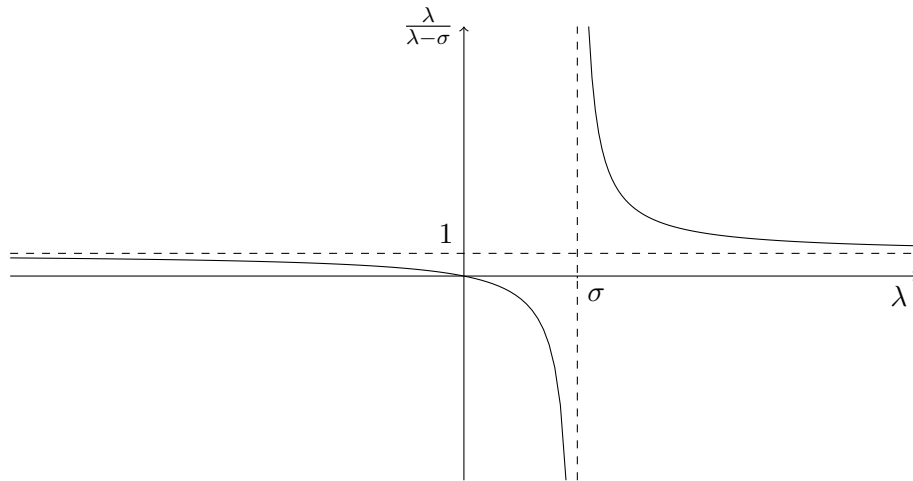
Proposition 3.14. *Suppose K is a symmetric positive definite matrix and that $\sigma \neq 0$ is a scalar shift. Then $(\lambda, \mathbf{x})$ is an eigenpair of*

$$K\mathbf{x} = \lambda M\mathbf{x} \tag{3.31}$$

if and only if $(\frac{\lambda}{\lambda-\sigma}, \mathbf{x})$ is an eigenpair of

$$(K - \sigma M)^{-1} K\mathbf{x} = \mu \mathbf{x} \tag{3.32}$$

Proof. Firstly, let $\mu = \frac{\lambda}{\lambda-\sigma}$. This is equivalent to $\lambda = \frac{-\sigma\mu}{1-\mu}$. Now suppose (3.31) holds.

$$K\mathbf{x} = \lambda M\mathbf{x}$$
$$\iff \qquad K\mathbf{x} = \tfrac{-\sigma\mu}{1-\mu} M\mathbf{x}$$
$$\iff \quad (1-\mu)K\mathbf{x} = -\sigma\mu M\mathbf{x}$$
$$\iff \quad K\mathbf{x} - \mu K\mathbf{x} = -\sigma\mu M\mathbf{x}$$
$$\iff \qquad K\mathbf{x} = \mu(K - \sigma M)\mathbf{x}$$
$$\iff \quad (K - \sigma M)^{-1}K\mathbf{x} = \mu\mathbf{x}$$

$\square$

Proposition 3.15. *The spectral transformation given in Proposition 3.14 maps the smallest positive eigenvalues of (3.31) to the largest eigenvalues of (3.32).*

Proof. Suppose $\sigma > 0$ and that $\lambda < \sigma$ is an eigenvalue of (3.31). Then $\lambda - \sigma < \lambda$ and so dividing by $\lambda - \sigma$ gives $1 > \frac{\lambda}{\lambda-\sigma}$. Hence all eigenvalues that lie to the left of the shift

get mapped to an eigenvalue of the system (3.31) that is less than 1.

Suppose now that $\lambda > \sigma$. Hence $0 < \lambda - \sigma < \lambda$, thus $1 < \frac{\lambda}{\lambda - \sigma}$ so any eigenvalues to the right of the shift are mapped to eigenvalues of (3.32) that are larger than those that were to the left of the shift. If there are two eigenvalues to the right of the shift so that $0 < \sigma < \lambda_1 < \lambda_2$ then

$$\lambda_1 < \lambda_2$$
$$\iff \quad -\lambda_1 > -\lambda_2$$
$$\iff \quad -\sigma\lambda_1 > -\sigma\lambda_2$$
$$\iff \quad \lambda_1\lambda_2 - \sigma\lambda_1 > \lambda_1\lambda_2 - \sigma\lambda_2$$
$$\iff \quad \lambda_1(\lambda_2 - \sigma) > \lambda_2(\lambda_1 - \sigma)$$
$$\iff \quad \frac{\lambda_1}{\lambda_1 - \sigma} > \frac{\lambda_2}{\lambda_2 - \sigma}$$

$\square$

One possible method to calculate the required eigenpairs is the *Arnoldi method* which is implemented in ARPACK [100] for large sparse matrices and makes use of the spectral transformation from Proposition 3.14 ($M = -K_\sigma$ to solve the buckling equations). The drawback of this method is that it requires a linear solve of the form

$$(K - \sigma M)x = b \tag{3.33}$$

which can be computationally prohibitive when the number of design variables (and hence the dimension of (3.33)) increases.

Another method to compute the eigenpairs required is *subspace iteration* and has been implemented in the package HSL_EA19. This has the advantage that it does not require a solve of the form (3.33) as with ARPACK. It instead only requires that an approximation (or preconditioner) to (3.33) be supplied. When a full solve has been performed, the performance of HSL_EA19 is similar to that of ARPACK. The choice of the preconditioner is key to the performance of the algorithm. In general, the better this solve is approximated, the fewer iterations it will take to converge. The choice of preconditioner is exceptionally broad, and indeed it is possible to not do this operation which is equivalent to choosing the identity as the preconditioner. When the problem size is large, this is shown to pay off with the overall computation time of the algorithm significantly decreasing. However as the shift made will be small, a reasonable approximation to (3.33) is to use precomputed factors of K as a preconditioner. If the factors have already been computed then the performing a solve
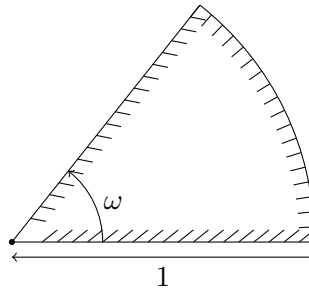
Figure 3-6: Wedge domain for the Laplace problem

with them is inexpensive and the subspace iteration algorithm should converge faster than using the identity as a preconditioner.

In the rest of this thesis when the buckling load of a structure is computed, HSL_EA19 is used with the Cholesky factorisation of K as the preconditioner. This will be used extensively in Chapter 6.

3.6   Re-entrant corner singularities

3.6.1   Laplace's equation

Consider Laplace's equation over the domain Ω with boundary $\partial\Omega$.

$$-\nabla^2 u = 0 \qquad \text{in } \Omega \tag{3.34a}$$

$$u = f \qquad \text{on } \partial\Omega \tag{3.34b}$$

In polar coordinates (r, θ) the Laplace operator in (3.34a) can be written as follows:

$$\nabla^2 u = \frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta} \tag{3.35}$$

If our domain is a wedge with angle ω from the horizontal as in Figure 3-6 then in the case of homogeneous Dirichlet boundary conditions adjacent to the origin it is possible to write down a solution to this problem as follows. The boundary conditions in (3.34b) become

$$u = 0 \text{ on } \theta = 0 \text{ and } \theta = \omega$$

and

$$u = f(\theta) \text{ on } r = 1.$$
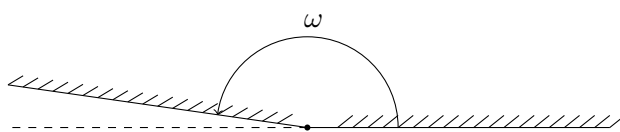
Figure 3-7: Domain for the Laplace problem with no singularity.
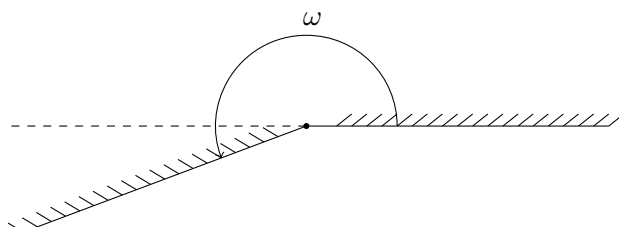
Figure 3-8: Domain for Laplace's equation with a re-entrant corner which gives a singularity at the origin.

Consider

$$u = \sum_{n=1}^{\infty} c_n r^{n\pi/\omega} \sin(\tfrac{n\pi\theta}{\omega}) \tag{3.36}$$

where c_n are the Fourier coefficients of $f(\theta)$ given by

$$c_n = \tfrac{2}{\omega} \int_0^{\omega} f(\theta) \sin \frac{n\pi\theta}{\omega} \ \mathrm{d}\theta.$$

A simple calculation shows that u given in (3.36) is the solution to Laplace's equation (3.34) on the wedge domain given in Figure 3-6. The growth of u is of interest as the origin approaches, so consider $\frac{\partial u}{\partial r}$.

$$\frac{\partial u}{\partial r} = \sum_{n=1}^{\infty} c_n \tfrac{n\pi}{\omega} r^{(n\pi/\omega - 1)} \sin(\tfrac{n\pi\theta}{\omega}) \tag{3.37}$$

As $r \to 0$, $\frac{\partial u}{\partial r} \to 0$ if $\left(\frac{n\pi}{\omega} - 1\right) > 0$ for all $n \in \mathbb{N}$. However if $\left(\frac{n\pi}{\omega} - 1\right) < 0$ for any $n \in \mathbb{N}$ then $\frac{\partial u}{\partial r} \to \infty$. This reduces to

$$\frac{\partial u}{\partial r} \to 0 \text{ when } \omega < \pi$$

$$\frac{\partial u}{\partial r} \to \infty \text{ when } \omega > \pi.$$

The domain shown in Figure 3-8 shows what is known as a *re-entrant corner*. This is a corner that is protruding into the interior of the domain, as opposed to a *salient*
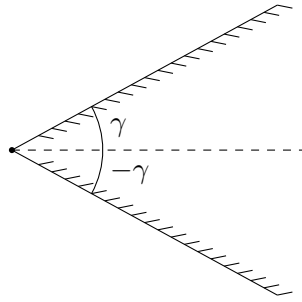
Figure 3-9: Wedge domain for the elasticity problem

corner which is seen in 3-7. As has just been shown, these two different types of corner can produce quantitatively different solutions.

3.6.2   Elasticity singularities

Consider a wedge with angle γ as shown in Figure 3-9. Green and Zerna 1968 [61] have stated that the equations of elasticity have the following form:

$$2\mu(u_x + iu_y) = \kappa\phi(z) - z\overline{\phi'(z)} - \overline{\psi(z)}$$

and seek to find solutions of the form

$$\phi(z) = A_1 z^\lambda + A_2 z^{\overline{\lambda}} \qquad (3.38a)$$

$$\psi(z) = B_1 z^\lambda + B_2 z^{\overline{\lambda}} \qquad (3.38b)$$

that satisfy the homogeneous boundary conditions

$$\sigma_{\theta\theta} - i\sigma_{r\theta} = 0 \qquad \text{on } \theta = \pm\gamma.$$

This boundary condition is written in the form

$$\phi(z) - z\overline{\phi'(z)} - \overline{\psi(z)} = 0 \qquad \text{on } \theta = \pm\gamma \qquad (3.39)$$

and represents a free Dirichlet boundary. This type of boundary will occur in the interior of a structure when material is removed from it. Hence this situation is representative of the elastic behaviour of a structure in the process of topology optimization.

Consider now (3.38) and imposing the condition that both ϕ and ψ be continuous immediately gives $\text{Re}(\lambda) > 0$. Also, it is clear that if $\text{Re}(\lambda) \geq 1$ then as $z \to 0$, $\phi(z) \to 0$ and $\psi(z) \to 0$ and so there is no singularity. Hence eigenvalues of (3.39) with

the property that $0 < \mathrm{Re}(\lambda) < 1$ are the eigenvalues of interest.

For completion, let us list some calculations:

$$\phi(z) = A_1 r^\lambda e^{i\theta\lambda} + A_2 r^{\overline{\lambda}} e^{i\theta\overline{\lambda}} \tag{3.40a}$$

$$\phi'(z) = A_1 \lambda r^{\lambda-1} e^{i\theta(\lambda-1)} + A_2 \overline{\lambda} r^{\overline{\lambda}-1} e^{i\theta(\overline{\lambda}-1)}$$

$$\overline{\phi'(z)} = \bar{A}_1 \overline{\lambda} r^{\overline{\lambda}-1} e^{-i\theta(\overline{\lambda}-1)} + \overline{A_2} \lambda r^{\lambda-1} e^{-i\theta(\lambda-1)}$$

$$z\overline{\phi'(z)} = \bar{A}_1 \overline{\lambda} r^{\overline{\lambda}} e^{-i\theta(\overline{\lambda}-2)} + \overline{A_2} \lambda r^{\lambda} e^{-i\theta(\lambda-2)} \tag{3.40b}$$

$$\psi(z) = B_1 r^\lambda e^{i\theta\lambda} + B_2 r^{\overline{\lambda}} e^{i\theta\overline{\lambda}} \tag{3.40c}$$

$$\overline{\psi(z)} = \overline{B_1} r^{\overline{\lambda}} e^{-i\theta\overline{\lambda}} + \overline{B_2} r^{\lambda} e^{-i\theta\lambda} \tag{3.40d}$$

Now put (3.40) into (3.39) and equating coefficients of r^λ on the boundary $\theta = \gamma$, $r^{\overline{\lambda}}$ on the boundary $\theta = \gamma$, r^λ on $\theta = -\gamma$ and $r^{\overline{\lambda}}$ on $\theta = -\gamma$ respectively, gives the following 4 equations:

$$A_1 e^{i\gamma\lambda} - \overline{A_2} \lambda e^{-i\gamma(\lambda-2)} - \overline{B_2} e^{-i\gamma\lambda} = 0$$

$$-\overline{A_1} \overline{\lambda} e^{-i\gamma(\overline{\lambda}-2)} + A_2 e^{i\gamma\overline{\lambda}} - \overline{B_1} e^{-i\gamma\overline{\lambda}} = 0$$

$$A_1 e^{-i\gamma\lambda} - \overline{A_2} \lambda e^{-i\gamma(\lambda-2)} - \overline{B_2} e^{i\gamma\lambda} = 0$$

$$-\overline{A_1} \overline{\lambda} e^{i\gamma(\overline{\lambda}-2)} + A_2 e^{-i\gamma\overline{\lambda}} - \overline{B_1} e^{i\gamma\overline{\lambda}} = 0$$

As nonzero A_i and B_i terms are required they can be removed from the formulation by looking for when the following condition on this determinant holds:

$$\begin{vmatrix} e^{i\gamma\lambda} & \lambda e^{-i\gamma(\lambda-2)} & 0 & e^{-i\gamma\lambda} \\ \overline{\lambda} e^{-i\gamma(\overline{\lambda}-2)} & e^{i\gamma\overline{\lambda}} & e^{-i\gamma\overline{\lambda}} & 0 \\ e^{-i\gamma\lambda} & \lambda e^{i\gamma(\lambda-2)} & 0 & e^{i\gamma\lambda} \\ \overline{\lambda} e^{i\gamma(\overline{\lambda}-2)} & e^{-i\gamma\overline{\lambda}} & e^{i\gamma\overline{\lambda}} & 0 \end{vmatrix} = 0 \tag{3.42}$$

Now let us seek a purely real eigenvalue, i.e. $\lambda = \overline{\lambda}$. Through a large calculation (or with the help of symbolic computations), this determinant can be reduced to the equation:

$$\lambda^2 \sin^2(2\gamma) - \sin^2(2\lambda\gamma) = 0$$

This equation has a trivial solution at $\lambda = 1$ for all γ. The rest of the solutions to this equation are plotted in Figure 3-10.

The smallest value of λ which solves the above equation for a given γ is of interest, as this eigenvalue will determine the singularity at the corner. In the range $0 < \gamma \leq \pi/2$ the smallest eigenvalue is $\lambda = 1$. Hence for all salient corners, there is no stress
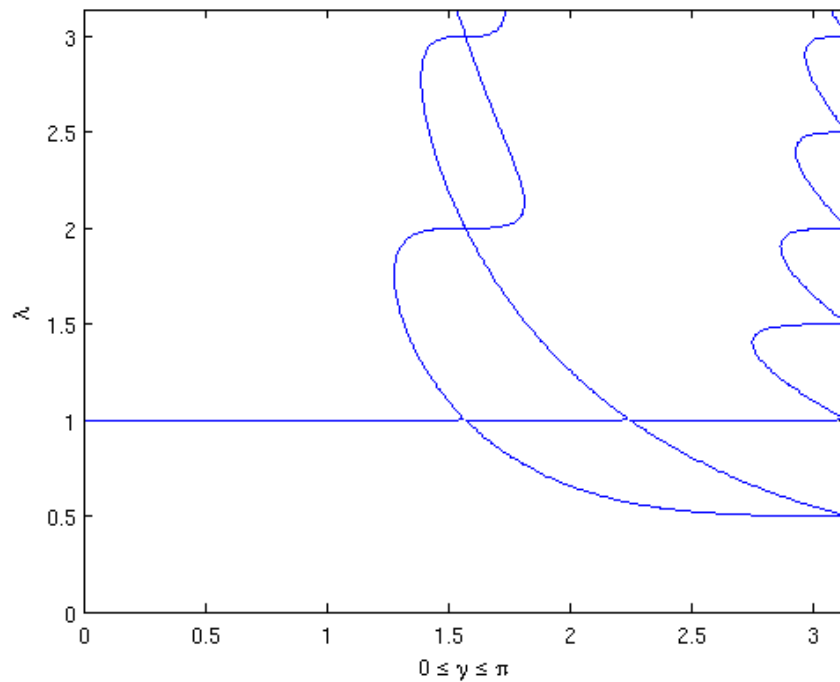
Figure 3-10: Solution space of $\lambda^2 \sin(2\gamma) - \sin^2(2\lambda\gamma) = 0$ for real valued λ.

singularity.

There is a clear bifurcation at the point $\gamma = \pi/2$, and for $\pi/2 < \gamma < \pi$ the eigenvalue is strictly less than 1. Hence for any re-entrant corner, stress singularities occur. The eigenvalue λ appears to be monotonically decreasing in this range, and the slit domain $\gamma \to \pi$ is the worst case singularity corresponding to $\lambda = 0.5$. At $\gamma = \frac{3\pi}{4}$, corresponding to a right angled re-entrant corner, $\lambda \approx 0.5445$.

We have so far not considered the possibility that an eigenvalue with non-zero imaginary part could have smaller real part than those found above. However Karp and Karal [84] claimed this is not the case. If there is a non-zero imaginary part, then (3.42) reduces to the following:

$$\lambda\overline{\lambda}(\sin^2(2\gamma)) = \sin(2\gamma\lambda)\sin(2\gamma\overline{\lambda}) \tag{3.43}$$

Now using double angle formulae and writing the complex eigenvalue $\lambda = x + iy$,

(3.43) can be rearranged to give the following:

$$\lambda\bar{\lambda}(\sin^2(2\gamma)) = \sin(2\gamma\lambda)\sin(2\gamma\bar{\lambda})$$

$$\Longleftrightarrow \quad (x^2 + y^2)(\sin^2(2\gamma)) = [\sin(2\gamma x)\cosh(2\gamma y) + i\cos(2\gamma x)\sinh(2\gamma y)]\times$$

$$[\sin(2\gamma x)\cosh(2\gamma y) - i\cos(2\gamma x)\sinh(2\gamma y)]$$

$$= \sin^2(2\gamma x)\cosh^2(2\gamma y) + \cos^2(2\gamma x)\sinh^2(2\gamma y) \quad (3.44)$$

$$= \sin^2(2\gamma x)\cosh^2(2\gamma y) + (1 - \sin^2(2\gamma x))\sinh^2(2\gamma y)$$

$$= \sin^2(2\gamma x)[\cosh^2(2\gamma y) - \sin^2(2\gamma x)] + \sinh^2(2\gamma y)$$

$$= \sin^2(2\gamma x) + \sinh^2(2\gamma y) \quad (3.45)$$

Thus

$$x^2\sin^2(2\gamma) - \sin^2(2\gamma x) = \sinh^2(2\gamma y) - y^2\sin^2(2\gamma) \quad (3.46)$$

(a) Small range $-0.1 < y < 0.1$ (b) Large range $-2 < y < 2$
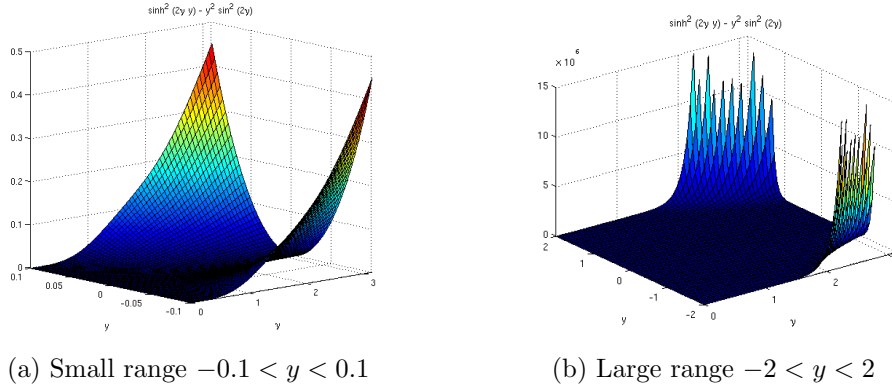
Figure 3-11: Plot of $\sinh^2(2\gamma y) - y^2\sin^2(2\gamma)$

Lemma 3.16. *For all $y \in \mathbb{R}\backslash\{0\}$ and $\gamma \in (0, \pi)$*

$$\sinh^2(2\gamma y) - y^2\sin^2(2\gamma) > 0 \quad (3.47)$$

Proof. When $\gamma \neq \pi/2$, $\sin^2(2\gamma) > 0$ and so (3.47) is equivalent to

$$\frac{\sinh^2(2\gamma y)}{\sin^2(2\gamma)} - y^2 > 0$$

Using Taylor's theorem to expand the left hand side in powers of γ and y gives

$$\frac{\sinh^2(2\gamma y)}{\sin^2(2\gamma)} = \frac{(2\gamma)^2 y^2}{\sin^2(2\gamma)} + \frac{(2\gamma)^4 y^4}{3\sin^2(2\gamma)} + \frac{2(2\gamma)^6 y^6}{45\sin^2(2\gamma)} + \frac{(2\gamma)^8 y^8}{315\sin^2(2\gamma)} + \dots \quad (3.48)$$

Now note that

$$\frac{(2\gamma)^2 y^2}{\sin^2(2\gamma)} - y^2 = \frac{(2\gamma)^2}{3} + \frac{(2\gamma)^4}{15} + \frac{2(2\gamma)^6}{189} + \frac{(2\gamma)^8}{675} + \frac{2(2\gamma)^{10}}{10395} + \dots \tag{3.49}$$

So combining (3.48) and (3.49) shows that every term in the summation is positive and thus the result holds for $\gamma \neq \pi/2$. When $\gamma = \pi/2$ then

$$\sinh^2(2\gamma y) - y^2 \sin^2(2\gamma) = \sinh^2(\pi y) > 0$$

and thus the result holds. $\qquad\square$
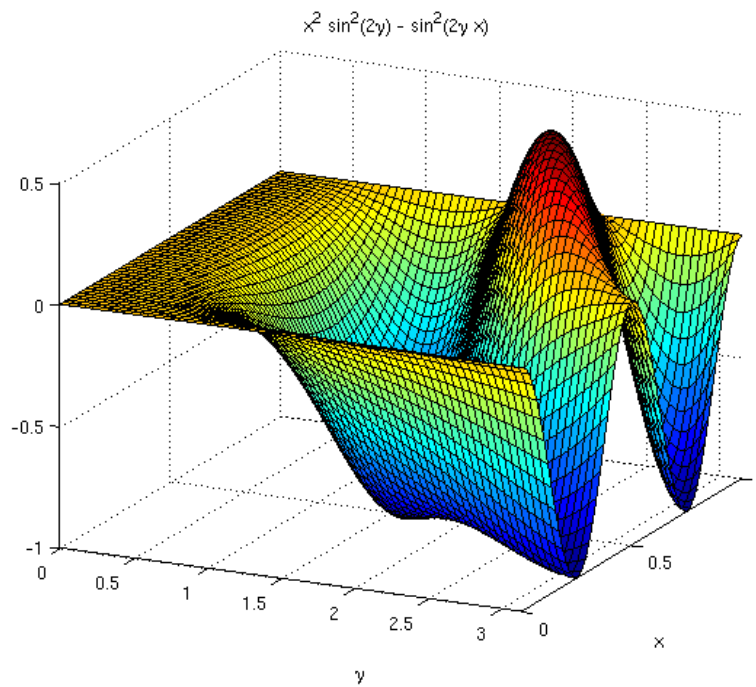
Figure 3-12: Plot of $x^2 \sin^2(2\gamma) - \sin^2(2\gamma x)$

Lemma 3.17. *For $0 < x < \bar{x}(\gamma)$*

$$x^2 \sin^2(2\gamma) - \sin^2(2\gamma x) < 0$$

where $\bar{x}(\gamma)$ is the smallest positive value of x for which $x^2 \sin^2(2\gamma) - \sin^2(2\gamma x) = 0$.

Proof.

$$\frac{\partial(x^2 \sin^2(2\gamma) - \sin^2(2\gamma x))}{\partial x} = -2\sin(4\gamma x) - x(\cos(4\gamma)) + x$$

$$\frac{\partial^2 (x^2 \sin^2(2\gamma) - \sin^2(2\gamma x))}{\partial x^2} = 1 - \cos(4\gamma) - 8\gamma^2 \cos(4\gamma x)$$

$$\left.\frac{\partial (x^2 \sin^2(2\gamma) - \sin^2(2\gamma x))}{\partial x}\right|_{x=0} = 0$$

$$\left.\frac{\partial^2 (x^2 \sin^2(2\gamma) - \sin^2(2\gamma x))}{\partial x^2}\right|_{x=0} = 1 - \cos(4\gamma) - 8\gamma^2 < 0$$

Hence these calculations show that the function is question is less than 0 infinitesimally after the line $x = 0$ and so remains less than 0 up until $x = \bar{x}(\gamma)$.   □

Theorem 3.18. *The eigenvalue with smallest real part which solves (3.42) is purely real.*

Proof. If an eigenvalue has non-zero imaginary part then (3.46) must hold. However, as $y \neq 0$ says that Lemma 3.16 must hold, this implies that

$$x^2 \sin^2(2\gamma) - \sin^2(2\gamma x) > 0.$$

Lemma 3.17 then ensures that $x \geq \bar{x}(\gamma)$ and so the real part of the solution has a larger real part than a purely real solution.   □

Hence the solution shown in Figure 3-10 is representative of the singularity which occurs in elasticity.

Remark 3.19. *Karp and Karal [84] give a proof that the purely real root of (3.43) has smaller real part than a complex root. They do so by examining the solutions to the simultaneous equations*

$$x \sin(2\gamma) = \sin(2\gamma x) \cosh(2\gamma y) \tag{3.50a}$$

$$y \sin(2\gamma) = \cos(2\gamma x) \sinh(2\gamma y) \tag{3.50b}$$

and looking at the properties of these solutions. Squaring and adding the equations (3.50) gives the equation (3.44). However, the solutions to (3.50) are only a particular solution to (3.44) as we could, for example, examine the equations

$$y \sin(2\gamma) = \sin(2\gamma x) \cosh(2\gamma y)$$

$$x \sin(2\gamma) = \cos(2\gamma x) \sinh(2\gamma y)$$

and obtain different solutions which also solve (3.44). Therefore the proof given in [84] is incomplete, but their result still holds thanks to Theorem 3.18.

3.7   Summary

This chapter has shown the derivation of the state equations defining the response of material to an applied load. Properties of the resulting finite-element system have been noted and used to inform the choice of linear solver to be used. The buckling load of a structure has been defined and methods for solving the resulting generalised eigenvalue problem have been reviewed.

Re-entrant corner singularities have been investigated as they occur relentlessly in the ESO method for structural optimization. Categorising these singularities is necessary to understand the behaviour of the algorithm. It is noted that these singularities are inherent in the linear elasticity equations and not simply a numerical error.

4
Survey of optimization methods

In this chapter mathematical optimization methods are surveyed. The knowledge of these methods will inform the choice of optimization strategy which will be employed in the later chapters. Beginning with general definitions in Section 4.1, the simplex method for linear programming is discussed in Sections 4.2 and 4.3. Integer programming methods are covered in Sections 4.4 to 4.6. Nonlinear continuous programming methods are explored in Sections 4.7 to 4.11.

4.1 Preliminary definitions

Consider the general optimization problem as follows.

$$\min_{x} f(x) \tag{4.1a}$$

$$\text{subject to} \quad c_i(x) = 0 \qquad i \in \mathcal{E} \tag{4.1b}$$

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{4.1c}$$

Definition 4.1 (Active set). *The active set of the optimization problem (4.1) at a point x is defined as*

$$\mathcal{A}(x) = \{i \in \mathcal{E} \cup \mathcal{I} \text{ such that } c_i(x) = 0\} \tag{4.2}$$

Definition 4.2. *The Linear Independence Constraint Qualification (LICQ) holds at x^* when the set*

$$\{\nabla c_i(x^*), i \in \mathcal{A}(x^*)\} \tag{4.3}$$

is linearly independent.
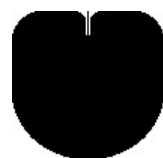
Definition 4.3 (MFCQ). *Let x^* be feasible for* (4.1), *and let $\mathcal{A}_{\mathcal{I}}(x^*) := \{i \in \mathcal{I} : c_i(x) = 0\}$. The Mangasarian-Fromowitz constraint qualification (MFCQ) holds at x^* if there exist a vector $w \in \mathbb{R}^n$ such that*

$$\nabla c_i(x^*)^T w > 0 \quad (i \in \mathcal{A}_{\mathcal{I}}(x^*)),$$
$$\nabla c_i(x^*)^T w = 0 \quad (i \in \mathcal{E}),$$
$$\{\nabla c_i(x^*) : i \in \mathcal{E}\} \quad linearly\ independent$$

Definition 4.4 (Lagrangian function). *The Lagrangian function of* (4.1) *is given by*

$$\mathcal{L}(x, \lambda) := f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x) \tag{4.4}$$

where the variables λ_i are known as Lagrange multipliers.

Definition 4.5 (First order necessary KKT conditions). *Suppose x^* is a local solution of* (4.1) *and $f(x)$ & $c(x)$ are continuously differentiable. Suppose also that the LICQ holds at x^*. Then there exists Lagrange multipliers λ^* with components $\lambda_i^*, i \in \mathcal{E} \cup \mathcal{I}$ such that*

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \tag{4.5a}$$
$$c_i(x^*) = 0 \qquad \forall i \in \mathcal{E} \tag{4.5b}$$
$$c_i(x^*) \geq 0 \qquad \forall i \in \mathcal{I} \tag{4.5c}$$
$$\lambda_i^* \geq 0 \qquad \forall i \in \mathcal{I} \tag{4.5d}$$
$$\lambda_i^* c_i(x^*) = 0 \qquad \forall i \in \mathcal{E} \cup \mathcal{I} \tag{4.5e}$$

A point x that satisfies the first order necessary KKT conditions is known as a *KKT point*.

4.2 Theory of Simplex Method

The simplex algorithm dates from 1947, and owes its origins to Dantzig [40]. At the turn of the millennium, it was named as one of the top 10 algorithms of the 20th century by Simpson [163]. Before discussing the simplex method, some fundamentals about linear programming need to be established.

Definition 4.6 (Linear Programming Problem and Canonical form). *A problem of the*

form

$$\max z = c^T x \qquad subject\ to \tag{4.6a}$$

$$Ax \leq b \tag{4.6b}$$

$$x \geq 0 \tag{4.6c}$$

is known as a linear programming problem *(LPP). Here* $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ *and* $A \in \mathbb{R}^{m \times n}$. *An LPP is said to be in* canonical form *if* (4.6b) *is an* equality *constraint*.

It is a simple exercise to convert any LPP into canonical form by introducing slack variables (see for example Soni 2007 [155] Section 3.3.1.).

Definition 4.7 (Convex set). *A non-empty set* $S \in \mathbb{R}^n$ *is* convex *if for all* x_1, $x_2 \in \mathbb{R}^n$ *and* $\lambda \in (0, 1)$, $\lambda x_1 + (1 - \lambda) x_2 \in S$.

Theorem 4.8 (Dantzig and Thapa [41]). *Any LPP has a feasible region which is either empty or a closed convex polyhedron.*

Proof. The proof is left as an exercise in Dantzig and Thapa [41] exercises 1.11 to 1.13 and is included here for completeness.

Consider the set $\Gamma := \{x \in \mathbb{R}^n | w^T x \leq d\}$ where $w \in \mathbb{R}^n$ and $d \in \mathbb{R}$ are given. Then for $\lambda \in (0, 1)$ and $x_1, x_2 \in \Gamma$:

$$w^T(\lambda x_1 + (1 - \lambda) x_2) = \lambda w^T x_1 + (1 - \lambda) w^T x_2$$
$$\leq \lambda d + (1 - \lambda) d$$
$$= d$$

So $\lambda x_1 + (1 - \lambda) x_2 \in \Gamma$. Note that the same holds if the set Γ is defined by an equality not an inequality.

Now suppose that there are k convex sets $\Gamma_1, \ldots, \Gamma_k$ in $\mathbb{R}^n$ with $\cap_{i=1}^{k} \Gamma_i \neq \emptyset$. It will subsequently be shown that the intersection of these convex sets is also convex. Consider $\lambda \in (0, 1)$ and $x_1, x_2 \in \cap_{i=1}^{k} \Gamma_i$. Note that for all j, $x_1 \& x_2 \in \Gamma_j$. Hence by the convexity of Γ_j, $\lambda x_1 + (1 - \lambda) x_2 \in \Gamma_j$. Since this is true $\forall j = 1, \ldots, k$ then $\lambda x_1 + (1 - \lambda) x_2 \in \cap_{i=1}^{k} \Gamma_i$.

Hence the intersection of a finite number of convex sets is itself a convex set. To see why the feasible region is a polyhedron, note that each row of (4.6b) characterises all the points lying on one side of a hyperplane. These points then form a half space which is trivially closed. By definition, the intersection of a finite number of closed half spaces is called a closed polyhedron.

Justification of *closed* in the above is given by the following. If $C_i, i = 1, \ldots, k$ are closed sets and $\{x^j | j \in \mathbb{N}\}$ is a sequence of points in $\cap_{i=1}^k C_i$ with accumulation point x, then since each C_i is closed, $x \in C_i$. This is true for all $i \in 1, \ldots, k$ and hence $x \in \cap_{i=1}^k C_i$. Hence the intersection of a finite number of closed sets is closed. $\qquad\square$

Definition 4.9 (Extreme point). *Given any nonempty convex set Γ, say $x \in \Gamma$ is an extreme point if x is not an interior point of any line segment in Γ. i.e. $\nexists\, x_1$ & $x_2 \in \Gamma$ and $\lambda \in (0, 1)$ with $x_1 \neq x_2$ such that $x = \lambda x_1 + (1 - \lambda)x_2$.*

It is clear that any closed convex polyhedron, defined by at least the same number of constraints as dimensions of the problem (and hence any feasible region), has at least 1 extreme point and at most a finite number of extreme points.

Definition 4.10 (Basic solutions). *Consider an LPP in canonical form. Say $Ax = b$ has a* basic solution *x if*

$$\mathcal{I} := \{i \mid x_i \neq 0\} \qquad and \qquad |\mathcal{I}| \leq rank(A) \tag{4.7}$$

Moreover, if $x \geq 0$ then x is called a basic feasible solution.

Theorem 4.11. *Suppose that x is a basic feasible solution to a linear programming problem in canonical form. Then x is an extreme point of the feasible region $\mathcal{F}$.*

Proof. Can be found in Nocedal and Wright [120] Theorem 13.3. $\qquad\square$

Theorem 4.12 (Fundamental Theorem of Linear Programming). *Consider a linear programming problem in canonical form. Then if there exists a finite optimal solution then there exists an optimal basic feasible solution.*

Proof. See for example Luenberger and Ye [103]. $\qquad\square$

4.3 Simplex Algorithm

Suppose there is a feasible solution of the form

$$x = \begin{pmatrix} 0_{n-m} \\ x_B \end{pmatrix} \tag{4.8}$$

where $x_B > 0$ represent the basic variables corresponding to the decomposition $A = (A^0 \,|B)$. Note that $x_B = B^{-1}b$.

Now suppose one basic variable is removed from the current solution and another variable is introduced, giving us a feasible solution of the form

$$y = \begin{pmatrix} y^0 \\ y_B \end{pmatrix} \tag{4.9}$$

where y^0 are the first $n - m$ entries and y_B correspond to the old basis variables. As this is feasible this gives

$$Ay = A^0 y^0 + By_B = b$$

which can be rearranged to give

$$y_B = B^{-1}b - B^{-1}A^0 y^0. \tag{4.10}$$

Now considering the objective function at this new solution gives

$$z(y) = c^T y = (c^0)^T y^0 + c_B^T y_B \tag{4.11}$$
$$= (c^0)^T y^0 + c_B^T (B^{-1}b - B^{-1}A^0 y^0) \tag{4.12}$$
$$= c_B^T x_B + [(c^0)^T - c_B^T B^{-1} A^0] y^0 \tag{4.13}$$
$$= z(x) + [(c^0)^T - c_B^T B^{-1} A^0] y^0. \tag{4.14}$$

Thus, in order to improve the objective function, choose a component j such that $[(c^0)^T - c_B^T B^{-1} A^0]_j > 0$ and then set y_j^0 to be non-zero and a component of y_B to zero.

Algorithm 1 Simplex method for LPP

1: Choose a basic feasible solution $(x_0 \ x_B)^T$.
2: Do until $[(c^0)^T - c_B^T B^{-1} A^0]_j \leq 0$:
3: Choose $i \in \{\ell \text{ nonbasic} \mid [(c^0)^T - c_B^T B^{-1} A^0]_\ell > 0\}$
4: **if** $B^{-1} A_i \leq 0$ **then**
5:     {Note A_i is the ith column of A.}
6:     Stop as the problem is unbounded.
7: **else**
8:     Choose $j \in \text{basic}$ so $j = \min_k \{(B^{-1}b)_k/(B^{-1}A_i)_k : (B^{-1}A_i)_k > 0\}$
9: **end if**
10: Make non-basic variable i basic and make basic variable j non-basic.
11: End do
12: Stop as an optimal basic feasible solution has been found.

4.4 Branch-and-Bound

Branch-and-bound methods were first suggested by Land and Doig [97]. Consider the problem

$$\min c^T x \qquad \text{subject to} \qquad (4.15a)$$
$$Ax = b \qquad (4.15b)$$
$$x \geq 0 \qquad (4.15c)$$
$$x \in \mathbb{Z}^n \qquad (4.15d)$$

The first step is to solve (4.15) with the integer constraint on the variables (4.15d) removed, that is $x \in \mathbb{R}^n$. The solution of this problem is then not guaranteed to have integer components. Then choose a variable $j \in \{1, \ldots, n\}$ with noninteger component and define $I_j := \lfloor x_j \rfloor$.

Now it is possible to make the first *branch* into left and right child problems. The left-child problem is to solve

$$\min c^T x \qquad \text{subject to} \qquad (4.16a)$$
$$Ax = b \qquad (4.16b)$$
$$x \geq 0 \qquad (4.16c)$$
$$x_j \leq I_j \qquad (4.16d)$$

and the right-child problem is to solve

$$\min c^T x \qquad \text{subject to} \qquad (4.17a)$$
$$Ax = b \qquad (4.17b)$$
$$x \geq 0 \qquad (4.17c)$$
$$x_j \geq I_j + 1 \qquad (4.17d)$$

This whole process can be recursively applied to create what is known as the binary *enumeration tree*. Repeating this process enough times will find an integer solution. The values of the objective function of these integer solutions are retained and used to prune the tree. If the solution to the continuous problem at a node has objective function higher than that of the current best integer solution then the rest of that branch may be disregarded as any integer solutions belonging to that branch must also have worse objective function.

The methods for choosing the noninteger component x_j on which to branch, and the

choice of where to next look for a solution if the working branch is pruned are important, and different strategies for these form the basis for different implementations. For further details on branch-and-bound methods see for example Moré and Wright [114], Winston [181] or Martí and Reinelt [106].

4.5   Cutting plane methods

Cutting plane methods were first proposed by Gomory [59]. Consider a problem

$$\min c^T x \qquad \text{subject to} \tag{4.18a}$$

$$Ax = b \tag{4.18b}$$

$$x \geq 0 \tag{4.18c}$$

$$x \in \mathbb{Z}^n \tag{4.18d}$$

then, like the branch-and-bound method, solve (4.18) with the integer constraint (4.18d) removed. As this is a linear problem, it is known that the solution must occur at a vertex of the n-dimensional simplex. If this solution is an integer solution then the algorithm stops, as the optimum has been found.

If this solution has non-integer components then a hyperplane is found that lies between this vertex and all the feasible integer points. This hyperplane is then added in as a constraint into (4.18) to exclude that vertex, and this is known as making a *cut*. The new linear programming problem is then solved and the process repeated with added constraints until the solution found has integer components.

4.6   Branch-and-cut methods

As the name suggests, branch-and-cut is a hybrid of branch-and-bound and cutting plane methods.

