

Applications of the Cellular Automata Paradigm in Structural Analysis and Design

Applications of the Cellular Automata Paradigm in Structural Analysis and Design

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 13 December 2004 te 3.30 uur

door

Mostafa M. ABDALLA

M. Sc. in Aerospace Engineering, Faculty of Engineering, Cairo University

geboren te Cairo, Egypte

Dit proefschrift is goedgekeurd de promotor:
Prof. dr. Z. Gürdal

Smenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. Z. Gürdal,	Technische Universiteit Delft, Promotor
Prof. dr. ir. F. van Keulen,	Technische Universiteit Delft
Prof. dr. A. Rothwell,	Technische Universiteit Delft
Prof. dr. P.M.A Sloot,	Universiteit van Amsterdam
Prof. dr. ir. P. Wesseling,	Technische Universiteit Delft

Published and distributed by: DUP Science

DUP Science is an imprint of Delft University Press P.O. Box 98 2600 MG Delft
The Netherlands Telephone: +31 15 27 85 678 Telefax: +31 15 27 85 706 Email:
info@library.tudelft.nl

ISBN 90-407-2565-9

Keywords: Cellular Automata/ Structural Optimization/ Topology Design

Copyright ©2004 by M.M. Abdalla

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: Delft University Press

Printed in the Netherlands

*To my mother
In Memoriam*

Acknowledgements

I wish to thank my academic advisor Prof.dr. Zafer Gürdal first for giving me the wonderful opportunity of working with him. The amount of freedom he offered me in conducting the research reported herein, his unfailing support, and insightful advice can be hardly overestimated. I wish also to extend my thanks to my committee members, Prof. dr. ir. F. van Keulen, Prof. dr. A. Rothwell, Prof.dr. P. Slood, and Prof. dr. ir. P. Wesseling for accepting to serve in my exam committee.

I also take this chance to thank a number of my close friends and colleagues for their support and for the many illuminating discussions we had. Namely, I wish to thank Mr. O. Seresta for the amusing times we spent sharing one office as graduate students. Special thanks goes to my friends dr. W.F. Faris and Mr. C.K. Reddy. Our collaboration in the MEMS microbeam design work was quite fruitful. It has been quite a pleasure interacting with them.

Summary

A recently emerging approach based on the use of the cellular automata (CA) paradigm is aimed at addressing the automated combined analysis and design of one-, two-, and three-dimensional elastic systems within a massively parallel computational environment. Implementation of the methods of cellular automata to the design of structural systems is, in some sense, an attempt for a simultaneous solution of the state and design variables that appear in the highly nonlinear governing differential equations and associated optimality conditions. The main advantages of CA for structural optimizations are 1) its potential for massively parallel implementation on advanced hardware, 2) both the analysis and design are treated locally, and 3) it lends itself readily to optimality based approaches where the solution of the field problem and the design problem are arrived at simultaneously.

An algorithm for designing structures for eigenvalue requirements was presented. Conceptually, this was an important area of investigation because the local nature of CA algorithms would, at least apparently, be challenged by the global nature of the eigenvalue response. The proposed algorithm was designed to be fully local in nature, and thus suitable for CA type implementation. The algorithm was applied to the design of Euler-Bernoulli columns against buckling. The analysis rule used to predict displacements was, for the first time, derived using energy principles. The CA combined analysis and design algorithm proved effective in accurately predicting optimal column shapes and the corresponding buckling mode shapes.

CA topology design algorithm was presented where the design rules for minimum compliance design of two-dimensional linearly elastic continuum topology were derived using variational calculus. The CA design rule was obtained based on the continuous optimality criteria interpreted as local Kuhn-Tucker conditions. As such, the design rule at each cell involved the solution of a simple one-dimensional optimization problem. The CA analysis rule was derived, similarly to the eigenvalue design work, based on energy minimization. Numerical experiments with the proposed algorithm indicated that the CA design rule is quite robust and does not suffer from checkerboard patterns, mesh-dependent topologies, or numerical instabilities. Given the simplicity of the CA algorithm, it became clear that CA is a good candidate as a topology design tool.

Topology optimization to minimize the compliance of a structure required a detailed

CA lattice to capture topological features. In this respect, it was found that the CA analysis rule converged rather slowly. The deterioration of CA convergence rate with lattice refinement was to be expected given that CA relied completely on local exchange of information. In a hypothetical ideal CA computing machine, where all cells are updated in parallel at a high rate, convergence rate deterioration would pose no particular problems. The efficiency of CA would still be considerable due to the simplicity of the processing elements. On the other hand, such deterioration in convergence rate would be a significant limiting factor when CA is used on existing serial processors.

Encouraged by the success of optimality-based CA design rules in topology optimization, CA design rules for nonlinear problems showing limit point behavior were next derived using rigorous optimality conditions. The design rule was successfully formulated as a local cell-level optimization problem. The CA design rule was coupled to nonlinear finite element analysis to solve the problem of shape design of a MEMS microbeam. The convergence of the CA design rule was quite fast requiring only twenty to thirty nonlinear finite element analyses. The results confirmed that local design rules based on optimality perform satisfactorily and could be indeed considered a general method for deriving CA design rules.

The structural response of MEMS microbeam is nonlinear due to the nonlinear dependence of electrostatic load on the microbeam deflection, and exhibits limit point (pull-in) behavior. We considered optimizing the shape of a capacitive micro-beam for maximum pull-in voltage. Extensive results for different beam boundary conditions were generated. The optimization results indicated that substantial increase in pull-in voltage can be achieved by varying the width and/or thickness distribution.

The implementation of CA as combined analysis and design tool where both CA analysis rules and CA design rules are applied to obtain a final converged design *together* with the corresponding displacements can be considered to be established at the algorithmic level. A combination of energy minimization for the derivation of the analysis rule and optimality for the design rule is generally applicable for a wide range of structural problems. The main challenges lie mainly in devising suitable hardware and software implementations where the CA computational advantage stemming from massive parallelism would be clearly demonstrated.

Samenvatting

Recent is een techniek ontstaan voor de aanpak van gecombineerde automatische analyse en ontwerp van één- of meerdimensionale elastische systemen in een omgeving van meervoudig parallel rekenen. De aanpak is gebaseerd op het paradigma van "cellular automata (CA)". De implementatie van deze techniek voor het ontwerpen van constructies is in zekere mate een poging tot het simultaan bepalen van de toestands- en ontwerpvariabelen via sterk niet-lineaire differentiaalvergelijkingen die het system beschrijven, alsmede van de daarbij behorende optimaliteitsvoorwaarden. De grootste voordelen van CA voor constructief optimaliseren zijn 1) zijn potentie voor implementatie van massief parallel gebruikte geavanceerde hardware, 2) de lokale aanpak van zowel de analyse als het ontwerp, en 3) zijn vanzelfsprekende geschiktheid voor een simultane aanpak via optimaliseringstechnieken van zowel het veldprobleem als het ontwerp.

Als onderdeel van het onderzoeksproject werd een algoritme voor door eigenwaardeberekening bepaalde aspecten in het ontwerpen van constructies gepresenteerd. Voor het begrip was dit een belangrijk onderzoeksgebied, omdat de lokale aard van CA algoritmen op zijn minst schijnbaar wordt beïnvloed door de globale aard van de eigenwaarde response. Het voorgestelde algoritme werd zo ontworpen dat het volledig lokaal was van aard en dus bruikbaar voor CA type implementatie. Het algoritme werd toegepast op het ontwerp van kolommen met weerstand tegen Euler-Bernouilli knik. De analyse beschrijving die werd gebruikt voor een eerste schatting van de verplaatsing, werd bepaald met energie-principes. De CA analyse gecombineerd met ontwerp-algoritmen heeft bewezen een efficiënte en nauwkeurige voorspelling te geven van optimale kolomvormen en de bijbehorende knikpatronen.

Vervolgens is een op CA gebaseerd topologisch ontwerp-algoritme gepresenteerd, waarbij de ontwerp-regels voor een qua stijfheid optimaal ontwerp van een tweedimensionale lineair elastische continuüm topologie, is ontwikkeld met gebruik van variatierekening. De CA ontwerp-regel is gebaseerd op de continue optimaliserings criteria, geïnterpreteerd als lokale Kuhn-Tucker condities. De CA analyse regel was, zoals bij het eigenwaarde-ontwerp probleem, gebaseerd op minimalisering van de energie. Numerieke experimenten met het voorgestelde algoritme gaven een aanwijzing dat de CA ontwerp regel zeer robuust is en niet leidt aan "schaakbord"-achtige patronen in de materiaal verdeling, noch aan numerieke instabiliteit. Met het gegeven van de eenvoud van het CA algoritme werd duidelijk, dat CA een

veelbelovende methodiek is voor topologisch optimaal ontwerpen.

Topologische optimalisering met betrekking tot minimum vervormbaarheid van een constructie vereist een gedetailleerd CA raster om de topologische kenmerken voldoende te beschrijven. In dit verband werd geconstateerd dat de CA analyse slechts langzaam convergeerde. De met toenemende raster-dichtheid slechter wordende CA convergentie kon worden verwacht op grond van het feit, dat CA geheel vertrouwt op informatie-uitwisseling op lokaal niveau. In een hypothetisch ideale CA machine waar alle cellen gelijktijdig snel worden aangepast zou dit geen probleem zijn. Dan zou de efficiency van CA nog altijd aanzienlijk zijn door de eenvoud van de elementen. Aan de andere kant zou een dergelijke afnemende convergentiegraad een significante beperkende factor betekenen, indien gebruik wordt gemaakt van de bestaande seriële processoren.

Na het succes met CA voor ontwerp regels in topologisch optimalisering, werd besloten zulke regels voor niet-lineaire problemen met "limit point" gedrag en gebruik van strikte optimaliteits condities, af te leiden. De ontwerpregel werd met succes geformuleerd als een probleem op lokaal, cel niveau. De ontwerpregel werd gekoppeld aan niet-lineaire Eindige Elementen Analyse voor het bepalen van de optimale vorm van een MEMS balkje op micro-schaal. De convergentie van de CA ontwerpregel bleek zeer snel en had slechts tussen de 20 en 30 EE analyses nodig. De resultaten bevestigden dat lokale regels voor een optimaal ontwerp behoorlijk presteren en inderdaad kunnen worden beschouwd als een algemene methode om CA-ontwerp regels af te leiden. De mechanische response van een MEMS micro-balkje is niet-lineair van aard vanwege de niet-lineaire relatie tussen elektrostatische belasting en de doorbuiging van het balkje. Er is sprake van limit-point (pull-in) gedrag. We beschouwden de optimalisering van de vorm van een capacitieve micro-balk voor maximaal intrek voltage. Uitvoerige resultaten werden gegenereerd voor verschillende randvoorwaarden. De optimaliseringresultaten gaven aan, dat een substantiele vergroting van het intrek-voltage kon worden bereikt door de breedte- en dikteverdeling te variëren.

De implementatie van CA als een gecombineerd analyse- en ontwerpgereedschap, waarin zowel CA analyse regels als CA ontwerp regels worden toegepast, kan worden beschouwd als te zijn gebouwd op algorithmes. De methodiek leidt uiteindelijk tot het verkrijgen van een geconvergeerd optimaal ontwerp. Een combinatie van minimalisering van de energie voor de afleiding van de analyse regels en de eis van

optimaliteit voor het bepalen van de ontwerp regels, is algemeen toepasbaar op een breed gebied van constructieve problemen. De grootste uitdagingen liggen in het bedenken van bruikbare hardware- en software-implementaties waar de typische rekenkundige voordelen van CA, voortkomend uit het massief parallelle karakter, duidelijk kunnen worden aangetoond.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Introduction to Cellular Automata	4
1.3	CA on Configurable Computers	8
1.4	Literature Review	10
1.5	Scope and Objectives	16
1.6	Organization of the Dissertation	18
2	Cellular Automata Algorithm for Eigenvalue Problems	21
2.1	Introduction	21
2.2	Eigenvalue Requirement Design Algorithm	23
2.3	Buckling Design of Columns	26
2.4	Numerical Examples	34
2.5	Conclusion	40

- 3 Topology Design Using Optimality Based Cellular Automata 43**
 - 3.1 Introduction 43
 - 3.2 Formulation of Minimum Compliance Design 47
 - 3.3 Topology Design 54
 - 3.4 CA Implementation 56
 - 3.5 Results 63
 - 3.6 Conclusion 72

- 4 Optimal Design of an Electrostatically Actuated MicroBeam for Maximum Pull-in Voltage 75**
 - 4.1 Introduction 75
 - 4.2 MEMS Optimization: an overview 77
 - 4.3 Micro-Beam Model 79
 - 4.4 Optimization Problem 81
 - 4.5 Optimality Criterion 83
 - 4.6 Results 86
 - 4.7 Conclusions 95

- 5 Conclusion 97**
 - 5.1 Summary 97
 - 5.2 Overview of the state of Cellular Automata research 100

5.3	Prospects of future research	102
A	Nonlinear Pull-in Point Prediction for Electrostatically Actuated microbeam	105
A.1	Finite Element Model	105
A.2	Nonlinear Response	107

Chapter 1

Introduction

1.1 Motivation

The rapid advancement of computing hardware gives a strong incentive to a more automated approach to design. Optimization of structural systems is a proven methodology for automating the structural design process [1, 2, 3]. In structural optimization, the design process is formulated as a mathematical optimization problem where a certain objective is to be maximized or minimized subject to constraints that represent available resources, limitations of materials, stability, or manufacturing technology. In order to maximize the applicability of structural optimization techniques, especially to large and multi-physics problems, new algorithms capable of exploiting advanced computing hardware potential are required.

Two basic approaches have been the standard in structural optimization: mathematical programming, and optimality criteria. In the optimality criteria approach, the problem can be either a continuous (infinite-dimensional) sizing problem, where the solution of the problem consists of finding the functional dependence of structural properties to minimize an integral cost functional, or a finite-dimensional problem, such as truss design, where the objective is to find a finite set of design variables to minimize a give cost function. In the finite dimensional case, constraints on the global behavior of the structure and/or constraints on the individual design vari-

Chapter 1

ables can be imposed. In the continuous case, integral constraints and/or pointwise constraints are imposed. In either case, the solution proceeds by writing the necessary optimality criteria and solving these equations for the optimal distribution of *both* the design and the corresponding field variables. In the continuous case, the necessary conditions are obtained through variational calculus.

In the mathematical programming approach, the problem is assumed to consist of finding a finite set of design variables to minimize a given function, and the solution usually proceeds by determining a search direction, searching along that direction, updating the direction, and repeating the process to convergence. Continuous problems are handled by first discretizing the problem, typically using finite elements, and linking the design variables to element properties.

The mathematical programming approach leads to a dichotomy of the structural optimization process into a separate response evaluation (analysis), and search (optimization) processes. This has been the approach mostly used in structural optimization applications. On the other hand, the optimality criteria method attacks the problem in a more integral manner by trying to find the optimal design together with the corresponding solution of the underlying field problem [4].

The prevalence of mathematical programming methods in structural optimization is largely due to the above-mentioned dichotomy. In this approach, the evaluation of the structural response, which either yields the objective function values or constraint values or both, can be accomplished using any type of analysis: analytic, semi-analytic, or numerical. The type of the numerical method used for response evaluation is in itself immaterial. Available methods for solving mathematical programming problems include gradient based search methods that require the derivatives of the responses with respect to the design variables, and non gradient based methods usually termed zero order methods. For smooth problems, gradient-based methods are generally believed to be superior in performance to zero order methods. As such, analysis techniques capable of generating gradient (also called sensitivity) information at minimum cost is usually preferred to analysis techniques that do not provide sensitivity information readily.

Since the engineering design process (whether using optimization or not) necessarily needs analysis tools, the development of software for the analysis of engineering structures has been, and continues to be, an active field of research. Consequently,

powerful analysis tools are readily available for use in optimization. The use of these powerful tools only required, at most, the introduction of sensitivity analysis that provides the essential ingredient for accurate and reliable search direction computation. Sensitivity analysis has been incorporated into many existing commercial finite element based analysis software. When semi-analytic and analytic sensitivity analyses [4] are incorporated into the analysis software, sensitivity information can be obtained with a small computational overhead, hence the general preference for the mathematical programming approach and its intensive use in practical engineering applications [5]. Optimality based methods, on the other hand, remain largely on the research side, where exact analytic solutions are generated for special cases and used as test cases for mathematical programming methods.

In an effort to advance the state-of-the art in engineering design, researchers have been investigating novel numerical methods that imitate the development and growth of biological systems. Evolutionary programming techniques, genetic algorithms, and neural networks are among those approaches that have been inspired by natural phenomenon and that have subsequently been implemented for design of structural systems. Although some of these approaches are advocated as suitable for modern parallel computing environments, they are usually used strictly as traditional design optimization tools. In this capacity, they rely on traditional analysis techniques for the computation of the various responses of the structures. Hence, computational parallelization efforts associated with these approaches are mostly coarse-grained parallelization implemented by simultaneous/parallel execution of the analysis for different design and loading conditions repetitively.

A recently emerging approach based on the use of the cellular automata (CA) paradigm is aimed at addressing the automated combined analysis and design of one-, two-, and three-dimensional elastic systems within a massively parallel computational environment. Implementation of the methods of cellular automata to the design of structural systems is, in some sense, an attempt for a simultaneous solution of the state and design variables that appear in the highly nonlinear governing differential equations and associated optimality conditions. The main advantages of CA for structural optimizations are 1) its potential for massively parallel implementation on advanced hardware, 2) both the analysis and design are treated locally, and 3) it lends itself readily to optimality based approaches where the solution of the field problem and the design problem are arrived at simultaneously. The following brief introduction to CA, adopted from [6], will shed more light on the characteristics of

CA that give rise to the above advantages.

1.2 Introduction to Cellular Automata

Cellular automata are generally attributed to Ulam, S. (1952, "Random Processes and Transformations) [7], and von Neumann, J. (1966, "Theory of Self-Reproducing Automata,") [8] who introduced the concept in the late forties to provide a realistic model for the behavior of complex systems. The literature on the subject is not pure in the sense that cellular automata type methods seem to be invented many times under different names, and under somewhat different implementations. Initially they seem to be introduced under the name automata networks, which are used to model discrete dynamical systems in time and space. In that sense, they can roughly be defined by a finite or infinite graph where each vertex can take on discrete values from a finite set. The state of each vertex changes following transition rules which take into account the vertex's current state as well as that of its neighbors in the graph. The network may be updated either synchronously or sequentially. In the synchronous mode, which is also called parallel mode, all the sites are updated in a discrete time scale simultaneously. The sequential update is applicable to only finite networks, and the sites are updated one by one in a prescribed order.

A particular case of automata networks is the cellular automata, in which the graph is a regular lattice and the updating mode is synchronous. Moreover, the update (transition) rules and the neighborhood structure are the same for all sites. A variant of the cellular automata uses continuous lattice site values, and is sometimes referred to as a coupled map-lattice or cell-dynamic scheme. In their modern engineering implementation, cellular automata are simple mathematical idealizations of natural systems, and are used successfully to represent a variety of phenomena such as diffusion of gaseous systems, solidification and crystal growth in solids, fracture mechanics, and hydrodynamic flow and turbulence. In most of the previous applications, they are used to represent macroscopic behavior of a system, which are governed by partial differential equations of the continuum under consideration. This is generally accomplished using simple rules that represent the micro-mechanics of the medium. Using a sufficiently large number of cells, however, it was possible to represent a complex continuum response. In this sense, cellular automata can be viewed as a solution strategy for governing partial differential equations.

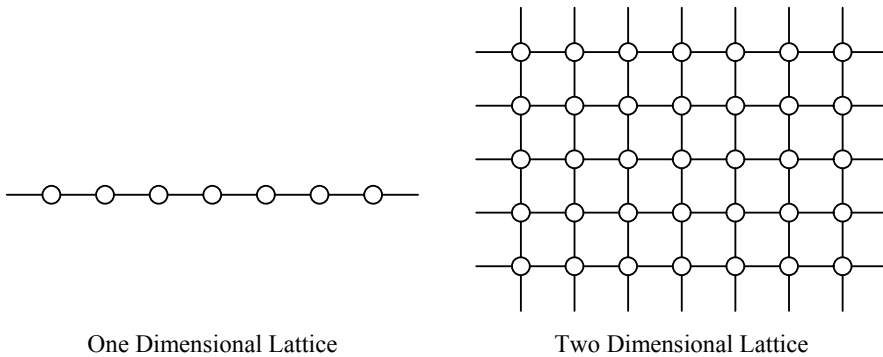


Figure 1.1: One-dimensional CA lattice (left) and two-dimensional rectangular CA lattice (right).

A typical CA algorithm is defined by few basic elements:

1.2.1 CA Lattice

The form of the cellular space directly reflects the physical dimensions of the problem being solved. Two sample lattice structures, representing one- and two-dimensional cellular spaces are shown in Fig. 1.1, where cell locations are indicated by open circles. A three-dimensional space can be constructed by layering several of the two-dimensional ones, spaced equally so that the distance between them is the same as the distance between the cells in the plane. The lattice structures, however, are not limited to the rectangular ones shown in the figure. Cellular automata based on other lattice systems such as two-dimensional trigonal and hexagonal lattices are also possible. Wolfram [9], for example, used a regular two-dimensional lattice of hexagonal cells for a cellular automaton fluid model. Each lattice site has a value or set of values, which are updated over the course of iterations. The site values may be discrete or continuous values allowed to change in a range, or they may be binary (0/1) variables.

1.2.2 The Neighborhood

The neighborhood structure is one of the most important characteristics of a CA lattice. In updating the values of a site, it is necessary to consider the site's own value and the values of the sites in its neighborhood. The set of sites that is utilized for the update is highly problem dependent, and relies heavily on the nature of the physical phenomenon that is being modeled. Some common examples of neighborhood structures used in the literature are shown in Fig. 1.2. The cell to be updated is labeled as C, and the adjacent ones are labeled with letters representing the East, West, North, and South directions. Again, these are not the only neighborhood structures. For example, a neighborhood commonly referred to as the "MvonN Neighborhood" combines the nine sites of the Moore neighborhood with four more sites lying in the north, south, east and west directions two cell spacing distant from the center cell.

1.2.3 Boundaries

Since every cell has the same neighborhood structure, even the cell at the boundary of a physical domain has neighboring cells that are outside the domain. Traditionally, border cells are assumed to be connected to the cells on the opposite boundary as neighbors forming a closed domain. For example, for a two-dimensional rectangular domain, a site on the left border has the site in the same row on the right border as its left (west) neighbor. With the same update rule applied to all the cells, this yields what is called a periodic boundary condition which is representative of an infinite system. With its classical representation of moving particles, for example, a particle leaving the domain from one side enters the domain from the opposite side in the same row or column.

Of course, the type of the boundary condition to be used in a simulation depends on the physical application under consideration. Other types of boundary conditions may be modeled by using preset values of the cell values for the boundary nodes or writing unique update rules for the cells at the boundary. Writing new update rules provides a substantial flexibility in introducing different boundary conditions. Using such techniques, boundaries that reflect or absorb particles, as well as moving boundaries and sources have been created [10, 11].

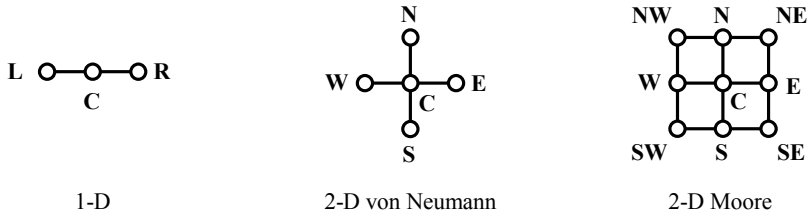


Figure 1.2: Traditional CA Neighborhoods.

1.2.4 CA Update Rules

In a computer implementation, the update rules that are applied to every cell of the lattice are like function subroutines. The arguments for the function subroutine are the values of the sites of the neighborhood, and the value returned by the function is the new value of the cell at which the function is being applied. For example, for the von Neumann neighborhood, the function has 5 arguments, $f(C, E, W, N, S)$ which returns the value of the site C at the time/iteration $t + 1$. Since the update rule is applied to all the cells simultaneously, the incoming arguments are the values of all the cells in the previous cycle (t), and at the new cycle ($t + 1$) all the cells have new values.

The fundamental feature of cellular automata, which makes them highly useful computational tools for large systems, is their inherent parallelism. It is conceivable that by assigning a simple processor to every so many cells of a large system of cells, one can increase the detail or the size of the system without increasing the time it takes to update the entire system. There does not seem to be a practical limitation or an overhead associated with splitting the problem into small pieces and distributing it. Thus, cellular automata simulations are highly suited for massively parallel computers.

1.3 CA on Configurable Computers

The local nature of CA algorithms motivates the implementation of CA algorithms in a massively parallel computing environment. Moreover, the local computations done at the cell level are usually either fixed for the entire CA simulation or needs to be changed infrequently (i.e., for a structural application, the change can be a switch of local rules from analysis to design update modes). General-purpose processors do not achieve maximum efficiency when executing specific applications; they are designed to give high average performance for a variety of tasks but hardly excel in any particular one. In a typical processor, different parts of the chip are optimized for performing certain instructions. While a specific instruction is executed, most of the chip is inactive. Moreover, due to its general-purpose nature, the control logic is generally complicated and adds significant overhead to the computations.

Due to the simplicity of the cellular automata calculations and the minimal storage space required, a classic CA application would only use a fraction of the full capabilities of a general-purpose CPU. Furthermore, since serial processors are only able to perform one operation at a time, the majority of the computation time for each individual cell would consist of waiting for the processor to complete other operations until their local calculation has priority. Thus, the majority of the processor's computational power and speed is wasted. A more favorable architecture for cellular automata would consist of a large number of independent processors that can perform simple operations and that use very little storage space. The most conventional solution to this problem is parallel computing. Multiple serial processors configured in parallel can significantly improve the performance of algorithms that are based on many independent calculations. They can be programmed to eliminate much of the waiting time inherent to individual serial processors by dividing the cell updates among the available processors [10, 12]. Yet, at the processor-level, the efficiency will still be rather low due to the above mentioned inefficiencies inherent in general-purpose processors.

