

**Multigrid Implementation of Cellular Automata for  
Topology Optimisation of Continuum Structures  
with Design-Dependent Loads**



# Multigrid Implementation of Cellular Automata for Topology Optimisation of Continuum Structures with Design-Dependent Loads

Proefschrift

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus Prof.dr.ir. J.T. Fokkema,  
voorzitter van het College voor Promoties,  
in het openbaar te verdedigen op 8 Juni om 10.00 uur

door

Ramzi ZAKHAMA

Master of Science in Civil Engineering, Ecole Nationale d'Ingénieurs de Tunis,  
Tunisia  
geboren te Verviers, België

Dit proefschrift is goedgekeurd door de promotoren:

Prof.dr. Z. Gürdal  
Prof.dr. H. Smaoui

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr. Z. Gürdal	Technische Universiteit Delft, promotor
Prof.dr. H. Smaoui	Ecole Nationale d'Ingénieurs de Tunis, promotor
Dr. M. M. Abdalla	Technische Universiteit Delft, adjunct supervisor
Prof.dr P. Duysinx	Université de Liège
Prof.dr.ir. A. van Keulen	Technische Universiteit Delft
Dr. D. Bassir	Technische Universiteit Delft, adviseur
Dr. J. van der Tempel	Technische Universiteit Delft, adviseur
Prof.dr. A. Rothwell	Technische Universiteit Delft, reservelid

Cover picture: Courtesy of TOSHIBA

Publisher: TUDelft, Faculteit Luchtvaart-en Ruimtevaarttechniek

Printed by: PrintPartners Ipskamp, Enschede

ISBN 978-90-9024227-9

Keywords: Topology optimisation, Continuum structures, Multigrid, Cellular automata, Design-dependent loads.

Copyright ©2009 by Ramzi Zakhama

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the publisher: TUDelft, Faculteit Luchtvaart-en Ruimtevaarttechniek

Printed in the Netherlands

*To my mother and  
my father.*



## **Acknowledgements**

On completion of this thesis, I would like to express my sincere gratitude to everyone whose help made this work possible. I wish to thank my advisors Prof. Dr. Zafer Gürdal and Prof. Dr. Hichem Smaoui for their guidance, their generosity and for giving me the wonderful opportunity to work with them. Special thanks go to my co-advisor Dr. Mostafa M. Abdalla for his continuous inspiration and smart ideas, without his support I would not have finished or even started this research. I wish also to extend my thanks to my committee members, Prof. Dr. Pierre Duysinx, Prof. Dr. Ir. Fred van Keulen, Prof. Dr. Allan Rothwell, Dr. David Bassir and Dr. Ir. Jan van der Tempel for serving in my dissertation committee. I would also like to thank Miranda Aldham - Breary for her corrections of my English, Roeland De Breuker for his translations of my abstract and my propositions, and Annemarie van Lienden, Jan Hol, Gillian Saunders - Smits and Angela de Gier for their cooperations and for helping me with the administrative papers. Finally, I would like to thank my friends and colleagues in the group Eelco, Tanvir, Glenn, Agnes, Claudio, Gasser, Sam, Julien, Attila, Christian, Ahmad, Farid, Lotfollah, Noh and Ali.





## Summary

Topology optimisation of continuum structures has become mature enough to be often applied in industry and continues to attract the attention of researchers and software companies in various engineering fields. Traditionally, most available algorithms for solving topology optimisation problems are based on the global solution approach and require a large number of costly analyses. An alternative methodology, based on cellular automata (CA) and accelerated with a multigrid discretisation scheme, can be a good candidate to solve topology optimisation problems in a reasonable amount of computational time. The main advantages of using CA paradigm in structural design are the local analysis and design resolutions, and its parallel nature. The multigrid acceleration method is used to improve its efficiency. This innovative approach offers a new paradigm for design and analysis with higher efficiency compared to existing design optimisation algorithms. In this thesis, the topology optimisation of continuum structures with design-dependent loads is solved using a multigrid accelerated cellular automata algorithm.

The CA paradigm is applied to two and three dimensional continuum topology optimisation problems. An optimisation formulation based on minimum compliance design subject to a volume constraint is used in the present work. The cellular automata based design scheme uses local update(s) rules for both field variables (displacements) and design variables (cell densities). The analysis rules are derived from the principle of minimum total potential energy, and the design rules are derived based on continuous optimality criteria interpreted as local Kuhn-Tucker conditions. Numerical experiments demonstrate the robustness of the proposed CA algorithm to solve topology optimisation problems at the expense of slow convergence. The deterioration of CA convergence rate is due to slow propagation of information from cell to cell as the number of variables increases. A methodology based on multigrid scheme is used to accelerate the CA design algorithm.

The multigrid acceleration scheme uses different discretisation levels of grids. The CA iterations on the finest grid are coupled with the iterations of the correction solution on the coarse grids. The multigrid accelerated CA algorithm is demonstrated to be a powerful tool for solving topology optimisation problems compared to other algorithms based on finite element analysis. The computational cost using this scheme is found to be proportional to the number of cells.

The topology optimisation of continuum structures is also performed under the influence of design-dependent loads. A direct density based method that does not involve explicit construction of the loading surfaces is developed to define the design-dependent loads. The optimisation problem is solved by the modified Method of

Moving Asymptotes coupled with line search (MMALS), and by multigrid accelerated CA algorithms. An explicit constraint is added to the formulation to lead a design to a close pure black/white solution. Numerical examples demonstrate that the multigrid accelerated CA algorithm convergence is more stable and more robust than the MMALS algorithm. The proposed algorithms demonstrates the effect of design-dependent loading onto the topology optimisation problem as illustrated by creating extra elements, shape modifications or void appearing.

## Samenvatting

Topologie-optimalisatie van continue constructies heeft een voldoende volwassen stadium bereikt om vaak toegepast te worden in de industrie, en blijft de aandacht trekken van onderzoekers en softwarebedrijven in diverse ingenieursdisciplines. Traditioneel gezien zijn de meeste beschikbare algoritmes voor het oplossen van topologie-optimalisatie-problemen gebaseerd op de globale oplossing aanpak en vereisen een groot aantal kostelijke analyses. Een alternatieve methode, gebaseerd op cellular automata (CA) en versneld met een multigrid discretiseermethode, kan een goede kandidaat zijn om topologie-optimalisatieproblemen op te lossen binnen een redelijke hoeveelheid rekentijd. De belangrijkste voordelen van CA voor structureel ontwerp zijn de lokale analyse en ontwerpregels, en de parallelle aard. De multigrid acceleration methode wordt gebruikt om de efficiëntie te verhogen. Deze innovatieve aanpak creëert een nieuw paradigma voor ontwerp en analyse met een grote efficiëntie in vergelijking met bestaande optimalisatiealgoritmen. In dit proefschrift wordt het topologie-optimalisatieprobleem opgelost voor continue constructies met ontwerpafhankelijke belastingen door gebruik te maken van een cellular automata algoritme versneld door de multigridtechniek.

Het CA paradigma is toegepast op twee- en driedimensionale continue topologie-optimalisatieproblemen. Een optimalisatieformulatie gebaseerd op een ontwerp met minimale flexibiliteit onderhevig aan randvoorwaarden met betrekking tot het volume is beschreven in dit werk. Het ontwerpschema, gebaseerd op cellular automata, maakt gebruik van lokale update regels voor zowel veldvariabelen (verplaatsingen) als ontwerpvariabelen (celdichtheden). De analyse-regels zijn afgeleid van continue optimaliteitscriteria die geïnterpreteerd kunnen worden als lokale Kuhn-Tucker voorwaarden. Numerieke experimenten tonen de robuustheid aan van het voorgestelde CA algoritme om topologie-optimalisatieproblemen op te lossen ten koste van trage convergentie. De verslechtering van de CA convergentiesnelheid komt door de tragere verspreiding van informatie van de ene cel naar de andere bij toenemend aantal variabelen. Een methodologie, gebaseerd op een multigrid schema, wordt gebruikt om het CA ontwerp-algoritme te versnellen.

Het multigrid versnellings-schema maakt gebruik van verschillende discretisatieniveaus van rasters. De CA iteraties op het fijnste raster zijn gekoppeld met de iteraties van de correctie-oplossing op de groffere rasters. Het is van het multigrid accelerated CA algoritme aangetoond dat het een krachtig instrument is om topologie-optimalisatieproblemen op te lossen in vergelijking met andere algoritmen die gebaseerd zijn op een eindige elementenanalyse. De rekenkost gebruik makend van dit schema is proportioneel met het aantal cellen.

De topologie-optimalisatie van continue constructies is ook uitgevoerd onder invloed van ontwerpafhankelijke belastingen. Een directe methode, gebaseerd op dichtheid, die de expliciete constructie van de belastingsvlakken niet behelst, is ontwikkeld om de ontwerpafhankelijke belastingen te bepalen. Het optimalisatieprobleem wordt opgelost door de aangepaste Method of Moving Asymptotes gekoppeld aan een Line Search (MMALS), en door multigrid accelerated CA algoritmen. Een expliciete randvoorwaarde is toegevoegd aan de formulatie om een ontwerp te bekomen dat dicht aanleunt tegen een zuivere zwart/wit oplossing. Numerieke voorbeelden tonen aan dat de convergentie van het multigrid accelerated CA algoritme stabiel en robuuster is dan het MMALS algoritme. Het beschreven algoritme toont het effect aan van een ontwerpafhankelijke belasting op het topologie-optimalisatieprobleem zoals wordt geïllustreerd door het creëren van extra elementen, vormaanpassingen of het opduiken van leegtes.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background and motivation . . . . .	1
1.2	Topology optimisation . . . . .	3
1.3	Cellular automata for topological optimisation of structures . . . . .	4
1.4	Multigrid acceleration method for topology optimisation . . . . .	6
1.5	Design-dependent loads on the topology optimisation . . . . .	7
1.5.1	Hydrostatic pressure loads . . . . .	8
1.5.2	Transmissible loads . . . . .	9
1.6	Thesis layout . . . . .	10
<b>2</b>	<b>Cellular Automata for Topology Design</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Introduction to cellular automata . . . . .	14
2.2.1	Cell states . . . . .	15
2.2.2	Lattice geometry . . . . .	15

2.2.3	Neighbourhood . . . . .	16
2.2.4	Local rules of transition . . . . .	17
2.2.5	Boundary conditions . . . . .	17
2.3	Minimum compliance design optimality criteria . . . . .	17
2.4	CA implementation for structural design . . . . .	22
2.4.1	Analysis update rule . . . . .	22
2.4.2	Design update rule . . . . .	25
2.4.3	Updating the Lagrange multiplier . . . . .	25
2.4.4	Cellular automata scheme . . . . .	26
2.4.5	Multiple load cases . . . . .	27
2.5	Numerical examples . . . . .	28
2.6	Concluding remarks . . . . .	31
<b>3</b>	<b>Multigrid Accelerated Cellular Automata</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Idea of multigrid . . . . .	34
3.3	Transfer operators . . . . .	36
3.4	Multigrid design algorithm . . . . .	38
3.5	Full multigrid design algorithm . . . . .	39
3.6	Numerical examples . . . . .	41
3.6.1	Comparison . . . . .	41
3.6.2	Parametric studies . . . . .	43

3.6.3	Compression bridge . . . . .	47
3.6.4	Arch bridge . . . . .	49
3.6.5	Multiple arch bridge . . . . .	51
3.7	Concluding remarks . . . . .	53
<b>4</b>	<b>Wind Load Effect in Topology Optimisation Problems</b>	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Topology design with wind loading . . . . .	56
4.2.1	Wind loading function . . . . .	57
4.2.2	Sensitivity analysis . . . . .	58
4.2.3	Optimisation . . . . .	59
4.3	Explicit constraint . . . . .	61
4.4	Numerical examples . . . . .	63
4.4.1	Two-dimensional example . . . . .	64
4.4.2	Three-dimensional example . . . . .	66
4.5	Concluding remarks . . . . .	68
<b>5</b>	<b>Multigrid Accelerated Cellular Automata with Design-Dependent Loads</b>	<b>69</b>
5.1	Introduction . . . . .	69
5.2	Minimum compliance design optimality criteria . . . . .	70
5.3	Convex approximations . . . . .	73
5.4	Relaxation . . . . .	75

5.5	Explicit constraint . . . . .	75
5.6	Algorithm . . . . .	76
5.7	Numerical examples . . . . .	78
5.7.1	Clamped column . . . . .	78
5.7.2	Off-shore wind turbine support structure . . . . .	87
5.7.3	Multiple arch bridge . . . . .	95
5.8	Concluding remarks . . . . .	99
<b>6</b>	<b>Conclusions</b>	<b>101</b>
6.1	Summary . . . . .	101
6.2	Future work . . . . .	103
<b>A</b>	<b>Local Rule of Analysis</b>	<b>105</b>
A.1	Two-dimensional structures . . . . .	105
A.2	Three-dimensional structures . . . . .	106



# Chapter 1

## Introduction

### 1.1 Background and motivation

In the past the process of determining the optimal design of structures such as bridges, cars, or airplanes relied heavily on the engineer's intuition and experience. The designers tried to find optimal forms with a series of tests on many prototypes. This type of procedure is very expensive and inaccurate. The introduction of mathematical optimisation theory into structural design in the early 1960's [1] helped to reduce prototyping costs in manufacturing. Since then research has been carried out, and is intensifying, to find ways to automate the design process. This increase in research has been made possible in particular by the development of the Finite Element Method (FEM) promoting fast structural analyses. Also, the use of multiple optimisation techniques made it possible to reshape or resize structural elements modelled with thousands of degrees of freedom and with thousands of design variables. The goal of the optimisation process is to obtain an optimal solution for a given engineering problem under certain loads and constraints. For more than twenty years, interest in structure optimisation has continued to increase. Today, structural optimisation techniques are classified into the three large families given below.

- Size optimisation, see figure 1.1(a): only the cross-section or the thickness of the different elements of the structure can be modified while the shape and the topology are fixed.

## Chapter 1

- Shape optimisation, see figure 1.1(b): the shape of the structure can be changed while the topology is fixed. Shape optimisation is characterised by modifying the border domain regarding the original structure or by changing the transversal dimensions, but modifying the connectivity or the nature of the structural members is not allowed.
- Topology optimisation, see figure 1.1(c): fundamental modifications of a structural nature can be achieved. In this case, the modifications of connectivity or structural members are possible. Topology optimisation commonly leads to significant improvements in performance.

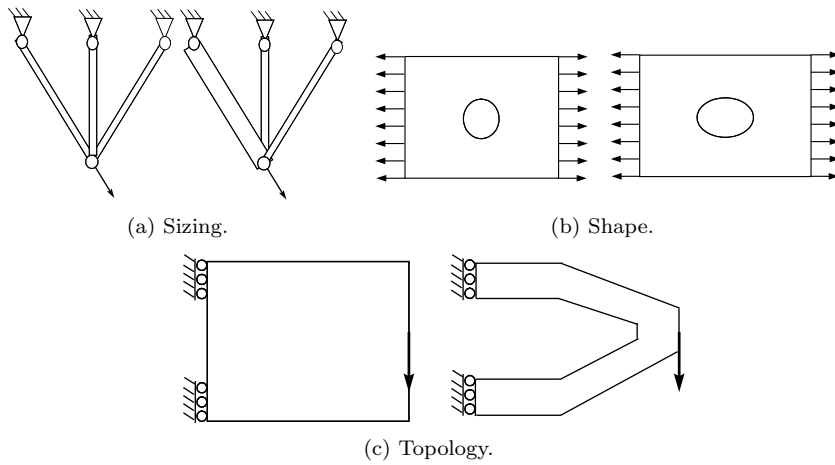


Figure 1.1: The three families of the optimisation structural.

These techniques have been used in designing structures for different fields of application such as space, aeronautics, automobile, naval architecture, and civil engineering, for example. A good design is very important for its resistance to various loadings, its lifespan and its service usage.

In this work, special emphasis is placed on solving, quickly and effectively, the topology optimisation problem for two and three dimensional continuum structures with design-dependent loads on serial computers. The traditional techniques used to solve this problem, based on a FEM solver for large scale models, require a large memory for storage and a large number of costly analyses. In this thesis an attempt is made to use a nontraditional scheme, called Cellular Automata (CA) accelerated with a Multigrid (MG) discretisation scheme, to find solutions for the topology optimisation problem with design-dependent loads using a combined analysis/design approach.

This technique does not require a large amount of memory and may offer additional computational advantages that offset costly analysis requests.

Three main areas are highlighted in this thesis, namely:

1. cellular automata for topology optimisation of structures
2. multigrid acceleration of cellular automata technique for topology optimisation of structures
3. design-dependent loads for topology optimisation of structures

## 1.2 Topology optimisation

Topology optimisation of structures is the most difficult class of problems of structural design because of its combinatorial character that leads to exponential complexity which requires an exponentially increasing amount of resources (e.g. time, computer memory) as the problem size increases. In the last decades, effort to solve the challenge of the increasing complexity of the systems studied, and the capacity of the optimisation methods used to solve these problems in a reasonable time have continued to increase. In spite of some recorded progress, the search is continuing for more powerful and more robust programs that are capable of handling geometric and material nonlinearities and allowing the integration of multiple disciplines, including topology, in the design.

The topology optimisation problem of the structural domain is to find the optimal distribution of structural materials. Starting from given domain, boundary conditions and load cases, the optimiser proceeds to determine where structural material should be assigned. The complete process is highly automated and thus allows the designer to quickly explore the design space and generate highly intuitive and innovative structural designs.

The most commonly used approaches to solve the topology optimisation problem are the homogenisation approach [2–6] and the SIMP approach (Solid Isotropic Material with Penalisation) [7–11]. In the homogenisation approach, which was originally developed by Bendsøe and Kikuchi [2], the domain in this method is a composite material made of a periodic microstructure. Each microstructure consists of material and void. The topology optimisation problem for this approach is solved by finding the optimal distribution of the voids in the design domain. Bendsøe [7] also proposed

another technique to solve topology optimisation problem called the SIMP approach. This approach consists of defining one design variable, a fictitious density  $\rho$ , in each finite element. Thus, the elastic modulus of each element is defined as

$$E_i(x_i) = x_i^p E_0 \quad (p \geq 1) \quad (1.1)$$

$$\rho_i(x_i) = x_i \rho_0 \quad (0 \leq x_i \leq 1), \quad (1.2)$$

where  $E_0$  is the base elasticity modulus,  $p$  is a penalisation parameter,  $\rho_i$  is a local density variable,  $x_i$  is a design variable associated to the  $i^{th}$  finite element and  $\rho_0$  is the base density.

Fictitious material density variables  $\rho$  are associated with the finite elements of the topology design region. Each fictitious density can assume a value between 1 representing solid material and 0 representing a void region. The stiffness of each element material is linked to the fictitious density using the SIMP power law (1.1) such that the converged optimised design topology is driven to crisp black/white designs where most elements are at the upper or lower limit.

### 1.3 Cellular automata for topological optimisation of structures

Recently, there has been a growing interest in solving complex structural problems using massively parallel algorithms. Most available algorithms are serial in nature and require a large number of costly analyses, mostly based on the global solution approach. An alternative methodology, based on the Cellular Automata paradigm, can be implemented on both traditional and parallel hardware architectures. The CA is characterised by five properties: lattice geometry, neighbourhood, cell states, local rules of transition and boundary conditions.

The cellular automata paradigm is a mapping of discrete dynamic systems in time and space. Each cell of the discrete domain communicates only with its neighbourhood through simple local rules of transition. These rules are functions of the states of the cell itself and its neighbouring cells. Global convergence is sought by applying the local rules repetitively to the entire domain.

Kita and Toyoda [12] were among the first to use the cellular automata paradigm for solving topology optimisation problems. They constructed CA design rules to obtain two-dimensional topologies based on an Evolutionary Structural Optimisation

(ESO) approach [13–15], where the analysis of the structure is performed using the finite element method.

Another pioneering work is attributed to Gürdal and Tatting [16]. Using the CA paradigm they performed the analysis and design tasks in an integrated scheme to solve the topology design of trusses that exhibit linear and geometrically nonlinear responses. A Stress Ratio (SR) method [17] was chosen for the design update rule. The analysis rules are derived from local equilibrium, and a simple design rules that are based on fully stressed design [17] are used to size the truss members, and the cross sectional areas of the members that connect the neighbouring cells become design variables. The authors of this article demonstrated the efficiency of CA within a design environment.

The CA method was extended to encompass the design of two-dimensional continuum structures by the same authors Tatting and Gürdal [18]. The two-dimensional continuum is modelled at the cell level by a truss layout that is equivalent to the continuum cell according to an energy criterion. The relationship between the thickness of the continuum structure and the cross sectional areas of the truss members is established by equating the strain energy of the continuum cell and that of the truss cell for given nodal displacements. The local analysis rules are derived from the equilibrium condition of the cell. A fully stressed material condition is selected to construct the local design rules. Numerical examples are made and compared to GENESIS software to demonstrate the efficiency of the combined CA analysis and design.

Encouraged by the success of applying the CA paradigm to structural design, Abdalla and Gürdal [19] extended CA to design Euler-Bernoulli column for minimum weight under buckling constraint. This problem is solved using two local rules. One, the analysis rules for CA are derived by minimisation of the total potential energy in a cell neighbourhood. Two, the design rules are formulated as a local mini-optimisation problem involving force resultants. The proposed CA algorithm is shown to converge correctly to an analytical optima for a number of classical test cases.

Abdalla and Gürdal [20] have also successfully applied CA to treat the problem of topology optimisation of the elastic plate. In their work, CA design rules are formulated for the first time using rigorous optimality criteria based on SIMP material [7–11] approach. The CA analysis rules were derived from the principle of minimal total potential energy. An extension of this work was made by Setoodeh et al. [21] to combine fiber angle and topology design of anisotropic laminae. Fibre angles and density measures at each cell of a domain are updated based on the optimality criteria for the minimum compliance. Numerical examples for cantilever

plates show the robustness of the implemented algorithm based on the CA paradigm.

Recently, the CA paradigm has been applied to topology optimisation of 2D elastic continuum structures subject to in plane loads and exhibiting geometric nonlinearities Zakhama et al. [22]. The methodology applied in this paper is based on the SIMP technique which is successfully defined in the work of Abdalla and Gürdal [20]. The analysis is based on an equivalent truss model [18]. Relations between the thickness of the isotropic plate and the cross-sectional areas of the truss members are derived so that cell strain energy due to in-plane deformation is the same for both models. It builds on the method developed in Gürdal and Tatting [16] for the topology design of linear and nonlinear trusses while allowing systems large enough to simulate continuum domains. Diverse examples are presented to demonstrate the difference in topology between a linear and nonlinear analysis.

The cellular automata paradigm is also well known to be an inherently massively parallel algorithm. Slotta et al. [23] have successfully implemented Gürdal and Tatting's [16] work using standard programming languages and parallelisation libraries. The domain is decomposed into different groups of cells. Each group is assigned to a processor and the same local rules are applied for all the processors. Results demonstrate that the CA method is perfectly suited for parallel computation. Setoodeh et al. [24] proposed solving topology optimisation for a continuum structure, based on the work of Abdalla and Gürdal [19], using a pipeline parallel implementation of cellular automata on distributed memory architecture. Numerical results show that the pipeline implementation converges successfully and generates optimal designs.

## 1.4 Multigrid acceleration method for topology optimisation

In CA studies, it has been observed that the CA convergence rate deteriorates considerably as the grid is refined. This is due to the slow propagation of information across the domain. Additionally when a CA algorithm is implemented on a serial machine it loses its most attractive feature: parallelism [23, 24]. A methodology based on the Multigrid scheme is proposed in this thesis to accelerate the CA convergence process on serial machines. In earlier papers, it has been demonstrated that the CA method takes advantage of the acceleration effect of multigrid schemes [25–27]. In this thesis, the multigrid acceleration scheme is proposed as a means to accelerate the CA algorithm for traditional hardware architecture. The motivation behind this incorporation is that the cellular automata paradigm and the multigrid scheme are closely related in their nature. The main idea in the multigrid concept is to use

different discretisation levels of grids, where the iterations of a classical iterative method on the finer grid are coupled with the iterations for the correction of the solution on the coarser grids. This concept is illustrated in depth by Wesseling [28]. Hackbush [29] presents some applications of the multigrid algorithm such as solving elliptic partial differential equations and Poisson's equation.

In one of the earlier papers Maar and Shulz [30] incorporated the multigrid method to accelerate the convergence of nonlinear interior point algorithm applied to topology optimisation problems. Kim and Yoon [31] described the new concept of multi-resolution multi-scale topology optimisation. The authors formulated the design optimisation variable in a wavelet-based variable space, not in a direct density variable space. Using this method, major numerical instabilities such as mesh-dependencies and local minima are resolved. Dreyer et al. [32] presented two formulations of multigrid methods for optimisation problems: the reduced Sequential Quadratic Programming (SQP) [33–35] with multigrid solution of the linearised model equation and the simultaneous multigrids for solution of quadratic subproblems in a SQP-algorithm. Shape optimisation of turbine blades and topology optimisation of elastic structures are chosen as numerical examples for these two formulations. Kwon et al. [36] incorporated the multigrid method into a multi-scale method to improve numerical efficiency.

Only in the work of Kim et al. [25] is the multigrid method applied to accelerate the convergence of the cellular automata paradigm for structural design optimisation of a beam. In this paper, equilibrium update rules are derived for a cell-centered and a vertex centered lattice based on the minimisation of total potential energy, which makes the update rules general enough to solve any Euler-Bernoulli beam model. The design update rules are based on local optimality conditions that yield fully stressed design updates. Simple examples are used to demonstrate the methodology and various iterative aspects of the update rules are studied.

## 1.5 Design-dependent loads on the topology optimisation

Research into the optimisation of the topology of continuum structures [2] is well established and continues to attract the attention of researchers and software companies in various engineering fields. However, few researches take into account the effect of design-dependent loads on topology optimisation problems. This class of problems is present in different engineering domains: the fluid pressure loading of structures in hydrostatic problems, snow loading in civil engineering structures, and

wind and hydrodynamic loading on off-shore structures. In these problems, the location, direction, and magnitude of loading are dependent on the design of the structure and cannot be prescribed *a priori*.

Optimisation of structural topology under design-dependent loads is a comparatively recent topic. The research in this area focuses on two types of problems: design of structures under hydrostatic pressure and design of structures under transmissible loads. Recent interest in structural optimisation for design-dependent loading within the context of modern topology optimisation can be traced to the publication of the papers by Hammer and Olhoff [37] on pressure loading, and Fuchs and Moses [38] on transmissible loads.

### 1.5.1 Hydrostatic pressure loads

When structures are designed to resist hydrostatic pressure, the magnitude and the direction of load will change as the topology of the structure is evolved. More significant is the fact that the location of the applied load is defined by the interface between the material regions and the void regions. Since the location of the interface is not known *a priori*, special techniques and precautions are necessary to properly model and solve such problems.

In the work of Hammer and Olhoff [37] these problems are addressed for the two-dimensional case. The approach is based on the popular density-based approach and the SIMP model discussed earlier (see equations (1.1) and (1.2)). The pressure loading surface is identified by finding the curve of constant fictitious density corresponding to a chosen cut-off value. The pressure loads are applied on this curve and they are transformed to nodal forces within each finite element. The iso-volumetric curve is represented by a set of Bezier cubic splines and approximated by a straight line within the finite elements that it intersects.

The methodology of Hammer and Olhoff [37] is extended in the work of Du and Olhoff [39]. The authors modify the isoline technique to make the process of the identification of the loading surface more robust. The authors also replace the traditional SIMP model by an energy-based model with weighted unit cost constraints to realise the topology optimisation process using general design-dependent loads. The same authors [40] provided an extension of their two-dimensional work [39] to the three-dimensional case.

In the above cited work, the loading surface was determined from the fictitious density distribution, thus avoiding the need to introduce additional design variables



to describe the loading surface. In contrast, Fuchs and Shemesh [41] explicitly introduce a parameterised loading surface and a set of design variables, independent of the density distribution, which control the shape of the pressure interface into the problem of the topology optimisation. In order to maintain continuity and enhance numerical stability, a very low Young's modulus is assumed for any material on the pressure side and a nominal modulus is employed for the material region. Again, the interface shape is constructed using Bezier splines. The main application described in their paper is the design of dams subject to hydrostatic water pressure.

Another contribution of the design-dependent loading application for the topology optimisation problem is attributed to Chen and Kikuchi [42]. In this work, the authors propose an approach to simulate design-dependent loads using fictitious thermal loads. The topology optimisation problem is transformed from a two-phase to a three-phase material distribution problem within the design domain in which the solid, void, and hydrostatic fluid phases are optimally distributed. More recently, Sigmund and Clausen [43] have suggested a new way to solve pressure load problems in topology optimisation by using a mixed displacement-pressure formulation for the underlying finite element problem.

### 1.5.2 Transmissible loads

The idea of transmissible loads is best explained by considering the design of a bridge. The load on the bridge is known in magnitude and direction, but the exact location of the force along the line of action depends on the topology of the bridge. In contrast to hydrodynamic loading the direction of the force is not design-dependent. This is the same situation with hydrodynamic loading where the magnitude and direction of the load are known, but its point of application is the unknown surface of the structure. Design of the structural topology when the applied forces are transmissible along their line of action is first introduced within modern topology optimisation formulations by Fuchs and Moses [38]. The authors introduce design variables to parameterise load magnitudes along the line of action. The sum of the loads along the line of action is prescribed to keep the total magnitude constant. In this fashion, the net load of each line of action is given but the exact location of it is a part of the optimisation process.

The main ideas described above are not all applicable to wind or hydrodynamic water pressure loading problem addressed in this thesis. For example, wind loading cannot be represented using the method of transmissible loads, especially in three dimensions, since no wind loading is applied when there is no surface obstructing the wind. Although the iso-volumetric surface method can be applied, it is interesting

to develop a direct density based method that does not involve explicit construction of the loading surfaces [44, 45].