These methods start by applying a cutting plane method to (4.18) until either a solution is found or no more cutting planes can be computed. If no more cuts can be made, then the branch-and-bound method is started. Some non-integer components of the solution are chosen on which to branch. These new subproblems can then be tackled using cutting plane methods again [107].

4.7   Quadratic Programming

Definition 4.13. *A quadratic program (QP) is a problem of the form*

$$\min_{x\in\mathbb{R}^n} \quad \tfrac{1}{2}x^T G x + x^T c \tag{4.19a}$$

$$\text{subject to} \quad a_i^T x = b_i \qquad i \in \mathcal{E} \tag{4.19b}$$

$$a_i^T x \geq b_i \qquad i \in \mathcal{I} \tag{4.19c}$$

where $c, x \in \mathbb{R}^n$, $G \in \mathbb{R}^{n\times n}$ and $G = G^T$. If G is positive semidefinite, then (4.19) is called a convex QP. The set $\mathcal{E}$ defines the equality constraints, and the set $\mathcal{I}$ defines the inequality constraints.

It has been shown that a QP can always be solved or shown to be infeasible in a finite amount of computation [120]. In the convex QP case, the problem is similar in difficulty to that of a linear program.

Definition 4.14. *A equality constrained quadratic program (EQP) is a problem of the form*

$$\min_{x\in\mathbb{R}^n} \quad q(x) := \tfrac{1}{2}x^T G x + x^T c \tag{4.20a}$$

$$\text{subject to} \quad Ax = b. \tag{4.20b}$$

where $c, x \in \mathbb{R}^n$, $G \in \mathbb{R}^{n\times n}$ and $G = G^T$. $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m\times n}$.

The first order necessary conditions for x^* to be a solution of (4.20) says $\exists \lambda^* \in \mathbb{R}^m$ such that

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix} \tag{4.21}$$

i.e.

$$Gx^* - A^T \lambda^* = -c$$

$$Ax^* = b$$

(4.21) can be rewritten as

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix} \tag{4.22}$$

where $h = Ax - b$, $g = c + Gx$ and $p = x^* - x$.

The matrix

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \tag{4.23}$$

is known as the *KKT matrix*.

Theorem 4.15. *Let Z denote the $n \times (n-m)$ matrix whose columns are a basis for the kernel of A. That is, Z has full rank and satisfies $AZ = 0$. Assume A has full row rank (i.e. all the constraints are linearly independent). Assume $Z^T G Z$ is positive definite (i.e. the reduced Hessian matrix is SPD).*

Then the KKT matrix (4.23) is nonsingular and $\exists (x^*, \lambda^*)$ satisfying (4.21).

Proof. See for example Nocedal and Wright [120] Lemma 16.1. □

If fact, the second order conditions are the satisfied too, so x^* is a strict local minimiser of the EQP. Note that a stronger result than Theorem 4.15 holds, that is, if $Z^T G Z$ is positive definite then the KKT matrix has precisely n positive, m negative and 0 zero eigenvalues (Forsgren et al. 2002 [54]).

Theorem 4.16. *Let A have full row rank and assume $Z^T G Z$ is positive definite. Then x^* satisfying (4.21) is a unique global solution of (4.20).*

Proof. See Nocedal and Wright [120] Theorem 16.2. □

So, if the above assumptions hold, in order to find the global solution to the EQP, only one equation of the form (4.21) must be solved. There are many ways to do this, and choosing the most efficient linear algebra technique is important. Note: the KKT matrix is always indefinite if $Z^T G Z \succ 0$ and $m > 0$.

4.7.1 Inequality constrained Quadratic Programming

An Inequality constrained Quadratic Program (IQP) is a problem of the following form.

$$\min_{x \in \mathbb{R}^n} \quad q(x) = \tfrac{1}{2} x^T G x + x^T c \tag{4.24a}$$

$$\text{subject to} \quad a_i^T x = b_i \qquad i \in \mathcal{E} \tag{4.24b}$$

$$a_i^T x \geq b_i \qquad i \in \mathcal{I} \tag{4.24c}$$

The Lagrangian for the IQP (4.24) is

$$\mathcal{L}(x, \lambda) = \tfrac{1}{2} x^T G x + x^T c - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i (a_i^T x - b_i) \tag{4.25}$$

So any solution x^* of the IQP (4.24) satisfies the first order KKT conditions for some Lagrange multipliers $\lambda_i^*, i \in \mathcal{A}(x^*)$ with

$$Gx^* + c - \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* a_i = 0 \tag{4.26a}$$

$$a_i^T x^* = b_i \qquad \forall i \in \mathcal{A}(x^*) \tag{4.26b}$$

$$a_i^T x^* \geq b_i \qquad \forall i \in \mathcal{I} \backslash \mathcal{A}(x^*) \tag{4.26c}$$

$$\lambda_i^* \geq 0 \qquad \forall i \in \mathcal{I} \cap \mathcal{A}(x^*) \tag{4.26d}$$

Note here there is no need to have the LICQ as, in QP problems, the constraints are linear and thus the LICQ are automatically satisfied.

Theorem 4.17. *If x^* satisfies (4.26) for some $\lambda_i^*, i \in \mathcal{A}(x^*)$ and the matrix G is positive semidefinite then x^* is a global solution of the IQP (4.24).*

Proof. See for example Nocedal and Wright [120], Theorem 16.4. □

If the contents of the optimal active set were known in advance, the solution x^* could be found by applying the techniques of EQP to

$$\min_x \quad q(x) = \tfrac{1}{2} x^T G x + x^T c$$
$$\text{subject to} \qquad a_i^T x = b_i \qquad \forall i \in \mathcal{A}(x^*)$$

Normally $A(x^*)$ is not known so determining this set is the main challenge of active set methods for IQPs. The simplex method is an active set method of linear programming. Active set methods for QPs differ in that the iterates (and the solution) are not necessarily vertices of the feasible region.

Interior point methods

Interior point methods may be extended from linear programming to convex QPs and an alternative to active set methods. Consider the problem

$$\min_x \quad q(x) = \tfrac{1}{2} x^T G x + x^T c$$
$$\text{subject to} \qquad Ax \geq b$$

The KKT conditions are then

$$
\begin{aligned}
Gx - A^T\lambda + c &= 0 \\
Ax - b &\geq 0 \\
(Ax - b)_i\lambda_i &= 0 \qquad i = 1, \ldots, m \\
\lambda &\geq 0
\end{aligned}
$$

Now introduce a *slack* vector $y \geq 0$ so that

$$
\begin{aligned}
Gx - A^T\lambda + c &= 0 \\
Ax - y - b &= 0 \\
y_i\lambda_i &= 0 \qquad i = 1, \ldots, m \\
(y, \lambda) &\geq 0
\end{aligned}
$$

As $G \succeq 0$, these KKT conditions are necessary and sufficient for optimality, however it may be impossible to satisfy these conditions if there are no feasible points. Given a current feasible iterate (x, y, λ) with $(y, \lambda) \geq 0$, it is possible to define a complementarity measure μ by $\mu = y^T\lambda/m$. Now consider the perturbed KKT conditions given by,

$$
F(x, y, \lambda; \sigma\mu) = \begin{bmatrix} Gx - A^T\lambda + c \\ Ax - y - b \\ y_1\lambda_1 - \sigma\mu \\ \vdots \\ y_m\lambda_m - \sigma\mu \end{bmatrix} = 0, \qquad \sigma \in [0, 1] \tag{4.27}
$$

The solutions (y, λ) of (4.27) define the *central path*, which is a trajectory that leads to the solution of the QP as $\sigma\mu$ tends to 0. If fixing μ and applying Newton's method to (4.27) leads to the linear system

$$
\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \mathrm{diag}(\lambda) & \mathrm{diag}(y) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -y_1\lambda_1 + \sigma\mu \\ \vdots \\ -y_m\lambda_m + \sigma\mu \end{bmatrix} \tag{4.28}
$$

where

$$r_d = Gx - A^T\lambda + c$$
$$r_p = Ax - y - b$$

Then

$$(\widetilde{x}, \widetilde{y}, \widetilde{\lambda}) = (x, y, \lambda) + \alpha(\Delta x, \Delta y, \Delta \lambda)$$

where α is chosen to retain the inequality $(\widetilde{y}, \widetilde{\lambda}) \geq 0$. For a comprehensive discussion of interior point methods, see for example Nocedal and Wright [120].

4.8   Line search methods for unconstrained problems

In a line search method, given a point x_k the goal is to iterate to the minimiser x^*. The next point is given by moving a distance α_k along a search direction p_k. The next iteration point is then

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.29}$$

α_k is known as the *step length*.

If the search direction p_k has the property that $p_k^T \nabla f_k < 0$ then p_k is known as a *descent direction*. This ensures that f must take a lower value at some point along the search direction. In the steepest descent method, the search direction is taken to be

$$p_k = -\nabla f_k \tag{4.30}$$

In Newton's method, the search direction is taken to be

$$p_k = -\nabla^2 f(x_k) \nabla f_k \tag{4.31}$$

Quasi-Newton methods use a (normally positive definite) approximation to the Hessian instead of the exact Hessian to compute the search direction.

The computation of the step length α_k has to ensure that there is sufficient decrease in the objective function, however this choice must be made efficiently. The best choice of the step length is

$$\alpha_k = \arg\min_\alpha f(x_k - \alpha p_k), \qquad \alpha > 0 \tag{4.32}$$

In general, finding even a local minimum of this one dimensional minimisation problem is expensive and unjustified and so practical methods choose α_k so that it achieves sufficient improvement in the objective function according to some measure.

4.9   Trust region methods

Key to the idea of a trust region, is a model m_k that approximates the objective function f. The *trust region* is then the area around the current iterate within which the model is trusted to be a fair representation of the objective function. They then choose the next iterate to be the minimizer of the model within this trust region.

The choice of the size of the trust region is crucial to the performance of the method. The region must be large enough for each step to allow good improvement of the objective function, but it must not be so big that the model function m_k no longer approximates the objective function f effectively.

If the region is too large, and the minimiser of the model function within the trust region actually gives an increase in the objective function f, then the step is rejected and the trust region may be reduced. If over the history of the iterations the model function sufficiently tracks the objective function then it is assumed that the trust region is conservative. In this case the size of the trust region is increased in order to speed up convergence.

4.10   Sequential Quadratic Programming

4.10.1   Newton Formulation

Consider a equality constrained problem:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{4.33a}$$

$$\text{subject to } c(x) = 0 \tag{4.33b}$$

The Lagrangian function for this problem is then given by

$$\mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x) \tag{4.34}$$

Let $A(x)$ denote the Jacobian of the constraints, i.e. with $c_i(x)$ denoting the ith component of vector $c(x)$ then $A(x)^T = [\nabla c_1(x), \nabla c_2(x), \ldots, \nabla c_m(c)]$.

The first order KKT conditions of (4.33) are

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0 \tag{4.35}$$

Any solution (x^*, λ^*) of (4.33) for which $A(x^*)$ has full rank satisfies (4.35). Newton's

method to find the solution of (4.35) is to solve

$$
\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x, \lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} \nabla f_k(x) + A_k(x)^T \lambda_k \\ -c_k(x) \end{bmatrix} \tag{4.36a}
$$

then

$$
\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} \tag{4.36b}
$$

The matrix in (4.36a) is nonsingular if both

$$
A(x) \text{ has full row rank, and} \tag{4.37a}
$$

$$
d^T \nabla_{xx}^2 \mathcal{L}(x, \lambda) d > 0 \quad \forall d \neq 0 \text{ s.t. } A(x)d = 0. \tag{4.37b}
$$

Equations (4.37) are equivalent to the LICQ holding and that the reduced Hessian matrix is positive definite. Compare these with the assumptions of Theorem 4.15. In this case solving the system in (4.36b) gets closer to the solution locally, but a merit function is required to ensure global convergence.

4.10.2 Taylor's series expansion

Consider the Taylor's series expansion of $f(x)$ in one variable.

$$
\begin{aligned}
f(x) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n \\
&= f(x_0) + \frac{\partial f}{\partial x}(x_0)(x - x_0) + \tfrac{1}{2}\frac{\partial^2 f}{\partial x^2}(x_0)(x - x_0)^2 + \tfrac{1}{6}\frac{\partial^3 f}{\partial x^3}(x_0)(x - x_0)^3 + \text{h.o.t.}
\end{aligned}
\tag{4.38}
$$

which can be generalised to the multidimensional case as follows.

$$
f(\mathbf{x}) = f(\mathbf{x_0}) + (\mathbf{x} - \mathbf{x_0})^T \nabla f(\mathbf{x_0}) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x_0})^T \nabla^2 f(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0}) + \text{h.o.t.} \tag{4.39}
$$

This expansion can be used to locally model a general nonlinear function f as a quadratic function, and is used extensively in SQP.

4.10.3 SQP Formulation

In this section, the notation is shortened to drop the dependency on x and λ, i.e. $A_k(x)$ becomes A_k and $\mathcal{L}_k(x, \lambda_k)$ becomes $\mathcal{L}_k$ etc.

Alternatively to (4.36), (4.33) can be viewed as a quadratic program: At iterate

(x_k, λ_k),

$$\min_{p} \quad f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \tag{4.40a}$$

$$\text{subject to} \quad A_k p + c_k = 0 \tag{4.40b}$$

Hence, if the assumptions (4.37) hold, then (4.40) has a unique solution (p_k, ℓ_k) satisfying

$$\nabla_{xx}^2 \mathcal{L}_k p + \nabla f_k - A_k^T \ell_k = 0 \tag{4.41a}$$

$$A_k p_k + c_k = 0 \tag{4.41b}$$

This pair (p_k, ℓ_k) can be identified with the solution of the Newton system. To see this, consider (4.36a):

$$\nabla_{xx}^2 \mathcal{L}_k p_k - A_k^T p_\lambda = -\nabla f_k + A_k^T \lambda_k$$

$$A_k p_k = -c_k$$

This first equation can be rearranged to give

$$\nabla_{xx}^2 \mathcal{L}_k p_k - A_k^T (p_k + \lambda_k) = \nabla_{xx}^2 \mathcal{L}_k p_k - A_k^T \lambda_{k+1} = -\nabla f_k$$

Hence $\lambda_{k+1} = \ell_k$ & p_k solve both the Newton step (4.36) and the SQP subproblem (4.40).

4.10.4 Line search SQP method

The basic line search method for SQP is to iterate over k the update formula

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} \alpha_k^x p_k \\ \alpha_k^\lambda p_\lambda \end{bmatrix} \tag{4.42}$$

where the α_k^x and α_k^λ are nonnegative stepsizes that are to be found.

To find the stepsizes, it is necessary to have the concept of a *merit function* $\phi(x)$. This merit function is a measure of distance to a critical point. Thus the stepsize is α_k^x is found by requiring that $\phi(x_k + \alpha_k^x p_k)$ be sufficiently smaller than $\phi(x_k)$.

For the choice of different merit functions, see Conn et Al, [38]. There they discuss Augmented Lagrangian penalty functions and smooth/nonsmooth exact penalty functions.

4.10.5 Trust region SQP method

A typical trust region SQP method is, at an iteration point x_k, solve a subproblem of the form

$$\min_{p} \quad f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \tag{4.43a}$$

$$\text{subject to} \quad A_k p + c_k = 0 \tag{4.43b}$$

$$\text{and} \quad ||p|| \leq \Delta_k \tag{4.43c}$$

for some suitable trust region radius Δ_k with corresponding choice of norm. The solution of this subproblem will only be accepted as the next iterate $(x_{k+1}) = (x_k + p_k)$ if the merit function at that point $\phi(x_k + p_k)$ is significantly less than the merit function at the current point $\phi(x)$.

If this relationship between the values of the merit function does not hold, then the trust region radius Δ_k is rejected and reduced to a smaller value. It may be the case that there is no feasible solution to the trust region subproblem. In this circumstance, the linearised constraints (4.43b) are not satisfied at every step and are simply improved with the hope they are satisfied when the trust region constraint allows. This can be achieved via a filter, penalty or relaxation method (see Nocedal and Wright [120]).

4.11 The Method of Moving Asymptotes

MMA was developed by Svanberg in 1987 [167]. It is a method developed specifically for structural optimization and started off as a somewhat heuristic method. Since then globally convergent methods [169] have been implemented but these can be very slow.

The idea behind MMA is to approximate the objective and constraints by functions for which the minimum can be found efficiently. These functions are chosen to be separable and convex. They arise from a Taylor's series expansion in a shifted and inverted variable.

Given the objective function or a constraint $F(x)$. the approximating functions are given by

$$F(x) \approx F(x^0) + \sum_{i=1}^{n} \left(\frac{r_i}{U_i - x_i} + \frac{s_i}{x_i - L_i} \right) \tag{4.44}$$

where r_i and s_i are defined as

$$\text{if } \tfrac{\partial F}{\partial x_i} > 0 \text{ then } r_i = (U_i - x_i^0)^2 \tfrac{\partial F}{\partial x_i}(x^0) \text{ and } s_i = 0$$

$$\text{if } \tfrac{\partial F}{\partial x_i} < 0 \text{ then } s_i = -(x_i^0 - L_i)^2 \tfrac{\partial F}{\partial x_i}(x^0) \text{ and } r_i = 0$$

The variables U_i and L_i are asymptotes for the convex approximating functions, which move dependent on previous iterations (hence the name MMA). The asymptotes are given by the relations

$$L^{(k)} - x^{(k)} = \gamma^{(k)}(L^{(k-1)} - x^{(k-1)})$$
$$U^{(k)} - x^{(k)} = \gamma^{(k)}(U^{(k-1)} - x^{(k-1)})$$

where $\gamma^{(k)}$ is a scalar defined by

$$\gamma^{(k)} = 1.2 \quad \text{if} \quad (x^{(k)} - x^{(k-1)})(x^{(k-1)} - x^{(k-2)}) > 0$$
$$\gamma^{(k)} = 0.7 \quad \text{if} \quad (x^{(k)} - x^{(k-1)})(x^{(k-1)} - x^{(k-2)}) < 0$$
$$\gamma^{(k)} = 1.0 \quad \text{if} \quad (x^{(k)} - x^{(k-1)})(x^{(k-1)} - x^{(k-2)}) = 0.$$

Thus the asymptotes are moved away from the current iteration point if the two previous iterations moved in the same direction. Similarly the asymptotes are moved towards the current iteration point if the two previous iterations moved in opposite directions and they remain in place if in the last two iterations the point x has not moved.

(a) MMA approximations

(b) MMA approximations with narrower asymptotes
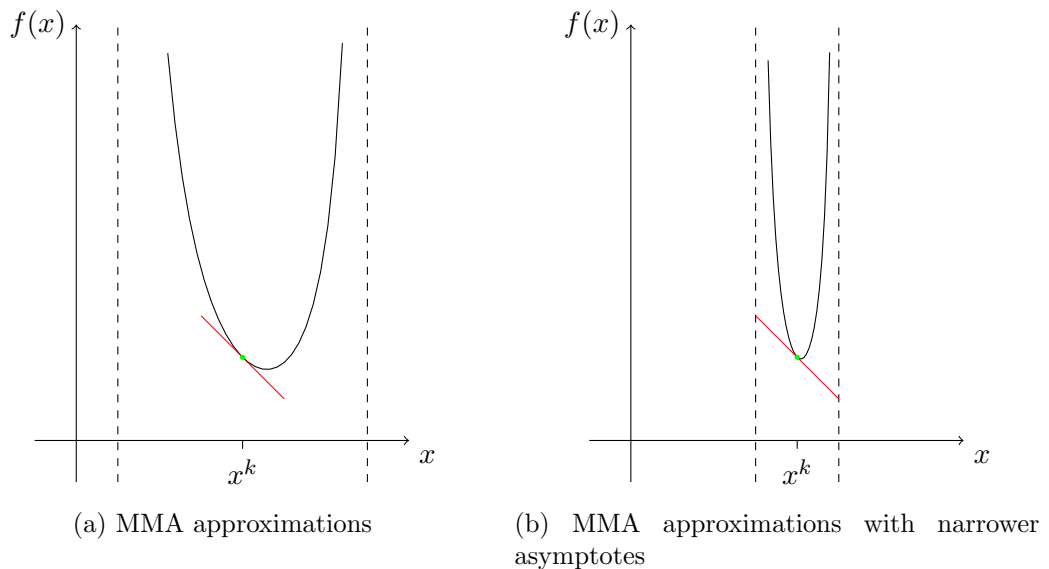
Figure 4-1: MMA approximating functions

The convex approximations to the objective function and constraints are brought together to form the approximating subproblem. These subproblems are separable

and convex, and can be solved using an interior point method. The solution of this subproblem is used as the starting point for the subsequent problem.

If, however, the solution of the subproblem becomes infeasible for the underlying optimization problem, or indeed the value of the objective function is increased, then the corresponding asymptotes are approximating the appropriate function are constricted. This has the affect of making the approximating function more convex, and hence limiting the distance any variables can move along that direction.

Iterating this process of forming and solving subproblems occurs until a KKT point is reached. MMA has become the de facto standard optimization method to use when solving topology optimization problems [19].

4.12   Summary

Optimization methods considered here can be spilt into 2 categories: discrete and continuous. Both of these perspectives have their advantages and will be returned to in subsequent chapters. Discrete optimization techniques can guarantee to find global optima but suffer from the curse of dimensionality. Continuous optimization techniques can avoid the curse of dimensionality but generally converge to local minima with no proof of global optimality.

Detailed knowledge of the theory behind the different optimization methods is needed in order to assess how they apply to structural optimization problems. Continuous optimization techniques such as SQP and MMA will be investigated when the SIMP approach is used in Chapter 5, whereas knowledge of the simplex method and discrete optimization techniques will be used when discussing the ESO method in Chapter 7.

Minimisation of compliance subject to maximum volume

This chapter is concerned with the formulation of structural optimization as an mathematical programming problem that can be solved efficiently. To avoid the curse of dimensionality we immediately relax the binary constraint on the density variables and consider those which vary continuously. Sections 5.1 and 5.2 formulate the problem in the SIMP approach. Section 5.3 discusses appropriate optimization methods to solve the mathematical programming problem. Section 5.4 investigates the possibility of including the state equations directly in the optimization formulation. Section 5.5 introduces filters in order to regularise the problem and make it well posed. Finally Section 5.6 shows the latest results in solving this particular structural optimization problem.

5.1   Convex problem

Suppose we wish to solve the following problem

$$\min_{x} f^T u(x) \tag{5.1a}$$

$$\text{subject to} \quad K(x)u(x) = f \tag{5.1b}$$

$$e^T x \leq V_{\max} \tag{5.1c}$$

$$0 \leq x \leq 1 \tag{5.1d}$$

given that

$$K(x) = \sum_i x_i K_i \tag{5.2}$$
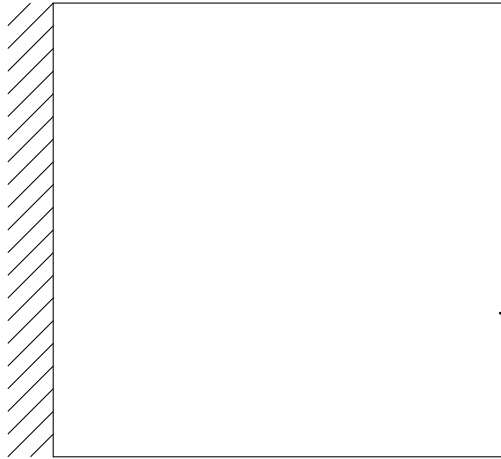
Figure 5-1: Design domain of a short cantilevered beam. The domain is a square that is fixed completely on the left hand side with a unit load applied vertically downwards in the middle of the right hand side of the domain.

where K_i is the element stiffness matrix associated with the variable i, $u(x)$ a vector of displacements, f a vector of applied loads and $V_{\max}$ a scalar defining a volume constraint.

Svanberg [166] showed that the problem (5.1) is convex by considering the Hessian of its Lagrangian and showing that it is positive definite. This means that this problem would be easily solved by most continuous optimization algorithms. For instance, if we consider a short cantilevered beam as shown in Figure 5-1 then we can find the solution to (5.1) which we show in Figure 5-2.

This solution is not desirable as in many cases the values of x_i that are not 0 or 1 do not have any physical meaning. In the Variable Thickness Sheet (VTS) approach where the variables x correspond to the thickness of a planar element then this approach is adequate. When the solution of this problem is to be used to design a structure where at any point we can state whether there is material there or not, we need to introduce a scheme to force the solution to be $x \in \{0, 1\}$.

5.2   Penalised problem

In order to force the solution of (5.1) to be either $x_i = 0$ or $x_i = 1$ for all $i = 1, \ldots, n$ we have to introduce what is known as a penalty function. Recall the construction of the stiffness matrix

$$K = \sum_i x_i K_i \tag{5.3}$$

Figure 5-2: Solution of convex problem on a short cantilevered beam domain. The density of material is plotted with black colour denoting the presence of material with density $x_i = 1$ and white colours the representing the absence of material, i.e. density $x_i = 0$. The colour scale is linear with the density of the material. This was solved using MMA on a mesh of 1600×1600 elements.

where K_i is the element stiffness matrix corresponding to the variable i. When we introduce the penalty function, this equation becomes

$$K = \sum_i \Psi(x_i) K_i \tag{5.4}$$

where Ψ is the penalty function. Note that if this penalty function is nonlinear then the problem (5.1) becomes nonconvex.

The penalty function Ψ is chosen so that it has a number of properties, namely

- Ψ is smooth

- Ψ is monotone

- $\Psi(0) = 0$ and $\Psi(1) = 1$

- $\Psi(x) \leq x$ for all $x \in [0, 1]$.

The penalty function is chosen to be smooth so as to retain the smoothness of the underlying problem. This allows us to use continuous optimization techniques to solve the problem. We want the penalty function Ψ to be monotone so as to avoid introducing extra local minima into the problem. $\Psi(0) = 0$ and $\Psi(1) = 1$ mean that
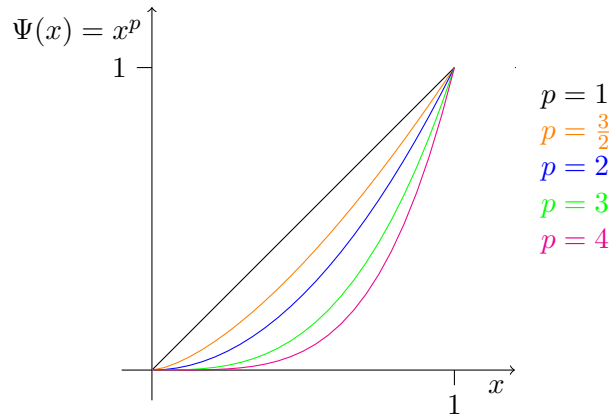
Figure 5-3: Power law penalty functions $\Psi(x) = x^p$ for various values of p in the SIMP method

the stiffness of elements at the points we desire correspond to the physical values that they should have.

The last point is where the penalisation occurs. This property states that the stiffness which we give to an element with an intermediate density is no greater than the physical value that it should have. Put another way, this states that intermediate density elements will provide lower stiffness to the structure than in the non-penalised case. Note that the convex problem is equivalent to choosing the identity as the penalty function.

This discourages elements of intermediate density from appearing in the solution of the optimization problem. To see this, consider the contribution of an element i with density $x_i = 0.5$ in the case $\Psi(x) = x^3$. Then $\Psi(x_i) = (\frac{1}{2})^3 = \frac{1}{8} = \frac{x_i}{4}$. Hence the element x_i is contributing only one quarter of the stiffness it would have in the non-penalised case, making it use up proportionally more volume for its stiffness.

Solid Isotropic Material with Penalisation (SIMP) is the name given to using a power law as the penalty function, i.e.

$$\Psi(x) = x^p, \qquad p \geq 1. \tag{5.5}$$

This will satisfy all the required conditions for the penalty function (see Figure 5-3).

So the SIMP problem of finding a structure with minimum compliance for a given maximum volume looks as follows.

$$\min_x f^T u(x) \tag{5.6a}$$

$$\text{subject to} \quad K(x)u(x) = f \tag{5.6b}$$

$$e^T x \le V_{\max} \tag{5.6c}$$

$$0 \le x \le 1 \tag{5.6d}$$

$$K(x) = \sum_i x_i^p K_i \tag{5.6e}$$

where p is the given penalty parameter.

5.3 Choice of optimization algorithm

We must select an appropriate constrained optimization method in order to solve the SIMP problem (5.6). Let us note some properties of (5.6).

1. This is a nonlinear optimization problem as the objective function, compliance, is nonlinear in the variables x.

2. The equilibrium equations are nonlinear in x, so we have nonlinear constraints.

3. The box constraints and the volume constraints are both linear in x.

4. If $p > 1$ then (5.6) is nonconvex.

5. If $x \in \mathbb{R}^n$ then we would like to be able to cope with n large, say $n = \mathcal{O}(10^6)$.

5.3.1 Derivative Free Methods

A commonly used derivative free method in optimization is the Simplex method for linear programming. This is not suitable for solving (5.6) as, by definition, it is designed for linear problems. Nonlinear programming simplex methods such as the Nelder-Mead Simplex Method are also inappropriate as they may converge to a non-stationary point [111]. More detrimental, however, is the curse of dimensionality which will affect these methods, in that they need $n+1$ function evaluations just to define the initial simplex.

Stochastic Optimization Methods

Stochastic, or evolutionary, methods for optimization have become increasingly popular with engineers over recent years. Along with the more common genetic algorithms and simulated annealing, biologically and physically inspired algorithms have been

proposed for solving constrained optimization problems. These include ant colony optimization, artificial immune systems, charged system search, cuckoo search, firefly algorithm, intelligent water drops and particle swarm optimization, to name but a few.

These methods have a pool of candidate solutions and some measure of the solution's fitness or objective function value. They then follow a set of rules to remove the worst performing candidate solutions from the pool and to create new ones either stochastically or by a defined combination of the best solutions. This evolutionary behaviour is repeated until the pool of candidate solutions cluster around the optimal solution, although this convergence is not guaranteed.

These methods are not going to be viable for solving the problem (5.6) because of the number of variables which we wish to consider. The box constraints (5.6d) mean that our feasible region is contained in the hypercube $[0, 1]^n$ where n is the number of variables in the problem. Hence, to have enough candidate solutions in an initial pool to be in each corner of this hypercube we need 2^n initial solutions. Say, for example, we had $n = 100$, a very modest number of variables. Then we would need $2^{100} > 10^{30}$ candidate solutions just to have one on each vertex. Each one of these candidate solutions would require function and constraint evaluations and so we can quickly see that these methods are not viable for high-dimensional problems such as (5.6).

A comprehensive comparison of stochastic methods for topology optimization with gradient based optimization was carried out by Sigmund, 2011 [153]. They found that when applied to these classes of problems, stochastic optimization methods require many orders of magnitude more function evaluations than derivative based methods and have not been shown to find solutions with improved objective functions.

5.3.2 Derivative based methods

Penalty and Augmented Lagrangian methods

In a penalty function method the idea is to move the constraints of the problem into the objective function and to penalise these terms so that the solution of this new unconstrained problem corresponds to the solution of the constrained problem. For instance, if we recall the general optimization problem

$$\min_x f(x) \tag{5.7a}$$

$$\text{subject to} \quad c_i(x) = 0 \qquad i \in \mathcal{E} \tag{5.7b}$$

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{5.7c}$$

we can define a quadratic penalty function as follows

$$Q(x, \mu) := f(x) + \tfrac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x) + \tfrac{\mu}{2} \sum_{i \in \mathcal{I}} (\min\{c_i(x), 0\})^2. \tag{5.8}$$

The parameter $\mu > 0$ is known as the *penalty* parameter. We can see that if μ is suitably large, then the minimizer of $Q(x, \mu)$ will require the final two terms in (5.8) to be 0, and hence the constraints of (5.7) to be satisfied. Typically, the unconstrained problem (5.8) will be solved repeatedly for a increasing sequence of μ_k until the solution satisfies the constraints.

One can see from the final term in (5.8) that due to the inequality constraints, the quadratic penalty function $Q(x, \mu)$ may be nonsmooth. Due to the box constraints (5.6d) and the volume constraint we would be introducing $2n + 1$ nonsmooth terms into the objective function which may hamper the performance of the solver for the unconstrained problem.

The case of an equality constrained optimization problem, where $\mathcal{I} = \emptyset$, the augmented Lagrangian function is defined as follows

$$\mathcal{L}(x, \lambda, \mu) := f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \tfrac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x). \tag{5.9}$$

Here the λ are an estimate of the Lagrange multipliers for the equality constrained problem. We can see that the augmented Lagrangian is simply the Lagrangian function (4.4) plus a quadratic term in the constraints. It is also an extension of the equality constrained penalty method in (5.8) by adding in the terms with Lagrange multipliers.

In order to use the augmented Lagrangian approach for a problem with inequality constraints we must add slack variables s_i so as to turn these into equality constraints in the following manner,

$$c_i(x) - s_i = 0, \qquad s_i \geq 0, \qquad \forall \, i \in \mathcal{I}. \tag{5.10}$$

If we include the slack variables within our notation x, we can then solve a bound-constrained problem of the form

$$\min_x \quad \mathcal{L}(x, \lambda, \mu) \tag{5.11a}$$

$$\text{subject to} \quad x_{\min} \leq x \leq x_{\max}. \tag{5.11b}$$

This can be solved by a gradient projection method. Practical Augmented Lagrangian methods generally converge only linearly [22] and efficient implementations require

partial separability of the problem, something that problem (5.6) does not possess.

Sequential Quadratic Programming

"[SQP is] probably the most powerful, highly regarded method for solving smooth nonconvex, nonlinear optimization problems involving nonlinear constraints", Conn et Al. 2000 [38].

Sequential Quadratic Programming methods appear to be a good method to try to solve the optimization problem (5.6). SQP methods have been outlined in Section 4.10 but we shall give a brief recap here.

Given a nonlinear programming problem such as (5.7), at an iterate denoted by the subscript k, the constraints are linearised and a quadratic approximation to the objective function is formed. This gives a quadratic subproblem like (4.40) where

$$\min_{p} \quad f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \qquad (5.12a)$$

$$\text{subject to} \quad a_{ki}^T p + c_{ki} = 0 \qquad \forall\, i \in \mathcal{E} \qquad (5.12b)$$

$$a_{ki}^T p + c_{ki} \geq 0 \qquad \forall\, i \in \mathcal{I} \qquad (5.12c)$$

This problem will either be solved to get a search direction with which to perform a line search for a given merit function, or solved with an additional trust region constraint to limit the step size and ensure a decrease in the merit function.

The solution to this problem is then used as the starting point for another linearisation and QP solve until either a KKT point is reached or the method breaks down.

In the topology optimization literature there is relatively little written about the use of SQP as the optimization method. One author has noted that "the application of sequential quadratic programming methods (SQP) ... is known as being not successful due to the lack of convexity of the resulting optimization problem with respect to the variable $[\rho]$", Maar, Schultz 2000 [104]. However, we wish to test this with modern implementations of SQP.

5.4   Simultaneous Analysis and Design (SAND)

In this section we consider solving the state equations by simply including them as constraints in the optimization formulation. This is known as Simultaneous Analysis and Design (SAND). In order to make the notation clearer, new notation will be introduced so that

$$x = \begin{bmatrix} \rho \\ u \end{bmatrix} \tag{5.13}$$

where $\rho \in \mathbb{R}^{n_\rho}$ represents the density of material in an element and $u \in \mathbb{R}^{n_u}$ represents the displacements of the nodes of the finite-element system. Note that $n_u = \mathcal{O}(n_\rho)$.

In this notation, the typical Nested Analysis and Design (NAND) formulation of the problem is written as follows:

$$\min_{\rho} f^T K^{-1}(\rho) f \tag{5.14a}$$

$$\text{subject to} \quad e^T \rho \leq V_{\text{frac}} \tag{5.14b}$$

$$0 \leq \rho \leq 1 \tag{5.14c}$$

As $\rho \in \mathbb{R}^{n_\rho}$ then we have n_ρ variables, 1 linear inequality constraint and n_ρ box constraints. The objective function is nonlinear.

The typical SAND formulation of the problem is similarly written as follows:

$$\min_{\rho, u} f^T u \tag{5.15a}$$

$$\text{subject to} \quad K(\rho) u = f \tag{5.15b}$$

$$e^T \rho \leq V_{\text{frac}} \tag{5.15c}$$

$$0 \leq \rho \leq 1 \tag{5.15d}$$

If the problem considered is in N-dimensional space and $\rho \in \mathbb{R}^{n_\rho}$ then there are $n_\rho + \mathcal{O}(Nn_\rho)$ variables, $\mathcal{O}(Nn_\rho)$ equality constraints (nonlinear in ρ but linear in u), 1 linear inequality constraint and n_ρ box constraints.

Compare these with the NAND formulation and it can be seen that in the SAND formulation there are an extra $\mathcal{O}(Nn_\rho)$ variables and an extra $\mathcal{O}(Nn_\rho)$ nonlinear equality constraints. However the objective reduces from nonlinear in the NAND formulation to linear in the SAND formulation.

This added complexity could be offset by the fact that the solution path is not restricted to the smaller manifold to which the NAND solution path is restricted. The thought is that the SAND method could then reach the solution faster than the NAND method as it is less restricted, or indeed it could find a better local optimum. This is tested subsequently in Section 5.4.1.
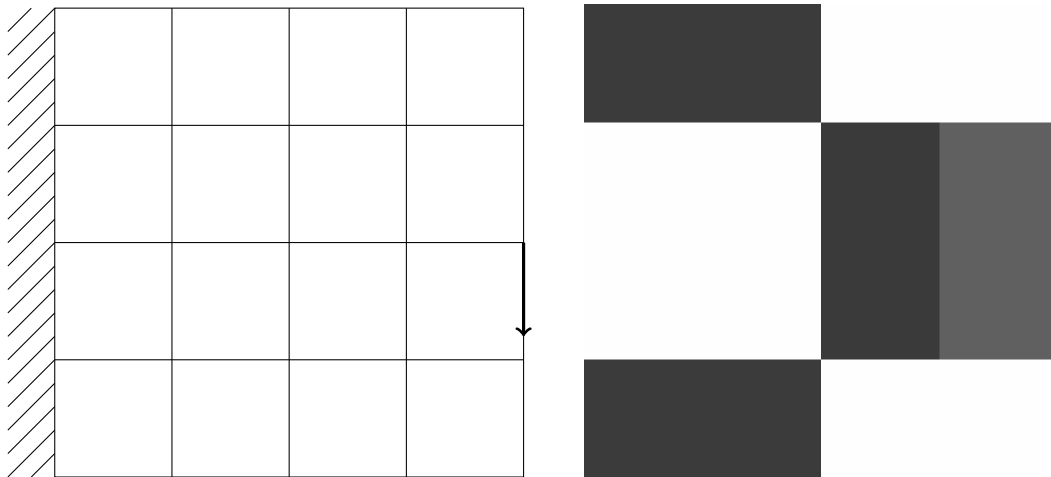
(a) Design domain of cantilevered beam and discretisation

(b) SAND solution of a cantilevered beam using S2QP

Figure 5-4: Design domain and solution using S2QP of a SAND approach to cantilevered beam problem

5.4.1 SQP tests

The SAND approach has been implemented using SQP solvers in order to test its effectiveness. The solvers used were S2QP [60] and SNOPT [56]. Limited results are shown in Figures 5-4 and 5-5. The problem considered in Figure 5-4 has $V_{\text{frac}} = 0.5$ and $p = 3$ and was found using S2QP. Note immediately the atrocious coarseness of the mesh. 4×4 elements is so small that this problem could potentially be solved by hand.

The problem considered in Figure 5-5 also has $V_{\text{frac}} = 0.5$ and $p = 3$ but was solved using SNOPT. Note that this was able to be solved on a mesh of size 10×10 which is still very coarse. An interesting point about this solution is the lack of symmetry. The solution is a verified local minima of the problem and also a verified local minima for the NAND formulation. The symmetry of the problem is not enforced at any stage as the equilibrium constraints only need to be satisfied at the solution. This freedom has allowed the SAND approach to find an asymmetric solution, something which the NAND approach would not produce.

The two figures 5-4 and 5-5 are actually very *atypical* of the results seen from the SAND approach. Typically the methods will fail to converge and these results shown were the product of hard-fought parameter testing and luck. Usually the optimization method claimed that the objective was unbounded below, when the equilibrium constraints were not satisfied. In the next section the reason for the SQP methods
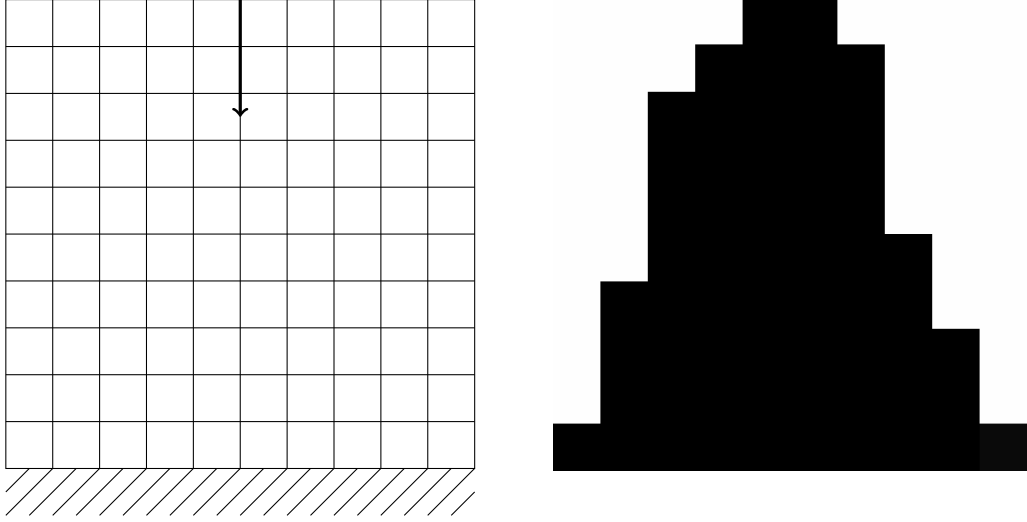
(a) Design domain of centrally loaded column and discretisation

(b) SAND solution of a centrally loaded column using SNOPT

Figure 5-5: Design domain and solution using SNOPT of a SAND approach to centrally loaded column problem. Note the lack of symmetry in the computed local minima suggesting that it is not globally optimal.

returning an unbounded infeasible solution will be investigated.

