DESIGN SYNTHESIS FOR MORPHING 3D MESO-SCALE STRUCTURE

A Thesis
Presented to
The Academic Faculty

by

Chen Chu

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2009

DESIGN SYNTHESIS FOR MORPHING 3D MESO-SCALE STRUCTURE

Approved by:

Dr. David Rosen, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Seung-Kyum Choi
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Richard Hague
School of Mechanical and Manufacturing Engineering
Loughborough University

Date Approved: May 15, 2009

ACKNOWLEDGEMENTS

I thank my advisor, Dr. David Rosen, for his helpful guidance and knowledge throughout my graduate school years that made this thesis possible. I also thank my other reading committee members, Dr. Richard Hague and Dr. Seung-Kyum Choi, for taking time out of their work to provide suggestions and expertise.

I would also like to thank all the members of the Georgia Tech Systems Realization Laboratory for erudition and friendship. I especially thank Greg Graf and Sara Engelbrecht, whom I had worked closely with and obtained help from in the development of my research and thesis.

I would like to thank many friends and old classmates who offered help and support when hardship came. They made me realize what true friend is.

I want to thank also my parents for being thoughtful and supportive. It is through their comforting and unconditional love that I could go on.

Last, I thank God for his mercy and keeping power that I still open to him and remain in the divine family.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS....iii

LIST OF TABLES....vii

LIST OF FIGURES....viii

LIST OF SYMBOLS AND ABBREVIATIONS....x

SUMMARY....xi

Chap.1 Background and Motivation....1

1.1 Background....1

1.2 Motivation....2

1.3 Problem statement....4

1.4 Research Questions and Hypotheses....5

1.4.1 Question and Hypothesis 1....5

1.4.2 Question and Hypothesis 2....6

1.5 Organization of thesis....6

Chap.2 Literature Review and Gap Analysis....10

2.1 Cellular Structure....10

2.2 Structural Analysis....10

2.3 Compliant Mechanism....12

2.4 Structural Optimization....14

2.5 Optimization techniques....14

2.6 Optimization methods....17

2.7 Gap Analysis....18

2.8 Summary....19

Chap.3 Problem Formulation....20

3.1 Primitive Instancing....20

3.1.1 Introduction....20

3.1.2 Mesh....20

3.1.3 Different Primitives....25

3.2 Problem Formulation....26

3.2.1 Strut Approach....26

3.2.2 Parametric Surface/Solid Approach....29

3.3 Summary....37

Chap.4 Optimization Methods....39

4.1 Particle Swarm Optimization....39

4.1.1 Swarm Size....41

4.1.2 Inertia Weight 43
4.1.3 Cognitive and Social Acceleration 45
4.2 Least Squares Minimization Formulation 45
4.2.1 Limitations 47
4.2.2 Initial Design 47
4.3 Comparison of PSO and LM 48
4.3.1 Pros and Cons 56
4.3.2 More Efficient Algorithm 57
4.4 Summary 58
Chap.5 Examples 59
5.1 A Simple 3D Cantilever Beam Example 59
5.1.1 Strut Approach 61
5.1.2 Parametric Solid Approach 63
5.1.3 Summary 64
5.2 A more complicated 3D Cantilever Beam Example 66
5.2.1 Strut Approach 68
5.2.2 Parametric solid approach 70
5.2.3 An alternative assignment of parametric value 72
5.2.4 Summary 73
5.3 A 2D Beam Example 75
5.3.1 Strut Approach 77
5.3.2 Parametric Surface Approach 80
5.3.3 Change of objective function 82
5.3.4 Summary 83
5.4 Wing Example 85
5.4.1 2D Airfoil 85
5.4.2 3D Wing 92
5.5 Optimization time 98
5.6 Summary 100
Chap.6 Closure 102
6.1 Summary 102
6.2 Conclusion 105
6.2.1 Base truss topology 105
6.2.2 Shape-Change Compliant Mechanism 106
6.2.3 Number of design variables 106
6.2.4 Parametric Surface/Solid Approach 107

6.2.5 Conclusions 109

6.3 Contribution 110

6.3.1 Level of optimization 110

6.3.2 Optimization methods 110

6.4 Future Work 111

6.4.1 Optimization methods 111

6.4.2 Topology and shape optimization 112

6.5 Closure 112

APPENDIX I 114

APPENDIX II 122

REFERENCES 125

LIST OF TABLES

Table 4.1: Parameters used for PSO swarm size experiment 42
Table 4.2: Parameters used for PSO inertia weight experiment 44
Table 4.3: Results of decreasing inertia weight 44
Table 4.4: Results of increasing inertia weight 45
Table 4.5: Results of experiments to compare PSO and LM 53
Table 4.6: Results of experiments to compare PSO with multiple runs of LM 57
Table 5.1: Parameters used for first cantilever beam example 62
Table 5.2: Summary of first 3D cantilever beam example 65
Table 5.3: Parameters used for second cantilever beam example 69
Table 5.4: Results of alternative parametric approach 73
Table 5.5: Summary of second cantilever beam example 75
Table 5.6: Comparison of starting configuration 77
Table 5.7: Parameters used for 2D beam example 79
Table 5.8: Graf’s optimization result for 2D beam example using strut approach 79
Table 5.9: Result of 2D beam example using strut approach with quadrilateral model 80
Table 5.10: Result of 2D beam example using parametric surface approach 81
Table 5.11: Result of 2D cantilever beam with a modified objective function 83
Table 5.12: Summary of 2D beam example 84
Table 5.13: Parameters used for 2D airfoil example 89
Table 5.14: Results of 2D airfoil example 90
Table 5.15: Parameters used to generate 3D wing 93
Table 5.16: Parameters used for 3D wing example 95
Table 5.17: Summary of 3D wing example 98
Table 5.18: Relationship between number of design variables and number of function calls 99

LIST OF FIGURES

Figure 1.1: A part made by FDM 2
Figure 1.2: An example of lattice structure 4
Figure 1.3: Organization of thesis 7
Figure 2.1: Octet-truss unit cell 11
Figure 2.2: Series of three unit truss structures that are connected between each node ... 12
Figure 2.3: An example of compliant mechanism 13
Figure 3.1: An airfoil example of FEM software generated mesh 22
Figure 3.2: Primitive Instancing Example 24
Figure 3.3: A simple example utilizing mapped mesh 25
Figure 3.4: Primitive for morphing structure 26
Figure 3.5: Primitive for stiff structure 26
Figure 3.6: Word formulation for strut approach 27
Figure 3.7: Math formulation for strut approach 29
Figure 3.8: Bezier surface illustration 31
Figure 3.9: Assigning Parametric Value 33
Figure 3.10: Relation between control diameters and unit cells 34
Figure 3.11: Word formulation for parametric surface/solid approach 35
Figure 3.12: Math formulation for parametric surface/solid approach 37
Figure 4.1: Problem 1 base truss topology 49
Figure 4.2: Problem 2 base truss topology 49
Figure 4.3: Problem 3 base truss topology 50
Figure 4.4: Problem formulation for PSO and LM comparison 51
Figure 4.5: Result structure of problem 1 by LM 54
Figure 4.6: Result structure of problem 2 by LM 54
Figure 4.7: Result structure of problem 3 by LM 54
Figure 4.8: Result structure of problem 1 by PSO 55
Figure 4.9: Result structure of problem 2 by PSO 55
Figure 4.10: Result structure of problem 3 by PSO 55
Figure 5.1: First 3D cantilever beam example 60
Figure 5.2: Base truss topology for first cantilever beam example 61
Figure 5.3: Problem formulation of 3D cantilever beam example strut approach 61
Figure 5.4: Graf's result for strut approach using LM/LSM 62
Figure 5.5: Problem formulation of first 3D beam example parametric solid approach 63
Figure 5.6: Parametric solid approach resulting structure 64

Figure 5.7: Second 3D cantilever beam example 67
Figure 5.8: Base truss topology for second cantilever beam example................................. 67
Figure 5.9: Relative stress distribution for second cantilever beam example 68
Figure 5.10: Problem formulation for second cantilever beam example with strut approach 69
Figure 5.11: Resulting structure of second cantilever beam example using strut approach 70
Figure 5.12: Problem formulation for second cantilever beam example with parametric solid approach 70
Figure 5.13: Resulting structure of second cantilever beam example using parametric solid approach 71
Figure 5.14: Alternative approach result structure for second beam example 73
Figure 5.15: Base truss topology used by Qi Xia and Yu Wang 76
Figure 5.16: Base truss topology for 2D beam used by Graf 77
Figure 5.17: Base truss topology for 2D beam example parametric surface approach 77
Figure 5.18: Graf’s resulting structure for 2D beam example using strut approach............. 79
Figure 5.19: Resulting structure of 2D beam example using strut approach with quadrilateral model 80
Figure 5.20: Resulting structure of 2D beam example using parametric surface approach . 81
Figure 5.21: Resulting structures for 2D beam example with modified objective function. 83
Figure 5.22: Airfoil cross section.. 85
Figure 5.23: Initial truss topology of 2D airfoil example 86
Figure 5.24: Problem formulation for 2D airfoil example 87
Figure 5.25: The 3x1 parametric surface used for 2D airfoil example 4x2 parametric approach 88
Figure 5.26: 2D airfoil starting truss topology 91
Figure 5.27: Resulting structure of 2D airfoil example using strut approach........................ 91
Figure 5.28: Resulting structure of 2D airfoil example using unit cell approach.................. 91
Figure 5.29: Resulting structure of 2D airfoil example using 3x2 parametric approach 92
Figure 5.30: Resulting structure of 2D airfoil example using 4x2 parametric approach 92
Figure 5.31: Structure of wing example at each cross section 93
Figure 5.32: Base truss topology of wing example 94
Figure 5.33: Problem formulation for 3D Wing example 94
Figure 5.34: Starting truss topology profile comparing to target profile 95
Figure 5.35: Resulting structure of 3D wing example using strut approach 96
Figure 5.36: Resulting structure of 3D wing example using parametric solid approach....... 97

LIST OF SYMBOLS AND ABBREVIATIONS

RP	Rapid Prototyping
MP	Mathematical Programming
AM	Additive Manufacturing
FDM	Fused Deposition Modeling
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
LM	Levenburg-Marquardt
FEM	Finite Element Modeling
cDSP	Compromise Decision Support Problem
LSM	Least-squares Minimization Methods
ρ	Relative Density
ψ_i	Interpolation Functions
δ	Deflection
vS	Von Mises Stress
yS	Yield Stress
Vol	Normalized Total Volume
w	Weighting Value
d_i	Diameter of Struts
d_{ij}/d_{ijk}	Control Diameters
D_{abc}	Diameter in unit cell
c_1	Cognitive Parameter
c_2	Social Parameter
φ	Sum of Cognitive and Social Parameter
c_2	Social Parameter
$\mathbf{J}$	Jacobian
μ^k	Scalar Damping Parameter
RepDefl	Difference Between Actual and Desired Deflection

SUMMARY

Rapid prototyping (RP) refers to manufacturing technologies that quickly produce parts from 3D data using an additive approach, as opposed to traditional machining process. It can be used to make complex shapes with very little or even no constraint on the form of the parts. New design methods are needed for parts that can take advantage of the unique capabilities of RP and thus expand the usage and stimulate the development of RP technology. Although current synthesis methods can successfully solve simple design problems, practical applications with thousands to millions elements are prohibitive to generate solution for.