## 1.6 Thesis layout

In this study, the topology optimisation of continuum structures subject to dead and design-dependent loading is considered. The optimisation problem is solved using a multigrid accelerated cellular automata algorithm. The optimisation problem is also solved using the Method of Moving Asymptotes (MMA) [46–49], modified by including a line search and by changing the formula for the update of asymptotes. The multigrid accelerated cellular automata technique for topology optimisation problems is demonstrated to be a valuable tool and may reduce considerably the computational time compared to other existing methods. Moreover, including design-dependent loads into the optimisation formulation can lead to a significant increase in structural stiffness with respect to these loads and can allow the appearance of void (or holes) in the optimal structure facing the loads.

This thesis is organised as follows. In chapter 2, a general overview of cellular automata is presented followed by a CA application to topology optimisation for two and three dimensional elastic continua problems. Then, the regularisation of the topology optimisation problem by the SIMP approach is detailed. The analysis rules are derived from the principle of minimum total potential energy, and the design rules are derived based on continuous optimality criteria interpreted as local Kuhn-Tucker conditions. Finally, the effectiveness of the CA algorithm in solving topology optimisation problems is demonstrated.

A methodology based on the multigrid scheme to accelerate the CA convergence on a serial machine for two and three dimensional topology optimisation problems is presented in chapter 3. The multigrid cellular automata algorithm is demonstrated to be effective in solving topology optimisation problems in a reasonable time. The chapter is concluded by an assessment of the computational complexity of the multigrid algorithm.

Chapter 4 deals with the inclusion of wind loading in the minimum compliance topology optimisation for two and three dimensional elastic continua problems. The wind loading is introduced into the formulation using standard expressions for the drag force. The modified MMA coupled with Line Search (MMALS) method are selected as a benchmark solver for the 5<sup>th</sup> chapter. An explicit constraint that controls the intermediate densities is introduced to obtain black/white designs. It

is shown that including wind loading in the formulation has an effect on the final design.

The same framework of inclusion of wind loading in the minimum compliance, defined in chapter 4, is extended and solved using the multigrid accelerated cellular automata algorithm in chapter 5. The design update rule is based on optimality criteria for minimum compliance design. The local optimisation problem is convexified to prevent oscillation in the topology optimisation process and the explicit constraint is used to control the intermediate densities. Numerical examples demonstrate the effect of design-dependent loading onto the topology optimisation problems.

A conclusion is presented and the accomplishments of the thesis are summarised in chapter 6.

## Chapter 1

## Chapter 2

# Cellular Automata for Topology Design

### 2.1 Introduction

The cellular automata (CA) paradigm has recently been applied successfully to several structural design problems [12, 16, 18–24]. The methodology is based on local resolution using simple interaction between a lattice of cells, and the CA update is done using simple local rules. For structural design problems, two local update rules are used to solve the optimisation problem; these are the update of field variables, e.g. cell displacements, and design variables, e.g. element thicknesses. This methodology does not require a large amount of memory storage and it can be easily parallelised due to its simple structure [23, 24].

In this chapter, the cellular automata paradigm is applied to two and three dimensional linearly elastic continua topology optimisation problems. We start by presenting an introduction to cellular automata. Formulation of the design problem is posed as compliance minimisation, which is based on continuous optimality criteria interpreted as local Kuhn-Tucker conditions to derive the CA design rules [20, 21]. The optimality criteria method has been successfully used by many researchers to solve a variety of structural optimisation problems [50–53]. The topology optimisation problem is regularised using the SIMP approach [7–11]. However, the analysis rules are derived from the principle of minimum total potential energy. The approach is

then used in three-dimensional problems with some necessary technical adaptation related to the change in dimension but with no modification in the concept. Some numerical examples are developed to demonstrate the efficiency of the CA algorithm for solving topology optimisation for two and three dimensional problems.

## 2.2 Introduction to cellular automata

Cellular automata are generally attributed to John von Neumann and Stanislas Ulam around 1950 [54]. Originally Stanislas Ulam referred to cellular automata as cellular space or automata networks. John von Neumann extended Stanislas Ulam's work with the aim of building a machine able to solve complex problems [55] based on human brain operations. He thought that such level of complexity in a machine would require mechanisms of self-control and self-reproduction. Adopting a more structural approach, von Neumann was led to wonder about the properties of self-replicating systems. The objective was to determine extremely simple rules to permit the building of very complex structures by self-reproducing biological systems.

Stanislas Ulam suggested to von Neumann to use a completely discrete universe of cells. Each cell characterised by internal states, precisely as a finite number of bits. This universe of cells evolves over discrete time steps. Each cell of the domain follows the same evolution rule according to its neighbouring cell states. This discrete dynamic systems of cells as described by von Neumann is called cellular automata.

Cellular automata are discrete dynamic systems in time and space, and a particular case of automata network. The automata network can be updated either synchronously or sequentially. For the synchronous mode, which is called the parallel mode, the entire sites are updated simultaneously in a discrete time step. The sequential mode is applied only for finite networks: the entire sites are updated successively in a prescribed order. The domain size can be increased without increasing the computational time of the entire system, by associating each processor to a set of cells. Cellular automata can be classified among the most powerful computational methods due to their parallelism.

The cellular automata technique has been successfully used in different fields such as modelling natural systems including biological systems, diffusion of gaseous systems, solidification, crystallisation, turbulence and hydrodynamic flow [56]. In most previous applications, the cellular automata can be viewed as a solution technique (strategy) for partial differential equations, which describe continuous dynamic systems. The idea is to build complex systems using extremely simple rules, rather

than using complicated equations. In other words, it is not the complicated equations that will be used to build complex systems, but the complexity will emerge as a result of simple interactions between cells. The cellular automata rule is a microscopic description of reality and an adequate rule can generate a realistic macroscopic behaviour. Using a sufficiently large number of cells, it is possible to represent a complex response governed by large number of nonlinear equations. The cellular automata are characterised by five properties: lattice geometry, neighbourhood, cell states, local rules of transition and boundary conditions.

### 2.2.1 Cell states

A cell is the basic element of cellular automata; it is a processor which allows storing and modification of a state. In the easiest case, the cell state consists of one bit; each cell can be either in state 1 or in state 0. The cell state can be more complex and consist of several attributes, but the number of possible states for a cell remains limited. The update of these states is done using a local rule of transition.

### 2.2.2 Lattice geometry

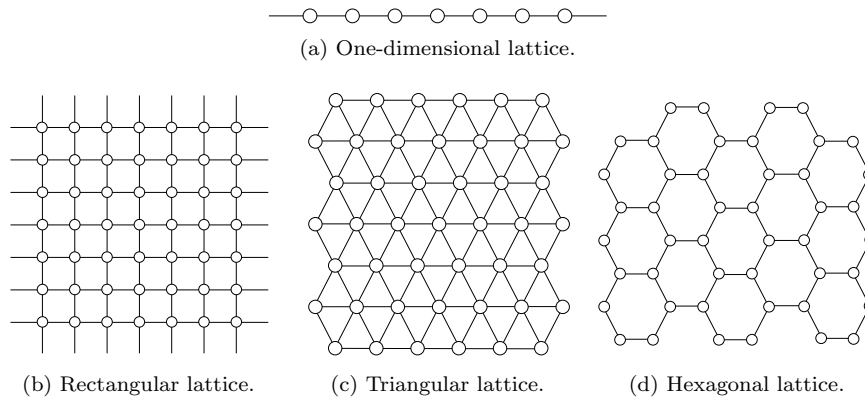


Figure 2.1: Lattice geometry.

The form of network depends on the system being modelled. In most of the cases the form of the network is typically one, two, or three dimensional. A two-dimensional lattice may consist of the identical juxtaposition of hypercubes as shown in figure

## Chapter 2

2.1. The three-dimensional lattice can be built by juxtaposition of several two-dimensional lattices with the same distance as between two cells in the plane. The network structures are not limited to one rectangular lattice but can take any other form such as triangular and hexagonal lattices (see figures 2.1(d) and 2.1(d)). For example, Wolfram [57,58] uses a hexagonal lattice of cells to model the fluid domain.

### 2.2.3 Neighbourhood

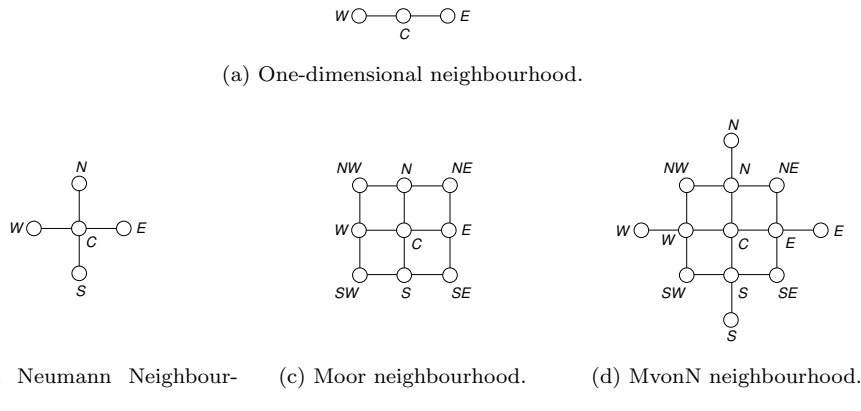


Figure 2.2: CA Neighbourhood.

The dynamics of cellular automata requires us to define the concept of cell neighbourhood: this is the set of cells that can interact with each cell of the domain at a given time. The cell neighbourhood contains all the necessary information required to update the cell states at each time step. For a two-dimensional lattice, the neighbourhoods most commonly used in the literature are von Neumann and Moor neighbourhoods (see figure 2.2). The von Neumann neighbourhood is a neighbourhood where each cell communicates only with the four cells that are orthogonally adjacent to it. For a Moor neighbourhood, each cell communicates with all eight cells surrounding it. More neighbourhood types can be used other than von Neumann and Moor neighbourhoods where the choice depends on the system that is being modelled. For example, the “MvonN neighbourhood” combines the eight cells of the Moor neighbourhood with the four cells orthogonally one space away from the current cell.



### 2.2.4 Local rules of transition

In computer implementation of CA, the update rules are applied for each cell of the domain using a certain function  $f$ . The arguments of this function are the cell neighbourhood states and the cell state in question, the output value is the new state of this cell. For example, for a von Neumann neighbourhood, the function  $f(C, E, W, N, S)$  has five arguments. It returns the state value of the cell  $C$  at the moment  $(t + 1)$ . Since the update rule can be applied simultaneously for each cell of the domain, the input arguments for this case are the state values of the cells at the moment  $(t)$  and at the moment  $(t + 1)$  all the cells have new state values.

### 2.2.5 Boundary conditions

The discretised design domain being finite, it is necessary to specify the concept of neighbourhood for the cells at the edges of the domain. A different local rule of transition then needs to be applied to the cells at the edges. To rather allow a unique (or single) local rule to be applicable to all cells, the grid is extended beyond the boundary of the actual design domain. Several boundary conditions can be defined depending on the nature of the system being studied such as (see figure 2.3)

- Periodic: for a two-dimensional lattice, the left and right edges and the upper and lower edges are connected, the domain can then be seen as a torus
- Fixed: the neighbourhood is extended with a set of virtual cells, and a constant state value is assigned for these cells
- Adiabatic: the neighbourhood is extended by duplicating the cells at the edges
- Reflecting: the neighbourhood is extended by duplicating the neighbourhood cell in the opposite direction

## 2.3 Minimum compliance design optimality criteria

The problem of topology design is posed according to minimal compliance aimed at finding the optimal distribution of material subject to the volume constraint. Consequently, the problem is equivalent to the problem of minimising the elastic strain

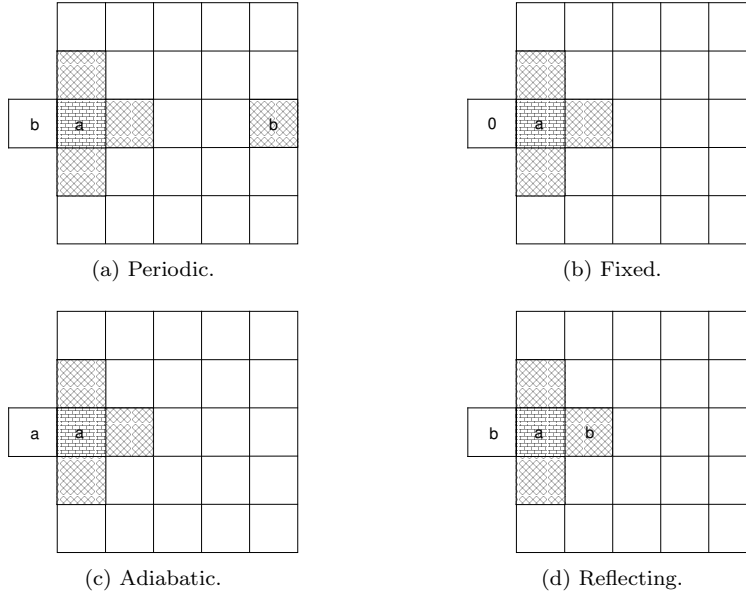


Figure 2.3: Boundary conditions.

energy of a structure, or maximising its total potential energy  $\Pi$  at equilibrium. Thus, the optimisation problem can be written as

$$\min_{\rho} W(\rho, \mathbf{u}^*) \quad \text{or} \quad \max_{\rho} \Pi(\rho, \mathbf{u}^*), \quad (2.1)$$

under the constraints:

$$\mathbf{g}(\rho) \leq 0, \quad (2.2)$$

and the volume constraint:

$$V = \int_{\Omega} \rho d\Omega \leq \eta V_{\Omega}, \quad (2.3)$$

where  $\rho$  is local density of material (or material distribution),  $\Omega$  is the prescribed design domain,  $\mathbf{u}^*$  is the displacement vector at equilibrium and  $\mathbf{g}$  are the local constraints. The volume  $V$  of the structure is limited to an available fraction  $\eta$  of the total volume of the design material domain  $V_{\Omega}$ .

The total potential energy at equilibrium is written as follows:

$$\Pi = \int_{\Omega} \Phi(\Gamma, \rho) d\Omega - \int_{\Gamma} \mathbf{t} \cdot \mathbf{u}^* \partial\Omega, \quad (2.4)$$

where  $\mathbf{t}$  is the applied surface traction,  $\Phi$  is the strain energy density of the structure and  $\Gamma$  is the generalised strain vector.

Thus, the Lagrangian for this problem is written as follows:

$$\mathcal{L} = \int_{\Omega} [-\Phi + \mu \cdot (\rho - \eta + \mathbf{c}^2) + \lambda \cdot (\mathbf{g} + \mathbf{s}^2)] d\Omega, \quad (2.5)$$

where  $\mu$  is a vector Lagrange multiplier associated with the point constraint (2.3) and  $\mathbf{c}$  is the corresponding slack vector variable,  $\lambda$  is a vector of Lagrange multipliers associated with the local constraints and  $\mathbf{s}$  is the corresponding slack vector.

Setting the variation of the Lagrangian function to zero, we obtain the following optimality conditions [21]:

1. Stationarity condition is written as

$$-\frac{\partial \Phi}{\partial \rho} + \mu + \lambda \cdot \frac{\partial \mathbf{g}}{\partial \rho} = 0. \quad (2.6)$$

2. Constraints

$$V \leq \eta V_{\Omega}, \quad (2.7)$$

$$\mathbf{g}(\rho) \leq 0. \quad (2.8)$$

3. Complementarity conditions

$$\lambda_i s_i = 0, \quad (2.9)$$

$$\mu_i c_i = 0, \quad (2.10)$$

$$\lambda_i \geq 0, \quad (2.11)$$

$$\mu_i \geq 0. \quad (2.12)$$

We introduce the Legendre transformation to define the complementary energy density, given by

$$\hat{\Phi}(\sigma, \rho) = \sigma \cdot \Gamma - \Phi(\Gamma, \rho), \quad (2.13)$$

where  $\sigma$  is the generalised stress defined by

$$\sigma = \frac{\partial \Phi}{\partial \Gamma}. \quad (2.14)$$

From the properties of Legendre transformations (2.13), the derivative of the complementary energy density with respect to the density  $\rho$  at constant stress can be written as

$$\left. \frac{\partial \hat{\Phi}}{\partial \rho} \right|_{\sigma} = \sigma \cdot \frac{\partial \Gamma}{\partial \rho} - \frac{\partial \Phi}{\partial \Gamma} \cdot \frac{\partial \Gamma}{\partial \rho} - \left. \frac{\partial \Phi}{\partial \rho} \right|_{\Gamma} \frac{\partial \rho}{\partial \rho}. \quad (2.15)$$

## Chapter 2

Consequently, from the definition of the generalised stress (2.14), it can be shown that:

$$\left. \frac{\partial \hat{\Phi}}{\partial \rho} \right|_{\sigma} = - \left. \frac{\partial \Phi}{\partial \rho} \right|_{\Gamma}. \quad (2.16)$$

This equation shows that maximising the strain energy density for a given strain and minimising the complementary energy at constant stress are equivalent.

Using (2.16), we can infer that the first order conditions (2.6), (2.8), (2.9) and (2.11) are equivalent to the optimality conditions of the following local minimisation problem:

$$\min_{\rho} \hat{\Phi}(\sigma, \rho) + \mu \rho, \quad (2.17a)$$

subject to,

$$\mathbf{g}(\rho) \leq 0. \quad (2.17b)$$

The Lagrange multiplier  $\mu$ , associated with the volume constraint, is the only global quantity that is involved in this local problem. It serves in updating the material densities in the domain. It is updated at the global level by satisfying the total volume constraint.

The topology optimisation design of two and three dimensional elastic continua is considered in this thesis. The material behaviour is assumed to be linear elastic and to obey plane stress law:

$$\mathbf{N} = \overline{\mathbf{Q}} \cdot \Gamma, \quad (2.18)$$

where  $\mathbf{N}$  is the vector of in-plane stress resultants and  $\overline{\mathbf{Q}}$  is the reduced stiffness matrix obtained after normalisation with respect to the initial elastic modulus  $E_0$ .

In the specialisation of the SIMP method, the local stiffness of the structure is a function of a fictitious local density distribution  $\rho$ , which is chosen as a design variable ( $0 < \rho < 1$ ). Thus, the expression of the strain energy density is written:

$$\Phi = \frac{1}{2} \rho^p \Gamma \cdot \overline{\mathbf{Q}} \cdot \Gamma, \quad (2.19)$$

where  $p \geq 1$  is a penalisation parameter that is introduced to lead the design to a black and white topology, by assigning sufficiently high values to  $p$ . The effect of this parameter is to penalise intermediate values of the density without causing numerical difficulties. The recommended value is  $p = 3$ .

With these definitions, the generalised stresses are obtained easily from (2.14):

$$\sigma = \rho^p \mathbf{N}, \quad (2.20)$$

and then the strain vector  $\Gamma$  can be expressed as function of the generalised stress  $\sigma$  as

$$\Gamma = \frac{1}{\rho^p} \overline{\mathbf{Q}}^{-1} \cdot \sigma. \quad (2.21)$$

Therefore, the complementary energy density  $\hat{\Phi}$  is obtained using (2.13), (2.19) and (2.21),

$$\hat{\Phi} = \frac{1}{2} \frac{1}{\rho^p} \sigma \cdot \overline{\mathbf{Q}}^{-1} \cdot \sigma. \quad (2.22)$$

Thus, the complementary energy density  $\hat{\Phi}$  can be written as

$$\hat{\Phi} = \frac{\Phi^*}{\rho^p}, \quad (2.23)$$

where

$$\Phi^* = \frac{1}{2} \sigma \cdot \overline{\mathbf{Q}}^{-1} \cdot \sigma. \quad (2.24)$$

The optimisation problem (2.17) is reduced to

$$\min_{\rho} \frac{\Phi^*}{\rho^p} + \mu \rho, \quad (2.25a)$$

subject to,

$$\epsilon \leq \rho \leq 1, \quad (2.25b)$$

where  $\epsilon > 0$  is a very small number, set as a lower bound on  $\rho$  to avoid numerical instability that may result from structural underdeterminacy when zero density is allowed.

This one-dimensional minimisation problem is convex. Its unconstrained solution is obtained analytically:

$$\hat{\rho} = \left( \frac{\Phi^*}{\bar{\mu}} \right)^{\frac{1}{p+1}}, \quad (2.26)$$

where  $\bar{\mu} = \mu/p$  is a modified Lagrange multiplier that has units of energy density.

The solution of the local minimisation problem (2.25) is given by

$$\rho = \begin{cases} \hat{\rho} & \epsilon < \hat{\rho} < 1 \\ \epsilon & \hat{\rho} \leq \epsilon \\ 1 & \hat{\rho} \geq 1 \end{cases} \quad (2.27)$$

## 2.4 CA implementation for structural design

In the previous section, optimality based local rules for updating the material density were derived. In this section, the CA discretisation of two and three dimensional structural domains is described. The elastic continuum domain (see figure 2.4(a)) is discretised by a lattice of regular cells which are equally spaced in the  $x$  and  $y$  directions (see figure 2.4(b)), or  $x$ ,  $y$  and  $z$  for a three-dimensional structural domain (see figure 2.4(c)). Traditional Moore neighbourhood is used to define the connectivity of the lattice as shown in figures 2.4(d) and 2.4(e). Each cell  $i$  communicates with its neighbours by a local rule and its state is denoted as  $\phi_i^k$  where  $k$  is the iteration number. For topology design in two and three dimensions, the state of the  $i^{th}$  cell is defined as

$$\phi_i = \{(u_{i(1\dots m)}), (f_{i(1\dots m)}), \rho_i\}, \quad (2.28)$$

where  $m$  corresponds to the dimensionality of the domain,  $m = 2$  or  $3$  for two or three dimensional, respectively. The components  $(u_{i(1\dots m)})$  are the cell displacements in the directions  $(1\dots m)$ ,  $(f_{i(1\dots m)})$  the external forces acting on the  $i^{th}$  cell in the respective  $(1\dots m)$  directions. Each cell of the discretised domain has its own density measure  $\rho_i$  at the node point independently of the densities of the elements that define the neighbourhood.

The update of the cells can be done simultaneously, which corresponds to the Jacobi scheme, as follows:

$$\phi_C^{k+1} = f(\phi_C^k, \phi_{NM}^k), \quad (2.29)$$

or sequentially, which corresponds to Gauss-Seidel scheme:

$$\phi_C^{k+1} = f(\phi_C^k, \phi_M^{k+1}, \phi_{NM}^k), \quad (2.30)$$

where  $M$  is the set of neighbouring cells whose states have been modified in the current iteration and  $NM$  is the set of remaining cells, which have not yet been modified.

The Gauss-Seidel method is used for the analysis update. For the design update, the Jacobi method is the appropriate one to use to preserve the symmetry on the solution [20].

### 2.4.1 Analysis update rule

The local analysis rule is derived from the equilibrium condition of the cell. The total potential energy associated with a cell is the sum of the strain energy in each

# Cellular Automata for Topology Design

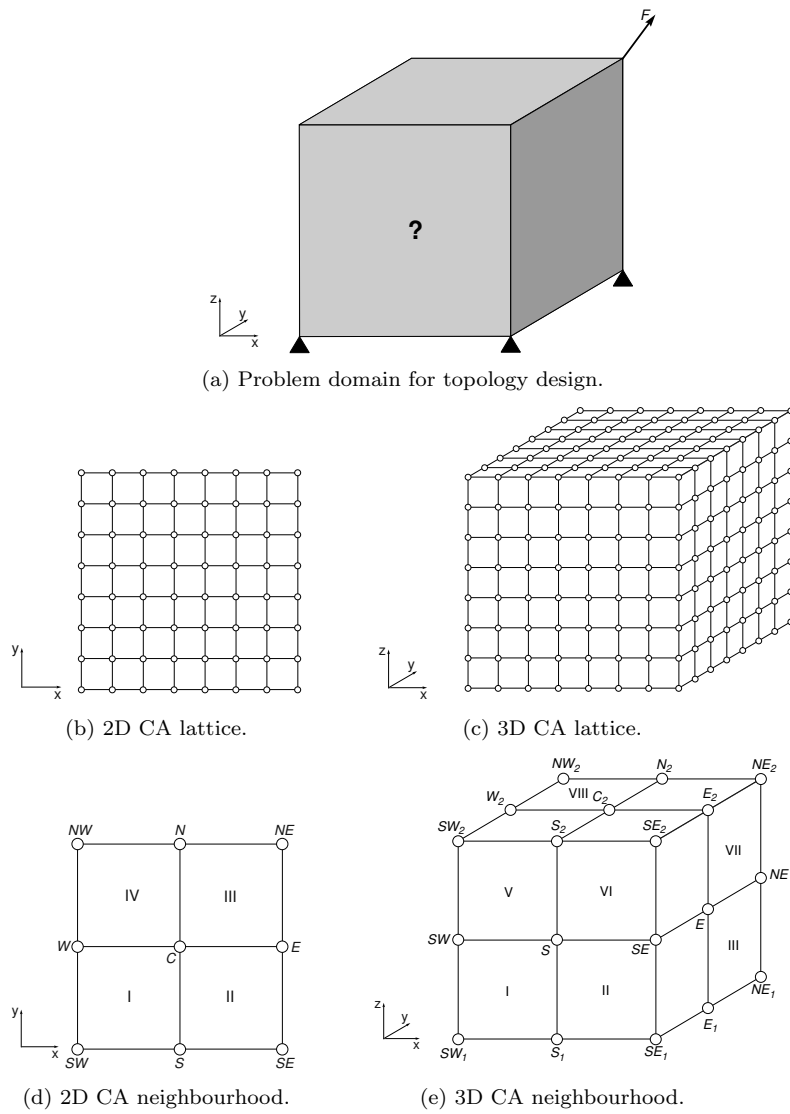


Figure 2.4: CA domain.

element of the neighbourhood structure, added to the potential energy due to the

external forces applied directly to the cell:

$$\Pi_i = \sum_{k=1}^{N_{element}} U_{i_k} - \mathbf{f}_i \cdot \mathbf{u}_i, \quad (2.31)$$

where  $N_{element}$  is the number of elements surrounding a cell,  $U_{i_k}$  is the strain energy for the  $k^{th}$  element,  $\mathbf{f}_i$  is the applied force vector and  $\mathbf{u}_i$  is the displacement vector for all the cell's neighbourhood including the cell itself.

The strain energy of an element is expressed in terms of the strain energy of the base material as follows:

$$U_k = \bar{\rho}^p \tilde{U}, \quad (2.32)$$

where

$$\tilde{U} = \frac{1}{2} \int_{element} \mathbf{\Gamma} \cdot \bar{\mathbf{Q}} \cdot \mathbf{\Gamma} \, dx dy dz, \quad (2.33)$$

is the strain energy of the base material,  $\mathbf{\Gamma}$  is the small-strain tensor, and  $\bar{\mathbf{Q}}$  is the reduced in-plane stiffness for isotropic materials. The elements densities  $\bar{\rho}$  are obtained by an average density interpolation [20] given by

$$\frac{1}{\bar{\rho}^p} = \frac{1}{N_{cell}} \sum_{i=1}^{N_{cell}} \frac{1}{\rho_i^p}, \quad (2.34)$$

where  $\rho_i$ 's are the density measures of the cells surrounding the element, and  $N_{cell}$  is the number of cells defining the element. For the two-dimensional neighbourhood structure  $N_{cell} = 4$  and for the three-dimensional neighbourhood structure  $N_{cell} = 8$ . Using this scheme checkerboard patterns are suppressed automatically during the optimisation process.

Thus, the equilibrium equations are obtained by minimising the total potential energy with respect to the cell displacements:

$$\min_{\mathbf{u}_C} \Pi_i. \quad (2.35)$$

The resulting equilibrium equations for each cell are written in a residual form:

$$\mathbf{R}_C(\mathbf{u}_C, \mathbf{u}_N) = \left\{ \begin{array}{c} \mathbf{G}_C(\mathbf{u}_C, \mathbf{u}_N) \\ \mathbf{G}_N(\mathbf{u}_C, \mathbf{u}_N) \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{f}_C \\ \mathbf{f}_N \end{array} \right\} = 0, \quad (2.36)$$

where  $\mathbf{u}_C$  and  $\mathbf{u}_N$  are the displacement vectors of the cell and the neighbourhood, respectively,  $\mathbf{G}_C$  and  $\mathbf{G}_N$  are the vectors of the internal forces,  $\mathbf{f}_C$  and  $\mathbf{f}_N$  are the vector of the applied forces relative to the cell and the vector of the internal forces relative to the neighbourhood, respectively.



Differentiating the vector  $\mathbf{R}_C$  with respect to the components of  $\mathbf{u}_C$ , the linear stiffness matrix can be written as

$$\mathbf{K} = -\frac{\partial \mathbf{R}_C}{\partial \mathbf{u}_C}(\mathbf{u}_C, \mathbf{u}_N). \quad (2.37)$$

The stiffness matrix  $\mathbf{K}$  can also be expressed as the Hessian of the total potential energy:

$$\mathbf{K}_{pq} = \frac{\partial^2 \Pi_i}{\partial u_p \partial u_q}. \quad (2.38)$$

Thus, the cell displacements are updated as follows:

$$\mathbf{u}_C^{t+1} = \mathbf{u}_C^t + \Delta \mathbf{u}_C, \quad (2.39)$$

$$\Delta \mathbf{u}_C = (\mathbf{K}_C)^{-1} \cdot (\mathbf{G}_C(\mathbf{u}_N^{t+1}) + \mathbf{f}_C), \quad (2.40)$$

where  $\mathbf{K}_C$  is the cell stiffness matrix,  $\mathbf{K}_C$  is a  $(2 \times 2)$  or  $(3 \times 3)$  matrix for two or three dimensional cases, respectively (see Appendix A for details).

## 2.4.2 Design update rule

The update of each cell density of the continuum structure is done using (2.27) where  $\Phi^*$  can be written as an average among the  $N_{element}$  elements of the Moore neighbourhood structure:

$$\Phi_c^* = \frac{1}{n V_C} \sum_{i=1}^{N_{element}} \rho_i^{-2p} \tilde{U}_i, \quad (2.41)$$

where  $n$  is the number of non-shadow elements with nonzero density. Since the CA algorithm should handle irregular domains, some cells will have shadow neighbours that lie outside the computational domain. Shadow cells are treated by setting their density  $\rho$  to zero. The value  $V_C$  is the volume of a cell,  $V_C = a^2$  or  $a^3$  for two or three dimensional case, respectively, and  $a$  is the distance between two immediate neighbour cells.