5.4.2   Constraint qualifications

If the constraints of the SAND formulation (5.15) are ordered so that

$$K(\rho)u - f = 0 \qquad \therefore \mathcal{E} = \{1, \dots, n_u\}$$
$$V_{\text{frac}} - e^T \rho \geq = 0 \qquad \therefore \mathcal{I} = \{n_u + 1\}$$

then for $i \in \{1, \dots, n_u\}$ the constraints are given by

$$c_i(x) = (K(\rho))_i u - f_i \tag{5.16}$$

where $(K(\rho))_i$ is the i-th row of the matrix $K(\rho)$.

The gradient of one of these constraints can be computed as follows.

$$\nabla c_i(x) = \begin{bmatrix} \frac{\partial (K(\rho))_i}{\partial \rho_1} u \\ \vdots \\ \frac{\partial (K(\rho))_i}{\partial \rho_{n_\rho}} u \\ (K(\rho))_i^T \end{bmatrix} \qquad i = \{1, \dots, n_u\} \tag{5.17}$$

Now assume there exists a node in the finite element mesh such that all the surrounding $\rho_i = 0$. Let i' and i'' be the indices corresponding to said node (in $3D$ there is also a third, i''', say). Then

$$\frac{\partial (K(\rho))i'}{\partial \rho_j} = \begin{cases} 0 & \text{if element } j \text{ not connected to node } i' \\ p\rho_j^{p-1} \left[K_j \right]_{i'} & \text{if element } j \text{ is connected to node } i' \end{cases}$$

If ρ_j is connected to node i' then $\rho_j = 0$. As $p > 1$ then this implies that

$$\frac{\partial (K(\rho))_{i'}}{\partial \rho_j} = \frac{\partial (K(\rho))_{i''}}{\partial \rho_j} = 0 \qquad \text{for all } j = 1, \ldots, n_\rho.$$

Note also that

$$(K(\rho))_{i'}^T = (K(\rho))_{i''}^T = 0.$$

Hence

$$\nabla c_{i'}(x) = \nabla c_{i''}(x) = 0 \ [= \nabla c_{i'''}(x)]$$

and therefore the MFCQ (see definition 4.3) does not hold.

As the MFCQ does not hold, this implies that the LICQ also does not hold. Convergence results for SQP methods rely heavily on these constraint qualifications and so the problem as written in SAND form is not one which can be solved reliably by SQP methods. The situation when the MFCQ does not hold will appear frequently in topology optimization. If the problem is thought of as finding where holes should be located, then the circumstance when MFCQ does not hold is precisely the situation which is hoped for in the solution.

If the situation occurs where the density of the elements around an applied load is 0 then the displacement of that node can be made arbitrarily negative, without at all effecting the constraint violation. Hence the solution appears unbounded whilst the constraints are not satisfied.

Due to the increased complexity that would be required in order to adapt the solution method to cope with the SAND approach, this thesis has found that SAND is not an effective formulation of the structural optimization problem. Indeed, any adaptation to SAND would be to use a primal feasible method, thus effectively turning SAND into NAND. This is due to the difficulty that the equations of linear elasticity pose as constraints in an optimization problem. It is therefore clear that the equilibrium equations should be removed from the formulation, solved by a dedicated linear algebra routine, and the problem tackled in a NAND approach.
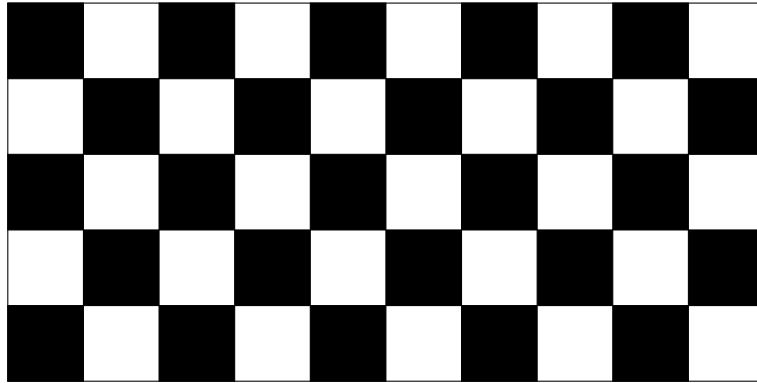
Figure 5-6: Chequerboard pattern of alternating solid and void regions

5.5 Regularisation of the problem by filtering

A problem of minimization of compliance subject to volume is known to be ill-posed (see for example Ambrosio and Buttazzo [9] and Kohn and Strang [93, 94, 95]). That is, improved structures can be found by taking increasingly smaller microstructure. Therefore the problem as stated in general has no solution. In a numerical calculation the solutions of the problem would therefore be dependent on the size of the mesh that is employed. In order to make the problem well-posed we must impose some form of minimum length scale on the problem.

5.5.1 Chequerboards

In an element-based topology optimization approach there may exist solutions that are not desired by engineers. These solutions typically exhibit *chequerboard patterns* as shown in Figure 5-6. In a actual example of minimising the compliance of a cantilevered beam this may manifest itself as in Figure 5-7.

These solutions are numerically driven and result in solutions with material elements connected to the rest of the structure only through corner contacts with other non-void elements. If the underlying mesh has no corner contacts (such as a hexagonal mesh) then these issues do not arise. This has been observed by Talischi et al. [171, 170]. However, automatic mesh generation techniques in general do not exclude corner contacts between elements so it is necessary to have a technique to eradicate chequerboard patterns from any mesh.

If an automatic mesh generation technique was developed to use hexagonal elements in 2D (or possibly rhombic dodecahedra in 3D) then chequerboard patterns would not occur but the solutions would not be mesh independent. Hence strategies to impose a
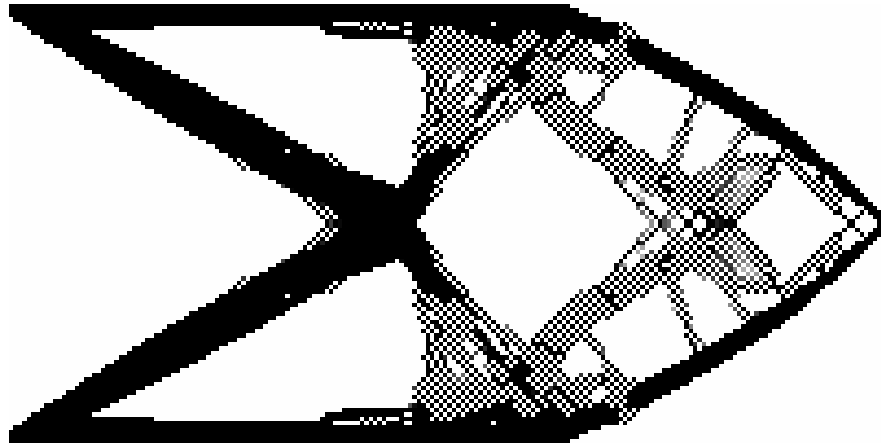
Figure 5-7: Chequerboard pattern appearing in the solution of a cantilevered beam problem.

minimum length scale on the problem would still be necessary.

One possible way of eradicating the chequerboard pattern is to constrain the total perimeter of the structure. This has been considered by, amongst others, Haber 1996 [65], Haber and Jog 1996 [79], Fernandes 1999 [52] and Petersson [127]. However, knowing *a priori* an appropriate value for the length of the parameter is not always possible. This makes it undesirable for us to consider it in this thesis.

5.5.2   Filters

Filtering is the established technique by which chequerboard patterns are eradicated and a minimum length scale applied to the problem. They can be thought of as a local smoothing operator which can be applied to different quantities relating to the optimization.

Bendsøe and Sigmund Filter

The mesh-independency filter [152] works by modifying the element sensitivities as follows:

$$\frac{\hat{\partial c}}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^{n} \hat{H}_f} \sum_{f=1}^{n} x_f \hat{H}_f \frac{\partial c}{\partial x_f} \tag{5.18}$$

The convolution operator (weight factor) $\hat{H}_f$ is written as

$$\hat{H}_f = \max\{r_{\min} - \text{dist}(e, f), 0\} \tag{5.19}$$

where the operator $\text{dist}(e, f)$ is defined as the distance between centre of element e and centre of element f. The convolution operator $\hat{H}_f$ is zero outside the filter area. The convolution operator decays linearly with the distance from element f.

Huang and Xie Filter

The filter [75] is given as follows. Firstly define nodal sensitivity values

$$s_j^\nu = \sum_{i=1}^{\kappa} \omega_{ij} s_i^e \qquad (5.20)$$

where s_i^e is the sensitivity value of element i, κ is the total number of elements connected to node j and ω_{ij} is a weighting given by

$$\omega_{ij} = \frac{1}{\kappa - 1} \left(1 - \frac{r_{ij}}{\sum_{i=1}^{\kappa} r_{ij}} \right) \qquad (5.21)$$

where r_{ij} is the distance from node j to the centroid of element i. The updated sensitivity value s_i is given by the formula

$$s_i = \frac{\sum_{j=1}^{n} w(r_{ij}) s_j^\nu}{\sum_{j=1}^{n} w(r_{ij})} \qquad (5.22)$$

where n is the total number of elements and $w(r_{ij})$ is the weight factor

$$w(r_{ij}) = \max\{0, r_{\min} - r_{ij}\} \qquad (5.23)$$

which is dependent on the variable $r_{\min}$ which defines the filter radius.

Choice of filter

Both the Bendsøe and Sigmund filter and the Huang and Xie filter are applicable to regularise the SIMP problem and are heuristic methods to impose a minimum length scale on the problem. The Sigmund filter [152] is a *density based* filter that is applicable everywhere the density of an element is greater than 0. Essentially it is a low-pass filter from image processing which is used to remove high variations in the gradients within a radius of $r_{\min}$. This is the standard filter which is used in the literature as it performs well with the SIMP approach.

The Huang and Xie filter [75] also removes high variations in the sensitivities of elements within a filter radius of $r_{\min}$ but differs in its implementation. It is designed for use with the BESO method as it can extrapolate sensitivities into regions where the element density is 0. For these reasons, when filtering the SIMP method we will

use the Bendsøe and Sigmund filter, and when we need to extrapolate sensitivities to areas of zero density, such as in Chapter 6, we shall employ the Huang and Xie filter.

5.6   Nested Analysis and Design (NAND)

In the Nested Analysis and Design (NAND) approach to topology optimization, the state equations are removed from the optimization formulation and solved separately by dedicated linear algebra routines. Hence the displacement vector is given as the solution to the equation

$$K(x)u(x) = f \tag{5.24}$$

In the NAND approach we must impose a lower bound on the variables x greater than 0. This is necessary so that the matrix $K(x)$ is positive definite and hence this gives a unique displacement vector $u(x)$. To see this, if we assume that $x_j = 0$ for some element j, then this is equivalent to setting the Young's modulus $E = 0$ for that element. From Definition 3.7 and equation (3.3) we can see that this is equivalent to the Lamé parameter $\mu = 0$. This violates the assumptions of Theorem 3.11 and thus the bilinear form is not coercive. It follows from Theorem 3.12 that the matrix $K(x)$ could be singular and thus we would not have a unique solution to the equilibrium equations (5.24).

Thus to ensure that we have a unique solution of the equilibrium equations, the box constraints on x given in (5.1d) are written as follows.

$$0 < x_{\min} \le x \le 1 \tag{5.25}$$

where $x_{\min}$ is chosen so that a linear solver would recognise that the matrix K is positive definite. Typically we choose $x_{\min}^p = 10^{-9}$, an empiricly found value. Thus the full formulation of the NAND approach to minimisation of compliance subject to a volume constraint is as follows.

$$\min_x f^T K^{-1}(x) f \tag{5.26a}$$

$$\text{subject to} \quad e^T x \le V_{\text{frac}} \tag{5.26b}$$

$$0 \le x \le 1. \tag{5.26c}$$

The remainder of this chapter is dedicated to showing the latest results in solving this mathematical programming problem. To solve this problem we use MMA which is freely available as part of NLopt [81] or directly from the original author Svanberg
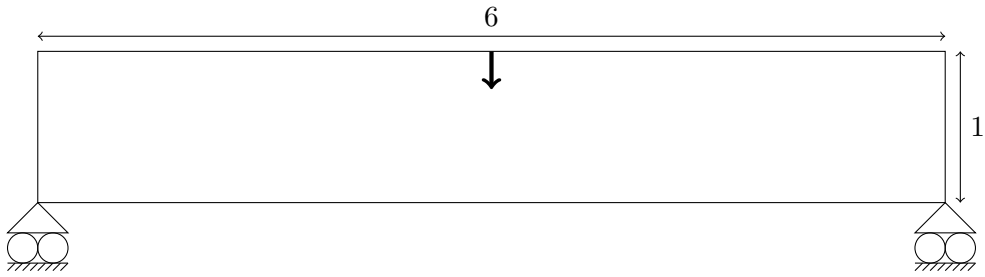
Figure 5-8: Design domain of MBB beam
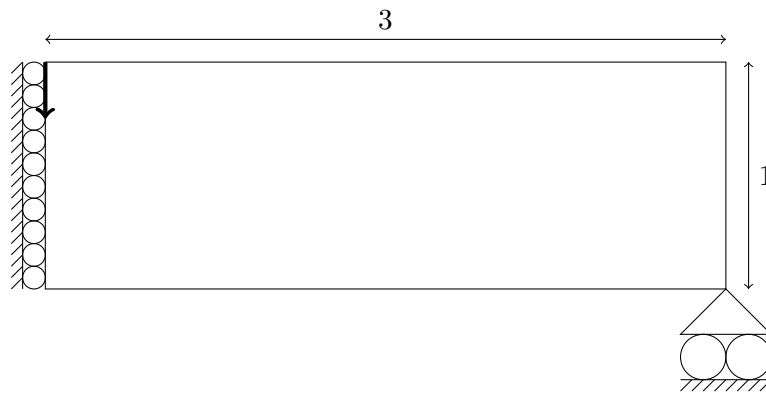
Figure 5-9: Computational domain of MBB beam

[167] supplied free for academic use. Specifically we use the Fortran 77 implementation supplied by Prof. Svanberg.

5.6.1 MBB beam

The MBB beam is named for the German aerospace company Messerschmitt-Bölkow-Blohm which first considered such a structure. The design domain of the MBB beam is given in Figure 5-8. Throughout this example the volume constraint is set to 0.3 of the total volume of the design domain. As usual the material properties are $E = 1$ and $\nu = 0.3$.

MBB Beam without filtering

Figures 5-10 and 5-11 show the resulting structure on the computational and full design domains respectively with no filtering applied. Note immediately the presence of chequerboard patterning in the structure. The very high fidelity of the finite-element discretisation is such that in print these structures may appear grey, whereas in fact the

Figure 5-10: NAND SIMP solution to MBB beam on computational domain without filtering
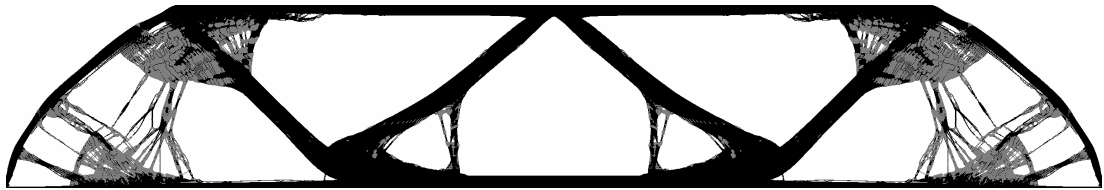
Figure 5-11: NAND SIMP solution to MBB beam on full domain without filtering

Mesh size	1200×400
N	480000
DOFs	962800
$r_{\min}$	0.0
Compliance	387.12
MMA Iterations	91

Table 5.1: Results for NAND SIMP approach to MBB beam without filtering
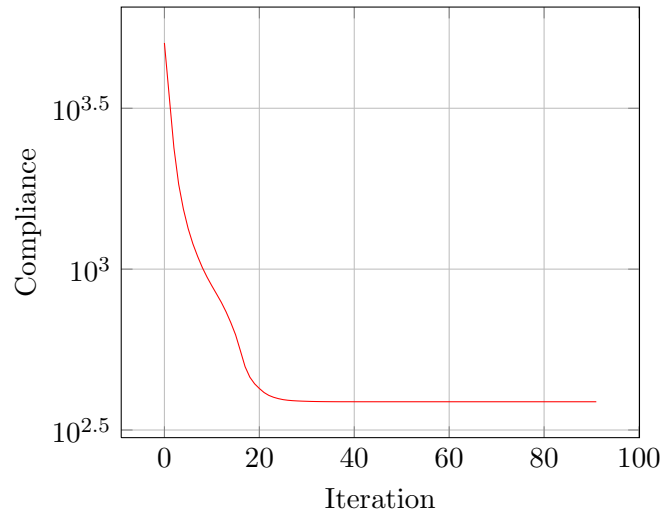
Figure 5-12: Compliance – iterations for NAND SIMP approach to the MBB beam without filtering

variables are very close to the box constraints thanks to the penalisation parameter.

Table 5.1 lists some of the interesting quantities about the optimization process. The values of compliance and the number of MMA iterations will be of interest when comparing to Table 5.2 and Table 5.3. Figure 5-12 shows a plot of compliance against the MMA iteration. Note that after 27 iterations the solution was within 1% of the objective function of the final solution.

The presence of the chequerboard pattern in the computed solution shows the need to apply filtering to the problem.

MBB Beam with filtering

Figures 5-13 and 5-14 show the resulting structure on the computational and full design domains respectively with the low-pass filter applied. Compare these with Figures 5-10 and 5-11 and note immediately the lack of chequerboard patterning in the structures. Details of the optimization process are given in Table 5.2 and a plot of the objective function against MMA iterations is given in Figure 5-15.

From Table 5.2 and Figure 5-15 it should be observed that after 500 iterations the optimization method has not converged. However after 126 iterations the solution was within 1% of the objective function of the solution after 500 iterations. The reason why MMA is failing to converge in this case is the filter passing incorrect derivative values to the optimization routine. It is precisely this feature of the filter which is regularising the problem and stopping the chequerboard patterns emerging early in the

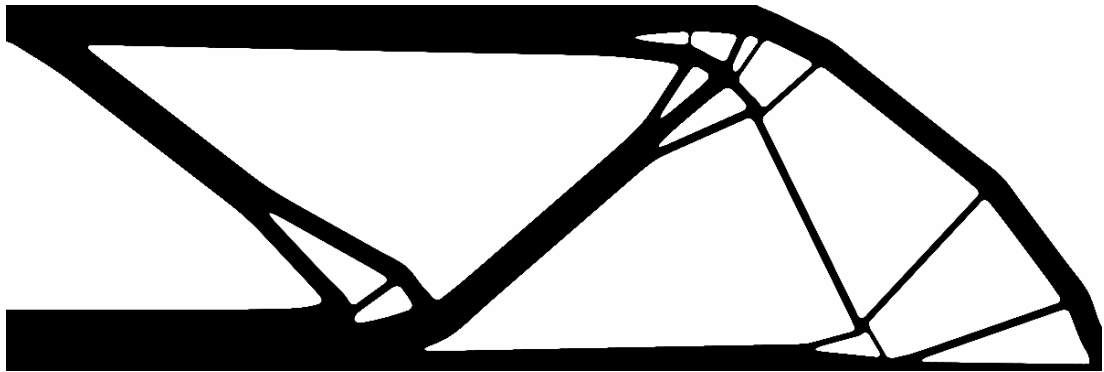Figure 5-13: NAND SIMP solution to MBB beam on computational with filtering
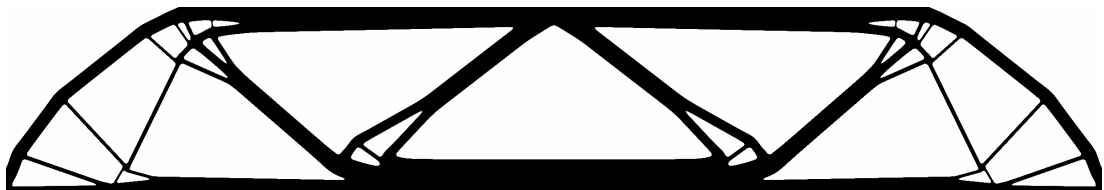
Figure 5-14: NAND SIMP solution to MBB beam on full domain with filtering

Mesh size	1200×400
N	480000
DOFs	962800
$r_{\min}$	7.5
Compliance	287.56
Iterations	500+

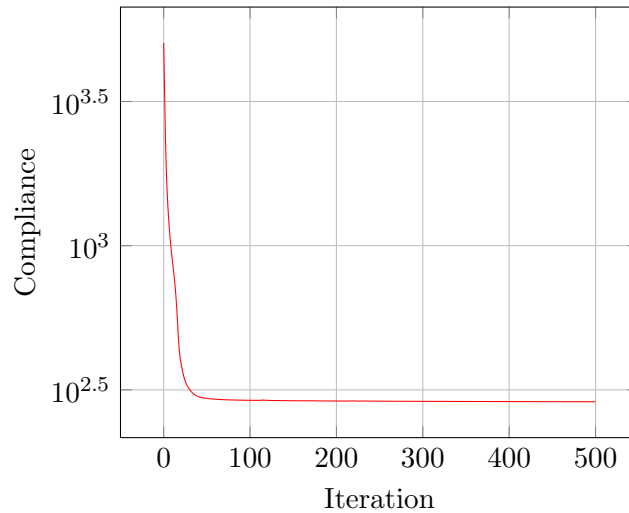Table 5.2: Results for NAND SIMP approach to MBB beam with filtering

Figure 5-15: Compliance – iterations for NAND SIMP approach to the MBB beam with filtering

optimization process, so this is not an unwanted feature.

In order to aid the optimization method to converge, it is necessary to provide it with the correct gradients, and so we choose a scheme to stop applying the filter after a certain period in the optimization process when the solution is close to the optimum.

MBB Beam with cessant filter

As Figures 5-12 and 5-15 have shown, typical objective functions found in the SIMP approach to structural optimization resemble long flat valleys. Hence when the solution is near the base of these valleys it would be advantageous to move in a very accurate search direction. As applying a low-pass filter to the gradient information gives inexact gradients to the optimization method, a scheme to turn off the filter when it can be detected that the solution is near the floor of the valley.

Hence for some tolerance *tol*, we choose to turn off the filter when the objective function at iteration k, denoted $\phi(k)$ satisfies

$$\frac{\phi(k) - \phi(k-1)}{\phi(k)} < tol. \tag{5.27}$$

This technique has been applied to the MBB beam with $tol = 10^{-5}$ and the results are shown in Figures 5-16 to 5-18 and Table 5.3.

From Figures 5-16 and 5-17 it can be seen that the use of the filter has removed the chequerboard patterns that are present in Figures 5-10 and 5-11. The resulting topology
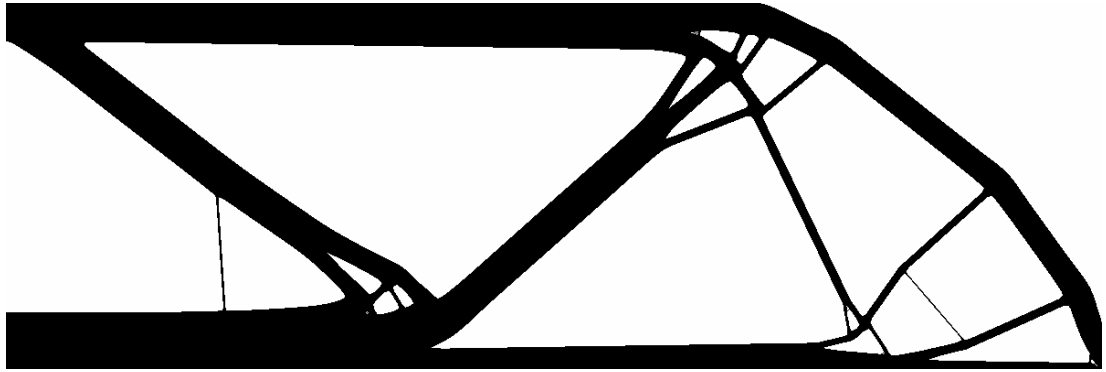
Figure 5-16: NAND SIMP solution to computational domain of MBB beam with cessant filter
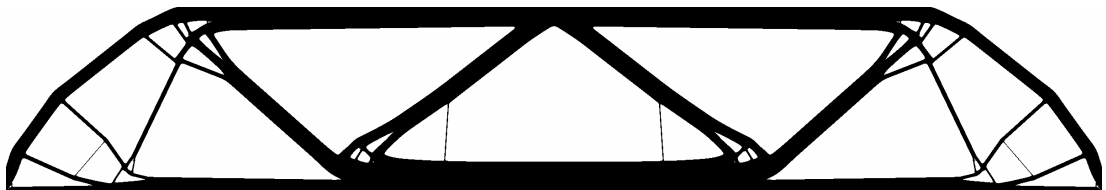
Figure 5-17: NAND SIMP solution to full MBB beam with cessant filter

Mesh size	1200×400
N	480000
DOFs	962800
$r_{\min}$	7.5
Filter *tol*	1×10^{-5}
Compliance	290.08
Iterations	192
Iterations with filtering	95

Table 5.3: Results for NAND SIMP approach applied to MBB Beam with cessant filter
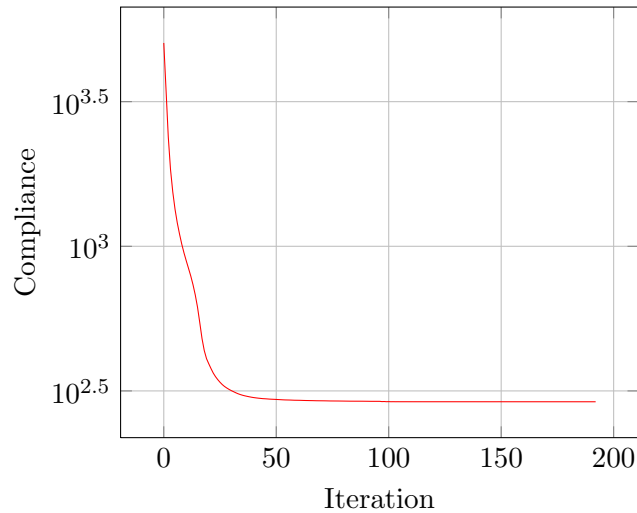
Figure 5-18: Compliance – iterations for NAND SIMP approach to the MBB beam with cessant filter

is very similar to the topology presented in Figure 5-14, the computed solution when the filter is applied constantly. Ceasing the filter has caused the number of iterations to drop markedly and the solution has this time converged to a local minima. This local minima is not quite a low as the solution computed when the filter is applied constantly (compare Tables 5.2 and 5.3) as stopping the filter has not continued to smooth out the fine features present in the solution.

The use of a cessant filter thus retains the chequerboard removing properties of filtering while allowing the optimization method to converge.

5.6.2 Michell Truss
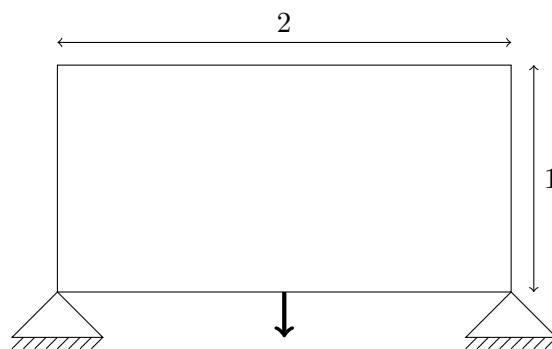
Figure 5-19: Design domain of Michell truss
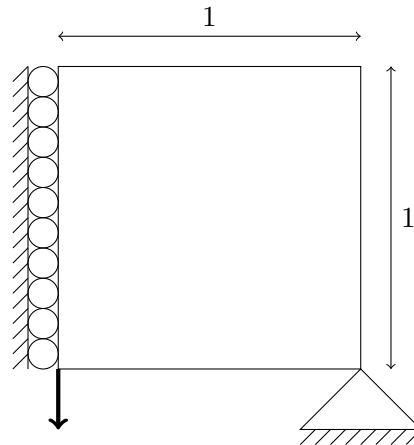
Figure 5-20: Computational design domain of Michell truss
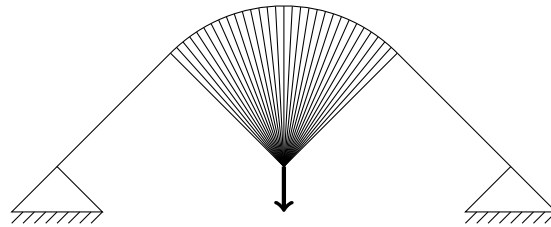
Figure 5-21: Analytic optimum to Michell truss

Figures 5-19 and 5-20 show the full design domain and the computational design domain of the Michell truss [112] respectively. The analytic optimum for this problem is given in Figure 5-21 with the thickness of the bars in the structure dependent on the volume constraint of the problem (Save and Prager 1985 [147]). The analytic optimum has an infinite number of infinitely thin bars (due to the ill-posedness of the problem) which cannot be represented on the finite-element discretisation of the design domain. However it is expected that the same basic shape with a finite number of bars should be present in a computed solution to this problem.

Figure 5-22 shows the NAND SIMP solution using MMA when applied to the Michell truss problem with a cessant filter of radius $7.5h$ where h is the width of an element in the finite-element mesh. Figure 5-23 shows the result of the same problem as Figure 5-22 but with a smaller filter radius of $2.5h$. Note the finer bars present in the case with the smaller filter radius and also how both structures very closely resemble the analytic optimum shown in Figure 5-21. The numerical values of the two optimization processes are given in Table 5.4 and the objective function history of both
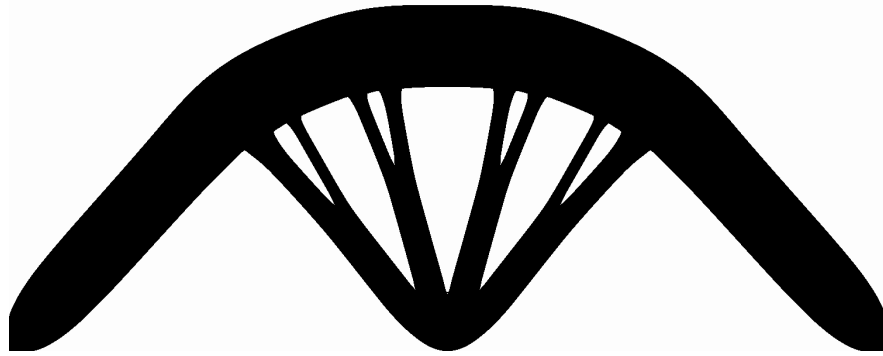
Figure 5-22: NAND SIMP solution to Michell truss problem on a 750×750 mesh with a cessant filter of radius $7.5h$ and $V_{\text{frac}} = 0.3$
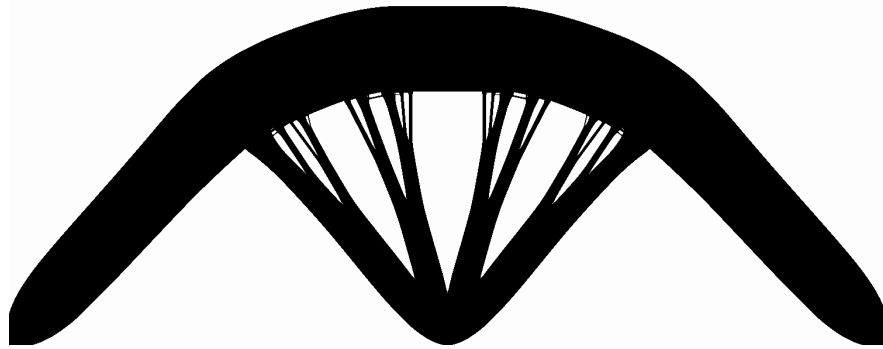
Figure 5-23: NAND SIMP solution to Michell truss problem on a 750×750 mesh with a cessant filter of radius $2.5h$ and $V_{\text{frac}} = 0.3$

examples are shown in Figure 5-24.

Firstly compare the number of iterations taken in the examples with different filter radii. The smaller the filter radii, the longer the optimization method takes to resolve the finer features of the structure. Note also the different objective function values of the two examples. The wider filter radius has provided more of a perturbation to the true gradients and thus has stopped the optimization process from falling into a local minima as early as the problem with the smaller filter radius.

In the convergence plot of objective function against MMA iterations in Figure 5-25, the point at which the filter is turned off is visible for the example with filter radius of 7.5. At this point MMA can step directly towards a local optimum and so the plot shows a marked decease in the objective function at this point.

Mesh size	750×750	750×750
N	562500	562500
DOFs	1127249	1127249
$r_{\min}$	7.5	2.5
Filter *tol*	1×10^{-5}	1×10^{-5}
Compliance	34.573	34.696
Iterations	212	269
Iterations with filtering	193	238

Table 5.4: Results for Michell truss with cessant filter
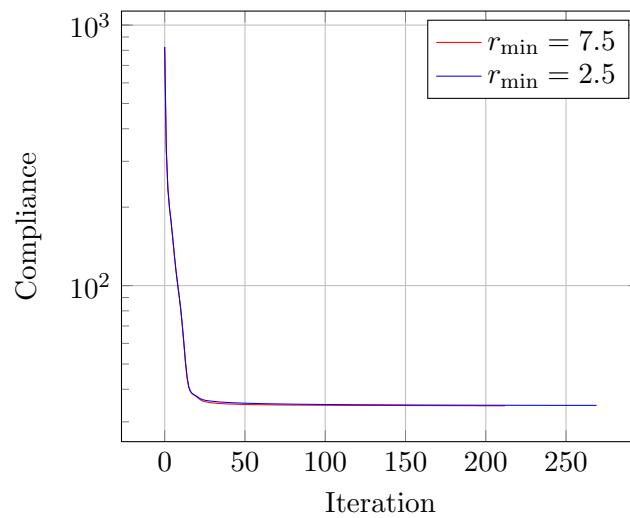
Figure 5-24: Compliance – iterations for NAND SIMP approach to the Michell truss with cessant filters of various radii
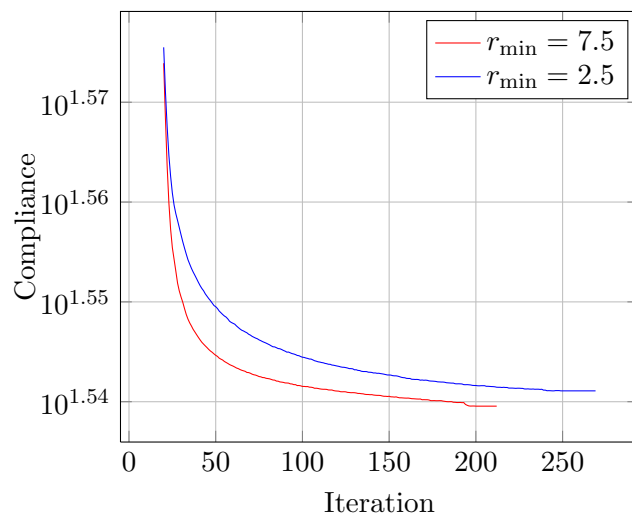
Figure 5-25: Compliance – iterations for NAND SIMP approach to the Michell truss with cessant filters of various radii after 20 iterations

5.6.3 Short cantilevered beam
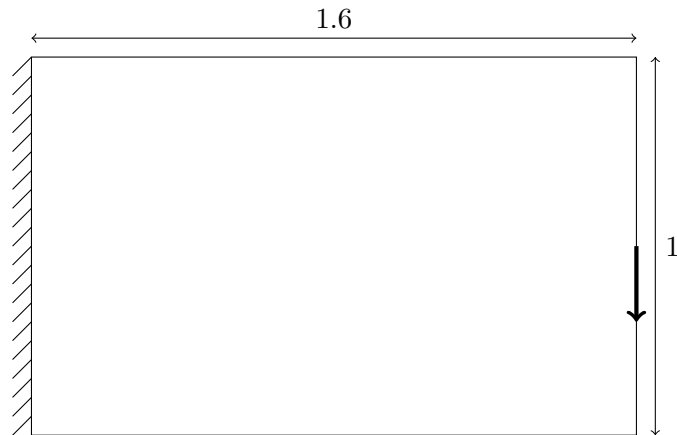
Figure 5-26: Design domain of the short cantilevered beam

The short cantilevered beam is a problem that will be considered again in later chapters so it included here to show the SIMP solution. The design domain is shown in Figure 5-26 and is a rectangle of aspect ratio $8 : 5$ fixed entirely on the left hand side with a unit load applied vertically in the middle of the right hand side. There is a symmetry present in this problem which could be removed to allow for a finer mesh, but leaving it shows that the NAND SIMP approach using MMA retains the inherent symmetry to the minimisation of compliance subject to a volume constraint problem.

The solution found on a mesh of size 1000×625 with a cessant filter of radius $7.5h$ and $V_{\text{frac}} = 0.3$ is shown in Figure 5-27 with the associated numerical values given in Table 5.5. Note the fanning of the bars around the corners of the structure similar to those seen in the Michell truss.

Mesh size	1000×625
N	625000
DOFs	1252000
r_{min}	7.5
Filter *tol*	1×10^{-5}
Compliance	59.881
Iterations	85
Iterations with filtering	63

Table 5.5: Results for short cantilever beam with cessant filter
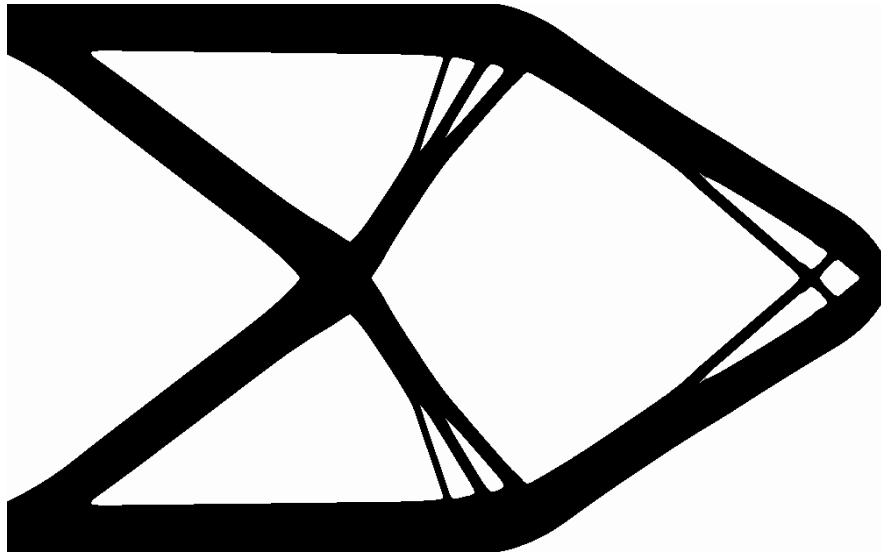
Figure 5-27: NAND SIMP solution to short cantilevered beam problem on a 1000×625 mesh with a cessant filter of radius $7.5h$ and $V_{\mathrm{frac}} = 0.3$
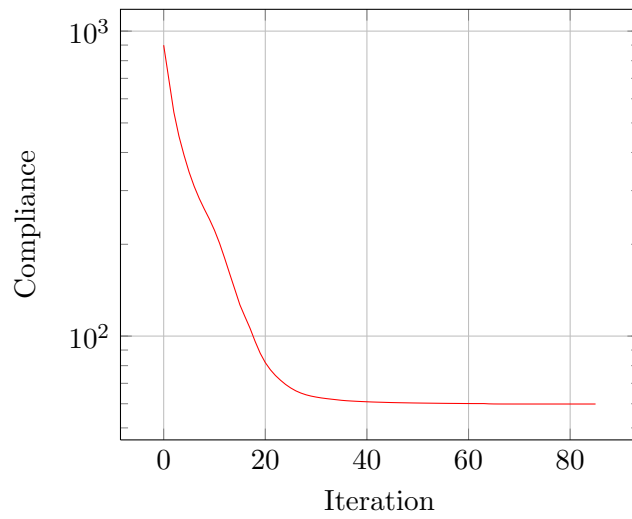
Figure 5-28: Compliance – iterations for NAND SIMP approach to the short cantilevered beam on a 1000×625 mesh with cessant filter of radius $7.5h$ and $V_{\mathrm{frac}} = 0.3$

5.6.4 Centrally loaded column

Here is presented a somewhat trivial optimization problem which is included for comparison with results in later chapters. The design domain is square and a unit load is applied vertically downwards at the centre of the top of the design domain and the base is fixed, as shown in Figure 5-29.
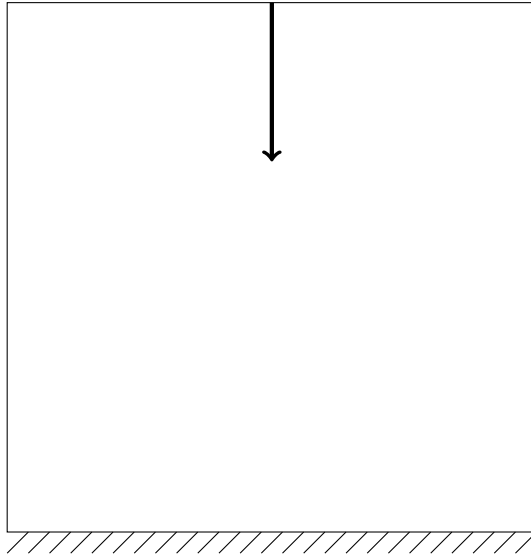
Figure 5-29: Design domain of model column problem. This is a square domain with a unit load acting vertically at the midpoint of the upper boundary of the space.