The ideal hardware environment for CA applications would consist of:

1. Highly optimized processing element (PE). The processing element should be built to include cell data (of all the cells assigned to the PE for update), and the required logic and computation modules,

2. Efficient bus communication between different processing elements,
3. High clock rate.

One possibility to achieve the above requirements is through the use of Very Large Scale Integrated Circuits (VLSI). These custom designed chips can be manufactured specifically for a computational application, and offer performance and speed several orders of magnitude greater than what can be achieved through software solutions on general-purposes computers. Cellular automata would be extremely efficient when implemented in a custom VLSI chip designed specifically toward CA rule updates and computations. Unfortunately, the economics of chip design and fabrication are such that a custom VLSI implementation for each CA application is not feasible. Moreover, an ideal design would connect the processing elements in the hardware in a similar pattern to the neighboring cells in the computational domain. Since the configuration of the cells depends largely on the specific problem at hand, such as the shape of the domain and the size of the model, a general chip design that is efficient for general implementations of CA may be difficult to determine.

Configurable computers, sometimes also known as configurable computing machines (CCMs), reconfigurable architectures, or adaptive computing systems, offer another innovative solution to problems that require specialized computations. More specifically, recent research in CCMs using Field Programmable Gate Arrays (FPGAs) demonstrated the ability to program specific computational needs at the hardware level. This method of programming the gate arrays within the chip for a specific application is attractive due to its inherent generality and adaptability. State of the art FPGA chips can be manufactured with up to 10 million gates and it is projected that this number could rise to 50 million by the year 2005. The present FPGA technology offers speeds measured at 30 billion MACs per second (a MAC is a standard multiply-accumulate operation used to measure performance) with similar gains projected in the future [13]. FPGAs can be programmed to implement a specific hardware design, achieving speeds only slightly below (only an order of magnitude) those achieved in a custom VLSI implementation. FPGAs are made in volume and can be reprogrammed as many times as desired, making it an economically viable option for high-performance implementation of applications. In aerospace applications, FPGA's are being considered for onboard processing in spacecraft [14] and more recently as a computational engine for structural analysis

Chapter 1

applications [15].

Configurable computers based on these FPGAs appear to be an excellent candidate for the CA paradigm [16]. Demonstration of a custom computing environment for solving engineering problems that involve local discretization has already been demonstrated. For example, an implementation of a finite difference formulation for the solution of a two-dimensional heat transfer problem was performed, and performance results were compared with those produced on a traditional workstation. The specific example was the analysis of a heat sink with cooling fins, including conduction, convection, and radiation effects [17]. Comparison of the results for a 1024×1024 finite difference mesh (more than 10^6 nodes) showed substantial speed-up with respect to the computing times achieved on a serial Sun SPARC-2 Workstation. This particular application was nearly linearly scalable in terms of performance versus the number of processing element involved. Depending on the clock rate of the CCM and the number of boards used, speed-ups of nearly 20,000 were demonstrated.

It is thus conceived that the cellular automata paradigm can lead through a combination of algorithms and advanced hardware (configurable computers or parallel clusters) to efficient tools for the design of complex structural systems. The initial investigation of this concept is one of the themes for this dissertation.

1.4 Literature Review

Cellular Automata models have been used extensively to model natural phenomena [18]. The ability of CA to produce complex global behavior based on simple cell-level update rules is an important research area in physics and mathematics [9]. In aerospace applications, automata networks have been used for variety of modeling and simulation tasks; simulation of aircraft subsystems [19], modeling of fluid flows (under the name of lattice Boltzmann method) [11, 20]. A comprehensive review of CA applications is beyond the scope of the present work. We focus on reviewing works pertaining to the application of CA to structural analysis and design.

1.4.1 CA for structural analysis/design

CA is definitely a new comer to the field of structural analysis and design. Nevertheless, a number of methods that appear in the structural optimization literature have a basic structure reminiscent of CA algorithms. These methods, especially in the area of topology design, are reviewed in the introduction to the paper by Kita and Toyoda [21]. The work of Kita and Toyoda [21] is the starting point of this review.

The topology design problem considered in [21] is to find the optimal thickness distribution of a two-dimensional continuum (plate) under inplane loads. The basic methodology advocated in [21] consists of,

1. Finite elements are identified as CA cells.
2. The cell neighborhood is identified as the elements sharing a common edge with the cell. For the rectangular FEM mesh used, this is a Moore neighborhood.
3. An update rule is devised, based on stresses in the neighborhood, to update cell thickness.

This work contained some far-reaching features. They formulated the CA design rule, for the first time, as a local optimization problem at the cell (element) level. They based the local update rule on the value of stress resultants in the neighborhood. Moreover, they provided an approximate sensitivity analysis as the basis for selecting the cell (element) level objective function.

The main drawback of their method is that they depended on the evolutionary structural optimization (ESO) method developed by Xie and Steven [22]. In ESO, the von Mises stress is used as a measure to eliminate elements in the domain that are not contributing to the load carrying capacity of the structure. This method is essentially heuristic and was criticized for its lack of mathematical foundations and premature convergence to suboptimal designs in a number of publications [23, 24]. Another disadvantage of this CA algorithm is the large number of iterations (in excess of a thousand) required to reach a converged topology. Given that each CA

Chapter 1

design iteration required a full finite element analysis of the structure, the overall computational cost proved to be excessive.

Another early contribution to the application of CA to structural problems appears in Hajela [25]. In this work, CA rules for the solution of two-dimensional elasticity problems are sought. A general weighted average method is postulated as the general form of the rule. The weight coefficients are determined through the use of a Genetic Algorithm (GA). The objective function of the genetic search was to minimize the norm of the difference between the converged CA solution and the *known* analytic solution of the test problem. This approach has the obvious drawback of the need to know the solution of the problem beforehand.

This line of attack has been further developed in [26, 27] where the need to know the exact solution is removed. In the new approach, competing CA rules are compared based on the value of the total potential energy of the converged design. Since the total potential energy assumes a minimum at the exact solution, CA rules that resulted in lower values of the total potential energy were favored. In this work, several features of the CA rule were open for selection including the neighborhood type, different choices of field variables (displacements, strains, ... etc), and the weights in the cell update formulae.

This approach has the advantage of automating the process of selecting CA rules. On the other hand, it is fairly computationally expensive. For each evaluation of the merit (fitness) of a CA rule, a complete CA simulation must be run. Since rule selection is based on GA which is notorious for the excessive number of function evaluations it needs to converge, the overall computational cost, even for coarse meshes, is quite large. It is also important to point out that this approach considers analysis only and does not address the question of design.

Other work on CA applications to structural analysis has been conducted combining both analysis and design capabilities. This work is described below in some detail for two reasons. First, combined analysis and design is going to be the main (though not the only) focus of CA applications presented in this dissertation, and, second, this work is the base point on which the research reported in this dissertation is conducted.

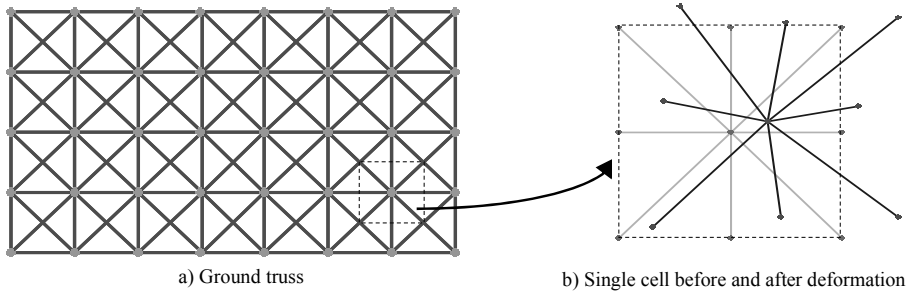


Figure 1.3: Truss ground structure and CA cell.

1.4.2 CA for combined analysis and design

The use of the cellular automata (CA) paradigm aimed at addressing the automated combined analysis and design of two-dimensional elastic systems was implemented recently by Gürdal and Tatting [6]. The basic elements of the methodology discussed in the previous section were demonstrated by using a simple two-dimensional domain occupied by a ground truss structures (see Fig. 1.3). In the truss ground structure, each cell was made up of eight truss members extending from the cell center to each of the eight neighbors in Moore neighborhood. The position of the cell center and the cross-sectional areas of each of the eight members are the unknown variables that need to be computed based on local rules. Two types of CA rules were derived for the optimization of truss structures, an analysis rule, and a design rule. The analysis rule was based on writing the equilibrium equations for each truss joint (cell center). In this fashion, two equations that can be solved for the cell displacements were derived. Analytic expressions for the cell displacements in terms of the displacements of the neighbors and the areas of connected members were obtained for different cell boundary conditions (free, pinned and roller). The design rule was based on the stress ratio method [4]. The stress ratio method is equivalent to the exact optimal solution for the design of trusses made of a single material for minimum weight and stress constraints in each member [4, 28].

The method was also extended to the analysis and design of continuum structures by Tatting and Gürdal [29] by representing a continuum cell as an assembly of orthogonal and diagonal truss members with properties related to the thickness and elastic properties of a two-dimensional continuum through the equivalence of the

Chapter 1

strain energy stored in the continuum and the truss domains. Numerical studies revealed several advantages of the new method. For both the truss and continuum domains, design results could be generated quite efficiently even for a very dense lattice of cells. For example, Fig. 1.4 shows the final designs of a U-shaped bracket represented by about 10,000 cells using both the truss and continuum definitions. The initial ground structure was defined by cells that have uniform non-zero cross-sectional area members inside domain represented by the dashed polygon, with zero cross-sectional properties elsewhere. During the design process, the geometric parameters of the cells were re-sized (nearly eighty thousand of them for the truss-domain) to minimize the cells' weight while keeping the local stresses (in all the members connected to the cells) below the stress allowable for the material.

Although the number of iterations needed for the combined analysis and design was large, the method proved to be more efficient than traditional iterative finite element analysis based optimization processes since each iteration for the current method required only a few computations per cell. In a similar example (for the design of a Mitchell truss), the cellular automata based design was two orders of magnitude faster than a state-of-the-art commercial design optimization package (GENESIS) with comparable accuracy.

The use of the cellular automata algorithm for design seemed to exhibit other potential advantages. In some cases, it was shown that designing a given configuration required fewer iterations than performing a single analysis of an arbitrary design [6]. This behavior was largely attributed to the formation of load paths (through disappearance of the unneeded material in certain parts of the domain) within the structure during the early stages of the design cycle that accelerated the convergence of local displacement fields that were most important for the deformation of the converged design of the structure.

Another interesting feature that was illuminated in the earlier study was the capability of the CA approach in handling material and geometric nonlinearities with almost no additional computational overhead [29]. Because the analysis and design iteration computations were all performed locally within each cell, the deformed positions of the neighboring cells were immediately available for the center cell computations. Therefore, local equilibrium conditions for the cells could be written based on deformed configurations. Hence, performing nonlinear analysis required very little increase in computational time compared to the linear analysis. In fact,

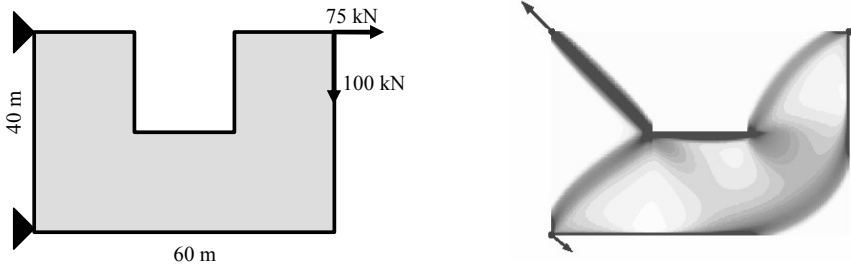


Figure 1.4: U shaped bracket; Problem definition (left), CA converged optimal topology (right).

nonlinear analyses typically required fewer iterations for convergence than linear analysis, making the method highly suitable for designing structures with nonlinearities.

The above work, although showing much promise, leaves much to be desired. First, the CA analysis update rule for a truss domain can be derived in a straightforward and logical manner because the truss structure is inherently discreet. The question arises as to what would be a more general approach to deriving CA analysis rules for continuum problems. The approach used in [29] where the continuum is modeled using an equivalent truss is not generally applicable. It is observed in [29] that the energy equivalence holds exactly for isotropic continuum only for a Poisson's ratio of $1/3$. The use of genetic algorithms to find the CA rules as discussed earlier is not attractive because of its computational cost. Another important aspect is that the design of trusses for minimum weight and stress constraints represents a simple problem for which the optimality conditions are well-known. Whether CA design rules can be extended to other important classes of structural optimization problems is left as an open area of further investigation

As discussed earlier, one of the main advantages of CA is its inherent parallelism. A parallel implementation of the method was demonstrated using standard programming languages and parallelization libraries by Slotta *et. al.*[12]. This work is based on the earlier work by Gürdal and Tatting [6, 29]. Numerical studies were performed using the ICASE parallel computing Beowulf-cluster (CORAL) to demonstrate the inherent parallelism of the method. By assigning groups of cells to different proces-

sors, the workload could be evenly distributed to dramatically lower computation time. The results revealed the perfect suitability of the CA approach for parallel computation, due to the local update algorithms and the minimal amount of information that needed to be shared between processors. Rudimentary numerical studies were also conducted to seek improvements in the computational performance for a parallel computing environment. In the same work [12], the stability of different schemes of ordering the cell updates (Gauss-Seidel vs. Jacobi) were compared numerically. The main conclusion was that the Gauss-Seidel updates are faster and more stable than the more straightforward Jacobi method. It is not possible to generalize these conclusions since the study relied entirely on numerical experiments.

1.5 Scope and Objectives

We can conclude from the literature reviewed that the Cellular Automata (CA) paradigm shows promise as a basic paradigm for structural analysis and/or design algorithms aimed at parallel and massively parallel hardware. The objective of this work is to investigate the application of the CA paradigm to a variety of structural analysis and design problems to demonstrate its potential. It is also important for this initial investigation to be able to come up with a set of recommendations as to what are the applications where CA will have a strong comparative advantage, viable strategies for combining the analysis and design, and methods for deriving the design update rules.

To achieve these objectives, detailed CA algorithms for the combined analysis and design of structures were developed for a number of selected applications. The first application was design for eigenvalue requirements. This class of problems seems, at first consideration, to preclude solution using a local algorithm since eigenvalues are properties of the whole structure rather than local quantities (e.g., stresses as used in previous studies). The successful development of a CA algorithm capable of solving this class of problems was the first contribution of this dissertation.

The second application was topology design of two dimensional continuum. The motivation behind the selection of topology design was manifold. First, this problem was addressed in earlier work on CA [21, 29] which was limited in generality as discussed before. Second, topology design methods have been the prey of nu-

merical instabilities that we felt can be efficiently suppressed by using local rather than global design strategies. Third, the application of CA to the analysis of multi-dimensional continuum was not yet fully investigated, and the integration of analysis and design rules was not fully considered. For these reasons, topology design was deemed to be the logical next step. The CA topology design algorithm developed in this dissertation accomplished elegantly all the above goals. For topology design, we considered CA both as a strict design tool to be coupled with traditional finite element analysis, and the case where CA was used for both design and analysis. Numerical instability was completely suppressed and a viable combined analysis and design methodology was developed that deals directly with the continuum model base on rigorous optimality criteria.

The applications of CA to structural optimization that were considered so far were governed by linear field equations. The only exception is the inclusion of nonlinearities in [29]. The development of CA algorithms capable of addressing strong nonlinear problems remained an untouched territory. The next application was the shape design of a MEMS (micro-electro-mechanical-systems) microbeam. The method of actuation of many MEMS devices is based on deformation-dependent electrostatic fields. The nonlinear relation between deformation and electric field leads to strong nonlinear behavior and physical instabilities that constitute an important research topic in the MEMS community. Shape design of micro-beams including the full nonlinear behavior has never been addressed before, and as such, the problem presented an interesting topic of research. At this stage, CA capability of tracing nonlinear response was only rudimentary investigated [29], for this reason it was decided to use a CA local design algorithm coupled with traditional nonlinear finite element analysis with arclength control for tracing the nonlinear response. The CA design algorithm performed satisfactorily for this problem showing robust convergence behavior.

The CA algorithms developed in this dissertation clearly demonstrate the potential of CA as a useful paradigm for structural analysis and design.

1.6 Organization of the Dissertation

After this introductory chapter, the dissertation is divided into four chapters and a concluding chapter. In chapter 2, an algorithm for designing structures for eigenvalue requirements is presented. The proposed algorithm, being fully local in nature, lends itself to CA type implementation. To illustrate the effectiveness of the proposed approach, the design of Euler-Bernoulli columns for a prescribed buckling load is considered. The proposed algorithm, features local analysis and update rules. The analysis rule is obtained through the minimization of the total potential energy in a cell neighborhood. The design rule is formulated as a heuristic cell-level optimization problem similar to the ideas proposed by Kita and Toyoda [21]. Excellent agreement between the CA results and known exact solutions is obtained. A more complex column design problem with local constraints is also considered. In the absence of a known analytic solution, the CA design is compared to the design obtained using the state of the art structural optimization software. The CA algorithm is demonstrated to be effective in solving unimodal optimal column problems.

By a careful study of the algorithm presented in chapter 2, it was recognized that the success of the algorithm was based on the exact correspondence between the proposed heuristic local optimization problem and the rigorous optimality criteria for the particular cases we considered. Based on this observation, it was determined that CA design rules can be obtained rigorously by using variational calculus. Under this new light, chapter 3 presents a CA topology design algorithm where the design rules for minimum compliance design of two-dimensional linearly elastic continuum topology are derived using variational calculus. The topology problem is regularized using the popular Solid Isotropic Material with Penalization (SIMP) approach. The CA design rule is derived based on continuous optimality criteria interpreted as local Kuhn-Tucker conditions. The CA design rule was linked with a finite element based global analysis of the two-dimensional elasticity problem. Also, CA based analysis rules were considered in combination with the CA design rule. Numerical experiments with the proposed algorithm indicated that the CA design rule is quite robust and does not suffer from checkerboard patterns, mesh-dependent topologies, or numerical instabilities. This presented a major departure from traditional topology design methods that relied either on heuristics or computationally expensive perimeter and slope control methods to suppress the numerical instabilities encountered in topology optimization. The ability of CA to perform

the role of an *optimizer* in combination with standard FEM analysis was also significant in that it demonstrated that CA design algorithms can be beneficial on their own right even when a combined CA analysis and design capability is not sought.

Encouraged by the success of optimality-based CA design rules in topology optimization, CA design rules for nonlinear problems showing limit point behavior were derived in chapter 4 using rigorous optimality conditions. The design rule was successfully formulated as a local cell-level optimization problem. The CA design rule was couple to nonlinear finite element analysis to solve the problem of shape design of a MEMS microbeam. The convergence of the CA design rule was quite fast requiring only twenty to thirty nonlinear finite element analyses. The results confirmed that local design rules based on optimality perform satisfactorily and could be indeed considered a general method for deriving CA design rules. The results indicate that substantial increase in pull-in voltage can be achieved by varying the width and/or thickness distribution.

The dissertation is concluded by chapter 5, where general conclusions regarding the domain of applicability of CA, its potential extensions and the relation between the design and analysis algorithms and software and hardware are discussed.

Chapter 2

Cellular Automata Algorithm for Eigenvalue Problems

2.1 Introduction

Cellular Automata (CA) algorithms, by their very nature, require a local formulation. This applies for both the update of field variables (e.g., cell deformation) and design variables (local cross section area). One of the first *challenges* to such a strategy that come to mind is eigenvalue problems. Eigenvalues are not point functions as is the case with displacements and stresses. The complete structure including boundary conditions and loading contribute to the determination of the eigenvalues. In this chapter, the objective is to demonstrate the use of CA for the design of continuum structures for a specified eigenvalue. This important class of problems includes the design of structures for a given buckling load or natural frequency. A general algorithm is proposed for this class of problems that relies on heuristic engineering common sense. The basic idea is to convert the eigenvalue design problem to a repetitive application of a simple stress based design sizing rule. The displacements of CA cells are computed according to an energy based local update rule. For the first time, CA analysis update is based on energy principles rather than direct equilibrium, which allows for greater generality. By basing the analysis rule on energy principles, CA rules can be derived for continuum structures without the need to

Chapter 2

find equivalent discreet models such as trusses [29]. Although the proposed method is general enough to solve more complex continuum problems, the work presented here specifically addresses the design of Euler-Bernoulli columns against buckling. The presentation given here is based on the work in reference [30].

Historically, two major approaches to the solution of the column buckling design problem can be identified. The first approach is based on continuous optimality criteria method and employs the calculus of variations. In this approach the continuous distribution of cross sectional area is the unknown to be determined, and the moment of inertia of the cross section is assumed to be proportional to the area raised to a fixed power. Both geometrically constrained (designs with minimum area constraint) and geometrically unconstrained problems are considered in the literature for various boundary conditions [31, 32, 33]. When the optimality conditions (a set of integro- differential equations) cannot be analytically solved, some numerical approximation method is employed such as the finite element method [34]. Although this approach leads to analytic solutions and considerable insight into the buckling design problem [35], it is not generally used in practice because of the limited freedom in the choice of the type of column cross section and the difficulty of incorporating local constraints.

The second traditional approach, which is generally used for practical problems, is based on mathematical optimization. The column is divided into a number of finite elements and the cross section area and moment of inertia (or other geometric dimensions) of each element are used as design variables. The problem is based as a mathematical programming problem and classical optimization methods [4] are used to find the optimal solution. When attacked in this manner, eigenvalue design problems require a repetitive determination of eigenvalues of a potentially large system of equations within an outer loop of a design optimization formulation. When a large number of structural properties are used as design variables, this formulation is computationally intensive. For that reason, approximation methodologies [36] are frequently employed to reduce the required number of eigenvalue evaluations. Simultaneous Analysis and Design (SAND), as used in [37] for buckling design, attempts at the simultaneous solution of the finite element equations and the mathematical optimization problem. Although SAND avoids the need of nesting design and analysis, it tends to produce large nonlinear systems which are difficult to solve.

More recently, novel approaches to eigenvalue design problems were introduced.

A genetic algorithm (GA) is used for buckling design of columns in [38]. The use of GA does not seem to introduce much computational savings, since it is not well suited to problems with large number of continuous variables.

CA, by its combined local analysis and design approach circumvents the inefficiency noted above by arriving at an improved design while simultaneously performing analysis. The algorithm presented here does not require the determination of eigenvalues or eigenvectors, thus potentially providing large savings in computational time. Massively parallel implementation, which the CA naturally calls for, is expected to improve computational efficiency in the future.

In the following, the general eigenvalue design algorithm is introduced, followed by a single degree of freedom example as a demonstration. As a specific application, the design of Euler-Bernoulli columns against buckling is considered. Equilibrium update rules for Euler-Bernoulli columns are derived next. The design algorithm for columns is then presented followed by a number of numerical examples that compare CA designs with known analytic solutions. Also, a design problem with cross-sectional manufacturing constraints is solved and the CA design is compared with GENESIS [39]. The CA algorithm is shown to perform satisfactorily for the problems considered and to reproduce known analytic solutions accurately.

2.2 Eigenvalue Requirement Design Algorithm

The generic equations governing the structure is assumed to be of the form:

$$\mathcal{L}(\mathbf{d})\mathbf{u} = \lambda\mathcal{H}(\mathbf{d})\mathbf{u} \quad (2.1)$$

where \mathcal{L} , \mathcal{H} are operators and λ is a given eigenvalue. \mathbf{u} represent the dependent (field) variables while \mathbf{d} represents the design variables, both are assumed to be defined over the domain of the problem Ω .

The operators \mathcal{L} and \mathcal{H} can be selected to describe structural problems in one, two or three space dimensions. Proper selection of these operators depends on the structural theory being used. At this point they are left completely arbitrary.

In order to excite the system (2.1), a fictitious source term \mathbf{f} may be introduced to

Chapter 2

the right hand side of (2.1) to give:

$$\mathcal{L}(\mathbf{d})\mathbf{u} = \lambda\mathcal{H}(\mathbf{d})\mathbf{u} + \mathbf{f} \quad (2.2)$$

Local to each point in the structure, a *stress measure* $\sigma(\mathbf{u}, \mathbf{d})$ is assumed to be defined in terms of the design and field variables. Also defined is a *strength measure* $S(\mathbf{d})$. The weight of the structure (objective function to be minimized) is assumed to be represented in integral form as:

$$W = \int_{\Omega} \rho(\mathbf{d}) d\Omega \quad (2.3)$$

where $\rho(\mathbf{d})$ is a pointwise defined *density measure*. The iterative algorithm for the solution of both the field and design variables is:

Algorithm

1. Initialize \mathbf{d} , \mathbf{f} and \mathbf{u} .
2. Solve the problem (at the $k + 1$ iteration):

$$\mathcal{L}(\mathbf{d}^k)\mathbf{u}^{k+1} = \lambda\mathcal{H}(\mathbf{d}^k)\mathbf{u}^k + \mathbf{f}^k \quad (2.4)$$

where $\{\mathbf{f}^k\}_{k=1}^{\infty}$ is such that:

$$\lim_{k \rightarrow \infty} \mathbf{f}^k = 0$$

This determines a new distribution of the field variables.

3. At each point solve the optimization problem:

Minimize: $\rho(\mathbf{d}^{k+1})$

Subject to:

$$\sigma(\mathbf{u}^{k+1}, \mathbf{d}^k) \leq S(\mathbf{d}^{k+1}), \quad \mathbf{g}(\mathbf{d}^{k+1}) \leq 0$$

where $\mathbf{g}(\mathbf{d})$ are pointwise defined side constraints.

4. Return to (2) and repeat until convergence.

When local update rules are used in step 2, the algorithm becomes completely local in nature, thus, completely consistent with the CA paradigm.

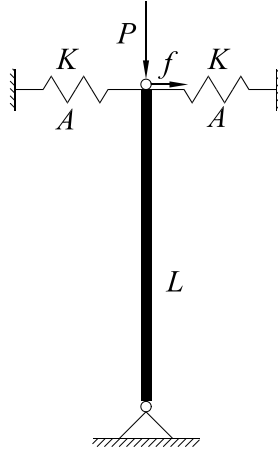


Figure 2.1: A rigid pole supported by springs under compression.