Two factors are considered. One is the number of design variables; the other is the optimization method. To reduce the number of design variables, parametric approach is introduced. Control diameters are used to control all strut size across the entire structure by utilizing a concept similar to control vertices and Bezier surface. This operation allows the number of design variables to change from the number of elements to a small set of coefficients.

In lattice structure design, global optimization methods are popular and widely used. These methods use heuristic strategies to search the design space and thus perform, as oppose to traditional mathematical programming (MP) methods, a better global search. This work propose that although traditional MP methods find local optimum near starting point, given a quick convergence rate, it will be more efficient to perform such method multiple times to integrate global search than using a global optimization method. Particle Swarm Optimization and Levenburg-Marquardt are chosen to perform the experiments.

The effectiveness of proposed approaches is tested through several examples, including both structural design and compliance mechanism design examples. The proposed method does greatly reduce computational time related to optimization. Some limitation of parametric approach existing in its nature is identified and discussed.

Chap.1 *Background and Motivation*

1.1 *Background*

Additive manufacturing (AM), or additive fabrication, refers to a group of processes that build up parts by adding material, often in layers. The advances in the equipment and materials make possible not only improvements in the “form, fit, and function” of so-called rapid prototypes, but also allow these processes to be used to produce a wider range of production parts [1]. Rapid prototyping (RP) technologies are the most widely applied and known fabrication methods that are based on additive fabrication principles [2].

Compared to conventional manufacturing methods, AM technologies advantage the manufacturing processes in several aspects. Instead of removing materials to obtain the part desired, constructing it by adding material reduce material required. AM allows the construction of parts that were formerly impossible to make. An example is presented in Figure 1.1. This part was made by Expedio Solutions SDN BHD [3] using Fused Deposition Modeling (FDM), which is a solid-based prototyping method that extrudes material layer-by-layer. The ability of making complicated parts gives designers the freedom to designing with less apprehension over the inability that exists in manufacturing methods. The utilization of these technologies in the late 1980s has enabled direct manufacturing from computer model to physical part. With 3D scanning technologies now available, a lot of parts that traditionally involve labor-intensive processes of molding and casting to create are now as easy as scanning and manufacturing [4]. Since it is driven by digital information, the skills and time needed for multi-part design, manufacturing and assembly are eliminated, human error opportunity can also be reduced [5].

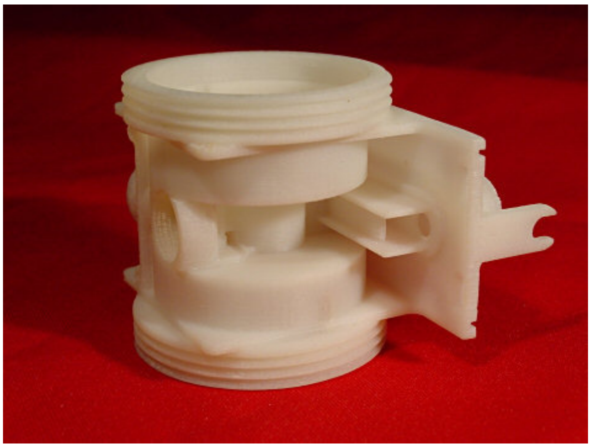


Figure 1.1: A part made by FDM

1.2 *Motivation*

Design methodology that exploits the design freedom given by AM is needed. To utilize the advantage brought forth by the introduction of AM, one of the applications is the use of meso-scale cellular structure. Since the advent of rapid manufacturing technology has made possible the manufacturing of cellular material, it is becoming increasingly desirable because it usually offers higher strength-to-weight ratio, good energy absorption characteristics and acoustic insulation properties [6, 7]. Cellular material includes foams, honeycombs, lattice and other similar constructions. It is defined as mesostructured materials when the characteristic lengths of the cells are in the range of 0.1 to 10mm [8]. They are of a scale which falls between the macro-scale of part geometry and the micro-scale of the base material. They often act as reinforcement within a large part [4].

Lattice structures consist of small struts as the example shown in Figure 1.2. Each strut can be placed and sized in a given space to design lattice structure. Current methodologies for designing these structures utilize optimization processes to automatically find an optimal design. For meso-scale lattice structure, a space as small as 15x15x15cm might already need up to thousands of struts. This makes the design space large, nonlinear and contains a lot of local optimum. The difficulty of finding optimal design increases exponentially with the number of struts. Genetic algorithms (GA) and particle swarm optimization (PSO) have been developed to execute both local and global search. The stochastic characteristic of these algorithms enables a more thorough search of the design space, but they also give different solution every time it is implemented. Also, the time that they require to find a solution for a rather modest design is prohibitive [9].

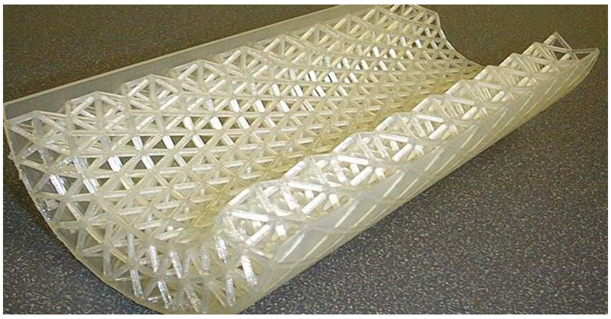


Figure 1.2: An example of lattice structure

1.3 *Problem statement*

In this work, design synthesis method for compliant cellular lattice structure is chosen to be the topic. The focus is on using strut size optimization to achieve assigned shape-change. Extending the work from [10], this thesis will attempt to improve it and include 3D structures as it is more applicable to real life design problem.

Current design synthesis method for lattice structure can be impractical for solving even a modest design problem. Thus, a less computational costly design method is needed. During the optimization process, the number of time needed for objective function value to be calculated is greatly related to the number of design variables. The calculation of objective function requires the entire structure to be analyzed through Finite Element Analysis code, which not only takes a considerable time, but also has no analytical gradient and/or hessian matrix available.

Two approaches are considered. One is the reduction of the design domain. By having a smaller number of design variables, the process of searching for an optimum can be shortened thus reducing the number of objective function evaluations and the time needed. A parametric approach is developed. It is applied to compliant mechanisms. For validating and testing purposes, structural examples are also used. Most of these structural examples are duplicated from the literature so that results from this work can be validated against the solutions from the literature. The other is the use of a more efficient optimization method. By formulating the achievement of target values of goals as a least-squares regression problem [9], Levenburg-Marquardt (LM) can be utilized which detail will be presented in chapter 4.

1.4 *Research Questions and Hypotheses*

1.4.1 *Question and Hypothesis 1*

Question 1: "How can a cellular structure be designed so that it responds to certain actuators and morphs to a desired shape?"

Hypothesis 1: A cellular structure that responds to given actuators by morphing to the desired shape can be designed by filling the initial shape with unit cells composed of struts to obtain a starting lattice structure, then through optimization to determine the thickness of each strut.

In chapter 3 and 4, each step of the methodology related to this hypothesis will be presented and described. Then the hypothesis will be further validated through a morphing wing example to show that the methodology proposed is doable and the shape change is attainable.

1.4.2 *Question and Hypothesis 2*

Question 2: "How can the number of design variables be reduced without changing the nature of strut sizing design problems?"

Hypothesis 2: The number of design variables can be reduced by changing the design variables from diameters of struts to a small number of coefficients that control the distribution of strut sizes across the structure using parametric surface/solid formulations.

This hypothesis will be validated mainly through chapter 3 by illustrating how the formulation is consummated and used. The benefit and limitation of such operation is then displayed and discussed in chapter 5 through multiple examples.

1.5 *Organization of thesis*

An overview of this thesis is presented in Figure 1.3.