Mesh size	750×750
N	562500
DOFs	1127250
$r_{\min}$	7.5
Filter *tol*	1×10^{-5}
Compliance	8.2047
Iterations	104
Iterations with filtering	87

Table 5.6: Results for centrally loaded column with cessant filter

The table of results for the centrally loaded column is given in Table 5.6 with the computed solution shown in Figure 5-30. The solution is a simple column which takes the load directly to the base of the design domain, with the thickness of the column dependent on the magnitude of the volume constraint parameter V_{frac}. This solution should be compared with the problems considered later in Section 6.6.3.
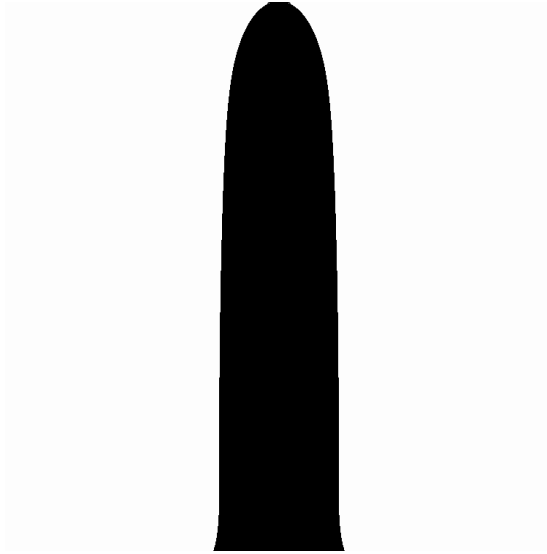
Figure 5-30: NAND SIMP solution to centrally loaded column problem on a 750×750 mesh with a cessant filter of radius $7.5h$ and $V_{\text{frac}} = 0.2$
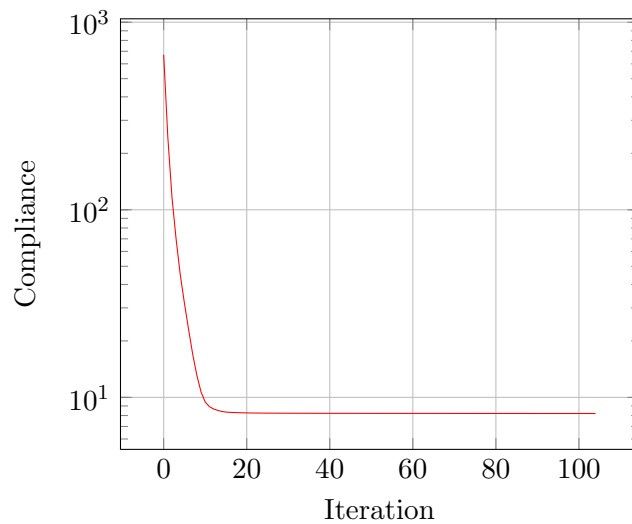
Figure 5-31: Compliance – iterations for NAND SIMP approach to the centrally loaded column on a 750×750 mesh with a cessant filter of radius $7.5h$ and $V_{\text{frac}} = 0.2$

5.7   Summary

This chapter has considered solving the problem of minimisation of compliance subject to a volume constraint by relaxing the binary constraint on the optimization variables and allowing the solution to vary continuously between 0 and 1. The theory of penalising intermediate densities has been reviewed and the SIMP approach has been motivated.

The choice of optimization algorithm to solve the problem has been considered and SQP methods have been used to try and solve the optimization problem in a SAND formulation. These, usually robust methods, generally failed to find a solution to these problems and it has been shown that this is due to constraint qualifications being violated in the SAND approach.

Chequerboard patterns have been observed as the problem as generally stated is ill-posed. Techniques for eradicating the chequerboard patterns have been discussed and the reasons for applying filters to the problem explained.

High fidelity examples of minimisation of compliance problems subject to a volume constraint in a NAND formulation to the SIMP approach using MMA have been presented. The use of filtering in these problems is shown to remove chequerboards but also to stop the optimization method from converging. A technique for turning off the filtering was introduced and shown to be robust and give good solutions without chequerboarding that also converged to local minima.

Buckling Optimization

In this chapter adding a buckling constraint to the standard structural optimization problem is considered. This adds a great deal of complexity and introduces a number of issues that do not arise in the more basic problem considered in Chapter 5. Section 6.1 introduces the buckling constraint and shows how a direct bound on the buckling constraint becomes non-differentiable when there is a coalescing of eigenvalues. Section 6.2 discusses the issues arising with spurious buckling modes. The problem is reformulated in Sections 6.3 to 6.4 and an analytic formula for the derivative of the stress stiffness matrix is presented. In Section 6.5 we then introduce a new method in order to efficiently compute a solution to an optimization problem involving buckling constraints.

6.1 Introduction and formulation

This chapter is motivated by a long standing realisation of a potential shortcoming of structural optimization:

"A process of optimization leads almost inevitably to designs which exhibit the notorious failure characteristics often associated with the buckling of thin elastic shells", Hunt 1971 [77].

In the finite-element setting, the buckling load of a structure is the smallest positive value of λ which solves the eigenvalue problem

$$(K + \lambda K_\sigma)v = 0 \qquad \text{for some} \qquad v \neq 0 \tag{6.1}$$

as described previously in Sections 3.4 and 3.5. In order to prevent the buckling of the structure, the eigenvalue λ must be kept larger than some safety factor. So consider a

bound of the form

$$\lambda > c_s \tag{6.2}$$

where c_s is a constant representing the safety factor. Note that this may never be feasible if c_s were chosen too large, and a problem with this specified constraint would have no solution. Now let us consider $\frac{\partial \lambda}{\partial x_i}$ noting that all the terms in (6.1) depend on x. Differentiating (6.1) gives

$$\frac{\partial K}{\partial x_i}v + K\frac{\partial v}{\partial x_i} + \frac{\partial \lambda}{\partial x_i}K_\sigma v + \lambda\frac{\partial K_\sigma}{\partial x_i}v + \lambda K_\sigma\frac{\partial v}{\partial x_i} = 0 \tag{6.3}$$

by the product rule. Rearranging this gives

$$-\frac{\partial \lambda}{\partial x_i}K_\sigma v = (\frac{\partial K}{\partial x_i} + \lambda\frac{\partial K_\sigma}{\partial x_i})v + (K + \lambda K_\sigma)\frac{\partial v}{\partial x_i}. \tag{6.4}$$

Multiplying on the left by v^T and noting again that K and K_σ are symmetric then the term on the right must vanish by (6.1) and thus

$$\frac{\partial \lambda}{\partial x_i}v^T K_\sigma v = -v^T(\frac{\partial K}{\partial x_i} + \lambda\frac{\partial K_\sigma}{\partial x_i})v. \tag{6.5}$$

At this point many authors make the assumption that the eigenvector v is normalised so that $v^T K_\sigma v = 1$. However as K_σ is not guaranteed to be positive definite this may lead to $v \in \mathbb{C}^n$ and thus increasing the computational complexity of the problem. To avoid this we choose to simply normalise v in a different norm ($v^T K v = 1$ as K is SPD) and keep track of the product $v^T K_\sigma v$ so that

$$\frac{\partial \lambda}{\partial x_i} = -\frac{v^T(\frac{\partial K}{\partial x_i} + \lambda\frac{\partial K_\sigma}{\partial x_i})v}{v^T K_\sigma v}. \tag{6.6}$$

Note also that $v^T K_\sigma v \neq 0$ as this would contradict K being positive definite.

Suppose however that λ is not a simple eigenvalue of (6.1). Then there exists another eigenvector $w \neq \pm v$, say, such that

$$(K + \lambda K_\sigma)w = 0. \tag{6.7}$$

In going from (6.4) to (6.5) it would be equally valid to multiply on the left by w^T to give

$$\frac{\partial \lambda}{\partial x_i} = -\frac{w^T(\frac{\partial K}{\partial x_i} + \lambda\frac{\partial K_\sigma}{\partial x_i})v}{w^T K_\sigma v}. \tag{6.8}$$

Indeed any linear combination of eigenvectors would cause the right hand term in (6.4) to vanish and would give an expression for $\frac{\partial \lambda}{\partial x_i}$. These are clearly different values and

shows that the derivative of the eigenvalue is not well defined when the eigenvalue in question is non-simple.

This is a major issue for continuous optimization using derivative-based methods. These approaches will naturally cause a coalescing of eigenvalues and hence may fail to converge to a solution. Semidefinite programming methods have been developed specifically to deal with such eventualities. A semidefinite matrix constraint on the matrix A has the form

$$A \succeq 0 \tag{6.9}$$

meaning that all the eigenvalues of the matrix A are bounded above 0. We now show how a bound on compliance and a bound on the buckling load of a system can be written as semidefinite matrix constraints.

Theorem 6.1. *Given a SPD matrix A then the symmetric matrix $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$ is positive semidefinite if and only if the* Schur complement $S = C - B^T A^{-1} B$ *is positive semidefinite.*

Proof. The proof is given in Boyd and Vandenberghe [25] appendix A.5.5 by considering the following:

$$\min_{\mathbf{u}} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \min_{\mathbf{u}} \mathbf{u}^T A \mathbf{u} + 2\mathbf{u}^T B \mathbf{v} + \mathbf{v}^T C \mathbf{v}.$$

$\square$

Corollary 6.2. *A constraint on the compliance of the system*

$$f^T u \leq c \tag{6.10}$$

may be written as a semidefinite matrix constraint of the form

$$\begin{bmatrix} K & f \\ f^T & c \end{bmatrix} \succeq 0 \tag{6.11}$$

Proof. In Theorem 6.1, set $A = K$, $B = f$ and $C = c$. Then as K is SPD it says

$$\begin{bmatrix} K & f \\ f^T & c \end{bmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad c - f^T K^{-1} f \succeq 0 \tag{6.12}$$

Using the relation $Ku = f$ and the fact that the matrix on the right hand side is of

dimension 1 we can rewrite this as

$$\begin{bmatrix} K & f \\ f^T & c \end{bmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad c - f^T u \geq 0 \tag{6.13}$$

or

$$\begin{bmatrix} K & f \\ f^T & c \end{bmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad f^T u \leq c \tag{6.14}$$

$\square$

Lemma 6.3 (Kočvara 2002 [90]). *Assume that K is positive definite and let $c_s > 0$. The matrix $[K + c_s K_\sigma]$ is positive semidefinite if and only if all the eigenvalues λ satisfying*

$$(K + \lambda K_\sigma)v = 0 \qquad for\ v \neq 0$$

lie outside of the interval $(0, c_s)$.

Proof. As K is SPD we can take its inverse and rewrite the condition of $[K + c_s K_\sigma]$ being positive semidefinite as

$$c_s^{-1} I + K^{-1} K_\sigma \succeq 0 \tag{6.15}$$

From the original eigenvalue problem (6.1) we have

$$- K^{-1} K_\sigma v = \frac{1}{\lambda} I v \tag{6.16}$$

so the eigenvalues of the matrix $[c_s^{-1} I + K^{-1} K_\sigma]$ are $(c_s^{-1} - \frac{1}{\lambda})$. Thus equation (6.15) holds if and only if $\frac{1}{c_s} - \frac{1}{\lambda} \geq 0$, i.e. either

$$\lambda \geq c_s \qquad \text{or} \qquad \lambda < 0 \tag{6.17}$$

$\square$

In a semidefinite approach to optimization, all the matrix entries of the constraints are effectively treated as variables. Hence if there are $\mathcal{O}(n)$ variables in the original formulation representing the densities of elements then an SDP approach to the problem would be considering $\mathcal{O}(n^2)$ variables. This significantly increases the computational cost of SDP methods in comparison to other methods.

Kocvara [91], and in conjunction with Stingl [92], have applied such methods to topology optimization problems. More recently, along with Bogani [23], they have applied an adapted version of their semidefinite codes to find noninteger solutions to

buckling problems. This made use of a reformulation of a semidefinite constraint using the indefinite Cholesky factorisation of the matrix, and solving a resulting nonlinear programming problem with an adapted version of MMA. With these techniques they were able to solve a non-discrete (convex) problem with 5000 variables in about 35 minutes on a standard PC.

6.2 Spurious Localised Buckling Modes

In this section we discuss the issue that occurs in the process of continuous optimization whereby the buckling load computed by standard means is numerically driven to be substantially lower than the physical load. Firstly we show this by means of a simple example.

6.2.1 Considered problem

Here we define a model problem which we consider in the rest of this section. As shown in Figure 6-1a we have a square design domain. The loading is vertically downwards at the top of the design domain and the base is fixed completely. The design domain is discretised into a mesh of 10×10 elements as shown in Figure 6-1b and the problem is minimisation of compliance subject to a volume constraint of 0.2 of the whole design domain. We solve this problem using the SIMP method with MMA in a nested approach as in Chapter 5.

6.2.2 Definition and eradication strategies

When an element's density is too low, the buckling mode calculated as the smallest positive eigenvalue of (6.1) may not correspond to the physically desired modeshape. The modeshape corresponding to the smallest positive eigenvalue can be seen to be localised in the regions where the elements have low density. In our formulation, low density elements represent areas with little or no material and so we wish the computed buckling mode to be driven by the elements containing material. Tenek and Hagiwara [173], Pedersen [125] and Neves et al. [116] all noted that spurious buckling (or harmonic) modes would be computed in which the buckling is confined to regions where the density of material is less than 10%.

Definition 6.4. *We define a* low density element *to be one where the density is below a threshold value. Here we consider this threshold to be 0.1, similarly to Pedersen [125] and Neves et Al. [118].*
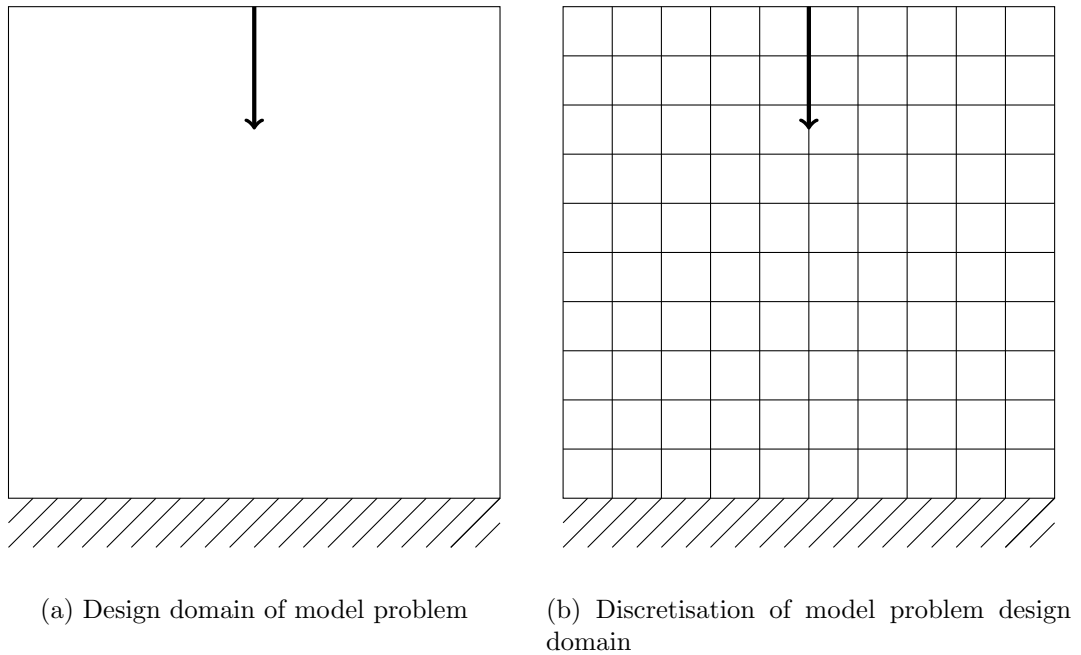
(a) Design domain of model problem

(b) Discretisation of model problem design domain

Figure 6-1: Considered problem in this section to show spurious buckling modes.

Definition 6.5. *A* spurious localised buckling mode *is an eigenvector that is a solution to* (6.1) *such that the displacements corresponding to nodes connected to non low density elements are all zero.*

Spurious localised buckling modes are elucidated in Figures 6-2 to 6-3 from a minimization of compliance optimization subject to a volume constraint *without* a buckling constraint.

Figure 6-3a shows the first occurrence of the spurious buckling modes. The elements in the top corners are first to get to a low value and we can see that in these areas the buckling mode is localised. This is the first time that the element density drops below 0.1, which is the critical value as found by Pedersen [125] and Neves et Al. [118].

Figure 6-3b corresponds to the smallest positive eigenvalue of the full system at the final solution of the optimization problem. The modeshape shown in Figure 6-4a shows the computed buckling mode when the void elements are completely removed from the eigenvalue calculations. Figure 6-4b is the 137th smallest positive eigenvalue of the full system as in Figure 6-3b.

Numerous options to deal with the problem of these spurious eigenvectors have been considered. These include

- Changing the stiffness/stresses associated with void elements

- Remeshing to remove the low density elements
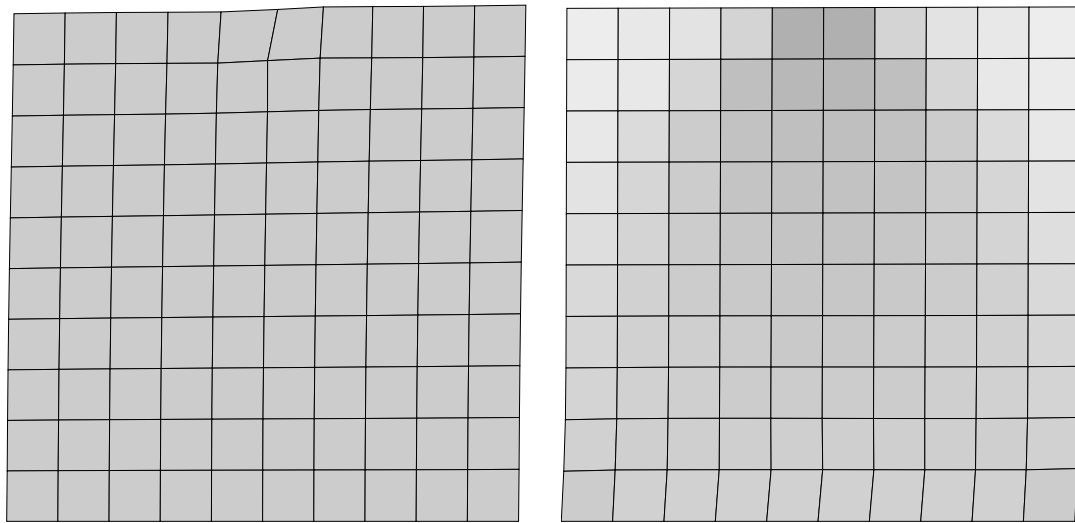
(a) Initial distribution of material and corresponding modeshape

(b) Distribution of material and corresponding modeshape after 1 iteration

Figure 6-2: Initial modeshape and modeshape after one iteration. Note no spurious localised buckling modes are observed.

(a) Distribution of material and corresponding modeshape after 2 iterations. Here we see the spurious buckling mode as the displacements are non-zero only in the top corners where the density is below 0.1.

(b) Final distribution of material and corresponding modeshape after 17 iterations. Here the spurious buckling mode is plain to see as the solid structure as not displaced at all.

Figure 6-3: Spurious localised buckling modes appearing in areas of low density.

(a) Actual modeshape computed when void elements are removed from the formulation

(b) 137th smallest positive eigenvalue of full system. This shows that the desired eigenvector is within the spectrum, but no longer is it the smallest positive eigenvalue.
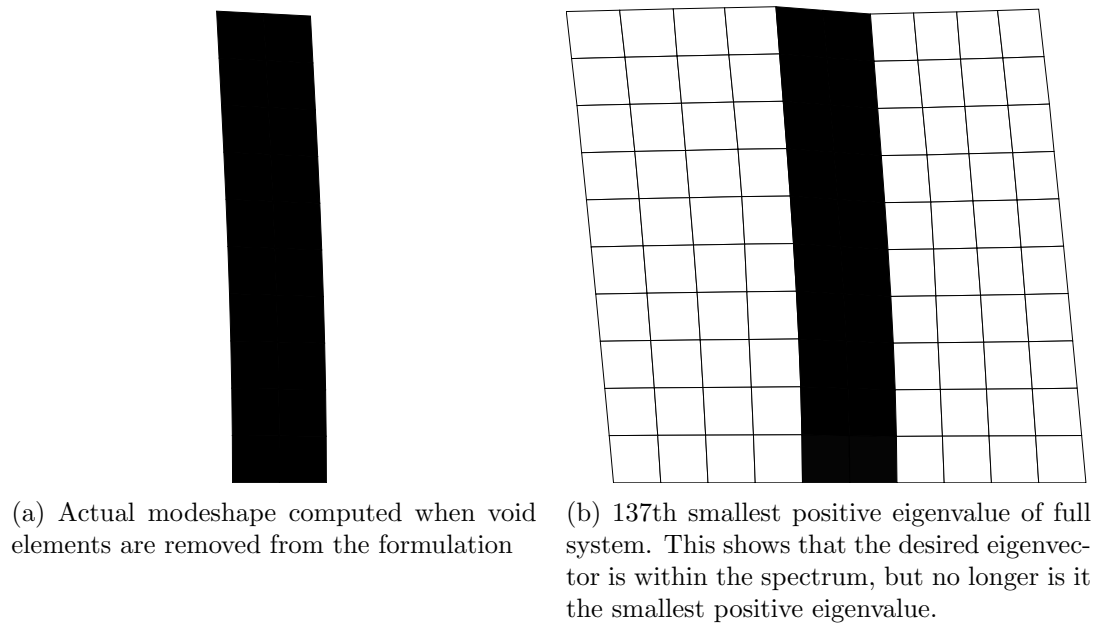
Figure 6-4: Modeshape of the solution in Figure 6-3b which are driven only by the elements containing material.

- More complete eigenvalue analysis

Finding the appropriate eigenpair from the unchanged spectrum that corresponds to the physically appropriate modeshape is a challenging problem. As Figure 6-4b shows the eigenpair may be found, but the eigenvalue seems not to occur at a significant point in the spectrum. That is to say, there is no distinct gap in the spectrum around the eigenvalue of interest and so it would be challenging to automatically detect the appropriate eigenvalue.

Remeshing would also be fraught with complications. Removing elements from the formulation would result in a lack of information about that specific area of the design domain. Doing so would lose all the information about elements with low densities, not just in terms of the buckling behaviour but also in terms of compliance.

Neves et Al. [118] have suggested reducing the stress in the elements with a density lower than 0.1 to an insignificant value of 10^{-15}. This very small value is necessary as they make the assumption that K_σ is SPD. As we are not making the assumption that K_σ is SPD, we have implemented this scheme for the same problem as in Figures 6-2a-6-3b with the difference that we set the stress to be zero.
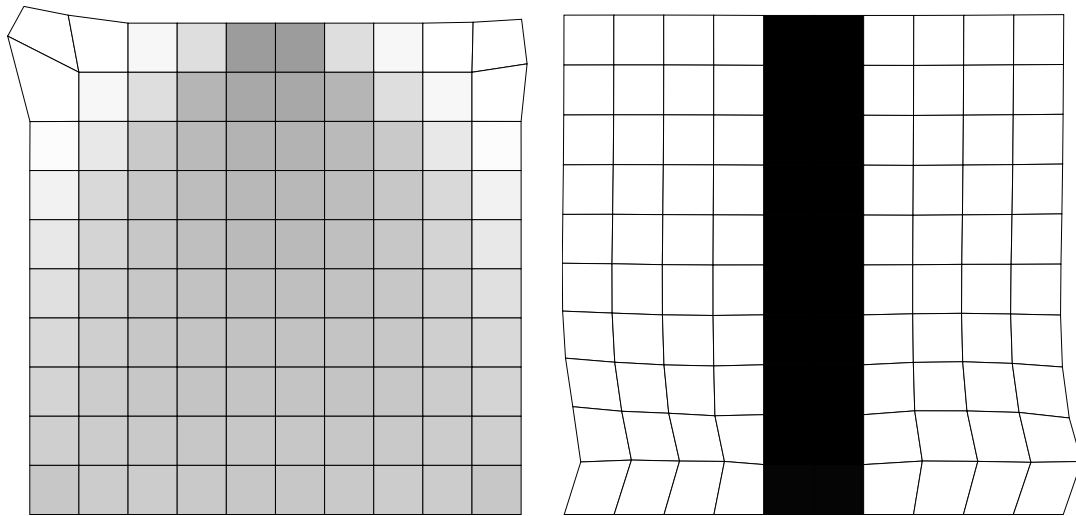
(a) Initial distribution of material and corresponding adjusted modeshape.

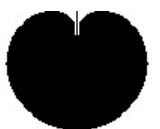(b) Distribution of material and corresponding adjusted modeshape after 1 iteration.
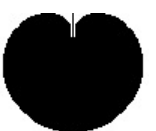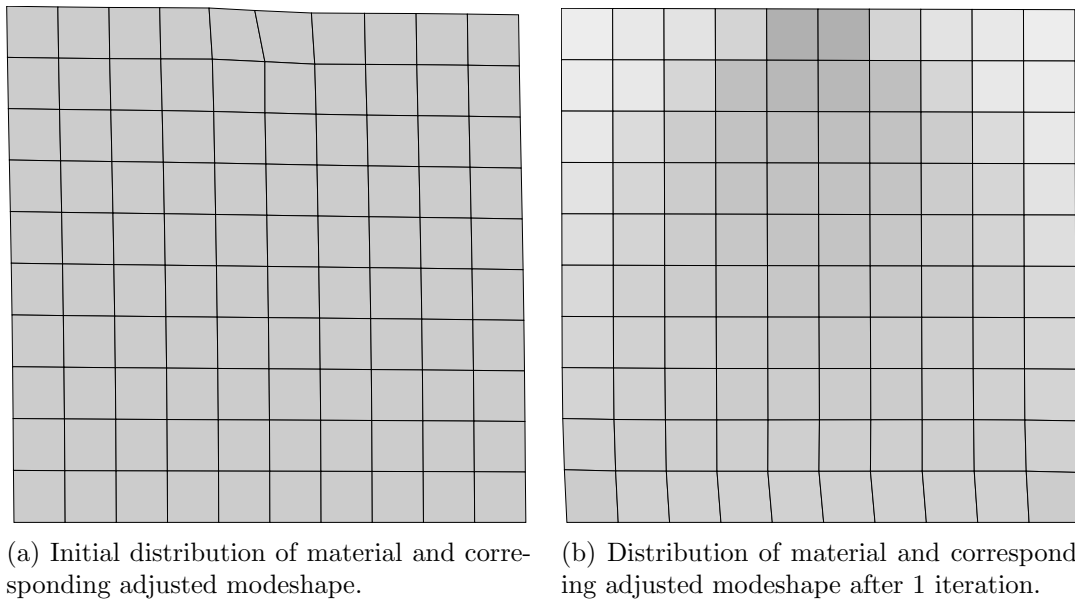
Figure 6-5: Initial material distributions and modeshapes using modified eigenvalue computation. Note that this is identical (up to sign change) to that in Figure 6-2b.

(a) Distribution of material and corresponding adjusted modeshape after 2 iterations. Here no spurious buckling mode is observed; compare with Figure 6-3a.

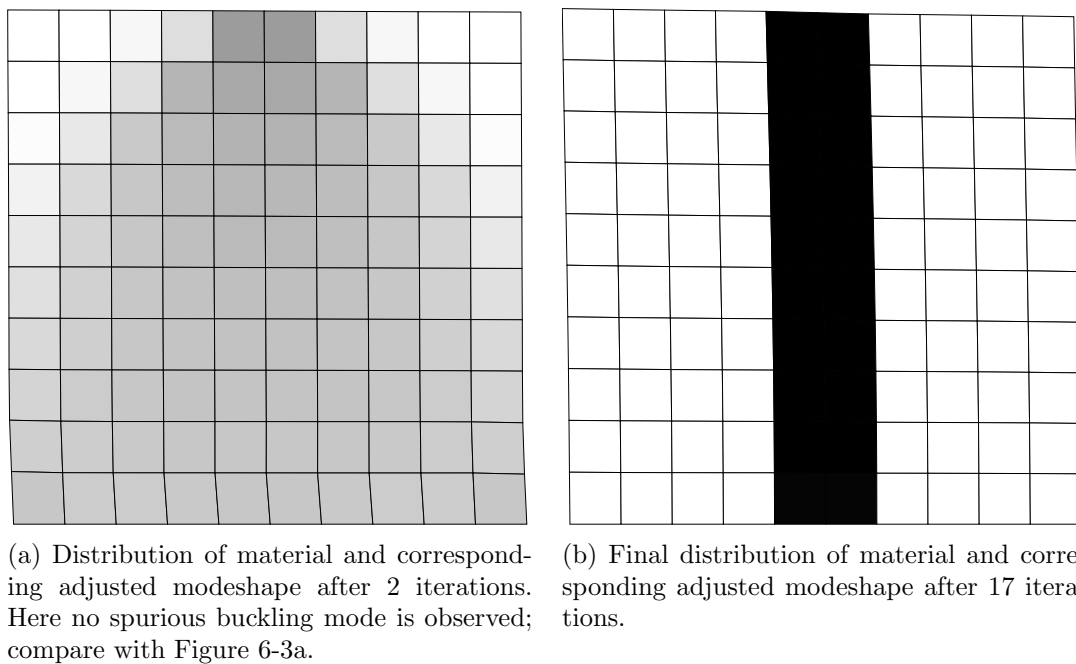(b) Final distribution of material and corresponding adjusted modeshape after 17 iterations.

Figure 6-6: Material distribution and modeshapes using modified eigenvalue computation. Note the lack of spurious localised buckling modes.

6.2.3 Justification for removal of stresses from low density elements

Theorem 6.6. *If all stresses in low density elements are set to zero in the construction of the stress stiffness matrix K_σ, and if the smallest positive eigenvalue of equation (6.1) is finite, it does not correspond to a spurious localised buckling mode.*

Proof. Let v_l be a spurious localised buckling mode. Hence v_l is a sparse vector with non-zero entries only corresponding to nodes that are entirely surrounded by low density elements.

Let us now consider the rows and columns of K_σ that correspond with nodes surrounded by low density elements. The only contribution to K_σ in these rows and columns comes from the surrounding elements and so from (3.29), if the stresses σ are set to 0 then these rows and columns will have zero entries.

Suppose for contradiction that v_l is a solution to the eigenvalue problem (3.30). Then we have

$$Kv_l + \lambda K_\sigma v_l = 0 \tag{6.18}$$

with λ finite. Multiplying on the left by v_l^T we obtain

$$v_l^T Kv_l + \lambda v_l^T K_\sigma v_l = 0 \tag{6.19}$$

Now the only non-zero components of v_l occur in the nodes that are completely surrounded by low density elements. But the corresponding columns of K_σ are all zero, and hence

$$K_\sigma v_l = 0 \tag{6.20}$$

Thus substituting into (6.18) we see that $v_l^T Kv_l = 0$. As the matrix K is SPD this implies that $v_l = 0$ and hence cannot be a solution of the eigenvalue problem (6.1), which is a contradiction. $\square$

Note that if λ is infinite then any constraint on a lower bound of this is trivially satisfied. As such this constraint could be removed from the optimization formulation at that point.

Figures 6-5a–6-6b show the newly calculated modeshapes when the adjusted method described in Section 6.2.2 is applied. Note that the buckling is not occurring in the regions of low density and the is driven by the material that is within the domain.

Whilst assigning zero stress stiffness (or mass in the harmonic analysis case) contributions from elements of low density can eradicate these spurious modes, this is not consistent with the underlying model of the structure given in Section 3.4. Indeed, if one were to consider a structure where a small fraction (less than 10%) of material

was equidistributed throughout the design domain, the stress stiffness matrix would be the zero matrix, and as a result the critical load of the structure would be computed as infinite. This would happen regardless of the load vector's magnitude or direction and so would lead to erroneous results. This may be avoided if the stress stiffness contributions were based on a "relative" density fraction, though care would have to be taken to ensure the theory was consistent with the derivation in Section 3.4.

## 6.3 Structural optimization with discrete variables

Finding a global solution to binary programming problems is notoriously difficult. The methods for finding such minima can be broadly put into three categories: implicit enumeration, branch-and-bound and cutting-plane methods. The most popular implementations involve hybrids of branch-and-bound and cutting-plane methods. For a comprehensive description of these binary programming methods see, for example, Wolsey [182]. These methods were popular for structural optimization from the late 1960s through to the early 1990s. In 1994, Arora & Huang [15] reviewed the methods for solving structural optimization problems discretely.

In 1968, Toakley [174] applied a combination of cutting-plane methods and branch-and-bound to solve truss optimization problems. Using what is now known as the branch-and-cut method, this method was resurged in 2010 by Stolpe and Bendsøe [162] to find the global solution to a minimisation of compliance problem, subject to a constraint on the volume of the structure.

In 1980, Farkas and Szabo [51] applied an implicit enumeration technique to the design of beams and frames. Branch-and-bound methods have been used by, amongst others, John et al. [80], Sandgren [143, 144] and Salajegheh & Vanderplaats [142] for structural optimization problems. In the latest of these papers, the number of variables in the considered problem was 100 and in some cases took over one week of CPU time on a modern server to compute the solution. Whilst these methods do find global minima, they suffer from exponential growth in the computation time as the number of variables increases.

In this chapter, we introduce an efficient method for binary programming and apply it to topology optimization problems with a buckling constraint. In doing so, we avoid the problem of spurious buckling modes and can find solutions to large two-dimensional problems ($\mathcal{O}(10^5)$ variables).

Due to the dimensionality of the problems, and the complexity of derivative-free methods for binary programs, we will use derivative information to reduce this complexity. The efficiency of topology optimization methods involving a buckling constraint is
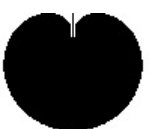
severely hindered by the calculation of the derivatives of the buckling constraint. This calculation typically takes an order of magnitude more time than the linear elasticity analysis. With this in mind, the proposed fast binary descent method we introduce will try to reduce the number of derivative calculations required.

The remainder of this chapter is organised as follows. In Section 6.4, we formulate the topology optimization problem to include a buckling constraint. Section 6.5 motivates and states the new method which we use to solve the optimization problem. Section 6.6 then contains implementation details and results for a number of two-dimensional test problems. Finally in Section 6.7, we draw conclusions about the proposed algorithm.

6.4   Formulation of topology optimization to include a buckling constraint

Given a safety factor parameter $c_s > 0$, a bound of the form $\lambda \geq c_s$, where λ is the critical load solving (6.1), is equivalent to the semidefinite constraint

$$K + c_s K_\sigma \succeq 0.$$

This means that all the eigenvalues of the system $(K + c_s K_\sigma)$ are non-negative. This happens only if $\sum_{i=1}^{M} v_i^T (K + c_s K_\sigma) v_i \geq 0$ where v_i are the M buckling modes that solve $(K + \lambda K_\sigma) v_i = 0$. If we let $x \in \{0, 1\}^n$ represent the density of material in each of the elements of the mesh, with $x_i = 0$ corresponding to an absence of material in element i and $x_j = 1$ corresponding to element j being filled with material, the problem to be solved becomes:

$$\min_x \sum x_j \tag{6.21a}$$

$$\text{subject to} \qquad c_1(x) := c_{\max} - f^T u(x) \geq 0 \tag{6.21b}$$

$$c_2(x) := \sum_{i=1}^{M} v_i(x)^T (K(x) + c_s K_\sigma(x)) v_i(x) \geq 0 \tag{6.21c}$$

$$x \in \{0, 1\}^n \tag{6.21d}$$

$$K(x) u(x) = f \tag{6.21e}$$

$$[K(x) + \lambda(x) K_\sigma(x)] v_i(x) = 0. \quad \forall i = 1, \ldots, M \tag{6.21f}$$

6.4.1 Derivative calculations

To use the binary descent method which will be introduced in Section 6.5 (or a SDP method) we need an efficient way of calculating the derivative of the constraints with respect to the variables x_i. As will be seen in Section 6.6, the computation of derivatives of the buckling constraint (6.21c) is the bottleneck in our optimization algorithm, so it is imperative that we have an analytic expression for this. To calculate the derivatives, the binary constraints on the variables are relaxed and assume that the following holds

$$K(x) = \sum_\ell x_\ell K_\ell,$$

where K_ℓ is the local element stiffness matrix. The derivative of this with respect to the density of an element x_i is given by

$$\frac{\partial K}{\partial x_i}(x) = K_i.$$

Calculating the derivative of the buckling constraint requires the derivation of an expression for $\frac{\partial K_\sigma}{\partial x_i}$. This quantity is non-trivial to compute, unlike the derivative of a mass matrix which would be in place of the stress stiffness matrix in structural optimization involving harmonic modes. The stress field σ_ℓ on an element ℓ is a 3×3 tensor with 6 degrees of freedom. This can be written in three dimensions as

$$\sigma_\ell = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix}_\ell = x_\ell E_\ell B_\ell u,$$

which in two dimensions reduces to

$$\sigma_\ell = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}_\ell = x_\ell E_\ell B_\ell u,$$

where u are the nodal displacements of the element, E_ℓ is a constant matrix of material properties and B_ℓ contains geometric information about the element. The indices $1, 2$ and 3 refer to the coordinate directions of the system.

We consider the two-dimensional case, and note that all the following steps have a direct analogue in three dimensions. We write the stress stiffness matrix given in (3.30)

as follows.

$$K_\sigma = \sum_{\ell=1}^{n} \int G_\ell^T \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 & 0 \\ \sigma_{12} & \sigma_{22} & 0 & 0 \\ 0 & 0 & \sigma_{11} & \sigma_{12} \\ 0 & 0 & \sigma_{12} & \sigma_{22} \end{bmatrix}_\ell G_\ell \mathrm{d}V_\ell, \tag{6.22}$$

where G_ℓ is a matrix containing derivatives of the basis functions that relates the displacements of an element ℓ to the nodal degrees of freedom [39] and n is the total number of elements in the finite-element mesh $\mathcal{T}$. Now define a map $\Theta : \mathbb{R}^3 \mapsto \mathbb{R}^{4\times 4}$ by

$$\Theta(\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}) := \begin{bmatrix} \alpha & \gamma & 0 & 0 \\ \gamma & \beta & 0 & 0 \\ 0 & 0 & \alpha & \gamma \\ 0 & 0 & \gamma & \beta \end{bmatrix}.$$

Note that Θ is a linear operator. Using this, (6.22) becomes

$$\begin{aligned} K_\sigma &= \sum_{\ell=1}^{n} \int G_\ell^T \Theta(x_\ell E_\ell B_\ell u) G_\ell \ \mathrm{d}V_\ell \\ &= \sum_{\ell=1}^{n} \int G_\ell(\xi)^T \Theta(x_\ell E_\ell B_\ell(\xi)u) G_\ell(\xi) \ \mathrm{d}V_\ell \\ &\approx \sum_{\ell=1}^{n} \sum_{j} \omega_j G_\ell(\xi_j)^T \Theta(x_\ell E_\ell B_\ell(\xi_j)u) G_\ell(\xi_j) \end{aligned} \tag{6.23}$$

where ω_j are the weights associated with the appropriate Gauss points ξ_j that implement a chosen quadrature rule to approximate the integral. Differentiating the equilibrium equation (6.21e) with respect to the density x_i yields

$$\frac{\partial K}{\partial x_i} u + K \frac{\partial u}{\partial x_i} = 0$$

and hence

$$\frac{\partial u}{\partial x_i} = -K^{-1} \frac{\partial K}{\partial x_i} u.$$

Now consider the derivative of the operator Θ with respect to x_i. Since Θ is linear

$$\begin{aligned} \frac{\partial \Theta(x_\ell E_\ell B_\ell u)}{\partial x_i} &= \Theta\Big(\frac{\partial}{\partial x_i} x_\ell E_\ell B_\ell u(x) \Big) \\ &= \Theta\Big(\delta_{i\ell} E_\ell B_\ell(\xi_j) u + x_\ell E_\ell B_\ell(\xi_j) \frac{\partial u}{\partial x_i} \Big) \end{aligned}$$

where $\delta_{i\ell}$ is the Kronecker Delta.