2.2.1 Example

To demonstrate the design algorithm in its simplest form, a single degree of freedom system is considered. Figure 2.1 depicts the system, where a rigid pole of length L is hinged at the bottom and supported by two identical springs of stiffness K at its tip. The objective is to find the correct design of the springs to avoid instability under the specified compressive load P . The equilibrium equation for the system takes the form:

$$\underbrace{2KL\theta}_{\mathcal{L}(\mathbf{u})} = \underbrace{P}_{\lambda} \underbrace{\theta}_{\mathcal{H}(\mathbf{u})} + \underbrace{f}_{\mathbf{f}} \quad (2.5)$$

where the field variable is the pole rotation θ . It follows directly from (2.5) that the critical value of the spring stiffness is $K_{cr} = P/(2L)$.

To formulate the problem in the terms introduced earlier, the stress measure is taken as the force in the spring $\sigma = KL\theta$ and the strength measure is assumed to be proportional to the cross sectional area of the spring $S = S_o A$. The spring stiffness is also proportional to the spring area A used as the design variable (i.e., $K = CA$, where C is a constant). With these definitions, step 2 of the design algorithm takes the form:

$$2K^k L \theta^{k+1} = P \theta^k + f^k \quad (2.6)$$

Chapter 2

Solving for θ^{k+1} and simplifying we obtain:

$$\theta^{k+1} = \frac{K_{cr}}{K^k} \theta^k + \tilde{f}^k \quad (2.7)$$

where $\tilde{f}^k = f^k / (2 K^k L)$.
and step 3 takes the form:

Minimize: A^{k+1}

Subject to: $K^k L \theta^{k+1} \leq S_o A^{k+1}$

The solution of this local optimization problem is simply:

$$A^{k+1} = K^k L \theta^k / S_o \quad (2.8)$$

From (2.7) we see that for $K^k < K_{cr}$ (*under-design*), the response is *accentuated*, while for $K^k > K_{cr}$ (*over-design*), the response is *attenuated*. This is the key point in the algorithm. Since the response of an under-design is accentuated, the stress measure increases, and in the next design step the stiffness of the design is also increased as seen from (2.8). The reverse happens when we have an over-design. The net effect is that the algorithm converges to the correct stiffness to support the load.

Note that the introduction of a the fictitious load f is not necessary for the algorithm to work. The only consequence of eliminating f altogether is that the solution procedure cannot be started from the undeflected position $\theta = 0$. Figure 2.2 shows the convergence of the algorithm for this simple case.

2.3 Buckling Design of Columns

To illustrate the ability of the algorithm to deal with practical problems, elastic column design for a specified buckling load is considered. Buckling design refers to finding the optimal material distribution of the column so that a given compressive load is supported without losing stability while minimizing the total volume of the

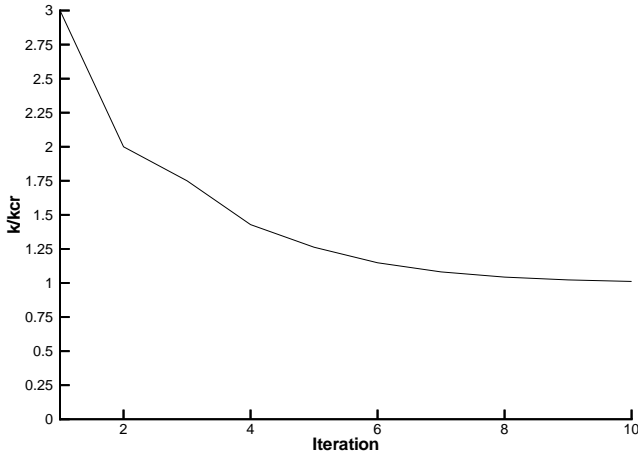


Figure 2.2: Convergence of spring stiffness.

column material. The governing equation is:

$$\frac{d^2}{dx^2} \left(EI(\mathbf{d}) \frac{d^2 w}{dx^2} \right) = -P \frac{d^2 w}{dx^2} + p(x), \quad 0 \leq x \leq L \quad (2.9)$$

where x is a coordinate along the column axis, E is Young's modulus of the column material, and I is the moment of inertia of the cross section which is assumed to be symmetric. Buckling in the plane of symmetry is exclusively considered. The dependent variable $w(x)$ is the lateral displacement, and $p(x)$ is a fictitious distributed load corresponding to \mathbf{f} in (2.2). The stress measure is taken to be the bending moment defined by:

$$M(x) = EI \frac{d^2 w}{dx^2} \quad (2.10)$$

and the strength measure is taken as the maximum allowable stress S_{all} multiplied by the section modulus Z defined by:

$$Z = \frac{I}{z} \quad (2.11)$$

Chapter 2

where z is the distance between the extreme fiber of the cross section and the neutral axis. The weight of the column is given by:

$$W = \int_0^L A(x) dx \quad (2.12)$$

thus, the density measure is the cross section area $A(x)$.

In the absence of side constraints, the solution of the cell-level optimization problem is fully stressed, this gives the design update rule as:

$$\frac{I}{z} = \frac{M}{S_{all}} \quad (2.13)$$

2.3.1 Equilibrium Update Rules

Derivation of cell-level local update rules for field variables is a key step of any CA implementation. Each cell should be capable of finding its deformation state for any given deformation state of its neighbors. For column design, we obtain the update rules by discretizing the governing equation (2.9). The domain of the solution is divided into a number of cells. The beam cross section is assumed to be constant over each cell and the field variables are associated with the midpoint of each cell as shown in fig. 2.3. The field variables are bending displacement and rotation, $\mathbf{u} = (w, \theta)$. The neighborhood of each cell comprises the cell itself (C) and two neighbors, called left (L) and right (R) neighbors. The displacement field is considered to be of the form:

$$w = w_i H_1(\xi) + h \theta_i H_2(\xi) + w_j H_3(\xi) + h \theta_j H_4(\xi) \quad (2.14)$$

where H_i are hermitian interpolation functions, $\xi = x/h$ is a non dimensional independent variable, and h is the lattice spacing. The displacement field is constructed in the form (2.14) for the four different segments of the control volume indicated by the dashed lines in fig. 2.3 by introducing two auxiliary sets of cell variables associated with the middle left (ML) and the middle right (MR) points. Thus, the kinematic variables are:

$$\mathbf{q} = \left(\mathbf{u}_C \mid \mathbf{u}_{ML} \mid \mathbf{u}_{MR} \right) \quad (2.15)$$

and neighbor displacements are:

$$\mathbf{p} = \left(\mathbf{u}_L \mid \mathbf{u}_R \right) \quad (2.16)$$

The equilibrium equation (2.9) is equivalent to the minimization of the total potential energy inside the control volume. The resulting equations are:

$$\mathbf{K} \cdot \mathbf{q} = \mathbf{K}_g \cdot \mathbf{q} + \mathbf{C}_g \cdot \mathbf{p} - \mathbf{C} \cdot \mathbf{p} + \mathbf{f}_{ex} \quad (2.17)$$

where the stiffness and geometric matrices \mathbf{K} and \mathbf{K}_g are given by:

$$\mathbf{K} = \frac{\partial^2 \Phi}{\partial \mathbf{q} \partial \mathbf{q}}, \quad \mathbf{K}_g = \frac{\partial^2 \Phi_g}{\partial \mathbf{q} \partial \mathbf{q}} \quad (2.18)$$

and the *clamp* matrices \mathbf{C} and \mathbf{C}_g are given by:

$$\mathbf{C} = \frac{\partial^2 \Phi}{\partial \mathbf{p} \partial \mathbf{q}}, \quad \mathbf{C}_g = \frac{\partial^2 \Phi_g}{\partial \mathbf{p} \partial \mathbf{q}} \quad (2.19)$$

where the strain energies Φ and Φ_g are given by:

$$\Phi = \int_{\Omega_c} EI \left(\frac{d^2 w}{dx^2} \right)^2 dx, \quad (2.20)$$

$$\Phi_g = P \int_{\Omega_c} \left(\frac{dw}{dx} \right)^2 dx \quad (2.21)$$

where Ω_c is the cell control volume. The external load vector \mathbf{f}_{ex} represents the effect of $p(x)$. Since $p(x)$ is arbitrarily chosen, the load vector is assumed to consist of a concentrated force and couple at each cell, thus:

$$\mathbf{f}_{ex} = \left(F \quad M \mid 0 \quad 0 \mid 0 \quad 0 \right) \quad (2.22)$$

Since the external load at the auxiliary points is zero, equilibrium equations enable the elimination of the variables associated with these neighbors. This process is similar to static condensation. We start by partitioning \mathbf{K} and \mathbf{C} as:

$$\mathbf{K} = \left[\begin{array}{c|c|c} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{K}_{13} \\ \hline \mathbf{K}_{12}^T & \mathbf{K}_{22} & \mathbf{0} \\ \hline \mathbf{K}_{13}^T & \mathbf{0} & \mathbf{K}_{33} \end{array} \right] \quad (2.23)$$

Chapter 2

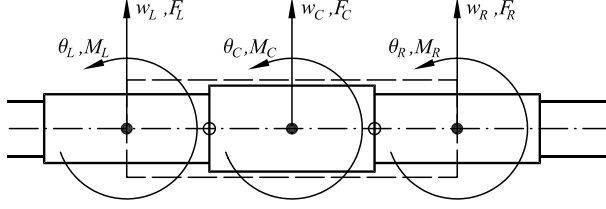


Figure 2.3: Cell neighborhood. ● cell variables, ○ auxiliary variables.

and

$$C = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \quad (2.24)$$

with similar partitions for K_g and C_g . The auxiliary variables are thus eliminated as:

$$\mathbf{u}_{ML} = -K_{22}^{-1} \cdot \left(K_{12}^T \cdot \mathbf{u}_C + C_2 \cdot \mathbf{p} \right), \quad (2.25)$$

$$\mathbf{u}_{MR} = -K_{33}^{-1} \cdot \left(K_{13}^T \cdot \mathbf{u}_C + C_3 \cdot \mathbf{p} \right) \quad (2.26)$$

This form of condensation neglects the geometric matrix contributions at the intermediate neighbors. This is deliberately done to make the geometric terms appear only as forcing terms. The consistent reduced equations of the system take the form:

$$\tilde{K} \cdot \mathbf{u}_C = \tilde{K}_g \cdot \mathbf{u}_C + \tilde{C}_g \cdot \mathbf{p} - \tilde{C} \cdot \mathbf{p} + \tilde{\mathbf{f}}_{ex} \quad (2.27)$$

where

$$\tilde{K} = K_{11} - K_{12} \cdot K_{22}^{-1} \cdot K_{12}^T - K_{13} \cdot K_{33}^{-1} \cdot K_{13}^T, \quad (2.28)$$

$$\begin{aligned} \tilde{K}_g = & K_{g11} + K_{12} \cdot K_{22}^{-1} \cdot K_{g22} \cdot K_{22}^{-1} \cdot K_{12}^T + \\ & K_{13} \cdot K_{33}^{-1} \cdot K_{g33} \cdot K_{33}^{-1} \cdot K_{13}^T - K_{g12} \cdot K_{22}^{-1} \cdot K_{12}^T - \\ & K_{12} \cdot K_{22}^{-1} \cdot K_{g12}^T - K_{g13} \cdot K_{33}^{-1} \cdot K_{13}^T - \\ & K_{13} \cdot K_{33}^{-1} \cdot K_{g13}^T, \end{aligned} \quad (2.29)$$

$$\tilde{C} = C_1 - K_{12} \cdot K_{22}^{-1} \cdot C_2 - K_{13} \cdot K_{33}^{-1} \cdot C_3, \quad (2.30)$$

$$\begin{aligned}
 \tilde{C}_g = & C_{g1} + K_{12} \cdot K_{22}^{-1} \cdot K_{g22} \cdot K_{22}^{-1} \cdot C_2 + \\
 & K_{13} \cdot K_{33}^{-1} \cdot K_{g33} \cdot K_{33}^{-1} \cdot C_3 - K_{g12} \cdot K_{22}^{-1} \cdot C_2 - \\
 & K_{12} \cdot K_{22}^{-1} \cdot C_{g2} - K_{g13} \cdot K_{33}^{-1} \cdot C_3 - \\
 & K_{13} \cdot K_{33}^{-1} \cdot C_{g3},
 \end{aligned} \tag{2.31}$$

and

$$\tilde{\mathbf{f}}_{ex} = \begin{pmatrix} F & M \end{pmatrix} \tag{2.32}$$

Thus, after simplification, the equilibrium relations for a cell, written exclusively in terms of its left and right neighbors, take the form:

$$\frac{8EI_C}{h^3} \begin{bmatrix} S_{11} & -S_{12} \\ -S_{12} & S_{22} \end{bmatrix} \cdot \begin{Bmatrix} w_C \\ h \theta_C \end{Bmatrix} = \begin{Bmatrix} \tilde{F} \\ \tilde{M} \end{Bmatrix} \tag{2.33}$$

where,

$$\begin{aligned}
 S_{11} &= 12[c(1+c) + 2d(d-1) + d(28+15c)] \\
 S_{12} &= 3(a-b)(3+c+11d) \\
 S_{22} &= c(7+c) + d(196+21c) + 2d(d-1)
 \end{aligned} \tag{2.34}$$

$$a = EI_L/EI_C, \quad b = EI_R/EI_C, \tag{2.35}$$

$$c = a + b, \quad d = ab \tag{2.36}$$

$$\tilde{F} = F + F_g + F_e, \quad \tilde{M} = (M + M_g + M_e)/h \tag{2.37}$$

$$F_e = \frac{8EI_C}{h^3} [6g_1(a)w_L + hg_2(a)\theta_L + 6g_1(b)w_R - hg_2(b)\theta_R] \tag{2.38}$$

$$M_e = -\frac{8EI_C}{h^3} [g_3(a)w_L - hg_1(a)\theta_L - g_3(b)w_R - hg_1(b)\theta_R] \tag{2.39}$$

$$\begin{aligned}
 F_g = & \tilde{P}_1 [f_2(a)w_C - hf_3(a)\theta_C - f_2(a)w_L - \\
 & hf_4(a)\theta_L] + \tilde{P}_2 [f_2(b)w_C + hf_3(b)\theta_C - \\
 & f_2(b)w_R + hf_4(b)\theta_R],
 \end{aligned} \tag{2.40}$$

Chapter 2

$$\begin{aligned}
 M_g = & -\tilde{P}_1 [f_3(a) w_C - h f_5(a) \theta_C - f_3(a) w_L - \\
 & h f_6(a) \theta_L] + \tilde{P}_2 [f_3(b) w_C + h f_5(b) \theta_C - \\
 & f_3(b) w_R - h f_6(b) \theta_R],
 \end{aligned} \tag{2.41}$$

$$\tilde{P}_1 = \frac{P}{30 f_1^2(a) h}, \quad \tilde{P}_2 = \frac{P}{30 f_1^2(b) h} \tag{2.42}$$

and,

$$\begin{aligned}
 f_1(r) &= 1 + 14r + r^2 \\
 f_2(r) &= 72(1 + 12r + 102r^2 + 12r^3 + r^4) \\
 f_3(r) &= 3(13 + 24r + 234r^2 - 16r^3 + r^4) \\
 f_4(r) &= 3(1 - 16r + 234r^2 + 24r^3 + 13r^4) \\
 f_5(r) &= 2(19 + 86r + 380r^2 + 26r^3 + r^4) \\
 f_6(r) &= (1 - 100r - 58r^2 - 100r^3 + r^4) \\
 g_1(r) &= 2r(1 + r)/f_1(r) \\
 g_2(r) &= 3r(1 + 3r)/f_1(r) \\
 g_3(r) &= 3r(3 + r)/f_1(r)
 \end{aligned} \tag{2.43}$$

When the applied axial load is not uniform (e.g., buckling of a column under its own weight), or design dependent (e.g., statically indeterminate frames), an additional axial degree of freedom can be added and updated using the axial equilibrium equation. In this paper, only columns with uniform applied compressive load are considered.

2.3.2 CA Design Algorithm

The equilibrium update rules of the previous section represent a discretization of the structural operators in (2.9). The next step in implementing the design algorithm is to calculate the stress measure, which will be used for the design update rule. Since boundary conditions are applied at the middle point of a cell, the bending moment is calculated at the end points of the cell to avoid numerical difficulties at free or hinged boundary conditions. The bending moment at the two ends of the cell is

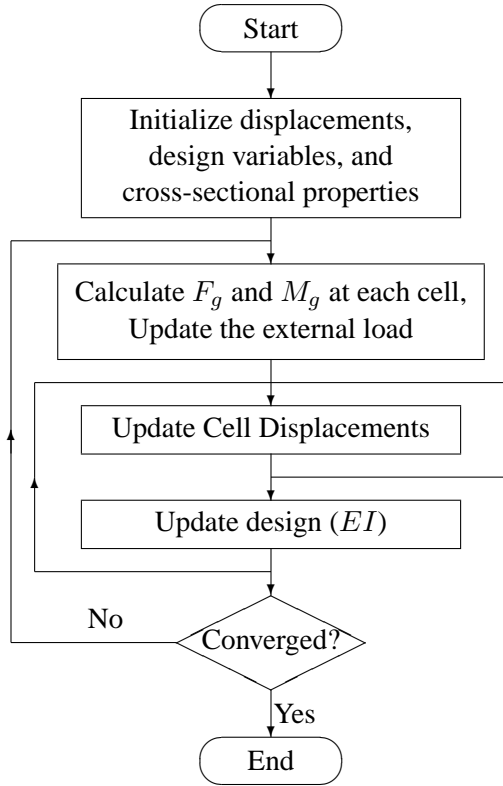


Figure 2.4: Flow chart of the column design algorithm.

given by:

$$M_L = \frac{EI_L}{f_1(a)h^2} (24(a-1)w_C - 4(a-5)h\theta_C - 24(a-1)w_L - 4(5a-1)h\theta_L), \quad (2.44)$$

$$M_R = \frac{EI_R}{f_1(b)h^2} (24(b-1)w_C + 4(b-5)h\theta_C - 24(b-1)w_R + 4(5b-1)h\theta_R) \quad (2.45)$$

Chapter 2

Thus, the stress measure is given by:

$$M_{max} = \max \{|M_L|, |M_R|\} \quad (2.46)$$

For geometrically similar cross sections, the design rule (2.13) simplifies to:

$$A = k \left(\frac{M_{max}}{S_{all}} \right)^{3/2} \quad (2.47)$$

where A is the area of the cross section and k is a constant that depends on the shape of the cross section (e.g., square, circular, ... etc.).

The flow chart of the design algorithm is shown in fig. 2.4. Three nested loops can be identified. The innermost loop consists of displacement (field variables) updates using (2.33). This loop is embedded in an intermediate loop in which the design is updated using (2.13). Throughout these computations the loads (F , M , F_g and M_g) are kept fixed. The outermost loop comprises updating the loads and checking for convergence. In our implementation, the inner loops are iterated a fixed number of times to reach a reasonable equilibrium distribution before updating the design. It was also found that under-relaxation of the geometric loads (F_g and M_g) is necessary. The amount of under-relaxation depends on the particular problem and boundary conditions.

2.4 Numerical Examples

The following examples demonstrate the ability of the CA methodology for designing continuum structures with constraints on eigenvalues. The examples cover a number of support conditions with and without geometric constraints.

2.4.1 Clamped-Free Column

First, we consider a clamped-free column. Following [31], we impose no minimum area constraint. The cross-sections are assumed square, so that geometric similarity

is satisfied. Figure 2.5 shows the normalized optimal area distribution as compared to the CA design using 30 cells. The agreement is excellent except near the tip, where the cross sectional area vanishes and the analytic solution is singular. This is further illustrated in fig. 2.6 which compares the mode shapes of the CA prediction (10 cells) and the analytically determined mode shape. NASTRAN simulation of the CA design is also shown and it gives identical results to CA predictions. Figure 2.7 depicts percentage weight saving (compared to uniform column design) using CA as the number of cells is varied. It is clear that as the number of cells increases, the CA design approaches the theoretical maximum weight saving.

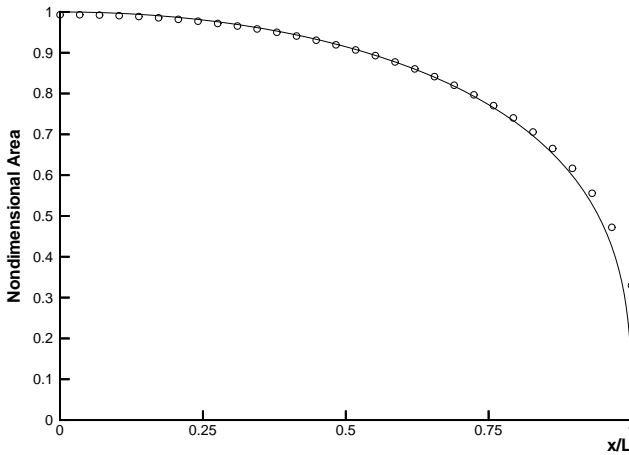


Figure 2.5: Clamped-free column; area distribution of the optimal column. — Analytic solution, \circ CA solution for 30 cells.

2.4.2 Simply Supported Column

The second example is a simply supported column with a minimum area constraint. The exact solution of [32] is applicable to this case. A column of length $L = 1$ m made of aluminum (Young's modulus $E = 70$ GPa and maximum pre-buckling stress $S_{all} = 270$ MPa) is designed to support a 500 kN compressive load. The cross sections are assumed to be square to maintain geometric similarity postulated

Chapter 2

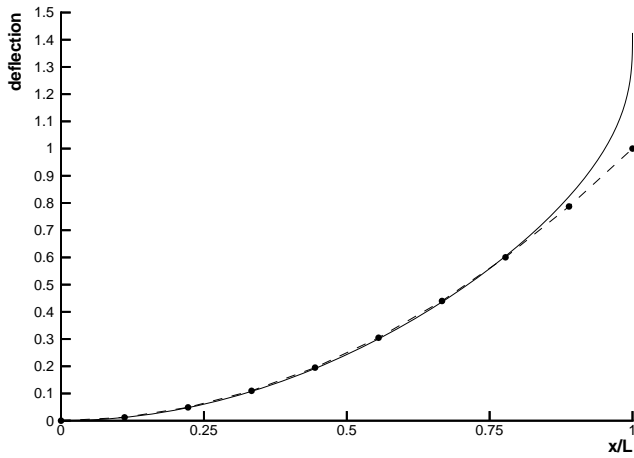


Figure 2.6: Clamped-free column; mode shape for the optimal design. — Analytic solution, - - CA solution for 10 cells, ● NASTRAN simulation of the CA design.

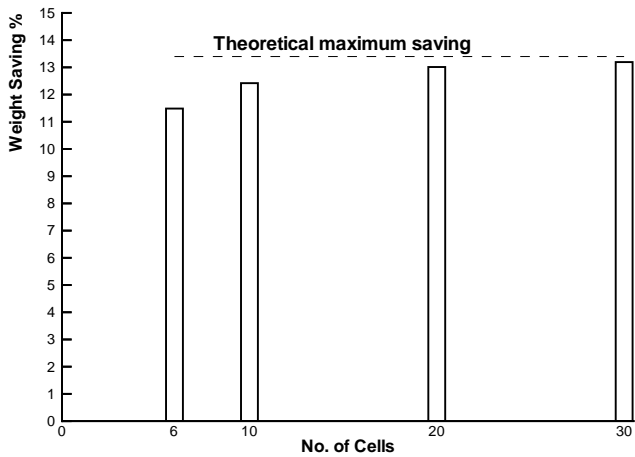


Figure 2.7: Clamped-free column; design improvement vs. number of cells.

in the analytic derivation. Due to symmetry, only half of the beam was discretized

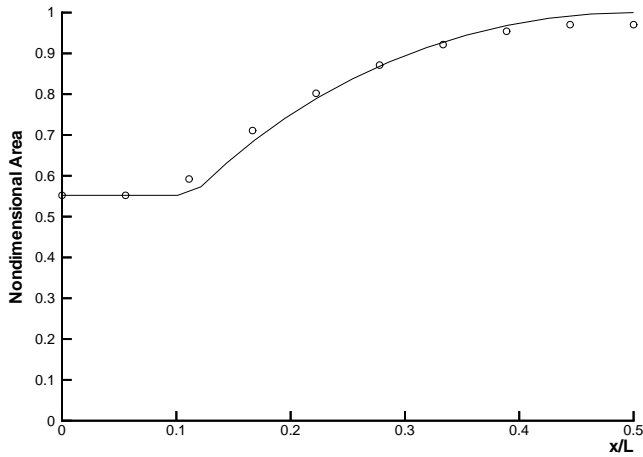


Figure 2.8: simply supported column; area distribution of the optimal column. — Analytic solution, \circ CA solution for 10 cells.

using 10 cells. Figure 2.8 depicts the analytic and CA normalized area distributions. The CA design tends to add more material towards the pinned end and reduce the maximum cross section area below the analytic prediction. This is because of the coarse lattice used. The volume of the CA design is within 0.3% of the analytic optimal solution.

2.4.3 Clamped-Clamped Column

The third example is a clamped-clamped column. This problem was proclaimed solved in [31], but it was found later that this solution actually maximizes the second buckling mode and hence is not optimal. The actual optimum is bimodal as reported in [33], meaning that the first two buckling modes have the same critical value. The column is discretized using 41 cells. The converged CA area distribution is plotted against the exact analytic prediction of [33] in fig. 2.9. The CA design is only 1% heavier than the analytic solution. A finite element analysis of the CA design using NASTRAN, reveals that the critical load for the second buckling mode is only

Chapter 2

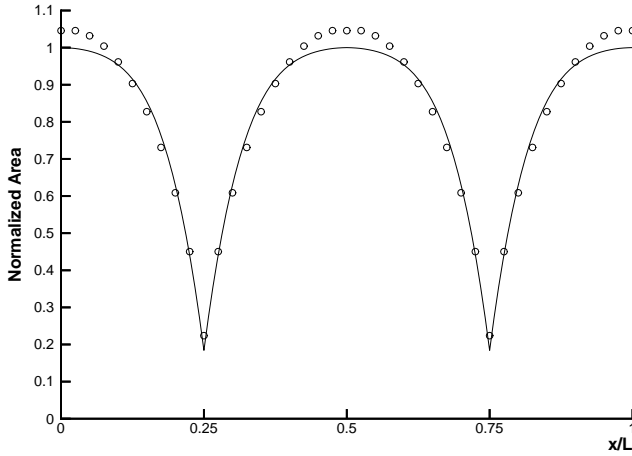


Figure 2.9: clamped-clamped column; area distribution of the optimal column. — Analytic solution, \circ CA solution for 41 cells.