## 2.4.3 Updating the Lagrange multiplier

The proposed solution algorithm is of the primal-dual type. Therefore, the Lagrange multiplier associated with the volume constraint plays a central role in the iterative

process (see algorithm 2.1). In each iteration its value  $\bar{\mu}$  in equation (2.26) is modified according to a simple update rule derived using Newton's method. The volume constraint is obtained by setting to zero the derivative of the Lagrangian function (2.5) with respect to  $\bar{\mu}$ :

$$\frac{\partial \mathcal{L}}{\partial \bar{\mu}} = 0. \quad (2.42)$$

The volume constraint, (2.3), is approximated as

$$\sum_{cells} \rho_C V_C - \eta \sum_{cells} V_C = 0. \quad (2.43)$$

Although there are several iterative methods to solve equation (2.42), the Newton-Raphson method is chosen to solve it, due to its quadratic convergence. Thus, the update rule of the Lagrange multiplier, based on the solution given by (2.27), is as follows:

$$\bar{\mu}^{k+1} = \bar{\mu}^k + \Delta \bar{\mu}, \quad (2.44)$$

where,

$$\Delta \bar{\mu} = - \frac{\frac{\partial \mathcal{L}}{\partial \bar{\mu}}}{\frac{\partial^2 \mathcal{L}}{\partial \bar{\mu}^2}}. \quad (2.45)$$

#### 2.4.4 Cellular automata scheme

In this chapter, the analysis and design iterations are nested. A flowchart of the CA design algorithm is presented in 2.1. Starting from a structure with zero displacements and from densities set to volume fraction  $\eta$ , analysis updates are performed repeatedly until the norm of the force imbalance (residual (2.46)) reaches a pre-specified tolerance  $\varepsilon_r$ .

$$\mathbf{R} = \sum_{cells} \mathbf{R}_C. \quad (2.46)$$

Next, the design is updated over the whole domain, then the volume constraint is checked. If the volume constraint is not satisfied, the Lagrange multipliers are updated and so is the design. The process continues until the relative difference between five successive compliance values is less than a pre-specified tolerance  $\varepsilon_c$  and the variation in cell densities is less than a tolerance  $\varepsilon_d$ .

---

**Algorithm 2.1** CA design algorithm

---

```

input
   $\mathbf{u}, \mathbf{f}, \rho$ 
output
   $\mathbf{u}, \rho$ 
% Initialisation
 $k = 0, \mu^{(0)}$ 
repeat
  for each element do
    % Element density interpolation
     $\bar{\rho}_e^{(k+1)}$ 
  end for
  repeat
    for each cell do
      % Analysis update rule
       $\mathbf{u}_C^{(k+1)}$ 
    end for
    until  $\frac{\|\mathbf{R}\|}{\|\mathbf{R}_0\|} \leq \epsilon_r$ 
    repeat
      for each cell do
        % Design update rule
         $\rho_C^{(k+1)}$ 
        if  $\left| \frac{V^{(k+1)}}{\eta V_G} - 1 \right| > \Delta V$  then
          % Update Lagrange Multiplier
           $\mu^{(k+1)}$ 
        end if
      end for
      until  $\left| \frac{V^{(k+1)}}{\eta V_G} - 1 \right| \leq \Delta V$ 
       $W^{(k+1)} = \mathbf{u}^{(k+1)} \cdot \mathbf{f}$ 
       $k = k + 1$ 
    until (five successive  $\left| \frac{W^{(k+1)}}{W^{(k)}} - 1 \right| \leq \epsilon_c$ ) and  $(|\rho^{(k+1)} - \rho^{(k)}| \leq \epsilon)$ 

```

---

### 2.4.5 Multiple load cases

In many fields the structures are subjected to multiple load cases such as in aerospace, civil engineering, and the automotive industry. Multiple loads are handled in the present work by minimising the weighted average of the compliances of all load cases.

The optimisation formulation including multiple load cases is as follows:

$$\min_{\rho} \sum_{k=1}^M \omega_k W_k(\rho, \mathbf{u}_k^*), \quad (2.47a)$$

subject to,

$$V = \int_{\Omega} \rho d\Omega \leq \eta V_{\Omega}, \quad (2.47b)$$

$$\epsilon \leq \rho \leq 1, \quad (2.47c)$$

where  $M$  corresponds to the number of load cases and  $\omega_k$  is the weighting factor. These parameters,  $\omega_k$ , are normalised so that their sum is equal to one.

In this formulation the displacement fields for each load case are independent, therefore, based on the formulation for a simple load case (2.25), the problem (2.47) becomes:

$$\min_{\rho} \sum_{k=1}^M \omega_k \frac{\Phi_k^*}{\rho^p} + \mu\rho, \quad (2.48a)$$

subject to,

$$\epsilon \leq \rho \leq 1. \quad (2.48b)$$

## 2.5 Numerical examples

In this section, we present some examples of two and three dimensional topology problems to show the effectiveness of the proposed cellular automata scheme and the capabilities of this method. The algorithm described previously (see algorithm 2.1) is implemented in a Linux C++ environment. In all runs the penalisation parameter is set by default to 3. The tolerance for the residual ratio is set to  $10^{-2}$ , the design tolerance is set to 0.05 and a lower bound of  $10^{-3}$  is adopted for the density. The relative compliance tolerance is fixed at  $10^{-3}$ . The value of the Lagrange multiplier is calculated at each redesign step to satisfy the volume constraint within  $10^{-3}$ . The Poisson ratio is set to 0.3 and the Young modulus  $E$  used is 1000 N/mm<sup>2</sup>.

The chosen problem is a classical cantilever problem with a tip load  $P = 100$  N applied at the bottom and clamped from the left edge (see figure 2.5(a)). The plate

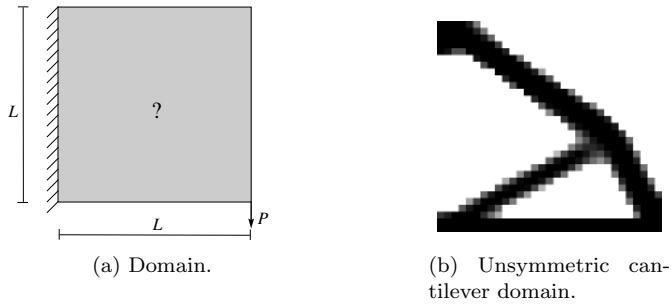


Figure 2.5: Unsymmetric cantilever 2D problem.

dimensions are 300 mm in length, 300 mm in width and 100 mm in thickness. The two-dimensional domain is discretised with  $31 \times 31$  cells and the volume fraction is set to  $\eta=0.3$ .

The two-dimensional optimum topology obtained for this problem is given in figure 2.5(b). Note that the optimal structure is a truss of three members that carry the tip load. The solution is obtained after 46 design iterations in about 18 seconds. The compliance of the final design is 4.57 Nmm. Figure 2.6 shows the compliance history and the evolution of the solution with respect to the design iterations. The compliance convergence is reasonably smooth.

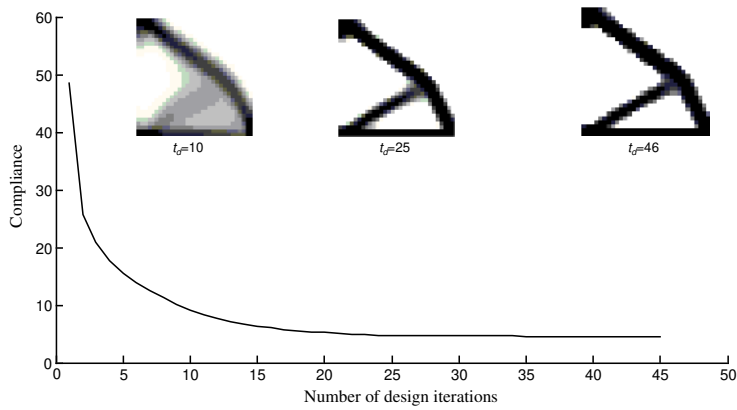


Figure 2.6: Compliance and evolution of unsymmetric cantilever 2D problem.

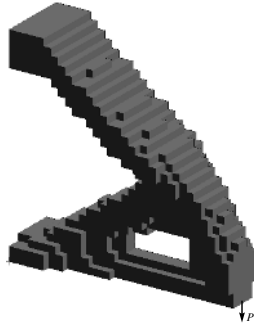


Figure 2.7: Unsymmetric cantilever 3D topology.

The same problem is studied for the three-dimensional case. The domain is discretised using  $31 \times 11 \times 31$  cells. The volume fraction and the tip load are kept the same. Figure 2.7 illustrates the optimum topology for the three-dimensional case. The solution is converged after 36 design iterations in about 826 seconds with a compliance of 7.84 Nmm. The topology obtained for the three-dimensional case is seen to be very similar to the topology obtained for the two-dimensional case.

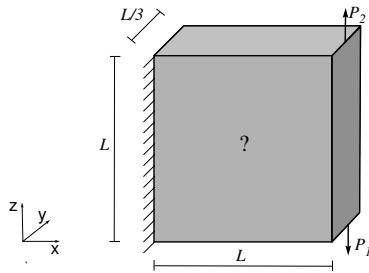


Figure 2.8: Cantilever domain for two load cases.

The classical cantilever problem is extended to study two load cases for two and three dimensional cases. The problem is defined as shown in figure 2.8. The loads have the same magnitude as defined for the one load case problem. The weighting factors are chosen to be equal to 0.5. For the two and three dimensional domains the discretisation and the volume fraction are kept the same as before. The topology for two and three dimensional cases are represented in figures 2.9(a) and 2.9(b), respectively. The compliances for both optimum structures are 7 Nmm for the two-

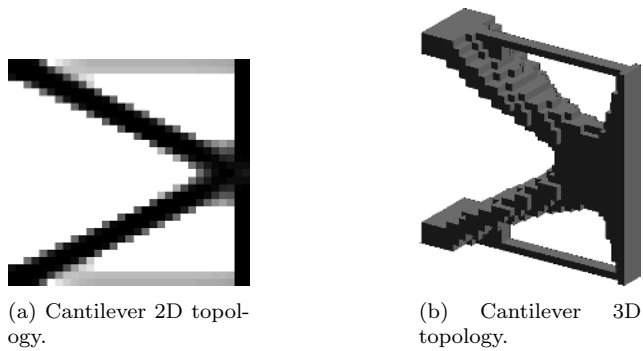


Figure 2.9: Cantilever for two load cases.

dimensional problem and 9.27 Nmm for the three-dimensional problem. The higher compliance in the 3D problem is obviously due to the larger set of possible topologies allowed by the variability of the thickness.

## 2.6 Concluding remarks

A cellular automata implementation for two and three dimensional topology optimisation of linear elastic structures has been presented. The optimality criteria, interpreted as local Kuhn-Tucker condition, were used to construct the local design rule and the analysis rule was derived from the principle minimum total potential energy. Multiple load cases are considered in the optimisation formulation. Some examples have been treated that illustrate the successes of the CA technique in solving topology optimisation problems. When CA is implemented on a serial machine it loses its most attractive feature: parallelism. The propagation of the information has to travel from neighbour to neighbour throughout the domain which causes a deterioration in rate of convergence. A traditional method to accelerate the rate of convergence, for a serial machine is to use the multigrid scheme [25], which is discussed in the following chapter.

## Chapter 2



## Chapter 3

# Multigrid Accelerated Cellular Automata

### 3.1 Introduction

The convergence of the cellular automata algorithm described in the previous chapter (see algorithm 2.1) for topology optimisation problems deteriorates considerably as the number of cells is increased. This deterioration is due mainly to the slow propagation of information across the domain. Each cell of a domain communicates only with its direct neighbours. This means, for each iteration, the cell level information where the loads are applied has to travel from neighbour to neighbour. As the grid is refined, it takes longer and longer for cell level displacements to propagate to the support and reach “steady state” values, which leads to a poor rate of convergence. A traditional solution to this problem is to use Multigrid (MG) acceleration scheme [23].

In this chapter, an alternative methodology based on the multigrid scheme is proposed to accelerate the CA convergence on a serial machine for two and three dimensional topology optimisation problems. The methodology of multigrid scheme and cellular automata paradigm are closely related in their nature. The concept of multigrid is to use different discretisation levels of grids, in which the iterations of the cellular automata algorithm performed on the finest grid are coupled with the iterations of the correction solution on the coarse grids. The topology optimisation

problem is regularised using the traditional SIMP approach [7–11] as described in chapter 2. The update rules used in this chapter are the ones derived in chapter 2. Two versions of the algorithm are implemented: a baseline multigrid for analysis acceleration and a full multigrid design algorithm. In the context of this study full multigrid refers to acceleration of both the analysis and design iterations. It is shown that the multigrid accelerated cellular automata scheme is a powerful tool for solving topology optimisation problems. This is demonstrated quantitatively by comparing the convergence time of the multigrid algorithm for different discretisation levels, with that of the same design algorithm where the analysis is performed by a commercial finite element code.

## 3.2 Idea of multigrid

The convergence rate of a classical iterative schemes, such as Gauss-Seidel and Jacobi, for solving linear systems deteriorates considerably as the number of variables increases. The iteration of these schemes acts with the greatest efficiency on propagation/convergence of short wavelength, high frequency components of the field information, while it is the components of low frequency information which persist that destroy the rate of convergence [59]. In order to improve the performance of an iterative procedure, an initial approximation of the solution can be used, for example, by relaxation on a coarse grid using a classical iterative method. Since the variables on a coarse grid are fewer, the low frequency components of the field information can be reduced without losing much precision and the computational cost for one relaxation is also much smaller than that on the fine grid. This relaxation method can then be used in order to obtain a better approximation for the finest grid solution.

The basic idea underlying the multigrid implementation is to use different discretisation levels of grids (see figure 3.1); where the iterations of the classical iterative method (or the CA analysis method presented in this thesis) on the finest grid are coupled with the iterations of the correction solution on the coarser grids [28]. It is well known that the classical iterative methods act directly on the high frequency components in the error  $\mathbf{e} = \mathbf{u} - \mathbf{v}$  of an approximation  $\mathbf{v}$  for the solution  $\mathbf{u}$  of the equation  $\mathbf{A} \cdot \mathbf{u} = \mathbf{b}$ . The error  $\mathbf{e}$  which is obtained from the residual equation  $\mathbf{A} \cdot \mathbf{e} = \mathbf{r}$  where  $\mathbf{r} = \mathbf{b} - \mathbf{A} \cdot \mathbf{v}$  can be eliminated after a few iterations on the finest grid. However, after few iterations on the finest grid the convergence rate deteriorates due to the low frequency components [59]. By using the relaxation scheme on a coarse grid one obtains an approximation to the error  $\mathbf{e}$  which corrects the fine grid approximation solution, thus the low frequency components will be eliminated.

The coarse grid correction scheme is defined as follows:

- Relax on  $\mathbf{A} \cdot \mathbf{u} = \mathbf{b}$  over  $\Omega^k$  in order to obtain  $\mathbf{v}^k$ , where  $\Omega^k$  is a grid finer than  $\Omega^{k-1}$ .
- Compute the residual  $\mathbf{r} = \mathbf{b} - \mathbf{A} \cdot \mathbf{v}^k$ .
- Relax on  $\mathbf{A} \cdot \mathbf{e} = \mathbf{r}$  over  $\Omega^{k-1}$  to obtain an approximation to the error  $\mathbf{e}^{k-1}$ .
- Correct the approximation obtained on  $\Omega^k$  with the error estimate on  $\Omega^{k-1}$  by  $\mathbf{v}^k = \mathbf{v}^k + \mathbf{e}^{k-1}$ .

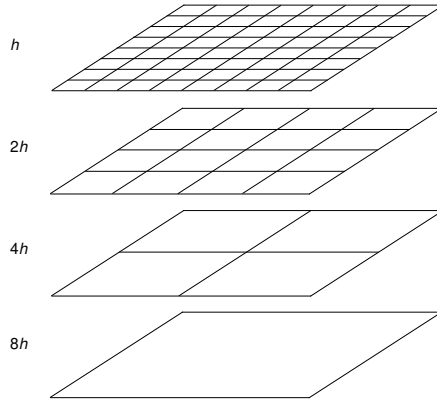


Figure 3.1: Multi-level grids for MG for two-dimensional problems.

In this work, the multigrid scheme is used to accelerate the CA convergence for two and three dimensional topology optimisation problems. The main difference between the MG design algorithm described in this work and the CA design algorithm described in Abdalla and Gürdal [20] (see chapter 2) is the acceleration of the analysis convergence. The strategies most used to visit the different grids are the **V** and **W** cycles [28]. Figure 3.2 shows the order in which the grids are visited. A parameter  $\gamma_k$  represents the number of visiting times to a grid,  $\gamma_k = 1$  is assigned for the **V** cycle and  $\gamma_k = 2$  is assigned for the **W** cycle, respectively. A dot filled circle is used to represent a smoothing operation which corresponds to few CA analysis iterations.

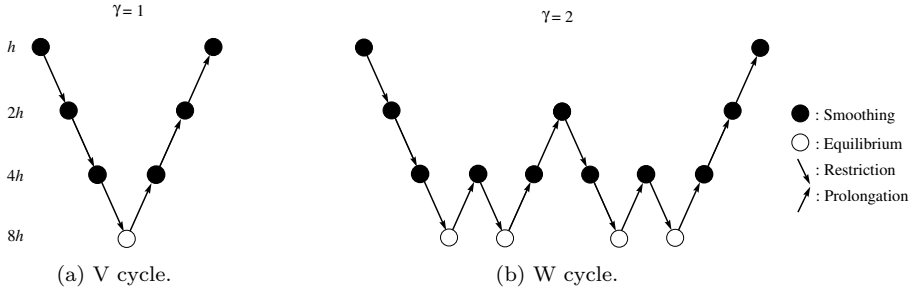


Figure 3.2: Multigrid strategies.

### 3.3 Transfer operators

The relaxation scheme, also called multigrid scheme, is based on transformation operators between coarse and fine grids. Two operators, called prolongation and restriction operators, define the multigrid scheme (see figure 3.2). The prolongation, or interpolation, operator denoted by  $\mathbf{I}_{k-1}^k$  transforms functions from coarse grid  $\Omega^{k-1}$  to a fine grid  $\Omega^k$ . It maps the error  $\mathbf{e}^{k-1}$  obtained from the coarse grid  $\Omega^{k-1}$  onto the fine grid  $\Omega^k$ . The restriction operator, denoted by  $\mathbf{I}_k^{k-1}$ , is needed for transferring residual  $\mathbf{r}^k = \mathbf{b} - \mathbf{A} \cdot \mathbf{v}^k$  from a fine grid  $\Omega^k$  to a coarse grid  $\Omega^{k-1}$ .

In this work, the prolongation and the restriction operators for the displacement fields are obtained by using bilinear interpolation between two generic grids  $\Omega^{k-1}$  and  $\Omega^k$ . In particular, the prolongation operator maps corrections  $\mathbf{e}$  of a solution in a coarse grid onto a fine grid as

$$\mathbf{e}^h = \mathbf{I}_{2h}^h \cdot \mathbf{e}^{2h}, \quad (3.1)$$

where the superscript  $h$  indicates the fine grid and  $2h$  indicates the coarse one.

The prolongation operator for a two-dimensional grid is illustrated in figure 3.3. The correction of a cell on a coarse grid is projected unchanged onto a matching cell on a fine grid as

$$\mathbf{u}^h(3,3) = \mathbf{u}^{2h}(2,2), \quad (3.2)$$

where  $\mathbf{u}$  represents the displacement vector of a cell.

The corrections of a fine grid cell which belongs to an edge of a coarse grid element are approximated as

$$\mathbf{u}^h(3,4) = \frac{1}{2} [\mathbf{u}^{2h}(2,2) + \mathbf{u}^{2h}(2,3)]. \quad (3.3)$$

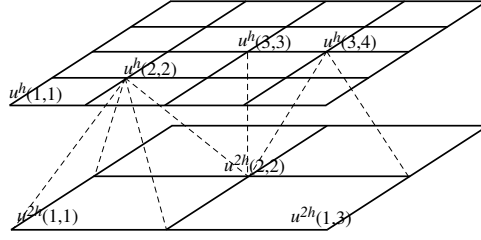


Figure 3.3: Two-dimensional prolongation.

The corrections of a fine grid cell which is located at the middle of a coarse element are approximated using the bilinear interpolation as

$$\mathbf{u}^h(2, 2) = \frac{1}{4} [\mathbf{u}^{2h}(1, 1) + \mathbf{u}^{2h}(2, 2) + \mathbf{u}^{2h}(2, 1) + \mathbf{u}^{2h}(1, 2)]. \quad (3.4)$$

Thus, the prolongation operator can be written in stencil notation as

$$\mathbf{I}_{2h}^h = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}. \quad (3.5)$$

Similarly, the prolongation operator in three-dimensions is obtained and can be written in stencil notation as follows:

$$\mathbf{I}_{2h}^{h(-1)} = \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}, \quad \mathbf{I}_{2h}^{h(0)} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix},$$

$$\mathbf{I}_{2h}^{h(1)} = \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}. \quad (3.6)$$

where the superscript  $(-1)$ ,  $(0)$  or  $(1)$  refers to the position of the matrix in the three-dimensional stencil  $\mathbf{I}_{2h}^h$ .

The restriction operator is needed for transferring the residual from a fine grid to a coarse grid as

$$\mathbf{r}^{2h} = \mathbf{I}_h^{2h} \cdot \mathbf{r}^h. \quad (3.7)$$

By means of Galerkin's approximation, the restriction operator is defined by

$$\mathbf{I}_h^{2h} = \left( \mathbf{I}_{2h}^h \right)^T. \quad (3.8)$$

However, the design is only performed on the finest grid. Then, the density in the fine grid is transformed by restriction to that in the coarse grid. For a two-dimensional problem, the density of each element in a coarse grid is determined by averaging the densities of the matching four elements in the finer grid:

$$\bar{\rho}_{2h}^p = \frac{1}{4} \sum_{i=1}^4 \bar{\rho}_{i_h}^p. \quad (3.9)$$

For a three-dimensional case, the density for each element in a coarse grid is obtained as

$$\bar{\rho}_{2h}^p = \frac{1}{8} \sum_{i=1}^8 \bar{\rho}_{i_h}^p. \quad (3.10)$$

### 3.4 Multigrid design algorithm

The multigrid design algorithm is very similar to the CA design algorithm described in chapter 2. The main difference is that in the former the structural analysis update is implemented according to a multigrid enhanced CA method instead of the base CA method. The multigrid algorithm starts from the finest grid and visits all the hierarchy of coarse grids (see algorithm 3.1). Starting from a fine grid  $\Omega^h$ , the CA analysis update is applied  $S$  pre-relaxations times. The residual  $\mathbf{r}$  obtained from a fine grid  $\Omega^h$  is then mapped onto a coarse grid using the restriction operator  $\mathbf{I}_{2h}^h$ . The density of each element in a coarse grid is also restricted using equation (3.9) (resp. equation (3.10)) for a two-dimensional (resp. three-dimensional) problem. Arriving to the coarsest grid, the CA analysis update is executed until the residual ratio reaches a pre-specified tolerance or an exact solution is obtained if possible. Next, the corrections obtained from a coarse grid  $\Omega^{2h}$  are mapped onto a fine grid  $\Omega^h$  using the prolongation operator  $\mathbf{I}_h^{2h}$  followed by  $S$  post-relaxations. This scheme is repeated until the residual ratio reaches a pre-specified tolerance.

---

**Algorithm 3.1** MG analysis algorithm

---

```

input
   $\mathbf{u}^h, \mathbf{f}^h, \rho^h, h$ 
output
   $\mathbf{u}^h$ 
if  $h =$  coarsest grid then
  repeat
    CA analysis( $\mathbf{u}^h, \mathbf{f}^h, \rho^h$ )
  until  $\frac{\|\mathbf{r}^h\|}{\|\mathbf{r}_0^h\|} \leq \epsilon_r$ 
else
  for  $i=1$  to  $S$  do
    CA analysis( $\mathbf{u}^h, \mathbf{f}^h, \rho^h$ )
  end for
  for each cell do
    % Compute the residual forces for each cell
    %  $\mathbf{r}_C^h$ 
     $\mathbf{f}_C^{2h} = \mathbf{I}_h^{2h} \cdot \mathbf{r}_C^h$ 
     $\mathbf{u}_C^{2h} = 0$ 
  end for
  for each element do
    % Restrict the element densities by using equation (3.9) or (3.10)
    %  $\rho^{2h}$ 
  end for
  for  $i=1$  to  $\gamma$  do
    MG analysis( $\mathbf{u}^{2h}, \mathbf{f}^{2h}, \rho^{2h}, 2h$ )
  end for
  for each cell do
     $\mathbf{u}_C^h = \mathbf{u}_C^h + \mathbf{I}_{2h}^h \cdot \mathbf{u}_C^{2h}$ 
  end for
  for  $i=1$  to  $S$  do
    CA analysis( $\mathbf{u}^h, \mathbf{f}^h, \rho^h$ )
  end for
end if

```

---

### 3.5 Full multigrid design algorithm

The idea behind the full multigrid (FMG) algorithm is to use an initial approximation to the solution of analysis and design for each given grid level in order to

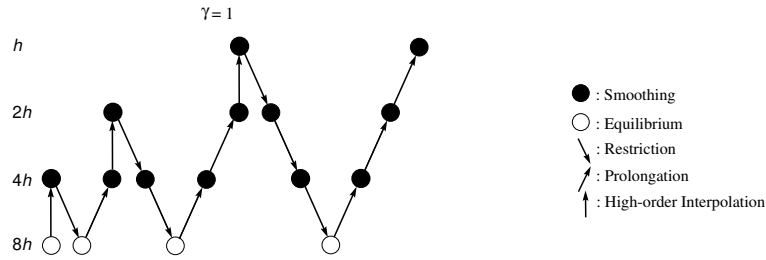


Figure 3.4: Full Multigrid strategy.

accelerate the convergence process. The full multigrid strategy for  $\mathbf{V}$  cycle is shown in figure 3.4. Starting from the coarsest grid ( $i = 1$ ) which requires less computational time to converge, the displacements and the design variables are interpolated recursively to a fine grid using the prolongation operator  $\mathbf{I}_h^{2h}$ . At a given fine grid in the FMG algorithm, the multigrid scheme and the design update rule are applied, whereas in the MG algorithm the design process is carried out only at the finest grid level (see algorithm 3.2)).

---

**Algorithm 3.2** FMG design algorithm
 

---

```

input
   $\mathbf{f}$ 
output
   $\mathbf{u}, \rho$ 
  % CA design algorithm for the coarsest grid
  CA design( $\mathbf{f}^{cg}, \rho^{cg}$ )
for  $i = 2$  to finest grid do
  % Interpolation of the displacements and densities
  for each cell do
     $\mathbf{u}_C^h = \mathbf{I}_{2h}^h \mathbf{u}_C^{2h}$ 
     $\rho_C^h = \mathbf{I}_{2h}^h \rho_C^{2h}$ 
  end for
  MG design( $\mathbf{u}^h, \mathbf{f}^h, \rho^h, h$ )
end for

```

---




















### 3.6 Numerical examples

In this section, some examples of topology optimisation are considered and the efficiency of the multigrid accelerated algorithm is examined. All algorithms described in this chapter are implemented under a Linux C++ environment and tested on a Dual core AMD Opteron(tm) machine with a processor frequency of 2400 Mhz and 8 GByte memory. In all runs, the penalisation parameter is set to 3. The tolerance for the compliance and the design is set to  $10^{-3}$  and 0.05, respectively, and a lower bound of  $10^{-3}$  is adopted for the density.

#### 3.6.1 Comparison

Table 3.1: Optimal topologies and compliances.

Cell number	Optimal topology using NASTRAN	Optimal topology using MG	Optimal topology using FMG
$129 \times 33$	 4273.6	 4258.7	 4189.3
$257 \times 65$	 4064.1	 4062.7	 3859.6
$513 \times 129$	 3985.7	 3984.2	 3727.9
$1025 \times 257$	 3983	 3980.9	 3668.8
$2049 \times 513$	 3994	 3992	 3642
$4097 \times 1025$	lack of memory	 3998.4	 3634.3

To demonstrate the efficiency of the multigrid accelerated CA algorithm in solving the topology optimisation problem, it is compared with existing methods. Since the same CA design update rule is used in all tested algorithms, the comparison concerns design algorithms based on different analysis processes, namely the different

multigrid schemes and the commercial NASTRAN code. The example studied is a symmetric cantilever with 1000 mm in length, 250 mm in height and 1 mm in thickness. The volume fraction is set to 0.5, the tip load considered is  $P = 100$  N, the Poisson ratio is 0.4 and the Young modulus  $E$  used is  $1000$  N/mm<sup>2</sup>. The tolerance for the residual ratio is set to 0.1 and 0.5 for the finest and the coarsest grids, respectively. The  $\mathbf{W}$ -cycle is used for the multigrid process. The smoothing parameter is set to  $S = 2$ . These values were chosen arbitrarily; however, the effect of these different parameters on the multigrid accelerated CA algorithm in solving the topology optimisation problem are discussed in Section 3.6.2.

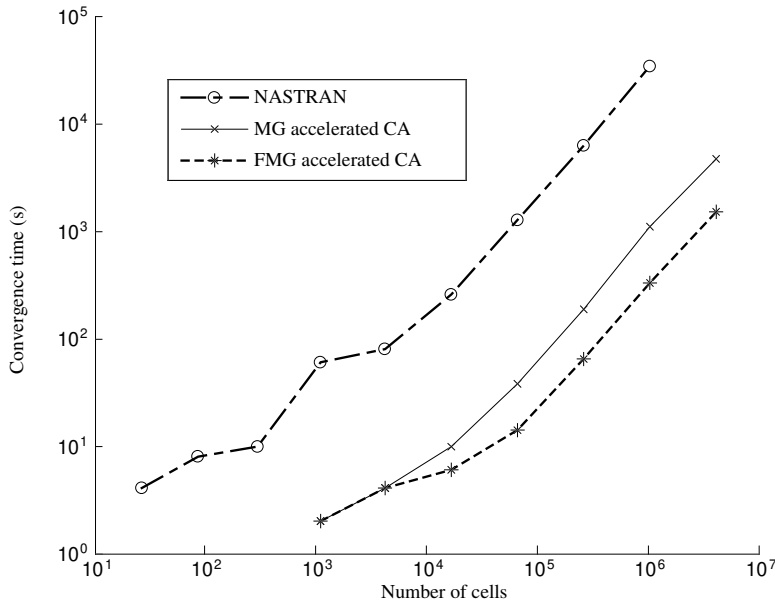


Figure 3.5: Convergence time using NASTRAN, MG accelerated CA and FMG.