Applying the chain rule to (6.23) we obtain

$$\frac{\partial K_\sigma}{\partial x_i} \approx \sum_{l=1}^{n} \sum_{j} \omega_j G_\ell(\xi_j)^T \frac{\partial \Theta(x_\ell E_\ell B_\ell(\xi_j)u)}{\partial x_i} G_\ell(\xi_j)$$

$$\frac{\partial K_\sigma}{\partial x_i} \approx \sum_{l=1}^{n} \sum_{j} \omega_j G_\ell(\xi_j)^T \Theta(\delta_{i\ell} E_\ell B_\ell(\xi_j)u - x_\ell E_\ell B_\ell(\xi_j) K^{-1} \frac{\partial K}{\partial x_i} u) G_\ell(\xi_j), \qquad (6.24)$$

where the approximation is due to the error in the quadrature rule used. This matrix can now be used to find the derivative of the buckling constraint which we require. For each variable $x_i = 1, \ldots, n$, (6.24) must be computed. As (6.24) contains a sum over $\ell = 1, \ldots, n$, it can be seen that computing $\frac{\partial K_\sigma}{\partial x_i}$ has computational complexity of $\mathcal{O}(n)$ for each i and hence computing (6.24) for all variables has complexity of $\mathcal{O}(n^2)$.

6.5 Fast Binary Descent Method

In this section, we motivate and describe a new method that we propose for solving the binary programming problem. If we solve the state equations (6.21e) and (6.21f) then problem (6.21) takes the general form

$$\min_x e^T x \qquad (6.25a)$$

$$\text{subject to} \qquad c(x) \geq 0 \qquad (6.25b)$$

$$x \in \{0, 1\} \qquad (6.25c)$$

with $x \in \mathbb{R}^n$, $c \in \mathbb{R}^m$ and $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$. Typically m will be small (less than 10) and $m << n$. We also assume that $x^0 = e$ is an initial feasible point of (6.25). Let k denote the current iteration, and x^k is the value of x on the k-th iteration.

The objective function $e^T x$ is a linear function of x that can be optimized by successively reducing the number of nonzero terms in x and we need not worry about errors in approximating this. However, the constraints are nonlinear functions of x and ensuring that (6.25b) holds is difficult. Accordingly, we now describe how a careful linearisation of the constraint equations can lead to a feasible algorithm. Taylor's theorem can be used to approximate $c(x^k)$

$$c(x^{k+1}) = c(x^k) + \sum_{i=1}^{n} \frac{\partial c(x^k)}{\partial x_i} (x_i^{k+1} - x_i^k) + \text{ higher order terms}$$

where $\frac{\partial c(x^k)}{\partial x_i}$ is determined using the explicit derivative results of the previous section.

The method will take discrete steps so that

$$x_i^{k+1} - x_i^k \in \{-1, 0, 1\} \qquad \forall i = 1, \ldots, n,$$

and so we must assume that the higher order terms will be small, but later a strategy will be introduced to cope with this when they are not.

Consider now variables x_i^k such that $x_i^k = 1$ that we wish to change to $x_i^{k+1} = 0$. Since $x_i^{k+1} - x_i^k = -1$, for the difference in the linearised constraint functions

$$c(x^{k+1}) - c(x^k) = \sum_{i=1}^n \frac{\partial c(x^k)}{\partial x_i}(x_i^{k+1} - x_i^k)$$

to be minimal, all the terms of $\frac{\partial c(x^k)}{\partial x_i}$ need to be as small as possible. However, since there are multiple constraints, the variables for which the gradient of one constraint is small may have a large gradient for another constraint.

Assuming a feasible point such that $c(x^k) > 0$ and ignoring the higher order terms,

$$c(x^{k+1}) = c(x^k) + \sum_{i=1}^n \frac{\partial c(x^k)}{\partial x_i}(x_i^{k+1} - x_i^k). \qquad (6.26)$$

We have to ensure $c(x^{k+1}) > 0$, so

$$c(x^k) + \sum_{i=1}^n \frac{\partial c(x^k)}{\partial x_i}(x_i^{k+1} - x_i^k) > 0$$

or equivalently

$$1 + \sum_{i=1}^n \frac{\partial c_j(x^k)}{\partial x_i}/c_j(x^k)(x_i^{k+1} - x_i^k) > 0 \qquad \forall j = 1, \ldots, m.$$

If $x_i^{k+1} \neq x_i^k$ then each normalised constraint $c_j(x^k)$ is changed by $\pm\frac{\partial c_j(x^k)}{\partial x_i}/c_j(x^k)$.

Define the sensitivity of variable i to be

$$s_i(x^k) = \max_{j=1,\ldots,m} \frac{\partial c_j(x^k)}{\partial x_i}/\max\{c_j(x^k), 10\epsilon\} \qquad (6.27)$$

where ϵ is the machine epsilon that guards against round off errors. For each variable, $s_i(x^k)$ is the most conservative estimate of how the constraints will vary if the value of the variable is changed. In one variable, this has the form shown in Figure 6-7. Figure 6-7a shows the absolute values of the linear approximations to the constraints based on their values and corresponding derivatives. Figure 6-7b shows the calculation that we

make based on normalising these approximations to compute which of the constraints would decrease the most if the variable x_i^k were changed. β_j is the point at which the line associated with the constraint c_j crosses the y-axis and so $\beta_j = 1 - \frac{\partial c_j(x^k)}{\partial x_i}/c_j(x^k)$. The amount that the normalised constraint c_j would change if the variable x_i^k were changed is then given by $1 - \beta_j = \frac{\partial c_j(x^k)}{\partial x_i}/c_j(x^k)$.

In this case the derivatives indicate that if the variable x_i^k were to be decreased, the second constraint is affected relatively more than the first constraint (as $\max\{a, b\} = b$), and hence the sensitivity associated with this variable x_i is given the value $s_i(x^k) = \frac{\partial c_2(x^k)}{\partial x_i}/c_2(x^k)$.
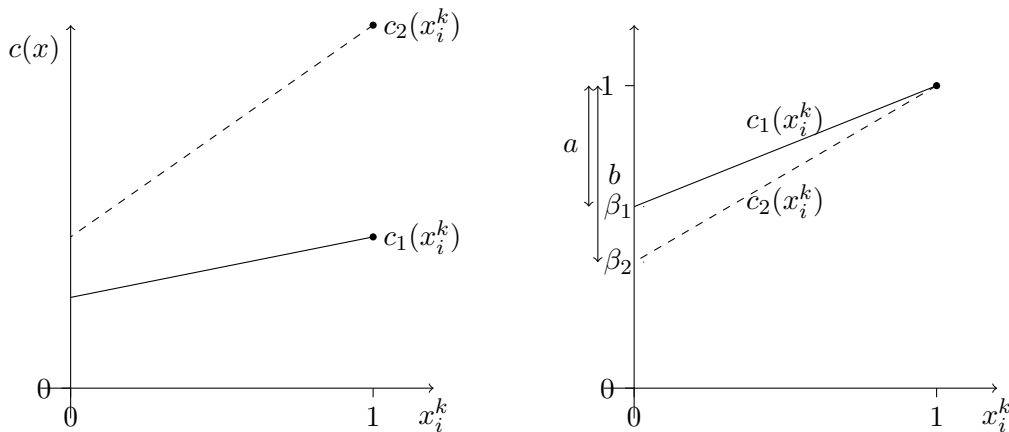
(a) Linear approximations to the constraints $c(x_i^k)$ in the case where $m = 2$. In this situation $x_i^k = 1$.

(b) Sensitivity calculation in one variable. Here $s_i(x^k) = \max\{a, b\} = b$.

Figure 6-7: Sensitivity calculation in one variable for the case when $m = 2$.

This sensitivity measure also provides an ordering so that if we choose to update variables in increasing order of their sensitivity, the changes in the constraint values are minimised. Now for ease of notation, let us assume that the variables are ordered so that

$$s_1 \le s_2 \le \ldots \le s_p \qquad \forall s_i \text{ s.t. } x_1^k, x_2^k, \ldots, x_p^k = 1 \tag{6.28}$$

$$s_{p+1} \ge s_{p+2} \ge \ldots \ge s_n \qquad \forall s_i \text{ s.t. } x_n^k, x_{n-1}^k, \ldots, x_{p+1}^k = 0 \tag{6.29}$$

To be cautious, instead of requiring $c(x^{k+1}) \ge 0$, we allow for the effects of the nonlinear terms and so are content if instead $c(x^{k+1}) \ge (1-\alpha)c(x^k)$ for some $\alpha \in (0, 1)$.

This implies that

$$c(x^k) + \sum_{i=1}^{n} \frac{\partial c^T(x^k)}{\partial x_i}(x_i^{k+1} - x_i^k) \geq (1 - \alpha)c(x^k),$$

or equivalently

$$\alpha c(x^k) + \sum_{i=1}^{n} \frac{\partial c^T(x^k)}{\partial x_i}(x_i^{k+1} - x_i^k) \geq 0.$$

To update the current solution, we consider the variables ordered so that (6.28) and (6.29) hold and find for some $\alpha \geq 0$

$$L := \max_{1 \leq \ell \leq p} \ell \text{ s.t. } \alpha c_j(x^k) - \sum_{i=1}^{\ell} \frac{\partial c_j(x^k)}{\partial x_i} > 0 \qquad \text{for all } j \in 1, \ldots, m. \qquad (6.30)$$

Then we decrease from 1 to 0 those variables $x_1^k, \ldots, x_L^k$ so as to reduce the objective function by a value of L.

However, there is the possibility that increasing variables from 0 to 1 could further reduce the objective function by reducing yet more variables from 1 to 0. This is tested by finding (or attempting to find) $J > 0$ such that

$$J := \max_{0 \leq \ell \leq (p-L)/2} \ell \text{ s.t. } \sum_{i=1}^{\ell} \frac{\partial c_j(x^k)}{\partial x_{p+i}} - \sum_{i=1}^{2\ell} \frac{\partial c_j(x^k)}{\partial x_{L+i}} \geq 0 \qquad \text{for all } j \in 1, \ldots, m. \quad (6.31)$$

So the variables corresponding to the terms in the first sum are increased from 0 to 1 but for each of these, two variables are decreased from 1 to 0, corresponding to the terms in the second summation. As there are more terms in the second summation the objective function improves whilst remaining a feasible solution. Hence the variables $x_{L+1}^k, \ldots, x_{L+2J}^k$ are decreased from 1 to 0 and the variables $x_{p+1}^k, \ldots, x_{p+J}^k$ are increased from 0 to 1. Note that in (6.30) and (6.31) the equations have to hold for each of the constraints $j = 1, \ldots, m$.

The coefficient α is a measure of how well the linear gradient information is predicting the change in the constraints. If the problem becomes infeasible, then the method has taken too large a step, so α is reduced in order to take a smaller step. However, recall the goal of this method is to compute the gradients as few times as possible, and so we wish to take steps that are as large as possible. If the step has been accepted for the previous two iterations without reducing α then α is increased to try and take larger steps and thus speed up the algorithm.

Note that if α is too large and the solution becomes infeasible then α is reduced and a smaller step is taken *without* recomputing the derivatives. Hence increasing α

by too much is not too detrimental to the performance of the algorithm. Based on experience, α is reset to 0.7α when the solution becomes infeasible and α is set to 1.5α when we want to increase it. These values appear stable and give good performance for most problems.

To ensure that at least one variable is updated, α must be larger than a critical value α_c given by

$$\alpha_c = \max_{j=1,\dots,m} \{(\frac{\partial c_j(x^k)}{\partial x_1^k})/c_j(x^k)\}.$$

This guarantees that $L \geq 1$ and at least one variable is updated. The upper bound $\alpha \leq 1$ must also be enforced so that $c(x^{k+1}) \geq 0$.

If we cannot make any further progress with this algorithm, we stop. Making further progress would be far too expensive as we would have to switch to a different integer programming strategy and the curse of dimensionality for the problems that we wish to consider prohibits this. However, we believe the computed solution is good because if we try and improve the objective function by changing the variable for which the constraints are infinitesimally least sensitive, the solution becomes infeasible.

The fast binary descent algorithm is thus presented in Algorithm 2:

Algorithm 2 Fast binary descent method

 1: Initialise x^0 and α.
 2: Compute objective function (6.25a) and constraints (6.25b)
 3: **if** x^0 not feasible **then**
 4:    **if** $x^0 = e$ **then**
 5:       Stop
 6:    **else**
 7:       Increase x^0 towards e.
 8:    **end if**
 9: **else**
10:    Compute derivatives $\frac{\partial c(x^k)}{\partial x_i}$
11:    Sort s_i (6.27)
12:    Compute values L from (6.30) and J from (6.31)
13:    Update the variables x_i^k that correspond to L and J from (6.30) and (6.31)
14:    **if** no variables updated **then**
15:       {Algorithm has converged}
16:       **return** with computed solution
17:    **end if**
18:    Compute objective function and constraints from equations (6.25a) and (6.25b)
19:    **if** not feasible **then**
20:       {Reject update step}
21:       Reduce α.
22:       GO TO 12
23:    **else**
24:       {Accept update step}
25:       Increase α if desired
26:       $k = k + 1$
27:       GO TO 10
28:    **end if**
29: **end if**

6.6   Implementation and results

We consider optimising isotropic structures with Young's modulus 1.0 and Poisson's ratio 0.3. The design domains are discretised using square bilinear elements on a uniform mesh.

The fast binary descent method has been implemented in Fortran90 using the HSL
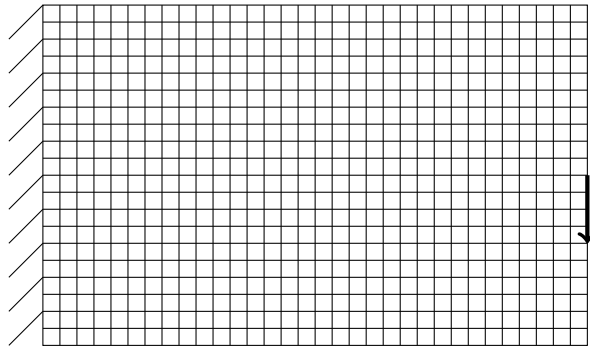

Figure 6-8: Design domain of a centrally loaded cantilevered beam. The aspect ratio of the design domain is 1.6 and a unit load is applied vertically from the centre of the right hand side of the domain.

mathematical software library [1] and applied to a series of two-dimensional structural problems. The linear solve for the calculation of displacements (6.21e) used HSL_MA87 [70], a DAG based direct solver designed for shared memory systems. For the size of problems considered, HSL_MA87 has been found to be very efficient and stable. The first 6 buckling modes of the system (3.30) were computed as these were sufficient to ensure all corresponding eigenvectors of the critical load were found. These eigenpairs were calculated using HSL_EA19 [124], a subspace iteration code, preconditioned by the Cholesky factorisation already computed by HSL_MA87. The sensitivities were passed through a standard low-pass filter[75] with radius $2.5h$ where h is the width of an element and ordered using HSL_KB22, a heapsort [180] algorithm.

The codes were executed on a desktop with an Intel® Core™2 Duo CPU E8300 @ 2.83GHz with 2GB RAM running a 32-bit Linux OS and were compiled with the gfortran compiler in double precision. All reported times are wall-clock times measured using `system_clock`.

6.6.1 Short cantilevered beam

We consider a clamped beam with a vertical unit external force applied to the free side as shown in Figure 6-8. Figures 6-9 to 6-11 refer to the solutions found with the same design domain and material properties but with differing buckling and compliance constraints.

Figure 6-9 is the computed solution to the problem with parameters $c_s = 0.9$ in (6.21c) and $c_{\max} = 35$ in (6.21b). In this case the compliance constraint $c_1(x^0)$ is large initially but the buckling constraint $c_2(x^0)$ is small initially. We see that the method has produced a typical optimum grillage structure with 4 bars under compression and
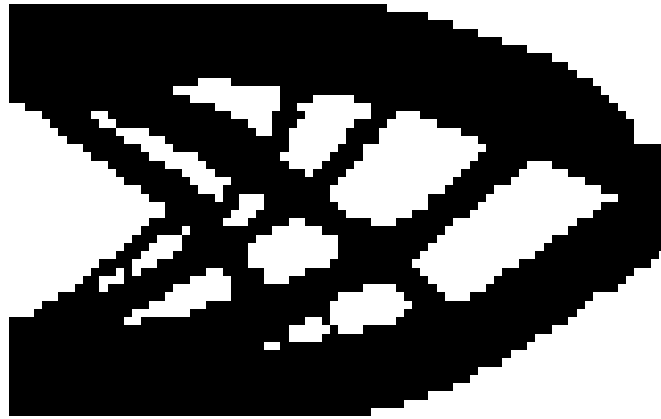
Figure 6-9: Solution found on mesh of 80×50 elements. The buckling constraint is set to $c_s = 0.9$ and the compliance constraint $c_{\max} = 35$. A volume of 0.6255 is attained. The buckling constraint c_2 is active and the compliance constraint c_1 is not.

only 3 bars under tension. Note that in the upper bar near the point of loading there is a distinct corner in the computed solution. This type of formation attracts high concentrations of strain energy and so if the problem were minimization of compliance then an optimization method would wish to avoid such situations. However, in this case optimization of this region is primarily dominated by the buckling constraint and the compliance is not the critical constraint.

Figure 6-10 is the computed solution to a problem with the same buckling constraint as in Figure 6-9 ($c_s = 0.9$) but is allowed to be more flexible with $c_{\max} = 60$ (i.e. the compliance constraint is not as restrictive). This results in a clear asymmetry in the computed solution in which the lower bar is much thicker than the upper bar. This lower bar is under compression with this loading, and hence would be prone to buckling. Thus optimization reinforced the lower bar to meet the buckling constraint.

Figure 6-11 was obtained as the solution for a problem with $c_s = 0.1$ and $c_{\max} = 30$. In this case the initial value of $c_1(x^0)$ is close to 0. The computed solution has only the compliance constraint active and hence the computed solution is more symmetrical than the solutions shown in Figures 6-9 and 6-10.

From Figures 6-9 to 6-11 it is possible to see a clear difference in the topology of the resulting solution depending on the parameters c_s and $c_{\max}$. Note that whilst one constraint may be violated if the updating process were to proceed, the other constraints have been utilized throughout the computation and have affected the path taken and resulting solution of the algorithm. The history of the algorithm when applied to the problem solved in Figure 6-9 where $c_{\max} = 35$ and $c_s = 0.9$ is displayed in Figures 6-12 to 6-14.

Figure 6-10: Solution found on mesh of 80×50 elements. The buckling constraint is set to $c_s = 0.9$ and the compliance constraint $c_{max} = 60$. A volume of 0.5535 is attained. Here the buckling constraint c_2 is active and the compliance constraint c_1 is not.
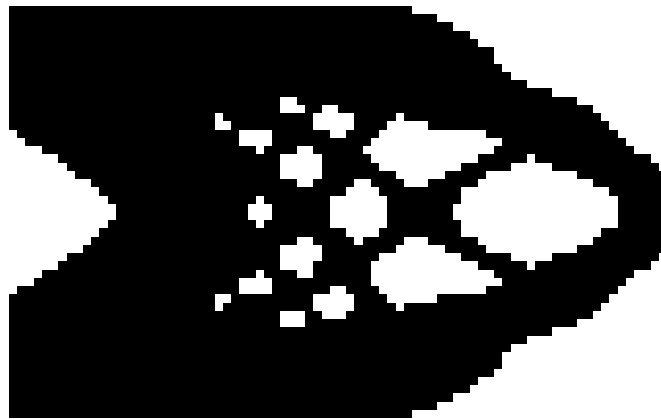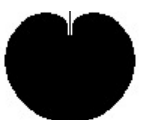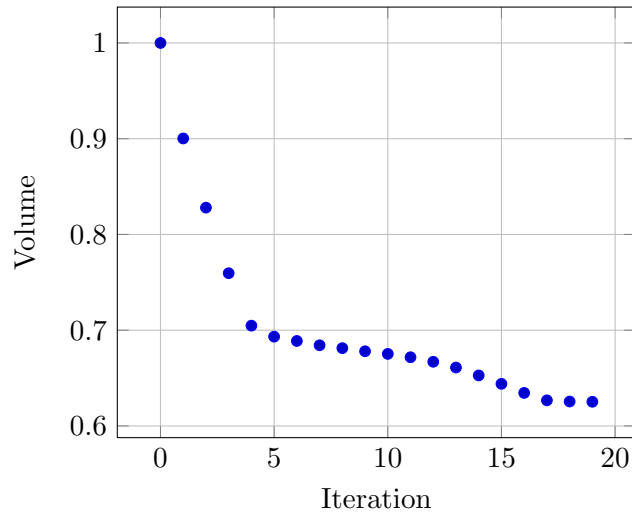

Figure 6-11: Solution found on mesh of 80×50 elements. The buckling constraint is set to $c_s = 0.1$ and the compliance constraint $c_{max} = 30$. A volume of 0.692 is attained. Here the compliance constraint c_1 is active and the buckling constraint c_2 is not.
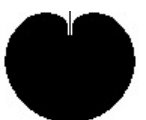
Figure 6-12: Volume – iterations of the fast binary descent method applied to the short cantilevered beam with $c_{\max} = 35$ and $c_s = 0.9$.

The plot of the objective function against iteration number shown in Figure 6-12 is monotonically decreasing and so shows that the method as described in Section 6.5 is indeed a descent method. Note that in the initial stages of the computation large steps are made and this varies as the computation progresses. Until iteration 4 large steps have been made and thus the objective function is swiftly decreasing. When going to iteration 5 taking a large step would make the current solution infeasible so the method automatically decreases the step size and hence the decrease in the objective function is reduced.

Figure 6-13 shows that the compliance constraint is inactive at the solution of this problem. Note that at all points the compliance of the structure is below the maximum compliance $c_{\max}$ and so the solution is feasible at all points with respect to c_1. If this plot is compared with Figure 6-12 then the large changes in compliance can be seen to occur where there are large reductions in volume and similarly when there is a small change in the volume the change in compliance is also small.

Figure 6-14 shows the lowest 6 eigenvalues of the system as the binary descent method progresses. We see that on the 20-th iteration the lowest eigenvalue is below the constraint c_s and so the computed solution is at iteration 19. At iterations 5 and 8 we see that the eigenvalue constraint is close to being violated. The increase in the lowest eigenvalue at the subsequent steps corresponds to a local thickening of the structure around the place where the buckling is most concentrated. This shows that the method has re-introduced material in order to move away from the constraint
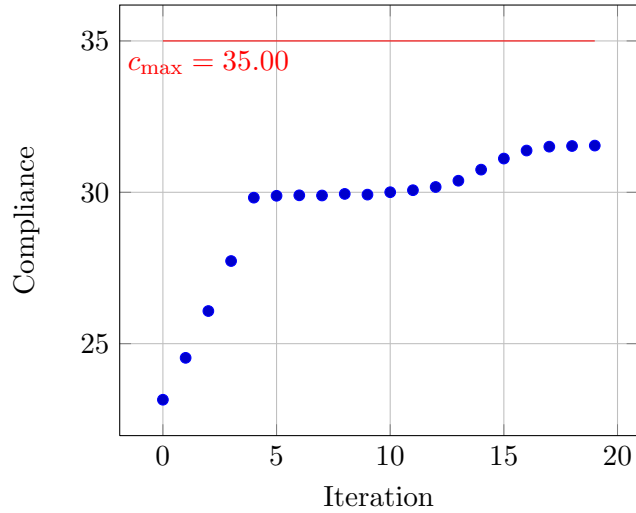
Figure 6-13: Compliance – iterations of the fast binary descent method applied to the short cantilevered beam with $c_{\max} = 35$ and $c_s = 0.9$.
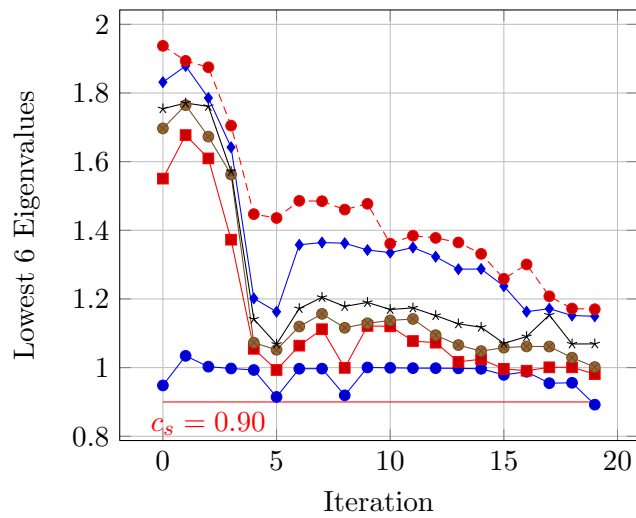

Figure 6-14: Eigenvalues – iterations of the fast binary descent method applied to the short cantilevered beam with $c_{\max} = 35$ and $c_s = 0.9$.. Note that on the 20-th iteration the eigenvalue constraint is violated, thus the computed solution is at the 19-th iteration.

(a) Design domain with width to height ratio $3 : 10$.

(b) Optimal design on a 30×100 mesh with $c_s = 0.225$ and $c_{max} = 22.5$. Here c_2 is active and c_1 is not.

(c) Optimal design on a 30×100 mesh with $c_s = 0.001$ and $c_{max} = 60$. Here c_1 is active and c_2 is not.
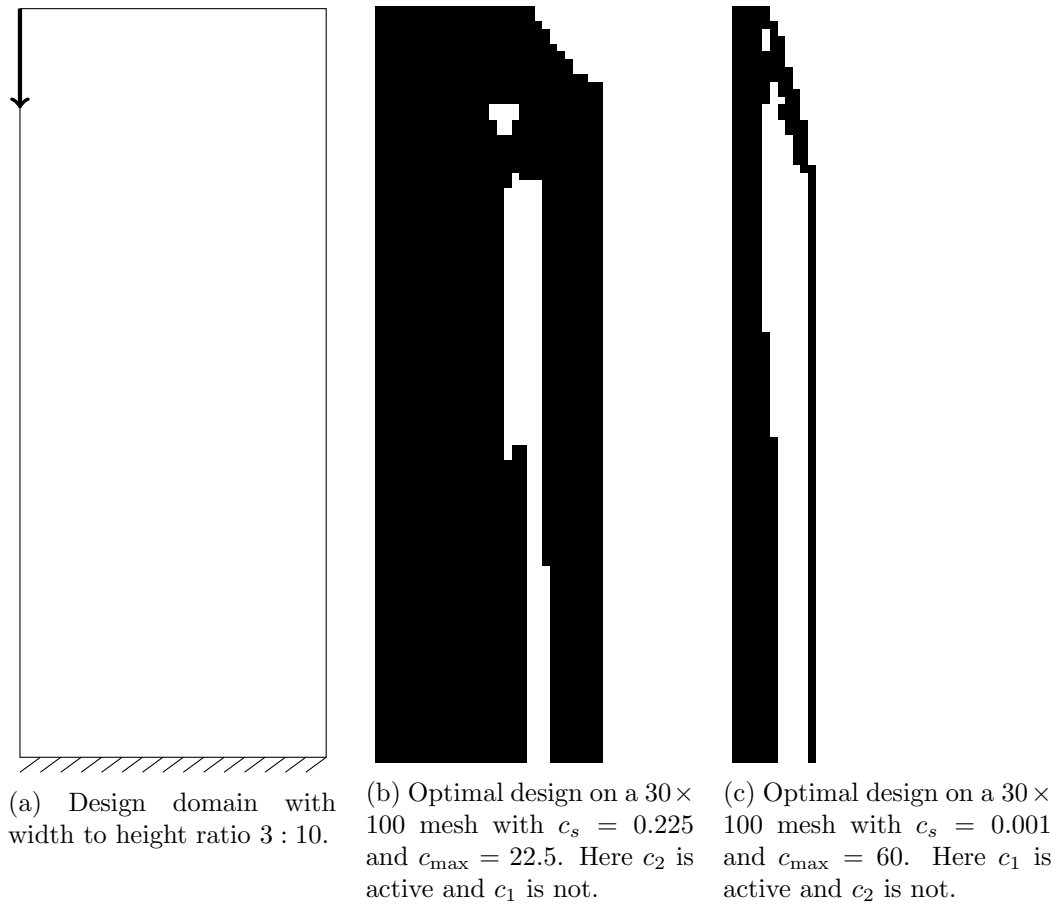
Figure 6-15: Design domain and results from the fast binary descent method applied to a column loaded at the side.

boundary. The nonlinearity in $c_2(x)$ is clear from the non-monotonic behaviour seen in Figure 6-14. Generally we do see the eigenvalues converging and that supports the intuitive optimality criteria of coincidental eigenvalues.

Figure 6-14, when viewed in combination with Figure 6-13 shows that for the history of the algorithm the solutions are all feasible.

6.6.2 Side loaded column

In this section we consider a tall design domain fixed completely at the bottom carrying a vertical load applied at the top corner of the design domain. The design domain is shown in Figure 6-15a and the computed solutions to this problem with differing constraints are shown in Figures 6-15b and 6-15c. The problem solved in Figure 6-15b has $c_s = 0.225$ and $c_{max} = 22.5$. The problem solved in Figure 6-15c has $c_s = 0.001$ and $c_{max} = 60$.
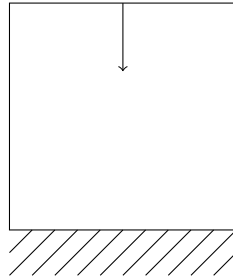
Figure 6-16: Design domain of model column problem. This is a square domain of side length 1 with a unit load acting vertically at the midpoint of the upper boundary of the space.

In Figure 6-15c as the constraints are relaxed compared with the problem in Figure 6-15b, the computed solution has a significantly lower objective function. However, it follows the same structural configuration where the main compressive column directly under the load resists the buckling and the slender column on the side provides additional support in tension to reduce bending. In both of these structures the path of the optimization is driven by the first buckling mode.

6.6.3 Centrally loaded column

We consider a square design domain (Figure 6-16). A unit load is applied vertically downwards at the centre of the top of the design domain and the base is fixed.

Figures 6-17 to 6-20 present results for a mesh of 60×60 elements for a range of values of the constraints. Figures 6-17 and 6-18 have $c_s = 0.5$ with $c_{\max} = 5$ and $c_{\max} = 5.5$, respectively. This small change in the compliance constraint results in two distinct configurations. Figure 6-18 with the higher compliance constraint achieves a lower volume and has the compliance constraint active as opposed to the buckling constraint which is active in Figure 6-17.

Distinct *"Λ-like"* structures have been found in Figures 6-19 and 6-20. These problems share the parameter $c_{\max} = 8$ but vary in that they have $c_s = 0.4$ and $c_s = 0.1$, respectively. The higher buckling constraint of Figure 6-19 leads to the development of thick regions in the centre of the supporting legs. These regions help to resist the first order buckling mode of the individual legs and are not seen in Figure 6-20 as the buckling constraint is lower. Figure 6-21 is the solution to a problem with the same parameters as the problem considered in Figure 6-20 but is solved on a much finer 200×200 mesh.

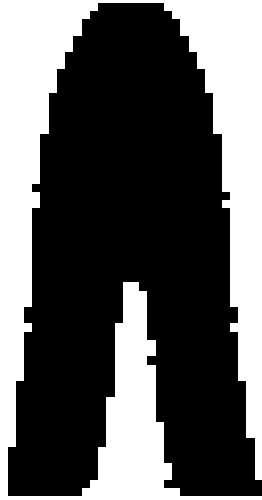These results can be compared directly with those found by Kočvara and Stingl

Figure 6-17: Solution computed on a mesh of 60×60 elements. The buckling constraint is set to $c_s = 0.5$ and the compliance constraint $c_{max} = 5$. Here, the compliance constraint is active and the buckling constraint is inactive.

Figure 6-18: Solution computed on a mesh of 60×60 elements. The buckling constraint is set to $c_s = 0.5$ and the compliance constraint $c_{max} = 5.5$. In this case, compared with Figure 6-17, the higher compliance constraint has led to a solution where this constraint is inactive and the buckling constraint is now active.
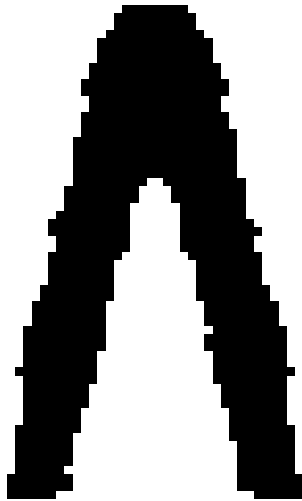

Figure 6-19: Solution computed on a mesh of 60×60 elements. The buckling constraint is set to $c_s = 0.4$ and the compliance constraint $c_{max} = 8$. A volume of 0.276 is attained.
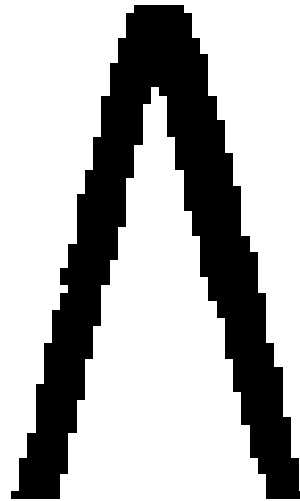

Figure 6-20: Solution computed on a mesh of 60×60 elements. The buckling constraint is set to $c_s = 0.1$ and the compliance constraint $c_{max} = 8$. A volume of 0.183 is attained.

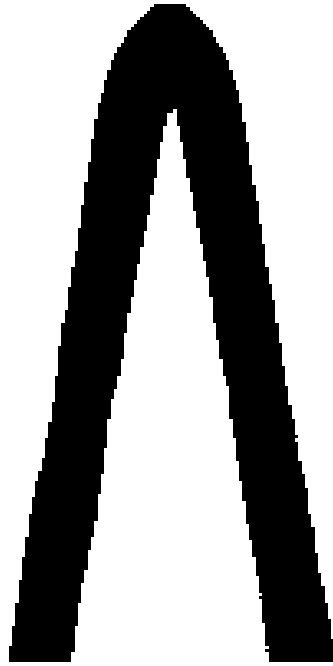Figure 6-21: Solution computed on a mesh of 200×200 elements. The buckling constraint is set to $c_s = 0.1$ and the compliance constraint $c_{\max} = 8$. A volume of 0.1886 is attained. Compare with Figure 6-20.

[92]. The design domain and loading are comparable, however they use SDP methods to solve a non-penalised problem in a VTS setting and hence find intermediate densities. The *"Λ-like"* structure is visible in their solutions, although the interior of the structure is filled with material of intermediate density.

From Figures 6-17 to 6-21 we see that the symmetry of the problem is not present in the computed solution. As Stolpe [161] and Rozvany [136] have shown, since we do not have continuous variables we do not necessarily expect the optimal solution to these binary programming problems to be symmetric. The asymmetry in the computed solutions arise from (6.30) and (6.31) as only a subset of elements with precisely the same sensitivity values may be chosen to be updated and so the symmetry may be lost.

Table 6.1 summarises the results obtained when solving the problem considered in Figures 6-20 and 6-21 but with varying mesh sizes. Note the problem size that the fast binary method has been able to solve. A computation on a two-dimensional mesh of 3×10^4 elements took less than 8 hours on a modest desktop and 4×10^4 elements took around 12 hours. This speed is attained because the number of derivative calculations appears to not be dependent on the number of variables. Figure 6-22 shows a log-log plot of the number of optimization variables against the wall-clock time taken to compute a solution. As the plot appears to have a gradient close to 2 this indicates

Problem size n	Objective	Derivative calculations	Analyses	Time (mins) to 3 s.f.	Proportion of time on $\partial c_2/\partial x$
$30 \times 30 = 900$	0.266	11	26	$4.21E-01$	0.623
$40 \times 40 = 1600$	0.229	12	22	$1.10E+00$	0.782
$50 \times 50 = 2500$	0.213	11	21	$2.29E+00$	0.857
$60 \times 60 = 3600$	0.183	26	31	$6.73E+00$	0.901
$70 \times 70 = 4900$	0.187	24	28	$1.16E+01$	0.931
$80 \times 80 = 6400$	0.185	21	24	$1.81E+01$	0.948
$90 \times 90 = 8100$	0.184	20	22	$2.85E+01$	0.948
$100 \times 100 = 10000$	0.184	18	23	$4.06E+01$	0.966
$110 \times 110 = 12100$	0.188	19	21	$6.12E+01$	0.973
$120 \times 120 = 14400$	0.187	18	20	$8.45E+01$	0.978
$130 \times 130 = 16900$	0.184	19	23	$1.19E+02$	0.980
$140 \times 140 = 19600$	0.188	17	18	$1.54E+02$	0.984
$175 \times 175 = 30625$	0.173	20	22	$3.86E+02$	0.985
$180 \times 180 = 32400$	0.191	20	23	$4.58E+02$	0.989
$200 \times 200 = 40000$	0.188	21	24	$7.34E+02$	0.990
$317 \times 317 = 100489$	0.181	19	20	$4.23E+03$	0.996

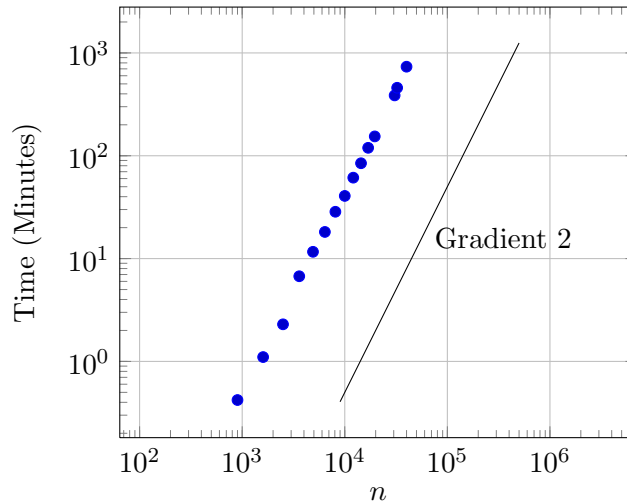Table 6.1: Table of results for the centrally loaded column

Figure 6-22: Log–log plot of time against the number of optimization variables. The gradient of this plot appears to be 2, suggesting that the time to compute the solution to a problem with n variables is $O(n^2)$.

that the time to compute a solution is $\mathcal{O}(n^2)$.

This problem can be compared to that solved by Bogani et al. [23]. They solve the continuous problem using modern SDP methods in a non-penalised manner i.e. the convex problem. On a similar machine they solved a problem with the same loading conditions discretised into 5000 variables in around 35 minutes. Compare this with the 4900 variable problem detailed in Table 6.1 and it can be seen that the fast binary descent method finishes in around 12 minutes for a similar sized problem.

A detailed examination of the computational cost indicates that the vast majority of the computational cost is in the computation of the derivative of the buckling constraint (see the final column of Table 6.1). A massively parallel implementation of this step is possible and it is anticipated that it should achieve near optimal speedup as no information transfer is required for the calculation of the derivative with respect to the individual variables.

Finally, the solution found when the design domain was discretised into 175×175 elements had the lowest objective function. It is possible that this is due to the slight difference in the symmetries of the problem when the domain is split into an odd number of elements as opposed to splitting into an even number of elements. The reasons for this are not fully understood and warrant future investigation.

6.7 Conclusions

Spurious buckling modes have been observed and investigated. The technique for eradicating these spurious eigenvectors from the computations has been shown to fully remove the numerically driven modes. However, it is also been shown that this technique makes the results inconsistent with the underlying state equations and thus a large amount of error is involved if this is employed.

The main computational cost associated with topology optimization involving buckling is the calculation of the derivatives of the buckling load. We have presented an analytic formula for this but it still remains the most expensive part of the algorithm. To reduce the computational cost we have developed an algorithm that aims to minimise the number of these computations that are required. The method is a descent method that enforces feasibility at each step and thus could be terminated early and would still result in a feasible structure.

We have numerically shown that the algorithm scales quadratically with the number of elements in the finite-element mesh of the design domain. This corresponds to the analytical result that the derivative of the stress-stiffness matrix with respect to each of the design variables is an $\mathcal{O}(n^2)$ operation. The numerical experiments demonstrate the efficiency of the method for binary topology optimization using compliance and buckling constraints.

7

Analysis of Evolutionary Structural Optimization

This chapter is concerned with the convergence of the ESO algorithm. Section 7.1 begins the chapter by introducing the algorithm. This is followed by a typical example of the convergence behaviour of the algorithm. The choice of strain energy density as the sensitivity is demonstrated in Section 7.3. Sections 7.4 and 7.5 find analytic examples of nonlinear and linear behaviour of the linear elasticity equations respectively. A motivating example in the continuum setting is presented in Section 7.6 that shows the nonlinear behaviour of the algorithm and inspires the modified ESO algorithm which is given in Section 7.7. This modified algorithm is then applied to the tie beam problem in Section 7.8 in order to show its effectiveness.

7.1 The ESO algorithm

Evolutionary Structural Optimization (ESO) is a technique for topology optimization developed by Xie and Steven in 1993 [184] and has been improved upon continuously since then.

In its simplest form, ESO starts with a discretised mesh of the design domain and fully populates each of the elements with material. Some form of *sensitivity* is then calculated, and those elements which the sensitivity value deem to be of least worth to the structure are removed. New sensitivities are then computed on the updated structure and this process is repeated. This is summarised in Algorithm 3.

ESO has been employed to try and optimize the compliance-volume product (CV) of a structure. In order to do so, a number of different sensitivity measures have been proposed, namely the Van der Waals stress of an element, or the element strain energy density. It is the latter on which we concentrate our thoughts.