3% higher than the first mode. The CA algorithm cannot handle bimodal optima without modification, since it considers one mode only. For that reason, further refinement of the lattice causes the CA design to deviate from the bimodal optimum, following the symmetric mode (which becomes the second mode rather than the first). However, for practical purposes, the CA design is seen to approximate the bimodal optimum quite well.

2.4.4 Clamped-Free Column with Manufacturing Constraints

All the previous examples assume the cross-sections to be geometrically similar. This artificial restriction is removed in this final example. Consider a clamped-free column of rectangular cross section of height H and width W . The column length is 1 m, and is made of aluminum (Young's modulus $E = 70$ GPa) is designed to support a 500 kN compressive load. The following manufacturing constraints are imposed:

$$H \geq H_{min}, \quad W \geq W_{min}, \quad \text{and} \quad H \leq RW \quad (2.48)$$

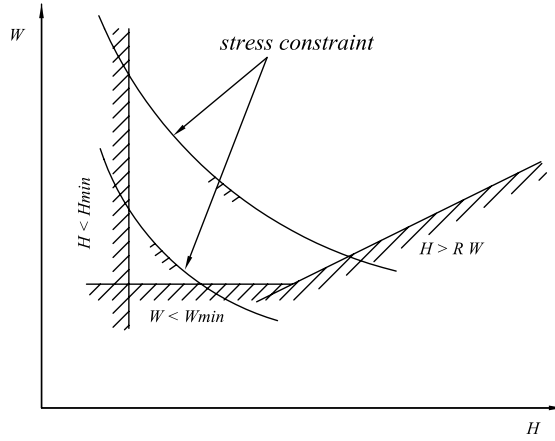


Figure 2.10: Design domain with manufacturing constraints.

where $H_{min} = 5$ cm, $W_{min} = 5$ cm, and the maximum allowable aspect ratio $R = 10$.

The fully stressed condition (2.13) evaluates to one of the following design points (see the sketch in fig. 2.10 for the design domain arrangement for two different cases depending on the value of M_{max}):

$$H = \left(\frac{6 M_{max}}{W_{min} S_{all}} \right)^{1/2}, \quad W = W_{min} \quad (2.49)$$

$$H = \left(\frac{6 R M_{max}}{S_{all}} \right)^{1/3}, \quad W = H/R \quad (2.50)$$

$$W = \left(\frac{6 M_{max}}{H_{min}^2 S_{all}} \right), \quad H = H_{min} \quad (2.51)$$

Another candidate solution is:

$$W = W_{min}, \quad H = H_{min} \quad (2.52)$$

Of the feasible candidate solutions the one with minimum area is chosen.

Since—to the authors' knowledge—no analytic solution exists for this problem, the CA design is compared to the design obtained from traditional finite element based

Chapter 2

software GENESIS [39]. The column is divided into 10 cells for the CA design, and 21 elements for GENESIS linked to only 20 design variables to correspond directly to the CA model. Table 2.1 contains the results of both methods and indicates the active constraint(s). The total volume of the CA design, shown in figure 2.11 is 1640.7cm^3 as compared to 1640cm^3 for GENESIS design. The agreement is satisfactory between the two designs, and they predict the same active constraints.

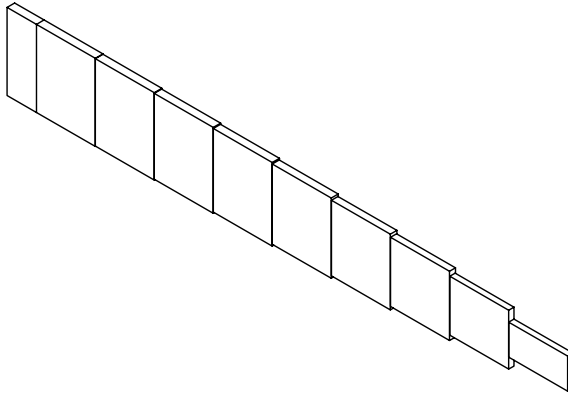


Figure 2.11: Clamped-free column; 3D view of CA design.

2.5 Conclusion

In this chapter, important steps are taken forward towards a CA based structural optimization methodology. An algorithm based on local rules for both analysis and design is proposed to solve structural design problems with eigenvalue requirements. The general algorithm converts the design problem into a repetitive application of a simple design rule based on local stresses. The formulation of the local update rule is in the form of a local minimization problem. The local nature of the algorithm lends it to Cellular Automata (CA) type implementation. The update of the structural displacements is also performed using a local analysis rule based on the minimization of the total potential energy in a cell neighborhood. This approach allows the direct modeling of continuum structures and generalizes easily to other structure types other than beam structures considered here.

Table 2.1: CA vs. GENESIS designs (dimensions in cm).

Cell	CA		GENESIS	
	H	W	H	W
1 [†]	14.402	1.440	14.635	1.468
2 [†]	14.402	1.440	14.687	1.473
3 [†]	14.283	1.428	14.444	1.449
4 [†]	14.040	1.404	14.078	1.412
5 [†]	13.659	1.366	13.605	1.365
6 [†]	13.116	1.312	13.004	1.304
7 [†]	12.364	1.236	12.178	1.221
8 [†]	11.313	1.131	10.959	1.099
9 [‡]	9.631	1.000	9.385	1.000
10 [‡]	5.731	1.000	5.168	1.000

[†]Aspect ratio constraint active.

[‡]Minimum width constraint active.

To demonstrate the ability of the algorithm to handle practical problems, column design for buckling is considered in detail. The numerical examples show that the CA design algorithm converges to the analytic optimum for columns made of geometrically similar cross sections; with and without geometric constraints. A design example with manufacturing constraints is also considered, and the CA design compares very favorably to the design determined through classical optimization coupled to finite element analysis. The proposed algorithm has the benefit, inherent in CA algorithms, of being easily implemented on parallel architectures. Although the algorithm is not yet implemented on parallel architecture, considerable savings can be gained by parallel implementation for large problems.

Chapter 3

Topology Design Using Optimality Based Cellular Automata

3.1 Introduction

In the previous chapter, we developed a CA algorithm to treat design problems for eigenvalue requirements. Buckling critical design of elastic Euler- Bernoulli columns under compressive load for weight minimization was considered. The CA algorithm presented therein converges to the analytic optimal solution for different boundary conditions and geometric constraints. The design formulation was heuristic in following the methodologies of earlier investigations, and depends on the use of stress constraints for cross sectional sizing. Lack of analytic justification of the excellent results obtained using this formulation limits the confidence in the extension of this heuristic algorithm to more complex problems. A major insight comes from the observation that the stress ratio method used for truss design by Gürdal *et al* [6] although advocated there as a method of removing unnecessary material is actually equivalent to the optimality conditions of minimum compliance design of

Chapter 3

trusses. This connection suggests that CA design rules can be successfully obtained by writing the optimality conditions in terms of the local state of stress.

Development of local design rules based on a sound theoretical basis is thus deemed essential for the success of CA application to structural design. In this chapter, the local design rules are formulated on the basis of fundamental optimality criteria using the methods of calculus of variations. The continuous optimality conditions are interpreted as local Kuhn-Tucker conditions. This approach naturally leads to rigorous local design rules that satisfy optimality.

The optimality criteria method was used by many researchers to solve a variety of structural optimization problems. After the first order optimality conditions are obtained in terms of the displacement field, the equations are rearranged to obtain a resizing rule [40, 41]. On the other hand, Zhou and Rozvany [42, 43, 44] developed the COC and the DCOC methods which write the optimality conditions in terms of the stress field not the displacement field. Their derivation is requires the system to be linear and is based on the flexibility method.

Our approach for the derivation of the optimality conditions is based on displacement field as a primary unknown which greatly simplifies the derivation of the optimality conditions and the subsequent numerical implementation. After the optimality conditions are written, the resizing (design) rule is obtained in terms of the local stresses by introducing the Legendre transformation into the optimality conditions. This method is more general than the COC since it works for both linear and nonlinear elastic continua.

Two-dimensional minimum compliance topology design is considered as an application for the optimality based CA design rules. Topology design refers to the determination of regions of a given domain that should comprise the structure, versus the regions that should be left empty of material. This material/void distribution is usually visualized using a black color for material regions, and white color for void regions. Topology design thus aims at finding a black/white distribution of material. When cast as a structural optimization problem, the optimal topology will be driven by the objective and constraints. A popular objective function is the minimization of compliance under applied loads. Usually, the main constraint is a constraint on the total available material volume. Minimizing compliance under a volume constraint is the standard topology optimization problem. Other formu-

lations where the objective is to maximize buckling loads or natural frequencies have also been considered. The black/white topology problem can be thought of as a discrete optimization problem with a given material point assigned a value of 1 and a void point assigned a value of 0. This black/white design problem is ill-conditioned and is thus not computationally solvable. Topology optimization seeks to define well-conditioned problems approximating (also called a regularization of) the black/white problem and devising appropriate numerical solutions techniques.

Since the original work of Bendsøe and Kikuchi [45], many numerical and theoretical approaches to topology optimization were attempted (see Rozvany [46] and the references therein). In almost all these formulations, the design domain is divided into a number of finite elements, and certain design variables are ascribed to each element. The design variables determine element volume and stiffness and the distribution of the optimal design variables is used to interpret the topology. Topologies produced in this manner can contain areas where it can't be interpreted as purely material or purely void regions. These areas are referred to as grey areas. Topology optimization formulations differ in their approach to suppressing these grey areas. Good formulations produce as little grey areas as possible.

Another aspect of topology optimization is numerical stability. Many of the numerical solution procedures suffer from instability problems in the form of checkerboard patterns and mesh dependency [47]. Checkerboards, as the name implies, are regions where adjacent elements alternate being black (fully occupied by material) and white (void). This means that the change in material properties between adjacent elements is oscillatory and severe. Under these conditions, the finite element mesh no longer provide a valid mathematical description of the structure, leading to significant over-prediction of the stiffness of checkerboard regions, thus falsely leading the optimization procedure to generate non-physical solutions. The other problem that has been encountered is mesh dependency. When topologies obtained using successively finer finite element meshes don't show convergence to a solution, but rather the topology keeps acquiring finer and finer new features, the topology is said to be mesh-dependent. Since the mesh size is an arbitrary choice, lack of convergence with mesh refinement calls into question the validity of the obtained topology designs. Avoiding checkerboard patterns and mesh dependency is a major objective of a good topology design algorithm,

In this chapter, we start by formulating the minimum compliance design problem in

Chapter 3

generic terms. Variational calculus is then used to derive the optimality conditions which are written in terms of the local state of stress. The optimality criteria for topology design are developed by specializing the optimality conditions for general minimum compliance design to the Simple Isotropic Microstructure with Penalization (SIMP) [48] topology design formulation, and closed form analytic expressions for the optimal material distribution are derived in terms of local strain energy and a constant Lagrange multiplier. The Lagrange multiplier is associated with the volume constraints and shows as a global quantity (not dependent on the position inside the design domain). Two approaches to the treatment of the volume constraint and the Lagrange multiplier are proposed.

Once the theoretical foundations are expounded, we turn our attention to numerical aspects. The CA lattice and neighborhood are described, and the cell variables are defined. CA analysis update rules for two-dimensional linearly elastic continuum are derived from the principle of minimum total potential energy. These analysis update rules are equivalent to the set of linear equations generated by a finite element mesh corresponding to the CA lattice (cells corresponding to FEM nodes). In this fashion, the excellent convergence properties of the finite element method in structural problems are maintained. The main difference between the CA rules as an analysis tool and finite elements is thus seen to be in how the solution is obtained not in the system of equations to be solved. CA obtains the solution iteratively and using limited interaction between cells, while the corresponding finite element solution would usually involve matrix factorization where the interaction between cells (nodes) extends across the solution domain. As explained before, there have been difference on whether to use CA for both analysis and design [6], or using CA for design only and finite elements for analysis [21]. Here, we develop both methodologies and do some limited numerical experiments using both.

The CA design rule is obtained by averaging the strain energy in the neighborhood and substituting this value into the analytic solution of the optimality criteria. The CA design rule presented avoids the need for sensitivity analysis or the solution of large-scale mathematical programming problems. The CA design rules will also be shown to produce checkerboard-free mesh-independent optimal topologies, thus eliminating in a natural and elegant way the need for special procedures such as sensitivity filtering [49]. in addition to the stabilizing effect of averaging the strain energy over a neighborhood [49], which is known to eliminate checkerboard patterns, further stabilization and mesh independency is obtained by associating topology

variables not with elements but with cells (nodes). Moreover, in calculating strain energy a new interpolation scheme for the topology variables called compliance averaging is introduced. The combined effect of compliance and strain energy averaging is a robust and stable CA topology optimization procedure. The robustness of the CA topology algorithm is explored through a number of test cases, including a Mitchell truss problem. The CA algorithm produces crisp black/white topologies which are free of checkerboard patterns and insensitive to mesh variations. The presentation given here expands on the previous work by the author [50].

3.2 Formulation of Minimum Compliance Design

Minimum compliance design attempts to find the optimal distribution of material in a given domain to minimize the compliance of the structure under given loads with constraints on material availability. The distribution of the material throughout the domain is described by certain design functions $\mathbf{b}(\mathbf{x})$, that determine the local stiffness of the material and the local use of resources. An example is the topology design of variable stiffness panels considered in [], where the local design functions, $\mathbf{b}(\mathbf{x})$, are the local material density and local fiber angle. Material availability becomes a constraint on functionals of the design functions. Thus, variational calculus can be used to find the optimality conditions.

The compliance of the structure is measured by the complementary work done by the external loads W_c , which can be related to the total potential energy at equilibrium, Π_0 by,

$$W_c = -\Pi_0, \quad (3.1)$$

Thus the optimization problem is to find the distribution of design functions $\mathbf{b}(\mathbf{x})$ to minimize W_c , thus,

$$\min_{\mathbf{b}} (-\Pi_0), \quad (3.2)$$

We also assume that the design functions $\mathbf{b}(\mathbf{x})$ have to satisfy local and integral constraints. Integral constraints define constraints on global resources, such as maximum material volume, while local constraints represent limits on available local resources, such as maximum possible thickness.

Chapter 3

The local point by point constraints take the form,

$$\mathbf{g}(\mathbf{x}, \mathbf{b}(\mathbf{x})) \leq 0, \quad (3.3)$$

and the integral constraints take the form,

$$\int_{\Omega} [\mathbf{f}(\mathbf{b}) - \mathbf{f}_0] d\Omega \leq 0. \quad (3.4)$$

The functions \mathbf{f} represent the local contribution to global cost (e.g., local material volume that adds up to total material volume of the structure). The constants, \mathbf{f}_0 , represent a measure of the maximum available resources.

The equilibrium value of the total potential energy can be obtained by minimizing the total potential over kinematically admissible displacement fields $\mathbf{u}(\mathbf{x})$,

$$\Pi_0 = \min_{\mathbf{u}} \Pi, \quad (3.5)$$

where the total potential of the structure, ignoring body forces, is defined by,

$$\Pi = \int_{\Omega} \Phi(\mathbf{x}, \boldsymbol{\gamma}; \mathbf{b}) d\Omega - \int_{\Gamma_1} \mathbf{t} \cdot \mathbf{u} \partial\Omega, \quad (3.6)$$

where \mathbf{t} is the applied surface traction, and Φ is the strain energy density of the structure, and $\boldsymbol{\gamma}(\mathbf{x}, \mathbf{u})$ is the generalized strain vector. The strain energy density Φ is parameterized by the design functions $\mathbf{b}(\mathbf{x})$.

Since the minimization of the total potential with respect to the displacement field (3.5) will reduce to the equilibrium equations, and since all the integral (3.4) and side point constraints (3.3) are not functions of the displacements, we can derive the optimality criterion by combining (3.2) and (3.5), and restricting ourselves to variations in the design functions, to obtain,

$$\min_{\mathbf{b}} -\Pi, \quad (3.7)$$

The Lagrangian in this case can be written as,

$$\mathcal{L} = \int_{\Omega} \left[-\Phi + \boldsymbol{\mu} \cdot (\mathbf{f} - \mathbf{f}_0 + \mathbf{c}^2) + \boldsymbol{\lambda} \cdot (\mathbf{g} + \mathbf{s}^2) \right] d\Omega, \quad (3.8)$$

where $\boldsymbol{\lambda}(\mathbf{x})$ is a vector of Lagrange multiplier functions associated with the point constraints (3.3), $\mathbf{s}(\mathbf{x})$ is the corresponding vector of slack functions, $\boldsymbol{\mu}$ is a constant vector of Lagrange multipliers associated with the integral constraints (3.4), and \mathbf{c} is the corresponding constant vector of slack variables.

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

1. stationarity

$$-\frac{\partial \Phi}{\partial \mathbf{b}} + \boldsymbol{\mu} \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{b}} + \boldsymbol{\lambda} \cdot \frac{\partial \mathbf{g}}{\partial \mathbf{b}} = 0. \quad (3.9)$$

2. switching conditions

$$\lambda_i(\mathbf{x}) s_i(\mathbf{x}) = 0, \quad (3.10)$$

$$\mu_i c_i = 0. \quad (3.11)$$

3. non-negativity

$$\lambda_i \geq 0, \quad (3.12)$$

$$\mu_i \geq 0, \quad (3.13)$$

in addition to the integral, (3.4), and side, (3.3), constraints.

We introduce the generalized stresses $\boldsymbol{\sigma}$, defined by,

$$\boldsymbol{\sigma} = \frac{\partial \Phi}{\partial \boldsymbol{\gamma}}, \quad (3.14)$$

and we define the complementary energy density $\hat{\Phi}(\mathbf{x}, \boldsymbol{\sigma}; \mathbf{b})$ by the Legendre transformation,

$$\hat{\Phi}(\mathbf{x}, \boldsymbol{\sigma}; \mathbf{b}) = \boldsymbol{\sigma} \cdot \boldsymbol{\gamma} - \Phi(\mathbf{x}, \boldsymbol{\gamma}; \mathbf{b}), \quad (3.15)$$

It can be shown from the definition of the generalized stress, (3.14), and the Legendre transformation,(3.15), that,

$$\left. \frac{\partial \hat{\Phi}}{\partial \mathbf{b}} \right|_{\boldsymbol{\sigma}} = - \left. \frac{\partial \Phi}{\partial \mathbf{b}} \right|_{\boldsymbol{\gamma}}. \quad (3.16)$$

Chapter 3

This equation represents the fact that maximization of the strain energy density for a given strain (displacement field) is equivalent to the minimization of the complementary energy at constant stress.

Using (3.16), we can show that the first order conditions (3.9), (3.10), and (3.12) are equivalent to those of the following pointwise minimization problem,

$$\min_{\mathbf{b}} \hat{\Phi}_{(\mathbf{x}, \boldsymbol{\sigma})} + \boldsymbol{\mu} \cdot \mathbf{f}_{\mathbf{x}}, \quad (3.17)$$

subject to,

$$\mathbf{g}_{\mathbf{x}}(\mathbf{b}) \leq 0, \quad (3.18)$$

where the subscripts under different functions indicate variables that are held constant. Thus, the minimum compliance problem for the structure is reduced to a local mathematical optimization problem. This local optimization problem can be either solved analytically, or using standard numerical techniques [4]. Note, however, that the local optimization problem presented above (3.18) contains an unknown vector of Lagrange multipliers associated with the global integral constraints. Their values, therefore, are dependent on the global integral constraints requiring a scheme for their computation, which is described in the following section.

3.2.1 The Lagrange Multipliers

The Lagrange multipliers associated with the integral constraints, $\boldsymbol{\mu}$, are obtained by solving the active integral constraints of (3.4). The Lagrange multipliers associated with the inactive integral constraints will be zero. In this fashion, the minimum compliance design problem is split into a set of local update rules (3.18), and a global iteration to obtain the Lagrange multipliers. The philosophy of the solution is simple. The iterations start by an assumed set of design functions and Lagrange multipliers. The corresponding displacement field that satisfies equilibrium equations is obtained, then the local optimization problem can be set-up and solved point by point. This leads to an update of the design functions. Since the displacement field is no longer compatible with the design functions, equilibrium is no longer satisfied, therefore, the displacement field needs to be updated. The Lagrange multipliers also need to be updated to satisfy the integral constraints. Thus, two iterative processes can be identified, the first to converge the design and the displacement

field to satisfy both optimality and equilibrium, and the second to converge the Lagrange multipliers to satisfy integral constraints.

The two iterative processes identified above can be nested in two different ways. The first approach is illustrated in the flowchart in fig. 3.1, where the displacements are updated in the outer loop, while the Lagrange multipliers are updated in the inner loop. For each determination of the displacement field, for a given set of design functions, the design is updated over the whole domain, and then the integral constraints are checked. If the integral constraints are not satisfied, the Lagrange multipliers are updated and the design is recalculated in the inner loop. The process continues until all integral constraints are satisfied. This leads to an updated design which is then used to update the displacement field in the outer loop. The outer loop is repeated until the design is converged.

The second approach to nesting the iterations is illustrated in the flowchart in fig. 3.2. In this approach, the Lagrange multipliers are chosen initially, then the inner loop comprises a repeated update of the displacement field and the design until the design converges. This converged design will not necessarily satisfy the integral constraints. To satisfy the integral constraints, the Lagrange multipliers are updated in an outer loop, and the inner loop repeated, until the integral constraints are satisfied.

It might be argued that the second approach is inherently wasteful, since fully converged designs are obtained again and again that do not satisfy the integral constraints. These designs, however, are physically meaningful. The integral constraints limit the availability of global resources. It is often interesting to investigate the optimal design as the amount available resources are parametrically varied. Each of the designs obtained in the inner loop in fig. 3.2 represents a valid optimum solution for some value of the available resources. By varying the Lagrange multipliers, the compromise between available resources and performance (mean compliance) can be investigated. This approach has been successfully used in [51] where the effect of available volume on the average compliance was studied for the combined topology and fiber angle design of composite panels.

Chapter 3

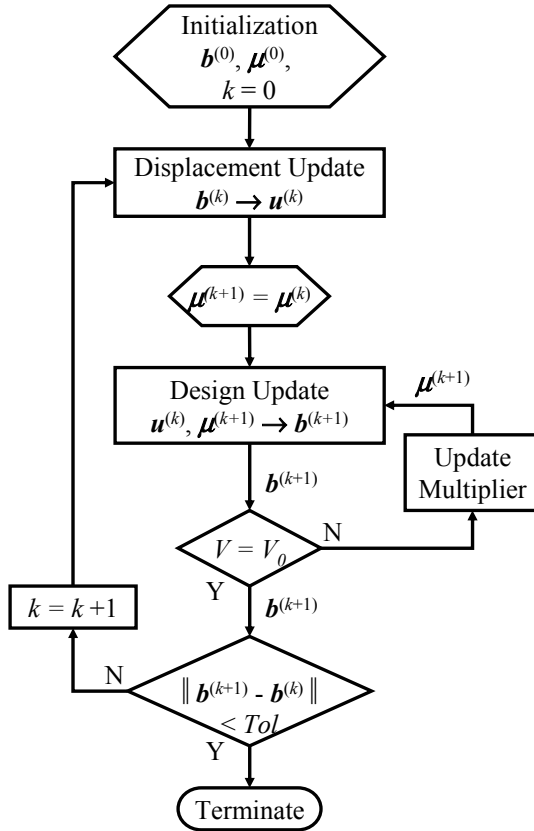


Figure 3.1: Nesting of the design iterations: displacement update in the outer loop, Lagrange multiplier update in the inner loop.

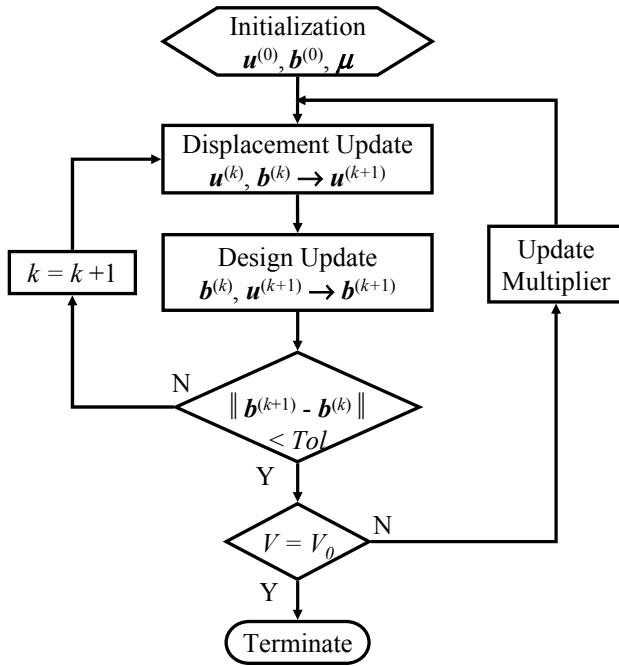


Figure 3.2: Nesting of the design iterations: displacement update in the inner loop, Lagrange multiplier update in the outer loop.

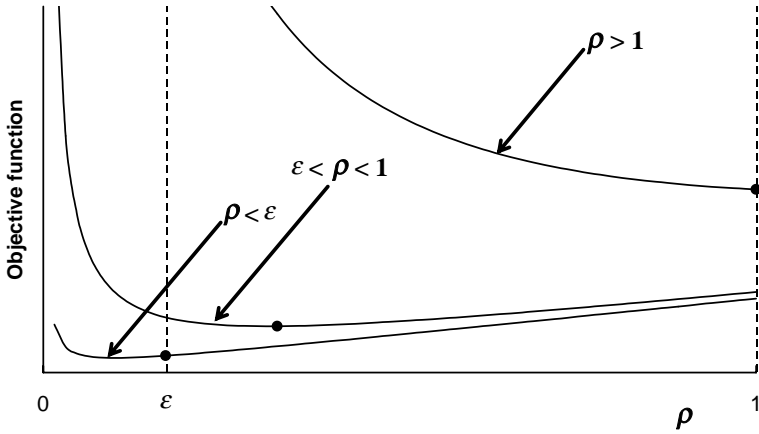


Figure 3.3: Local optimization problem.