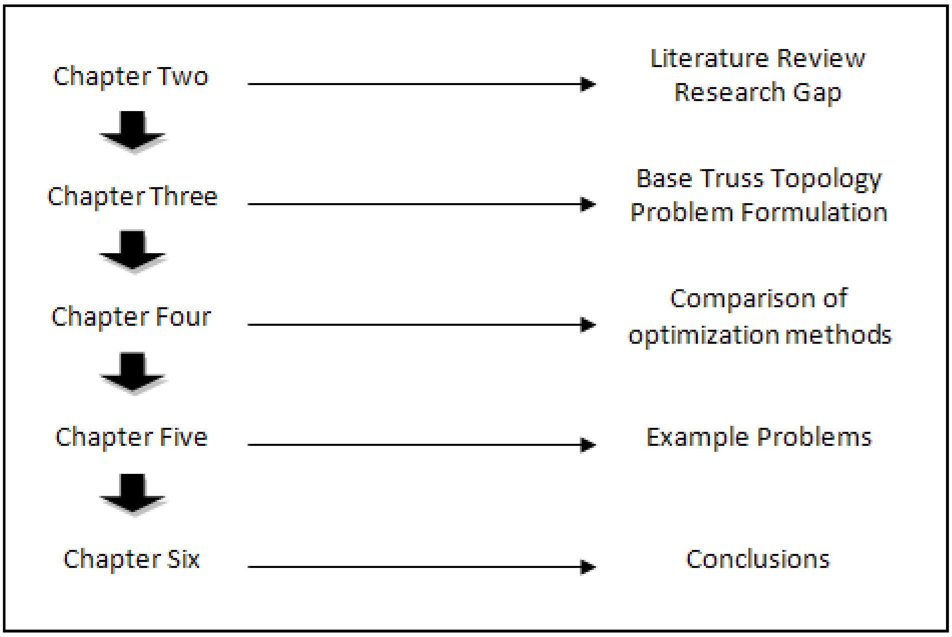


Figure 1.3: Organization of thesis

Chapter two contains a literature review of works that are related to this thesis. First a brief overview of cellular structures and the reasons behind choosing designed lattice structure to focus on is presented. Then several available analysis methods are shown, and the choice of using unit truss approach to model the behavior of truss structure accurately is explained. Last we show the design of compliant mechanisms, an overview of structural optimization and the shortcoming in the common optimization procedure shared in these synthesis methods.

Chapter three provides the synthesis method used in this thesis. Base truss topologies are obtained by inserting primitives into unit cells that are divided from the design domains. A new design synthesis method, parametric surface/solid approach, is formed by utilizing a set of coefficients that are called control diameters as design variables instead of diameters of all the struts. In this way, the number of design variables can be greatly reduced, thus reduced the time needed for optimization.

Chapter four compares the efficiency of using PSO and LM in terms of number of function evaluations and the resulting objective function value. Three simple beams are used as design problem examples. The results show that although LM converges fast and often to local optimum, adding the stochastic characteristic by using randomly generated starting points can ensure LM finding better solution than PSO and still faster.

Chapter five demonstrates the advantages and limitations of the parametric surface/solid approach using several examples. The first two examples are 3D cantilever beam examples. One is a duplication example from [4]. The other is an extension, a more complex version of it, so that the advantages of parametric solid approach can be observed. The third example is a 2D, three point bending beam, which is also a duplication example. The final example is a wing design example as an extension of [10]. This example will be used to show the proposed synthesis method both in solving 2D and 3D problem.

Chapter six will conclude the studies in this thesis, summarize the synthesis method, show both the pros and cons of using the parametric surface/solid approach and also discuss future work possible.

Chap.2 ***Literature Review and Gap Analysis***

2.1 ***Cellular Structure***

Cellular structures can be categorized by the nature of placement of internal voids into stochastic and ordered cellular materials [8].

Stochastic cellular materials (e.g. foam materials), as the name implies, are produced using stochastic processes. These materials' characteristics can be controlled, but not explicitly defined. The designing of these structures is faster and relatively low-cost [4, 6] . Stochastic materials suffer from the lack of freedom provided to the designers in terms of the topology of the mesostructure [11].

Designed or ordered cellular structures, on the other hand, have a more time-consuming designing process, but give higher stiffness and strength compared to stochastic cellular structures [12]. When the strength of a lattice structure scales as ρ, with the same relative density ρ, the strength of foam scales as $\rho^{1.5}$ [13]. Thus, given a relative density $\rho = 0.1$, a lattice structure will be approximately three times stronger than a corresponding foam structure. For the advantage stated above, this thesis will focus on designed lattice structures.

2.2 ***Structural Analysis***

Ashby and co-workers wrote a book on metal foam design and analysis [14]. They and others have applied similar methods in lattice structure analysis. The octet truss in Figure 2.1 has been extensively analyzed. Deshpande et al. [13] consider the struts in the octet truss unit cells as tension and compression bars that are pin-jointed. This often under-predicts the strength and stiffness of the structure due to the assumption of pin-jointed vertices; and it does not capture bending, shear, etc [15]. Wang and McDowell [16] extended the study and included some other lattice cells.

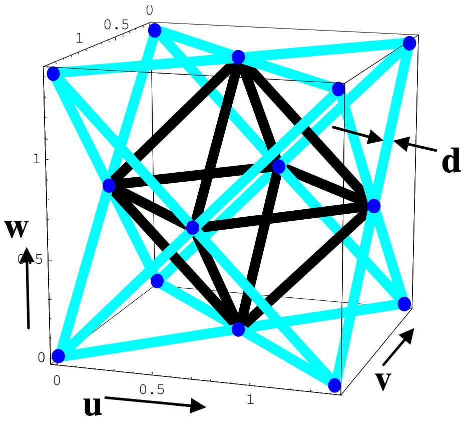


Figure 2.1: Octet-truss unit cell

A more general analytical model of lattice behavior was developed by Johnson et al. [15]. This general lattice model use each vertex and a set of half struts it connects as a unit cell like that shown in Figure 2.2. The analysis used a concept similar to the finite element method’s frame element. The unit-truss finite element program in MATLAB has been developed [17]. When combined with the tangent stiffness method, truss structures can be analyzed under axial forces, bending, torsion, nonlinearity, and buckling [18]; thus, the unit truss approach will be used throughout this thesis as the analysis method to more closely model the behavior of truss structures.

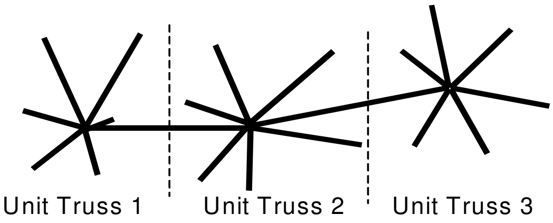


Figure 2.2: Series of three unit truss structures that are connected between each node

2.3 *Compliant Mechanism*

Compliant mechanisms are single-piece flexible structures that generate desired motion by undergoing elastic deformation as opposed to rigid linkages or joints of conventional mechanisms. An example is shown in Figure 2.3 [19]. The advantages related to using compliant mechanisms are many. They do not have the backlash error and have lower production and maintenance costs associated with the multiple piece assembly since it is hingeless [20]. Also, the smooth deformation field reduces stress concentrations. Especially for airfoils, without the connecting hinges to create discontinuities over the wing surface, earlier airflow separation can be avoided [21].

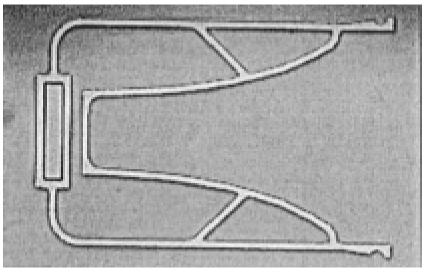


Figure 2.3: An example of compliant mechanism

Although there exist other design synthesis methods that exclude optimization process [4], when considering compliant mechanisms, which, instead of maximizing or minimizing compliance, often requires the structure to have specific deflection or shape, most methods share common procedures in designing these structures. They are: First, obtain a base topology, which is an initial guess of the design, boundary condition and other information; then, utilize optimization methods to find the optimal design.

The design and synthesis with systematic methods of compliant mechanisms have been developed and studied. Ananthasuresh et al. developed a continuum-based approach that uses the techniques of structural optimization and the homogenization method [22, 23]. Sigmund [24] and Larsen et al. [25] focused on using structural optimization to design mechanisms. Methods developed by Midha and his associates used kinematic techniques such as graph theory, Burmester theory, and pseudo rigid-body model, where the compliant mechanism behavior is modeled with a combination of rigid links and torsional springs [26]. Frecker et. al [26] used topology optimization to attain the force-deflection relationship desired. Many works have since tried to include both topology optimization and shape/size optimization.

2.4 *Structural Optimization*

Structural optimization refers to optimizing the parameters of a structure to achieve the required performance. Available methods may be subdivided into two categories called analytical methods and numerical methods. Analytical methods emphasize the conceptual aspect while numerical methods are concerned mainly with the algorithmical aspect [27].

Analytical methods employ mathematical theories in studies of optimal layouts or geometrical forms of simple structural components. They are usually not intended to handle larger structural systems [27], thus are not suitable for lattice structure design.

Numerical methods utilize mathematical programming. These usually start with an initial guess. Through iterative search of better design, an optimal design is considered to be found when the stopping criteria are reached. These methods are used and called optimization methods in the following sections of this thesis.

2.5 *Optimization techniques*

Optimization techniques are named differently in various works [4]. In general, design variables usually are considered from two aspects, the topology aspect and dimensional aspect. Take a one-piece elastic body as an example, the topology can be defined as the number of holes inside the exterior boundary, and the dimensions would be the size of these holes [21]. Topology optimization deals with the topology aspect of the design whereas pure size optimization deals only with the dimensional aspect. While topology and size optimization are more specifically defined and commonly agreed on, shape or geometry optimization is more of an ambiguity. By Rozvany [28], topology refers to spatial sequence or connectivity of members or elements, geometry indicates the location of intersections of member axes and size denotes the cross-sectional dimensions. While it is harder to define for continuum, most work relate the shape/geometry aspect as dimensional for truss structures, since the location of nodes change the length of elements and can be considered as dimensional aspect [20, 21, 29].

Many works now implement all aspects sequentially or simultaneously [20, 28, 30]. Typical synthesis methods employ a two-step approach to implement topology optimization first, then the size/shape optimization. But as Lu and Kota stated in [20], for a shape changing design task, the problem depends greatly on the precise deformation (direction and magnitude) of all or several discrete output points along the shape-changing boundary. With multiple output points, it might be inappropriate to use the two-step approach that focuses on the performance of only one particular output point. Thus in their work, the two aspects are addressed simultaneously [21].

Typical topological design optimization techniques include ground truss (discrete) approach, the homogenization (continuum) method and the material density method (continuum) [29]. Since topology optimization is intrinsically a discrete optimization problem, [29], and using discrete optimization algorithms can be unstable [31]. These three are based on transforming the discrete problem into a continuous one by using continuous design variables.