There have been attempts to analyse the convergence of ESO. For example, Tan-

---

Algorithm 3 Evolutionary Structural Optimization (ESO)

 1: Mesh design domain
 2: Define a rejection ratio RR
 3: **loop**
 4:    Perform structural analysis of structure
 5:    Calculate elemental sensitivities s_i for all elements i
 6:    {Filter sensitivities (optional)}
 7:    Remove elements i with $s_i \leq RR \min_j \{s_j\}$
 8: **end loop**

---

skanen has shown that ESO updates follow the same path as a form of the simplex method would take [172]. This type of analysis gives a theoretical basis for ESO as an optimization algorithm, but does not address the fact that it is using a linear programming method to optimize a nonlinear function. This chapter of this thesis will investigate this aspect.

7.2   Typical convergence behaviour of ESO

The motivation for this chapter stemmed from graphs such as Figure 7-1 which is a replica of the results of Edwards [47]. This type of convergence graph is typical of those generated by ESO.
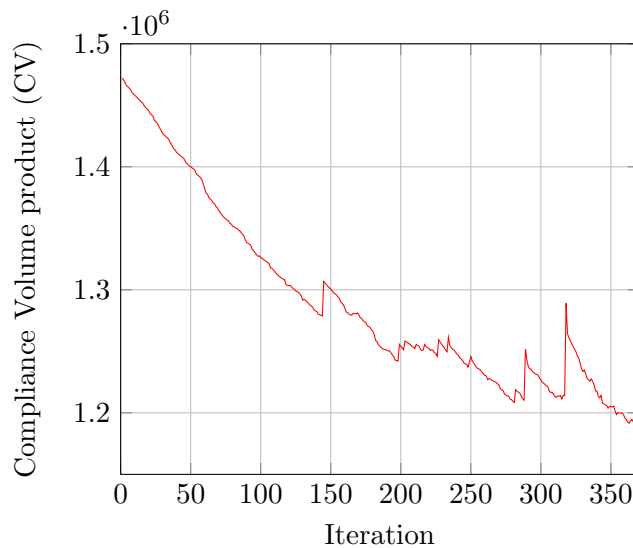
Figure 7-1: Compliance volume (CV) plot for ESO applied to the short cantilevered beam.

We can see in Figure 7-1 that it is clear the graph is not monotonically decreasing. As ESO is inherently a discrete algorithm the notion of optimality is that of a global optimum. However, in this example there are $64 \times 40 = 2560$ design variables. Hence there are $2^{2560} \approx 4 \times 10^{770}$ different possible solutions, among which we are trying to find the optimum. Note that there are an estimated 10^{80} atoms in the observable universe!

It is clear that there is nothing inherent in the ESO algorithm which will guarantee that the global optimum will be attained. In this chapter we ask two tractable questions:

1. Why does the ESO algorithm not reduce the objective function monotonically?

2. Can we adapt the ESO algorithm so it does reduce the objective function monotonically?

7.3 Strain energy density as choice of sensitivity

Let us begin by defining the strain energy density, U_e on an element e.

$$U_e := \tfrac{1}{2} u_e^T K_e u_e \tag{7.1}$$

where u_e is the vector of displacements associated with the element e, and K_e is the local element stiffness matrix of element e.

From the equilibrium equations, we have

$$Ku - f = 0$$

where K is the finite element stiffness matrix, u is the displacement vector and f is an applied force. Differentiating this with respect to an element x_e we obtain

$$\frac{\partial K}{\partial x_e} u + K \frac{\partial u}{\partial x_e} = 0$$
$$\frac{\partial u}{\partial x_e} = -K^{-1} \frac{\partial K}{\partial x_e} u. \tag{7.2}$$

If we now consider compliance:

$$C = f^T u \tag{7.3}$$

Differentiating with respect to an element x_e we get

$$
\begin{aligned}
\frac{\partial C}{\partial x_e} &= \frac{\partial u^T}{\partial x_e} K u + u^T \frac{\partial K}{\partial x_e} u + u^T K \frac{\partial u}{\partial x_e} \\
&= u^T \frac{\partial K}{\partial x_e} u + 2 u^T K (-K^{-1} \frac{\partial K}{\partial x_e} u) \\
&= -u^T \frac{\partial K}{\partial x_e} u.
\end{aligned}
\tag{7.4}
$$

In an ESO context, the stiffness matrix K is given by the following equation.

$$
K = \sum_e x_e K_e
\tag{7.5}
$$

and so the derivative of this with respect to an element x_e is

$$
\frac{\partial K}{\partial x_e} = K_e.
\tag{7.6}
$$

Substituting this in, we have

$$
\begin{aligned}
\frac{\partial C}{\partial x_e} &= -u_e^T K_e u_e \\
&= -2 U_e
\end{aligned}
\tag{7.7}
$$

The volume of the structure V is given by

$$
V := \sum_e x_e.
\tag{7.8}
$$

Hence the derivative of CV is given by

$$
\begin{aligned}
\frac{\partial CV}{\partial x_e} &= \frac{\partial C}{\partial x_e} V + C \frac{\partial V}{\partial x_e} \\
&= -2 U_e V + f^T u.
\end{aligned}
\tag{7.9}
$$

As we wish to minimise CV, we want to change from 1 to 0, those elements that have maximum $\frac{\partial CV}{\partial x_e}$.

$$
\arg\max_e \frac{\partial CV}{\partial x_e} = \arg\max_e \left(-2 U_e V + f^T u \right) = \arg\max_e -2 U_e V = \arg\min_e U_e
\tag{7.10}
$$

hence those elements with least strain energy density are precisely the elements for which the derivative of the objective function, CV, is maximum. So whilst the algorithm

only considers the strain energy density of an element, we can equivalently analyse the method by instead talking about this as the derivative of CV.

7.4 Nonlinear behaviour of the elasticity equations

Definition 7.1. *A function $f : X \to \mathbb{R}$ is convex if for all $x, y \in X$ and $\lambda \in [0, 1]$*

$$\lambda f(x) + (1 - \lambda)f(y) \geq f(\lambda x + (1 - \lambda)y).$$

Theorem 7.2. *Suppose $g, h : \mathbb{R}^n \mapsto \mathbb{R}$ are non-negative and convex. Then the product gh is convex.*

Note this is set as Exercise 3.32 in Boyd and Vandenburghe [25] and we include the proof here for completeness.

Proof. Let $\lambda \in [0, 1]$, and let $x, y \in \mathbb{R}^n$.

$$\begin{aligned}
gh(\lambda x + (1 - \lambda)y) &= g(\lambda x + (1 - \lambda)y)h(\lambda x + (1 - \lambda) \\
&\leq [\lambda g(x) + (1 - \lambda)g(y)][\lambda h(x) + (1 - \lambda)h(y)] & (7.11) \\
&= \lambda^2 g(x)h(x) + \lambda(1 - \lambda)g(y)h(x) + \\
&\qquad \lambda(1 - \lambda)g(x)h(y) + (1 - \lambda)^2 g(y)h(y) \\
&\leq \lambda^2 g(x)h(x) + (1 - \lambda)^2 g(y)h(y) & (7.12) \\
&\leq \lambda g(x)h(x) + (1 - \lambda)g(y)h(y) & (7.13)
\end{aligned}$$

Where in (7.11) we have used the fact that g and h are convex. (7.12) uses the non-negativity of g and h, and (7.13) makes use of the fact that $\lambda, (1 - \lambda) \in [0, 1]$. Thus gh is convex. $\square$

Corollary 7.3. *Compliance-volume product, CV, is convex, over the domain $x \in [0, 1]^n$.*

Proof. Svanberg [166] showed that compliance is convex. As compliance can be written as $u(x)^T K(x)u(x)$ and the matrix $K(x)$ is known to be SPD, this gives us the necessary non-negativity of the compliance.

Volume can be written as $e^T x$ where $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$. As this is linear, it is trivially convex, and non-negativity is trivial on the domain $x \in [0, 1]^n$. Hence Theorem 7.2 can be used immediately to give the result. $\square$

Definition 7.4. *We define Δ to be the difference between the linear approximation to the CV of a structure based on the derivative information $\frac{\partial CV}{\partial x_e}$ at a point x^k and the actual value of CV attained at the next iterate x^{k+1}. i.e.*

$$\Delta = CV(x^{k+1}) - (CV(x^k) - \sum_e \frac{\partial CV}{\partial x_e}(x^k))$$

where e denotes the elements to be updated from iterate k to iterate $k+1$.

Now we show, by way of three lemmata, the following theorem about the convergence of the ESO method.

Theorem 7.5. $\Delta \geq 0$ *and there exist structural configurations for which $\Delta = 0$ and $\Delta > M$ for any $M \in \mathbb{R}$.*

Lemma 7.6. *If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable then f is convex if and only if*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

for all x and y in $\mathbb{R}^n$.

Proof. Given in Boyd and Vandenburghe [25] Section 3.1.3. $\square$

Note that some sources will uses this expression as a definition of a convex function. However, our definition allows for non-differentiable functions to be considered convex.

Corollary 7.7. $\Delta \geq 0$

Proof. CV is convex by Corollary 7.3. As CV is differentiable then Lemma 7.6 states

$$CV(y) \geq CV(x) + \nabla CV(x)^T (y - x) \tag{7.14}$$

Let $x = x^k$ and $y = x^{k+1}$, so

$$(y - x)_e = \begin{cases} 0 & \text{if } x_e^k = x_e^{k+1} \\ -1 & \text{if } x_e^k \neq x_e^{k+1} \end{cases} \tag{7.15}$$

Using (7.15), (7.14) becomes

$$CV(x^{k+1}) \geq CV(x^k) - \sum_e \frac{\partial CV}{\partial x_e}(x^k) \tag{7.16}$$
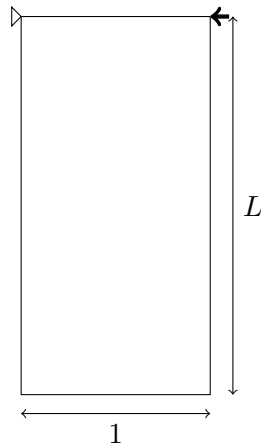
and thus $\Delta \geq 0$.

$\square$
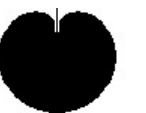
Figure 7-2: A frame consisting of 4 beams. The horizontal beams are of unit length, and the vertical beams have arbitrary length L. The frame is fixed in the top left corner completely and there is a unit load applied horizontally in the top right corner. The top and bottom beams are of interest to us.

What we have shown up until now is that an improved change in CV cannot exceed that given by the linear approximation to CV. Using an alternative version of the definition of convexity (as given in Lemma 7.6) this is clear. Now we show how if we consider a simple system we can show both that this bound will be tight and that there will be no upper bound on this quantity.

Lemma 7.8. *In a rectangular 4 beam system as (shown in Figure 7-2) with horizontal length 1 and arbitrary vertical length L that is fixed completely in one corner and loaded under compression with a horizontal load at the horizontally opposite corner, then $\Delta > M$ for any $M \in \mathbb{R}$.*

Proof. Consider the system in Figure 7-2. We model this as a frame with only 4 beam elements. Cook [39] section 4.2, gives the element stiffness matrix for a beam element. We assume that the beams have unit Young's modulus, unit cross-sectional area and unit moment of inertia of cross-sectional area. This allows us to compute some values of interest (the calculations are made symbolically with MATLAB's MuPAD feature). The volume of the whole structure is $2L + 2$ which reduces to $2L + 1$ when we remove either the top or bottom beam.

We build a finite element matrix where the nodes are ordered top left, bottom left, top right and then bottom right. Within this nodal ordering, we arrange the degrees of freedom with the horizontal displacement first, followed by vertical displacement and then the anticlockwise moment.

The element corresponding to the left beam has the following global stiffness matrix:

$$
\begin{pmatrix}
\frac{12}{L^3} & 0 & \frac{6}{L^2} & -\frac{12}{L^3} & 0 & \frac{6}{L^2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{L} & 0 & 0 & -\frac{1}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{6}{L^2} & 0 & \frac{4}{L} & -\frac{6}{L^2} & 0 & \frac{2}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{12}{L^3} & 0 & -\frac{6}{L^2} & \frac{12}{L^3} & 0 & -\frac{6}{L^2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{1}{L} & 0 & 0 & \frac{1}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{6}{L^2} & 0 & \frac{2}{L} & -\frac{6}{L^2} & 0 & \frac{4}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Similarly, the right beam has the following global stiffness matrix:

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{12}{L^3} & 0 & -\frac{6}{L^2} & -\frac{12}{L^3} & 0 & -\frac{6}{L^2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L} & 0 & 0 & -\frac{1}{L} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{6}{L^2} & 0 & \frac{4}{L} & \frac{6}{L^2} & 0 & \frac{2}{L} \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{12}{L^3} & 0 & \frac{6}{L^2} & \frac{12}{L^3} & 0 & \frac{6}{L^2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 & \frac{1}{L} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{6}{L^2} & 0 & \frac{2}{L} & \frac{6}{L^2} & 0 & \frac{4}{L}
\end{pmatrix}
$$

The top beam corresponds to this stiffness matrix in the global coordinates:

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 12 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & 6 \\
0 & 6 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -12 & -6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & -6 \\
0 & 6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 4
\end{pmatrix}
$$

Finally, the bottom beam has this stiffness matrix in the global coordinate system:

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 12 & 6 & 0 & -12 & 6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6 & 4 & 0 & -6 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -12 & -6 & 0 & 12 & -6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6 & 2 & 0 & -6 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

For the global system, we can combine all 4 of these element stiffness matrices together and invert this matrix to get K^{-1} (see Appendix A.1). We can then apply

the specified loading to obtain the displacement vector u which has the following form:

$$u = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{L^3}{2(L^3+3L^2+12)} \\ 0 \\ 0 \\ -\frac{L^3+12}{2(L^3+3L^2+12)} \\ \frac{3L}{2(L^3+3L^2+12)} \\ \frac{3L}{L^3+3L^2+12} \\ \frac{6}{(L^3+3L^2+12)} - 1 \\ \frac{3L}{2(L^3+3L^2+12)} \\ \frac{3L}{L^3+3L^2+12} \end{pmatrix}$$

The compliance of the whole structure is then

$$C = 1 - \frac{6}{(L^3 + 3L^2 + 12)} \tag{7.17}$$

We can write down the derivative of the compliance volume product w.r.t. the densities of the top and bottom elements via (7.9).

$$\frac{\partial CV}{\partial x_{\text{top}}} = -\frac{L(2L^6 + 13L^5 + 24L^4 + 33L^3 + 96L^2 + 36L + 72)}{(L^3 + 3L^2 + 12)^2} \tag{7.18}$$

$$\frac{\partial CV}{\partial x_{\text{bot}}} = 1 - \frac{12(2L^3 + 3L^2 + 6L + 12)}{(L^3 + 3L^2 + 12)^2} \tag{7.19}$$

Hence, the linear approximations to CV when we remove the top and bottom beams respectively are:

$$CV - \frac{\partial CV}{\partial x_{\text{top}}} = 4L - \frac{12(3L^4 + 10L^3 + 6L^2 + 30L + 24)}{(L^3 + 3L^2 + 12)^2} + 3 \tag{7.20}$$

$$CV - \frac{\partial CV}{\partial x_{\text{bot}}} = 2L - \frac{12L(L^3 + 2L^2 + 6)}{(L^3 + 3L^2 + 12)^2} + 1 \tag{7.21}$$

It is straightforward to compute the displacement of the structure when either the top or bottom beam is removed. This beam is simply not included in the construction of the global stiffness matrix, and computer algebra software can again find the inverse of the stiffness matrix as a function of the beam length L (see Appendices A.2 and

A.3.)

When we do this, and apply the specified loading, we obtain the displacements as follows:

$$u_{\text{top}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{L^3}{6} \\ 0 \\ \frac{L^2}{2} \\ \frac{L^3}{6} - 1 \\ \frac{L(L+1)}{2} \\ \frac{L(L+2)}{2} \\ -\frac{2L^3}{3} - L^2 - 1 \\ \frac{L(L+1)}{2} \\ L(L+1) \end{pmatrix} \tag{7.22}$$

$$u_{\text{bot}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \tag{7.23}$$

Hence,

$$CV_{\text{top}} = (2L+1)(\frac{2L^3}{3} + L^2 + 1) \tag{7.24}$$

$$CV_{\text{bot}} = 2L + 1 \tag{7.25}$$

Now to understand all the calculations we have made, we look at the difference between the linear approximation to removing each bar and the actual values attained when removing each bar.
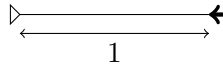
Figure 7-3: A frame consisting of 2 overlapping beams. Both beams are of unit length. The frame is fixed in the left hand side completely and there is a unit load applied horizontally at right free end.

$$\Delta_{\text{top}} := CV_{\text{top}} - (CV - \frac{\partial CV}{\partial x_{\text{top}}}) =$$

$$\frac{L(4L^9 + 32L^8 + 87L^7 + 180L^6 + 465L^5 + 558L^4 + 702L^3 + 936L^2 + 216L + 216)}{3(L^3 + 3L^2 + 12)^2}$$

$$(7.26)$$

$$\Delta_{\text{bot}} := CV_{\text{bot}} - (CV - \frac{\partial CV}{\partial x_{\text{bot}}}) = \frac{12L(L^3 + 2L^2 + 6)}{(L^3 + 3L^2 + 12)^2} \qquad (7.27)$$

We can see that as $L \to \infty$, $\Delta_{\text{top}} \to \infty$ and so for any $M > 0$ we can choose an L such that $\Delta_{\text{top}} > M$.

Note, as $L \to \infty$, $\Delta_{\text{bot}} \to 0$.

$\square$

We can see that as L gets larger, Δ_{top} can get arbitrarily large, and Δ_{bot} can become arbitrarily small. This means that if we remove the bottom bar from Figure 7-2, CV behaves linearly as $L \to \infty$. However if the top beam is removed CV behaves incredibly nonlinearly and as $L \to \infty$ the linear approximation becomes arbitrarily bad.

7.5   Linear behaviour of the elasticity equations

We can find that $\Delta = 0$ in a different system, which does not require us to take the limit. If we consider the system shown in Figure 7-3 where there are 2 beams modelled to occupy the same physical space.

Using the same procedure as above, we find that the stiffness matrix for this system has the form

$$K = \begin{pmatrix} 2 & 0 & 0 & -2 & 0 & 0 \\ 0 & 24 & 12 & 0 & -24 & 12 \\ 0 & 12 & 8 & 0 & -12 & 4 \\ -2 & 0 & 0 & 2 & 0 & 0 \\ 0 & -24 & -12 & 0 & 24 & -12 \\ 0 & 12 & 4 & 0 & -12 & 8 \end{pmatrix}$$

which reduces to the matrix

$$K = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 24 & -12 \\ 0 & -12 & 8 \end{pmatrix}$$

when we fix the system completely in the top left corner. Its inverse is given by

$$K^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{6} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix}.$$

From this we can deduce that the CV of the system is 1, and that the derivative of the CV product with respect to the density of either element is given by $\frac{\partial CV}{\partial x_e} = 0$.

If we compute the CV for the system when one of the bars is completely removed, we find firstly that the stiffness matrix and the inverse of the stiffness matrix have the following formulae:

$$K = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 12 & -6 \\ 0 & -6 & 4 \end{pmatrix}$$

$$K^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 \end{pmatrix}.$$

From these we compute that the CV product of the system with a single bar is 1. Putting this into the definition of Δ when we remove one of these elements gives

$$\Delta = 1 - (1 - 0) = 0$$

In this specific example we find that CV is behaving linearly, regardless of the size of step taken. In this case, the overlapping nature of the elements is similar to the form used in the SIMP method to represent the structure, where a continuous variable can be thought of as representing the number of whole elements present in the structure at that corresponding point.

7.6 A motivating example of nonlinear behaviour in the continuum setting

Consider again the typical example of ESO's convergence behaviour which is shown again in Figure 7-4 with some added annotations.
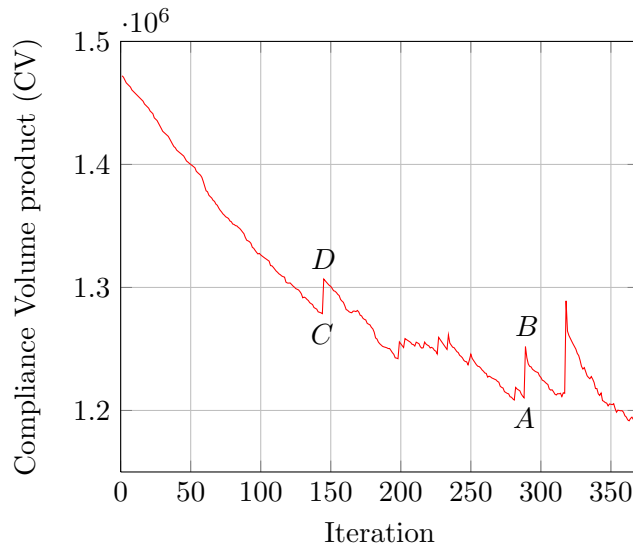
Figure 7-4: Compliance volume (CV) plot for ESO applied to the short cantilevered beam.

If we take a specific look at one of the most notable increases in this graph, when the method goes from iteration 288, point A, to iteration 299, point B, we will see the nonlinearity discussed in the previous section present in this calculation. The structures at these points are shown in Figure 7-5 and Figure 7-6 respectively.

To see what is going on in these different structures, the principal stress vectors are plotted in Figures 7-7 and 7-8. From these figures it can be seen that the small change in the structure, notably the disconnection of one of the "bars", has caused a large redistribution of stresses in the structure. It has also led to the remainder of these "bars" becoming effectively redundant, thus causing a more far-reaching change in the structure, not confined to the local region around the elements that were removed. This is the cause of nonlinear behaviour in the compliance resulting in an increase of the objective function.

In order to picture the nonlinearity we linearly interpolate the densities between the structures A and B. We then plot the corresponding CV product. Note that this type of calculation is not usually done with the ESO method, as it requires elements
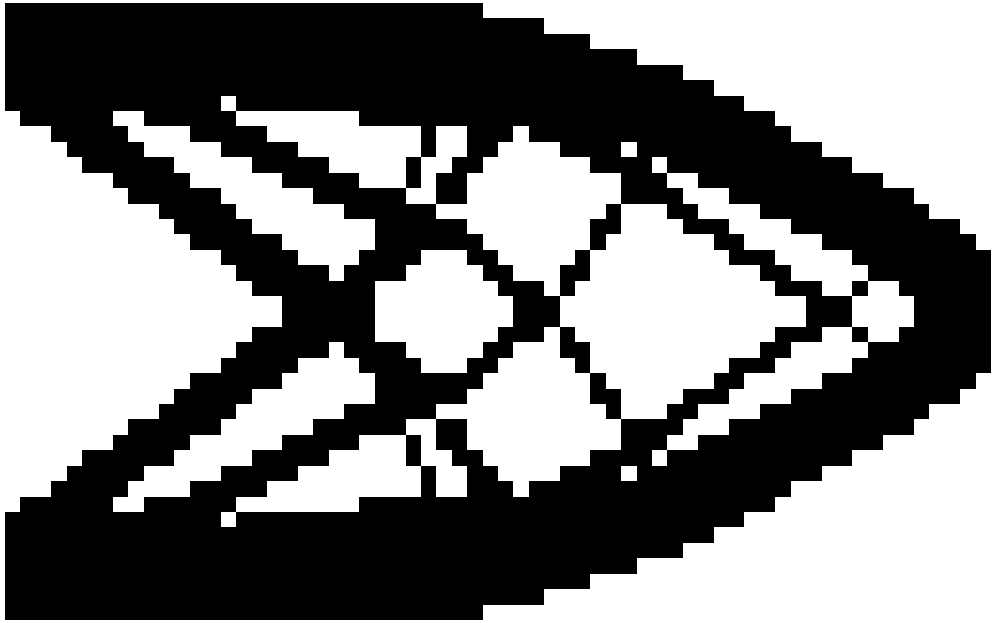
Figure 7-5: Structure at iteration number 288 corresponding to point A of Figure 7-4.

Figure 7-6: Structure at iteration number 289 corresponding to point B of Figure 7-4.
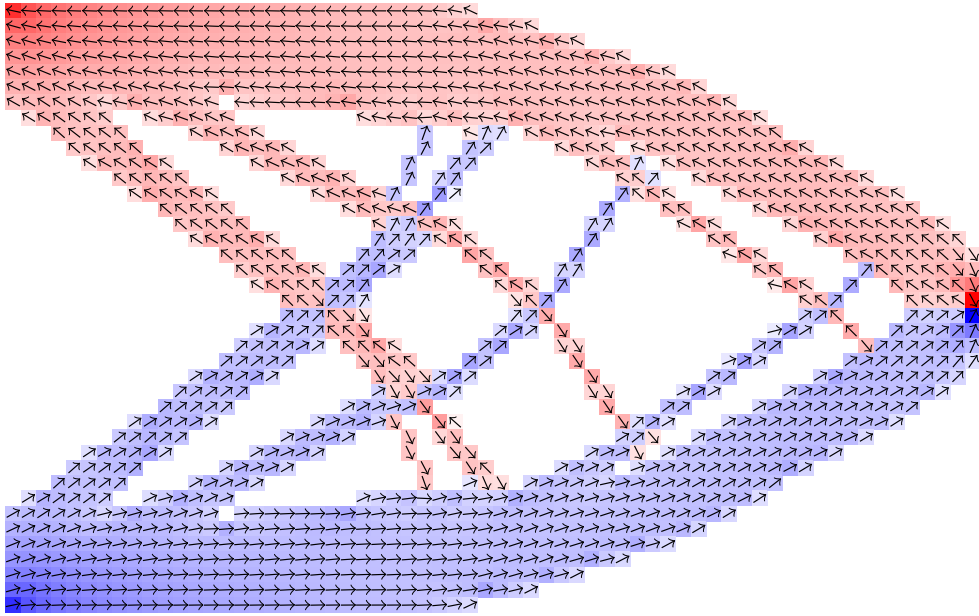
Figure 7-7: Force paths at iteration number 288 corresponding to point A of Figure 7-4. The relative colour intensity denotes the magnitude of the principal stress vector. Red colouring denotes tension and blue colouring denotes compression.
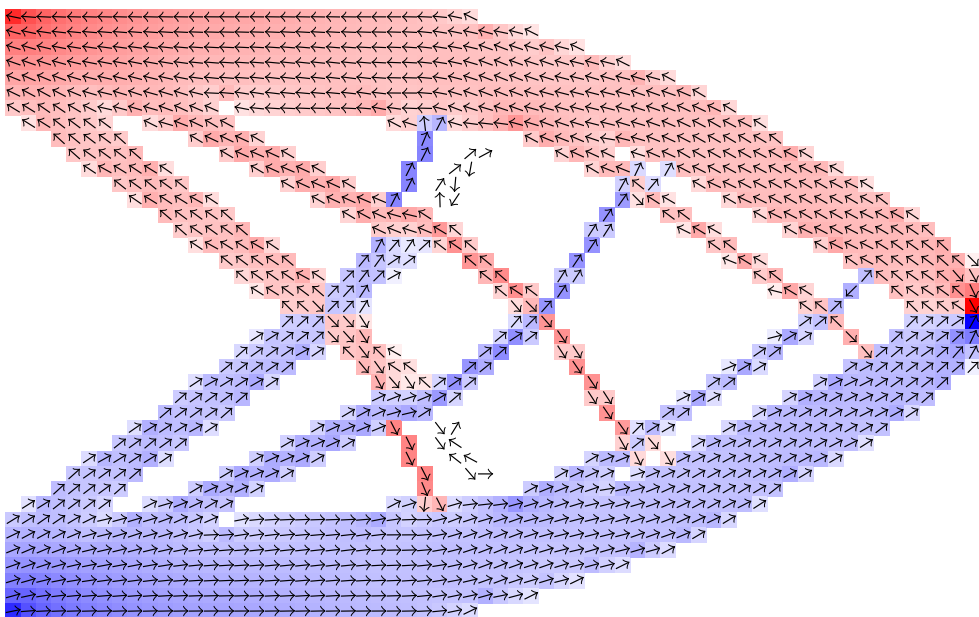

Figure 7-8: Force paths at iteration number 289 corresponding to point B of Figure 7-4. The relative colour intensity denotes the magnitude of the principal stress vector. Red colouring denotes tension and blue colouring denotes compression.

having intermediate density, instead of discrete densities. The plot is given in Figure 7-9.
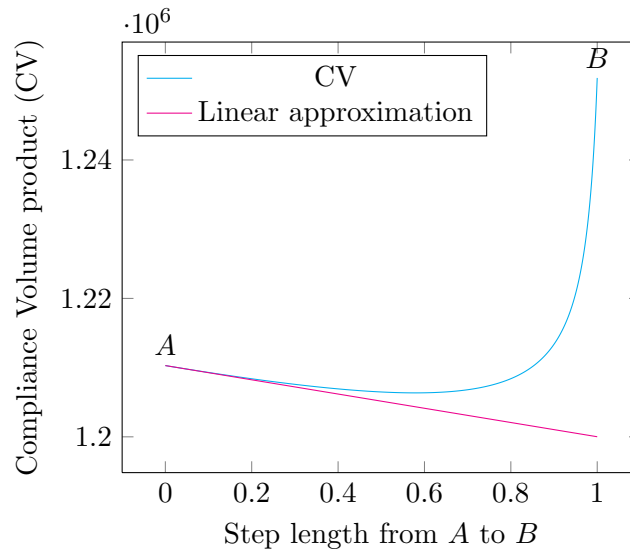
Figure 7-9: Compliance volume (CV) plot as we interpolate the structures from iteration 288 to 289.

There are a number of things that we see from this graph. Firstly, the direction in which we move is indeed a descent direction. The optimal step length would be around 0.58 times the actual unit step taken. The other, more notable thing is that this graph is nonlinear. It is this nonlinearity which causes the objective history of ESO to jump, i.e. have non-monotonic convergence. In fact, if the step length is more than 0.853 then CV increases.

This increase in the objective function occurs as one of the connections in the continuum structure is broken, leading to a marked topological change in the structure. We will now look at the jump which occurs when we go from points C to D in Figure 7-4.

The structures at points C and D in Figure 7-4 are shown in Figures 7-10 and 7-11 respectively. Interpolating the density of material in the same manner as Figure 7-9 leads to the plot shown in Figure 7-12. In this case, there is no connection which is being broken, as we previously had in Figure 7-9. There is however, the same nonlinear behaviour.

From this we can see that the step size taken by ESO is too large. This is equivalent to the step size being too large in a line search optimization method. If ESO had the ability to choose a smaller step then it may not exhibit this non-monotonic convergence
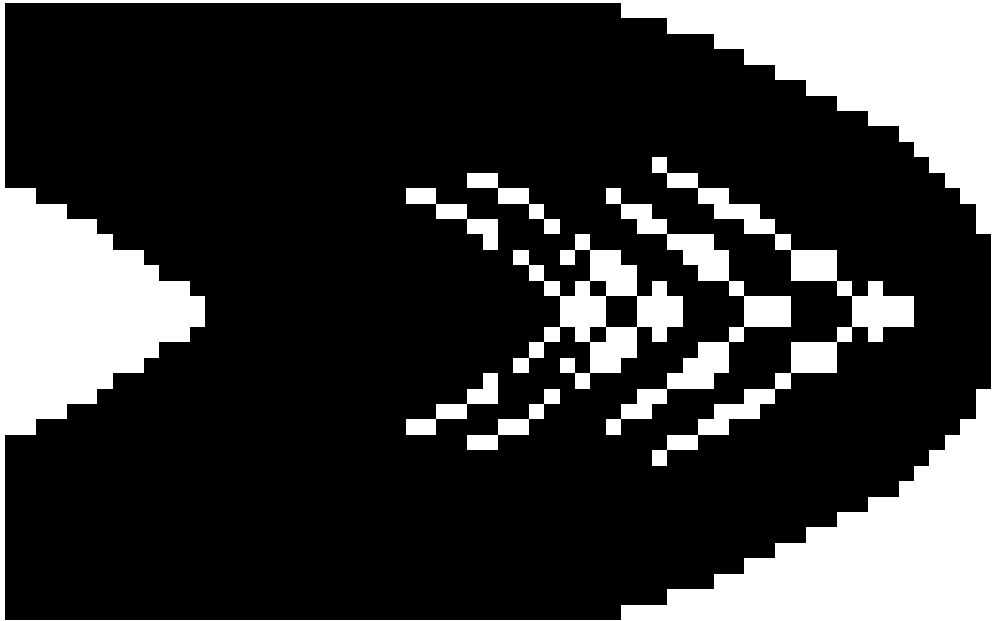
Figure 7-10: Structure at iteration number 144 corresponding to point C of Figure 7-4.
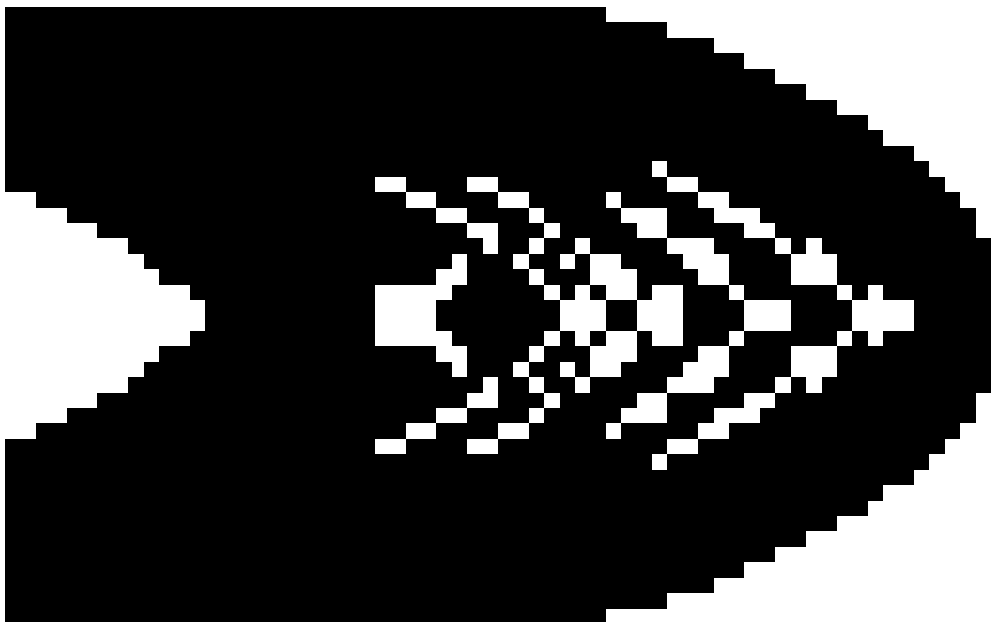
Figure 7-11: Structure at iteration number 145 corresponding to point D of Figure 7-4.
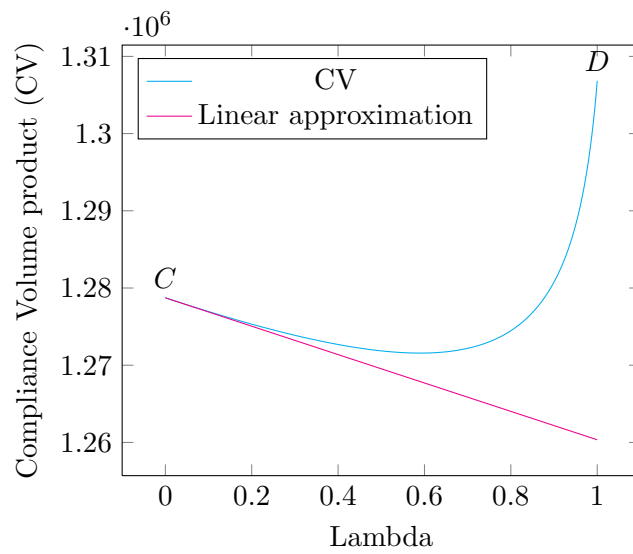
Figure 7-12: Compliance volume (CV) plot as we interpolate the structures from iteration 144 to 145.

behaviour. In the following section we introduce a change to the ESO algorithm to allow it to automatically take a smaller step size.

7.7   ESO with h-refinement

We have seen that due to the nonlinearity inherent in the equations of elasticity, ESO will often take too large a step and this will cause the objective function to increase. We now modify the ESO algorithm so that if it is allowed to take a smaller step then we hope to see the objective function decrease monotonically. The simple modification consists of checking that the objective function has not increased, but if it has, instead of removing the elements we instead refine them and continue the ESO process.

There are 3 general types of mesh refinement, r-, p- and h-refinement. r-refinement, or relocation refinement, is the least noted of these in which the location of the mesh connections are moved to areas of interest. p-refinement works by varying the order of the polynomial basis functions in the underlying finite-element discretisation of the problem. The goal in our case with mesh refinement is to have a more detailed representation of the domain of the structure. As such, p-refinement is not suitable as it does not change how the domain is represented. r-refinement would also lead to difficulties as it would, by definition, relocate parts of the domain and so great care and complications would be needed to represent one structure on a mesh that has been moved in space.

h-refinement by contrast simply works by recursively dividing elements into smaller ones. Hence any structure represented on a coarse mesh can be exactly represented on a h-refined mesh. This type of adaptivity is therefore ideal to allow the ESO method to take a smaller optimization step.

Mesh refinement has been previously employed in topology optimization for structural problems. For instance in the SIMP approach, Maute and Ramm (1995) [110] employed mesh refinement in order better represent structural boundaries and Stainko (2006) [156] used adaptive global and local h-refinement in in order to improve computational efficiency. In the ESO method, global mesh refinement has been used by Akin and Arjona-Baez (2001) [14] in order to control the finite-element error in the structural analysis. Huang and Xie (2007) [72] used a posteriori global mesh refinement with the ESO method in order to avoid local minima, but this is only achieved by enforcing the maintenance of boundary conditions.

The adapted ESO method is described in Algorithm 4 as uses h-refinement with the sole aim to remove the non-monotonic convergence behaviour of the ESO algorithm. In the results shown, ESO with h-refinement was implemented in Ansys, a commercial finite element package. This has the feature which allows a mesh to be automatically refined in given elements. This was applied to the short cantilevered beam problem considered previously in Figure 7-1 and results are shown in Figures 7-13 and 7-14.

Algorithm 4 Evolutionary Structural Optimization with h-refinement

---

1: Mesh design domain
2: Define a rejection ratio RR
3: **loop**
4:     Perform structural analysis of structure
5:     **if** Objective has increased **then**
6:         Reinstate removed elements and refine them
7:     **else**
8:         Calculate elemental sensitivities s_i for all elements i
9:         {Filter sensitivities (optional)}
10:        Remove elements i with $s_i \leq RR \min_j \{s_j\}$
11:     **end if**
12: **end loop**
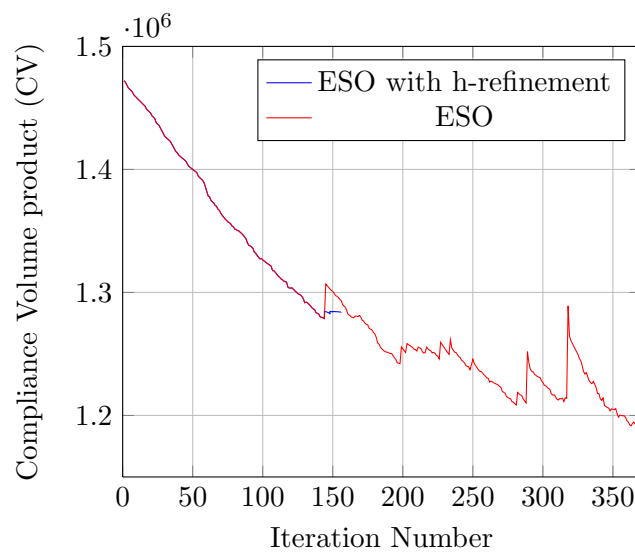
Figure 7-13: Convergence of ESO with h-refinement applied to the short cantilevered beam
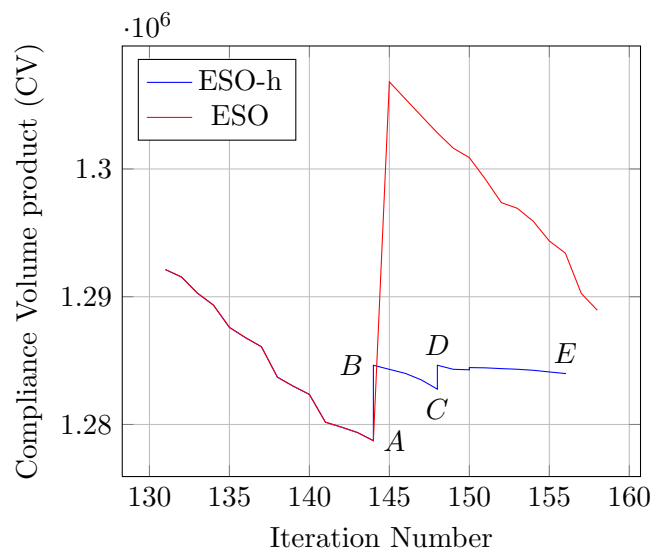
Figure 7-14: Magnified view of convergence of ESO with h-refinement applied to the short cantilevered beam

As we can see from Figures 7-13 and 7-14, the ESO algorithm with h-refinement is identical to the original ESO algorithm until the objective function increases. At that point, the algorithm refines the mesh and the ESO algorithm continues. The meshes used at points A, B, C, D and E are shown in Figures 7-15, 7-16, 7-17, 7-18 and 7-19 respectively.