3.3 Topology Design

The general design formulation for minimum compliance design of continuum structures developed in the previous section is specialized for the case of topology design of two-dimensional elastic continuum of constant thickness (i.e., a uniform plate). Instead of solving the discrete topology design problem, artificial density variables are introduced to determine the local contribution of the material to stiffness. This approach converts the discrete topology problem into a continuous sizing problem for which the analytic framework derived earlier applies. The specialization of the formulation requires two simple steps. The design functions vector $\mathbf{b}(\mathbf{x})$ is defined, and the dependence of global integral resources (3.4), local resources (3.3), and strain energy density on the design functions is specified. Once these relations are defined, the local optimization problem can be explicitly formulated, and in many cases, solved analytically.

To simplify the presentation, we adopt notation from plate theory. We assume all the loads to be acting in the plane of the plate, so that no bending deformations are induced, and neglect body forces. We further assume that the material behavior is linearly elastic, and obeys plane stress equations,

$$\mathbf{N} = \bar{\mathbf{Q}} \cdot \boldsymbol{\gamma}, \quad (3.19)$$

where \mathbf{N} is the vector of inplane stress resultants, and $\bar{\mathbf{Q}}$ is the reduced inplane stiffness matrix.

In the present approach, fictitious density distribution ($0 < \rho < 1$) is used as the design function. For topology optimization using SIMP, it is assumed that the local stiffness of the structure is a function of a fictitious density $\rho(\mathbf{x})$, and the strain energy density, $\bar{\Phi}$, is parameterized as,

$$2\bar{\Phi} = \rho^p \boldsymbol{\gamma} \cdot \bar{\mathbf{Q}} \cdot \boldsymbol{\gamma}, \quad (3.20)$$

For $\rho \equiv 1$, (3.20) reduces to the usual strain energy density for a plate governed by (3.19). For $\rho = 0$, the strain energy density vanishes, signifying a void region. The exponent p is chosen high enough so that intermediate densities are penalized, so that the final distribution of $\rho(\mathbf{x})$ will consist almost entirely of black ($\rho \approx 1$), and white ($\rho \approx 0$) regions; typically $p \geq 3$ is used.

With these definitions, the generalized stresses $\boldsymbol{\sigma}$, are obtained using (3.14) as,

$$\boldsymbol{\sigma} = \rho^p \mathbf{N}, \quad (3.21)$$

and then the complementary energy $\hat{\Phi}$ is obtained using (3.15) as,

$$\hat{\Phi} = \frac{\bar{\Phi} \boldsymbol{\sigma}}{\rho^p}, \quad (3.22)$$

where

$$2\bar{\Phi} \boldsymbol{\sigma} = \boldsymbol{\sigma} \cdot \bar{\mathbf{Q}}^{-1} \cdot \boldsymbol{\sigma}. \quad (3.23)$$

The total volume of the material, V , is limited to a fraction η of the total volume of the domain, V_0 . This constraint can be expressed in the standard form (3.4) as an integral constraint,

$$\int_{\Omega} (\rho - \eta) d\Omega \leq 0 \quad (3.24)$$

Thus the local optimization problem (3.18) reduces to a one dimensional minimization problem over the fictitious density ρ , and takes the form,

$$\min_{\rho} \frac{\bar{\Phi}^{(k)} \boldsymbol{\sigma}}{\rho^p} + \mu \rho, \quad (3.25)$$

Chapter 3

subject to,

$$\epsilon \leq \rho \leq 1, \quad (3.26)$$

where ϵ is a small number used to avoid numerical ill conditioning, and k is used to signify that $\bar{\Phi}_\sigma$ is calculated based on the design variables in the previous iteration.

The topology one-dimensional problem (3.25) is convex. The solution can be obtained analytically by setting the derivative of the function to zero. The value at which the gradient is zero is denoted by $\hat{\rho}$ and is given by,

$$\hat{\rho} = \left(\frac{\bar{\Phi}_\sigma}{\bar{\mu}} \right)^{\frac{1}{p+1}}, \quad (3.27)$$

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

Depending on the value of $\hat{\rho}$, the solution of the local topology optimization problem is given by one of three values to enforce the bounds on density (3.26),

$$\rho^{(k+1)} = \begin{cases} \hat{\rho} & \epsilon < \hat{\rho} < 1 \\ \epsilon & \hat{\rho} < \epsilon \\ 1 & \hat{\rho} > 1 \end{cases} \quad (3.28)$$

This is shown in Fig. 3.3, where the minimum is shown graphically for three different values of the parameter $\hat{\rho}$.

The Lagrange multiplier $\bar{\mu}$ can be loosely interpreted as an average strain energy density in the structure. So, instead of pre-specifying the volume fraction η , $\bar{\mu}$ may be specified as input. This eliminates the need to iteratively determine the Lagrange multiplier to satisfy the volume constraint. This corresponds to the iterative scheme in fig. 3.2. Alternatively, the approach in fig. 3.2 can be used. This point will be further discussed when the numerical implementation is described in Sec. 1.4.

3.4 CA Implementation

In the previous section, optimality based local rules for updating the material density were derived. We now turn our attention to the CA discretization of a two-dimensional structural domain. A generic two-dimensional topology design problem is depicted in Fig. 3.4-a. We use a lattice of regularly spaced cells with the same

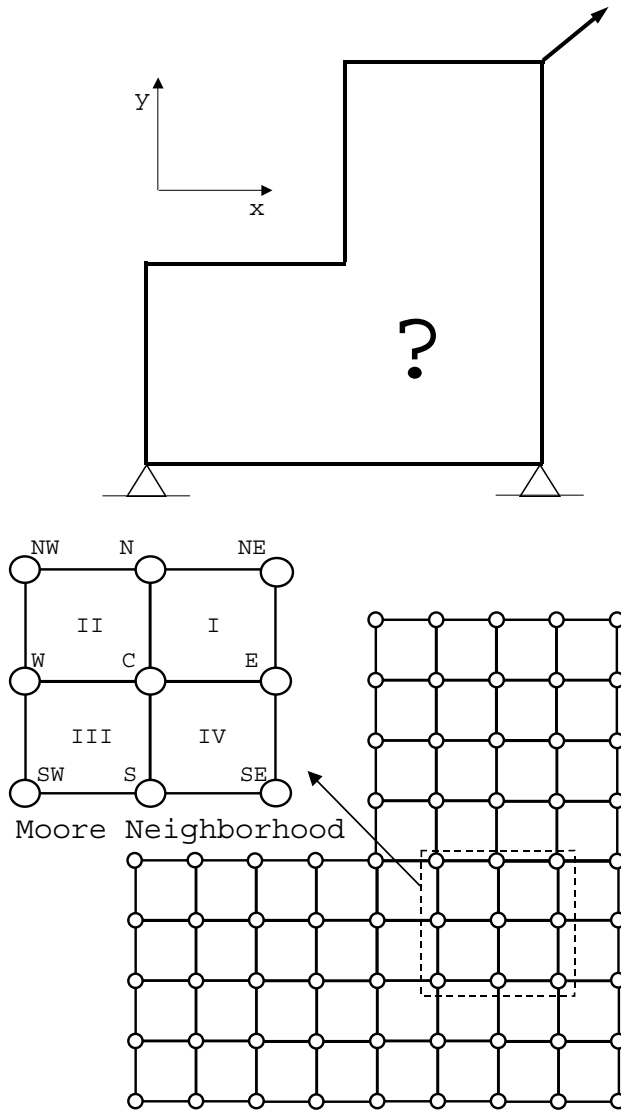


Figure 3.4: a) Sample domain for topology design, b) CA lattice and Moore neighborhood.

Chapter 3

spacing h in both x and y directions. Traditional Moore neighborhood is used to define the connectivity of the lattice as shown in the inset of Fig. 3.4.b. Each cell (C) has eight neighboring cells ($N, S, E, W, NW, NE, SW, SE$), and the neighborhood is split into four quadrants. The state of a cell i is denoted as ϕ_i^k where k is the iteration number. For topology design in two dimensions we define, similar to [6], a generic cell as,

$$\phi_i = \{(u_i, v_i), (f_{x_i}, f_{y_i}), \rho_i\} \quad (3.29)$$

where u_i and v_i are the cell displacements in x - and y - directions, respectively, and f_{x_i} and f_{y_i} are the external forces acting on the i -th cell in the respective directions. Note that each cell has its own density measure ρ_i at the cell point independent of the thickness of the quadrants that are used to define the neighborhood.

For isotropic materials, the reduced inplane stiffness can be written in terms of Young's modulus E , Poisson's ratio ν , and the plate thickness t as:

$$\bar{Q} = \frac{Et}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix}. \quad (3.30)$$

We restrict our attention to linear small deformation problems. The small-strain tensor is given by:

$$\gamma = (\varepsilon_x, \varepsilon_y, \varepsilon_{xy}) \quad (3.31)$$

where

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \\ \varepsilon_{xy} &= \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \end{aligned} \quad (3.32)$$

The strain energy in the cell neighborhood is calculated by summing the contribution of each quadrant. To simplify the strain energy calculations, the thickness of each quadrant is assumed to be constant. The thickness is determined by finding an average value $\bar{\rho}$ of the four cells sharing the quadrant. Recalling the definition of SIMP, (3.20), the strain energy in a quadrant can be written in terms of the strain energy of the base material as,

$$U_q = \frac{\bar{\rho}^p}{2} \int_{quadrant} \gamma \cdot \bar{Q} \cdot \gamma \, dx dy \quad (3.33)$$

is expressed as a quadratic form in cell displacements. The coefficients of this quadratic form are functions of material parameters and the average density $\bar{\rho}$. The value of the average density, $\bar{\rho}$, for each quadrant is determined through a new interpolation scheme as described below.

3.4.1 Compliance averaging interpolation scheme

Each quadrant is assumed to have a constant density $\bar{\rho}$ given by:

$$\frac{1}{\bar{\rho}^p} = \frac{1}{N_{cell}} \sum_{cells} \frac{1}{\rho_i^p} \quad (3.34)$$

where ρ_i 's are the density measures of the four cells surrounding the quadrant, and N_{cell} is the number of cells defining the quadrant. For the chosen neighborhood structure $N_{cell} = 4$. We note that since the compliance is inversely proportional to ρ^p (which measures the local stiffness), this scheme effectively assigns each quadrant an average value of the compliance at the four cells.

This compliance averaging interpolation scheme is chosen so that any cell with a density measure below the threshold value ϵ (or zero) would turn-off (force the assigned density to zero) all four quadrants in which that cell participates. This makes cells in white (void) regions have a negligible (or no) effect on the equilibrium equations of cells in the black regions. Another interesting aspect of this approach is that by smoothly interpolating cell densities, checkerboard patterns are automatically suppressed [52, 53].

3.4.2 Displacement update

On each quadrant (*I* to *IV*), each displacement component is expressed in terms of cell values using bilinear interpolation. The approximate equilibrium equations are found by minimizing the total potential energy over the cell neighborhood with respect to cell displacements, this gives the following form of equilibrium equations:

$$E \begin{bmatrix} A & B \\ B & A \end{bmatrix} \cdot \begin{Bmatrix} u_C \\ v_C \end{Bmatrix} = \begin{Bmatrix} f_{x_C} + f_x^e \\ f_{y_C} + f_y^e \end{Bmatrix} \quad (3.35)$$

Chapter 3

where A and B are parameters that depend on Poisson's ratio and quadrant average density. f_x^e and f_y^e are elastic forces that depend on average densities and material parameters and linearly on neighbor displacements. These forces arise because neighbor displacements are assumed to be fixed at their values in the previous iteration, while cell displacements and/or forces are updated to restore equilibrium during the new iteration. Closed form expressions for these quantities are generated using *MATHEMATICA*TM. This 2×2 system of equations is solved for cell displacements, cell forces or a mixture of both according to the type of displacement boundary conditions, or the lack thereof, at the cell.

3.4.3 Design update

Cell densities are updated using (3.28). To this end, the cell strain energy density $\hat{\Phi}_e$ is calculated by averaging over cell neighborhood. Since the CA algorithm should handle irregular domains, some cells will have shadow neighbors that lie outside the computational domain. Shadow cells are treated by setting their density ρ to zero. Due to the density interpolation scheme (3.34), these cells automatically decouple from the solution. The area of the quadrants corresponding to shadow cells is not considered in averaging. In summary, we have:

$$\Phi_\sigma = \frac{1}{n h^2} \sum_I^{IV} \bar{\rho}_i^{2p} \tilde{U}_i \quad (3.36)$$

where n is the number of quadrants with nonzero density.

3.4.4 Cell update scheme

The CA analysis and design rules are applied at each cell, keeping the values of other cells fixed. Cells can be either updated *simultaneously* which corresponds to a Jacobi iteration, in which case:

$$\phi_C^{k+1} = f(\phi_C^k, \phi_N^k, \phi_S^k, \phi_E^k, \phi_W^k, \phi_{NW}^k, \phi_{NE}^k, \phi_{SW}^k, \phi_{SE}^k) \quad (3.37)$$

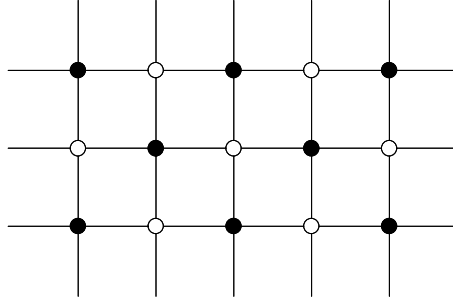


Figure 3.5: Checkerboard ordering for Gauss-Seidel iteration.

or *sequentially*, where the cell is updated using the latest information from its neighbors, which corresponds to a Gauss-Seidel iteration. In order to preserve the symmetry of solutions (when the domain and the loading are symmetric), a variant of the Gauss-Seidel method is used where the cells are updated in a checkerboard ordering (see fig. 3.5). The update rule for *black* cells takes the form:

$$\phi_C^{k+1} = f(\phi_C^k, \phi_N^k, \phi_S^k, \phi_E^k, \phi_W^k, \phi_{NW}^k, \phi_{NE}^k, \phi_{SW}^k, \phi_{SE}^k) \quad (3.38)$$

and for *white* cells:

$$\phi_C^{k+1} = f(\phi_C^k, \phi_N^k, \phi_S^k, \phi_E^k, \phi_W^k, \phi_{NW}^{k+1}, \phi_{NE}^{k+1}, \phi_{SW}^{k+1}, \phi_{SE}^{k+1}) \quad (3.39)$$

The Gauss-Seidel method with checkerboard ordering is used for the analysis update. This is done to reinforce symmetry in symmetric problems. For the design update, the Jacobi method is used.

In previous work on CA [6, 29], the analysis and design are nested. A fixed number of analysis update N_a is performed followed by a design update. The design changes were damped to prevent divergence of the nonlinear iteration. Fixing the number of analysis updates is not completely satisfactory, since the convergence rate of the Gauss-Seidel method deteriorates significantly as the total number of cells is increased [54].

In this work, the analysis and design are likewise nested, but instead of using a fixed number of repetitions of the analysis update, the analysis updates are performed until the norm of the residual forces (unbalanced equilibrium) reaches a pre-specified

Chapter 3

tolerance. Instead of damping design changes [6, 29], a minimum value of cell density ρ_{min} is used as,

$$\rho_{min} = \max \{ \varepsilon, (1 - \alpha) \rho^k \} \quad (3.40)$$

where α is a prescribed move limit. The overall iteration is terminated when the maximum change in cell densities is less than a pre-specified tolerance.

From a computational perspective, the attractive feature of CA is its inherent parallelism. When this parallelism is not fully exploited, CA algorithms can be quite slow to converge, especially for the analysis update. The communication between cells is limited only to immediate neighbors. The information from the cells where the loads are applied has to travel by neighbor-to-neighbor interaction throughout the domain. As the lattice is refined, the number of lattice updates needed to reach equilibrium significantly increases manifesting the deterioration in the rate of convergence alluded to above. Thus, when CA is implemented on a serial machine it loses its most attractive feature as far as the analysis update is concerned. The design features of CA, though, remain effective. For this reason, it is important to investigate use of finite element analysis for the analysis update, while using CA for the design update.

3.4.5 Updating the Lagrange Multiplier

When the volume fraction η is pre-specified, the modified Lagrange multiplier $\bar{\mu}$ needs to be determined. This corresponds to the approach in fig. 3.1. The update rule for the Lagrange multiplier can be obtained by applying Newton's method, leading to a very simple update rule. The volume constraint, (3.24), is approximated as:

$$\sum_{lattice} \bar{\rho}_e V_e - \eta \sum_{lattice} V_e = 0 \quad (3.41)$$

where V_e is the volume of a quadrant.

The quadrant average density is given by (3.34), whereas the cell densities are calculated based on (3.28). Since the average strain energy in each quadrant is calculated based on the values of the cell density of the previous design iteration, the quadrant density can change only due to changes in the modified Lagrange multiplier $\bar{\mu}$. In

this fashion, the volume constraint in equation (3.41) is considered as a nonlinear equation to be satisfied by finding the proper value of the modified Lagrange multiplier. The solution to this equation can be conveniently obtained by applying the Newton-Raphson method. The derivative of (3.41) with respect to $\bar{\mu}$ is obtained by applying the chain rule to (3.34) and (3.28).

After algebraic manipulation, the update rule of the Lagrange multiplier is simplified to,

$$\bar{\mu}^{k+1} = \bar{\mu}^k (1 + (p + 1) \xi) \quad (3.42)$$

where,

$$\xi = 1 - \eta \frac{\sum_{lattice} V_e}{\sum_{lattice} \bar{\rho}_e V_e} \quad (3.43)$$

For the uniform lattice used herein, V_e is simply the area of any of the rectangular quadrants h^2 .

3.5 Results

The CA based iterative local analysis and design update formulation described in the preceding sections was implemented using a Fortran90 code, while a MATLABTM code was developed for the finite element analysis based approach. Both approaches are tested on a single processor Pentium III machine. Example results for both analysis and design are provided to illustrate the possibilities of the CA combined analysis and design.

3.5.1 CA for both analysis and design

We consider solving topology design problems using CA rules for analysis and design. As a demonstration, we consider the design of a cantilever. Geometry of a symmetric cantilever beam like domain is depicted in Figure 3.6. Poisson's ratio is taken as 0.3 for all subsequent examples. The aspect ratio of the domain is 4,

Chapter 3

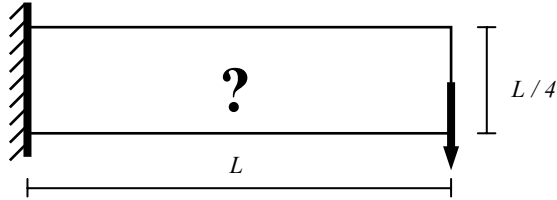


Figure 3.6: Symmetric cantilever problem.

and the volume fraction is set to 0.5. The minimum density ε is set to 10^{-3} , and the penalization parameter is chosen as $p = 3$. Since the analysis is linearly elastic, Young's modulus, sheet thickness and load value do not influence the optimal topology.

The first step is to investigate the convergence of the analysis rule. The problem was run on analysis only mode on a 81×21 cells lattice and assuming the density of all cells $\rho_i = 1$ to study the analysis convergence behavior. The convergence history is shown in fig. 3.7. It is clear that the CA iteration initially achieves considerable reduction in the norm of residual (unbalanced cell forces) and then the convergence rate deteriorates. Also, it is noteworthy to see that a very large number of iterations is required for convergence.

In light of the slow convergence of the analysis update, it is quite inefficient to converge the displacements completely before applying the design rule, especially given that after the design is updated the converged displacements will be no longer correct, and will need to be updated. For this reason, the first nesting approach of fig. 3.1 is not efficient when CA analysis rule is used for finding the displacements. For this case, the better alternative is to follow the nesting approach of fig. 3.2 where the Lagrange multiplier is kept fixed. The displacement field is only partially converged before the design is updated. To stabilize the iteration a tight move limit is applied to the design variables. In this fashion, both the displacement field and the design will evolve towards a solution that satisfies both equilibrium and optimality.

The combined analysis and design algorithm was run with a move limit $\alpha = 5\%$ in equation (3.40) for the cantilever design problem. The Lagrange multiplier was adjusted to produce a volume fraction of approximately 0.5. The algorithm converged in a total of 294,414 analysis updates. The run time on a Pentium III machine is

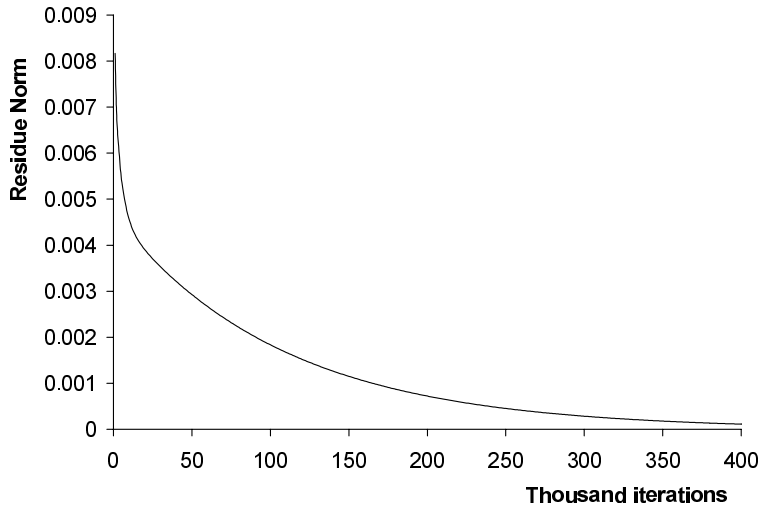
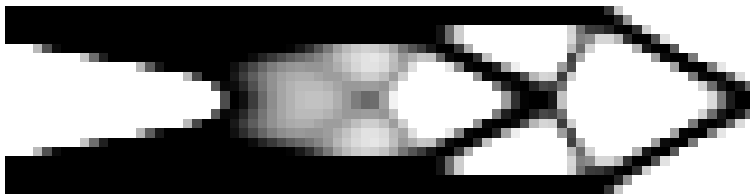
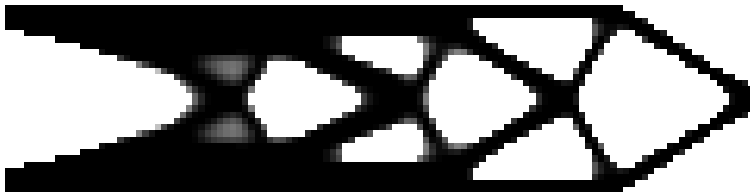


Figure 3.7: Convergence of CA for analysis and design.



(a) CA analysis



(b) FEM analysis

Figure 3.8: Converged optimal topology using the CA topology algorithm.

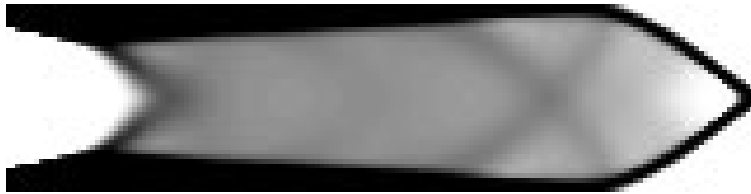
about 908 seconds. The optimal topology is shown in fig. 3.8-a attains a minimum nondimensional compliance of 440.5.

3.5.2 CA for design combined with FEM analysis

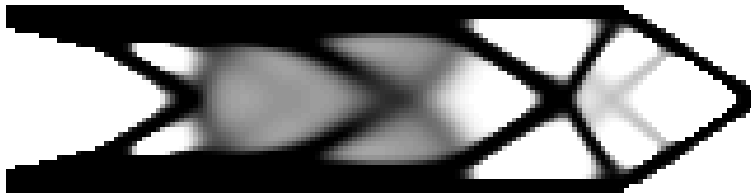
As discussed earlier, the CA analysis update is slow on a serial machine because of the large number of cell updates needed. It is instructive to investigate the case where the CA analysis update is replaced by a direct finite element solution. The same symmetric cantilever problem solved earlier is re-solved using FEM for the analysis for the same cell density. Since for finite element analysis, the displacement update is costly and exact, it is more efficient to use the nesting approach of fig. 3.1. The value of the Lagrange multiplier was calculated at each re-design step to satisfy the volume constraint within 10^{-3} . The optimal topology, shown in fig. 3.8-b, has a nondimensional compliance of 410.6.

A close look at fig. 3.8 will reveal that both solutions are qualitatively similar. In both cases the cantilever is given a *sandwich structure* where most of the material is at the top and bottom region acting similar to *face sheets*. This is natural since the cantilever is primarily resisting bending loads. The thickness of the face sheets increases as the cantilever root is approached. The rest of the material is distributed to form a truss-like core which efficiently resists the shear loading.

Nevertheless, the agreement is not complete. The CA analysis approach seems to generate more straight features and more grey areas, while the FEM analysis seems to produce more curved boundaries and less grey areas. The difference is also reflected in about 7% difference in compliance between the two designs. It is well known that the topology optimization problem with $p > 1$ is not convex and multiple optima exist [47]. Since the CA analysis solution and the FEM analysis solution use different strategies for updating the Lagrange multiplier, they ended converging to different optima.



(a) after 20 iterations.



(b) after 40 iterations.



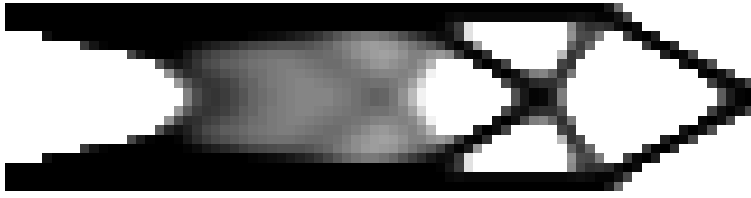
(c) after 60 iterations.