The material density method used a material density function ρ as the design variable. Using $0 < \rho < 1$ when 0 corresponds to a void and 1 to solid, the on-off material distribution problem is converted to a sizing problem. The homogenization method associates the material density with the elasticity tensor that is required for the finite element analysis [32]. It considers a mechanical element as a body occupying a subdomain Ω^{m}, which is a finite section of the entire domain Ω. The goal is to find the optimal elasticity tensors over the domain Ω of statically admissible stress field. The ground truss approach finds the optimum topology in subsets of the ground truss which is a complete graph of struts among all nodes. The design variables are the cross-sections of the ground truss members. When the cross-sectional area reaches the vanishing value, it is removed to obtain the optimum [33]. The ground truss approach can only provide a rough estimate for the geometry of designed structures whereas the homogenization method provides better results but is more computationally expensive and might produce non-realizable elements [10].

Pure size optimization is somewhat easier to implement because it is a continuous optimization problem. The topological aspect of the design might still be integrated through eliminating vanishing element during size optimization, but difficulties may arise from the discontinuity brought forth.

2.6 *Optimization methods*

Optimization methods, as stated earlier, can also be referred to as mathematical programming (MP) methods. Typical MP methods find the optimum in problems where there exists a single global optimum [27]. In a lattice structure design however, a large design space is to be expected where there are usually many local optima. As a result, global optimization methods are more commonly applied. These methods use heuristic strategies to search the design space, and therefore are also called global search heuristics.

The evolutionary optimization, like Genetic Algorithm (GA), is a very popular optimizer used by many [20, 21, 29]. Wang et. al [10] found that Particle Swarm Optimization (PSO) shares similarities with GA, but it often converges more quickly than GA for the design synthesis of cellular structures [34]. Since the design of a compliant mechanism can be formulated as a least-squares problem, the Levenburg-Marquardt method (LM) which solves least-squares minimization problem will be investigated in chapter 4.

Regardless of what optimization procedure is considered, the number of design variables is proportional to the number of elements either with binary variables that determine if each element remains or eliminated [20, 21] or/and the dimension of the elements (width, density, diameters, etc.) [10, 26, 29, 35, 36]. When designing a lattice structure, the number of elements often is of thousands to millions, resulting in a large design domain and requiring a time consuming optimization process.

2.7 *Gap Analysis*

In designing compliant mechanism, none of the current synthesis methods can avoid optimization procedure. While they are more or less proved to be effectual, it often becomes too computationally costly when lattice structure design is involved. Lattice structures usually consist of thousands or even millions of elements, issuing a design space that is highly nonlinear and that contains many local minima due to the many design variables.

Instead of creating a totally different new method, this thesis will focus on reducing computing time associated with optimization procedures from two aspects. One is to reduce the number of design variables; the other is to use a more efficient optimization method. Both of the proposed approaches do not change the nature of design method but the formulation of problem.

2.8 *Summary*

In this chapter, some of previous research relating to lattice structure design is reviewed. The advantage of using cellular structures, especially design cellular structures comparative to stochastic cellular structure is discussed. Several structure analysis methods are also presented, along with the reasons why the unit truss approach is chosen to be used in this thesis. Finally, a brief overview of structural optimization is presented. Topology, geometry and size optimization definition is given. Lattice optimization methods used and their common shortcoming are also discussed.

A research gap was identified in designing compliant mechanisms of large scale lattice structures. The proposed work does not focusing on creating a brand new design method, but rather a different way to formulate optimization procedure so that the number of design variables and computing time for optimization can be reduced.

Chap.3 *Problem Formulation*

In this chapter, formulations for problems to utilize the design synthesis method are established. The idea is to obtain a truss structure suitable for a given initial shape to morph to a desired shape under the application of actuators. The process is accomplished in two steps: first, the initial structure is acquired by mapped mesh and primitive instancing; then, the initial structure goes through the optimization process which automatically finds the optimal combination of the strut diameters according to user defined objective function. Two different approaches are used, one finds each strut diameter directly, one finds coefficients (control diameters) which control strut diameters throughout the design space. More corresponding details of these steps to attain the goal will be discussed in the following sections.

3.1 *Primitive Instancing*

3.1.1 *Introduction*

Before a suitable structure can be obtained, an initial structure topology is needed. A good initial structure is very important for optimization processes. It might not only reduce a good amount of computing time needed, but it also leads to obtaining better results. The purpose is to fill the volumetric region with struts systematically.

3.1.2 *Mesh*

Possible ways to implement trusses into a design domain are many. Instead of randomly placing struts in the design domain, some automated (or semi-automated) methods were considered below.

i FEM Mesh

Probably the easiest way to obtain an initial truss topology is to use finite element modeling (FEM) software to automatically generate a mesh, then, with the mesh, obtain a corresponding truss structure, as shown in Figure 3.1. As is known, many FEM tools can divide a given volume into small elements for analysis purposes. Utilizing that, a corresponding truss structure can be defined.

The advantage of this approach is that it is almost fully automatic and is easier and faster to obtain an initial strut configuration, but the drawback is that the truss structure generated is usually not suitable for designing a morphing structure. For optimization process, a good starting design is needed to ensure good result gained from the optimization process to be discussed in chapter 4.

The mesh gained from FEM tools is for the purpose of analyzing; therefore, at certain locations that are considered to need more detailed analysis, smaller elements will be created, and often forming triangular structures with neighboring elements. The purpose of designing cellular structures is to only use material where it is needed. With the FEM mesh generating fine mesh for analysis purpose; it will not be a very good starting structure for optimization. Also, more triangular structures make the overall structure more inflexible thus not good for morphing.

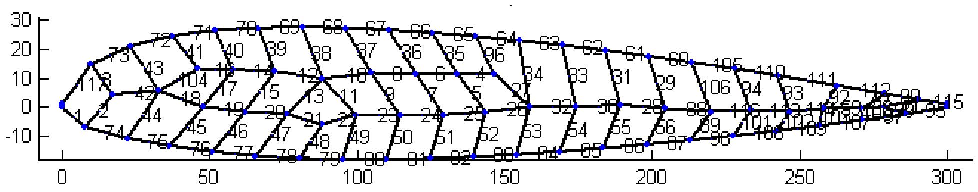


Figure 3.1: An airfoil example of FEM software generated mesh

Figure 3.1 shows a starting topology of an airfoil example used in Wang and Rosen's paper [10]. The structure was taken from the mesh generated in ANSYS. As can be observed, the topology contains a lot of "stiff" structure, thus, for the reasons stated above, it is not a very good truss topology to start with.

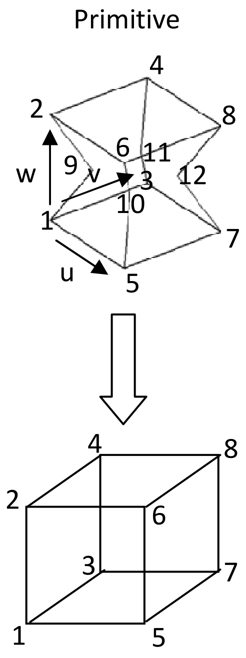
ii Mapped Mesh

Instead of using FEM software to generate mesh, we approach this by mapped mesh. Different from using a FEM free meshing method, mapped mesh is used to create a base for primitive instancing instead of using it as the starting truss topology. The idea is to first divide the design domain into quadrilaterals (2D) or hexahedrons (3D) as geometric bounds for later replacing the volumetric region with chosen truss primitives. An example is shown in Figure 3.2. The primitive has parameters u, v, w (from 0 to 1) to indicate three directions. Each point in a primitive is defined by its u, v, w value. With the global coordinates of the eight points that bound a cell, the coordinate of the nodes can be calculated relatively according to the u, v and w value. The equation for calculating the coordinate of nodes is:

$$\sum_{i=1}^{8} P_i \psi_i \tag{3.1}$$

where P_i is the x, y, z (global) coordinate of the eight points bounding the cell, and ψ_i is the interpolation function provided in equation (3.2). These interpolation functions are created through the inspiration of Lagrange interpolation functions used for rectangular elements [37] and expanded it into three dimensional. Lagrange interpolation is a well known, classical technique for interpolation technique. It was used to map finite element nodes into arbitrary mesh in [37], which is very similar to mapping nodes into unit cells.

$$\begin{gathered}
\psi_1 = (1-u) \times (1-v) \times (1-w) \\
\psi_2 = (1-u) \times (1-v) \times w \\
\psi_3 = (1-u) \times v \times (1-w) \\
\psi_4 = u \times (1-v) \times (1-w) \\
\psi_5 = u \times (1-v) \times w \\
\psi_6 = u \times v \times (1-w) \\
\psi_7 = (1-u) \times (1-v) \times w \\
\psi_8 = u \times v \times w
\end{gathered} \tag{3.2}$$



	u	v	w
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1
9	0.25	0.25	0.5
10	0.75	0.25	0.5
11	0.25	0.75	0.5
12	0.75	0.75	0.5

Figure 3.2: Primitive Instancing Example

The mapped mesh approach is to utilize Bezier curves or surfaces to approximate the outer shape of the design domain, and obtain a parametric representation of it. Unit cells can then be defined parametrically. Although it doesn't seem easy to always divide an area or a volume into all quadrilaterals or hexahedrons, studies of this issue have already shown that it can be done automatically [38]. Figure 3.3 is a simpler example, which had most of its design domain divided into unit cells and some parts left without struts. The approach was appropriate for the application.