Different discretisation levels are used for the comparison; the results are generated for 11 grid levels, starting from the coarsest grid level of  $9 \times 3$  cells, up to the finest grid level of  $4097 \times 1025$  cells. Convergence time for the FEM-CA solution using the commercial NASTRAN code and for both the multigrid accelerated CA algorithm and the full multigrid design algorithm are illustrated in figure 3.5. The vertical and horizontal axes represent the convergence time and the number of cells, respectively, on a log-log scale. First, it is observed that the curves corresponding to the MG accelerated CA algorithm and the FMG design algorithm have approximately the

same slope with some time gain for the FMG algorithm when the number of variables is large enough. The commercial NASTRAN code showed a higher convergence time than the other two algorithms. Moreover, the commercial NASTRAN code suffers lack of memory while running the grid level of  $4097 \times 1025$  cells. On the contrary, the cellular automata paradigm can handle large problems because of its local nature which makes the storage of the global stiffness matrix unnecessary. The run time to convergence relative to the MG and FMG algorithms appears to be nearly proportional to the number of cells, which reveals a computational effort in the order of  $O(N)$ . As for the optimal topologies, from table 3.1 it can be seen that those obtained by the MG algorithm and by the use of NASTRAN for analysis are practically the same with a slightly (0.005% to 0.03%) but persistently lower compliance in the MG results. The FMG algorithm, however, produces a remarkably distinct topology, with a gain between 2% and 9% in compliance, compared to the topologies given by the former algorithms. This performance is mainly attributed to the fact that the FMG algorithm starts successively from a better design as the grid level goes up.

### 3.6.2 Parametric studies

In this section, we consider the influence of different parameters on the convergence of MG and FMG acceleration algorithms. The control parameters which are considered here are the parameter  $\gamma$  to select either the  $\mathbf{V}$  or  $\mathbf{W}$  cycle, the smoothing parameter  $S$ , the number of grid levels used  $N_G$ , and the tolerances for the residual ratio for the finest and the coarsest grids. These parameters are actually the most influential on the performance of the MG acceleration algorithm.

The symmetric cantilever problem of the previous example 3.6.1 was chosen as the example for the parametric studies. The finest grid level is  $1025 \times 257$  cells. The parameter  $\gamma$  to select either  $\mathbf{V}$  or  $\mathbf{W}$  cycle for MG or FMG algorithms is first examined by varying different parameters while some parameters are kept as constants (see section 3.6.1). The  $\mathbf{V}$  and  $\mathbf{W}$  cycles results for MG and FMG algorithms are presented in figure 3.6 where the vertical axis is log scale and represents the total number of cell updates  $N_t$ . From these figures we can remark that the performance of the FMG algorithm with  $\mathbf{V}$  cycle is better than the other three algorithms with respect to the variation of different parameters. In general, the computational efforts of MG with  $\mathbf{V}$  and  $\mathbf{W}$  cycles appear to be almost the same for various combinations of parameters. However, in case where the tolerance of residual ratio for the finest grid  $\epsilon_{r,f}$  is increased (see figure 3.6(b)), the MG algorithm with  $\mathbf{W}$  cycle is more efficient than with  $\mathbf{V}$  cycle. Figure 3.6(c) shows that when the tolerance of the coarsest grid is decreased, the performance of MG with  $\mathbf{V}$  cycle is better than the

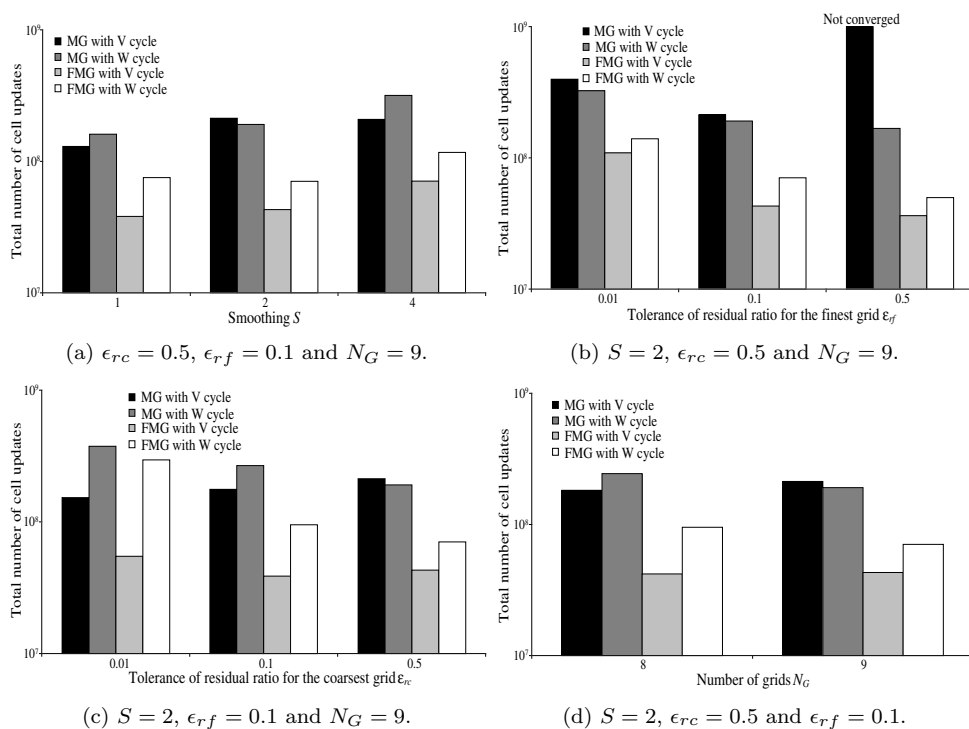


Figure 3.6: Efficiency of MG and FMG algorithms with **V** or **W** cycles: influence of parameters.

#### MG with **W** cycle.

Next, the influence of the rest of the parameters is analysed solely on the MG acceleration algorithm with the **W** cycle, which will be used to solve the example problems of the next sections. The influence of the smoothing parameter  $S$  is shown in figure 3.7 for various combinations of  $\epsilon_{rf}$ ,  $\epsilon_{rc}$  and  $N_G$ . The vertical axis represents the total number of cell updates  $N_t$  on a log scale. As can be seen from this figure, the total number of cell updates for  $S = 1$  and  $S = 2$  are almost the same for the various combinations of these parameters, except for  $\epsilon_{rf} = \epsilon_{rc} = 0.5$ . When  $S = 1$  and  $\epsilon_{rf} = 0.5$ , convergence is obtained with less computational effort. Moreover, the total number of cell updates increases exponentially with the smoothing parameter regardless of the values of the different parameters. When  $S = 1$  (figure 3.8), the plot is slightly unstable, especially for coarsely discretised problems, hence, the choice of  $S = 2$  is a good compromise between efficiency and stability of convergence.

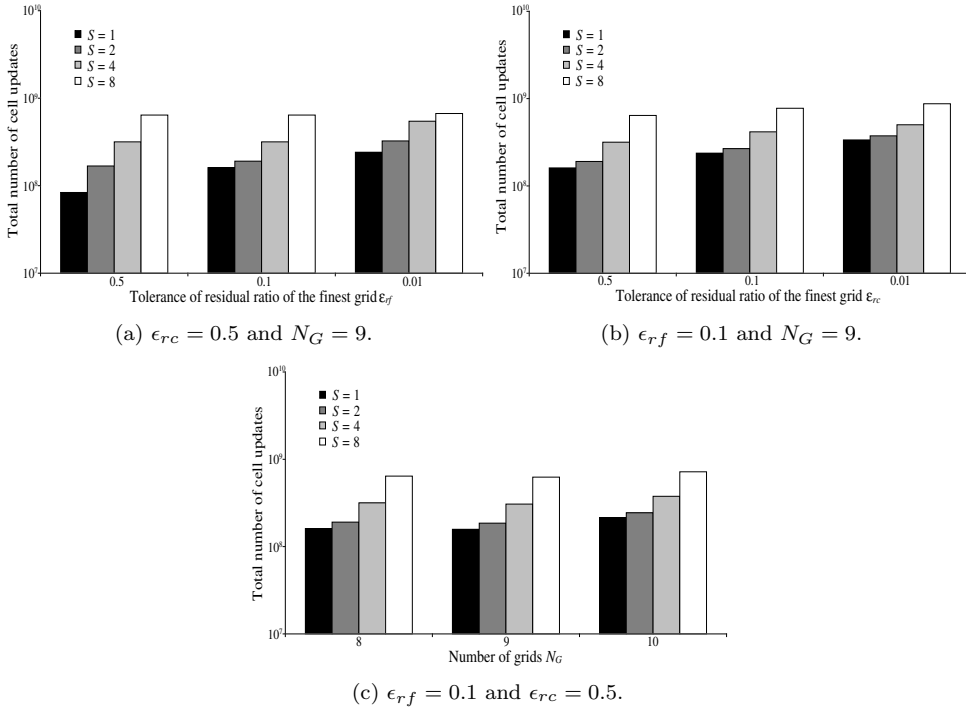


Figure 3.7: The total number of cell updates  $N_t$  for MG with  $S$  variations.

The influence of the tolerance ratio for the finest grid  $\epsilon_{rf}$  on the performance of the MG accelerated CA algorithm with **W** cycle is illustrated in figure 3.9. The results show that beyond the value  $\epsilon_{rf} = 0.3$  the total number of cell updates  $N_t$  becomes constant. Therefore, we can conclude that a value of  $\epsilon_{rf} = 0.2$  is sufficient to obtain convergence. Nevertheless, to be conservative, and to preserve convergence the value of 0.1 is chosen as a tolerance ratio for the finest grid. The influence of the tolerance ratio for the coarsest grid  $\epsilon_{rc}$  on the total number of cell updates  $N_t$  is shown in figure 3.10. As can be seen,  $\epsilon_{rc} = 0.6$  is the minimum value that assures convergence with less computational efforts. The choice of  $\epsilon_{rc} = 0.6$  is confirmed by varying the number of grids  $N_G$  (see figure 3.11). The total number of cell updates decreases as the number of grids increases as depicted in figure 3.11, consequently, as the discretisation of the coarsest grid becomes small enough less computational effort is needed to obtain the convergence.

In all next runs, the multigrid accelerated cellular automata algorithm with **W** cycle

### Chapter 3

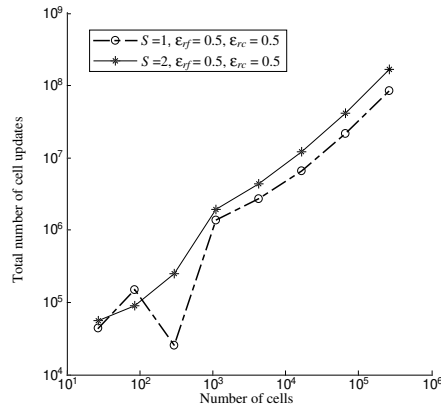


Figure 3.8: The total number of cell updates  $N_t$  for MG when  $S = 1$  and  $S = 2$ .

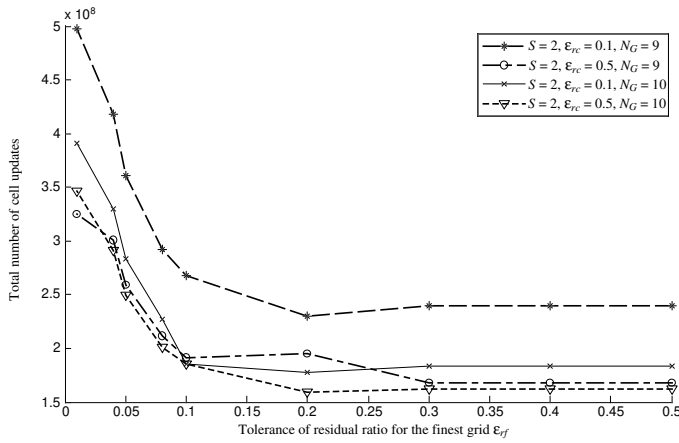


Figure 3.9: Total number of cell updates  $N_t$  as a function of  $\epsilon_{rf}$  for MG with **W** cycle.

is used to solve the topology optimisation problem for two and three dimensional problems. The smoothing parameter is set to  $S = 2$ . The tolerance ratios for the finest grid and the coarsest grid are set to 0.1 and 0.6, respectively. The coarsest grid is defined by the grossest discretisation permitted by the geometry of the design domain. These parameters are fixed to obtain a fast and a stable convergence.

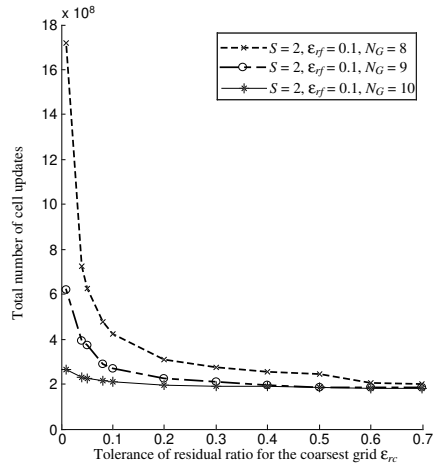


Figure 3.10: Total number of cell updates  $N_t$  as a function of  $\epsilon_{rc}$  for MG with **W** cycle.

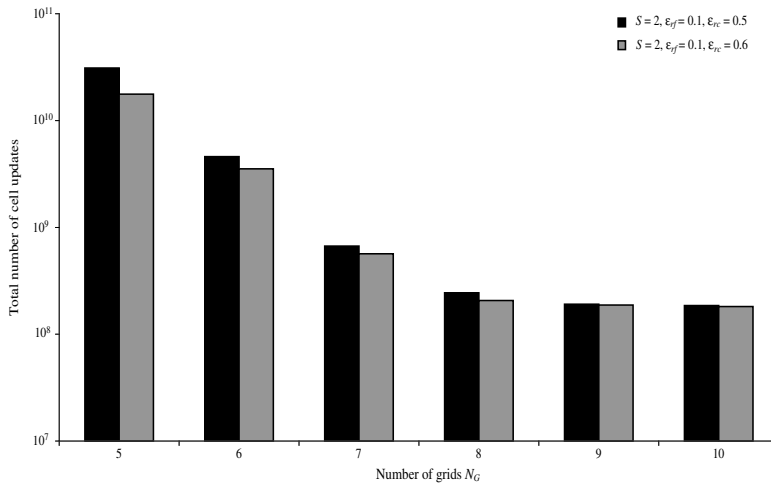


Figure 3.11: The total number of cell updates  $N_t$  for MG with  $N_G$  variations.

### 3.6.3 Compression bridge

In this example, the objective is to find an optimal topology for a bridge which crosses a river and supports a uniformly distributed traffic loading. The design domain,

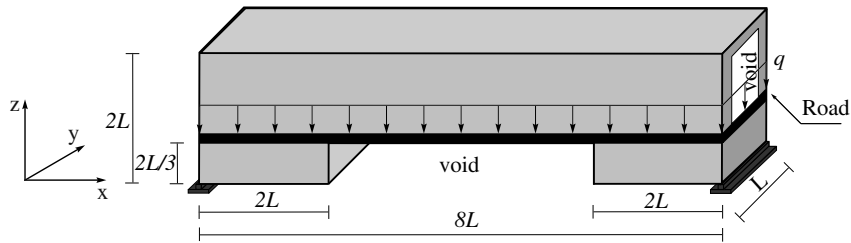


Figure 3.12: Compression bridge domain.

the loading and the boundary conditions of the bridge problem are represented in figure 3.12. Requirements of waterway traffic underneath and road traffic on the bridge translate into the definition of imposed zones: an empty (white) zone for the waterway and a dense (black) one for the deck, as represented in figure 3.12. The design domain is discretised with  $257 \times 65$  cells for the two-dimensional case and with  $257 \times 65 \times 33$  for the three-dimensional case including the empty zone. The volume fraction is set to 0.1 and the Poisson ratio to 0.3.



Figure 3.13: 2D topology of compression bridge.

The final topology for the two-dimensional case performed by the MG design algorithm is represented in figure 3.13. It corresponds to a compression arch which holds a three span deck. The first and the third spans are cantilevers which are supported each by a compression member, whereas the central span is suspended via a series of tension members. Different views for the three-dimensional version of the topology of the bridge are shown in figure 3.14. The topology obtained with



the three-dimensional model presents some similarity, in the XZ plane, with the topology generated by the two-dimensional model (see figures 3.14(a) and 3.13) and with the design of the compression arch bridge reported in [60]. The algorithm for two-dimensional case converges in a total of 57 design updates in 30 seconds, and for the three-dimensional case converges in a total of 56 design updates and the run time is 7973 seconds.

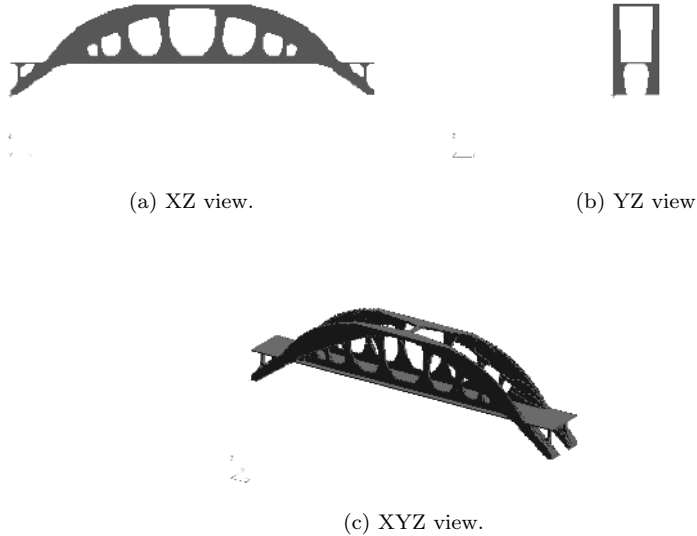


Figure 3.14: 3D topology of compression bridge.

### 3.6.4 Arch bridge

The problem definition is similar to the previous example, except that the deck is located at the top of the cuboid structure. The bridge structure is simply supported at the lower outer left and right edges, with a uniformly distributed traffic loading applied at the top surface as shown in figure 3.15. The void volume should allow passage of boats under the bridge.

The MG design algorithm is run with  $257 \times 65$  cells for the two-dimensional case and

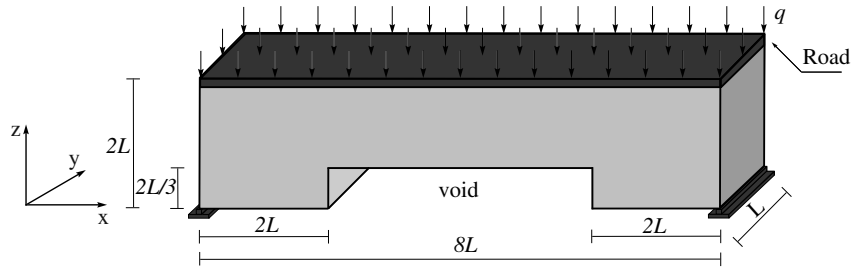


Figure 3.15: Arch bridge domain.

$257 \times 65 \times 33$  cells for the three-dimensional case. The two-dimensional converged design for a Poisson ratio  $\nu = 0.3$  and a volume fraction  $\eta = 0.15$  is shown in figure 3.16. The converged topology corresponds to a compression arch that supports the deck by means of compression members. The two-dimensional final topology is shown in figure 3.16. Its layout is similar to that of the XZ view of the final three-dimensional topology presented in figure 3.17. The converged topology in figure 3.16 requires 57 design iterations in 21 seconds, and the topology in figure 3.17 requires 59 design updates for a run time of 2673 seconds.



Figure 3.16: 2D topology of arch bridge.

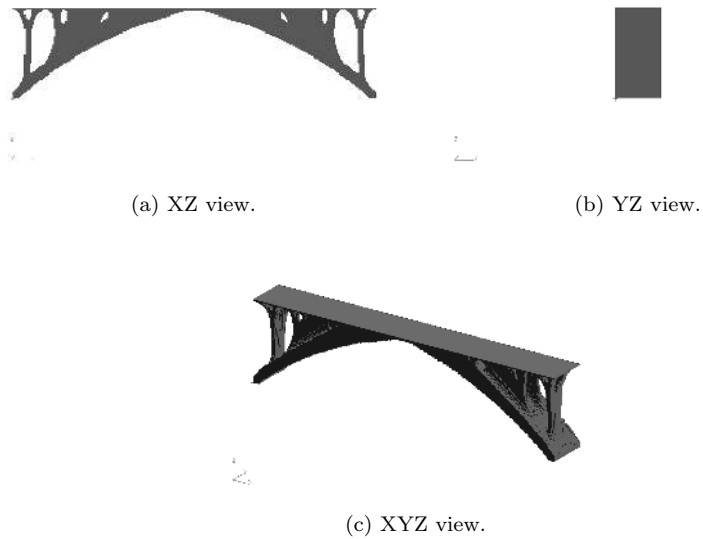


Figure 3.17: 3D topology of arch bridge.

### 3.6.5 Multiple arch bridge

In the following example, we try to find a suitable design for a bridge that crosses a river. This bridge is designed to support road traffic and no waterway traffic is

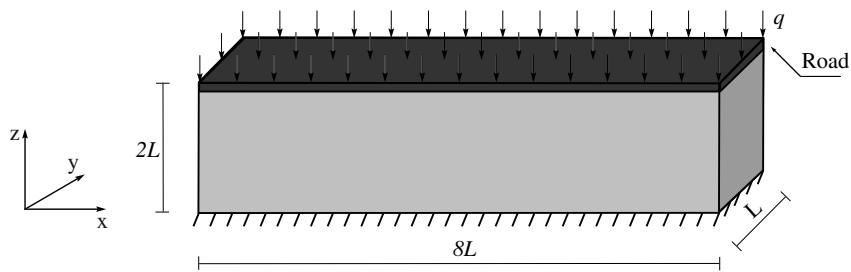


Figure 3.18: Multiple arch bridge domain.



Figure 3.19: 2D topology of multiple arch bridge.

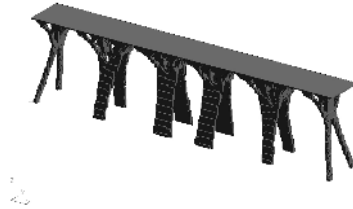
allowed on this river. The bridge domain is shown in figure 3.18. The structure should support a uniformly distributed traffic loading at the top surface and is clamped at the bottom surface. The deck is represented as a dense zone (see figure 3.18).



(a) XZ view.



(b) YZ view.



(c) XYZ view.

Figure 3.20: 3D topology of multiple arch bridge.

The design domain is discretised with  $257 \times 65$  and  $257 \times 65 \times 33$  cells for the two and three dimensional cases respectively. The Poisson ratio used is  $\nu = 0.3$  and the volume fraction is set to 0.1. The topology for two-dimensional case is shown in figure 3.19. The topology corresponds to a four arch bridge that holds a deck.

The three-dimensional version of this bridge with different views is depicted in figure 3.20. It is almost similar to the two-dimensional topology but with five arches. The multigrid accelerated cellular automata algorithm for the two-dimensional problem converges in a total of 94 design updates in 51 seconds, and it converges in a total of 72 design updates in 8352 seconds for the three-dimensional problem.

### 3.7 Concluding remarks

Combined multigrid cellular automata implementations for the topology optimisation of continuum structures have been presented in this work. Two design algorithms have been proposed, a multigrid and a full multigrid design algorithms. Numerical tests for a symmetric cantilever example illustrate the efficiency of the combined multigrid cellular automata design algorithm in solving the topology optimisation problem compared to the use of the commercial NASTRAN code for the analysis phase. The multigrid scheme is shown to accelerate the convergence of the analysis phase and the full multigrid to improve convergence of both analysis and design in the topology optimisation problem. The computational cost for the multigrid algorithms is indeed found to be proportional to the number of cells, that is an effort of order  $O(N)$ . Applied to 2D and 3D topology optimisation of example bridge structures, the multigrid accelerated cellular automata design algorithm generates realistic topologies such as the familiar compression arch bridge architectures.

## Chapter 3

## Chapter 4

# Wind Load Effect in Topology Optimisation Problems

### 4.1 Introduction

In the last two decades, increasing interest has been observed in solving the topology optimisation of continuum structures [2]. However, optimisation of structural topology under design-dependent loads is a comparatively recent topic. Few researchers take into account the effects of design-dependent loads on the topology optimisation problem. Most of the available work consists of design for structures under hydrostatic pressure [37, 39–43] and design for structures under transmissible loads [38]. Much of this work is not applicable to the wind loading problem, given that no wind loading is applied when there is no surface obstructing the wind.

In this chapter, we propose a scheme that does not involve explicit construction of the loading surfaces. The topology optimisation of two and three dimensional structures subject to dead and wind loading is considered. The wind loading is introduced into the formulation using standard expressions for the drag force, and a strategy is devised so that wind pressure is ignored where there is no surface obstructing the wind. A design problem formulation is constructed for a minimum compliance design subject to a volume constraint, and using the popular SIMP material model [7–11].

The Method of Moving Asymptotes (MMA) [46–49] is a well known solver that is commonly used to solve the topology optimisation problem [61–63], in this chapter it is used as a benchmark solver for the next chapter. The MMA is modified by including line search and changing formula for the update of asymptotes. In order to obtain black/white design, intermediate density values, which are used as design variables, are controlled by imposing an explicit constraint. Numerical examples are used to demonstrate that the proposed formulation is successful in incorporating the effect of wind loading into the topology optimisation problem.

## 4.2 Topology design with wind loading

In this chapter, the topology optimisation problem is posed following the commonly used minimum compliance formulation, with a material volume constraint as described in chapter 2 (see equations (2.1)-(2.3)); the formulation can be written as

$$\min_{\rho} W(\mathbf{u}^*) = \mathbf{F} \cdot \mathbf{u}^*, \quad (4.1a)$$

subject to,

$$V = \int_{\Omega} \rho \, d\Omega \leq V^*, \quad (4.1b)$$

$$0 \leq \underline{\rho} \leq \rho \leq 1, \quad (4.1c)$$

where  $W$  is the compliance of the structure which can be written either as the work done by the external forces or as twice the total elastic energy at equilibrium. The vector  $\mathbf{F}$  denotes the external forces, and  $\mathbf{u}^*$  is the displacement field of the domain  $\Omega$  at equilibrium. The local density distribution of material  $\rho$  of the discretised model is chosen as a design variable, and  $\underline{\rho}$  is a small number set as a lower bound on  $\rho$  to avoid numerical instability. The bound  $V^*$  is the allowable limit on the material volume  $V$  of the structure.

The well known SIMP approach [7–11] was selected to achieve a black/white topology design. Local material stiffness  $E$  is defined as

$$E = \bar{\rho}^p E_0, \quad (4.2)$$

where  $E_0$  is the base elasticity modulus, and  $p$  is a penalisation parameter  $p \geq 3$ . In the present formulation the local density measures  $\rho_i$  of the discretised structure are associated with the nodes of the finite element mesh. The element densities  $\bar{\rho}$  are



obtained by an average compliance interpolation [20] for the  $m$  nodes surrounding the element and given by

$$\frac{1}{\bar{\rho}^p} = \frac{1}{m} \sum_{i=1}^m \frac{1}{\rho_i^p}. \quad (4.3)$$

Using this scheme checkerboard patterns are suppressed automatically during the optimisation process.

This section begins with an explanation of how the wind loading is introduced into the topology optimisation problem. A sensitivity analysis is then carried out. Finally, the optimisation problem is solved using the modified Method of Moving Asymptotes (MMA) [46] coupled with Line Search (MMALS).

### 4.2.1 Wind loading function

Wind typically blows in many directions at varying speeds. Ideally multiple load cases need to be considered in the topology optimisation process to account for the consequent variabilities in wind forces acting on a structure. However, in this study only one load case is considered for simplicity; with uni-directional incoming wind blowing from the West (see Figure 4.1(a)). This assumption remains valid for any incoming wind directions.

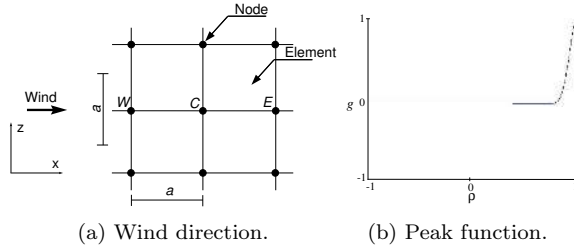


Figure 4.1: Wind direction and peak function.

Wind loading is included in the formulation using a direct density based method that does not involve explicit construction of the loading surfaces. The wind load is defined at each node of the discretised domain by introducing a loading function  $g$  using the standard formula for drag forces:

$$F_{Wind_C} = C_D Q_\infty A g, \quad (4.4)$$

where  $F_{Wind_C}$  is the wind load value at the node  $C$ ,  $Q_\infty$  is the incoming wind dynamic pressure,  $C_D$  is the drag coefficient, and  $A$  is the area of the surface side facing the wind in YZ directions. Computation of the area  $A$  depends on the node position of  $C$ , i.e. whether the node is at a corner, a side, or the middle of the discretised domain. For example for a middle node, in the 2D case,  $A$  is equal to  $a$  times the thickness of the structure and in the 3D case it is equal to  $a^2$  where  $a$  is the distance between the nodes of a uniform mesh. The factor  $g$  is a peak function interpolation [64] given by

$$g = e^{-\frac{[1-(\rho_C-\rho_N)]^2}{2\sigma^2}}, \quad (4.5)$$

where  $\rho_C$  is the density of the node  $C$  and  $\rho_N$  is the density of the neighbouring node in the direction opposite to wind direction ( $\rho_N = \rho_W$  when the incoming wind flows from the West), and  $\sigma$  is a standard deviation parameter. The force  $F_{Wind_C}$  is applied at every node of the domain. However, the use of a peak function interpolation for  $g$ , see Figure 4.1(b), ensures that the wind loading is applied only at nodes where a large change in density occurs, indicating a structural surface, and that the loading is unidirectional. In other words, the load is applied only when the node  $C$  is in a solid region and the node  $N$  is in void. In the present formulation, we are interested in truss like structures. Thus, we make the assumption that no wind shadowing is present.