Each time we refine, i.e. go from points A to B and points C to D, the CV of the structure increases. More specifically, the volume remains exactly the same as the structure has not changed, only the mesh describing it. The compliance increases as the refined mesh can more accurately resolve the gradients of the stress field. These increases are thus not due to the optimization, but rather caused by the more accurate representation of the structure. Appendix B shows a mesh refinement study which shows and explains this behaviour.

Following from where the mesh is refined, one can then see that ESO automatically continues to improve the objective function. It does this by choosing to take a smaller size (i.e. remove a smaller amount of volume of the structure) and the nonlinearity of the compliance does not adversely affect the convergence.

This method stops when the stiffness matrix K describing the structure becomes singular (as measured by the linear solver). This is the same criteria used to stop the original ESO method.

It is possible to introduce an actual stopping criterion for use in the ESO with h-refinement algorithm. That is if, for some given value $\mathtt{tol}$ for which $0 \leq \mathtt{tol} < 1$

$$\sum_e \frac{\partial CV}{\partial x_e}(x^k) \leq \mathtt{tol} CV \qquad (7.28)$$

then stop and we would consider the current point x^k to be a local minima of the problem. We apply this stopping criterion in the following section.
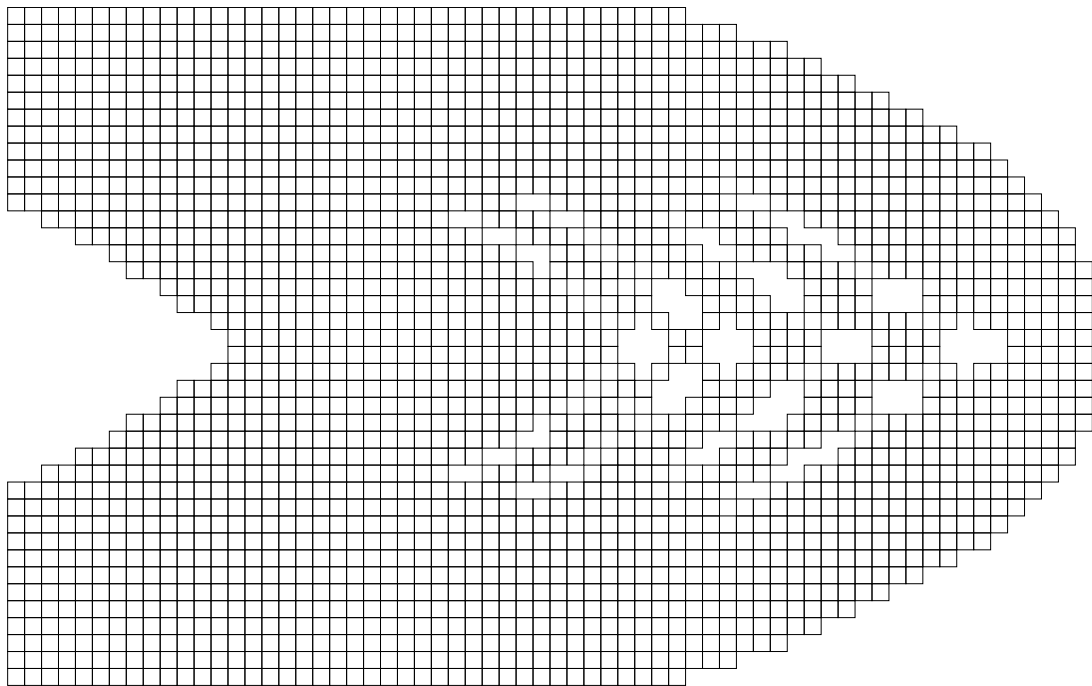
Figure 7-15: The mesh after 144 iterations of both the ESO algorithm and the ESO with h-refinement when applied to the short cantilevered beam. This corresponds to point A in Figure 7-14.
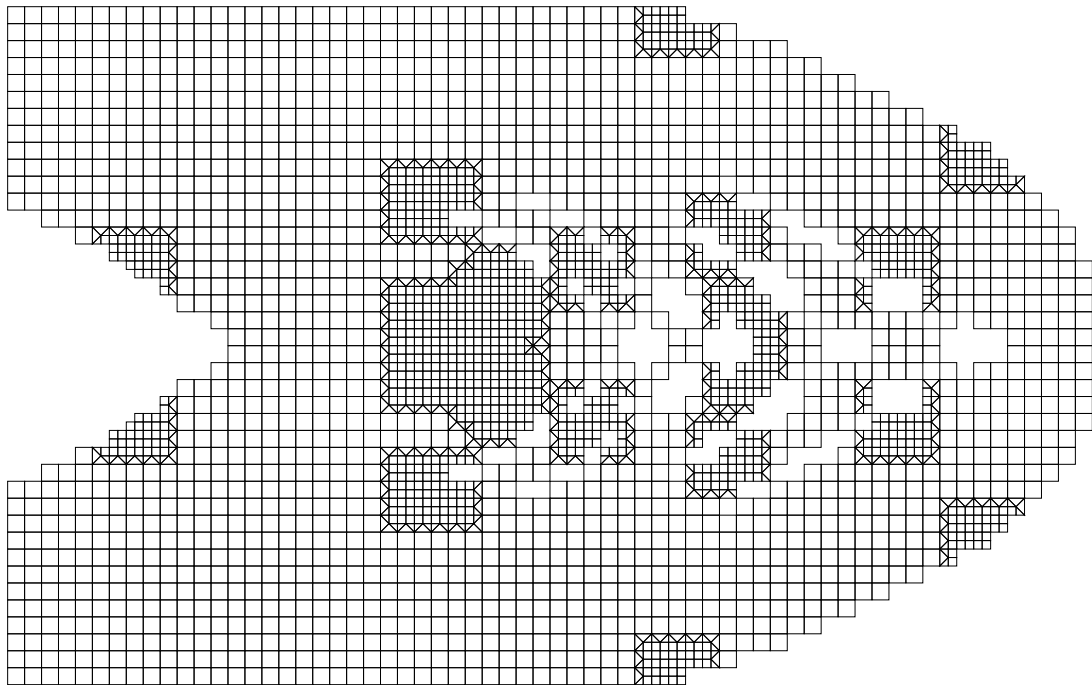
Figure 7-16: The mesh after being refined from the mesh shown in Figure 7-15. This corresponds to point B in Figure 7-14.
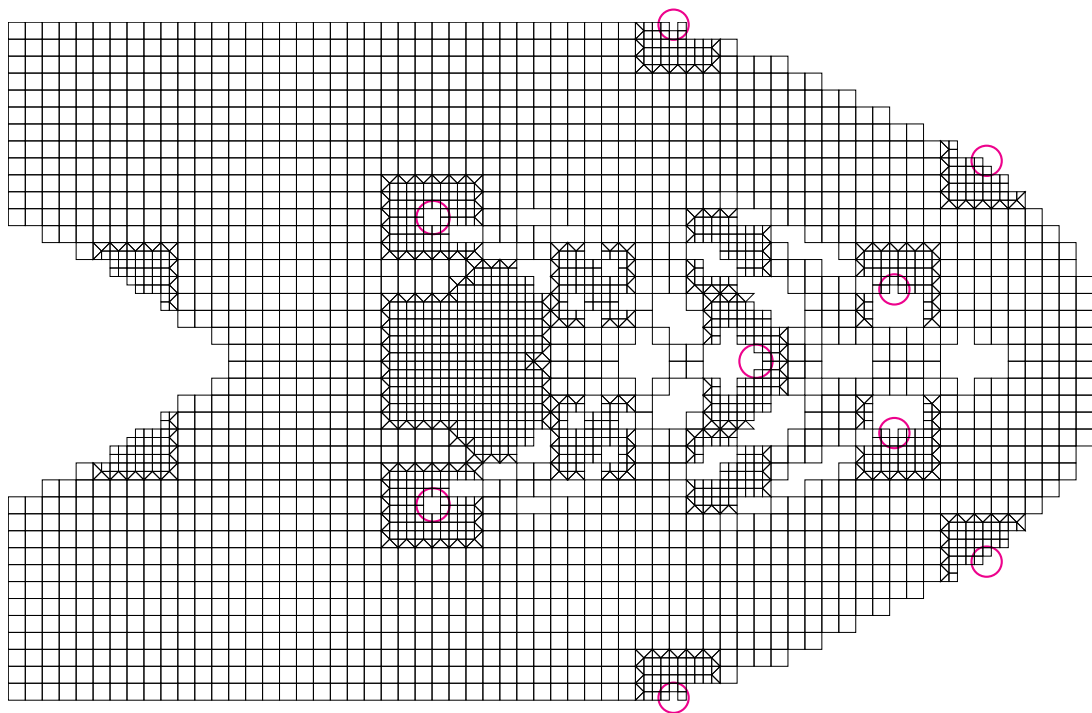

Figure 7-17: The mesh at point C of Figure 7-14. The elements which have been removed from since this mesh was generated at point B have been highlighted.
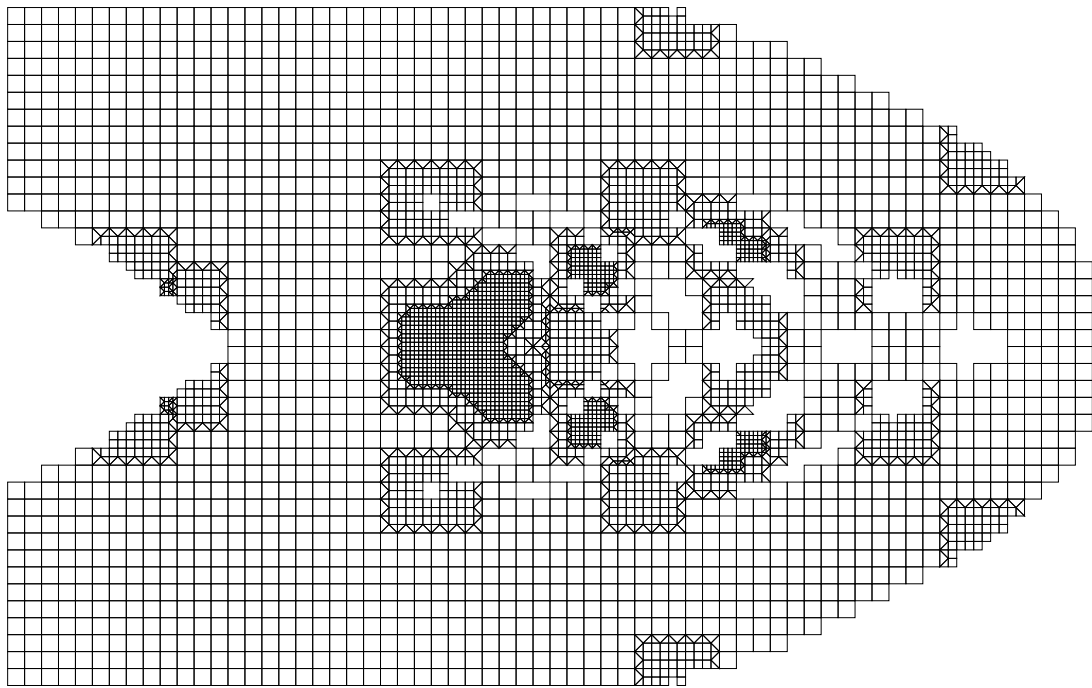
Figure 7-18: The mesh at point D of Figure 7-14 that results from Figure 7-17 being refined.
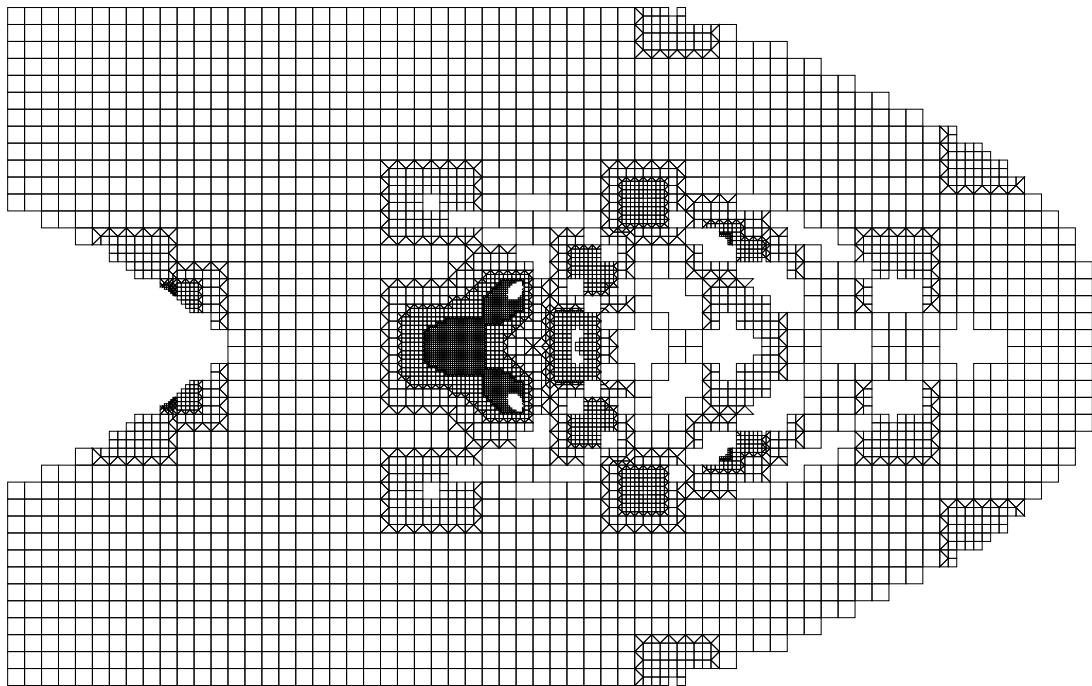
Figure 7-19: The final mesh coming from the ESO with h-refinement algorithm applied to the short cantilevered beam.

7.8   Tie-beam with h-refinement

The "tie-beam" was introduced by Zhou and Rozvany in 2001 [187] and is a notoriously difficult topology optimization problem. The design domain is shown in Figure 7-20 and consists of 100 elements, where there is a *tie* connecting what would be a cantilevered beam to a roller support on a fixed ceiling. The loading in the horizontal direction is 3 times the magnitude of the loading in the vertical direction.
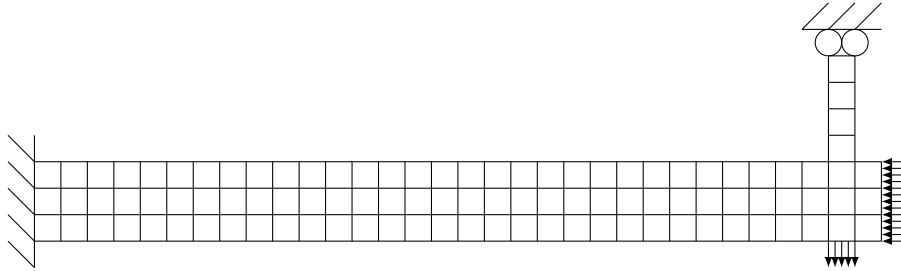
Figure 7-20: Tie-beam problem as stated by Zhou and Rozvany [187].

The global solution to the problem as stated by Zhou and Rozvany was given in 2010 by Stolpe and Bendsøe [162]. In order to compute the global solution they had to resort to using branch-and-cut methods and a great deal of patience (over a week of CPU time to find the minimal compliance for a given volume structure).

The methods used by Stolpe and Bendsøe were generic optimization methods which are unsuitable for more realistic large-scale topology optimization problems. The computational cost of the branch-and-cut methods is far too high to deal with problems that have substantially more variables, such as those we have seen in Chapters 5 and 6.

When ESO is applied to the tie-beam, the structure with minimal objective function is the structure given in the initial configuration. The objective function history is shown in Figure 7-21. In the initial step in the ESO process the *tie* connecting the main structure to the ceiling is cut, and the objective increases dramatically, resulting in a highly non-optimal structure.

Applying the ESO with h-refinement algorithm to this problem does not behave in the same way. Instead of cutting the tie, the algorithm instead performs a local refinement of the mesh in this region. In doing so, ESO with h-refinement is able to find a structure that has a lower objective function than the initial configuration, and hence better than the solution found by the basic ESO algorithm. The objective function history is shown in Figure 7-22.

The meshes automatically generated are shown in Figures 7-23 to 7-27 and the
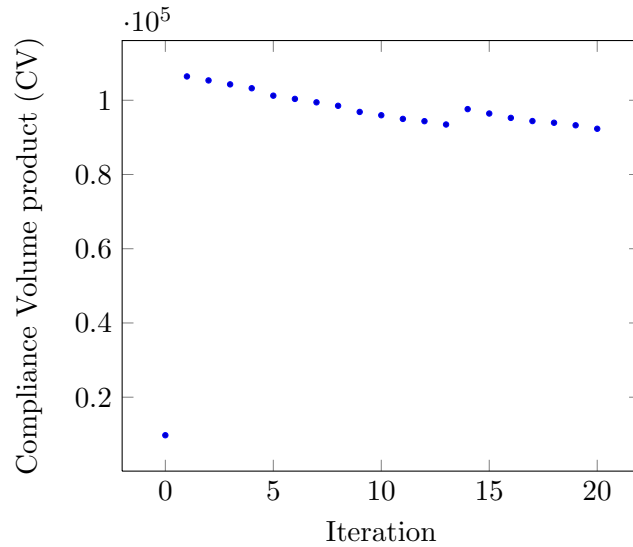
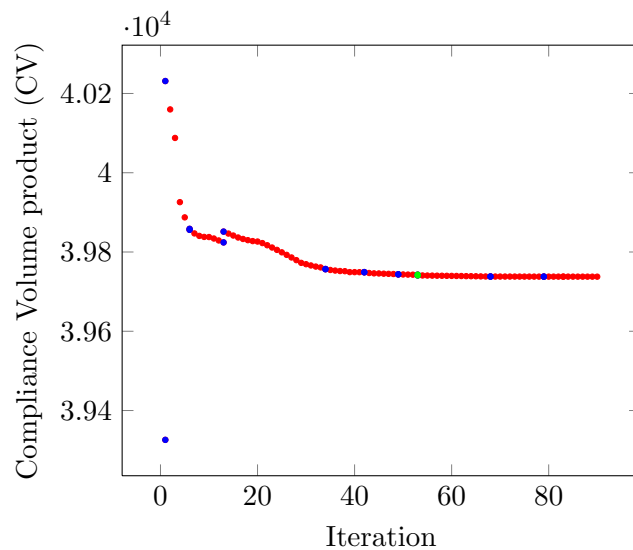Figure 7-21: ESO objective function history for the tie-beam problem.

Figure 7-22: Compliance volume (CV) plot for ESO with H-refinement applied to the short cantilevered beam. Note again that the increases in the objective function are caused only by refining the mesh to get a more accurate resolution of the structure, rather than changing the structure itself. These increases are marked in blue. One instance of refinement decreasing the objective function is seen and marked in green. Red colours represent the progress of ESO without changing the mesh in that step.
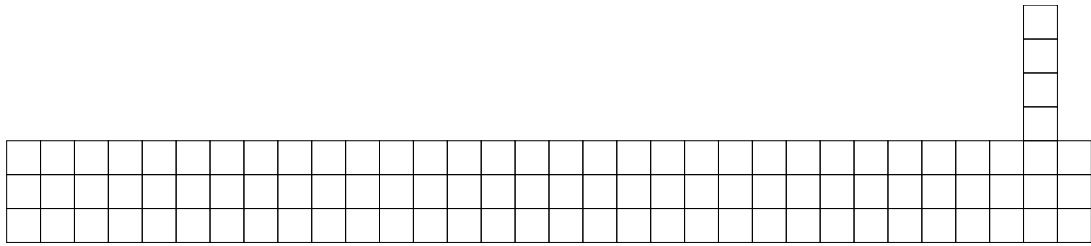
Figure 7-23: Initial mesh from ESO with h-refinement applied to the tie-beam problem.
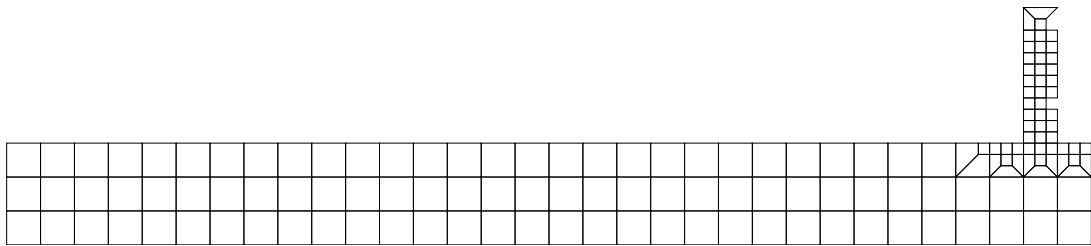

Figure 7-24: First mesh showing h-refinement from ESO with h-refinement from the tie-beam problem.
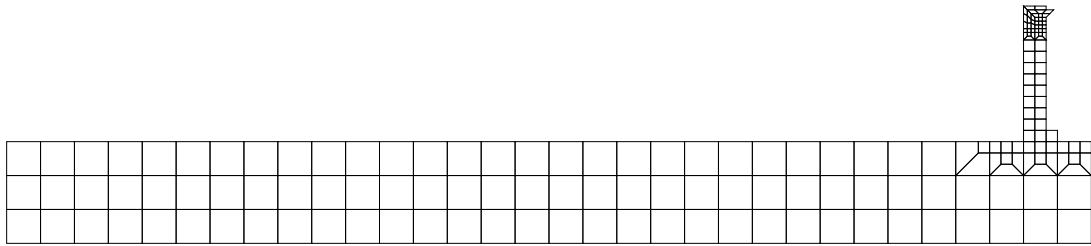

Figure 7-25: Mesh showing 2 levels of refinement from ESO with h-refinement from the tie-beam problem.
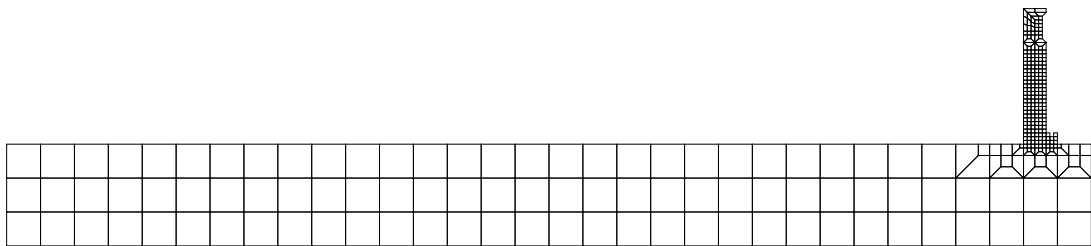

Figure 7-26: Mesh showing refinement in a different position from ESO with h-refinement from the tie-beam problem.
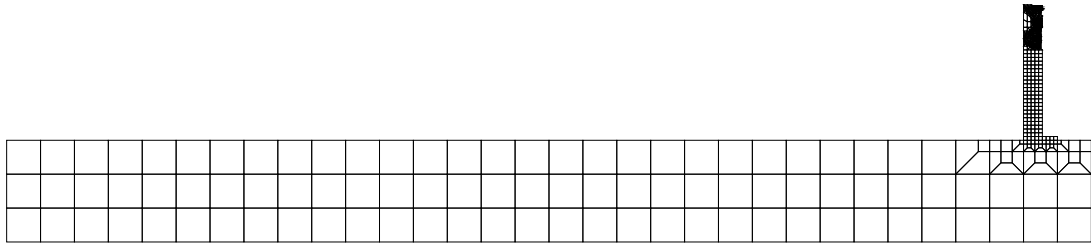
Figure 7-27: Final mesh from ESO with h-refinement applied to the tie-beam problem.

Figure 7-28: Final structure given by ESO with h-refinement applied to the tie-beam problem.

final structure shown in Figure 7-28. Notice that we start with a uniform mesh as depicted in Figure 7-23. After one iteration, the mesh has been refined and two of the smaller refined elements have been removed. This is shown in Figure 7-24. Note that the refinement process used has introduced non-rectangular elements in order to avoid *hanging nodes*.

In the sixth iteration the structure has been refined again and is shown in Figure 7-25. The next refinement occurs on the thirteenth iteration occurs in a different part of the structure compared to the refinement in the sixth iteration. This is shown in Figure 7-26. The final mesh is shown in Figure 7-27. Notice that all of the refinement of the mesh has occurred around the tie. This allows the method accurately represent a structure with a thinner tie that was in the original problem statement.

In this example we used the convergence criterion set out in (7.28) with $\texttt{tol} = 10^{-8}$. We plot the values of $\frac{\partial CV}{\partial x_e}(x^k)/CV$ in Figure 7-29. As this quantity approaches 0, the structure is converging to a stationary point. Hence ESO with h-refinement is approaching a local minimum of the unconstrained continuous optimization problem.
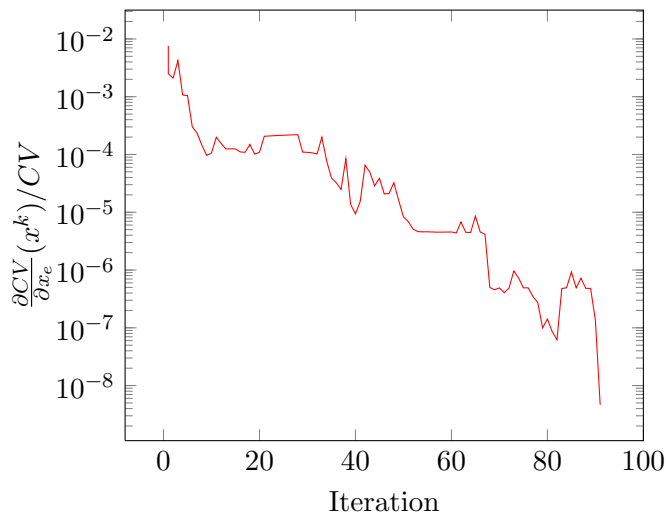
Figure 7-29: Convergence criteria – iterations of eso with h-refinement applied to the tie-beam

7.9 ESO as a stochastic optimization algorithm

The tie-beam is a particularly difficult optimization problem, with many methods failing to find a solution better than the initial state. When ESO is applied to this problem it clearly is amongst those methods that are not good in this instance. To its credit however, if one refers back to the original objective function history shown in this chapter resulting from the ESO algorithm applied to the short cantilevered beam (Figure 7-1) it can be seen that ESO finds multiple solutions which appear close to distinct local minima.

As ESO progresses it is able to leave local minima and (in this case) find a better solution than the first local minima it exposes. Due to this behaviour of ESO, the solution it finds that has an objective function of around 1.192×10^6 is considerably better than the local solution found by ESO with h-refinement that has an objective function of around 1.285×10^6. In this way, as ESO takes some steps which increase the objective function, it is similar to stochastic methods of optimization such as simulated annealing (see for example Kirkpatrick, Gelatt and Vecchi 1983 [88], Černý 1985 [175] or Aarts and Korst 1989 [2]).

The tie-beam example shows us that the ESO solution is not guaranteed to be the global solution of the problem. It is possible to combine the ESO and ESO with h-refinement methods in order to obtain multiple local minima for the same problem. When ESO chooses a step which increases the objective function, the method can be branched so that ESO with h-refinement finds the local minima around that point, but

ESO continues to search along the same path for other solutions closer to the global optimum.

7.10 Conclusions

1. The nonmonotonic convergence behaviour of ESO can be explained by the fact that the underlying state equations of linear elasticity are nonlinear with respect to varying the domain of the problem. This nonlinearity can lead to ESO taking a step which increases the objective function which is sometimes catastrophic for the quality of solution which ESO finds.

2. When ESO does increase the objective function of the solution, this is equivalent to taking too large a step in line search method.

3. ESO can display descent if the elements that is selects for removal subsequently cause the objective to increase are refined as opposed to removed. ESO then naturally chooses a smaller step length in the line search and this then leads to descent to a point which approximates a stationary point in unconstrained optimization.

4. As ESO with h-refinement can now be shown to approximate a stationary point of an unconstrained optimization problem, it has a much more sound theoretical background.

8

Conclusions and future work

8.1 Achievements of the thesis

The key findings and developments of this thesis are:

- The SAND formulation of minimisation of compliance problem subject to a volume constraint violates the MFCQ and as a result SQP methods struggle to find feasible solutions.

This result makes it undesirable to use a SAND formulation for topology optimization problems as its disadvantages outweigh its potential benefits.

- In a NAND formulation of minimisation of compliance problem subject to a volume constraint, filtering provides an excellent way to regularise the problem and remove chequerboard patterns, though impairs the convergence of the problem.

Using a low-pass filter to remove the high-frequency variation in derivative values is a simple and effective way of imposing a minimum length scale on the topology optimization problem. This minimum length scale can be defined a priori and so is preferable to a perimeter constraint where the maximum allowable perimeter of a structure is generally uncertain. The filtering also keeps the solution away from many local minima, as the perturbation to the true gradients does not allow the solution to fall into the local minima.

- A robust criterion for detecting when to stop filtering the problem has been developed and shown to work well on very high resolution test problems.

When the objective function stops decreasing by any meaningful amount, and the chequerboard pattern has been avoided, it is desirable to converge to a KKT point.

The filtering scheme may avoid many local minima, but cannot be guaranteed to find the global minima. Hence providing the optimization routine with the correct gradients allows the solution to fall into a nearby optima and thus using a cessant filter keeps the regularising properties of the filter and the local convergence properties of the unperturbed optimization routine.

- Spurious localised buckling modes have been observed and proven to be eradicated by setting the contributions to the stress stiffness matrix from low density elements to zero, though this is not consistent with the underlying equations.

These unwanted numerical features arise due to the representation of the structure using continuous variables. Any continuous optimization approach to topology optimization involving buckling or harmonic modes will exhibit this characteristic and so care must be taken to ensure the analysis is performed accurately.

- An analytic expression for the derivative of the stress stiffness matrix with respect to the density of an element has been presented.

Often, when performing optimization involving harmonic modes, the stress stiffness matrix is considered *similar* to the mass matrix. At first glance this appears reasonable as they appear in the same place in a generalised eigenvalue problem and have the same sparsity structure. However the construction of a mass stiffness matrix is a forward problem, whereas the construction of the stress stiffness matrix involves the solution of an inverse problem. As such computing the derivative of the stress stiffness matrix is by no means trivial. The analytic expression for this allows it to be computed efficiently but its complexity means it remains an expensive step in an overall computation.

- A new optimization method has been developed specifically for the minimisation of weight subject to a volume constraint and a buckling constraint in order to minimise the number of derivative calculations needed and to avoid the problem of computing spurious localised buckling modes.

This method is designed to provide an efficient technique for a topology optimization problem with buckling as a constraint. It has been developed due to the difficulties associated with existing methods and has been shown to scale well up to large problem sizes of use in practical applications.

- Singularities at a re-entrant corner occur naturally in the equations of linear elasticity and are not simply a numerical error.

In an element based formulation of a topology optimization problem, re-entrant corners are an inevitable feature in an optimal design and are most pronounced in the ESO approach to topology optimization. Only in the case of a stress constrained problem may this present an issue and so may require special care.

- The nonmonotonic convergence behaviour of ESO has been explained by observing the nonlinear behaviour of the underlying state equations of linear elasticity with respect to varying the domain of the problem.

The observation that the ESO uses infinitesimal information to determine the direction of a unit step is crucial to understanding the ESO algorithm. The analytic examples of the values of the objective function deviating from their linear approximation based on that infinitesimal information show the nonlinear behaviour that is not considered by the ESO method. This observation shows that whilst the change to the structure may be relatively very small in terms of volume, it is still a unit step in the infinity-norm and can have a drastic effect on the behaviour of the structure.

- ESO with h-refinement has been observed to approximate a stationary point of an unconstrained optimization problem and thus give a much more sound theoretical background to ESO.

Building on the previous observation, the natural manner to validate ESO as an optimization method was to allow it to take a smaller step. The simple addition of h-refinement to the ESO algorithm achieved this and allowed the modified ESO method to exhibit monotonic convergence. In combination with the previous observation, many of the questions regarding the convergence of ESO have been answered.

8.2 Application of the results of the thesis and concluding remarks

This thesis has been a mathematical exploration of a problem which is very much of interest to mechanical, civil and aeronautical engineers. While technical details have been the main focus of this thesis, how the problem is formulated is the most important feature of solving the problem and underpins the statements which can be made about the resulting solution.

If the problem is unconstrained then the ESO method may provide a quick way of searching through the design space which may easily escape local minima. The choice of sensitivity measure should be based on the gradient of the objective function,

and not some other physical quantity. However the result given from ESO should not be considered a local minimum but should give a good starting point for another optimization algorithm, such as ESO with h-refinement or a SIMP approach.

The most robust technique available for solving a topology optimization problem is to use a SIMP approach and a mathematical programming technique. These allow for constraints on the system in the way that a method like ESO do not. Provided the objective and the constraints remain differentiable, and the underlying equations do not exhibit unwanted numerical features when low density elements are represented, then the SIMP approach will give a solution for which local optimality can be claimed.

This thesis has discussed in detail the difficulties associated with including the solution of the underlying state equations in the formulation of the optimization problem. The advantages of *not* including them in the problem should be highlighted. Whilst it reduces the number of optimization variables, more importantly to an engineer, it provides meaningful quantities about the solution at all times throughout the optimization process. That is to say, given any solution, the state equations can be solved and so can be interpreted physically.

Removing solutions of the state equations from the formulation and adopting a NAND approach also allows for dedicated PDE solvers to be employed. This transfers all the difficulty of solving the PDE, and hence finding a feasible solution, to a code which may have been optimized for such a purpose.

The fast binary descent method can be used for the specific problem it was developed for, or any other problem where derivative calculations are very expensive. Its derivation may be used as a model to build a different optimization algorithm if certain aspects of a SIMP approach do not lend them selves to being solved efficiently.

ESO with h-refinement can be used to further investigate the ESO method and can validate the solutions given by ESO.

8.3 Future work

There are many different avenues for future work in topology optimization which could be explored. For instance, the mesh refinement techniques used in the ESO-h could be investigated to see if the resulting structure could be independent of the mesh refinement technique employed. It should also be incorporated into Bidirectional Evolutionary Structural Optimization (BESO) to assess the optimality of structures produced by the BESO method. As a like for like comparison, the structures found by the ESO-h algorithm should be computed using the final refined mesh so that the effect of refinement is removed from the analysis of the method.
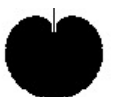
Most work done to date on comparing iterative and direct solvers for topology optimization has focused on having one linear solve for each optimization step. Fully coupling an iterative solver with the optimization process could be investigated, so that the equilibrium equations of elasticity are only satisfied to a very tight tolerance when the optimal structure is found. This type of approach could significantly improve the efficiency of the method.

If an iterative solver is used, preconditioning for these problems could be further investigated, making full use of the knowledge of the problem at previous optimization iterations. This would lead nicely into considering fully non-linear elastic material. The optimization process and convergence when effects such as contact are included are not yet fully understood.

Further, to apply topology optimization in other situations, the methods needed to optimize coupled systems such as electro-mechanical systems or fluid structure interactions should be investigated. Issues such as the variables to be included in the optimization formulation and possible iterative optimization schemes between the different systems is a rich area of research with promising impact and applications.
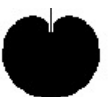
Stiffness matrices of few bar structures

A.1 Stiffness matrix and inverse of 4 bar structure.

Here is stated the stiffness matrix and its inverse for the structure considered in Figure 7-2 comprised of 4 beams.

$$
K = \begin{pmatrix}
\frac{12}{L^3}+1 & 0 & -\frac{6}{L^2} & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{L}+12 & 6 & 0 & -12 & 6 & 0 & 0 & 0 \\
-\frac{6}{L^2} & 6 & \frac{4}{L}+4 & 0 & -6 & 2 & 0 & 0 & 0 \\
-1 & 0 & 0 & \frac{12}{L^3}+1 & 0 & -\frac{6}{L^2} & -\frac{12}{L^3} & 0 & -\frac{6}{L^2} \\
0 & -12 & -6 & 0 & \frac{1}{L}+12 & -6 & 0 & -\frac{1}{L} & 0 \\
0 & 6 & 2 & -\frac{6}{L^2} & -6 & \frac{4}{L}+4 & \frac{6}{L^2} & 0 & \frac{2}{L} \\
0 & 0 & 0 & -\frac{12}{L^3} & 0 & \frac{6}{L^2} & \frac{12}{L^3}+1 & 0 & \frac{6}{L^2} \\
0 & 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 & \frac{1}{L}+12 & -6 \\
0 & 0 & 0 & -\frac{6}{L^2} & 0 & \frac{2}{L} & \frac{6}{L^2} & -6 & \frac{4}{L}+4
\end{pmatrix}
$$

As the inverse of this matrix is dense and symmetric, we list only the lower triangular part of the inverse, giving each column separately in equations (A.1) to (A.9).

$$K^{-1}(1:9,1) = \begin{pmatrix} \frac{L^3(51L^5+298L^4+355L^3+816L^2+1392L+96)}{24(15L+1)(L+1)(L^3+3L^2+12)} \\ -\frac{3L^3}{30L+2} \\ \frac{L^2(3L^2+7L+\frac{1}{2})}{(15L+1)(L+1)} \\ \frac{L^3(51L^5+298L^4+355L^3+636L^2+1200L+84)}{24(15L+1)(L+1)(L^3+3L^2+12)} \\ \frac{L^2(12L^5+79L^4+102L^3+151L^2+336L+24)}{8(15L+1)(L+1)(L^3+3L^2+12)} \\ \frac{L^2(3L^4+13L^3+L^2+36L+3)}{(15L+1)(L^3+3L^2+12)} \\ \frac{L^3}{2(L^3+3L^2+12)} \\ \frac{L^2(31L^4+66L^3+7L^2+192L+24)}{8(15L+1)(L+1)(L^3+3L^2+12)} \\ \frac{L^2(L^3+2L^2+6)}{2(L+1)(L^3+3L^2+12)} \end{pmatrix} \tag{A.1}$$

$$K^{-1}(2:9,2) = \begin{pmatrix} \frac{L(9L+1)}{15L+1} \\ -\frac{3L^2}{15L+1} \\ -\frac{3L^3}{30L+2} \\ \frac{L(12L+1)}{30L+2} \\ -\frac{3L^2}{15L+1} \\ 0 \\ \frac{1}{30}-\frac{1}{(450L+30)} \\ 0 \end{pmatrix} \tag{A.2}$$

$$K^{-1}(3:9,3) = \begin{pmatrix} \frac{L(6L^2+14L+1)}{(15L+1)(L+1)} \\ \frac{L^2(3L^2+7L+\frac{1}{2})}{(15L+1)(L+1)} \\ \frac{L(6L^2+14L+1)}{2(15L+1)(L+1)} \\ \frac{L(12L+1)}{30L+2} \\ 0 \\ \frac{L(8L+1)}{2(15L+1)(L+1)} \\ \frac{1}{2}-\frac{1}{2(L+1)} \end{pmatrix} \tag{A.3}$$

$$K^{-1}(4:9,4) = \begin{pmatrix} \frac{\frac{17L^8}{8}+\frac{149L^7}{12}+\frac{355L^6}{24}+34L^5+103L^4+52L^3+93L^2+96L+6}{(15L+1)(L+1)(L^3+3L^2+12)} \\ \frac{L(12L^6+79L^5+102L^4+151L^3+516L^2+216L+12)}{8(15L+1)(L+1)(L^3+3L^2+12)} \\ \frac{L(L+1)(3L^4+10L^3-9L^2+45L+3)}{(15L+1)(L^3+3L^2+12)} \\ \frac{L^3+12}{2(L^3+3L^2+12)} \\ \frac{L(31L^5+66L^4+7L^3+372L^2+216L+12)}{8(15L+1)(L+1)(L^3+3L^2+12)} \\ \frac{L(L^4+2L^3+12L+6)}{2(L+1)(L^3+3L^2+12)} \end{pmatrix} \tag{A.4}$$
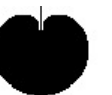
$$K^{-1}(5:9,5) = \begin{pmatrix} \frac{3L}{5} - \frac{1}{8(L+1)} - \frac{1}{600(15L+1)} - \frac{3L^2}{8(L^3+3L^2+12)} + \frac{22}{75} \\ \frac{(L+1)(3L^4+10L^3-9L^2+45L+3)}{(15L+1)(L^3+3L^2+12)} \\ -\frac{3L}{2(L^3+3L^2+12)} \\ -\frac{1}{8(L+1)} - \frac{1}{120(15L+1)} - \frac{3L^2}{8(L^3+3L^2+12)} + \frac{3}{10} \\ -\frac{(L^3+\frac{3L^2}{2}+3)}{(L+1)(L^3+3L^2+12)} + \frac{1}{2} \end{pmatrix} \tag{A.5}$$

$$K^{-1}(6:9,6) = \begin{pmatrix} \frac{2(L+1)(3L^4+10L^3-9L^2+45L+3)}{(15L+1)(L^3+3L^2+12)} \\ -\frac{3L}{L^3+3L^2+12} \\ \frac{4L^4+L^3+48L+3}{(15L+1)(L^3+3L^2+12)} \\ \frac{L^3+12}{2(L^3+3L^2+12)} \end{pmatrix} \tag{A.6}$$

$$K^{-1}(7:9,7) = \begin{pmatrix} -\frac{6}{(L^3+3L^2+12)} + 1 \\ -\frac{3L}{2(L^3+3L^2+12)} \\ -\frac{3L}{L^3+3L^2+12} \end{pmatrix} \tag{A.7}$$

$$K^{-1}(8:9,8) = \begin{pmatrix} -\frac{1}{8(L+1)} - \frac{1}{24(15L+1)} - \frac{3L^2}{8(L^3+3L^2+12)} + \frac{1}{3} \\ -\frac{(L^3+\frac{3L^2}{2}+3)}{(L+1)(L^3+3L^2+12)} + \frac{1}{2} \end{pmatrix} \tag{A.8}$$

$$K^{-1}(9,9) = \begin{pmatrix} -\frac{(2L^3+3L^2+6)}{(L+1)(L^3+3L^2+12)} + 1 \end{pmatrix} \tag{A.9}$$

A.2 Stiffness matrix and inverse of 4 bar structure without the top bar.