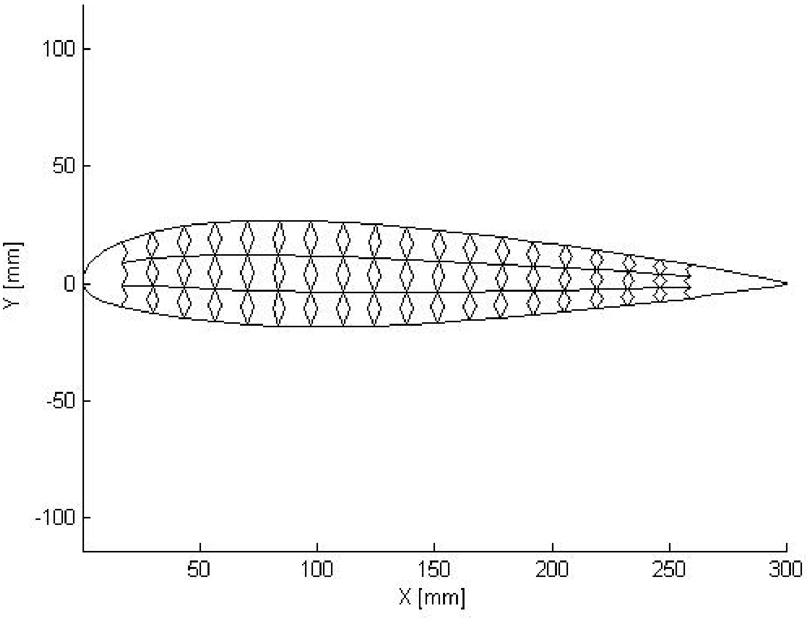


Figure 3.3: A simple example utilizing mapped mesh

3.1.3 *Different Primitives*

Once the unit cells are set up, different primitives can be inserted in. The development of different primitives is not in the scope of this thesis. We chose two (showed in Figure 3.4 and Figure 3.5) that are seemingly more suitable for the example problems that are discussed in chapter 5. The former is more suitable for morphing with, instead of upright struts, bent, concave struts at the sides that are more likely to deform under loading. On the contrary, the latter has elements connecting all the nodes which make it less likely to deform under loading in different directions.

Different primitive topologies can be established in the future to get more variety of the structure design.

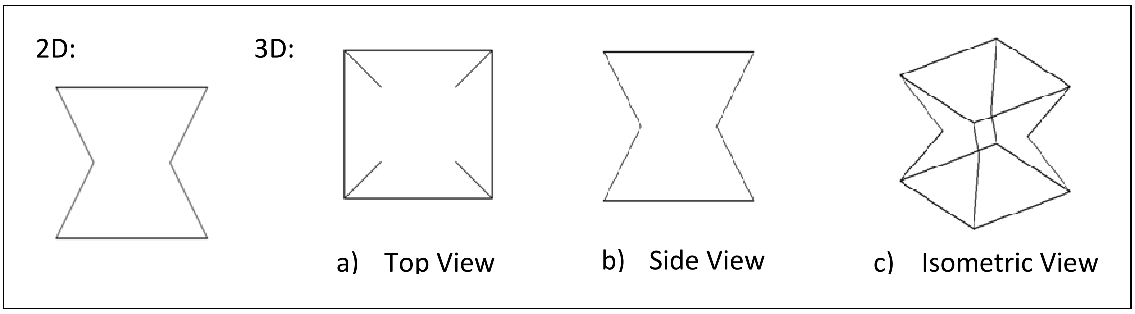


Figure 3.4: Primitive for morphing structure

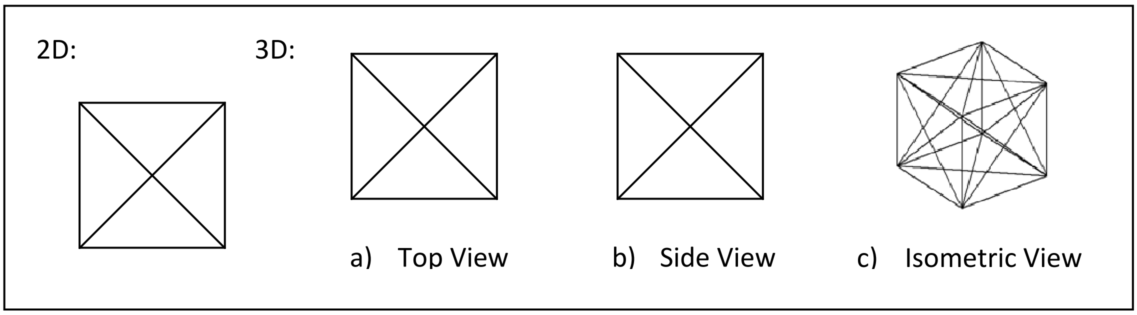


Figure 3.5: Primitive for stiff structure

3.2 *Problem Formulation*

Two different formulations of the problem corresponding to two different approaches will be given here. The first is to optimize each strut size (diameter) and find the optimal composition, a straightforward size optimization of the struts. The other is to find the control diameters that best control the strut size in each unit cell. More detail is given in the following sections.

3.2.1 *Strut Approach*

To utilize the diameter of each strut as the design variable is easy to implement and fairly straightforward. A general word formulation for structural design problem is given in Figure 3.6.

Given:

Truss Structure: Position and orientation of nodes and elements

Specified load and boundary condition

Find: diameter of each strut

Satisfy:

Goals:

Minimize nodal deflection deviation from target.

Minimize truss volume

Minimize plastic deformation in struts (optional constraint)

Bounds:

Minimum and maximum diameter constraints

Minimize:

Weighted sum of square deviation from goal values of the objectives

Figure 3.6: Word formulation for strut approach

This problem is formulated using the Compromise Decision Support Problem (cDSP) formulation [39-41]; therefore, the multi-objective function to be minimized is a weighted-sum of goal deviations, and each goal is specified as deviation from a target value. The optional constraint can be added in the objective function as a penalty function [42]. In this thesis, the prevention of plastic deformation is done by calculating the Von Mises equivalent stress in each strut, and when it exceeds the yield stress of the strut, the difference is then used in the penalty function. This might not be as important if designing a stiff structure when expecting small deformation. But, when designing a compliant mechanism, that is, when large deformation and ability to switch between two shapes in reasonable number of times is to be expected, prevention of plastic deformation would be important in order to ensure the feasibility and usability of the designed structure.

A math formulation is given in Figure 3.7 where δ is the deflection and subscript “actual” and “desired” denotes whether it is the actual deflection or the target. Vol is the volume of the structure. vS is the Von Mises stress calculated in the middle of each strut and yS is the yield stress that can be obtained from material property.

Given:

$\mathrm{x, y, z}$ coordinate of each node

Each element defined by two nodes, indicating it connects those two nodes.

Load each defined by a node number and the magnitude in the global x, y, and z direction, they can be either force, moment or both.

Boundary condition can be specified by the node number and in what direction it has constraint.

$\delta_{\mathrm{desired}}$ for each node in interest.

Find: $\mathrm{d_j}$, j = 1, 2, 3...number of struts (diameter of each strut)

Satisfy:

Goals:

Minimize:

$|\delta_{\mathrm{k_{actual}}} - \delta_{\mathrm{k_{desired}}}|$ for node k (nodes in interest)

$\mathrm{Vol} = \sum \pi d^2/4$ (sum of the volume of all the struts)

$\mathrm{vS_j} - \mathrm{yS_j}$, the Von Mises stress minus the yield stress in each strut, only calculated when $\mathrm{vS_j} > \mathrm{yS_j}$. (optional)

Bounds:

$0.5 \le \mathrm{d_j} \le 5$ (mm)

Minimize:

$$\mathrm{Z} = \mathrm{w_d} \times \sum (\delta_{\mathrm{k_{actual}}} - \delta_{\mathrm{k_{desired}}})^2 + \mathrm{w_v} \times \mathrm{Vol}^2 + \mathrm{w_y} \times \sum (\mathrm{vS} - \mathrm{yS})^2$$

Where $\mathrm{w_d}$, $\mathrm{w_v}$, $\mathrm{w_y}$ is the importance of preference for each part.

Figure 3.7: Math formulation for strut approach

3.2.2 *Parametric Surface/Solid Approach*

Although the strut approach is easy to implement and straightforward, the design domain it creates is usually very large and contains hundreds or thousands of design variables. Since optimized structures often have smoothly varying changes in density/strut size, the parametric surface/solid approach is introduced. In this parametric surface/solid approach, the number of design variables can be greatly reduced. It was created utilizing a concept similar to the Bezier surface, in which the coordinates of the points on the surface are calculated by:

$$\mathrm{p}(\mathrm{u},\mathrm{v})=\sum_{\mathrm{i}=0}^{\mathrm{m}}\sum_{\mathrm{j}=0}^{\mathrm{n}}\mathrm{p}_{\mathrm{ij}}\cdot\mathrm{B}_{\mathrm{i},\mathrm{m}}(\mathrm{u})\cdot\mathrm{B}_{\mathrm{j},\mathrm{n}}(\mathrm{v})$$

(3.3.a)

$$0\leq\mathrm{u}\leq1,\ \ 0\leq\mathrm{v}\leq1$$

$$\mathrm{B}_{\mathrm{i},\mathrm{m}}(\mathrm{u})=\frac{\mathrm{m}!}{\mathrm{i}!\,(\mathrm{m}-\mathrm{i})!}\mathrm{u}^{\mathrm{i}}(1-\mathrm{u})^{\mathrm{m}-\mathrm{i}}$$

(3.3.b)

p_{ij} are the control vertices. m and n will be the degree of the surface. With an array of $(\mathrm{m}+1)\times(\mathrm{n}+1)$ control vertices, points on a Bezier surface with degree $\mathrm{m}\times\mathrm{n}$ can be created [43]. An illustration is shown in Figure 3.8 below.