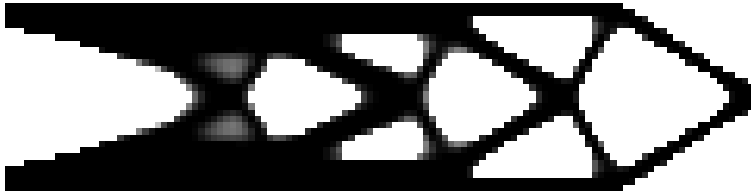
(d) after 80 iterations.

Figure 3.9: Evolution of cantilever topology (161×41 cells).

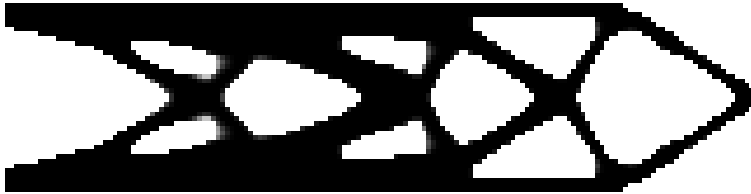
Chapter 3



(a) 81×21 cells



(b) 121×31 cells



(c) 161×41 cells

Figure 3.10: Cantilever topology after 100 design iterations ($p = 3$).

(a) 161×41 cells, $p = 4$ (b) 161×41 cells, $p = 5$

Figure 3.11: Effect of penalization parameter on converged cantilever topology.

3.5.3 Symmetric cantilever: Mesh Independency

The symmetric cantilever designs in fig. 3.8 show some grey areas, most notably the design based on FEM analysis. It is important to investigate the behavior of the design as the mesh is refined and how the design evolves. In particular, it is important to investigate whether grey areas are inherent in the designs obtained using the CA algorithm or incidental to the particular lattice density used. Because the CA analysis is not efficient on serial machines especially for fine meshes, we restrict ourselves to FEM analysis results for the rest of the chapter.

Figure 3.9 shows the evolution of the topology on a denser 161×41 cells mesh. The CA design algorithm converges to an almost black/white topology in about 80 design iterations. The figure clearly indicates that no checkerboard patterns are encountered.

To investigate the mesh independency of the obtained topology, the same topology problem with the same parameters as above is solved on three successively finer meshes. Figure 3.10 shows the converged optimal topologies for the three meshes. We note that the design features are the same in all three solutions. As the number of cells is increased, the grey areas encountered on coarser meshes disappear giving a more crisp solution. The run time for the MATLABTM implementation ranges

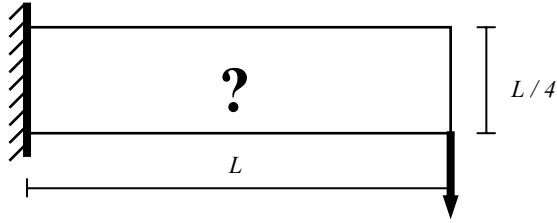


Figure 3.12: Unsymmetric cantilever problem.



Figure 3.13: Unsymmetric cantilever optimal topology.

from 226 seconds for the coarsest mesh to about 2000 seconds for the finest mesh.

The effect of changing the value of the penalization parameter p is shown in fig. 3.11. The converged designs are similar for the cases $p = 4$ and $p = 5$ to the design for the case $p = 3$. The popular value used $p = 3$ is quite sufficient to suppress grey areas.

3.5.4 Unsymmetric cantilever

To further illustrate the robustness of the CA design rule, we consider an unsymmetric cantilever of aspect ratio 4 as shown in fig. 3.12. The algorithm was run on a 161×41 cells mesh and $p = 3$ for a volume fraction $\eta = 0.5$. The converged topology as shown in fig. 3.13 is close to the optimal layout calculated by [52].

3.5.5 Michell truss

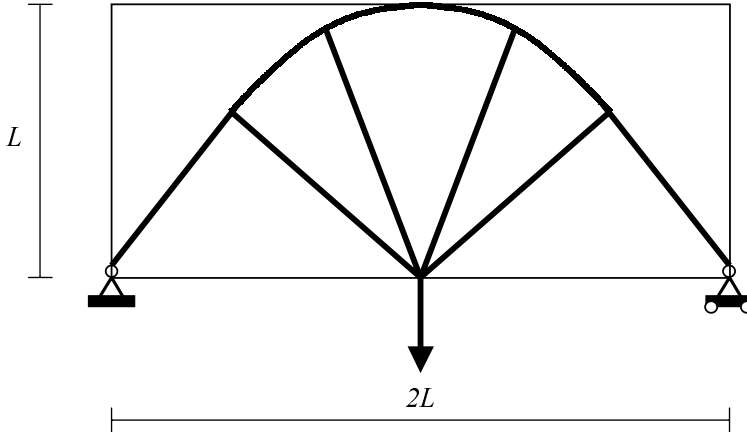


Figure 3.14: Michell truss, domain and optimal layout.



Figure 3.15: Michell truss optimal topology.

The problem under study is a rectangular ground structure supported at the lower corners, with a downward force applied at the center of the bottom edge (see fig. 3.14). This problem roughly corresponds to the Mitchell truss, a classical topology optimization problem. The *ideal* solution is shown superimposed on the ground structure.

The CA design algorithm is run on a 81×81 cells modeling half the domain by using symmetry. The converged design for $p = 3$ and a volume fraction $\eta = 0.3$ is shown in fig. 3.15. The converged topology corresponds reasonably well to the exact solution. It is noteworthy that the Michell truss solution can be approached only in the limit of vanishingly small volume fraction.

3.6 Conclusion

A cellular automata (CA) design algorithm is presented for two-dimensional minimum compliance design. The design rule was rigorously derived based on the continuous optimality criteria interpreted as local Kuhn-Tucker condition. The topology optimization formulation is based on the popular SIMP method. Although SIMP is widely used in the topology optimization literature, the method suffers from instabilities in the form of checkerboard patterns and mesh dependency.

In this chapter, we presented a novel approach to suppress these instabilities. The design variables are associated with cells and are interpolated smoothly over the lattice. A new averaging scheme for the density based on the averaging the compliance of the material rather than a direct mathematical average was proposed. The proposed compliance averaging scheme proved effective in suppressing numerical instabilities. Numerical experiments demonstrated that CA generated topologies do not suffer from checkerboard patterns and that mesh refinement does not change the optimal topology but rather increase its crispness.

The analysis rule was again derived using energy principles applied to cell neighborhoods. Numerical experiments indicate that the CA combined analysis and design algorithm performs satisfactorily even on a serial machine for moderately fine lattices. The computational cost of running the CA algorithm for fine lattice discretization on serial machines was prohibitive. For this reason, the use of CA design rule combined with finite element analysis was investigated. The CA design algorithm performed satisfactorily for all considered numerical examples.

In their pioneering work on applying CA to topology optimization, Kita and Toyoda [21] report a number of finite element analyses in excess of a thousand, while our

results indicated that even for dense lattices (by far denser than in [21]), the required number of finite elements analysis was less than a hundred. It is also important to note that the CA algorithm demonstrated numerical robustness manifested in the absence of checkerboard patterns and mesh dependency in contrast to the results of [21]. Moreover, the proposed CA design rule is based on rigorous optimality rather than heuristic reasoning. It is concluded that the optimality based CA approach is effective in solving topology optimization problems.

Further numerical experiments on parallel architectures are required to determine the relative merits of using matrix finite element techniques versus CA local analysis rules.

Chapter 4

Optimal Design of an Electrostatically Actuated MicroBeam for Maximum Pull-in Voltage

4.1 Introduction

In the previous chapters, CA was applied to two different types of problems. Each problem posed its distinct challenges. In the case of eigenvalue design, treated in chapter 2, the challenge was in the global nature of the design objective. The main question was how to devise local design rule to design the structure to satisfy a global (eigenvalue) requirement. In the case of topology design, treated in

Chapter 4

chapter 3, the question was how to devise a local design rule to rigorously satisfy the optimality criteria of the problem. The rigorous optimality-based derivation of the design rule was the main improvement over the approach adopted in chapter 2. Moreover, by exploiting the flexibility of CA in defining cells, we defined the cells in a way that eliminated the sometimes problematic numerical instability problems that traditionally appeared in topology design algorithms. Using the proposed CA design approach, we obtained crisp, well-defined, converged optimal topologies and demonstrated the effectiveness of CA in several topology design examples.

In both eigenvalue and topology design CA algorithms, the analysis rule was based on energy minimization over a cell neighborhood. The use of energy minimization to derive approximate equilibrium conditions is a well established procedure in structural analysis. It seems that the most challenging part of CA combined analysis and design formulation would be formulating the rigorous optimality conditions of the problem in the form of local design rules.

This is by no means to downplay the importance of the analysis rule. Local analysis rules are an integral part of the CA formulation. Nevertheless, due to the slow convergence of CA analysis rules on serial architectures, an issue that was raised and discussed in section 3.5.1, further development of CA analysis rules will be tied more intimately with parallelization and acceleration studies. Parallelization and convergence acceleration research is being carried out by other researchers of the CA group under Prof. Gürdal, and is outside the scope of this dissertation. The main features of these efforts is highlighted in the next chapter, where the state of the CA research is surveyed.

In this chapter, we derive CA design rules for the design of systems subject to nonlinear instability. Many structural elements, particularly shells, show limit point behavior where there is a maximum value of load after which no static equilibrium of the system exists. The load capacity of such structures is inherently limited by their geometric behavior even when the material is operating in the linear elastic range. As such, this is an important structural optimization problem. The nonlinear behavior of the system means that the optimality conditions derived in chapter 3 are no longer applicable. The optimality conditions for limit point design are derived in this chapter and used to formulate the CA design rule as a local optimization problem.

The problem considered in this chapter is motivated by a Micro-electro-mechanical-systems MEMS application. A brief introduction to optimization studies in MEMS is given in section 2. In section 3, the mathematical model of a MEMS microbeam is developed and a qualitative description of the pull-in instability is provided. In section 4, the shape optimization problem is posed as a continuous variational problem. In section 5, we develop an optimality criterion based CA design update rule to solve the optimization problem. This CA design rule is coupled with traditional finite element analysis to obtain numerical solutions of the optimal MEMS structures. Finally, we present results for both thickness and width optimization in section 6, and then present conclusions. The presentation given here follows closely the paper by Abdalla *et al* [55].

4.2 MEMS Optimization: an overview

Micro-electro-mechanical devices (MEMS) are rapidly gaining popularity in a variety of industrial applications such as the aerospace, automotive, and biomedical industries. MEMS devices are generally classified according to their actuation mechanisms. Actuation mechanisms for MEMS vary depending on the suitability to the application at hand. The most common actuation mechanisms are electrostatic, pneumatic, thermal, and piezoelectric [56]. Electrostatically actuated devices form a broad class of MEMS devices due to their simplicity as they require few mechanical components, and small voltage levels for actuation [56]. The structural elements that are used in MEMS devices are typically simple elements like beams, plates, and membranes. Electrostatically actuated microbeams are used in many MEMS devices such as capacitive microswitches and resonant sensors. Manufacturing and design of these devices are, to some extent, in a more mature stage than some other MEMS devices.

MEMS microbeams are liable to an instability known as the pull-in instability. When the applied voltage is increased beyond a critical value, called pull-in voltage, stable equilibrium positions of the microbeam cease to exist. Pull-in instability [57] greatly limits the stable range of operation of microbeams. In most cases it would be highly desirable to delay the onset of pull-in for better performance of the device. Various control schemes have been proposed in the literature to overcome the pull-

Chapter 4

in instability. For example, Pelesko and Triolo [58] studied a number of control methodologies. They showed that pull-in voltage can be effectively increased by the proposed control schemes. The pull-in voltage depends on the interaction of the electrostatic forces generated by the applied voltage, and the structural stiffness of the microbeam. Since the shape of the microbeam influences both the electrostatic forces and the structural stiffness, it is natural to seek optimized shapes of the microbeam to passively maximize the pull-in voltage instead of using control schemes. This avoids the need to add control circuitry, and will eliminate the energy expended in control, resulting in higher efficiency.

Shape optimization is fairly new in the MEMS literature. The optimization procedure depends on the design goals. [59] studied the relation between the sensitivity of area-constrained MEMS devices like sensors and actuators and their geometry, the objective was to maximize their sensitivity by optimizing the geometry. [60] studied theoretically and experimentally the effect of optimizing the metal film thickness to obtain the maximum thermal sensitivity for bi-material temperature sensors and actuators. [61] described an optimization technique that was used to predict the shapes of the relay switch which lead to design improvement. They provided an example that optimization leads to an actuation force decreased by a factor of two. Shape and topology optimization of classes of MEMS devices are considered by a number of authors [62, 63].

In this chapter, we study the shape optimization of electrostatically actuated MEMS microbeams. Shape changes of the microbeam can be affected through either thickness or width manipulation. A microbeam with varying thickness can be manufactured using deposition whereas etching can be used to manufacture a microbeam with varying width [56]. We investigate the optimal shapes of the microbeam in either case. Specifically, we find the thickness and width distribution that maximizes the pull-in voltage. We compare the thickness-optimized and width-optimized microbeam designs in terms of pull-in voltage gains and microbeam deflections at pull-in.

4.3 Micro-Beam Model

A sketch of the micro-beam actuator under consideration is shown in Fig. 4.1. The micro-beam is homogeneous with length L , a width distribution $b(x)$ and a thickness distribution $h(x)$. The beam is made of a conducting material and forms the upper electrode. The lower electrode remains stationary. The zero voltage separation between the electrodes is d . When a voltage is applied across the actuator, electrostatic forces are generated on the beam; causing it to deflect. In this paper, we consider static response only, where a DC voltage V is applied across the actuator. The beam is assumed to be placed in vacuum. The fringing effects at the electrode edges are ignored; the electrostatic force on the beam, thus, will act in the z direction only.

The choice of structural model of the microbeam depends on the magnitude of the deflection compared to the thickness of the microbeam. Since the microbeam will deform in the order of the gap width d , the proper form of the structural model will depend on the ratio d/h . For microbeams with a value of $d/h \leq 1$, a linear Euler-Bernoulli model will be an adequate representation of the microbeam behavior. For moderate values of d/h , bending-stretching coupling terms need to be taken into account. For large values of d/h , or when the microbeam is initially under large tension, the microbeam should be modeled as a membrane. In this paper, we assume that d/h is less than unity, and that the microbeam is not under initial tension. For this case, the total potential energy Π of the beam is written as,

$$\Pi = \mathcal{U} + \mathcal{V} \quad (4.1)$$

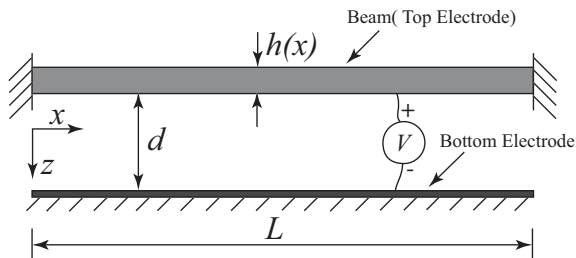


Figure 4.1: The micro-beam configuration

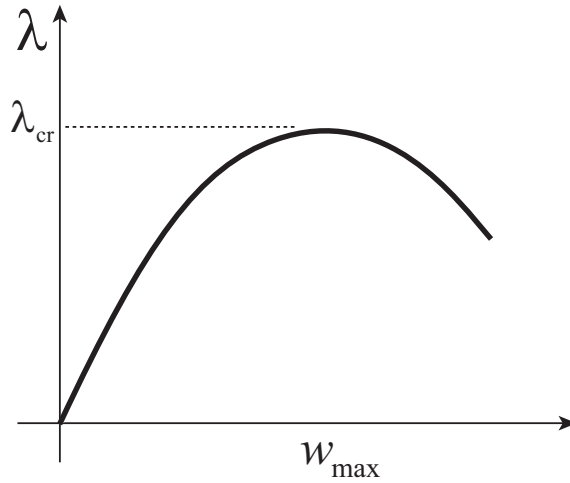


Figure 4.2: Typical load-displacement response of the micro-beam.

where, considering small deflections, the strain energy due to bending, \mathcal{U} is:

$$\mathcal{U} = \frac{1}{2} \int_0^L EI w''^2 dx \quad (4.2)$$

and the potential energy, \mathcal{V} , due to the electrostatic force,

$$p(x) = \frac{1}{2} \frac{\varepsilon_o b V^2}{(d-w)^2}, \quad (4.3)$$

is,

$$\mathcal{V} = -\frac{1}{2} \varepsilon_o V^2 \int_0^L \frac{b}{(d-w)} dx \quad (4.4)$$

In these equations, E is Young's modulus of the beam material, I is the second moment inertia of the beam cross-section, w is the beam displacement along the z direction and ε_o is the dielectric constant of vacuum. A prime indicates derivatives with respect to the beam coordinate x .

The nondimensional potential energy, $\tilde{\Pi}$, is written as:

$$\tilde{\Pi} = \frac{1}{2} \int_0^1 \tilde{I} \tilde{w}''^2 d\tilde{x} - \tilde{\lambda} \int_0^1 \frac{\tilde{b}}{(1-\tilde{w})} d\tilde{x} \quad (4.5)$$

where the corresponding non-dimensional variables are:

$$\begin{aligned}\tilde{x} &= \frac{x}{L}; \tilde{w} = \frac{w}{d}; \tilde{b} = \frac{b}{b_o}; \tilde{h} = \frac{h}{h_o}; \tilde{I} = \frac{1}{12} \tilde{b} \tilde{h}^3 \\ \tilde{\Pi} &= \frac{\Pi L^3}{EI_o d^2}; \tilde{\lambda} = \frac{\varepsilon_o b_o V^2 L^4}{2EI_o d^3}; (\cdot)' = \frac{d(\cdot)}{d\tilde{x}}\end{aligned}\quad (4.6)$$

where b_o and h_o are the uniform nominal width and thickness. As $\tilde{\lambda} \propto V^2$, we treat $\tilde{\lambda}$ as the non-dimensional load parameter. For the rest of the paper, we work exclusively in terms of the nondimensional variables. The tilde above nondimensional quantities will be dropped for convenience.

The Euler-Lagrange equation of the above energy functional yields the equilibrium equation as:

$$(I w'')'' = \frac{\lambda b}{(1-w)^2} \quad (4.7)$$

The term on the left hand side of (4.7) represents the mechanical restoring force and the right hand side term represents the electrostatic force. As the voltage is increased, the electrostatic force increases. This deflects the beam further till a new equilibrium position is reached. The nonlinear nature of the electrostatic force leads to a softening effect as shown in Fig. 4.2. There is a critical voltage beyond which equilibrium solutions cease to exist. This critical voltage is called *pull-in voltage*, which corresponds to a critical load λ_{cr} . In structural mechanics literature, this instability is referred to as *limit point* and as *saddle-node bifurcation* in mathematics literature [64].

4.4 Optimization Problem

The objective of the present paper is to optimize the dimensions the micro-beam to maximize λ_{cr} , given a maximum available amount of material. The material volume constraint is expressed as a non-dimensional integral constraint,

$$\int_0^1 b h dx = 1 \quad (4.8)$$

Chapter 4

We also impose a minimum width and thickness constraints,

$$\bar{b} - b(x) \leq 0, \quad \bar{h} - h(x) \leq 0 \quad (4.9)$$

where \bar{b} and \bar{h} are some minimum allowable width and thickness. The optimization problem for the continuous beam is then posed as,

$$\begin{aligned} & \max_{y(x)} \lambda_{cr} \\ & \text{such that : } \int_0^1 y \, dx = 1 \\ & \bar{y} - y(x) \leq 0 \end{aligned} \quad (4.10)$$

In the above optimization problem, the width distribution $b(x)$ or the thickness distribution $h(x)$ is the unknown function $y(x)$ to be determined. When optimizing the width distribution, we consider the thickness to be constant at the nominal value (e.g., $h(x) \equiv 1$). Similarly, when optimizing the thickness distribution, the width is assumed to be constant at the nominal value. The lower bound on $y(x)$ is correspondingly taken as $\bar{y} = \bar{b}$ or $\bar{y} = \bar{h}$.

The necessary optimality conditions for this problem can be formulated using the techniques of variational calculus. This is the approach adopted in chapter 3 for the topology optimization problem. On the other hand, we can first discretize the problem into a finite number of CA cells. A constant width and thickness are associated with each cell, and one set of these values are considered as design variables, thus the problem is converted to a problem having a finite number of variables. For these problems, the optimality conditions can be readily derived using the Kuhn-Tucker conditions. Theoretically, the optimal design for the CA lattice will asymptotically approach to that of the continuous beam as the number of cells increases.

The nonlinear equations of equilibrium is derived based on the finite element method. The details of the derivation and the numerical method used to trace the nonlinear response of the beam are presented in Appendix A.

4.5 Optimality Criterion

The optimization problem for the discretized beam takes the form,

$$\begin{aligned} \max_{\mathbf{y}} \quad & \lambda_{cr} \\ \text{such that : } & \mathbf{y}^T \cdot \boldsymbol{\ell} = 1 \text{ and } \mathbf{g}(\mathbf{y}) \leq \mathbf{0} \end{aligned} \quad (4.11)$$

where \mathbf{y} is the vector of design variables, $\boldsymbol{\ell}$ is the vector of cell lengths and g_i is the lower bound constraint on the i th design variable (cell height or width of the i th cell). The Lagrangian \mathcal{L} for the above problem takes the form [4],

$$\mathcal{L} = -\lambda_{cr} + \mu(\mathbf{y} \cdot \boldsymbol{\ell} - 1) + \boldsymbol{\gamma} \cdot (\mathbf{g} + \mathbf{s}^2) \quad (4.12)$$

where $\boldsymbol{\gamma}$ is a vector of constant Lagrange multipliers, \mathbf{s} is a vector of constant slack variables and μ is the constant Lagrange multiplier associated with the volume constraint. Setting the first variation of the Lagrangian to zero, we get the optimality conditions,

1. stationarity,

$$-\frac{\partial \lambda_{cr}}{\partial \mathbf{y}} + \mu \boldsymbol{\ell} + \boldsymbol{\gamma} \cdot \frac{\partial \mathbf{g}}{\partial \mathbf{y}} = \mathbf{0} \quad (4.13)$$

2. constraints

$$\begin{aligned} \mathbf{y} \cdot \boldsymbol{\ell} - 1 &= 0 \\ \mathbf{g} &\leq \mathbf{0} \end{aligned} \quad (4.14)$$

3. switching conditions

$$\gamma_i s_i = 0 \quad (4.15)$$

The sensitivity of λ_{cr} can be obtained by differentiating the equilibrium equation (see Appendix A, equation (A.5)) with respect to y_i ,

$$\frac{\partial \mathbf{K}}{\partial y_i} \mathbf{q} + \mathbf{K}_t \frac{\partial \mathbf{q}}{\partial y_i} - \lambda_{cr} \frac{\partial \mathbf{p}}{\partial y_i} - \frac{\partial \lambda}{\partial y_i} \mathbf{p} = \mathbf{0} \quad (4.16)$$

Chapter 4

where, \mathbf{K}_t is the tangent stiffness matrix, \mathbf{q} is the vector of nodal displacements, and \mathbf{p} is the electrostatic load vector.

Left multiplying (4.16) by the tangent vector \mathbf{r} (see Appendix A for details) and noting that at the limit point, $\mathbf{r} \cdot \mathbf{K}_t = \mathbf{0}$, we get,

$$\frac{\partial \lambda}{\partial y_i} = \mathbf{r} \cdot \frac{\partial \mathbf{K}}{\partial y_i} \cdot \mathbf{q} - \lambda_{cr} \mathbf{r} \cdot \frac{\partial \mathbf{p}}{\partial y_i} \quad (4.17)$$

where \mathbf{r} is normalized such that $\mathbf{r} \cdot \mathbf{p} = 1$. Recalling that,

$$\mathbf{K}_i^e = I_i \hat{\mathbf{K}}_i^e, \quad \mathbf{p}_i^e = b_i \hat{\mathbf{p}}_i^e$$

we obtain,

$$\frac{\partial \lambda_{cr}}{\partial y_i} = \frac{\partial I_i}{\partial y_i} \mathbf{r}_i^e \cdot \hat{\mathbf{K}}_i^e \cdot \mathbf{q}_i^e - \lambda_{cr} \mathbf{r}_i^e \cdot \hat{\mathbf{p}}_i^e \frac{\partial b_i}{\partial y_i} \quad (4.18)$$

where \mathbf{q}_i^e is the displacement vector corresponding to the i th cell and \mathbf{r}_i^e is the corresponding component of the eigenvector \mathbf{r} .

The dual potential \mathcal{V} represents negative the work done per unit length by the actual electrostatic forces \mathbf{p}_i^e on the eigenvector displacements \mathbf{r}_i^e , and is given by,

$$\mathcal{V} = -\lambda_{cr} \Psi_i b_i \quad (4.19)$$

where,

$$\Psi = \frac{\mathbf{r}_i^e \cdot \hat{\mathbf{p}}_i^e}{\ell_i} \quad (4.20)$$

The dual complementary strain energy density \mathcal{U}^* represents the complementary work done by actual elastic forces over the eigenvector displacements, and is given by,

$$\mathcal{U}^* = \frac{\Phi_i}{I_i} \quad (4.21)$$

where,

$$\Phi_i = \frac{I_i^2 \mathbf{r}_i^{eT} \hat{\mathbf{K}}_i^e \mathbf{q}_i^e}{\ell_i} \quad (4.22)$$

We express the sensitivity in terms of \mathcal{U}^* , and \mathcal{V} as,

$$\frac{\partial \lambda_{cr}}{\partial y_i} = \frac{1}{I_i^2} \frac{\partial I_i}{\partial y_i} \Phi_i \ell_i - \lambda_{cr} \ell_i \Psi_i \frac{\partial b_i}{\partial y_i} = -\ell_i \frac{\partial}{\partial y_i} (\mathcal{U}^* - \mathcal{V}) \quad (4.23)$$

From this relation, we can interpret the optimality conditions for the optimization problem (4.11) as optimality conditions for the following local optimization problem,

$$\begin{aligned} \min_{y_i} \quad & \frac{\Phi_i}{I_i} + (\mu + \lambda_{cr} \Psi_i) b_i \\ \text{such that :} \quad & \bar{y} - y_i \leq 0 \end{aligned} \quad (4.24)$$

The above local optimization problem is applicable for all the cells in the CA lattice. Similar to the topology optimization case, the local optimization problem can be analytically solved. The general solution to the above one-dimensional problem is obtained as,

$$y^* = \begin{cases} \sqrt{\frac{12 \max(\Phi_i, 0)}{\mu + \lambda \Psi_i}} & \text{width sizing} \\ \sqrt[4]{\frac{36 \max(\Phi_i, 0)}{\mu}} & \text{thickness sizing} \end{cases} \quad (4.25)$$

The minimum width, or thickness, constraint is satisfied by updating the design variables as,

$$y_i = \begin{cases} y^* & y^* \geq \bar{y} \\ \bar{y} & \text{otherwise} \end{cases} \quad (4.26)$$

The Lagrange multiplier μ in (4.24,4.25) is again determined using the volume constraint. We refer the reader to section 3.2.1 for a more detailed discussion of Lagrange multipliers. From (4.25), we see that the width/thickness values depend upon μ . This suggests a one-dimensional map of the form,

$$\mu \mapsto v \quad (4.27)$$

where $v = \mathbf{y} \cdot \boldsymbol{\ell}$ is the total volume.