Here is stated the stiffness matrix and its inverse for the structure considered in Figure 7-2 comprised of only 3 beams, where the top bar has been omitted.

$$K = \begin{pmatrix} \frac{12}{L^3}+1 & 0 & -\frac{6}{L^2} & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L}+12 & 6 & 0 & -12 & 6 & 0 & 0 & 0 \\ -\frac{6}{L^2} & 6 & \frac{4}{L}+4 & 0 & -6 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & \frac{12}{L^3}+1 & 0 & -\frac{6}{L^2} & -\frac{12}{L^3} & 0 & -\frac{6}{L^2} \\ 0 & -12 & -6 & 0 & \frac{1}{L}+12 & -6 & 0 & -\frac{1}{L} & 0 \\ 0 & 6 & 2 & -\frac{6}{L^2} & -6 & \frac{4}{L}+4 & \frac{6}{L^2} & 0 & \frac{2}{L} \\ 0 & 0 & 0 & -\frac{12}{L^3} & 0 & \frac{6}{L^2} & \frac{12}{L^3} & 0 & \frac{6}{L^2} \\ 0 & 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 & \frac{1}{L} & 0 \\ 0 & 0 & 0 & -\frac{6}{L^2} & 0 & \frac{2}{L} & \frac{6}{L^2} & 0 & \frac{4}{L} \end{pmatrix}$$
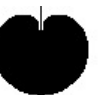
Again, the lower triangular part of the symmetric matrix is given in equations (A.10) to (A.18).

$$
K^{-1}(1:9,1) = \begin{pmatrix} \frac{L^3}{3} \\ 0 \\ \frac{L^2}{2} \\ \frac{L^3}{3} \\ \frac{L^2}{2} \\ \frac{L^2}{2} \\ -\frac{L^3}{6} \\ \frac{L^2}{2} \\ \frac{L^2}{2} \end{pmatrix} \tag{A.10}
$$

$$
K^{-1}(2:9,2) = \begin{pmatrix} L \\ 0 \\ 0 \\ L \\ 0 \\ 0 \\ L \\ 0 \end{pmatrix} \tag{A.11}
$$

$$
K^{-1}(3:9,3) = \begin{pmatrix} L \\ \frac{L^2}{2} \\ L \\ L \\ -\frac{L^2}{2} \\ L \\ L \end{pmatrix} \tag{A.12}
$$

$$
K^{-1}(4:9,4) = \begin{pmatrix} \frac{L^3}{3} + 1 \\ \frac{L^2}{2} \\ \frac{L^2}{2} \\ 1 - \frac{L^3}{6} \\ \frac{L^2}{2} \\ \frac{L^2}{2} \end{pmatrix} \tag{A.13}
$$

$$K^{-1}(5:9,5) = \begin{pmatrix} 2\,L + \frac{1}{3} \\ L + \frac{1}{2} \\ -\frac{L\,(L+1)}{2} \\ 2\,L + \frac{1}{3} \\ L + \frac{1}{2} \end{pmatrix} \tag{A.14}$$

$$K^{-1}(6:9,6) = \begin{pmatrix} L + 1 \\ -\frac{L\,(L+2)}{2} \\ L + \frac{1}{2} \\ L + 1 \end{pmatrix} \tag{A.15}$$

$$K^{-1}(7:9,7) = \begin{pmatrix} \frac{2\,L^3}{3} + L^2 + 1 \\ -\frac{L\,(L+1)}{2} \\ -L\,(L+1) \end{pmatrix} \tag{A.16}$$

$$K^{-1}(8:9,8) = \begin{pmatrix} 3\,L + \frac{1}{3} \\ L + \frac{1}{2} \end{pmatrix} \tag{A.17}$$

$$K^{-1}(9,9) = \begin{pmatrix} 2\,L + 1 \end{pmatrix} \tag{A.18}$$

A.3 Stiffness matrix and inverse of 4 bar structure without the bottom bar.

Here is stated the stiffness matrix and its inverse for the structure considered in Figure 7-2 comprised of only 3 beams, where the bottom bar has been omitted.

$$K = \begin{pmatrix} \frac{12}{L^3} & 0 & -\frac{6}{L^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{6}{L^2} & 0 & \frac{4}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{12}{L^3} & 0 & -\frac{6}{L^2} & -\frac{12}{L^3} & 0 & -\frac{6}{L^2} \\ 0 & 0 & 0 & 0 & \frac{1}{L} & 0 & 0 & -\frac{1}{L} & 0 \\ 0 & 0 & 0 & -\frac{6}{L^2} & 0 & \frac{4}{L} & \frac{6}{L^2} & 0 & \frac{2}{L} \\ 0 & 0 & 0 & -\frac{12}{L^3} & 0 & \frac{6}{L^2} & \frac{12}{L^3} + 1 & 0 & \frac{6}{L^2} \\ 0 & 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 & \frac{1}{L} + 12 & -6 \\ 0 & 0 & 0 & -\frac{6}{L^2} & 0 & \frac{2}{L} & \frac{6}{L^2} & -6 & \frac{4}{L} + 4 \end{pmatrix} \tag{A.19}$$

The simplicity of the representation of the inverse of this stiffness matrix allows us
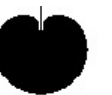
to present the entire matrix in equation (A.20).

$$
K^{-1} =
\begin{pmatrix}
\frac{L^3}{3} & 0 & \frac{L^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & L & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{L^2}{2} & 0 & L & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{L^3}{3} + L^2 + 1 & \frac{L}{2} & \frac{L^2}{2} + L & 1 & \frac{L}{2} & L \\
0 & 0 & 0 & \frac{L}{2} & L + \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{2} \\
0 & 0 & 0 & \frac{L^2}{2} + L & \frac{1}{2} & L + 1 & 0 & \frac{1}{2} & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & \frac{L}{2} & \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{2} \\
0 & 0 & 0 & L & \frac{1}{2} & 1 & 0 & \frac{1}{2} & 1
\end{pmatrix}
\tag{A.20}
$$

Mesh refinement studies

In this appendix the effect of mesh refinement on the values of compliance are studied.

B.1 Cantilevered beam with point load
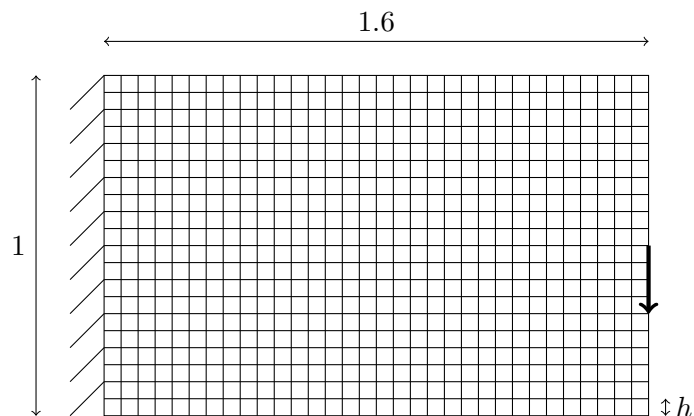
Figure B-1: Design domain of a centrally loaded cantilevered beam. The aspect ratio of the design domain is 1.6 and a unit load is applied vertically from the centre of the right hand side of the domain.

When a load is applied to a single node in the finite-element mesh this corresponds to the underlying f in the continuous setting being a Dirac-delta distribution. This follows because if the underlying function had more than a single point value, its support would intersect with the support of at least two of the finite-element basis functions. As such we see that $f = \delta$ and it is known that δ is not a function in the classical sense and is therefore not in L^2. Hence the standard finite element theory does not apply, specifically that $u_h \to u$ as $h \to 0$.
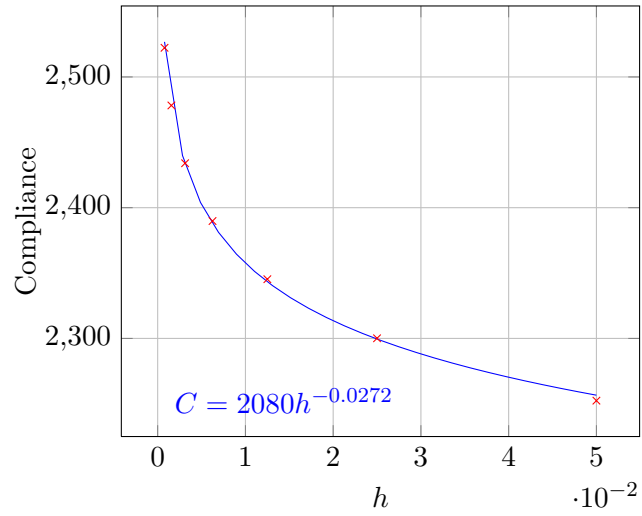
Figure B-2: Compliance plot for different mesh sizes h applied to a short cantilevered beam. The red crosses are the values of the compliance. The blue line is a best fit line calculated from the below log–log plot.
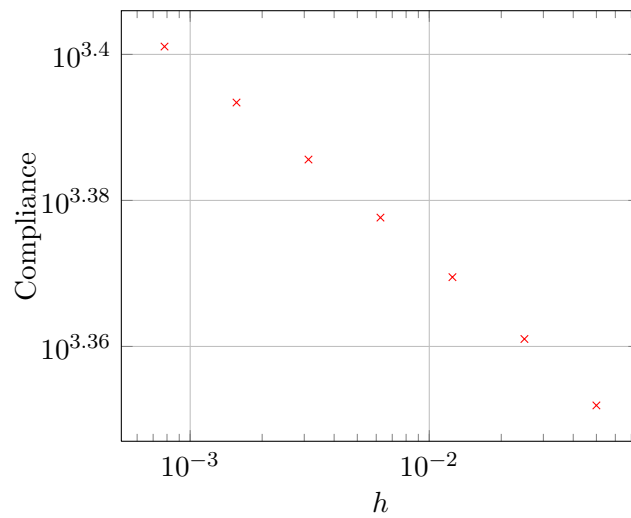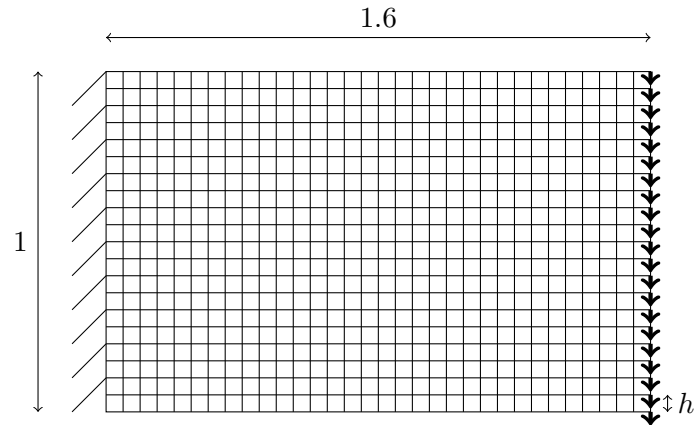

Figure B-3: Log–log plot of compliance against the mesh size for the short cantilevered beam. This plot appears to have a gradient of -0.0272.

179

B.2 Cantilevered beam with distributed load

In the case of a distributed load, the right hand side f is in L^2 and as expected we find that the compliance converges as $h \to 0$.

Figure B-4: Design domain of a cantilevered beam. The aspect ratio of the design domain is 1.6 and a unit load is applied vertically distributed over the right hand side of the domain.
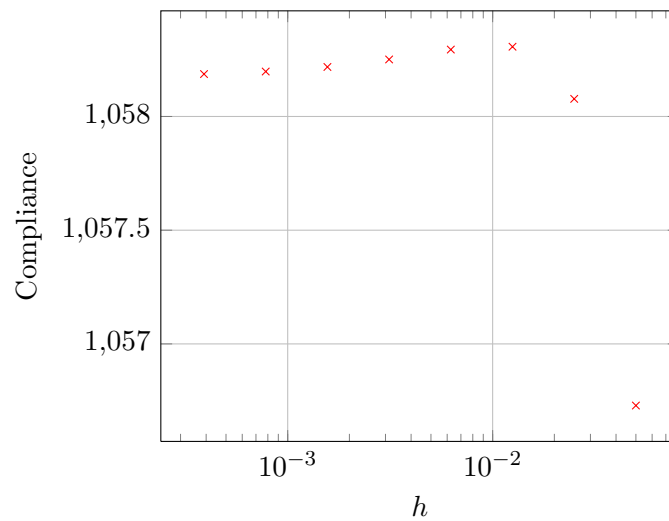

Figure B-5: Compliance plot for different mesh sizes h applied to a short cantilevered beam with distributed load. The red crosses are the values of the compliance.

Bibliography

[1] HSL(2011). A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk.

[2] E.H.L. Aarts and J. Korst. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1989.

[3] W. Achtziger. On simultaneous optimization of truss geometry and topology. *Structural and Multidisciplinary Optimization*, 33(4):285–304, 2007.

[4] W. Achtziger and M. Kočvara. On the maximization of the fundamental eigenvalue in topology optimization. *Structural and Multidisciplinary Optimization*, 34(3):181–195, 2007.

[5] W. Achtziger and M. Kočvara. Structural topology optimization with eigenvalues. *SIAM Journal on Optimization*, 18(4):1129–1164, 2007.

[6] Wolfgang Achtziger and Mathias Stolpe. Truss topology optimization with discrete design variables - Guaranteed global optimality and benchmark examples. *Structural and Multidisciplinary Optimization*, 34(1):1–20, December 2007.

[7] Wolfgang Achtziger and Mathias Stolpe. Global optimization of truss topology with discrete bar areas - Part I: theory of relaxed problems. *Computational Optimization and Applications*, 40(2):247–280, November 2008.

[8] Wolfgang Achtziger and Mathias Stolpe. Global optimization of truss topology with discrete bar areas - Part II : Implementation and numerical results. *Computational Optimization and Applications*, 44(2):315–341, 2009.

[9] Luigi Ambrosio and Giuseppe Buttazzo. An optimal design problem with perimeter penalization. *Calculus of Variations and Partial Differential Equations*, 1:55–69, 1993.

[10] Oded Amir, Martin P. Bendsøe, and Ole Sigmund. Approximate reanalysis in topology optimization. *International Journal for Numerical Methods in Engineering*, 78:1474–1491, 2009.

[11] Oded Amir and Ole Sigmund. On reducing computational effort in topology optimization: how far can we go? *Structural and Multidisciplinary Optimization*, 44(1):25–29, October 2010.

[12] Oded Amir, Mathias Stolpe, and Ole Sigmund. Efficient use of iterative solvers in nested topology optimization. *Structural and Multidisciplinary Optimization*, 42(1):55–72, December 2009.

[13] Samuel Amstutz. Augmented Lagrangian for cone constrained topology optimization. *Computational Optimization and Applications*, 49(1):101–122, July 2009.

[14] Javier Arjona-Baez and J.E. Akin. Enhancing structural topology optimization. *Engineering Computations*, 18(3/4):663–675, 2001.

[15] J.S. Arora and M.W. Huang. Methods for optimization of nonlinear problems with discrete variables: a review. *Structural Optimization*, 8:69–85, 1994.

[16] Klaus-Jürgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Inc., 1982.

[17] Muriel Beckers and C. Fleury. A primal-dual approach in truss topology optimization. *Computers & Structures*, 64(1-4):77–88, 1997.

[18] A. Ben-Tal, F. Jarre, M. Kočvara, A. Nemirovski, and J. Zowe. Optimal design of trusses under a nonconvex global buckling constraint. *Optimization and Engineering*, 1(2):189–213, 2000.

[19] Martin P. Bendsøe and Ole Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2003.

[20] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, November 1988.

[21] MP Bendsøe. Optimal shape design as a material distribution problem. *Structural and multidisciplinary optimization*, 1(4):193–202, 1989.

[22] E.G. Birgin and J.M. Martınez. Practical augmented lagrangian methods. *Encyclopedia of Optimization,*, pages 3013–3023, 2007.

[23] Claudio Bogani, Michal Kočvara, and Michael Stingl. A new approach to the solution of the VTS problem with vibration and buckling constraints. In *8th World Congress on Structural and Multidisciplinary Optimization*, pages 1–10, Lisbon, Portugal, 2009.

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

[25] S.P. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge Univ Pr, 2004.

[26] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer Verlag, 2008.

[27] P.A. Browne, C. Budd, N.I.M. Gould, H.A. Kim, and J.A. Scott. A fast method for binary programming using first-order derivatives, with application to topology optimization with buckling constraints. *International Journal for Numerical Methods in Engineering*, 2012.

[28] M Bruggi and P Venini. A mixed FEM approach to stress-constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 73:1693–1714, 2008.

[29] M. Bruyneel, B. Colson, and A. Remouchamps. Discussion on some convergence problems in buckling optimisation. *Structural and Multidisciplinary Optimization*, 35:181–186, 2008.

[30] Sean Buckeridge. *Numerical Solution of Weather and Climate Systems.* PhD thesis, University of Bath, November 2010.

[31] T. Buhl, C.B.W. Pedersen, and O. Sigmund. Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19(2):93–104, 2000.

[32] Martin Burger, Benjamin Hackl, and Wolfgang Ring. Incorporating topological derivatives into level set methods. *J. Comput. Phys.*, 194(1):344–362, February 2004.

[33] Jane Burry, Peter Felicetti, Jiwu Tang, Mark Burry, and Mike Xie. Dynamical structural modeling - A collaborative design exploration. *International Journal of Architectural Computing*, 03(01):27–42, 2005.

[34] A. Canelas, J. Herskovits, and J.C.F. Telles. Shape optimization using the boundary element method and a SAND interior point algorithm for constrained optimization. *Computers & Structures*, 86:1517–1526, 2008.

[35] V.J. Challis. A discrete level-set topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 41(3):453–464, 2010.

[36] Ting-yu Chen, Hsien-Chie Cheng, and Kuo-Ning Chiang. Optimal Configuration Design of Elastic Structures for Stability. In *Proceedings of the The Fourth International Conference on High-Performance Computing in the Asia-Pacific Region-Volume 2*, pages 1112–1117, 2000.

[37] Gengdong Cheng and Xiaofeng Liu. Discussion on symmetry of optimum topology design. *Structural and Multidisciplinary Optimization*, July 2011.

[38] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-region methods*. Mathematical Programming Society and the Society for Industrial and Applied Mathematics, 2000.

[39] Robert D. Cook, David S. Malkus, and Michael E. Plesha. *Concepts and Applications of Finite Element Analysis*. John Wiley and Sons, 1989.

[40] G.B. Dantzig. Reminiscences about the origins of linear programming. *Operations Research Letters*, 1(2):43–48, 1982.

[41] G.B. Dantzig and M.N. Thapa. *Linear Programming: 2: Theory and Extensions*. Springer Series in Operations Research. Springer, 2003.

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

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

[44] Jianbin Du and Niels Olhoff. Topology optimization of continuum structures with respect to simple and multiple eigenfrequencies. In *6th World Congresses of Structural and Multidisciplinary Optimization*, pages 1–9, Rio de Janeiro, 2005.

[45] Ian S Duff, Albert Maurice Erisman, and John K Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, 1986.

[46] Peter Donald Dunning. *Introducing Loading Uncertainty in Level Set-Based Structural Topology Optimisation*. PhD thesis, University of Bath, 2011.

[47] Caroline Suzanne Edwards. *The analysis and development of efficient and reliable topology optimisation schemes*. PhD thesis, University of Bath, February 2008.

[48] Abderrahman El maliki, Michel Fortin, Nicolas Tardieu, and Andr Fortin. Iterative solvers for 3d linear and nonlinear elasticity problems: Displacement and mixed formulations. *International Journal for Numerical Methods in Engineering*, 83(13):1780–1802, 2010.

[49] L. Euler. Sur la force des colonnes. *Mem. Acad., Berlin*, 13, 1759.

[50] L. Euler. *Novi commentarii Academiae Scientiarum Imperialis Petropolitanae*, volume 5, pages 299–316. Petropolis, Typis Academiae Scientarum, 1760. http://www.biodiversitylibrary.org/bibliography/9527.

[51] J. Farkas and L. Szabo. Optimum design of beams and frames of welded I-sections by means of backtrack programming. *Acta Technica*, 91(1):121–135, 1980.

[52] P. Fernandes, J.M. Guedes, and H. Rodrigues. Topology optimization of three-dimensional linear elastic structures with a constraint on "perimeter". *Science*, 73:583–594, 1999.

[53] R. Fletcher. Semi-Definite Matrix Constraints in Optimization. *SIAM Journal on Control and Optimization*, 23(4):493, 1985.

[54] Anders Forsgren, Philip E Gill, and Margaret H Wright. Interior Methods for Nonlinear Optimization. *SIAM Review*, 44(4):525–597, 2002.

[55] Stephane Garreau, Philippe Guillaume, and Mohamed Masmoudi. The Topological Asymptotic for PDE Systems: The Elasticity Case. *SIAM Journal on Control and Optimization*, 39(6):1756, 2001.

[56] Philip E. Gill, Walter Murray, and Michael A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[57] G.M.L. Gladwell. *Contact problems in the classical theory of elasticity*. Springer, 1980.
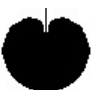
[58] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.

[59] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–279, September 1958.

[60] N.I.M. Gould and D.P. Robinson. A second derivative SQP method: Local convergence and practical issues. *SIAM Journal on Optimization*, 20(4):2049–2079, 2010.

[61] A.E. Green and W. Zerna. *Theoretical Elasticity*. Oxford University Press, 2nd edition, 1968.

[62] Michael Griebel, Daniel Oeltz, and Marc Alexander Schweitzer. An Algebraic Multigrid Method for Linear Elasticity. *SIAM Journal on Scientific Computing*, 25(2):385, 2003.

[63] Y.X. Gu, G.Z. Zhao, H.W. Zhang, Z. Kang, and R.V. Grandhi. Buckling design optimization of complex built-up structures with shape and size variables. *Structural and Multidisciplinary Optimization*, 19(3):183–191, 2000.

[64] X. Guo, G.D. Cheng, and N. Olhoff. Optimum design of truss topology under buckling constraints. *Structural and Multidisciplinary Optimization*, 30(3):169–180, 2005.

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

[66] Raphael T. Haftka. Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103, July 1985.

[67] Raphael T. Haftka and Zafer Gürdal. *Elements of Structural Optimization*. Kluwer Academic Publishers, third edition, 1991.

[68] J. Herskovits, G. Dias, G. Santos, and C.M. Mota Soares. Shape structural optimization with an interior point nonlinear programming algorithm. *Structural and Multidisciplinary Optimization*, 20(2):107–115, October 2000.

[69] J.D. Hogg, J.K. Reid, and J.A. Scott. A DAG-based Sparse Cholesky Solver for Multicore Architectures. Technical report, Science and Technology Facilities Council, 2009.

[70] J.D. Hogg, J.K. Reid, and J.A. Scott. Design of a multicore sparse cholesky factorization using DAGs. *SISC*, 32:3627–3649, 2010.

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

[72] X. Huang and Y.M. Xie. A new look at ESO and BESO optimization methods. *Structural and Multidisciplinary Optimization*, 35(1):89–92, May 2007.

[73] X. Huang and Y.M. Xie. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, 43(14):1039–1049, October 2007.

[74] X. Huang and Y.M. Xie. A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization*, 41(5):671–683, 2010.

[75] X. Huang and Y.M. Xie. *Evolutionary Topology Optimization of Continuum Structures*. Wiley, 2010.

[76] X. Huang, Z.H. Zuo, and Y.M. Xie. Combining genetic algorithms with BESO for topology optimization. *Structural and Multidisciplinary Optimization*, 38:511–523, 2009.

[77] G.W. Hunt and J.M.T. Thompson. *A General Theory of Elastic Stability*. Wiley-Interscience, 1973.

[78] J. Jensen and N. Pedersen. On maximal eigenfrequency separation in two-material structures: the 1D and 2D scalar cases. *Journal of Sound and Vibration*, 289(4-5):967–986, February 2006.

[79] C. Jog and R. Haber. Stability of finite element models for distributed-parameter optimization and topology design. *Computer Methods in Applied Mechanics and Engineering*, 130(3-4):203–226, April 1996.

[80] K.V. John and C.V. Ramakrishnan. Optimum design of trusses from available sections - use of sequential linear programming with branch and bound algorithm. *Engineering Optimization*, 13:119–145, 1988.

[81] S. Johnson. The NLopt nonlinear-optimization package. *URL: http://ab-initio.mit.edu/nlopt*, 2010.

[82] Yoshihiro Kanno and Xu Guo. A mixed integer programming for robust truss topology optimization with stress constraints. *International Journal for Numerical Methods in Engineering*, 83(13):1675–1699, 2010.

[83] Şükrü Karakaya and Ömer Soykasap. Natural frequency and buckling optimization of laminated hybrid composite plates using genetic algorithm and simulated annealing. *Structural and Multidisciplinary Optimization*, 43(1):61–72, July 2011.

[84] Frank C. Karal and Samuel N. Karp. The elastic-field behavior in the neighborhood of a crack of arbitrary angle. *Communications on Pure and Applied Mathematics*, 15(4):413–421, 1962.

[85] E. Karer and J.K. Kraus. Algebraic multigrid for finite element elasticity equations: Determination of nodal dependence via edge-matrices and two-level convergence. *International Journal for Numerical Methods in Engineering*, 83(5):642–670, 2010.

[86] N.S. Khot. Nonlinear analysis of optimized structure with constraints on system stability. *AIAA Journal(ISSN 0001-1452)*, 21:1181–1186, 1983.

[87] N.S. Khot, V.B. Venkayya, and L. Berke. Optimum structural design with stability constraints. *International Journal for Numerical Methods in Engineering*, 10(5):1097–1114, 1976.

[88] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[89] U. Kirsch and G.I.N. Rozvany. Alternative formulations of structural optimization. *Structural Optimization*, 7:32–41, 1994.

[90] M. Kočvara. On the modelling and solving of the truss design problem with global stability constraints. *Structural and multidisciplinary optimization*, 23(3):189–203, 2002.

[91] Michal Kočvara. On the modelling and solving of the truss design problem with global stability constraints. *Structural and Multidisciplinary Optimization*, 23(3):189–203, April 2002.

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

[93] R.V. Kohn and G. Strang. Optimal design and relaxation of variational problems, i. *Communications on Pure and Applied Mathematics*, 39(1):113–137, 1986.

[94] R.V. Kohn and G. Strang. Optimal design and relaxation of variational problems, ii. *Communications on Pure and Applied Mathematics*, 39(2):139–182, 1986.

[95] R.V. Kohn and G. Strang. Optimal design and relaxation of variational problems, iii. *Communications on Pure and Applied Mathematics*, 39(3):353–377, 1986.

[96] R. Lakes. Material with structural hierarchy. *Nature*, 361(6412):511–515, 1993.

[97] A.H. Land and A.G. Doig. An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 28(3):497–520, 1960.

[98] Ulrik Darling Larsen, Ole Sigmund, and Siebe Bouwstra. Design and Fabrication of Compliant Micromechanisms and Structures with Negative Poisson's Ratio. *Journal of Microelectromechanical Systems*, 6(2):99–106, 1997.

[99] T.H. Lee. Adjoint Method for Design Sensitivity Analysis of Multiple Eigenvalues and Associated Eigenvectors. *AIAA journal*, 45(8):1998, 2007.

[100] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK Users' Guide*. SIAM, 1998.

[101] Esben Lindgaard and Erik Lund. Nonlinear buckling optimization of composite structures. *Computer methods in applied mechanics and engineering*, 199:2319–2330, 2010.

[102] Esben Lindgaard and Erik Lund. Optimization formulations for the maximum nonlinear buckling load of composite structures. *Structural and Multidisciplinary Optimization*, 43(5):631–646, November 2010.

[103] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 116. Springer, 2008.

[104] B. Maar and V. Schulz. Interior point multigrid methods for topology optimization. *Structural and Multidisciplinary Optimization*, 19(2):214–224, 2000.

[105] Y. Maeda, S. Nishiwaki, K. Izui, M. Yoshimura, K. Matsui, and K. Terada. Structural topology optimization of vibrating structures with specified eigenfrequencies and eigenmode shapes. *International Journal for Numerical Methods in Engineering*, 67(5):597–628, 2006.

[106] Rafael Martí and Gerhard Reinelt. Branch-and-bound. In *The Linear Ordering Problem*, volume 175 of *Applied Mathematical Sciences*, pages 85–94. Springer Berlin Heidelberg, 2011.

[107] Rafael Martí and Gerhard Reinelt. Branch-and-cut. In *The Linear Ordering Problem*, volume 175 of *Applied Mathematical Sciences*, pages 95–116. Springer Berlin Heidelberg, 2011.

[108] J.M. Martinez. A note on the theoretical convergence properties of the SIMP method. *Structural and Multidisciplinary Optimization*, 29(4):319–323, October 2005.

[109] H. Mateus, C.M. Mota Soares, and C.A. Mota Soares. Buckling sensitivity analysis and optimal design of thin laminated structures. *Computers & Structures*, 64(1-4):461–472, August 1997.

[110] K. Maute and E. Ramm. Adaptive topology optimization. *Structural Optimization*, 10:100–112, 1995.

[111] K.I.M. McKinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal of Optimization*, 9:148–158, 1998.

[112] A.G.M. Michell. LVIII. The limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(47):589–597, 1904.

[113] Radomir Mijailovi. Optimum design of lattice-columns for buckling. *Structural and Multidisciplinary Optimization*, 42:897–906, 2010.

[114] J.J. Moré and S.J. Wright. *Optimization software guide*. Society for Industrial Mathematics, 1993.

[115] Attila P. Nagy, Mostafa M. Abdalla, and Zafer Gürdal. Isogeometric design of elastic arches for maximum fundamental frequency. *Structural and Multidisciplinary Optimization*, 43(1):135–149, August 2011.

[116] Miguel M. Neves, Ole Sigmund, and Martin P. Bendsøe. Topology optimization of periodic microstructures with a penalization of highly localized buckling modes. *International Journal for Numerical Methods in Engineering*, 54(6):809–834, June 2002.

[117] M.M. Neves, H. Rodrigues, and J.M. Guedes. Generalized topology design of structures with a buckling load criterion. *Structural and Multidisciplinary Optimization*, 10(2):71–78, 1995.

[118] M.M. Neves, O. Sigmund, and M.P. Bendsøe. Topology optimization of periodic microstructures with a penalization of highly localized buckling modes. *International Journal for Numerical Methods in Engineering*, 54(6):809–834, 2002.

[119] Fei Niu, Shengli Xu, and Gengdong Cheng. A general formulation of structural topology optimization for maximizing structural stiffness. *Structural and Multidisciplinary Optimization*, 43:561–572, November 2011.

[120] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer verlag, 1999.

[121] J.T. Oden. Calculation of geometric stiffness matrices for complex structures. *AIAA Journal*, 4(8):1480–1482, August 1966.

[122] Carlos E. Orozco and Omar N. Ghattas. A reduced SAND method for optimal design of non-linear structures. *International Journal for Numerical Methods in Engineering*, 40(15):2759–2774, August 1997.

[123] Carlos E. Orozco and Omar N. Ghattast. Sparse Approach to Simultaneous Analysis and Design of Geometrically Nonlinear Structures. *AIAA Journal*, 30(7):1877, 1992.

[124] E.E. Ovtchinnkov and J.K. Reid. A preconditioned block conjugate gradient algorithm for computing extreme eigenpairs of symmetric and hermitian problems. Technical Report RAL-TR-2010-019, RAL, 2010.

[125] N.L. Pedersen. Maximization of eigenvalues using topology optimization. *Structural and Multidisciplinary Optimization*, 20(1):2–11, 2000.

[126] N.L. Pedersen and A.K. Nielsen. Optimization of practical trusses with constraints on eigenfrequencies, displacements, stresses, and buckling. *Structural and Multidisciplinary Optimization*, 25(5-6):436–445, December 2003.

[127] J. Petersson. Some convergence results in perimeter-controlled topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 171(1-2):123–140, March 1999.

[128] O.M. Querin, G.P. Steven, and Y.M. Xie. Evolutionary structural optimisation using an additive algorithm. *Finite Elements in Analysis and Design*, 34:291–308, 2000.

[129] O.M. Querin, V. Young, G.P. Steven, and Y.M. Xie. Computational efficiency and validation of bi-directional evolutionary structural optimisation. *Computer Methods in Applied Mechanics and Engineering*, 189:559–573, 2000.

[130] S. Rahmatalla and C.C. Swan. Continuum topology optimization of buckling-sensitive structures. *AIAA journal*, 41(6):1180–1189, 2003.

[131] S.F. Rahmatalla and C.C. Swan. A Q4/Q4 continuum structural topology optimization implementation. *Structural and Multidisciplinary Optimization*, 27(1):130–135, 2004.

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

[133] Ulf T. Ringertz. On topology optimization of trusses. *Engineering Optimization*, 9:209–218, 1985.

[134] Ulf T. Ringertz. A branch and bound algorithm for topology optimization of truss structures. *Engineering Optimization*, 10:111–124, 1986.

[135] Ulf T. Ringertz. On Methods for Discrete Structural Optimization. *Engineering Optimization*, 13(1):47–64, 1988.

[136] George I. N. Rozvany. Authors reply to a discussion by Gengdong Cheng and Xiaofeng Liu of the review article On symmetry and non-uniqueness in exact topology optimization by George I.N. Rozvany (2011, Struct Multidisc Optim 43:297317). *Structural and Multidisciplinary Optimization*, 44(5):719–721, September 2011.

[137] George I. N. Rozvany. On symmetry and non-uniqueness in exact topology optimization. *Structural and Multidisciplinary Optimization*, 43(3):297–317, September 2011.

[138] G.I.N. Rozvany. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, 21(2):90–108, April 2001.

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

[140] G.I.N. Rozvany and M. Zhou. Applications of the COC method in layout optimization. In H. Eschenauer, C. Mattheck, and N. Olhoff, editors, *Proceedings of*

the International Conference of Engineering Optimization in Design Processes, Karlsruhe, 1990, pages 59–70, Berlin, 1991. Springer Verlag.

[141] S. Sadiku. Buckling load optimization for heavy elastic columns: a perturbation approach. *Structural and multidisciplinary optimization*, 35(5):447–452, 2008.

[142] E. Salajegheh and G.N. Vanderplaats. Optimum design of trusses with discrete sizing and shape variables. *Structural Optimization*, 6:79–85, 1993.

[143] E. Sandgren. Nonlinear integer and discrete programming for topological decision making in engineering design. *Journal of Mechanical Design*, 112(1):118–122, 1990.

[144] E. Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, 112(2):223–229, 1990.

[145] S. Sankaranarayanan, Raphael T. Haftka, and Rakesh K. Kapaniaf. Truss topology optimization with simultaneous analysis and design. *American Institute of Aeronautics and Astronautics*, 32(2):420–424, 1994.

[146] O. Sardan, V. Eichhorn, D.H. Petersen, S. Fatikow, O. Sigmund, and P. Bøggild. Rapid prototyping of nanotube-based devices using topology-optimized microgrippers. *Nanotechnology*, 19(49):495–503, December 2008.

[147] M. Save, W. Prager, G. Sacchi, and W.H. Warner. *Structural Optimization: Optimality criteria*. Mathematical concepts and methods in science and engineering. Plenum Press, 1985.

[148] Lucien A. Schmit and Richard L. Fox. An integrated approach to structural synthesis and analysis. *AIAA Journal*, 3(6):1104–1112, June 1965.

[149] Shahriar Setoodeh, Mostafa M. Abdalla, Samuel T. IJsselmuiden, and Zafer Gürdal. Design of variable-stiffness composite panels for maximum buckling load. *Composite Structures*, 87(1):109–118, 2009.

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

[151] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural and Multidisciplinary Optimization*, 16(1):68–75, August 1998.

[152] Ole Sigmund. On the Design of Compliant Mechanisms Using Topology Optimization*. *Mechanics of Structures and Machines*, 25(4):493–524, January 1997.

[153] Ole Sigmund. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43(5):589–596, March 2011.

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

[155] S.G. Soni. *Operations Research.* Prentice-Hall Of India Pvt. Limited, 2007.

[156] Roman Stainko. An adaptive multilevel approach to the minimal compliance problem in topology optimization. *Communications in Numerical Methods in Engineering*, 22(2):109–118, 2006.

[157] James H. Starnes Jr. and Raphael T. Haftka. Preliminary Design of Composite Wings for Buckling, Strength, and Displacement Constraints. *Journal of Aircraft*, 16(8):564–570, 1979.

[158] Rob Stevenson. Robustness of multi-grid applied to anisotropic equations on convex domains with re-entrant corners. *Numerische Mathematik*, 66:373–398, 1993.

[159] M. Stingl, M. Kočvara, and G. Leugering. Free material optimization with fundamental eigenfrequency constraints. *SIAM Journal on Optimization*, 20(1):524–547, 2009.

[160] M. Stolpe and K. Svanberg. On the trajectories of penalization methods for topology optimization. *Structural and Multidisciplinary Optimization*, 21(2):128–139, 2001.

[161] Mathias Stolpe. On some fundamental properties of structural topology optimization problems. *Structural and Multidisciplinary Optimization*, 41(5):661–670, 2010.

[162] Mathias Stolpe and Martin P. Bendsøe. Global optima for the Zhou-Rozvany problem. *Structural and Multidisciplinary Optimization*, 43(2):151–164, 2010.

[163] Francis Sullivan. The joy of algorithms. *Computing in Science and Engineering*, page 2, 2000.

[164] Krishnan Suresh. A 199-line matlab code for pareto-optimal tracing in topology optimization. *Structural and Multidisciplinary Optimization*, 42(5):665–679, 2010.

[165] Katsuyuki Suzuki and Noboru Kikuchi. A homogenization method for shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 93:291–318, 1991.

[166] K. Svanberg. On the convexity and concavity of compliances. *Structural Optimization*, 7:42–46, 1994.

[167] Krister Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal For Numerical Methods in Engineering*, 24:359–373, 1987.

[168] Krister Svanberg. The Method of Moving Asymptotes - A New Method for Structural Optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.

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

[170] C. Talischi, G.H. Paulino, and C.H. Le. Honeycomb Wachspress finite elements for structural topology optimization. *Structural and Multidisciplinary Optimization*, 37(6):569–583, 2009.

[171] Cameron Talischi, Glaucio H. Paulino, and Chau H. Le. Topology Optimization Using Wachspress-Type Interpolation with Hexagonal Elements. *Multiscape and Functionally Graded Materials*, pages 309–314, 2008.

[172] P Tanskanen. The evolutionary structural optimization method: theoretical aspects. *Computer Methods in Applied Mechanics and Engineering*, 191(47-48):5485–5498, November 2002.

[173] Lazarus H. Tenek and Ichiro Hagiwara. Eigenfrequency Maximization of Plates by Optimization of Topology Using Homogenization and Mathematical Programming. *JSME International Journal Series C*, 37(4):667–677, 1994.

[174] A.R. Toakley. Optimum design using available sections. *Proc Amer Soc Civil Eng, J Struct Div,*, 94:1219–1241, 1968.

[175] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.

[176] S. Venkataraman and R.T. Haftka. Structural optimization complexity: what has moore's law done for us? *Structural and Multidisciplinary Optimization*, 28(6):375–387, 2004.

[177] Shun Wang, Eric de Sturler, and Glaucio H. Paulino. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69(12):2441–2468, 2007.

[178] Ryo Watada, Makoto Ohsaki, and Yoshihiro Kanno. Non-uniqueness and symmetry of optimal topology of a shell for minimum compliance. *Structural and Multidisciplinary Optimization*, pages 459–471, November 2010.

[179] Peng Wei, Michael Yu Wang, and Xianghua Xing. A study on x-fem in continuum structural optimization using a level set model. *Comput. Aided Des.*, 42(8):708–719, August 2010.

[180] J.W.J. Williams. Algorithm 232: heapsort. *Communications of the ACM*, 7(6):347–348, 1964.

[181] Wayne L. Winston. *Introduction to Mathematical Programming Application & Algorithms*. PWS-Kent, 1991.

[182] L.A. Wolsey. *Integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1998.

[183] Qi Xia, Tielin Shi, and Michael Yu Wang. A level set based shape and topology optimization method for maximizing the simple or repeated first eigenvalue of structure vibration. *Structural and Multidisciplinary Optimization*, 43(4):473–485, November 2010.

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

[185] Kazuo Yonekura and Yoshihiro Kanno. Global optimization of robust truss topology via mixed integer semidefinite programming. *Optimization and Engineering*, 11:355–379, 2010.

[186] J. Zhan, X. Zhang, and J. Hu. Maximization of Values of Simple and Multiple Eigenfrequencies of Continuum Structures Using Topology Optimization. In *International Conference on Measuring Technology and Mechatronics Automation*, pages 833–837. IEEE Computer Society, 2009.

[187] M. Zhou and G.I.N. Rozvany. On the validity of ESO type methods in topology optimization. *Structural and Multidisciplinary Optimization*, 21(1):80–83, March 2001.

Index