### 4.2.2 Sensitivity analysis

The sensitivity of the compliance with respect to the design variables, which are the node densities in this case, is obtained from:

$$\frac{dW}{d\rho} = \frac{d\mathbf{F}}{d\rho} \cdot \mathbf{u} + \mathbf{F} \cdot \frac{d\mathbf{u}}{d\rho}. \quad (4.6)$$

The sensitivity of the displacement in the above equation can be obtained by differentiating the equilibrium equation with respect to the node densities and is expressed as

$$\frac{d\mathbf{u}}{d\rho} = \mathbf{K}^{-1} \cdot \left( \frac{d\mathbf{F}_{Wind}}{d\rho} - \frac{d\mathbf{K}}{d\rho} \cdot \mathbf{u} \right), \quad (4.7)$$

where  $\mathbf{K}$  is the global stiffness matrix and  $\mathbf{F}_{Wind}$  is the wind load vector. Thus, the sensitivity of the compliance is given by

$$\frac{dW}{d\rho} = 2 \mathbf{u} \cdot \frac{d\mathbf{F}_{Wind}}{d\rho} - \mathbf{u} \cdot \frac{d\mathbf{K}}{d\rho} \cdot \mathbf{u}. \quad (4.8)$$

As can be observed from equation (4.8), the sensitivity of the compliance is decomposed into two terms; the first term is due to the dependence of external forces on

the design variables, and the second one is due to the dependence of the stiffness on the same variables. We denote these two terms by  $\psi$  and  $\phi$ , respectively. Furthermore, external forces applied to a given structure can be decomposed into two parts, which are the static loads and the incoming wind forces. Dead loads are assumed to be independent of the structural configuration. Thus, the expression of  $\psi$  for each node  $C$  depends only on wind loading and simplifies to

$$\psi_C = \frac{C_D Q_\infty A}{\sigma^2} \left\{ \left[ [1 - (\rho_C - \rho_W)] e^{-\frac{[1 - (\rho_C - \rho_W)]^2}{2 \sigma^2}} \right] u_C - \left[ [1 - (\rho_E - \rho_C)] e^{-\frac{[1 - (\rho_E - \rho_C)]^2}{2 \sigma^2}} \right] u_E \right\}, \quad (4.9)$$

where the subscripts  $W$  and  $E$  refer to the nodal points to the West and to the East of node  $C$ , respectively, and the wind direction is assumed to be from West to East. Using the same notation,  $u_C$  and  $u_E$  are the horizontal displacement at node  $C$  and its East neighbour, respectively.

### 4.2.3 Optimisation

The optimisation problem is solved by the Method of Moving Asymptotes (MMA) [46]. According to [46], the compliance is approximated as

$$W^{(k)}(\rho) = r_0^{(k)} + \sum_{j=1}^n \left( \frac{p_{0j}^{(k)}}{U_j^{(k)} - \rho_j} + \frac{q_{0j}^{(k)}}{\rho_j - L_j^{(k)}} \right), \quad (4.10)$$

where  $k$  is the MMA iteration index,  $n$  is the total number of nodes of the discretised domain,  $r_0^{(k)}$  is a correction term to ensure that the approximating objective function equals the original function at the current iteration point,  $p_{0j}^{(k)}$  and  $q_{0j}^{(k)}$  are determined by a first-order approximation of the first Kuhn-Tucker condition as

$$p_{0j}^{(k)} = \begin{cases} (U_j^{(k)} - \rho_j^{(k)})^2 \partial W_j / \partial \rho_j, & \partial W_j / \partial \rho_j > 0 \\ 0, & \partial W_j / \partial \rho_j \leq 0 \end{cases} \quad (4.11)$$

$$q_{0j}^{(k)} = \begin{cases} 0, & \partial W_j / \partial \rho_j \geq 0 \\ -(\rho_j^{(k)} - L_j^{(k)})^2 \partial W_j / \partial \rho_j, & \partial W_j / \partial \rho_j < 0, \end{cases} \quad (4.12)$$

and  $L_j^{(k)}$  and  $U_j^{(k)}$  are the lower and upper asymptotes respectively, defined by a heuristic strategy proposed by Svanberg [46] as

$$L_j^{(k)} = \rho_j^{(k)} - s \left( \rho_j^{(k-1)} - L_j^{(k-1)} \right) \quad (4.13)$$

$$U_j^{(k)} = \rho_j^{(k)} + s \left( U_j^{(k-1)} - \rho_j^{(k-1)} \right), \quad (4.14)$$

where the coefficient value  $s$  is equal to 0.5 for the iterations  $k = 1$  and  $k = 2$ . For  $k \geq 2$ , the coefficient value  $s$  is chosen as 0.7 when the sign of  $(\rho_j^{(k)} - \rho_j^{(k-1)})$  and  $(\rho_j^{(k-1)} - \rho_j^{(k-2)})$  are opposed. However, when  $(\rho_j^{(k)} - \rho_j^{(k-1)})$  and  $(\rho_j^{(k-1)} - \rho_j^{(k-2)})$  have the same sign,  $s$  is chosen equal to 1.2.

It is observed that, in some cases, the optimisation process tends to oscillate when the MMA solver tries to update the asymptotes  $L_j^{(k)}$  and  $U_j^{(k)}$ . The convergence is stabilised by modifying the lower asymptote according to the continuous optimality criteria interpreted as local Kuhn-Tucker conditions [20]. The update of the upper asymptote is kept the same as suggested by Svanberg [46]. Using the continuous optimality criteria interpreted as local Kuhn-Tucker conditions the objective function is approximated as

$$W^{(k)} = \sum_{j=1}^n \frac{\phi_j^{(k)}}{\left(\rho_j^{(k)}\right)^p}. \quad (4.15)$$

This approximation can be convexified by linearising the stiffness coefficient  $\rho_j^p$  around the most recent design point to obtain:

$$W^{(k)} = \sum_{j=1}^n \frac{q_{0j}^{(k)}}{\rho_j^{(k)} - L_j}, \quad (4.16)$$

where  $L_j = (1 - \frac{1}{p})\rho_j^{(k-1)}$  is the modified formula for the lower asymptote.

With the modifications discussed above the MMA solver does not always converge monotonously and in some cases it diverges, especially for strong wind loadings. To improve its performance, the calculation of the coefficients  $p_{0j}^{(k)}$  and  $q_{0j}^{(k)}$  defined in [46] is modified by considering the sensitivity contributions from the  $\phi$  and  $\psi$  terms independently:

$$p_{0j}^{(k)} = \begin{cases} (U_j^{(k)} - \rho_j^{(k)})^2 \psi_j, & \psi_j > 0 \\ 0, & \psi_j \leq 0 \end{cases} \quad (4.17)$$

$$q_{0j}^{(k)} = (\rho_j^{(k)} - L_j^{(k)})^2 \phi_j + \begin{cases} 0, & \psi_j \geq 0 \\ -(\rho_j^{(k)} - L_j^{(k)})^2 \psi_j & \psi_j < 0. \end{cases} \quad (4.18)$$

Moreover, it is also observed that the objective function does not always decrease after an MMA step and that the constraints are violated in some cases. In an earlier paper, Zillober [47] demonstrated that, by adding a line search in the optimisation

process the behaviour of the MMA method can be stabilised. Therefore, a line search is used in the present work to ensure satisfaction of the constraints and decrease in the objective value. Let the solution produced by MMA after the  $k^{th}$  iteration be denoted by  $\rho^*$ . A solution is sought in the form:

$$\rho^{(k+1)} = (\rho^* - \rho^{(k)})\alpha + \rho^{(k)}, \quad (4.19)$$

where  $\alpha \in [0, 1]$  is a parameter that needs to be determined such that the constraints be satisfied and the objective function be decreased. The value of  $\alpha$  is found by solving a one-dimensional approximate optimisation problem constructed from the original problem by using approximate cubic polynomial approximations for the objective and the constraints as functions of the independent variable  $\alpha$ .

### 4.3 Explicit constraint

Preliminary results for a numerical example illustrating an off-shore wind turbine support structure exhibit a grey design as the optimal topology (see Figure 4.2(a)) as opposed to a black/white one as expected. The explanation for this is related to the distribution of the wind loads over the nodes as shown in Figure 4.2(b). Each column in the figure is a histogram of the magnitude of the loads on nodes along a specific vertical line. Note that the wind loads are distributed throughout the domain, but have significant magnitude only for a few nodes at the bottom of the structure; but, the distribution of the wind loads must be uniform on a surface facing the wind, which is not the case. To generate a black/white designs, an explicit constraint [65–68] was added to the formulation of problem (4.1) as a means to control the intermediate densities. The explicit constraint is defined as

$$h(\rho) = \int_{\Omega} (1 - \rho)(\rho - \underline{\rho})d\Omega \leq \epsilon_p. \quad (4.20)$$

In its numerical implementation the explicit constraint (4.20) is enforced gradually as follows. Initially, it is ignored, which results in a grey design. The value of the constraint function for this design is denoted by  $\epsilon_p^{(0)}$ . Next, a sequence of problems  $P_j$  are solved, each subject to the additional constraint (4.20) where the right hand side  $\epsilon_p$  is the  $j^{th}$  term of an associated decreasing sequence initialised at  $\epsilon_p^{(0)}$  and updated according to the formula:

$$\epsilon_p^{(j)} = \epsilon_p^{(j-1)}(1 - \beta). \quad (4.21)$$

**Algorithm 4.1** MMALS algorithm

---

```

input
   $\mathbf{u}, \mathbf{f}, \rho$ 
output
   $\mathbf{u}, \rho$ 
  % Initialisation
   $k = 0, j = 0$ 
  % Update wind forces
  %  $\mathbf{f}_{Wind}^{(1)}$ 
  FEM Analysis( $\mathbf{u}^{(0)}, \mathbf{f}^{(1)}, \rho^{(1)}$ )
  %  $\mathbf{u}^{(1)}$ 
   $h(\rho) = \sum_{c=1}^{N_{cells}} (1 - \rho_c)(\rho_c - \underline{\rho}) \bar{V}_c$ 
   $\epsilon_p^{(0)} = h(\rho)$ 
   $W^{(1)} = \mathbf{u}^{(1)} \cdot \mathbf{f}^{(1)}$ 
repeat
  repeat
    if  $W^{(k+1)} \leq W^{(k)}$  and constraints are satisfied then
      Modified MMA solver( $\rho^{(k)}$ )
      %  $\rho^{(k+1)}$ 
    else
      Line search( $\rho^{(k)}$ )
      %  $\rho^{(k+1)}$ 
    end if
    for each element do
      % Element density interpolation
      %  $\rho_e^{(k+1)}$ 
    end for
    % Update wind forces
    %  $\mathbf{f}_{Wind}^{(k+1)}$ 
    FEM Analysis( $\mathbf{u}^{(k)}, \mathbf{f}^{(k+1)}, \rho^{(k+1)}$ )
    %  $\mathbf{u}^{(k+1)}$ 
     $W^{(k+1)} = \mathbf{u}^{(k+1)} \cdot \mathbf{f}^{(k+1)}$ 
     $k = k + 1$ 
  until (five successive  $|\frac{W^{(k+1)}}{W^{(k)}} - 1| \leq \epsilon_c$ ) and  $(|\rho^{(k+1)} - \rho^{(k)}| \leq \epsilon)$ 
   $h(\rho) = \sum_{c=1}^{N_{cells}} (1 - \rho_c)(\rho_c - \underline{\rho}) \bar{V}_c$ 
  if  $\frac{h(\rho)}{V_T} \leq \epsilon_p^j$  then
     $\epsilon^{(j+1)} = \epsilon^{(j)}(1 - \beta)$ 
  else
    Break
  end if
   $j = j + 1$ 
until  $j = 101$ 

```

---

where the reduction  $\beta$  is defined by the user.

The process continues until the solver fails to converge for the updated value of  $\epsilon_p$  (see algorithm 4.1).

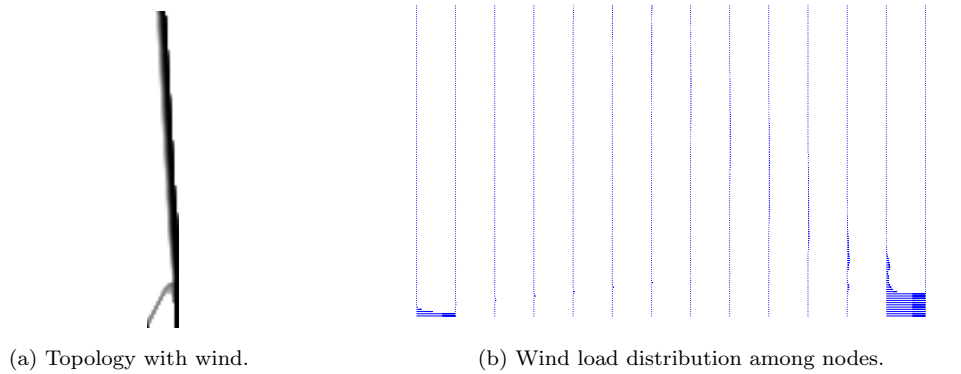


Figure 4.2: Off-shore wind turbine problem without explicit constraint.

## 4.4 Numerical examples

To evaluate the influence of wind loading on the optimum topology, the numerical example of an off-shore wind turbine support structure is considered. The design domain is a parallelepiped volume as shown in figure 4.3 which reduces to a rectangular domain in the 2D case. The hub and rotor weight  $P_{WR}$ , and the rotor thrust  $F_W$  are considered as dead loads with values of 2.55 MN and 0.125 MN, respectively. The spatial design domain considered is box shaped with dimensions of 120 m in height, 12 m in length and 12 m in width. The structure is clamped at the bottom. The incoming wind is assumed to flow from the West with a wind speed of 25 m/s. The problem is studied in both two and three dimensional cases with the assumption of linear small deformations. The material used is steel with Young's modulus  $E = 200000$  MPa, and Poisson's ratio  $\nu = 0.3$ . The penalisation parameter is set to 3, the volume fraction is set to be 0.3, and a lower bound  $\underline{\rho} = 10^{-3}$  is adopted for the density. The value of  $2\sigma^2$  for the peak function is fixed at 0.1 and the value of  $\beta$  to 0.05.

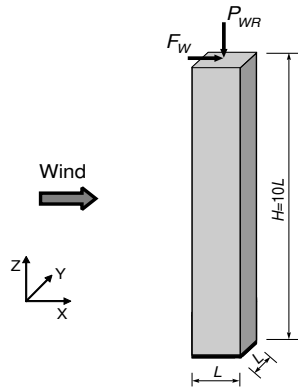


Figure 4.3: Domain.

#### 4.4.1 Two-dimensional example



(a) Optimum topology without wind.

(b) Optimum topology with wind.

Figure 4.4: 2D topology of an Off-shore wind turbine.

A 2D problem is first considered to investigate the topology optimisation of the wind turbine support structure. The results of this problem were generated with a discretised grid of  $15 \times 141$  nodes in the  $x$  and  $z$  directions, respectively. The final designs are represented in figures 4.4(a) and 4.4(b) for the cases with and without wind loading, respectively, while maintaining the hub and rotor weight  $P_{WR}$ , and the rotor thrust  $F_W$  for the both cases. It can be observed that the topology without wind is a column that supports only the dead loads. On the other hand, the topology



subject to wind loads requires more elements. The compliance calculated for the design in figure 4.4(a) is  $2.29 \cdot 10^{-2}$  MNm, while the compliance for the design in figure 4.4(b), where dead and wind loadings are applied, is  $1.39 \cdot 10^{-2}$  MNm. Thus, including wind loading in the design formulation leads to 40% increase in structural stiffness for the same material volume.

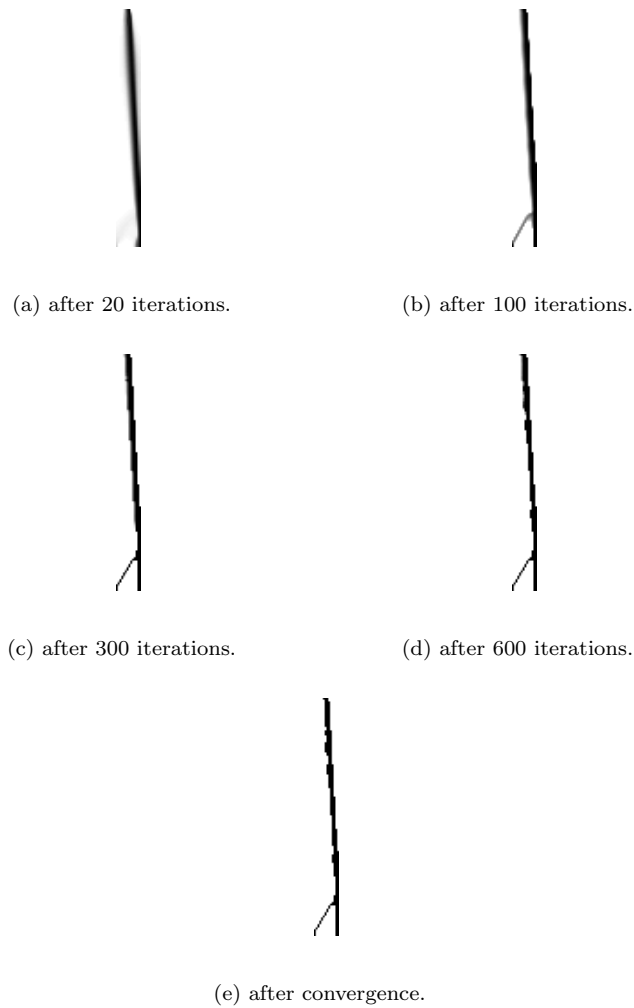


Figure 4.5: Evolution of the wind turbine support topology.

Figure 4.5 shows the evolution of the wind turbine support topology. The algorithm converges to a black/white topology in about 884 analysis iterations. It can be observed that the topology starts from a grey column design and ends up with a black/white design with apparition of an element that supports the loads coming from the wind.

Figure 4.6(a) shows the distribution of wind loads for the design obtained considering wind loading. It can be observed that the wind loads are uniformly distributed throughout the design domain. In some cases, especially for strong wind (Zakhama et al. [44]), the wind loads may be not uniformly distributed in some parts of the domain. This is due mainly to the way the coefficients  $p_{0j}^{(k)}$  and  $q_{0j}^{(k)}$  of the compliance approximation are updated and the parameter  $\epsilon_p$  of the explicit constraint is reduced. Further investigation is needed to determine an appropriate update scheme for these coefficients. Figure 4.6(b) shows the convergence history. The convergence is reasonably smooth with jumps in compliance corresponding to tightened tolerance on grey density. Although the introduction of the explicit constraint leads to an increased number of iterations to convergence, the obtained solution is physically more meaningful.

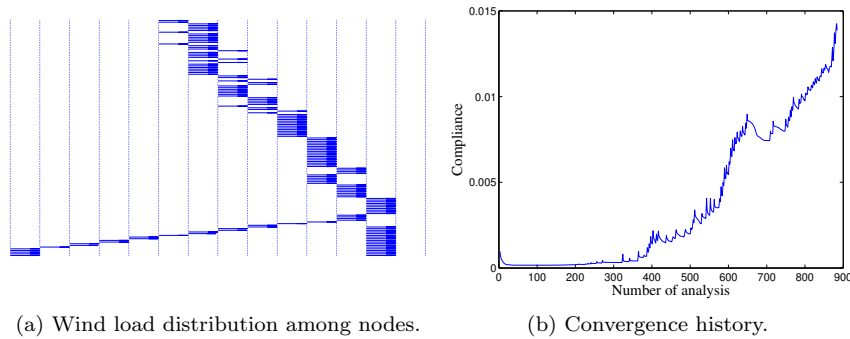


Figure 4.6: Wind load distribution and convergence history.

#### 4.4.2 Three-dimensional example

The off-shore wind turbine domain (see figure 4.3) is now discretised with  $15 \times 15 \times 141$  nodes. Figure 4.7 shows the different views of the optimal topology of wind turbine support with and without wind loads. To compare the compliances of the two solutions, the designs obtained without and with wind loads are postprocessed to force exact black/white topologies. The compliance without considering wind loads

## Wind Load Effect in Topology Optimisation Problems

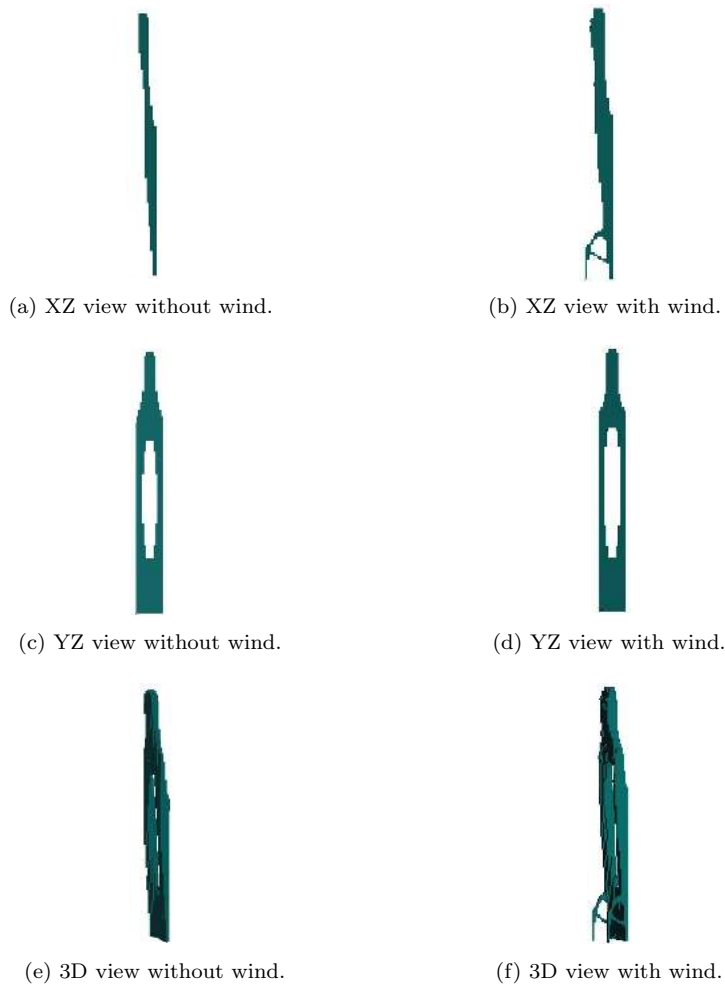


Figure 4.7: 3D topology of the wind turbine support.

is  $2.39 \cdot 10^{-2}$  MNm and the compliance considering wind loads is  $1.51 \cdot 10^{-2}$  MNm, which corresponds to 37% difference between the two solutions.

The topologies obtained for the 3D case are seen to be very similar to the topologies obtained for the 2D case in XZ plane (see figures 4.7(a), 4.7(b) and 4.4). The topology of the wind turbine support considering wind loading is supported by two

frames at the bottom of the structure. Figure 4.7(d) shows that, by including wind loading in the formulation, the topology allowed a larger hole facing the wind loads than the topology without wind. In the literature, most techniques for topology optimisation with design-dependent loads do not allow the formation of holes on a surface facing the wind.

## 4.5 Concluding remarks

A formulation for the inclusion of wind loading in the minimum compliance topology optimisation problem has been proposed. The method does not require the explicit construction of loading surfaces. The MMA method has been modified and a line search has been added to the process to guarantee the global convergence of the topology optimisation problem. An explicit constraint has been added into the topology optimisation formulation to control the intermediate density values and to ensure black/white topology. Numerical examples demonstrate the effect of the wind loads on the optimised topologies. Taking the wind loads into account in the formulation gives stiffer designs, with respect to these loads, with the same material volume and can allow holes facing the wind loads. Thus, the implementation of this method in the preliminary design phase can lead to significant reduction in structural weight which is reflected on the total cost of off-shore wind farms especially given the rising steel prices.

## Chapter 5

# Multigrid Accelerated Cellular Automata with Design-Dependent Loads

### 5.1 Introduction

The inclusion of wind loading in the minimum compliance topology optimisation was presented in chapter 4. The modified method of moving asymptotes [46] and its coupling with a line search method were proposed to solve the topology optimisation problem. The same framework, which includes wind loading in the minimum compliance topology optimisation problem, can be extended and solved using the multigrid accelerated cellular automata algorithm described in chapter 3.

In the present chapter, the topology optimisation of two and three dimensional structures subject to design-dependent loading is solved using the multigrid accelerated cellular automata algorithm. The optimality criteria for topology optimisation subject to design-dependent loading are developed using again the SIMP material model [7–11] to derive the local CA design rules. In the previous chapter there was no need to develop optimality criteria since topology optimisation was solved using the modified MMA method coupled with Line Search (MMALS) algorithm. On the other hand, the optimality criteria have been derived in chapter 2 but without considering the design-dependent loading in the formulation. The local optimisation

problem is convexified to prevent oscillation in the topology optimisation process. The Jacobi over relaxation scheme (JOR) [69, 70] is used to update the design variables and to help establish convergence. The intermediate densities are controlled by an explicit constraint [65–68], to obtain a black/white design. Some numerical examples are shown to highlight the efficiency of the multigrid accelerated cellular automata in solving the topology optimisation problem subject to design-dependent loading. Besides, the algorithm based on the multigrid accelerated cellular automata is seen to be more robust than the MMALS algorithm, and can lead a design to an almost pure black/white solution.

## 5.2 Minimum compliance design optimality criteria

The minimum compliance design optimisation problem used in this chapter is described earlier in chapter 4 (see problem (4.1) and is included here for completeness:

$$\min_{\rho} W(\mathbf{u}^*) = \mathbf{F} \cdot \mathbf{u}^*, \quad (5.1a)$$

subject to,

$$V = \int_{\Omega} \rho \, d\Omega \leq V^*, \quad (5.1b)$$

$$0 \leq \rho \leq 1. \quad (5.1c)$$

However, in chapter 4 there was no need for the optimality conditions since the problem was solved by using MMALS method. Although the optimality conditions have been derived in chapter 2 for a similar problem formulation, the previous formulation lacked the design-dependent loads which are included in the present chapter. The Lagrangian for this optimisation problem can be written as

$$\mathcal{L} = \mathbf{F} \cdot \mathbf{u}^* + \mu(V - V^* + c^2) + \lambda_1 \cdot (\rho - 1 + \mathbf{s}_1^2) + \lambda_2 \cdot (\rho - \rho + \mathbf{s}_2^2), \quad (5.2)$$

where  $\mu$  is a Lagrange multiplier associated with the volume constraint and  $c$  is the corresponding slack variable. The Lagrange multipliers  $\lambda_1$  and  $\lambda_2$  are associated with the bounds on the design variable vector, and  $\mathbf{s}_1$  and  $\mathbf{s}_2$  are the corresponding vectors of slack variables, respectively.

Setting the variation of the Lagrangian to zero, we obtain the first order necessary conditions as

1. Stationarity

$$\frac{\partial \mathbf{F}}{\partial \rho} \cdot \mathbf{u} + \mathbf{F} \cdot \frac{\partial \mathbf{u}}{\partial \rho} + \mu \frac{\partial V}{\partial \rho} + \lambda_1 - \lambda_2 = 0. \quad (5.3)$$

2. Constraints

$$V \leq V^*, \quad (5.4)$$

$$\underline{\rho} \leq \rho \leq 1. \quad (5.5)$$

3. Complementarity conditions

$$\mu c = 0, \quad (5.6)$$

$$\lambda_{1_i} s_{1_i} = 0, \quad \lambda_{2_i} s_{2_i} = 0, \quad (5.7)$$

$$\mu \geq 0, \quad (5.8)$$

$$\lambda_{1_i} \geq 0, \quad \lambda_{2_i} \geq 0. \quad (5.9)$$

Assuming that the lower and upper side constraints are not active, we obtain  $\lambda_1 = \lambda_2 = 0$ . When they are active, they can be excluded from the Lagrangian formulation (5.2). Thus, the stationarity condition can be written as

$$\frac{\partial \mathcal{L}}{\partial \rho} = \frac{\partial \mathbf{F}}{\partial \rho} \cdot \mathbf{u} + \mathbf{F} \cdot \frac{\partial \mathbf{u}}{\partial \rho} + \mu \frac{\partial V}{\partial \rho} = 0. \quad (5.10)$$

From the sensitivity of the displacement (4.7), the stationarity condition becomes:

$$\frac{\partial \mathcal{L}}{\partial \rho} = \mathbf{u} \cdot \frac{\partial \mathbf{F}}{\partial \rho} - \mathbf{F} \cdot \mathbf{K}^{-1} \cdot \frac{\partial \mathbf{K}}{\partial \rho} \cdot \mathbf{u} + \mathbf{F} \cdot \mathbf{K}^{-1} \cdot \frac{\partial \mathbf{F}}{\partial \rho} + \mu \frac{\partial V}{\partial \rho} = 0, \quad (5.11)$$

where  $\mathbf{K}$  is the global stiffness matrix.

Therefore, from the equilibrium equation  $\mathbf{K} \cdot \mathbf{u} = \mathbf{F}$ , and the external forces which can be decomposed into the static loads and the design-dependent loads, the stationarity condition can be written as

$$\frac{\partial \mathcal{L}}{\partial \rho} = 2\mathbf{u} \cdot \frac{\partial \mathbf{F}_d}{\partial \rho} - \mathbf{u} \cdot \frac{\partial \mathbf{K}}{\partial \rho} \cdot \mathbf{u} + \mu \frac{\partial V}{\partial \rho} = 0, \quad (5.12)$$

where  $\mathbf{F}_d$  is the design-dependent load vector defined in chapter 4 (4.4).