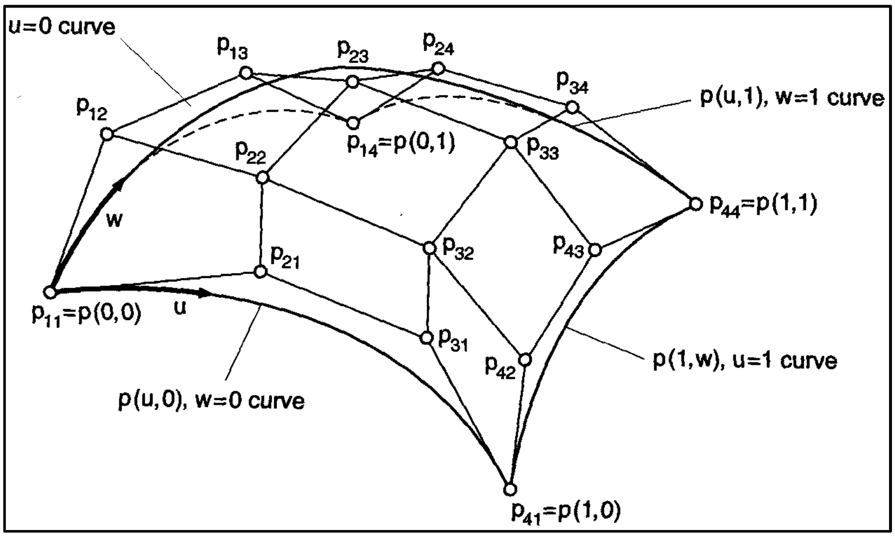


Figure 3.8: Bezier surface illustration

As stated above, the position of the points on the surface can be calculated with the given control vertices and the Bezier functions. Applying similar concept, instead of control vertices, several "control diameters" can be provided; along with the Bezier functions, strut diameters throughout the design space can be defined as exhibited in equation 3.4 and 3.5 below.

$$d(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} d_{ij} \cdot B_{i,m}(u) \cdot B_{j,n}(v) \tag{3.4}$$

or

$$\mathrm{d}(\mathrm{u},\mathrm{v},\mathrm{w}) = \sum_{\mathrm{i}=0}^{\mathrm{m}} \sum_{\mathrm{j}=0}^{\mathrm{n}} \sum_{\mathrm{k}=0}^{\mathrm{q}} \mathrm{d}_{\mathrm{ijk}} \cdot \mathrm{B}_{\mathrm{i,m}}(\mathrm{u}) \cdot \mathrm{B}_{\mathrm{j,n}}(\mathrm{v}) \cdot \mathrm{B}_{\mathrm{k,q}}(\mathrm{w}) \qquad \textbf{(3.5)}$$

$$0 \leq \mathrm{u} \leq 1,\ 0 \leq \mathrm{v} \leq 1,\ 0 \leq \mathrm{w} \leq 1$$

As shown in equation 3.5, adding the $\mathrm{B}_{\mathrm{k,q}}(\mathrm{w})$ term makes it applicable for three dimensional problems. The parameters here shall not be confused with the ones used in section 3.1.2. The ones in 3.1.2 are used to define the primitives in order to insert in the unit cells, whereas here the parameters are used after an initial structure is acquired to define the entire design space. Like how control vertices control the position of all the points on a Bezier surface, d_{ij} or $\mathrm{d}_{\mathrm{ijk}}$ are the control diameters used to define the diameters in each cell in a similar manner. Through assigning each cell with parameter values, the strut diameters in that cell can be calculated using equation 3.4 and 3.5. In this way, the number of design variables can be reduced to $(\mathrm{m}+1) \times (\mathrm{n}+1) \times (\mathrm{q}+1)$ depending on the degree of a parametric surface/solid that is used here. The function value $\mathrm{d}(\mathrm{u},\mathrm{v},\mathrm{w})$ denotes the diameter variation through u, v and w.

The parametric values (u, v and w) are assigned to the unit cells. The n$^{\mathrm{th}}$ unit cell along u's direction will have a u value of n/total number of unit cells along u direction. The struts are assigned to the unit cells during primitive instancing. When struts overlap, they are defined to be belonging to the unit cell first assigned. The sequence of this procedure is first along the u direction, then v and w respectively. Figure 3.9 is an illustration of this sequence. The order of assignment is: black, red, blue and green. Unit cells along u are assigned first while maintaining the same v value. Then the cells along u with the next v value are assigned. Since there are a total of two unit cells along u direction, the first unit cell will have a u value of 1/2 = 0.5. The same principle applies to the v value.

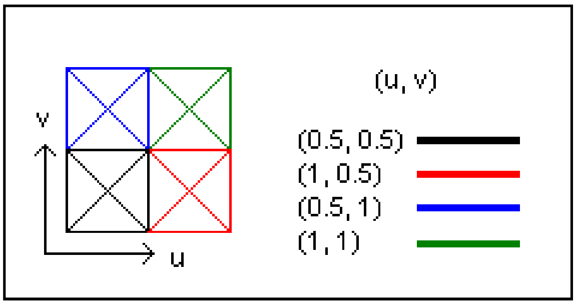


Figure 3.9: Assigning Parametric Value

A demonstration of the relation between unit cells and the control diameters is shown in Figure 3.10. A 1x1 parametric surface is used. Four control diameters (2x2) are represented by black circles. The unit cells are shown by blue dots. The control diameters' value decide the diameter value for each unit cell, just as the coordinates of the control vertices of Bezier surface decide the coordinates of the points on the surface.

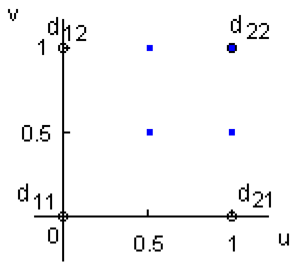


Figure 3.10: Relation between control diameters and unit cells

Assuming there are a total number of AxBxC unit cells with A unit cells in the u direction, B, C in v and w direction respectively. The diameter of struts in a unit cell that is the ath, bth, cth along u, v and w direction respectively, will have a diameter value of D_{abc}. This D_{abc} can be calculated by equation (3.6).

$$D_{abc} = \sum_{i=0}^{m}\sum_{j=0}^{n}\sum_{k=0}^{q} d_{ijk} \cdot B_{i,m}(a/A) \cdot B_{j,n}(b/B) \cdot B_{k,q}(c/C) \tag{3.6}$$

As Bezier functions create a smooth curve/surface, the parametric surface/solid approach is expected to give a smooth variation to diameters of struts in the design domain. Although the degree of function used (m, n, q in the previous paragraph) is user defined, it is reasonable to choose a degree that is one lower than the number of unit cells in the corresponding direction. For example, if a structure has a total of two unit cells in the x direction, no matter what the diameters should be, a linear function would be sufficient to represent the diameter variation.

The problem formulation for this approach is very similar to that used for the other approach, but instead of diameters, we now use just a set of control diameters, $\mathrm{d_{ij}}$ or $\mathrm{d_{ijk}}$, as our design variables. The number of design variables is decided by the diameter variation degree through the parameters that user decide on, but normally the amount of design variables is still greatly reduced. A word formulation of this problem approach is given in Figure 3.11 below.

Given:

Truss Structure: Position and orientation of nodes and elements

Specified load and boundary condition

Find: control diameters

Satisfy:

Goals:

Minimize nodal deflection deviation from target.

Minimize truss volume

Minimize plastic deformation in struts (optional constraint)

Bounds:

Minimum and maximum coefficient value

Minimize:

Weighted sum of square deviation from goal values of the objectives

Figure 3.11: Word formulation for parametric surface/solid approach

As can be observed, the word problem formulation is very similar to the one shown in Figure 3.6. The only difference is that instead of finding strut size directly, a set of control diameters is to be found; and, through the method stated earlier, the diameters of struts are then defined indirectly. The math formulation of the same problem is given in Figure 3.12.

Given:

$\mathrm{x, y, z}$ coordinate of each node

Each element defined by two nodes, indicating it connects those two nodes.

Load each defined by a node number and the magnitude in the global x, y, and z direction, they can be either force, moment or both.

Boundary condition can be specified by the node number and in what direction it has constraint.

$\delta_{\mathrm{desired}}$ for each node in interest.

Find: $\mathrm{d_{ij}}$ or $\mathrm{d_{ijk}}$ for 2D or 3D problem. $\mathrm{i, j, k}$ from 0 to the degree of variation in $\mathrm{u, v, w}$ (user defined).

Satisfy:

Goals:

Minimize:

$|\delta_{\mathrm{k_{actual}}} - \delta_{\mathrm{k_{desired}}}|$ for node k (nodes in interest)

$\mathrm{Vol} = \sum \pi d^2/4$ (sum of the volume of all the struts)

$\mathrm{vS_j} - \mathrm{yS_j}$, the Von Mises stress minus the yield stress in each strut, only calculated when $\mathrm{vS_j} > \mathrm{yS_j}$. (optional)

Bounds:

$0.5 \leq \mathrm{d_j} \leq 5$ (mm)

Minimize:

$$\mathrm{Z} = \mathrm{w_d} \times \sum (\delta_{\mathrm{k_{actual}}} - \delta_{\mathrm{k_{desired}}})^2 + \mathrm{w_v} \times \mathrm{Vol}^2 + \mathrm{w_y} \times \sum (\mathrm{vS} - \mathrm{yS})^2$$

Where $\mathrm{w_d}$, $\mathrm{w_v}$, $\mathrm{w_y}$ is the importance of preference for each part.

Figure 3.12: Math formulation for parametric surface/solid approach

3.3 *Summary*

In this chapter, problem formulations are established. Base truss topologies are obtained through primitive instancing. By dividing the design domain into quadrilaterals or hexahedrons, different truss primitives can be inserted into them to generate an initial truss structure. Two different approaches were discussed. Using the strut approach, the number of design variables is equal to the number of elements. But by the parametric surface/solid approach, the number of design variables can be reduced greatly; thus, in principle, optimization computing time can be greatly reduced also, since the design space to explore is much smaller. Experiments to support this will be presented with in chapter 5 using several examples. Optimization method used to solve these problems is provided in the next chapter.

So far, the problem formulation that can realize hypothesis two: “The number of design variables can be reduced by changing the design variables from diameters of struts to a small number of coefficients that control the distribution of strut sizes across the structure using parametric surface/solid formulations” is presented; and it was shown in principle that it is achievable. This will also be further employed in the example problems in chapter 5 to show its value and restriction.