A flow chart of above iterative scheme is shown in Fig. 4.3. First, the width/thickness distribution is initialized to one (the nominal nondimensional value). The design loop begins, with a nonlinear normal flow analysis to trace the nonlinear response of the actuator and bracket the pull-in point. The displacement vector at pull-in and the mode shape are then used to update the design using the optimality based CA design rule (4.26). At this stage, the Lagrange multiplier μ is not known, and an inner iteration loop to update μ according to the Newton- Raphson method applied to the condition $v = 1$ to satisfy the volume constraint.

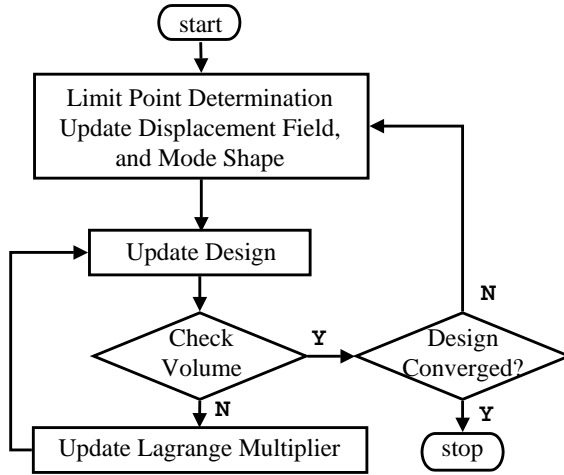


Figure 4.3: Iterative Procedure for finding the optimal design.

4.6 Results

The optimum width and thickness distributions are determined for microbeams with four different boundary conditions: (a) Simply Supported (SS); (b) Clamped-Clamped (CC); (c) Clamped-Free (CF) and (d) Clamped-Simply Supported (CS). The number of elements has a significant effect on the optimized pull-in value and the minimum thickness (or width) value. It is observed that for smaller number of elements, the minimum thickness (width) constraint might not be active. As the number of elements is increased, the value of the critical pull-in parameter, λ_{cr} , converges and the minimum thickness (width) constraint becomes active. Table. 4.1 gives a variation of λ_{cr} and the minimum thickness with the element size for thickness optimization of a CC microbeam and $\bar{h} = 0.2$. The convergence of the optimization iterations is quite fast as demonstrated in Fig. 4.4, where the convergence of the objective function (the pull-in parameter λ_{cr}) is plotted versus the number of iterations. We observe that after the first application of the optimization algorithm, the value of pull-in parameter is within 5% of the optimal value. It is also interesting to note that the algorithm converges monotonically.

Table 4.1: Variation of the design with element size

No. of elements	16	32	64	128*
λ_{cr}	9.4827	9.7947	9.8917	10.0136
h_{min}	0.3988	0.2915	0.2217	0.2

The * indicates that the minimum thickness constraint is active.

Table 4.2: Pull-in values for optimized width for different minimum width values and boundary conditions

Boundary Condition	λ_{cr}				
	$\bar{b}=1$	$\bar{b} = 0.8$	$\bar{b} = 0.6$	$\bar{b} = 0.4$	$\bar{b} = 0.2$
SS	1.1549	1.1549	1.1548	1.1541	1.1492
CC	23.107	15.196	11.19	8.442	5.8413
CF	0.7868	0.46257	0.31139	0.21688	0.14007
CS	7.5018	5.6800	4.6056	3.7592	2.788

Table 4.3: Pull-in values for optimized thickness for different minimum thickness values and boundary conditions

Boundary Condition	λ_{cr}				
	$\bar{h}=1$	$\bar{h} = 0.8$	$\bar{h} = 0.6$	$\bar{h} = 0.4$	$\bar{h} = 0.2$
SS	1.1492	1.4707	1.6115	1.6772	1.700
CC	5.8413	8.4088	9.3980	9.8597	10.0136
CF	0.1401	0.2350	0.2877	0.3209	0.3391
CS	2.7880	3.790	4.1902	4.3767	4.4405

Chapter 4

Table. 4.2 gives the optimal pull-in values for various values of \bar{b} for each boundary condition. We see that an optimal width distribution leads to a substantial increase in the pull-in voltage for some boundary conditions, and marginal increase for others. For example, for the case ($\bar{b} = 0.2$), the percentage increase in the pull-in for different boundary conditions is: (a)SS: 0.5%, (b)CC:295%, (c)CF: 461%, and (d)CS: 169%. Moreover, even when a small change in shape is allowed (e.g., $\bar{b} = 0.8$), the pull-in voltage increases substantially for all considered boundary conditions except the SS case that fails to show any significant improvement due to optimization. The optimal width distributions are shown in Fig. 4.5 for $\bar{b} = 0.2$, and Fig. 4.6 for $\bar{b} = 0.8$. The dashed line indicates the nominal design.

The optimal pull-in values for various values of \bar{h} for each boundary condition are given in Table. 4.3. We see that an optimal thickness distribution leads to a substantial increase in the pull-in voltage. For example, for the case ($\bar{h} = 0.2$), the percentage increase in the pull-in for different boundary conditions is: (a)SS: 47.93%, (b)CC:71.35%, (c)CF: 142.06%, and (d)CS: 59.27%. Once again, even when a small change in shape is allowed (e.g., $\bar{h} = 0.8$), the pull-in voltage increases substantially for all considered boundary conditions. The optimal thickness distributions are shown in Fig. 4.7 for $\bar{h} = 0.2$, and in Fig. 4.7 for $\bar{h} = 0.8$.

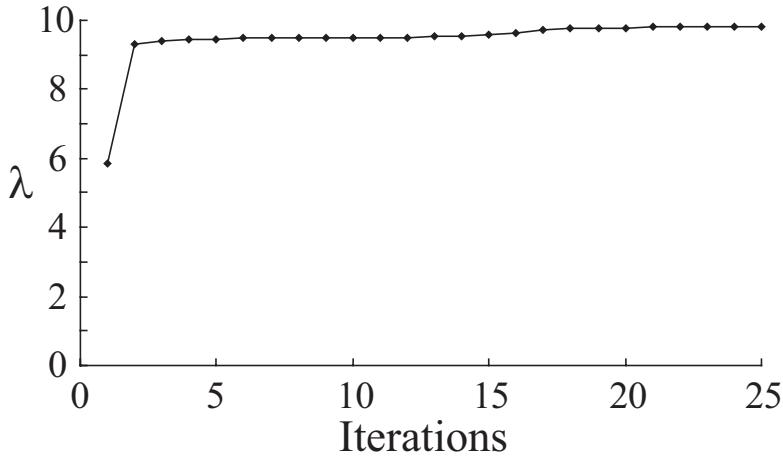
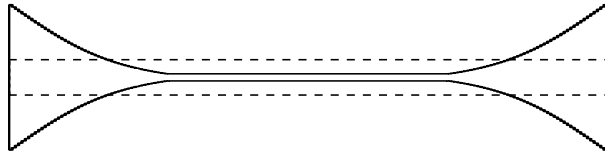


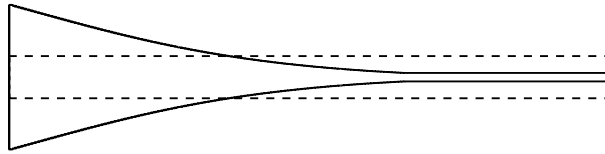
Figure 4.4: Convergence of the iterative design procedure for the CC thickness design case and 32 elements.



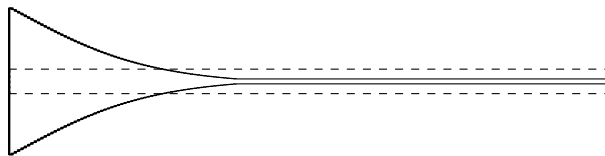
(a) SS



(b) CC



(c) CF



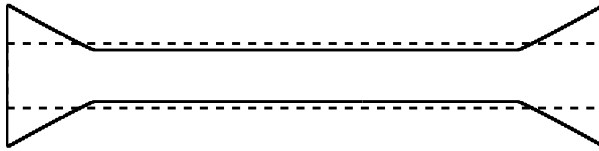
(d) CS

Figure 4.5: Optimal beam planform shapes for $\bar{b} = 0.2$.

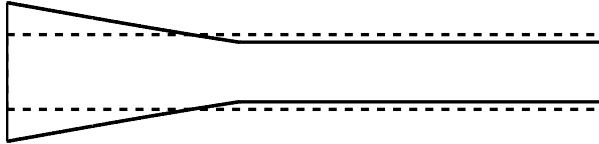
Chapter 4



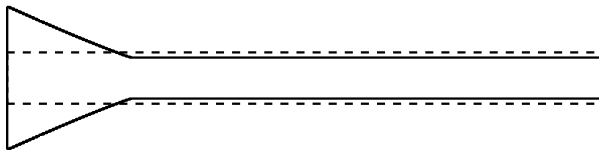
(a) SS



(b) CC

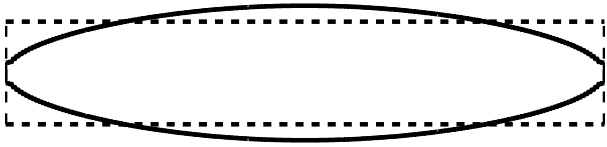


(c) CF

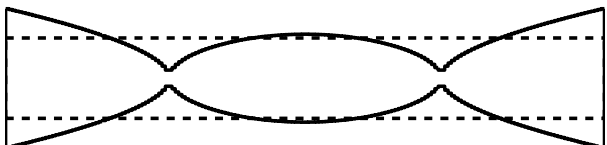


(d) CS

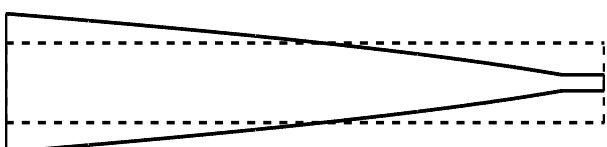
Figure 4.6: Optimal beam planform shapes for $\bar{b} = 0.8$.



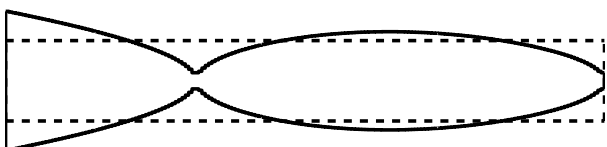
(a) SS



(b) CC



(c) CF



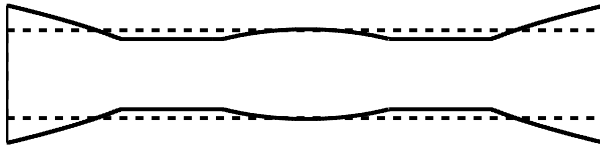
(d) CS

Figure 4.7: Optimal beam thickness distributions for $\bar{h} = 0.2$.

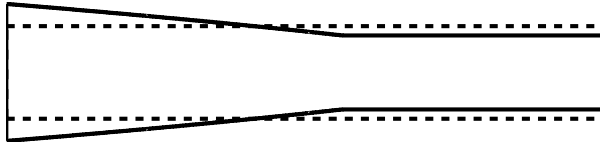
Chapter 4



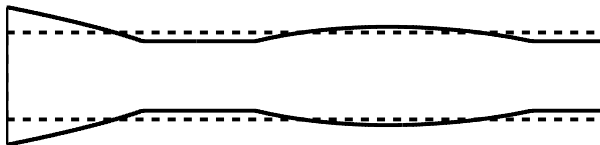
(a) SS



(b) CC



(c) CF



(d) CS

Figure 4.8: Optimal beam thickness distributions for $\bar{h} = 0.8$.

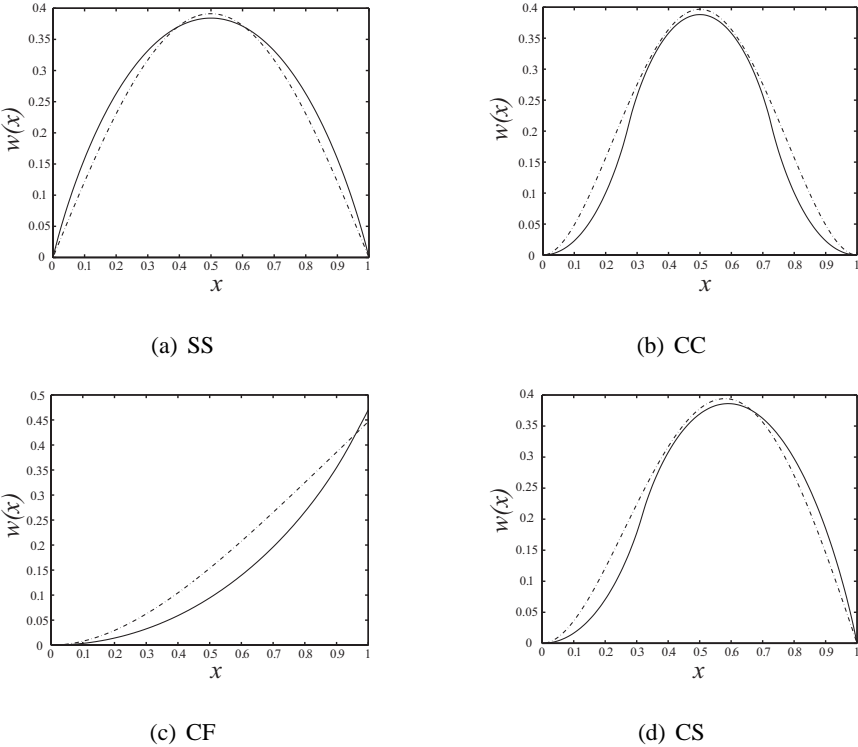


Figure 4.9: Nominal and optimal beam deflections for different boundary conditions with $\bar{h} = 0.2$

For both width and thickness optimization, the CC and CF cases show substantial pull-in gain. This is an important result because the above two cases, namely, CC and CF, are commonly used actuator configurations.

Deflection shapes for different boundary conditions are shown in Fig. 4.9 for thickness shaping, and in Fig. 4.10 for width shaping. Qualitatively speaking, the optimization algorithm tries to place more material where there is appreciable strain energy, and to simultaneously decrease the electrostatic load as much as possible. For thickness optimization, the algorithm can control only material placement at points with high strain energy, but not the electrostatic force which is independent of the thickness. Consequently, the optimal thickness distribution assigns higher thicknesses to regions with large curvature (strain energy), as shown in Fig. 4.9.

Chapter 4

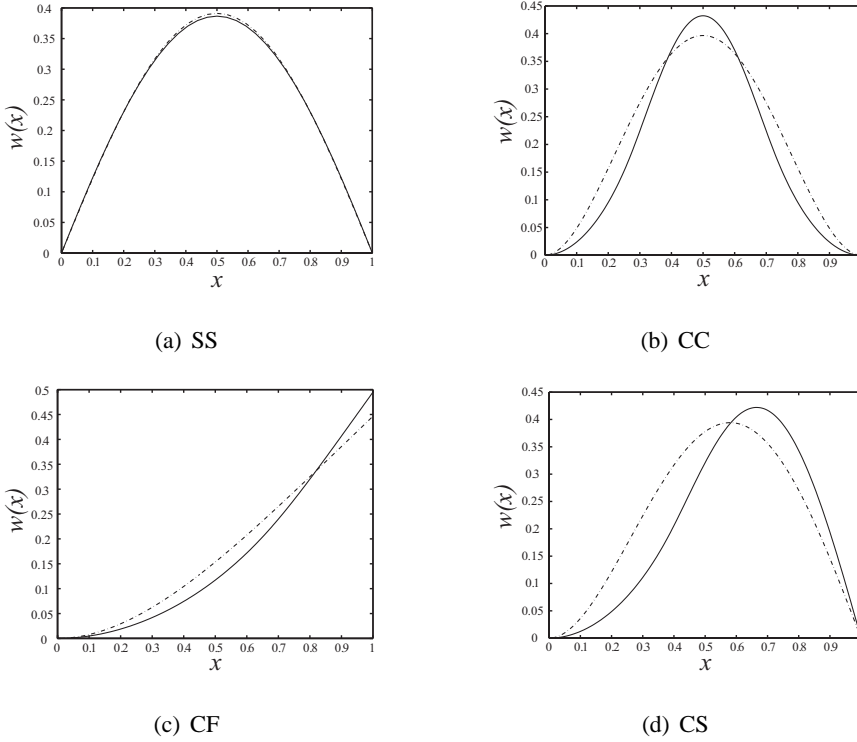


Figure 4.10: Nominal and optimal beam deflections for different boundary conditions with $\bar{b} = 0.2$

This leads to an increase of the structural stiffness of the microbeam. For this reason, although the pull-in voltage is increased significantly, the pull-in deflection is almost unchanged.

On the other hand, for width shaping, since the electrostatic force is proportional to width, the optimizer has the freedom either to place more material at points with high strain energy to increase stiffness, or to remove material to reduce the electrostatic loading. The optimizer will remove material from places with high flexibility, in the sense of high sensitivity of deflection to applied load. This can be observed in Fig. 4.5-b where for CC boundary conditions, the optimizer removes material from the center of the microbeam, while in Fig. 4.5-c for a CF boundary conditions the optimizer removes material from regions near the free end. Depending on

the boundary conditions, the effect of stiffening the beam and reducing the electrostatic load can add up or cancel out. For the SS case, width optimization tries to increase the stiffness by increasing the width around the center of the beam, this also increases the electrostatic load, and hence the improvement is nearly cancelled, leading to the marginal increase in pull-in value reported in Table. 4.2. Since width optimization can lead to a decrease in beam stiffness (while increasing the pull-in voltage), the pull-in deflections can actually increase due to the optimization. This is clearly seen from Fig. 4.10b-c.

4.7 Conclusions

In this chapter, a CA design rule for the design of structures showing nonlinear limit point behavior is formulated based on rigorous optimality conditions. Faithful to our methodology, the design rule is formulated as a local optimization problem. The design rule turns out to be quite similar to the design rule derived for topology optimization in chapter 3. Again, the design rule is a function of a suitably defined complementary energy. We demonstrated that the CA design algorithm is efficient and converges monotonously. The CA design algorithm was applied to the problem of shape design of a MEMS microbeam.

The stable range of operation of electrostatically actuated microbeams is limited by pull-in instability. In this chapter, we presented a way to passively (as contrasted to active control schemes) increase the pull-in voltage for an electrostatically actuated microbeam by changing the microbeam shape. Changes in the microbeam shape are assumed to be achieved either through thickness or width changes. We obtain optimal thickness and width designs for the microbeam that maximize the pull-in voltage.

It is demonstrated that a redistribution of width or thickness leads to substantial increase in the pull-in voltage and thus providing a greater range of operation for the device. The increase in pull-in for the optimized width distribution is even more dramatic when compared to that for thickness variation. It is also noteworthy that for thickness optimization, the pull-in deflection is not compromised, while for width optimization it is increased. On the whole, width optimization seems to be more

useful than thickness optimization. This is fortunate, since width shaping using etching is a potentially easier manufacturing process than deposition required for thickness shaping.

Chapter 5

Conclusion

5.1 Summary

The dissertation started with an introductory chapter stating the motivation for the work and providing a brief description of the basic elements of Cellular Automata. Although the Cellular Automata (CA) paradigm was known for decades, its application to structural design is quite recent. The existing literature on applying CA to problems in structural analysis and/or design was explored and the areas requiring further research were identified. This was followed by a description of the scientific contribution and the organization of the dissertation.

In chapter 2, an algorithm for designing structures for eigenvalue requirements was presented. Conceptually, this was an important area of investigation because the local nature of CA algorithms would, at least apparently, be challenged by the global nature of the eigenvalue response. The proposed algorithm, was designed to be fully local in nature, and thus suitable for CA type implementation. The algorithm was applied to the design of Euler-Bernoulli columns against buckling. The basic idea of the algorithm was very simple. For buckling design, the design load is applied

Chapter 5

to the structure and the displacement field is updated. Once the displacements are found, the local cross sections are sized to have minimum weight while maintaining the maximum stress less than a specified value. The process was repeated until convergence was obtained. The method proved effective in accurately predicting optimal column shapes and the corresponding buckling mode shapes.

The work presented in chapter 2, was closely modeled after previous applications of CA to structural optimization [21, 6], where local design rules were heuristically devised in the form of a local optimization problem involving the local state of stress. On the other hand, the analysis rule used to predict displacements was, for the first time, derived using energy principles. At this point it became clear that the derivation of analysis rules based on energy principle was the simplest general approach, the main remaining question was whether CA design algorithms could be equally rigorously derived from known principles of structural optimization.

By a careful study of the algorithm presented in chapter 2, it was recognized that the success of the algorithm was based on the exact correspondence between the proposed heuristic local optimization problem and the rigorous optimality criteria for the particular cases we considered. Based on this observation, it was determined that CA design rules could be obtained rigorously by using variational calculus and optimality conditions. A combination of an optimality based design rule and an energy based analysis rule would lead to rigorous formulations for both analysis and design.

In chapter 3, a CA topology design algorithm was presented where the design rules for minimum compliance design of two-dimensional linearly elastic continuum topology were derived using variational calculus. The CA design rule was obtained based on the continuous optimality criteria interpreted as local Kuhn-Tucker conditions. As such, the design rule at each cell involved the solution of a simple one-dimensional optimization problem. The CA analysis rule was derived, similarly to the work in chapter 2, based on energy minimization.

Numerical experiments with the proposed algorithm indicated that the CA design rule is quite robust and does not suffer from checkerboard patterns, mesh-dependent topologies, or numerical instabilities. Given the simplicity of the CA algorithm, it became clear that CA is a good candidate as a topology design tool.

Topology optimization to minimize the compliance of a structure required a detailed CA lattice to capture topological features. In this respect, it was found that the CA analysis rule converged rather slowly. The deterioration of CA convergence rate with lattice refinement was to be expected given that CA relied completely on local exchange of information. In a hypothetical ideal CA computing machine, where all cells are updated in parallel at a high rate, convergence rate deterioration would pose no particular problems. The efficiency of CA would still be considerable due to the simplicity of the processing elements. On the other hand, such deterioration in convergence rate would be a significant limiting factor when CA is used on existing serial processors.

At this point, it was decided that for serial architectures, the update of displacements is best achieved by a direct finite element analysis of the structure. In this scheme, the CA design rule is applied *after* the finite element analysis produces the analysis results. As such, CA was to perform the role of an *optimizer* in combination with standard FEM analysis. This hybrid approach corresponds to the original proposition by Kita and Toyoda [21]. The main departure between the approach presented in this dissertation and that earlier work lied in the rigorous derivation of the design rules, and the identification of finite element *nodes* with CA cells. This led to significant decrease in the number of finite element analyses required for convergence. It is also important to note that the CA algorithm demonstrated numerical robustness manifested in the absence of checkerboard patterns and mesh dependency.

Encouraged by the success of optimality-based CA design rules in topology optimization, CA design rules for nonlinear problems showing limit point behavior were derived in chapter 4 using rigorous optimality conditions. The design rule was successfully formulated as a local cell-level optimization problem. The CA design rule was couple to nonlinear finite element analysis to solve the problem of shape design of a MEMS microbeam. The convergence of the CA design rule was quite fast requiring only twenty to thirty nonlinear finite element analyses. The results confirmed that local design rules based on optimality perform satisfactorily and could be indeed considered a general method for deriving CA design rules.

The structural response of MEMS microbeam is nonlinear due to the nonlinear dependence of electrostatic load on the microbeam deflection, and exhibits limit point (pull-in) behavior. We considered optimizing the shape of a capacitive micro-beam for maximum pull-in voltage. Extensive results for different beam boundary condi-

tions were generated. The optimization results indicated that substantial increase in pull-in voltage can be achieved by varying the width and/or thickness distribution.

The implementation of CA as combined analysis and design tool where both CA analysis rules and CA design rules are applied to obtain a final converged design *together* with the corresponding displacements can be considered to be established at the algorithmic level. A combination of energy minimization for the derivation of the analysis rule and optimality for the design rule is generally applicable for a wide range of structural problems. The main challenges lie mainly in devising suitable hardware and software implementations where the CA computational advantage stemming from massive parallelism would be clearly demonstrated. Another approach would be to implement a suitable *convergence acceleration* method such as multigrid acceleration so that the benefits of the CA approach can be extended to serial machines and parallel clusters with a limited number of processors. Both these issues are under current research as will be discussed in the next section.

5.2 Overview of the state of Cellular Automata research

In the previous section, an overview of the dissertation was given. It was found that on the existing serial processors, CA do not perform satisfactorily in terms of computational time. Efforts by other researchers in Prof. Gürdal's group were undertaken to remedy this shortcoming [65]. The CA paradigm was used as the basis of a modular Fortran 90 package. The CA package allows for a general definition of cells. In this package, any number of field variables, design variables, material properties, and geometric dimensions can be used to define the cell. Cell neighborhood was also left general with the flexibility to select different cell neighborhood for different parts of the CA lattice. The package was designed such that matrix techniques can be easily used to construct the local analysis rules based on energy minimization. The package also allows the use of either analytic expressions for the solution of the cell-level local optimization problem associated with the design rule, or the ability to call a mathematical programming routine to numerically solve it. The package was also made parallel on a message passing architecture making it suitable for implementation on traditional parallel clusters with arbitrary number of processors.