The stiffness matrix for an element of the cellular automata discretisation is  $\mathbf{K}_i = \bar{\rho}_i^p \mathbf{K}_0$ , where  $\bar{\rho}_i$  is the element density obtained by an average density interpolation (2.34) for the  $N_{cell}$  cells surrounding the element. The parameter  $p$  represents the penalisation parameter ( $p \geq 3$ ) defined by the well known SIMP approach [7–11] and  $\mathbf{K}_0$  is the base stiffness matrix of an element. The cell volume can be written as  $V_C = \rho_C \bar{V}_C$ , where  $\rho_C$  is the cell density and  $\bar{V}_C$  is the base cell volume.

The local optimality for the cellular automata discretisation is obtained from the stationarity of the Lagrangian function (5.2) with respect to design variables:

$$\frac{\partial \mathcal{L}}{\partial \rho_C} = 0, \quad \text{for } C = 1 \dots N_{cells}. \quad (5.13)$$

From the stiffness matrix relation  $\mathbf{K}_i = \bar{\rho}_i^p \mathbf{K}_0$  and the average density interpolation definition (2.34), we can remark that only the  $N_{element}$  elements of the Moore neighbourhood structure (see figures 2.4(d) and 2.4(e)) are directly dependent on the cell density  $\rho_C$ . Moreover, from the relation of the design-dependent load defined in equation (4.4) it can be seen that only the forces applied at the cell  $C$  in question and its direct East cell neighbour  $E$ , when the design-dependent load direction is assumed to be from West, are dependent on the the cell density  $\rho_C$ . Thus, the derivative of the Lagrangian function (5.2) with respect to the design variable  $\rho_C$  becomes:

$$\frac{\partial \mathcal{L}}{\partial \rho_C} = \sum_{i=1}^{N_{element}} \left( -\mathbf{U}_{C_i} \cdot \frac{\partial \mathbf{K}_{C_i}}{\partial \rho_C} \cdot \mathbf{U}_{C_i} \right) + 2 u_E^x \frac{\partial F_{dE}^x}{\partial \rho_C} + 2 u_C^x \frac{\partial F_{dC}^x}{\partial \rho_C} + \mu \bar{V}_C = 0, \quad (5.14)$$

where the vector  $\mathbf{U}_{C_i}$  represents the displacements of all  $N$  cells surrounding the element  $C_i$ ,  $u_C^x$  and  $u_E^x$  are the cell displacements in  $x$  direction of the center cell  $C$  and the East cell  $E$  of the Moore neighbourhood structure, respectively. The forces  $F_{dC}^x$  and  $F_{dE}^x$  are the design-dependent loads (4.4) applied at the center cell  $c$  and the East cell  $E$  of the Moore neighbourhood structure, respectively.

Using the stiffness matrix relation  $\mathbf{K}_{C_i} = \bar{\rho}_{C_i}^p \mathbf{K}_0$ , equation (5.14) becomes:

$$\frac{\partial \mathcal{L}}{\partial \rho_C} = \sum_{i=1}^{N_{element}} \left( -\frac{\partial \bar{\rho}_{C_i}^p}{\partial \rho_C} \mathbf{U}_{C_i} \cdot \mathbf{K}_0 \cdot \mathbf{U}_{C_i} \right) + 2 u_E^x \frac{\partial F_{dE}^x}{\partial \rho_C} + 2 u_C^x \frac{\partial F_{dC}^x}{\partial \rho_C} + \mu \bar{V}_C = 0. \quad (5.15)$$

From the average density interpolation (2.34), the derivative of the element density  $\bar{\rho}_{C_i}^p$  with respect to the cell density  $\rho_C$  can be expressed as

$$\frac{\partial \bar{\rho}_{C_i}^p}{\partial \rho_C} = \frac{p}{N_{cell}} \frac{\bar{\rho}_{C_i}^{2p}}{\rho_C^{p+1}}. \quad (5.16)$$

Expressing the Lagrangian as a function of the design variable  $\rho_C$ , the stationarity condition (5.15) becomes:

$$\frac{\partial \mathcal{L}}{\partial \rho_C} = -\frac{p}{N_{cell}} \frac{1}{\rho_C^{p+1}} \sum_{i=1}^{N_{element}} \left( 2 \bar{\rho}_{C_i}^{2p} \tilde{U}_{C_i} \right) + 2 u_E^x \frac{\partial F_{dE}^x}{\partial \rho_C} + 2 u_C^x \frac{\partial F_{dC}^x}{\partial \rho_C} + \mu \bar{V}_C = 0, \quad (5.17)$$



where  $\tilde{U}_{C_i}$  is the element strain energy of the base material defined as

$$\tilde{U}_{C_i} = \frac{1}{2} \mathbf{U}_{C_i} \cdot \mathbf{K}_0 \cdot \mathbf{U}_{C_i}. \quad (5.18)$$

Based on the first order conditions (5.3) and (5.5), the optimisation problem (5.1) is equivalent to the following local minimisation problem:

$$\min_{\rho_C} \frac{\Phi^*}{\rho_C^p} + \Psi^*(\rho_C) + \frac{\mu}{2} \rho_C, \quad (5.19a)$$

subject to,

$$\rho \leq \rho_C \leq 1. \quad (5.19b)$$

where  $\Phi^*$  is the cell strain energy density which is calculated based on the design variable values of the previous iteration and is expressed by

$$\Phi^* = \frac{1}{N_{cell} \bar{V}_C} \sum_{i=1}^{N_{element}} \left( \bar{\rho}_{C_i}^{2p} \tilde{U}_{C_i} \right), \quad (5.20)$$

similarly to the expression (2.41) in chapter 2. The function  $\Psi^*(\rho_C)$ :

$$\Psi^*(\rho_C) = \frac{1}{\bar{V}_C} \left( u_E^x F_{d_E}^x + u_C^x F_{d_C}^x \right). \quad (5.21)$$

depends on the design variables.

The design update rules are derived from the solution of the local optimisation problem (5.19). This problem (5.19) is equivalent to the convex local minimisation problem (2.25) when the forces are independent of design variables. Its solution is given by (2.27). In the general case, when design-dependent loads are present, the local problem (5.19) is solved using a Brent method [71].

The Lagrange multiplier  $\mu$  is the only global quantity that is involved in the local problem (5.19) and it plays a key role in enforcing satisfaction of the volume constraint. The updating of this parameter is done iteratively (see chapter 2).

### 5.3 Convex approximations

From several numerical examples, it is observed that the solution tends to oscillate from one iteration to another and does not converge. As can be observed from the

plots of the local objective function (5.19) expressed in the design variable  $\rho_1$ , see figure 5.1, the solution of the local minimisation problem (5.19) for cell 1 of the example treated in numerical section 5.7.2.1 jumps from  $\rho_1 = \underline{\rho}$  at iteration 2 to  $\rho_1 = 1$  at iteration 3. Moreover, from the topologies represented in figure 5.1, it is clear that about half of the left edge and some cells at the top of the structure jump from low cell density solutions at iteration 2 to  $\rho_C = 1$  at iteration 3. This manifestation of divergence can be explained by the possible non-convexity of the objective function.

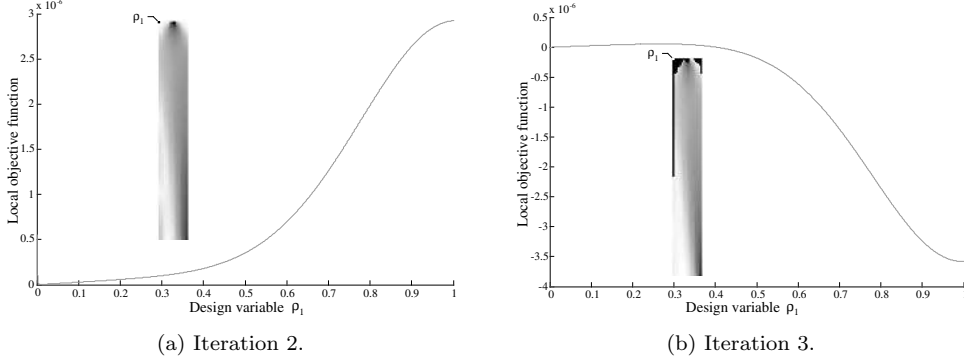


Figure 5.1: Non-convexity of local objective function and reversal of variation between iterations.

The objective function of the local minimisation problem (5.19) is convex when fixed loads are applied or the derivative of  $\Psi^*(\rho_C)$  with respect to design variable  $\rho_C$  vanishes. The solution for this type of local minimisation problem is given by (2.27), and the optimisation problem (5.1) converges for any given structure problems. However, when design-dependent loads are taken into account in the formulation, the derivative of the term  $\Psi^*(\rho_C)$  with respect to design variable  $\rho_C$  no longer vanishes. Therefore the local minimisation problem with respect to the design variable  $\rho_C$  (5.19) can non-convex and oscillation may appear in the optimisation process. To avoid this convergence difficulty, the function  $\Psi^*(\rho_C)$  is convexified with respect to the design variable  $\rho_C$ . The major factors that may cause the non-convexity of the function  $\Psi^*(\rho_C)$  are the peak function interpolation  $g$ , see equation(4.5), and the work  $u^x F_d^x$  of the load dependent forces. From the plot of the peak function interpolation  $g$ , see figure 4.1(b), it can be seen that it is almost convex as the parameter  $2\sigma^2$  is large enough (e.g.  $2\sigma^2 = 0.1$ ). When the displacement  $u^x$  is positive the work  $u^x F_d^x$  is convex, and when it is negative the work  $u^x F_d^x$  becomes concave and needs to be convexified. A linear approximation

is chosen to approximate the work  $u^x F_d^x$  when the displacements are negative. The function  $\Psi^*(\rho_C)$  becomes:

$$\Psi^*(\rho_C) = \frac{1}{\bar{V}_C} \left( \begin{array}{l} \left\{ \begin{array}{ll} u_E^x F_{d_E}^x, & u_E^x F_{d_E}^x \geq 0 \\ \rho_C u_E^x \frac{\partial F_{d_E}^x(\rho_C^{old})}{\partial \rho_C} + O(\rho_C^2), & u_E^x F_{d_E}^x < 0 \end{array} \right. + \\ \left\{ \begin{array}{ll} u_C^x F_{d_C}^x, & u_C^x F_{d_C}^x \geq 0 \\ \rho_C u_C^x \frac{\partial F_{d_C}^x(\rho_C^{old})}{\partial \rho_C} + O(\rho_C^2), & u_C^x F_{d_C}^x < 0 \end{array} \right. \end{array} \right). \quad (5.22)$$

## 5.4 Relaxation

The design variables are updated using a Jacobi scheme to preserve a symmetry on the solution (see chapter 2). It is observed that the optimisation process, in some cases, fails to converge. An alternative way to ensure convergence of a diverging Jacobi iterative process is to use the Jacobi Over Relaxation scheme (JOR) [69, 70]. The JOR scheme is defined as

$$\rho^{k+1} = \beta \rho^* + (1 - \beta) \rho^k, \quad (5.23)$$

where  $\rho^*$  is the solution produced by a Jacobi scheme and  $\beta$  is the relaxation factor ( $0 < \beta < 2$ ).

The value of  $\beta$  can affect the rate of convergence. The choice of this factor is not necessarily easy, and depends on many parameters of the optimisation process. In this chapter, we are interested in convergence rather than faster convergence. The value  $\beta > 1$  is used to speed up convergence while the value  $\beta < 1$  is used to help establish convergence, evidently the JOR scheme is reduced to Jacobi scheme for  $\beta = 1$ .

## 5.5 Explicit constraint

Optimal topologies based on design-dependent loads exhibit a grey design as observed in chapter 4. To obtain a black/white design, the intermediate densities are controlled by an explicit constrain [65–68] (see equation (4.20) in chapter 4) defined as

$$h(\rho) = \int_{\Omega} (1 - \rho)(\rho - \underline{\rho}) d\Omega \leq \epsilon_p, \quad (5.24)$$

where  $\epsilon_p$  is the allowable grey level.

The Lagrangian function (5.2) for the present optimisation problem (5.1) including

the explicit constraint (5.24) can be rewritten as

$$\mathcal{L} = \mathbf{F} \cdot \mathbf{u}^* + \mu (V - V^* + c^2) + \lambda_1 \cdot (\rho - 1 + \mathbf{s}_1^2) + \lambda_2 \cdot (\underline{\rho} - \rho + \mathbf{s}_2^2) + \kappa (h - \epsilon_p + m^2), \quad (5.25)$$

where  $\kappa \geq 0$  is a Lagrange multiplier associated with the explicit constraint and  $m$  is the corresponding slack variable.

Thus, the local minimisation problem (5.19) becomes:

$$\min_{\rho_C} \frac{\Phi^*}{\rho_C^p} + \Psi^*(\rho_C) + \frac{\mu}{2} \rho_C + \frac{\kappa}{2} (1 - \rho_C)(\rho_C - \underline{\rho}), \quad (5.26a)$$

subject to,

$$\underline{\rho} \leq \rho_C \leq 1. \quad (5.26b)$$

The term  $(1 - \rho_C)(\rho_C - \underline{\rho})$  in the local objective function (5.26) is a concave function. To convexify this function, a linear approximation around the most recent design is adopted. Therefore the local minimisation problem with respect to the design variable  $\rho_C$  can be written as

$$\min_{\rho_C} \frac{\Phi^*}{\rho_C^p} + \Psi^*(\rho_C) + \frac{\mu}{2} \rho_C + \frac{\kappa}{2} \rho_C (1 + \underline{\rho} - 2\rho_C^{old}), \quad (5.27a)$$

subject to,

$$\underline{\rho} \leq \rho_C \leq 1. \quad (5.27b)$$

In an early paper [66], it is proven that a solution of the optimisation problem (5.1) including the explicit constraint (5.24) exists. The choice of an appropriate value of the initial parameter  $\kappa$  and its update to get a black/white design are not obvious, a simple way to update the parameter  $\kappa$  is selected as follows:

$$\kappa^{j+1} = \kappa^j (1 + \gamma), \quad (5.28)$$

where  $\gamma$  is defined by the user.

## 5.6 Algorithm

The algorithm starts without including the explicit constraint (5.24) in the formulation. Then, the penalisation parameter  $p$  is increased from an initial value  $p_i$  to

---

**Algorithm 5.1** Multigrid design algorithm with design-dependent loading
 

---

```

input
     $\mathbf{u}, \mathbf{f}, \rho$ 
output
     $\mathbf{u}, \rho$ 
    % Initialisation
     $k = 0, j = 0, \mu^{(0)}, \kappa^{(0)} = 0$ 
repeat
     $\kappa^{(j+1)} = \kappa^{(j)}(1 + \gamma)$ 
    if  $j = 1$  then
         $p_i = p_f, \kappa^{(1)} = \kappa_i$ 
    end if
    for  $p = p_i$  to  $p_f$  do
        repeat
            for each element do
                % Element density interpolation
                 $\bar{\rho}^{(k+1)}$ 
            end for
            % Update forces
             $\mathbf{f}_d^{(k+1)}$ 
            repeat
                 $MG \text{ Analysis}(\mathbf{u}^{(k)}, \mathbf{f}^{(k+1)}, \rho^{(k+1)})$ 
                 $\mathbf{u}^{(k+1)}$ 
            until  $\frac{\|\mathbf{R}\|}{\|\mathbf{R}_0\|} \leq \epsilon_r$ 
            repeat
                for each cell do
                    % Design update rule (5.27),  $\rho_C^*$ 
                    if  $\left\| \frac{V^{(k+1)}}{\eta V_G} - 1 \right\| > \Delta V$  then
                        % Update Lagrange Multiplier
                         $\mu^{(k+1)}$ 
                    end if
                end for
                until  $\left| \frac{V^{(k+1)}}{\eta V_G} - 1 \right| \leq \Delta V$ 
                % Relaxation
                 $\rho^{k+1} = \beta \rho^* + (1 - \beta) \rho^k$ 
                 $W^{(k+1)} = \mathbf{u}^{(k+1)} \cdot \mathbf{f}^{(k+1)}$ 
                 $k = k + 1$ 
                until (five successive  $\left| \frac{W^{(k+1)}}{W^{(k)}} - 1 \right| \leq \epsilon_c$  and  $(|\rho^{(k+1)} - \rho^{(k)}| \leq \epsilon)$ 
            end for
             $h(\rho) = \sum_{C=1}^{N_{cells}} (1 - \rho_C)(\rho_C - \underline{\rho}) \bar{V}_C$ 
             $j = j + 1$ 
        until  $\frac{h(\rho)}{V_T} \leq \epsilon_p$  or  $j = 101$ 
    
```

---

a desirable penalisation value  $p_f$  via a continuation method to avoid local minima. The topology is obtained using the multigrid algorithm described in section 3.4 by including the updating of the forces that depend on the design variables. Next, the parameter  $\kappa$  is updated using (5.28). The process continues until the explicit constraint (5.24) is satisfied or a maximum of 100 iterations is reached (see algorithm 5.1).

## 5.7 Numerical examples

Some numerical examples of two and three dimensional topology optimisation problems with design-dependent loading are presented in the following. The obtained solutions are compared to results given by the MMALS algorithm described in chapter 4. In this section, wind loading and hydrodynamic water pressure loading are considered as design-dependent loadings. The same wind load function defined in equation (4.4) is used to define the hydrodynamic water pressure load function. The algorithm 5.1 is implemented in a Linux C++ environment and tested on a Dual core AMD Opteron(tm) machine with a processor frequency of 2400 Mhz and 8 GByte memory.

For all applications the penalisation parameter  $p$  is increased from 1 up to 3 by increments of 0.1 to avoid local minima. The tolerance for the design and for the compliance were set to 0.05 and  $10^{-3}$  respectively, and a lower bound of  $10^{-3}$  was adopted for the density. The multigrid accelerated cellular automata algorithm with **W** cycle is used to solve the topology optimisation problem for two and three dimensional problems. The smoothing parameter is set to  $S = 2$ . The residual tolerance ratio for the finest and the coarsest grids are set to 0.1 and 0.6 respectively. The peak function defined in chapter 4 is again used to regularise the design-dependent load function. The value of  $2\sigma^2$  is fixed at 0.1, and the relaxation factor  $\beta$  of equation (5.23) is set to 0.1. The initial Lagrange multiplier  $\kappa_i$  associated to the explicit constraint, the increment  $\gamma$  and the allowable grey level  $\epsilon_p$  are obtained from numerical experiments and set to  $3 \cdot 10^{-9}$ , 0.2 and  $10^{-6}$ , respectively.

### 5.7.1 Clamped column

To further illustrate the influence of design-dependent loads on topology, we consider a bi-clamped column of aspect ratio 8 as shown in figure 5.2. The column dimensions are 120 m in height, 15 m in length and 15 m in width. The structure is clamped

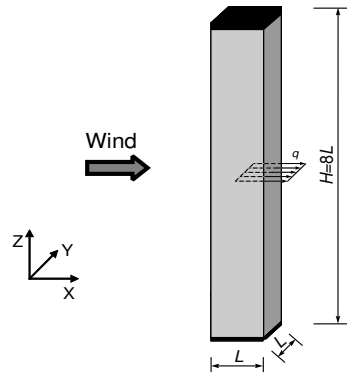


Figure 5.2: Domain.

from the top and bottom, and a uniform load is applied to the middle of the structure with a magnitude of  $8.33 \cdot 10^{-3}$  MN/m. The column is subjected to a wind coming from the West with a speed of 25 m/s. The material used is steel with a Young's modulus  $E = 200000$  MPa, and Poisson's ratio  $\nu = 0.3$ . The volume fraction is set to  $\eta = 0.5$ .



(a) Topology without wind loading.



(b) Topology with wind loading.

Figure 5.3: 2D topology of the bi-clamped column.

### 5.7.1.1 Clamped two-dimensional column

In this example the problem is treated as a two-dimensional case in the XZ plane. In the next example, a three-dimensional version of this problem will be studied. The

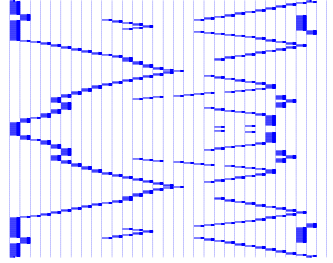


Figure 5.4: Wind distribution.

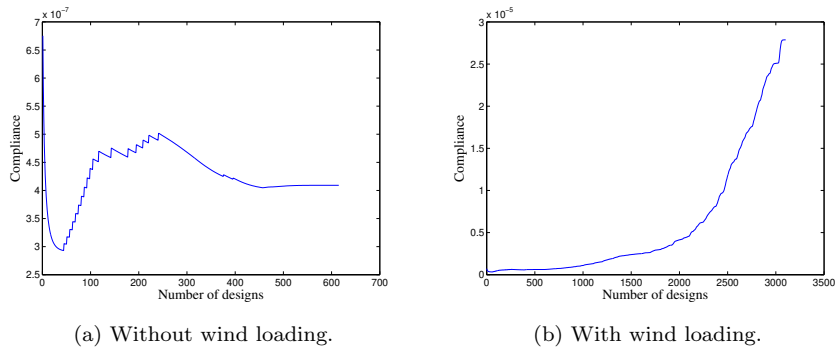


Figure 5.5: Compliance history.

algorithm was run on a  $33 \times 257$  cells mesh. The converged topologies are shown in figure 5.3. The converged topology in figure 5.3(a) requires 616 design iterations in 63 seconds, and the topology in figure 5.3(b) requires 3103 design updates for a run time of 424 seconds. The topology without wind loading (see figure 5.3(a)) is a symmetric design that carries the force  $F$ . It is composed of two cantilever like support structures at the top and bottom and a truss structure at the middle of the column. The topology with wind loading (see figure 5.3(b)) is an asymmetric



design. It consists of two cantilevers and a compression arch structure at the middle to support the incoming wind. Figure 5.4 shows the distribution of the wind loads over all the cells of the structure. The compliances computed with wind loads for both optimum structures are  $3.81 \cdot 10^{-5}$  MNm for the topology without wind loading and  $2.78 \cdot 10^{-5}$  MNm for the topology with wind loading, respectively, which corresponds to a 27% difference between the two solutions. Figure 5.5 shows the history of compliance for the cases with and without wind loading. It can be seen from figure 5.5(b) that the compliance is increased with respect to the number of designs, this is due mainly to the explicit constraint that leads the solution from a grey design to a black/white design, therefore, the wind loads at the surface facing the wind increase in value with respect to the formulation defined in equation (4.4).

### 5.7.1.2 Comparison with the MMALS algorithm

To make a comparison with the actual algorithm, based on multigrid accelerated CA, the same example was run using the MMALS algorithm. The value of  $\beta$  used is 0.05 and the penalisation parameter is fixed at 3. The optimum topologies for the cases without and with wind loading are represented in figures 5.6(a) and 5.6(b), respectively. The algorithm converged within a total of 1463 design updates for the case without wind loading and 2046 for the case considering wind loading. The run time for the MATLAB implementation is 13011 seconds without considering wind loading and 28104 seconds with wind loading. The huge difference in time between the designs given by the multigrid accelerated CA and the MMALS algorithms is due to the different programming environment.



(a) Topology without wind loading.



(b) Topology with wind loading.

Figure 5.6: 2D topology of the bi-clamped beam obtained with MMALS solver.

As can be seen, the optimum topologies without wind loading, using MMALS and multigrid accelerated CA algorithms, are approximately the same (see figures 5.3(a) and 5.6(a)). However, the topology obtained by considering wind loading in the formulation, obtained using MMALS algorithm, shows a pure compression arch at the middle of the structure (see figure 5.6(b)) as opposed to that shown in figure 5.3(b) where two extra elements connect the compression arch with the cantilevers. Figure 5.7 shows the history of compliance for the cases with and without wind loading. The compliance for the design in figure 5.6(a) is  $3.65 \cdot 10^{-5}$  MNm and for the design in figure 5.6(b) it is  $1.75 \cdot 10^{-5}$  MNm, this corresponds to a 52% difference. It can be seen from the compliance comparisons that the MMALS solver gives stiffer designs than the multigrid accelerated CA solver. However, the results of the numerous numerical examples tested reveal that the multigrid accelerated CA solver is more robust than the MMALS solver.

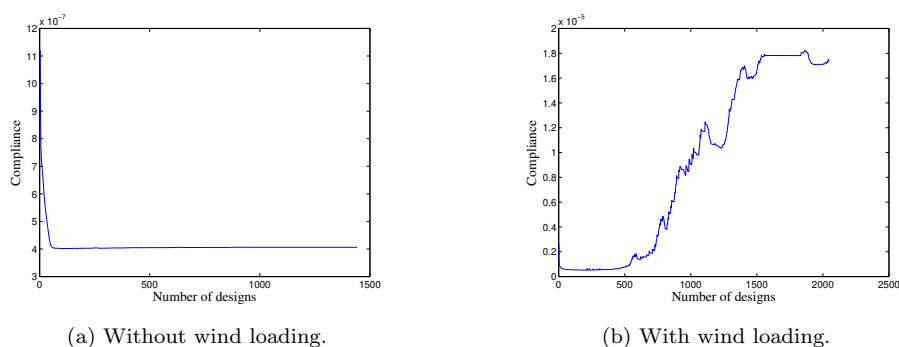


Figure 5.7: Compliance history obtained with MMALS solver.

### 5.7.1.3 Clamped three-dimensional column

The bi-clamped column example is now studied in the three-dimensional case. The problem is run with  $33 \times 33 \times 257$  cells using the multigrid accelerated CA algorithm for design-dependent. Figure 5.8 shows different views of the optimum topology for the cases with and without wind loading. Without wind loading the algorithm converges in a total of 445 design updates in 5382 seconds, and with wind loading it converges in a total of 1903 design updates in a run time of 28783 seconds.

From figures 5.8(a) and 5.8(b), it can be seen that the outer shapes of the optimum 3D topology design are close to those obtained for the 2D case (see figure 5.3). How-

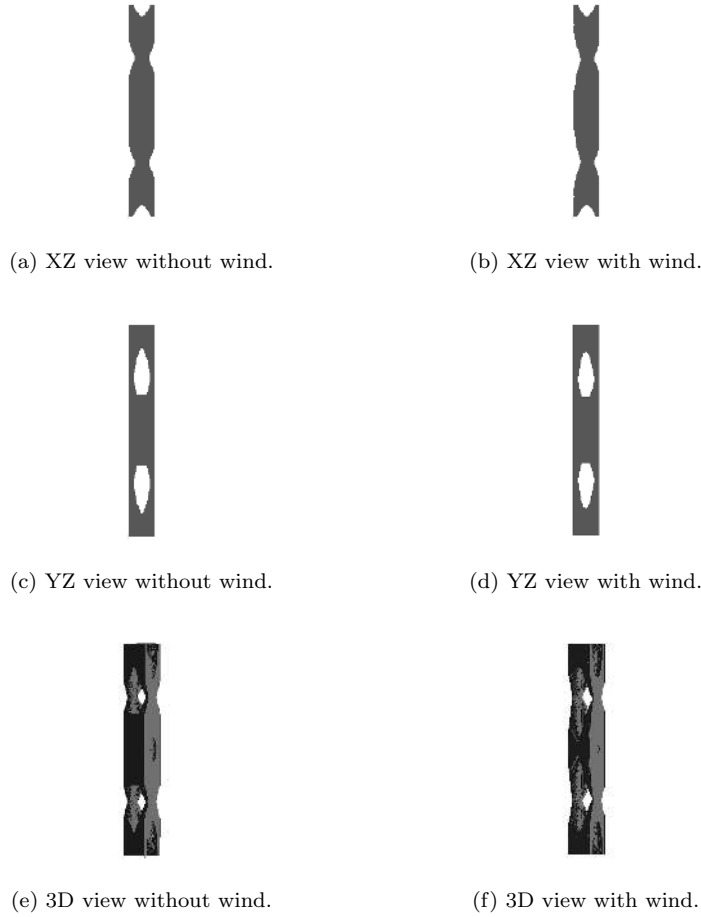


Figure 5.8: 3D topology of the bi-clamped column.

ever, the interior designs look different. The compliance for the design without wind loading is  $8.45 \cdot 10^{-4}$  MNm, while for the design with wind loading is  $1.25 \cdot 10^{-5}$  MNm. This means that including wind loading in the formulation increases structural stiffness by 98%. This huge difference in structural stiffness can be explained by the fact that including wind loads in the formulation tends to reduce the contact surface of the structure with wind (see figures 5.8(b) and 5.8(f)). Changing the value of volume fraction to  $\eta = 0.2$  gives a rather truss like structure that is closer to the 2D solution (see figure 5.9). The compression arch can be clearly seen on the sides of

the structure in figure 5.9(f). These sides are connected with a truss structure and two bar elements at the middle and at the west side of the structure, respectively (see figures 5.9(d) and 5.9(f)). As reported in chapter 4, including wind loading in the formulation can allow for holes facing the wind loads (see figures 5.8(d) and 5.9(d)) as opposed to the transmissible loads method [38].



(a) XZ view without wind.



(b) XZ view with wind.



(c) YZ view without wind.



(d) YZ view with wind.



(e) 3D view without wind.



(f) 3D view with wind.

Figure 5.9: 3D topology of the bi-clamped column with  $\eta = 0.2$ .

## 5.7.1.4 Clamped three-dimensional column with multiple loads

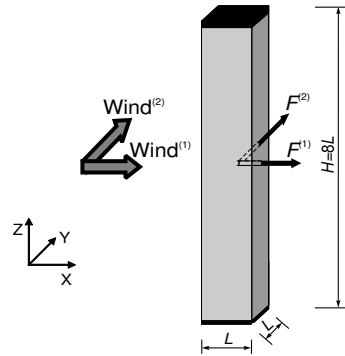


Figure 5.10: Domain.

The bi-clamped example will now be considered for design subject to two load cases. The different loads are represented in figure 5.10. The first load case is composed from a concentrated load applied at the middle of the structure in  $x$  direction with a magnitude of 0.125 MN, and a wind coming from West with a speed of 25 m/s. The second load case is defined by a concentrated load applied at the middle of the structure in the  $y$  direction with the same magnitude (0.125 MN) as defined in the first load case and a wind coming from the South with a speed of 30 m/s.