Chap.4 *Optimization Methods*

The reasons why a suitable optimization process is so significant for lattice structure design are several. First, with the strut diameter approach, the problems usually have hundreds and sometimes even thousands of design variables (struts), thus an optimization method which can handle large-scale problems is required. Second, there are usually a lot of local minima and it is very likely that the so-called optimum found would be just a local optimum. Last but the most important is that every time when the objective function value has to be calculated, it requires the structure be analyzed through Finite Element Analysis code. Not only does this take a lot of time, but also there is no analytical gradient and/or hessian matrix available.

To compare the efficiency, the PSO and the LM method with a Least-Squares Minimization formulation, which will be referred to as the nonlinear least-squares minimization methods (LSM), were chosen. The characteristics of the two optimization methods are discussed in the following sections of this chapter.

4.1 *Particle Swarm Optimization*

PSO simulates the behavior of a group of birds searching for food. It generates a group of random particles, and each particle's velocity then is governed by both its own and the society's experience of "the best" position. The algorithm is considered converged when particles start circling around a small area. With a group of particles “flying” through the design space, PSO is assumed to perform a better global search compared to general purpose mathematical programming methods, and, because of that, might take longer time. PSO was chosen since it was proved to be superior to the GA and is one of the more efficient algorithms for size and shape optimization [34]. The cooperative behavior modeled in PSO and competitive behavior in GA cause PSO to converge faster than GA [10].

The way each particle’s velocity and location are updated is shown in equation (4.1.a) and (4.1.b). The superscript k indicates the $\mathrm{k^{th}}$ iteration, and the subscript i signify the $\mathrm{i^{th}}$ particle. In (4.1.a), the velocity (v) is updated, then according to the velocity, the location (x) is changed. The ω in equation (4.1.a) is called inertia weight, which, when larger, gives each particle a larger speed and facilitates a global search; likewise it facilitates a local search while small. $\mathrm{c_1}$ and $\mathrm{c_2}$ are two constants, which are called cognitive and social parameters respectively. They regulate the relative velocity toward local and global optimum. $\mathrm{r_1}$, $\mathrm{r_2}$ are two random numbers in [0,1]. The p in the equations denotes the best position the particle has ever visited, but with the g subscript, it means the best position among the entire group.[44, 45]

$$\mathrm{v_i^{k+1}} = \omega \cdot \mathrm{v_i^k} + \mathrm{c_1} \cdot \mathrm{r_1} \cdot \left(\mathrm{p_i^k} - \mathrm{x_i^k}\right) + \mathrm{c_2} \cdot \mathrm{r_2} \cdot \left(\mathrm{p_g^k} - \mathrm{x_i^k}\right) \qquad \textbf{(4.1.a)}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{4.1.b}$$

As can be observed, there are several parameters that can affect the efficiency of this algorithm. For PSO to run more efficiently, better parameter values must be chosen. Since the computing time changes every time even using the same computing device, the efficiency is compared through number of function calls. For easier understanding of the results, Rastrigin function (equation 4.2 below), which was used in many research papers as one of the examples [44, 46, 47], was chosen to be used for the experiments. The minimum is already known as $f(x) = 0$ when all $x_i = 0$.

$$f(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$$

$$x \in (-5.12, 5.12)^n \tag{4.2}$$

The parameters focused on are the swarm size, inertia weight and cognitive and social acceleration.

4.1.1 ***Swarm Size***

There are some studies done on parameter choices for PSO that touched swarm size[45, 47, 48]. The general understanding is that a larger swarm size will improve the success rate but will also increase the number of objective function evaluations. Thus, it is a choice between better solution and shorter computing time. The suggestion given in Zhang, Yu and Hu's work is to choose a swarm size of 50 for more complicated problems and [20, 50] for simpler problems, and as they stated, 30 might be a good choice. But, believing that swarm size should have some sort of relation with the dimension of the problem, since the dimension infers the size of the design space, some experiment are conducted as follows using the parameters shown in Table 4.1. The maximum number of iterations denotes the maximum allowed iterations; when the number of iterations reaches this value, the optimization stops without finding optima. The optima is considered found when the objective function value is smaller than the error goal that the user defined. Cognitive and social acceleration controls the relative velocity of particles towards local and global optimum. The inertia weight controls how much a particle changes its velocity; with a larger value, the particles will tend to maintain its velocity as the last iteration.

Table 4.1: Parameters used for PSO swarm size experiment

Maximum Number of Iterations	10000
Error Goal	100
Cognitive Acceleration	2.0
Social Acceleration	2.0
Initial Inertia Weight	0.9
Final Inertia Weight	0.4

The results gained are in appendix I. For dimension 10, since the swarm sizes are all equal to or larger than the dimension itself and the global optimum is relatively easy to find because of the small design space, the result doesn't really tell us anything. When the dimension is 20 and swarm size is 10, half of the time PSO converged to local optima; this might imply that a swarm size of half of the dimension size is too small according to the general understanding stated in previous paragraph. But from the data obtained for dimensions 30, 40 and 50, we can see that PSO is actually very unpredictable. In principle, we would expect a smaller group of particles to converge faster and would be easier to trap in local optima, and that a larger group would either find the global optima or converge too slowly and exceed the maximum number of iteration. But it seems that even with the same swarm size, both situations happened and there is really no tendency as to either a larger or smaller swarm size is better. A conjecture of what happened would be that, since the initial position for each particle of the swarm is randomly generated, it might happen that when the particles started to be too close to each other, then they easily converge to a local optima, and when the particles started to be around some local optimum (but not the global optima), then they hovered between those local optima and could not converge. Even with a relatively small problem like this, PSO tends to call objective function evaluation a large number of times; considering that a bigger swarm size doesn't ensure a good solution, plus the significant computing time required for FEM analysis with each function call, a smaller swarm size (about 1/3 of dimension) might be preferred.

4.1.2 ***Inertia Weight***

In [44], Zheng, Ma and Qian stated that, on the contrary of what Y. Shi and R. Eberhart suggested, the inertia weight should start from 0.4 and linearly increase to 0.9 instead of decreasing from 0.9 to 0.4.

Table 4.2: Parameters used for PSO inertia weight experiment

Maximum Number of Iterations	10000	
Error Goal	100	
Cognitive Acceleration	2.0	
Social Acceleration	2.0	
Initial Inertia Weight	0.9	0.4
Final Inertia Weight	0.4	0.9

Using the parameters in Table 4.2, the results are shown in Table 4.3 and Table 4.4 for decreasing and increasing inertia weight respectively:

Table 4.3: Results of decreasing inertia weight

Number of iterations	Final Function Value	End Reason
10001	133.54	ExceedMaxIter
9295	160.4	Local Min
8617	123.52	Local Min
10001	136.45	ExceedMaxIter
6586	99.778	Global Min
10001	178.24	ExceedMaxIter
8558	148.32	Local Min
10001	167.29	ExceedMaxIter
7126	197.28	Local Min
7058	99.655	Global Min

Table 4.4: Results of increasing inertia weight

Number of Iterations	Final Function Value	End Reason
256	99.135	Global Min
527	139.37	Local Min
643	168.36	Local Min
353	98.751	Global Min
464	157.37	Local Min
494	169.43	Local Min
859	125.36	Local Min
634	112.43	Local Min
402	184.37	Local Min
448	118.42	Local Min

Although increasing the inertia weight doesn't necessarily give better solution, it does generally solve the problem faster.

4.1.3 *Cognitive and Social Acceleration*

Introduced by Zhang, Yu and Hu's work on parameter choice[45], a new parameter φ is set as the sum of c_1 and c_2 in equation (4.1.a). It was suggested that $\varphi = 4.05$ is appropriate for high multimodal functions, and $\varphi = 4.1$ for unimodal functions, similar from the conclusion drawn from [49]. Combine that with a suggested value [49] for c_1/c_2 ratio to be 2.8/1.3. The cognitive and social acceleration were chosen accordingly.

4.2 *Least Squares Minimization Formulation*

Since the number of design variables far exceeds the number of objectives, these problems can also be formed as data fitting problems, in which a large data set is fit to a low order polynomial model. The least-squares method finds a local minimizer to a function that is a sum of squares.

$$\mathrm{S(X)} = \sum_{\mathrm{i}} (\mathrm{P_{i,target}} - \mathrm{P_{i,actual}(X)})^2 \tag{4.3}$$

The least-squares formulation is given in equation 4.3 above, where $\mathrm{P_{i,target}}$ and $\mathrm{P_{i,actual}}$ are the desired and actual value of the $\mathrm{i^{th}}$ objective. Since this term is to be minimized, its derivative is set equal to zero:

$$\nabla \mathrm{S(X)} = 2\sum_{\mathrm{i=1}}^{\mathrm{n}} \left[\frac{\partial \mathrm{P_{i,actual}(X)}}{\partial \mathrm{X}}\right] \left[\mathrm{P_{i,target}} - \mathrm{P_{i,actual}(X)}\right] = 0 \tag{4.4}$$

where the partial derivative term is the Jacobian, **J**(**X**). Due to the nonlinearity of **J,** an iterative solution technique must be used to solve for the unknown coordinates, **X** of the system. Gauss-Newton methods are typically used to solve such problems [41]. The Levenburg-Marquardt (LM) method, which can be thought of as a modified Gauss-Newton algorithm, is used in this research since it tends to be more robust when sensitivities in the Jacobian are small.

The search direction that the LM method uses is a solution to equation 4.5 below:

$$\left((\mathbf{J}^{\mathrm{k}})^{\mathrm{T}}\mathbf{J}^{\mathrm{k}} + \mu^{\mathrm{k}}\mathrm{I}\right)\mathrm{d}^{\mathrm{k}} = -(\mathbf{J}^{\mathrm{k}})^{\mathrm{T}}\left[\mathrm{P_{i,target}} - \mathrm{P_{i,actual}}\right] \tag{4.5}$$

where μ^k is a scalar damping parameter, d^k is the direction and superscript k denotes the k_{th} iteration. When $\mu^k = 0$, the method is reduced to Gauss-Newton method, but when μ^k tends toward infinity, d^k turns toward the steepest descent direction. This implies that for some sufficiently large μ^k, descent of the objective function can be ensured even when second-order terms, which restrict the efficiency of the Gauss-Newton method, are encountered.[50] The iteration function can be computed as:

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \left[(\mathbf{J}^k)^T\mathbf{J}^k + \mu^k I\right]^{-1}(\mathbf{J}^k)^T\left[P_{i,target} - P_{i,actual}\right] \tag{4.6}$$

MATLAB's nonlinear least-squares (nonlinear data-fitting) solver *lsqnonlin* is used as LSM solver in this thesis. It selects from Gauss-Newton and LM algorithms to solve problems.