The topology optimization algorithm proposed in this dissertation was extended to fiber reinforced layer design. In the case of design of fiber composites, both the topology and/or the fiber angle direction are to be determined at each CA cell. Extensive topology optimization results for single and multiple load cases using CA for both analysis and design are reported on dense lattices [66, 51].

Another research effort was conducted in parallel to implement CA on a Field Programmable Gate Array (FPGA) hardware [67]. Initial work included the optimization of hardware description of CA rules. A new algorithm to use variable precision computing was proposed [68], and implemented. Some limited experiments were carried out on a beam bending problem. Cell update rates up to 3.5 billion cell updates per second were achieved on a medium-sized FPGA. The speed up compared to a single 1GHz Pentium IV processor was roughly 70. This initial investigation shows clearly that CA maps well onto FPGA hardware. With the rapid increase in FPGA size and speed, it is expected that a CA design tool running on FPGA will be highly competitive with finite element based design tools running on large parallel clusters; at a fraction of the cost.

In yet another parallel track, the acceleration of CA convergence using a multigrid approach was investigated [69]. In CA studies conducted so far, it was observed that the computational effort associated with CA iterations increases substantially as the number of cells is increased. It is generally known that fixed grid algorithms quickly damp short wavelength components of error, while long wavelength error components are damped slowly [70, 71]. A traditional solution to this problem is to use multigrid/full multigrid acceleration schemes [70, 71]. Multigrid acceleration of CA was carried out for minimum compliance design of Euler-Bernoulli beam under bending and it was shown that a multigrid accelerated CA attains optimal computational complexity. The total number of cell updated required for convergence was found to be linearly proportional to the number of cells. Since multigrid acceleration preserves to large degree the local nature of the CA algorithm, it is expected to map well onto FPGA. Work is currently underway in this direction.

CA methods attracted the attention of researchers as well as the group lead by Prof. Gürdal. Application of CA to topology design in the context of bone mimicking was presented in [72]. Other extensions to CA in areas such as reliability based topology optimization is also underway by the authors of the above-mentioned paper. Several other contributions to CA applications to structural optimization have appeared in

recent conferences [73, 74]. As practical experience with CA matures and more advances in CA software and hardware are achieved, the field will attract more researchers. It is the hope of this researcher that CA will find its place among the engineering design tools of the future.

5.3 Prospects of future research

There has been a substantial progress made in demonstrating the feasibility of the Cellular Automata (CA) paradigm for the analysis and design of one and two-dimensional structural systems under static loads as witnessed by the work presented herein and as the quick survey above indicates. Although the set of applications considered so far is not exhaustive of the potential of CA in structural and multidisciplinary design optimization, a fairly concrete idea of the *types* of problems for which CA has a strong potential for outperforming existing approaches can be gleaned, thus identifying applications where CA can substantially alter the state of the art in design practice.

The first important feature of the CA approach is its treatment of the underlying field problem (e.g., two-dimensional elasticity). CA formulation of the field problem is based on local update rules that are applied iteratively. From that perspective, the transition from solving linear problems to solving nonlinear problems appears to be minimal, both in terms of computational cost, and algorithm design. Moreover, this allows CA to benefit from efficient acceleration techniques such as multigrid.

The second, and more important feature of CA is its design capability. In many practical applications, the design problem is to find a design function over a pre-defined domain that maximizes or minimizes a given functional, subject to various integral (functional) constraints. Such problems, classically treated by the calculus of variations, are usually handled in engineering design by discretization. The original continuous problem is broken down using finite elements into a problem with a finite number of field unknowns (usually the value of field variables at nodes), and finite number of design variables (usually physical or geometric properties of elements). After the problem is discretized, it is solved as finite dimensional optimization problem, more often than not by employing an iterative search algorithm.

a major problem with this approach is that the number of design variables prohibitively increases when a fine mesh is used to capture the detailed physics of the problem. Search algorithms become less and less efficient as the number of design variables is increased, this approach greatly limits the size of problems that can be solved.

In the CA approach, the variational problem is approached directly. The necessary optimality conditions for the problem are written and formulated in the form of local cell level optimization problems involving only the design variables of a cell. By seeking solutions to the necessary optimality conditions, the search effort for optimum design is greatly reduced. The formulation of the optimality conditions as local rules provides a mathematically rigorous way of decomposing the search of the design space. for a problem discretized using N cells, and having m design variables per cell, instead of searching an $N \times m$ dimensional design space, CA searches N m dimensional subspaces. Moreover, the variational formulation of the problem lead to algorithms that are monotonically convergent.

Thus, the applications in which CA has the greatest potential are those where a nonlinear variational problem, with an underlying nonlinear field problem is to be solved. Foremost among such applications is three dimensional topology design of thin walled structures. An initial effort in this direction has been undertaken in the area of nonlinear topology design of trusses [75, 76], where both material and geometric nonlinearities are considered. By combining advanced hardware based on Field Programmable Gate Arrays (FPGAs), local CA rules, and multigrid acceleration, the three dimensional topology design of three dimensional structures using very dense lattices is anticipated to be within reach.

Appendix A

Nonlinear Pull-in Point Prediction for Electrostatically Actuated microbeam

A.1 Finite Element Model

The beam is discretized into n elements of constant width and thickness. Such an element is schematically shown in Fig. A.1. In what follows, we use subscript i to indicate scalar quantities associated with the i^{th} element/node. Let w_i and θ_i be the nodal degrees of freedom and f_i and M_i be the generalized nodal forces. The normalized local co-ordinate ξ is related to the global co-ordinate x as:

$$\xi(x) = \frac{2x - (x_i + x_j)}{l_i} \tag{A.1}$$

where l_i is the element length.

Appendix A

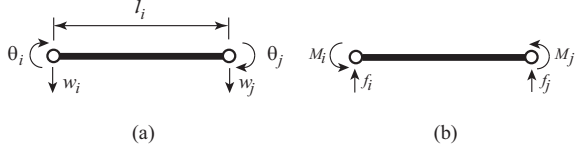


Figure A.1: (a) The element degrees of freedom. (b) The element nodal forces.

The displacement field w for the element is expressed as,

$$w(\xi) = \mathbf{N}_e \cdot \mathbf{q}_e \quad (\text{A.2})$$

where $\mathbf{N}_e = [N_1 \ N_2 \ N_3 \ N_4]^T$ is the interpolation function vector and,

$$\begin{aligned} N_1 &= \frac{1}{4}(-1 + \xi)^2(2 + \xi) \\ N_2 &= \frac{l_i}{8}(-1 + \xi)^2(1 + \xi) \\ N_3 &= \frac{-1}{4}(-2 + \xi)(1 + \xi)^2 \\ N_4 &= \frac{l_i}{8}(-1 + \xi)(1 + \xi)^2 \end{aligned} \quad (\text{A.3})$$

are the Hermitian shape functions. The element nodal displacement vector \mathbf{q}_e is:

$$\mathbf{q}_e = [w_i \ \theta_i \ w_j \ \theta_j]^T \quad (\text{A.4})$$

The equilibrium equation for the beam takes the form,

$$\mathbf{K} \cdot \mathbf{q} - \lambda \mathbf{p} = \mathbf{F} \quad (\text{A.5})$$

where \mathbf{K} is the global structural stiffness matrix, \mathbf{p} is the electrostatic load vector, \mathbf{F} is the external load vector and \mathbf{q} is the displacement vector. We write the tangent matrix \mathbf{K}_t as:

$$\mathbf{K}_t = \mathbf{K} - \lambda \mathbf{K}_L \quad (\text{A.6})$$

where the global electrostatic load matrix \mathbf{K}_L is given by,

$$\mathbf{K}_L = \frac{\partial \mathbf{p}}{\partial \mathbf{q}}$$

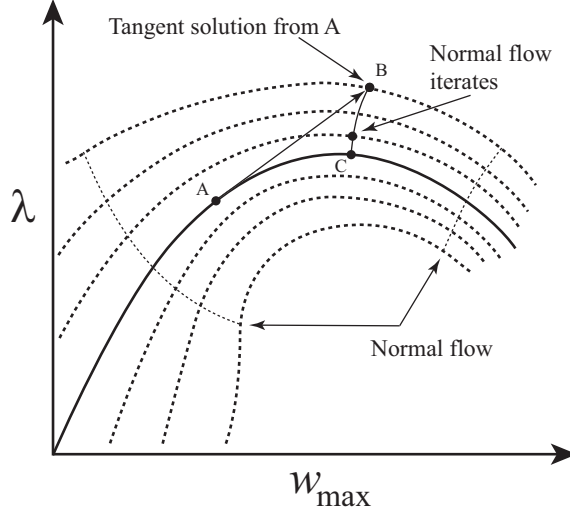


Figure A.2: Normal flow algorithm

The corresponding element equations are,

$$\begin{aligned}
 \mathbf{K}^e &= \frac{8}{l_i^3} \int_{-1}^1 I \ddot{\mathbf{N}}_e \otimes \ddot{\mathbf{N}}_e d\xi = I_i \hat{\mathbf{K}}^e \\
 \mathbf{p}^e &= \frac{b_i l_i}{2} \int_{-1}^1 \frac{1}{(1-w)^2} \mathbf{N}_e d\xi = b_i \hat{\mathbf{p}} \\
 \mathbf{K}_L^e &= \frac{b_i l_i}{2} \int_{-1}^1 \frac{1}{(1-w)^3} \mathbf{N}_e \otimes \mathbf{N}_e d\xi
 \end{aligned} \tag{A.7}$$

In the above equations, dots denote derivatives with respect to ξ . For the micro-beam under consideration, there are no external loads, i.e., $\mathbf{F} = \mathbf{0}$. The integrations in (A.7) are evaluated using a two point Gauss quadrature.

A.2 Nonlinear Response

The Normal flow algorithm [77] is used to trace the equilibrium curve and find the limit point. In the normal flow algorithm, successive Newton Raphson iterates

Appendix A

converge to the equilibrium solution along a path which is normal (in an asymptotic sense) to the so-called *Davidenko flow*. The *Davidenko flow* can be described by considering a small perturbation, δ , to the nonlinear system equations:

$$\mathbf{f}(\mathbf{q}; \lambda) = \delta \quad (\text{A.8})$$

As the perturbation parameter varies, small changes will occur in the solution curve for (A.8). The family of curves generated by varying δ is known as the *Davidenko flow*. The dashed lines in Fig. A.2 are a representative of the *Davidenko flow* for a one-dimensional problem.

In order to solve (A.8) we use the Newton-Raphson method. The Newton-Raphson iterate is given by:

$$D\mathbf{f} \mathbf{c} = -\mathbf{f}(\mathbf{q}_n; \lambda_n) \quad (\text{A.9})$$

where,

$$\mathbf{c} = [\Delta\mathbf{q} \mid \Delta\lambda]^T; D\mathbf{f} = [\mathbf{K}_t \mid -\mathbf{p}], \quad (\text{A.10})$$

and

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \Delta\mathbf{q}; \lambda_{n+1} = \lambda_n + \Delta\lambda \quad (\text{A.11})$$

The system given by (A.9) is a $m \times (m + 1)$ underdetermined system. We augment the system by the condition:

$$\mathbf{u} \cdot \mathbf{c} = 0 \quad (\text{A.12})$$

where \mathbf{u} is the kernel of $D\mathbf{f}$, partitioned as:

$$\mathbf{u} = \left[\mathbf{r}^T \mid \frac{d\lambda}{ds} \right]^T$$

where s is the arc length parameter along the equilibrium path.

An illustration of the normal flow algorithm for the solution of a one-dimensional problem is presented in Fig. A.2. Starting at a converged point on the equilibrium path, point **A**, an initial step is taken in the direction of the tangent vector to point **B**. Geometrically, the successive iterates (indicated using filled circular markers in the figure) return to the equilibrium solution along a path normal to the *Davidenko flow*. The normal flow algorithm will converge to a new point on the equilibrium path at point **C**.

The saddle-node (pull-in) is defined by the condition,

$$\frac{d\lambda}{ds} = 0 \tag{A.13}$$

The pull-in point is numerically obtained using a regula-falsi method applied to (A.13). At pull-in, we automatically obtain the mode shape \mathbf{r} which is the null-space of \mathbf{K}_t .

Bibliography

- [1] Levy, R. and Lev, O. E. Recent developments in structural optimization. *Journal of Structural Engineering*, 113(9):1939–1962, 1987.
- [2] Vanderplaats, G. N. Thirty years of modern structural optimization. *Advances in Engineering Software*, 16(1):81–88, 1993.
- [3] Sobieszczanski-Sobieski, J. and Haftka, R. T. Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization*, 14(1):1–23, 1997.
- [4] Haftka R. T. and Gürdal Z. *Elements of Structural Optimization*. Kluwer, 1993.
- [5] Sobieszczanski-Sobieski, J., Kodiyalam, S., and Yang, R. J. Optimization of car body under constraints of noise, vibration and harshness (nvh) and crash. *Structural and Multidisciplinary Optimization*, 22:295–306, 2001.
- [6] Gürdal, Z. and Tatting, B. Cellular automata for truss structures with linear and nonlinear response. In *AIAA-2000-1580, 41st AIAA/ASME/ AHS/ASC Structures, Structural Dynamics and Material Conference*, April 2000.
- [7] Ulam, S. Random processes and transformations. In *Proceedings of the International Congress of Mathematics*, volume 2, pages 85–87, April 1952.
- [8] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

- [9] Wolfram S. *Cellular Automata and Complexity: Collected Papers*. Addison-Wesley Publishing Company, 1994.
- [10] Lowekamp B.B., Watson L.T., and Cramer M.S. The cellular automata paradigm for the parallel solution of heat transfer problems. *Parallel Algorithms and Applications*, 9:119–130, 1996.
- [11] Meit, R., Luo, L., and Shyy, W. An accurate curved boundary treatment in the lattice boltzmann method. In *AIAA-99-3353*, 1999.
- [12] Slotta, D. J., Tatting, B., Watson, L. T., Gürdal, Z., and Missoum, S. Convergence analysis for cellular automata applied to truss design. *Engrg. Comput.*, 19:953–969, 2002.
- [13] *Virtex-ii Platform FPGA's: Introduction and Overview*. Xilinx, Inc., 2002.
- [14] Govindan, T., R. Space computing research in nasa's intelligent systems program. In *AIAA 2000-5101, AIAA Space 2000 Conference and Exposition, Long Beach, Ca.*, September 2000.
- [15] Singleterry, R. Jr., Sobieszczanski-Sobieski, J., and Brown, S. Field-programmable gate array computer in structural analysis: An initial exploration. In *AIAA 2002-1761, 43rd AIAA/ASME/ AHS/ASC Structures, Structural Dynamics and Material Conference, Denver, Co.*, April 2002.
- [16] Kobori, T., Maruyama, T., and Hoshino T. A cellular automata system with fpga's. In *IEEE Symposium on Field-Programmable Custom-Computing Machines*, 2001.
- [17] Paar K. J., Athanas, P. M., and Edwards C. M. Implementation of a finite difference method on a custom computing platform. In *High-Speed Computing, Digital Signal Processing, and Filtering Using Reconfigurable Logic, SPIE*, November 1996.
- [18] Gaylord, R. J. and Nishidate, K. *Modeling Nature: Cellular Automata Simulations with Mathematica*. Springer-Verlag, Inc., 1996.
- [19] Hess, R. Simulating aircraft subsystems using non-homogeneous automata networks. In *AIAA-2000-4499, AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, Co.*, August 2000.

- [20] Ricot, D. and Maillard, V. Bailly, C. Numerical simulation of unsteady cavity flow using lattice boltzmann method. In *AIAA 2002-2532, 8th AIAA/CEAS Aeroacoustics Conference and Exhibit, Breckenridge, Co.*, June 2002.
- [21] Kita, E. and Toyoda, T. Structural design using cellular automata. *Structural and Multidisciplinary Optimization*, 19:64–73, 2000.
- [22] Xie, Y. M. and Steven, G. P. *Evolutionary Structural Optimization*. Springer-Verlag, Berlin, 1997.
- [23] Zhou, M. and Rozvany, G., I., N. On the validity of eso type methods in topology optimization. *Structural and Multidisciplinary Optimization*, 21:80–83, 2001.
- [24] Sigmund, O. A 99 line topology optimization code written in matlab,” structural and multidisciplinary optimization. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.
- [25] Hajela, P. Implications of artificial life simulations in structural analysis and design. In *AIAA-1998-1775, 39th AIAA/ASME/ AHS/ASC Structures, Structural Dynamics and Material Conference, Long Beach, Ca.*, April 1998.
- [26] Hajela, P. and Kirn, B. On the use of energy minimization for ca based analysis in elasticity. In *AIAA-2000-2003, 41st AIAA/ASME/ AHS/ASC Structures, Structural Dynamics and Material Conference, Atlanta, Ga.*, April 2000.
- [27] Hajela, P. and Kim, B. On the use of energy minimization for ca based analysis in elasticity. *Structural and Multidisciplinary Optimization*, 23:24–33, 2001.
- [28] Zhou, M. and Haftka, R., T. A comparison of optimality criteria methods for stress and displacement constraints. In *AIAA paper No. 94-1364-CP*, 2000.
- [29] Tatting, B. and Gürdal, Z. Cellular automata for design of two-dimensional continuum structures. In *AIAA-2000-4832, Presented at the 8th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*,, September 2000.
- [30] Abdalla, M. M. and Gürdal, Z. Structural design using cellular automata for eigenvalue problems. *Structural and Multidisciplinary Optimization*, 2003.
- [31] Tadjbakhsh, I. and Keller J. B. Strongest columns and isoparametric inequalities for eigenvalues. *Journal of Applied Mechanics*, 29:159–164, 1962.

- [32] Trahair, N., S. and Booker, J., R. Optimum elastic columns. *International Journal of Mechanical Science*, 12:973–983, 1970.
- [33] Olhoff, N. and Rasmussen, S. On single and bimodal optimum buckling loads of clamped columns. *International Journal of Solids and Structures*, 13:605–614, 1970.
- [34] Szyszkowski, W., Watson, L., G., and Fietkiewicz, B. On single and bimodal optimum buckling loads of clamped columns. *Computers and Structures*, 32:1093–1104, 1989.
- [35] Gajewski, A. and Zyczowski, M. *Optimal Structural Design under Stability Constraints*. Kluwer Academic Publishers, 1989.
- [36] Canfield, R., A. Design of frames against buckling using a rayleigh quotient approximation. *AIAA Journal*, 31:1143–1149, 1993.
- [37] Shin, Y., S. and Haftka R., T. and Plaut R., H. Simultaneous analysis and design for eigenvalue maximization. *AIAA Journal*, 26:738–744, 1988.
- [38] Ishida, R. and Sugiyama, Y. Proposal of constructive algorithm and discrete shape design of the strongest column. *AIAA Journal*, 33:401–406, 1995.
- [39] *GENESIS User Manual*. Vandeplaats Research & Development, 1998.
- [40] Lam, Y.C., Manickarajah, D., and Bertolini, A. Performance characteristics of resizing algorithms for thickness optimization of plate structures. *Finite Elements in Analysis and Design*, 34:159–174, 2000.
- [41] Yin, L. and Yang W. Optimality criteria method for topology optimization under multiple constraints. *Computers and Structures*, 79(5):1839–1850, 2001.
- [42] Zhou, M. and Rozvany, G., I., N. An optimality criteria method for large systems. part i: Theory. *Structural Optimization*, 5:12–25, 1992.
- [43] Zhou, M. and Rozvany, G., I., N. An optimality criteria method for large systems. part ii: Algorithm. *Structural Optimization*, 6:250–262, 1996.
- [44] Zhou, M. and Rozvany, G., I., N. An improved approximation technique for the dco method of sizing optimization. *Computers and Structures*, 60(5):763–769, 1996.

- [45] Bendsøe, M., P. and Kikuchi, N. Generating optimal topologies in optimal design using a homogenization approach. *Comp. Meth. Appl. Mech. Engng.*, 71:197–224, 1988.
- [46] 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.
- [47] Sigmund O. and Petersson J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16:68–75, 1998.
- [48] Rozvany, G., I., N. Stress ratio and compliance based methods in topology optimization—a critical review. *Structural and Multidisciplinary Optimization*, 21:109–119, 2001.
- [49] Diaz, A. and Sigmund, O. Checkerboard patterns in layout optimization. *Structural and Multidisciplinary Optimization*, 10:40–45, 1995.
- [50] Abdalla, M. M. and Gürdal, Z. Structural design using optimality based cellular automata. In *AIAA-2002-1676, 43th AIAA/ASME/ AHS/ASC Structures, Structural Dynamics and Material Conference, Denver, Co*, April 2002.
- [51] Setoodeh, S., Abdalla, M. M., and Gürdal, Z. Combined topology and fiber path design of composite layers using cellular automata. *Structural and Multidisciplinary Optimization*, under review.
- [52] Maar, O. and Schulz, V. Interior point multigrid methods for topology optimization. *Structural and Multidisciplinary Optimization*, 19:214–224, 2000.
- [53] Salam Rahmatalla and Colby C. Swan. Continuum topology optimization of buckling-sensitive structures. *AIAA Journal*, 41:1180–1189, 2003.
- [54] Golub, G. and Ortega, J., M. *Scientific Computing, an Introduction with Parallel Computing*. Academic Press, 1993.
- [55] Abdalla, M., Reddy, K., Faris, F., and Gürdal, Z. Optimal design of an electrostatically actuated microbeam for maximum pull-in voltage. *Computers and Structures*, Accepted for publication.
- [56] S. Senturia. *Microsystem Design*. Kluwer, 2001.

- [57] Seeger, J., I. and Crary, S., B. Stabilization of electrostatically actuated mechanical devices. *Transducers*, pages 1133–1136, 1997.
- [58] J.A. Pelesko and A.A. Triolo. Nonlocal problems in mems device control. *Journal of Engineering Mathematics*, 41:345366, 2001.
- [59] D. Haronian. Maximizing microelectromechanical sensor and actuator sensitivity by optimizing geometry. *Sensors and Actuators A*, 50:223–236, 1995.
- [60] Lai, J., Perazzo, T., and Majumdar, A. Shi, Z. Optimization and performance of high-resolution micro-optomechanical thermal sensors. *Sensors and Actuators A*, pages 113–119, 1997.
- [61] M. Brenner, J. Lang, J. Li, J. Qiu, and A. Slocum. Optimum design of a mems switch. In *5th International Conf. on Modeling and Simulation of Microsystems*, U.S.A., April 2002.
- [62] S. Nishiwaki, K. Saitou, S. Min, and N. Kikuchi. Topological design considering flexibility under periodic loads. *Structural and Multidisciplinary Optimization*, pages 4–16, 2000.
- [63] O. Sigmund. Design of multiphysics actuators using topology optimization - part i: One-material structures. *Computer methods in applied mechanics and engineering*, pages 6577–6604, 2001.
- [64] A. H. Nayfeh and B. Balachandran. *Applied Nonlinear Dynamics*. Wiley, 1995.
- [65] Setoodeh, S., Adams, D., B., Gürdal, Z., and Watson, L. T. Pipeline implementation of cellular automata for structural design on message-passing multiprocessors. *Mathematical and Computer Modeling*, to appear.
- [66] Setoodeh, S and Gürdal, Z. Curvilinear fiber design of composite laminate using cellular automata. In *AIAA-2003-2003, 44th AIAA/ASME/ AHS/ASC Structures, Structural Dynamics and Material Conference, Norfolk, Va, April 2003*.
- [67] Hartka, T., Jones, M., Gürdal, Z., and Mostafa, M. A reconfigurable approach to structural engineering design computations. *in preparation*, 2004.

- [68] Jones, M., Plassmann, P., and Gürdal, Z. Efficient algorithms for iterative refinement based on configurable precision arithmetic. *Submitted for publication in SIAM Journal On Matrix Analysis and Applications.*, 2004.
- [69] Abdalla, M. M., Kim, S., and Gürdal, Z. 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, AIAA paper 2004-1644, Palm Springs, California*, April 2004.
- [70] Hackbusch, W. *Multi-grid Methods and Applications*. Springer-Verlag, Berlin, New York, 1985.
- [71] Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons Ltd., 1992-Corrected Reprint. Philadelphia: R. T. Edwards, Inc., 2004.
- [72] Tovar, A., Niebur, G., L., Sen, M., and Renaud, J., E. Bone structure adaptation as a cellular automaton optimization process. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Palm Springs, California*, April 2004.
- [73] Tovar, A., Patel, N., Kaushik A., K., Letona, G., A. Renaud, J., E., and Sanders, B. Hybrid cellular automata: a biologically-inspired structural optimization technique. In *AIAA 2004-4558, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York*, September 2004.
- [74] Canyurt, O. and Hajela, P. A sand approach based on cellular computation models for analysis and optimization. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Palm Springs, California*, April 2004.
- [75] Missoum, S, Abdalla, M. M., and Gürdal, Z. Nonlinear topology design of trusses using cellular automata. In *AIAA-2003-1445, 44th AIAA/ASME/AHS/ASC Structures, Structural Dynamics and Material Conference, Norfolk, Va*, April 2003.
- [76] Missoum, S, Abdalla, M. M., and Gürdal, Z. Nonlinear design of trusses under multiple loads using cellular automata. In *5th World Congress in Structural and Multidisciplinary Optimization, Lido di Jesolo, Italy*, May 2003.

- [77] S. A. Ragon, Z. Gürdal, and L. T. Watson. A comparison of three algorithms for tracing nonlinear equilibrium paths of structural systems. *Solids and Structures*, pages 689–698, 2002.

Curriculum Vitae

Mostafa M. Abdalla was born on March 1975 in Cairo Egypt. He finished his high school education in Cairo, where he later studied at the faculty of engineering at Cairo University majoring in aerospace engineering. After graduating with distinction in 1997, he joined the staff at the aerospace department in the same university as a teaching assistant, where he obtained his M. Sc. with distinction. In his M. Sc. thesis, he developed an analytic model for thin walled composite beams. His Ph. D. work focused on applying new massively parallel Cellular Automata algorithms to structural design with applications to microelectrical systems and topology design.