The algorithm is run with the same discretisation as defined earlier for the 3D case and with a volume fraction  $\eta = 0.2$ . The optimum topologies with and without wind loading are represented in different views in figure 5.11. The optimum topology without wind loading converges after 659 design updates in 15489 seconds, and with wind loading the topology converges after 3253 design updates in 76596 seconds. The compliances of the topologies with and without wind loads are  $2.61 \cdot 10^{-3}$  MNm and  $8.92 \cdot 10^{-3}$  MNm, respectively. An explanation of this difference in compliance, which is about 71%, is that including wind loading in the formulation causes the algorithm to try to minimise the wind contact on the structure by allowing the wind to go through the domain (see figure 5.11).

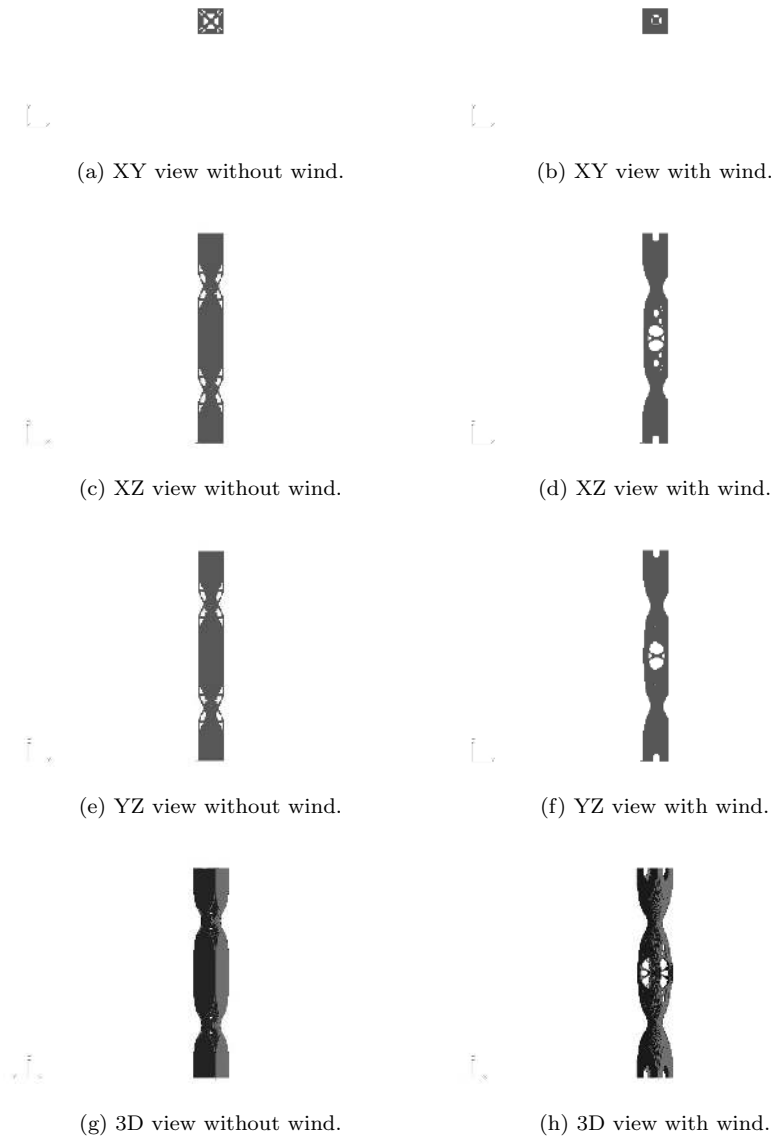


Figure 5.11: 3D topology of the bi-clamped beam with two load cases.

### 5.7.2 Off-shore wind turbine support structure

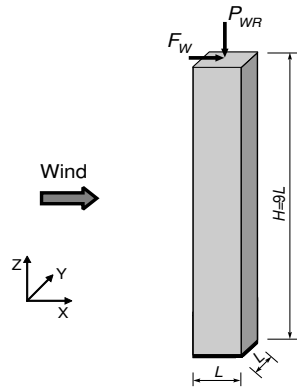


Figure 5.12: Domain.

The example of the off-shore wind turbine support structure studied in chapter 4 is reconsidered here (see figure 5.12) with the domain dimensions are 120 m in height, 13.34 m in length and 13.34 m in width. The base dimensions of the structural domain have been modified due to the multigrid discretisation requirement. The same hub and rotor weight  $P_{WR}$ , and the same rotor thrust  $F_W$  are considered in this example. The structure is clamped at the bottom. The incoming wind flows from West with a speed of 25 m/s. The material used is steel with a Young's modulus  $E = 200000$  MPa, and Poisson's ratio  $\nu = 0.3$ . The volume fraction is set to  $\eta = 0.5$ .

#### 5.7.2.1 Two-dimensional off-shore wind turbine support structure

The multigrid design algorithm with design-dependent loads is run for a  $17 \times 145$  cell grid model. The optimal topologies of the off-shore wind turbine support structure without and with wind loading are represented in figure 5.13. The algorithm without explicit constraint and without wind loading converges in a total of 240 design updates and the run time is 16 seconds. With wind loading the algorithm converges in a total of 255 design updates in 18 seconds. It can be observed that the topologies without and with wind loading but not including the explicit constraint (5.24) in the formulation are almost the same (see figures 5.13(b) and 5.13(a)). The grey elements appear more in the topology with wind loading at the surface facing the wind loads direction (see figure 5.13(b)). This means including wind loads in the formulation favours a grey design. Moreover, the wind forces acting on the grey

elements according to the wind load formulation defined in equation (4.4) are very small (see figure 5.14(a)).



(a) Topology without wind loading and without explicit constraint.



(b) Topology with wind loading and without explicit constraint.



(c) Topology without wind loading and with explicit constraint.



(d) Topology with wind loading and with explicit constraint.

Figure 5.13: 2D topology of the wind turbine support.

However, by considering the explicit constraint (5.24) in the formulation, the grey areas disappear and this gives more realistic solutions (see figures 5.13(c) and 5.13(d)). The converged solution without wind loading requires 716 design iterations with a run time of 60 seconds, and the solution with wind loading requires 2621 design updates in 216 seconds. It can be observed that the topology without wind is a simple column (see figure 5.13(c)), whereas the topology subject to wind loads is a column supported by extra elements at the bottom of the structure (see figure 5.13(d)). Moreover, the wind loads are distributed throughout the surface facing the wind with an approximately constant magnitude (see figure 5.14(b)). The compliance of



the design without considering wind loads is  $4.39 \cdot 10^{-3}$  MNm and the compliance of the design with wind is  $1.41 \cdot 10^{-3}$  MNm, which corresponds to 68% of difference. Figure 5.15 shows the compliance history. It is remarked that the compliance considering wind loading kept increasing with the number of design iterations (see figure 5.15(b)). This is due mainly to the elimination of grey areas by the explicit constraint (5.24).

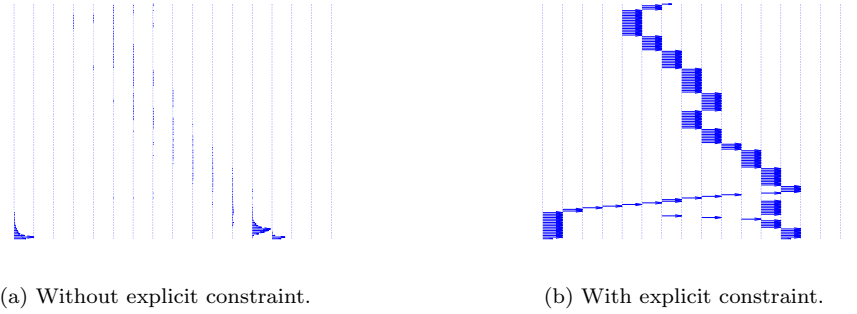


Figure 5.14: Wind distribution.

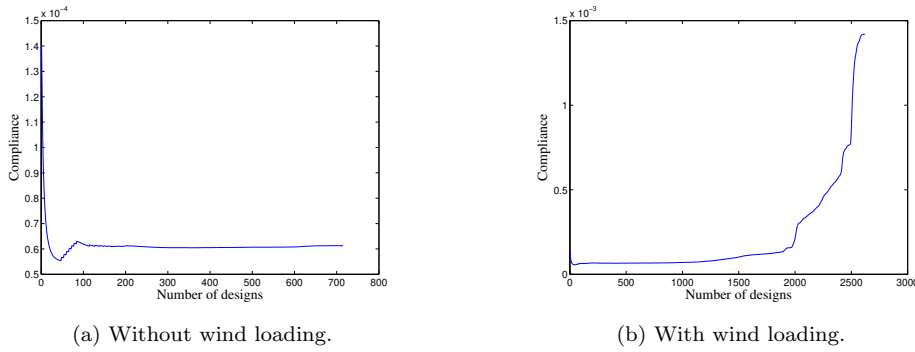


Figure 5.15: Compliance history.

For some cases, it occurred that the topology produced without design-dependent loading was stiffer than the topology generated considering design-dependent loading in the formulation (see figure 5.16). The volume fraction used for this specific case is  $\eta = 0.3$ . The compliance for the topology in figure 5.16(a) is  $8.12 \cdot 10^{-3}$  MNm, while

for the topology in figure 5.16(b) it is  $2.09 \cdot 10^{-2}$  MNm. In this specific example, the optimisation process probably could not avoid a local minimum for the problem with design-dependent loading and gave a weaker design than expected. A suggested idea to be explored to avoid being trapped at local minima while taking design-dependent loading into account is to consider two load cases: one without design-dependent loading and one with design-dependent loading.



(a) Topology without wind loading.

(b) Topology with wind loading.

Figure 5.16: 2D topology of the wind turbine support with  $\eta = 0.3$ .

### 5.7.2.2 Three-dimensional off-shore wind turbine support structure

The same off-shore wind turbine support structure example is studied in the three-dimensional case. The domain is discretised with  $17 \times 17 \times 145$  cells. The multigrid accelerated cellular automata algorithm without considering wind loads in the formulation converges in a total of 308 design updates in 524 seconds, and it converges in a total of 1682 design updates in 3826 seconds with wind loading. The optimum topologies of the off-shore wind turbine support structure designed with and without wind loading are represented in figure 5.17. The obtained topologies are almost similar to those obtained in the 2D case (see figure 5.13). The topology without considering wind loading is a simple column. With wind loading considered, the topology is a column held by extra elements at the base. The compliance calculated for the design subject to wind loads is  $1.30 \cdot 10^{-3}$  MNm, while the compliance for the design without wind loads is  $4.88 \cdot 10^{-3}$  MNm, this corresponds to 73% difference in stiffness between the two solutions. The compliances calculated for 2D and 3D are reasonably close.

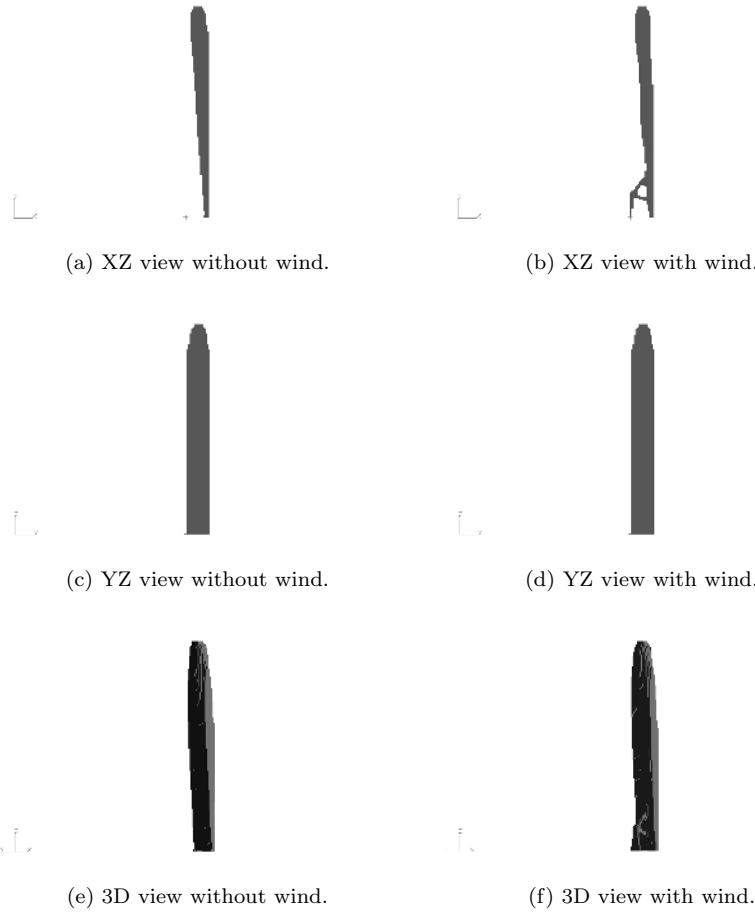


Figure 5.17: 3D topology of the wind turbine support.

### 5.7.2.3 Comparison with the MMALS algorithm

The same example was run using the MMALS algorithm. The value of  $\beta$  used is 0.05 and the penalisation parameter is fixed at 3. The optimum two-dimensional topologies for the case without and with wind loading are represented in figures 5.18(a) and 5.18(b), respectively. The obtained topologies are simple columns for both cases. The compliances after postprocessing the designs are  $1.17 \cdot 10^{-2}$  MNm for the topology in figure 5.18(a) and  $9.71 \cdot 10^{-3}$  MNm for the topology in figure 5.18(b),

respectively, this corresponds to a 17% difference between the two compliances. The solution obtained using multigrid accelerated CA algorithm considering wind loading (see figure 5.13(d)) in the formulation 85% stiffer with respect to wind loading than the solution obtained using the MMALS algorithm (see figure 5.18(b)).



(a) Topology without wind loading.

(b) Topology with wind loading.

Figure 5.18: 2D topology of the wind turbine support obtained with MMALS solver.

For the three-dimensional case, the topologies obtained using the MMALS algorithm are represented in figure 5.19. The topology without wind loading (see figure 5.19(e)) is seen to be very similar to the topology obtained using the multigrid accelerated CA algorithm (see figure 5.17(e)). However, the topology considering wind loading is a column supported by two frames at the bottom of the structure (see figure 5.19(f)). To compare the compliances, the designs are postprocessed to force exact black/white topologies. The compliance of the design without considering wind loads is  $5.63 \cdot 10^{-3}$  MNm and the compliance with wind loads is  $1.09 \cdot 10^{-3}$  MNm, which corresponds to 80% difference between the two solutions. For this case, the topology in figure 5.19(f) obtained using the MMALS algorithm is stiffer by 16%, with respect to wind load, than the design in figure 5.17(f) given by the multigrid accelerated CA algorithm. Although the solutions obtained using the MMALS algorithm are stiffer than the solution given by the multigrid accelerated CA algorithm, the convergence for the latter is more stable. Moreover, the multigrid accelerated CA algorithm converges to a pure black/white solution, whereas the MMALS algorithm sometimes fails to converge to a black/white design (see figure 5.18(b)).

Figure 5.20 represents the deflection in XZ view of the three-dimensional topologies of the wind turbine support structure considering wind loading for the multigrid accelerated CA, and the MMALS algorithms. The maximum deflection for both designs is at the tip of the structure where the dead loads are assumed to be applied.

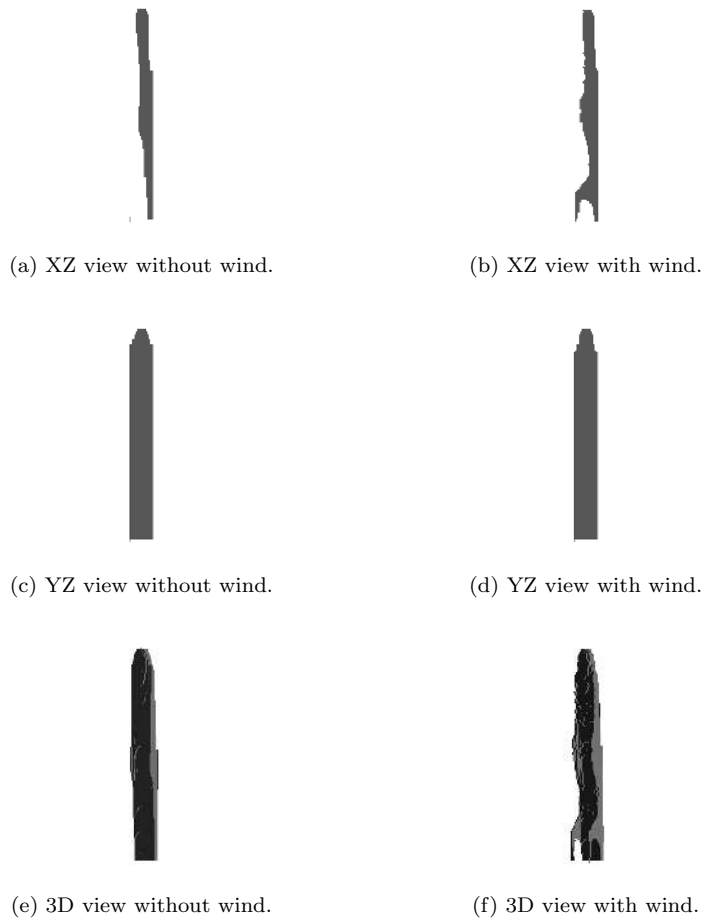


Figure 5.19: 3D topology of the wind turbine support.

The corresponding values are  $4.01 \cdot 10^{-3}$  m for the topology obtained using multigrid accelerated CA algorithm and  $2.82 \cdot 10^{-3}$  m for the topology obtained using MMALS algorithm.



(a) Multigrid accelerated CA algorithm.

(b) MMALS algorithm.

Figure 5.20: 3D topology deflection of the wind turbine support structure considering wind loading.

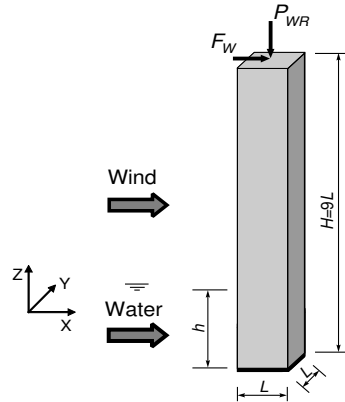


Figure 5.21: Domain.

#### 5.7.2.4 Three-dimensional off-shore wind turbine support structure subject to hydrodynamic water pressure loads

The off-shore wind turbine support structure is considered again to evaluate the influence of water depth combined with the wind effect. The structure is submerged by 30 m of sea water with 0.6 m/s water velocity (see figure 5.21). The hydrodynamic effect of water pressure is considered in this example. The same wind load function is used to define the water load function. The topology obtained subject to wind and water loads is represented in figure 5.22. The topology produced is almost the

same as the topology found when considering only wind loading in the formulation (see figure 5.17(f)) but with a larger base for the extra elements at the bottom of the structure. The compliance calculated without considering wind/water loading is  $6.15 \cdot 10^{-3}$  MNm and the compliance with wind and water loading is  $1.27 \cdot 10^{-3}$  MNm, which corresponds to 80% difference in structural stiffness with respect to these loads.

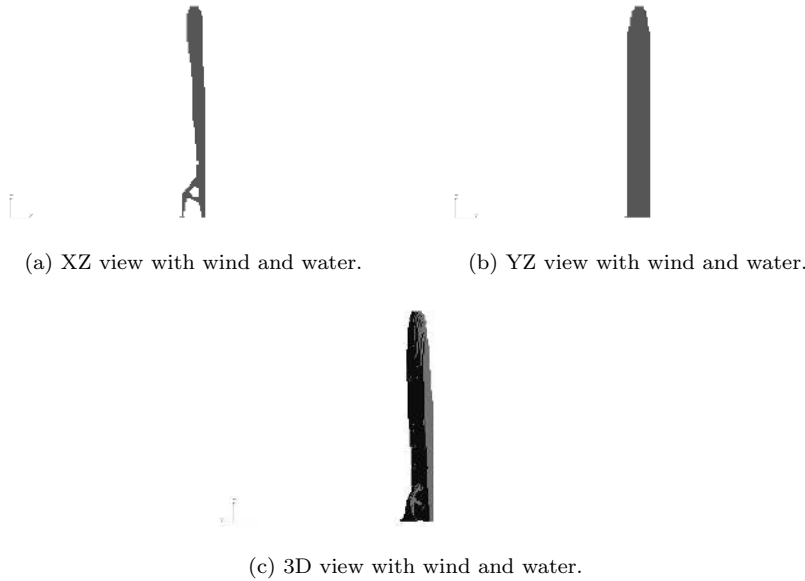


Figure 5.22: 3D topology of the wind turbine support with wind and water loading.

### 5.7.3 Multiple arch bridge

In this example, we study the multiple arch bridge problem defined in chapter 3 under the influence of hydrodynamic water pressure. The design domain is a cuboid with 120 m in length, 15 m in depth and 30 m in height. The considered bridge is clamped at the bottom of the domain and supports only a uniformly distributed traffic loading in addition to its own weight of magnitude  $0.15 \text{ MN/m}^2$  (see figure 5.23). The structure is submerged 7.5 m in the river where the water velocity is 3.5 m/s. The material used for the bridge is steel with a Young's modulus  $E = 200000 \text{ MPa}$ , and

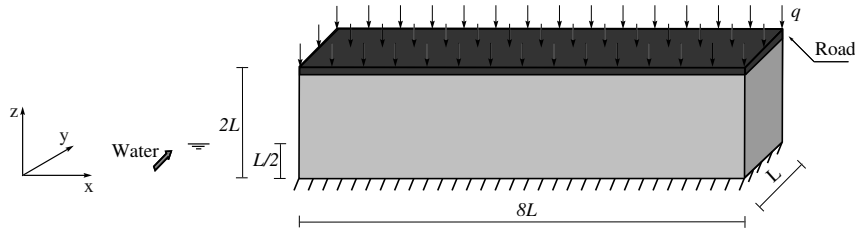


Figure 5.23: Domain.

Poisson's ratio  $\nu = 0.3$ . The residual tolerance ratio for the finest grid is fixed for this example at 0.07.

The domain is discretised into  $257 \times 33 \times 65$  cells and the volume fraction considered is  $\eta=0.1$ . The optimum bridges with and without the hydrodynamic water pressure influence are represented in figure 5.24. The solution in figure 5.24(e) converged after 977 design updates in 20813 seconds. However, the solution in figure 5.24(f) converged after 1418 design iterations for a run time of 30401 seconds. As can be observed from figure 5.24 there are slight differences between the two solutions. The topology under hydrodynamic water pressure produces larger columns at the surface facing the water flow with a slight difference in the shape of the columns at the base. The elements connecting the columns on both sides of the bridge are horizontal in the design without water flow and visibly oblique in the design with water flow. The compliance under the hydrodynamic water pressure influence is  $9.39 \cdot 10^{-2}$  MNm and the compliance without considering the hydrodynamic water pressure influence is 0.1 MNm, which corresponds to 6% in structural stiffness with respect to hydrodynamic loads.

Changing the initial Lagrange multiplier  $\kappa_i$  associated to the explicit constraint to  $3 \cdot 10^{-3}$  produced another local optimum (see figure 5.25). The compliance calculated for the topologies of figures 5.11(g) and 5.11(h) under the hydrodynamic water pressure influence are 3.33 MNm and 0.115 MNm, respectively. From these values it is clear that the local solutions given by  $\kappa_i = 3 \cdot 10^{-9}$  are stiffer designs, with respect to hydrodynamic loading, than those produced with  $\kappa_i = 3 \cdot 10^{-3}$ .





(a) XZ view without water flow.



(b) XZ view with water flow.



(c) YZ view without water flow.



(d) YZ view with water flow.



(e) 3D view without water flow.



(f) 3D view with water flow.

Figure 5.24: 3D topology of the multiple arch bridge with  $\kappa_i = 3 \cdot 10^{-9}$ .



(a) XZ view without water flow.



(b) XZ view with water flow.



(c) YZ view without water flow.



(d) YZ view with water flow.



(e) 3D view without water flow.



(f) 3D view with water flow.

Figure 5.25: 3D topology of the multiple arch bridge with  $\kappa_i = 3 \cdot 10^{-3}$ .

## 5.8 Concluding remarks

Multigrid accelerated cellular automata with design-dependent loading algorithm was implemented in this chapter. The optimality criteria interpreted as a local Kuhn-Tucker condition were formulated and used to construct the local design rule. The local optimisation problem was convexified to prevent convergence oscillation. The Jacobi over relaxation scheme was used instead of the Jacobi scheme to ensure convergence of the optimisation process. An explicit constraint was added to the optimisation formulation to control the intermediate density and lead the final design to almost pure black/white solutions. Optimal designs were generated under wind loading and hydrodynamic water pressure influences. It was shown that properly taking into account the design-dependent nature of loads such as wind pressure and hydrodynamic forces in the topology optimisation formulation may significantly influence the design solution. In particular, it leads to stiffer designs, with respect to these loads, by judiciously placing holes or elements in the structure, and tailoring the shape of these elements to simultaneously reduce the impact of these loads and better oppose this impact. The compliance difference between the designs with and without considering design-dependent loads can reach up to 98 percent in structural stiffness. It was observed from these numerical examples that the convergence of the multigrid accelerated cellular automata algorithm is more stable than the MMALS algorithm described in chapter 4. Further research is needed to avoid local minimum solutions.

## Chapter 5

# Chapter 6

## Conclusions

### 6.1 Summary

Research into topology optimisation of continuum structures has become increasingly popular. In spite of this recorded progress, further research has to be done to find more powerful and more robust programs to solve topology optimisation problems in a reasonable amount of computational time. In this thesis, a multigrid accelerated cellular automata (CA) scheme was selected as the method to solve the problems of topology optimisation of continuum structures. The multigrid scheme and the cellular automata paradigm are closely related in their nature. Moreover, the design-dependent character of loads such as wind and hydrodynamic pressure was carefully incorporated into the design problem formulation, and the influence of these loads on the optimum topology is demonstrated.

The dissertation started with an introductory chapter consisting outlining the background and motivation for the work and providing a list of the different structural optimisation techniques. This was followed by a literature reviews of the following topics:

- topology optimisation of structures
- previous applications of cellular automata in structural design
- multigrid technique for topology optimisation of structures

- topology optimisation of structures subject to design-dependent loads such as hydrostatic pressure and transmissible loads

The cellular automata paradigm is a recent methodology that has been used in structural design [12, 16, 18–24]. It is a local technique that is well known for its parallelism. In chapter 2, a background and basic of cellular automata were presented, followed by an application of cellular automata to topology optimisation of continuum structures [20]. The cellular automata elements are lattice geometry, neighbourhood, cell states, local rules of transition and boundary conditions. The elastic continuum domain was discretised using a lattice of regular cells. The topology optimisation problem was regularised using the traditional SIMP approach [7–11]. The analysis rules were derived from the principle of minimum total potential energy, and the design rules were formulated based on continuous optimality criteria interpreted as local Kuhn-Tucker conditions.

The proposed cellular automata design algorithm was verified to be robust in solving topology optimisation problems. The drawback, however, is that the CA analysis convergence tends to be rather slow. This deterioration in convergence rate is due to the slow propagation of information across the domain as the grid is refined. Abdalla and Gürdal [20] suggest using a direct finite element analysis of the structure to accelerate the analysis process, followed by CA design rules. This scheme was originally proposed by Kita and Toyoda [12]. The present thesis, the cellular automata analysis convergence was accelerated using multigrid method for serial machines.

The acceleration of CA convergence using a multigrid approach was presented in chapter 3. It was observed that the convergence rate of cellular automata scheme deteriorated considerably as the number of variables increases. The iteration of the CA scheme acts with greatest efficiency on the propagation of short wavelength error components, while long wavelength error components are slowly damped. Using multigrid technique, long wavelength error components can be reduced on coarse grids without losing too much precision with a computational cost for one relaxation much smaller than that on the fine grid. The numerical examples given in chapter 3 demonstrate the computational cost for the multigrid algorithms to be proportional to the number of cells. Moreover, the multigrid accelerated CA algorithm was shown to be an interesting candidate for solving topology optimisation for two and three dimensional continuum structures in a reasonable time.

In chapter 4, the study of topology optimisation of continuum structures was extended to investigate the effect of design-dependent loads on the optimal design. The finite element method was selected to determine the displacement fields, while the optimisation problem was solved using the Method of Moving Asymptotes (MMA)

[46–49]. This algorithm was also used as a benchmark later in the 5<sup>th</sup> chapter given the robustness of the well known MMA method in solving topology optimisation problems. In chapter 4, wind loading was introduced into the topology optimisation formulation for two and three dimensional problems as design-dependent loading by using a standard formula for drag forces (Zakhama et al. [44]).

Furthermore, the MMA method was modified by changing the formula for the update of asymptotes. Another modification is made in the MMA based algorithm by adding line search phase to the optimisation process to enhance the global convergence of the topology optimisation problem. To obtain a black/white design, the intermediate density values were controlled using an explicit constraint added into the topology optimisation formulation. It was demonstrated that taking the wind loads into account in the formulation tends to generate stiffer designs, with respect to these loads for the same volume and can lead to the apparition holes facing the incoming wind.

Encouraged by the success of the modified MMA method coupled with the Line Search (MMALS) algorithm in solving effectively the topology optimisation problems with design-dependent loads, the multigrid accelerated cellular automata algorithm was extended in chapter 5 to incorporate design-dependent loads. The optimality criteria for minimum compliance with design-dependent loads subject to a volume constraint were used to derive local design rules. The local objective function was convexified to prevent oscillation in the topology optimisation process, and the Jacobi Over Relaxation scheme (JOR) [69, 70] was used instead of the Jacobi scheme to update the design variables and to ensure convergence of the optimisation process. The explicit constraint was appended to the optimisation formulation, as done in chapter 4, to obtain near black/white design. The proposed algorithm was demonstrated to be quite fast and more robust than the MMALS to solve the topology optimisation problems with design-dependent loads. The numerical examples presented in chapter 5 showed the effect of design-dependent loading, such as wind and hydrodynamic water pressure loading, on the topology. It was shown, from the designs obtained, that taking into account the design-dependent loads in the formulation leads to stiffer designs with respect to these loads by allowing holes facing the loads, creating extra elements where needed or adapting shape.

## 6.2 Future work

The multigrid accelerated cellular automata algorithm presented in this thesis have been developed and implemented to solve topology optimisation for two and three di-

mensional linearly elastic continua with design-dependent loads. However, the small deformation assumption is not always valid. Such is the case of compliant mechanisms [72]. To accommodate the algorithms to more realistic designs, geometric nonlinearities of continuum structures must be considered for continuum structures that undergo large deformation. Research work on the CA paradigm as a tool for solving topology optimisation of continuum structures with geometric nonlinearities is limited, e.g. the nonlinear topology design of trusses [16, 73] and nonlinear topology design of continuum structures [22].

The multigrid algorithm presented in this thesis may fail to converge when geometric nonlinearities are considered in topology optimisation of continuum structures. Aside from the work reported in this thesis, in our experimentation with solving topology optimisation of geometrically nonlinear 2D continua we have observed severe convergence difficulties. With the subtle handling of discretisation and approximation errors by the multigrid approach, it is likely that a multigrid implementation of topology optimisation of continua exhibiting geometric nonlinearities would be more effective. Nonetheless, the extension of the multigrid design algorithm to nonlinear structures requires several adaptations. For instance, the residual relation  $\mathbf{Ae} = \mathbf{r}$  used for the multigrid (see chapter 3) does not hold for nonlinear problems since the matrix  $\mathbf{A}$  becomes dependent on the response and  $\mathbf{Au} - \mathbf{Av} \neq \mathbf{Ae}$ . An alternative methodology based on Full Approximation Storage (FAS) multigrid scheme [28, 74] seems to be a good direction to take to solve a nonlinear problem. This scheme shares many features with the multigrid algorithm described in chapter 3, the main differences lie in the solution process on the coarsest grid and in the relaxation process.

In this research, the design-dependent loads have been introduced into the formulation using standard expressions for the drag force which do not involve explicit construction of the loading surfaces. This scheme has been demonstrated to be successful in accommodating the topology to the specific nature of these loads using both the MMALS and the multigrid accelerated cellular automata algorithms. In some cases, however, the optimisation process could not avoid local minima. Moreover, the MMALS algorithm sometimes encounters difficulties converging to a pure black/white design. Further research is certainly needed to enhance the capability of these algorithms to converge to global minima and to better control the explicit constraint parameters to obtain a pure black/white design.



# Appendix A

## Local Rule of Analysis

### A.1 Two-dimensional structures

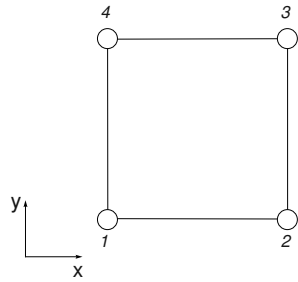


Figure A.1: Two-dimensional element.

For two-dimensional structures, the  $(18 \times 18)$  linear stiffness matrix  $\mathbf{K}$  is the assembly of element stiffness matrices of  $N_{element}$  elements of the Moore neighbourhood structure (see figure 2.4(d)):

$$\mathbf{K} = \sum_{i=1}^{N_{element}} \bar{\rho}_i^p \mathbf{K}^0, \quad (\text{A.1})$$

where  $\mathbf{K}^0$  is a stiffness matrix of the element represented in figure A.1. It is obtained from isotropic material and linear small deformation assumptions. The element

## Appendix A

stiffness matrix  $\mathbf{K}^0$  is defined as

$$\mathbf{K}^0 = \frac{E t}{1 - \nu^2} \left[ \begin{array}{c} \left( \begin{array}{cccccccc} \frac{1}{2} & \frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & 0 & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{4} \\ -\frac{1}{8} & \frac{1}{8} & \frac{1}{2} & -\frac{1}{8} & 0 & -\frac{1}{8} & -\frac{1}{8} & \frac{1}{4} \\ -\frac{1}{8} & 0 & -\frac{1}{8} & \frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{4} \\ -\frac{1}{8} & \frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & -\frac{1}{4} \\ 0 & -\frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & 0 \\ \frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & \frac{1}{2} \end{array} \right) + \\ \nu \left( \begin{array}{cccccccc} -\frac{1}{6} & \frac{1}{8} & -\frac{1}{12} & \frac{3}{8} & \frac{1}{12} & -\frac{1}{8} & \frac{1}{6} & -\frac{3}{8} \\ \frac{1}{8} & -\frac{1}{3} & -\frac{1}{8} & -\frac{1}{6} & -\frac{1}{8} & \frac{1}{12} & \frac{1}{8} & -\frac{1}{12} \\ -\frac{1}{12} & \frac{1}{3} & -\frac{1}{6} & -\frac{1}{8} & \frac{1}{6} & -\frac{1}{8} & -\frac{1}{12} & \frac{1}{8} \\ \frac{3}{8} & -\frac{1}{6} & -\frac{1}{8} & -\frac{1}{6} & -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} & -\frac{1}{12} \\ -\frac{1}{12} & \frac{1}{8} & \frac{1}{3} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{6} & -\frac{1}{12} & \frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{12} & -\frac{1}{8} & \frac{1}{12} & \frac{1}{8} & -\frac{1}{6} & -\frac{1}{12} & -\frac{1}{8} \\ \frac{1}{6} & \frac{1}{8} & -\frac{1}{12} & -\frac{1}{6} & -\frac{1}{12} & \frac{1}{8} & -\frac{1}{6} & \frac{3}{8} \\ -\frac{3}{8} & -\frac{1}{12} & \frac{1}{8} & \frac{1}{12} & \frac{1}{8} & -\frac{1}{12} & \frac{1}{6} & -\frac{1}{8} \end{array} \right) \right], \quad (\text{A.2})$$

where  $E$  is the Young's modulus,  $\nu$  is the Poisson's ratio and  $t$  is the plate thickness.

The cell displacements are updated using the Gauss-Seidel scheme as follows:

$$\mathbf{u}_C^{t+1} = \mathbf{u}_C^t + (\mathbf{K}_C)^{-1} \cdot (\mathbf{G}_C(\mathbf{u}_N^{t+1}) + \mathbf{f}_C), \quad (\text{A.3})$$

where  $\mathbf{K}_C$  is a  $(2 \times 2)$  matrix defined by the cell  $C$  indexes of the stiffness matrix  $\mathbf{K}$  which can be written as  $\mathbf{K}_{9,10 ; 9,10}$ . The vector  $\mathbf{G}_C(\mathbf{u}_N^{t+1})$  is the vector of internal forces defined as

$$\mathbf{G}_C(\mathbf{u}_N^{t+1}) = -\mathbf{K}_{N_1} \cdot \mathbf{u}_{N_1}^{t+1} - \mathbf{K}_{N_2} \cdot \mathbf{u}_{N_2}^{t+1}, \quad (\text{A.4})$$

where  $\mathbf{K}_{N_1}$  and  $\mathbf{K}_{N_2}$  are  $(2 \times 8)$  matrices defined by line 9 and 10 of the stiffness matrix  $\mathbf{K}$  as  $\mathbf{K}_{9,10 ; 1..8}$  and  $\mathbf{K}_{9,10 ; 11..18}$  respectively. The vectors  $\mathbf{u}_{N_1}^{t+1}$  and  $\mathbf{u}_{N_2}^{t+1}$  are their corresponding displacement vectors defined from  $\mathbf{u}_N^{t+1}$ .

## A.2 Three-dimensional structures

For the case of three-dimensional structures, the element stiffness matrix  $\mathbf{K}^0$  of figure A.2 is defined as

$$\mathbf{K}^0 = \begin{pmatrix} \mathbf{K}_{11}^0 & \mathbf{K}_{12}^0 & \mathbf{K}_{13}^0 \\ \mathbf{K}_{12}^{0T} & \mathbf{K}_{22}^0 & \mathbf{K}_{23}^0 \\ \mathbf{K}_{13}^{0T} & \mathbf{K}_{23}^{0T} & \mathbf{K}_{33}^0 \end{pmatrix}, \quad (\text{A.5})$$

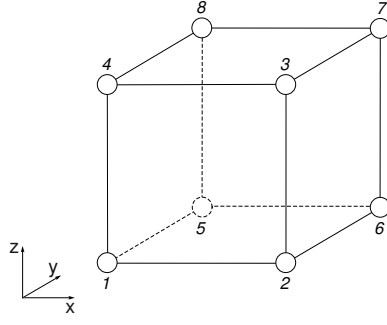


Figure A.2: Three-dimensional element.

where

$$\mathbf{K}_{11}^0 = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & \frac{c_5}{2} & c_6 & \frac{c_5}{2} \\ c_2 & c_1 & c_2 & -c_5 & -\frac{c_4}{2} & \frac{c_2}{2} & -\frac{c_5}{2} & c_7 \\ c_3 & c_2 & c_1 & -\frac{c_5}{2} & \frac{c_2}{2} & -\frac{c_4}{2} & -c_3 & \frac{c_5}{2} \\ c_4 & -c_5 & -\frac{c_5}{2} & c_1 & -c_2 & -c_3 & -\frac{c_4}{2} & -\frac{c_2}{2} \\ c_5 & -\frac{c_4}{2} & \frac{c_2}{2} & -c_2 & c_1 & c_2 & -\frac{c_2}{2} & -\frac{c_4}{2} \\ \frac{c_5}{2} & \frac{c_2}{2} & -\frac{c_4}{2} & -c_3 & c_2 & c_1 & -\frac{c_5}{2} & c_5 \\ c_6 & -\frac{c_5}{2} & -c_3 & -\frac{c_4}{2} & -\frac{c_2}{2} & -\frac{c_5}{2} & c_1 & -c_2 \\ \frac{c_5}{2} & c_7 & \frac{c_5}{2} & -\frac{c_2}{2} & -\frac{c_4}{2} & c_5 & -c_2 & c_1 \end{pmatrix}, \quad (\text{A.6})$$

$$\mathbf{K}_{22}^0 = \begin{pmatrix} c_1 & -\frac{c_5}{2} & -\frac{c_2}{2} & -\frac{c_4}{2} & -\frac{c_3}{2} & \frac{c_2}{2} & -\frac{c_1}{4} & \frac{c_5}{4} \\ -\frac{c_5}{2} & c_1 & c_2 & -c_3 & c_7 & -\frac{c_5}{2} & \frac{c_5}{4} & -\frac{c_1}{4} \\ -\frac{c_2}{2} & c_2 & c_1 & -c_2 & \frac{c_5}{2} & c_6 & c_2 & -\frac{c_2}{2} \\ -\frac{c_4}{2} & -c_3 & -c_2 & c_1 & -\frac{c_5}{4} & c_2 & c_6 & \frac{c_3}{2} \\ -\frac{c_3}{2} & c_7 & \frac{c_5}{2} & -\frac{c_5}{4} & c_1 & -c_2 & c_3 & c_4 \\ \frac{c_2}{2} & -\frac{c_5}{2} & c_6 & c_2 & -c_2 & c_1 & -c_2 & c_5 \\ -\frac{c_1}{4} & \frac{c_5}{4} & c_2 & c_6 & c_3 & -c_2 & c_1 & -\frac{c_5}{2} \\ \frac{c_5}{4} & -\frac{c_1}{4} & -\frac{c_2}{2} & \frac{c_3}{2} & c_4 & c_5 & -\frac{c_5}{2} & c_1 \end{pmatrix}, \quad (\text{A.7})$$

$$\mathbf{K}_{33}^0 = \begin{pmatrix} c_1 & -c_2 & \frac{c_2}{2} & -\frac{c_4}{2} & c_5 & -\frac{c_5}{2} & c_7 & \frac{c_5}{2} \\ -c_2 & c_1 & -\frac{c_5}{2} & -c_5 & c_4 & c_3 & -\frac{c_5}{2} & c_6 \\ \frac{c_2}{2} & -\frac{c_5}{2} & c_1 & c_2 & c_3 & c_4 & c_5 & \frac{c_5}{2} \\ -\frac{c_4}{2} & -c_5 & c_2 & c_1 & c_2 & -c_5 & -\frac{c_4}{2} & \frac{c_2}{2} \\ c_5 & c_4 & c_3 & c_2 & c_1 & -\frac{c_5}{2} & \frac{c_2}{2} & -\frac{c_4}{2} \\ -\frac{c_5}{2} & c_3 & c_4 & -c_5 & -\frac{c_5}{2} & c_1 & -c_2 & -c_3 \\ c_7 & -\frac{c_5}{2} & c_5 & -\frac{c_4}{2} & \frac{c_2}{2} & -c_2 & c_1 & c_2 \\ \frac{c_5}{2} & c_6 & \frac{c_5}{2} & \frac{c_2}{2} & -\frac{c_4}{2} & -c_3 & c_2 & c_1 \end{pmatrix}, \quad (\text{A.8})$$

Appendix A

$$\mathbf{K}_{12}^0 = \begin{pmatrix} -c_3 & -\frac{c_4}{2} & \frac{c_2}{2} & -\frac{c_5}{2} & -\frac{c_4}{2} & -c_5 & \frac{c_3}{2} & c_6 \\ -\frac{c_5}{2} & \frac{c_2}{2} & -\frac{c_4}{2} & -c_5 & c_5 & c_4 & c_5 & -c_2 \\ c_6 & \frac{c_5}{2} & c_5 & c_4 & \frac{c_3}{2} & -c_5 & -\frac{c_4}{2} & -\frac{c_5}{4} \\ \frac{c_5}{2} & c_6 & -\frac{c_5}{2} & c_3 & c_6 & c_2 & -\frac{c_5}{4} & -\frac{c_4}{2} \\ -c_5 & \frac{c_5}{2} & c_7 & -\frac{c_5}{2} & c_2 & c_6 & \frac{c_5}{2} & -c_5 \\ c_4 & c_3 & \frac{c_5}{2} & c_6 & \frac{c_5}{4} & -\frac{c_5}{2} & c_7 & -\frac{c_3}{2} \\ c_3 & c_4 & -c_5 & \frac{c_5}{2} & -\frac{c_1}{4} & \frac{c_2}{2} & -\frac{c_3}{2} & c_7 \\ -c_2 & c_5 & -\frac{c_4}{2} & -\frac{c_2}{2} & \frac{c_2}{2} & -\frac{c_1}{4} & \frac{c_2}{2} & -\frac{c_5}{2} \end{pmatrix}, \quad (\text{A.9})$$

$$\mathbf{K}_{13}^0 = \begin{pmatrix} -c_2 & \frac{c_5}{4} & -\frac{c_1}{4} & -\frac{c_2}{2} & -\frac{c_3}{2} & c_7 & -\frac{c_5}{2} & -\frac{c_5}{4} \\ c_6 & \frac{c_5}{2} & -\frac{c_2}{2} & -\frac{c_1}{4} & -\frac{c_2}{2} & \frac{c_5}{2} & c_6 & -c_2 \\ -\frac{c_5}{2} & c_7 & -\frac{c_3}{2} & -\frac{c_2}{2} & -\frac{c_1}{4} & \frac{c_5}{4} & -c_2 & c_6 \\ c_5 & -\frac{c_3}{2} & c_7 & \frac{c_5}{2} & \frac{c_5}{4} & -\frac{c_1}{4} & \frac{c_2}{2} & \frac{c_3}{2} \\ c_4 & c_5 & -\frac{c_5}{2} & c_6 & -c_2 & \frac{c_2}{2} & -\frac{c_1}{4} & -\frac{c_2}{2} \\ -c_5 & -\frac{c_4}{2} & -\frac{c_5}{4} & -c_2 & c_6 & \frac{c_3}{2} & -\frac{c_2}{2} & -\frac{c_1}{4} \\ \frac{c_5}{2} & -\frac{c_5}{4} & -\frac{c_4}{2} & c_5 & \frac{c_3}{2} & c_6 & c_2 & \frac{c_5}{4} \\ c_6 & c_2 & -c_5 & c_4 & -c_5 & c_2 & c_6 & -\frac{c_5}{2} \end{pmatrix}, \quad (\text{A.10})$$

$$\mathbf{K}_{23}^0 = \begin{pmatrix} c_2 & c_6 & \frac{c_3}{2} & c_5 & -\frac{c_4}{2} & -\frac{c_5}{4} & \frac{c_5}{2} & c_7 \\ -\frac{c_2}{2} & \frac{c_3}{2} & c_6 & -c_2 & -\frac{c_5}{4} & -\frac{c_4}{2} & -c_5 & -\frac{c_3}{2} \\ -\frac{c_1}{4} & \frac{c_2}{2} & -c_2 & c_6 & -\frac{c_5}{2} & c_5 & c_4 & -c_5 \\ \frac{c_2}{2} & -\frac{c_1}{4} & \frac{c_5}{4} & \frac{c_5}{2} & c_7 & -\frac{c_3}{2} & c_5 & -\frac{c_4}{2} \\ -c_5 & \frac{c_5}{2} & c_6 & -\frac{c_5}{2} & -c_3 & -\frac{c_4}{2} & -\frac{c_2}{2} & -\frac{c_5}{2} \\ -\frac{c_4}{2} & -\frac{c_2}{2} & \frac{c_5}{2} & c_7 & \frac{c_5}{2} & -\frac{c_4}{2} & -\frac{c_4}{2} & c_5 \\ -\frac{c_2}{2} & -\frac{c_4}{2} & -c_3 & -\frac{c_5}{2} & c_6 & \frac{c_5}{2} & -c_5 & c_4 \\ c_2 & -c_3 & -\frac{c_4}{2} & \frac{c_2}{2} & \frac{c_2}{2} & c_6 & \frac{c_5}{2} & c_3 \end{pmatrix}, \quad (\text{A.11})$$

where

$$\begin{aligned} c_1 &= \left(-\frac{\nu}{3} + \frac{2}{9}\right)b, & c_2 &= \frac{b}{24}, & c_3 &= \left(-\frac{\nu}{12} + \frac{1}{16}\right)b, \\ c_4 &= -\frac{b}{18}, & c_5 &= \left(\frac{\nu}{6} - \frac{1}{24}\right)b, & c_6 &= \left(\frac{\nu}{12} - \frac{5}{72}\right)b, \\ c_7 &= \left(\frac{\nu}{12} - \frac{1}{36}\right)b, & b &= \frac{E a}{(1 + \nu)(1 - 2\nu)}, \end{aligned} \quad (\text{A.12})$$

and  $a$  is the distance between the cells.

The analysis update rule remains the same as in two-dimensional case. However, the linear stiffness matrix  $\mathbf{K}$  of the Moore neighbourhood structure defined in figure 2.4(e) is a  $(81 \times 81)$  matrix. The matrix  $\mathbf{K}_C$  has dimensions  $(3 \times 3)$  ( $\mathbf{K}_C = \mathbf{K}_{40..42 ; 40..42}$ ), and  $\mathbf{K}_{N_1}$  and  $\mathbf{K}_{N_2}$  are  $(3 \times 40)$  matrices ( $\mathbf{K}_{N_1} = \mathbf{K}_{40..42 ; 1..40}$  and  $\mathbf{K}_{N_2} = \mathbf{K}_{40..42 ; 42..81}$ ).

# Bibliography

- [1] L. A. Schmit. Structural design by systematic synthesis. In *Proceedings of the 2nd Conference on Electronic Computation, ASCE, New York*, pages 105–132, 1960.
- [2] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.
- [3] K. T. Cheng and N. Olhoff. An investigation concerning optimal design of solid elastic plates. *International Journal of Solids and Structures*, 17:305–323, 1981.
- [4] K. Suzuki and N. Kikuchi. A homogenization method for shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 93:291–318, 1991.
- [5] A. Díaz and N. Kikuchi. Solutions to shape and topology eigenvalue optimization problems using a homogenization method. *International Journal for Numerical Methods in Engineering*, 53:1487–1502, 1992.
- [6] G. Allaire, E. Bonnetier, G. Francfort, and F. Jouve. Shape optimization by the homogenization method. *Numerische Mathematik*, 76:28–68, 1997.
- [7] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural and Multidisciplinary Optimization*, 1:193–200, 1989.
- [8] M. Zhou and G. I. N. Rozvany. The COC algorithm, part II: Topological, geometry and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89:197–224, 1991.
- [9] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural and Multidisciplinary Optimization*, 4:250–252, 1992.

- [10] G. I. N. Rozvany. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, 21:90–108, 2001.
- [11] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.
- [12] E. Kita and T. Toyoda. Structural design using cellular automata. *Structural and Multidisciplinary Optimization*, 19:64–73, 2000.
- [13] Y. M. Xie and G. P. Steven. A simple evolutionary procedure for structural optimization. *Computers and Structures*, 49:885–896, 1993.
- [14] Y. M. Xie and G. P. Steven. Optimal design of multiple load case structures using an evolutionary procedure. *Engineering Computations*, 11:295–302, 1994.
- [15] C. Zhao, G. P. Steven, and Y. M. Xie. Effect of initial non-design domain on optimal topologies of structures during natural frequency optimization. *Computers and Structures*, 62:119–131, 1997.
- [16] Z. Gürdal and B. Tatting. Cellular automata for design of truss structures with linear and non linear response. In *41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Atlanta, GA*, 2000.
- [17] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*. Kluwer, 1993.
- [18] B. Tatting and Z. Gürdal. Cellular automata for design of two-dimensional continuum structures. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach*, 2003.
- [19] M. M. Abdalla and Z. Gürdal. Structural design using cellular automata for eigenvalue problems. *Structural and Multidisciplinary Optimization*, 26:200–208, 2004.
- [20] M. M. Abdalla and Z. Gürdal. Structural design using optimality based cellular automata. In *43th AIAA/ASME/AHS/ASC Structures, Structural Dynamics and material Conference, Denver, Co*, April 2002.
- [21] S. Setoodeh, M. M. Abdalla, and Z. Gürdal. Combined topology and fiber path design of composite layers using cellular automata. *Structural and Multidisciplinary Optimization*, 30:413–421, 2005.
- [22] R. Zakhama, M. M. Abdalla, H. Smaoui, and Z. Gürdal. Topology design of geometrically nonlinear 2D elastic continua using CA and an equivalent truss model. In *11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2006.

- [23] D. J. Slotta, B. Tatting, L. T. Watson, Z. Gürdal, and S. Missoum. Convergence analysis for cellular automata applied to truss design. *Engineering Computations*, 19:953–969, 2002.
- [24] S. Setoodeh, D. B. Adams, Z. Gürdal, and L. T. Watson. Pipeline implementation of cellular automata for structural design on message-passing multiprocessors. *Mathematical and Computer Modelling*, 43:966–975, 2006.
- [25] S. Kim, M. M. Abdalla, Z. Gürdal, and M. Jones. Multigrid accelerated cellular automata for structural design optimization: A 1-D implementation. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Palm Springs, California*, 2004.
- [26] R. Zakhama, M. M. Abdalla, H. Smaoui, and Z. Gürdal. Multigrid implementation of cellular automata for topology optimization. In *First International Conference on Multidisciplinary Design and Applications, Besançon, France*, 2007.
- [27] R. Zakhama, M. M. Abdalla, H. Smaoui, and Z. Gürdal. Multigrid implementation of cellular automata for topology optimization of continuum structures. *Computer Modeling in Engineering and Sciences*. Manuscript accepted.
- [28] P. Wesseling. *An introduction to multigrid methods*. Wiley, 1992.
- [29] W. Hackbush. *Multigrid methods and applications*. Springer, 1985.
- [30] B. Maar and V. Schulz. Interior point multigrid methods for topology optimization. *Mathematical and Computer Modelling*, 19:214–224, 2000.
- [31] Y. Y. Kim and G. H. Yoon. Multi-resolution multi-scale topology optimization - a new paradigm. *International Journal of Solids and Structures*, 37:5529–5559, 2000.
- [32] T. Dreyer, B. Maar, and V. Schulz. Multigrid optimization in application. *Journal of Computational and Applied Mathematics*, 120:67–84, 2000.
- [33] U.C. Garcia-Palomares and O. L. Mangasarian. Superlinearly convergent quasi-newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming*, 11:1–13, 1976.
- [34] S. P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.
- [35] J. F. Bonnans, E.R. Panier, A.L. Tits, and J. L. Zhou. Avoiding the maratos effect by means of a nonmonotone line search. II. Inequality constrained problems-feasible iterates. *Journal on Numerical Analysis*, 29:1187–1202, 1992.

- [36] K. Kwon, J. E. Kim, G. W. Jang, and Y. Y. Kim. Multigrid approach for efficient integrated multiscale analysis and design optimization. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002.
- [37] V. B. Hammer and N. Olhoff. Topology optimization of continuum structures subjected to pressure loading. *Structural and Multidisciplinary Optimization*, 19:85–92, 2000.
- [38] M. B. Fuchs and E. Moses. Optimal structural topologies with transmissible loads. *Structural and Multidisciplinary Optimization*, 19:263–273, 2000.
- [39] J. Du and N. Olhoff. Topological optimization of continuum structures with design-dependent surface loading - part I: new computational approach for 2D problems. *Structural and Multidisciplinary Optimization*, 27:151–165, 2004.
- [40] J. Du and N. Olhoff. Topological optimization of continuum structures with design-dependent surface loading - part II: algorithm and examples for 3D problems. *Structural and Multidisciplinary Optimization*, 27:166–177, 2004.
- [41] M. B. Fuchs and N. N. Y. Shemesh. Density-based topological design of structures subjected to water pressure using a parametric loading surface. *Structural and Multidisciplinary Optimization*, 28:11–19, 2004.
- [42] B. C. Chen and N. Kikuchi. Topology optimization with design-dependent loads. *Finite Elements in Analysis and Design*, 37:57–70, 2001.
- [43] O. Sigmund and P. M. Clausen. Topology optimization using a mixed formulation: An alternative way to solve pressure load problems. *Computer Methods in Applied Mechanics and Engineering*, 196:1874–1889, 2007.
- [44] R. Zakhama, M. M. Abdalla, Z. Gürdal, and H. Smaoui. Wind load effect in topology optimization problems. *Journal of Physics: Conference Series*, 75, 2007.
- [45] R. Zakhama, M. M. Abdalla, Z. Gürdal, and H. Smaoui. Wind load modeling for topology optimization of continuum structures. *Structural and Multidisciplinary Optimization*. Manuscript accepted.
- [46] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.
- [47] C. Zillober. A globally convergent version of the method of moving asymptotes. *Structural Optimization*, 6:166–174, 1993.



- [48] K. U. Bletzinger. Extended method of moving asymptotes based on second-order information. *Structural Optimization*, 5:175–183, 1993.
- [49] M. Bruyneel, P. Duysinx, and C. Fleury. A family of MMA approximations for structural optimization. *Structural and Multidisciplinary Optimization*, 24:263–276, 2002.
- [50] Y. C. Lam, D. Manickarajah, and A. Bertolini. Performance characteristics of resizing algorithms for thickness optimization of plate structures. *Finite Elements in Analysis and Design*, 34:159–174, 2000.
- [51] L. Yin and W. Yang. Optimality criteria method for topology optimization under multiple constraints. *Computers and Structures*, 79:1839–1850, 2001.
- [52] M. Zhou and G. I. N. Rozvany. An optimality criteria method for large systems. Part I: theory. *Structural Optimization*, 5:12–25, 1992.
- [53] M. Zhou and G. I. N. Rozvany. An optimality criteria method for large systems. Part II: Algorithm. *Structural Optimization*, 6:250–262, 1996.
- [54] S. Ulam. Random processes and transformations. In *Proceedings of the International Congress of Mathematics*, volume 2, pages 85–87, April 1952.
- [55] J. V. Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.
- [56] T. Toffoli and N. Margolus. *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, Cambridge, MA, 1987.
- [57] S. Wolfram. *Cellular Automata and Complexity: Collected Papers*. Addison-Wesley Publishing Company, 1994.
- [58] S. Wolfram. *A New Kind of Science*. Wolfram Media, 2002.
- [59] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.
- [60] M. Beekers. Topology optimization using a dual method with discrete variables. *Structural Optimization*, 17:14–24, 1999.
- [61] M. Bruyneel and P. Duysinx. Note on topology optimization of continuum structures including self-weight. *Structural and Multidisciplinary Optimization*, 29:245–256, 2005.
- [62] T. Buhl, C. B. W. Pedersen, and O. Sigmund. Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19:93–104, 2000.

- [63] T. E. Bruns and D. A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190:3443–3459, 2001.
- [64] L. Yin and G. K. Ananthasuresh. Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. *Structural and Multidisciplinary Optimization*, 23:49–62, 2001.
- [65] C. S. Jog and R. B. Haber. Stability of finite element models for distributed-parameter optimization and topology design. *Computer Methods in Applied Mechanics and Engineering*, 130:203–226, 1996.
- [66] T. Borrvall and J. Petersson. Topology optimization using regularized intermediate density control. *Computer Methods in Applied Mechanics and Engineering*, 190:4911–4928, 2001.
- [67] G. K. Lau, H. Du, and M. K. Lim. Techniques to suppress intermediate density in topology optimization of compliant. *Computational Mechanics*, 27:426–435, 2001.
- [68] M. Y. Wang and S. Wang. Bilateral filtering for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 63:1911–1938, 2005.
- [69] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, 2000.
- [70] J. J. M. M. Rutten, P. Panangaden, and F. V. Breugel. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*. AMS Bookstore, 2004.
- [71] R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, 1973.
- [72] T. E. Bruns and D. A. Tortorelli. Topology optimization of nonlinear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190:3443–3459, 2001.
- [73] S. Missoum, M. M. Abdalla, and Z. Gürdal. Nonlinear topology design of trusses using cellular automata. In *44th AIAA/ASME/AHS/ASC Symposium on Structural Dynamics and Material Conference, Norfolk, Va*, 2003.
- [74] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.

## **Curriculum Vitae**

Ramzi Zakhama was born on June 1976 in Verviers, Belgium. He obtained his bachelor of civil engineering degree from Ecole Nationale d'ingénieur de Gabès, Tunisia, in 2001. After graduating, he started in 2003 his Masters degree program at Ecole Nationale d'ingénieurs de Tunis, Tunisia. In 2005, he started his PhD degree program in Aerospace structures group at the faculty of Aerospace Engineering of Delft university of Technology under the supervision of Professor Zafer Gürdal and Professor Hichem Smaoui.