4.2.1 *Limitations*

With one starting point, Least-squares minimization finds a nearby minimum. It finds a solution fairly quickly if compared with PSO (with similar stopping criteria), but its lack of global search ability often causes it to end up with a local minimum.

4.2.2 *Initial Design*

From the above statement, we know a good starting point in the design space, in our case a good initial combination of strut diameters, is very important. But if a good design is foreknown, an optimization will not be so meaningful, since it is used to find a solution that is not predictable. Although PSO seems to have, theoretically, a better global search ability, from experience [41] we found that a good starting point is very important to PSO also. Thus, some experiments done for efficiency comparison of these two methods are presented in section 4.3.

4.3 *Comparison of PSO and LM*

Three examples were investigated to include different numbers of design variables. The problems are created each with 9, 56 and 99 variables, using the base truss topologies shown in Figure 4.1~Figure 4.3. These structures are fixed at the left end and loaded with a 10 N point load at the top right end. The nodes of interest are chosen to be the nodes at the right end of the structure. And the target deflections are all set in regards to the deflection in y direction.

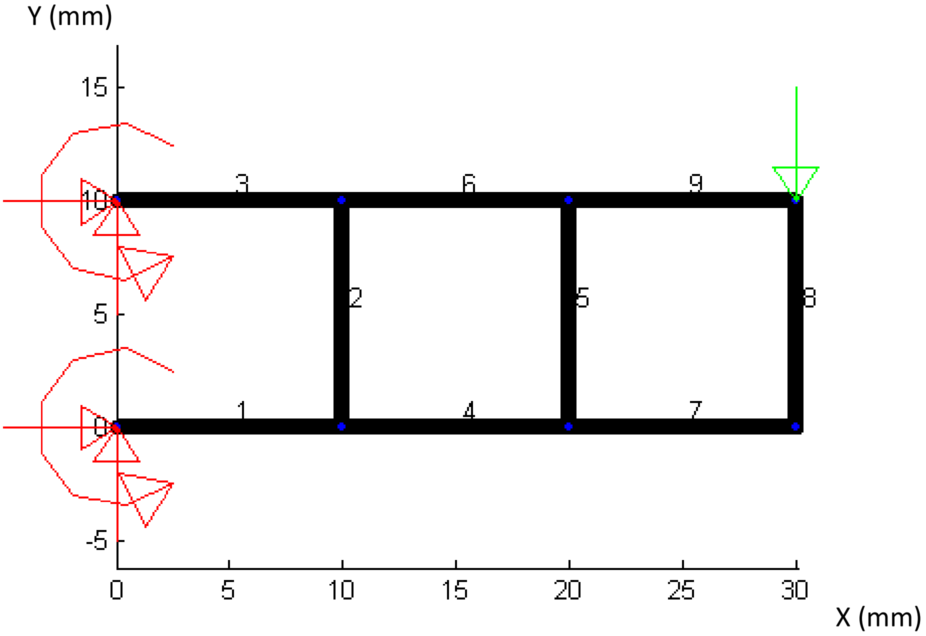


Figure 4.1: Problem 1 base truss topology

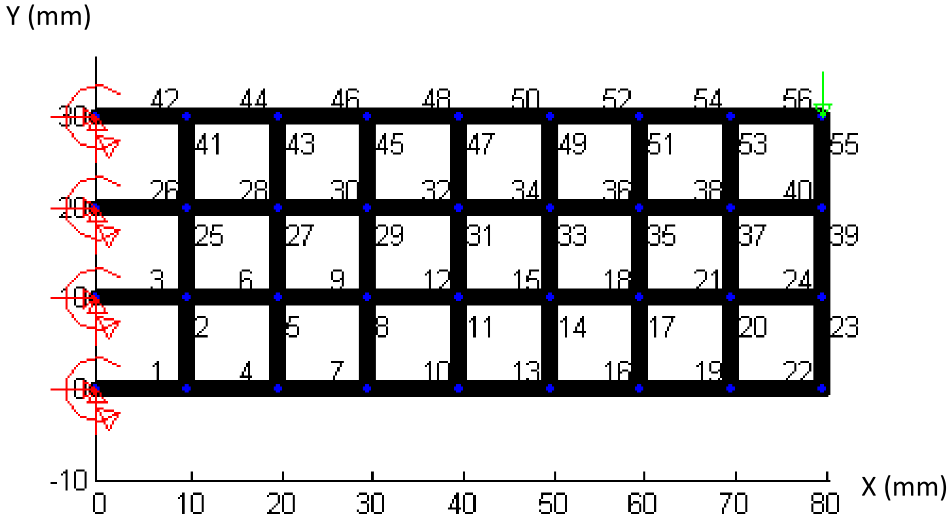


Figure 4.2: Problem 2 base truss topology

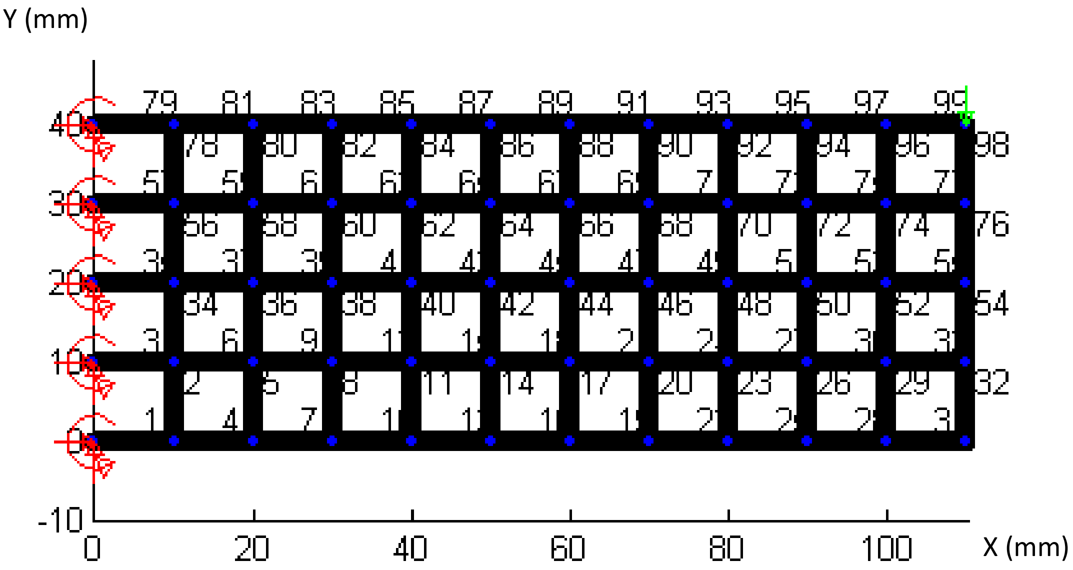


Figure 4.3: Problem 3 base truss topology

Given:

Truss Structure: Position and orientation of nodes and elements

Specified load and boundary condition: Nodes at X=0 are all fixed in space, 10N applied on the node at the top-right node

Desired deflection for the nodes in interest

Find: diameter of each strut $\mathrm{d_i}$ (*i* = 1...number of elements) (mm)

Satisfy:

Goals:

Minimize the difference between actual deflection and target deflection.

$\mathrm{RepDefl_j} = (\delta_{\mathrm{j_{actual}}} - \delta_{\mathrm{j_{target}}})$ mm, (j = 1...number of sample points)

Minimize generalized total truss volume

$\mathrm{Vol} = \omega_\mathrm{v} \times (\mathrm{TotalVolume_{actual}} - \mathrm{TotalVolume_{target}})$ mm^3

Minimize Stress exceeding Yield Stress

$yieldStress_k = (Stress_k - YieldStress_k)$ MPa, (k = 1...number of elements)

Bounds:

Minimum and maximum diameter constraints: $0.5\ \mathrm{mm} \leq \mathrm{d_i} \leq 5\ \mathrm{mm}$

Minimize:

$$\mathrm{ObjFunValue(d_i)} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{RepDefl_j}^2 + \mathrm{Vol}^2 + 10000 \times \sum_{\mathrm{k=1}}^{\mathrm{m}} \mathrm{yieldStress_k}^2$$

Figure 4.4: Problem formulation for PSO and LM comparison

The general optimization problem formulation for these example problems is given in Figure 4.4 where n is the number of sample nodes. $\mathrm{RepDefl_j}$ is the difference of the actual deflection and the desired deflection of each sample node. Vol is normalized total volume of the structure. m is the number of struts. $\mathrm{yieldStress_k}$ is the difference between the Von Mises stress at the middle node of the strut and the yield stress (only when Von Mises equivalent stress exceeds yield stress.) In the objective function, a large number (10^4) is multiplied to it so that this term can serve as a penalty function.

The starting point for all experiments conducted by LM is set such that all strut diameters are 2 mm. PSO, however, usually starts with a group of random particles. But, a seed (a particular particle in the starting swarm) of the same point was added to include a “starting point” in PSO.

For the objective function, there are three terms to consider:

1. The desired deflection for the sample nodes are set as 80% of the deflection in -y direction (pointing downward in fig. 2~4) when all the struts have the diameter of 2 mm.

2. The target volume was set as 80% of the volume of the starting structure. It is normalized by different ω_{v} for each example problem to ensure the scale of Vol will not exceed 1e-1 and that the square of Vol will be small enough that the deformed shape will have priority.

3. The third term is added to serve as a penalty function. If the Von Mises equivalent stress in the middle of the $\mathrm{k_{th}}$ strut is smaller or equal to the yield stress, then $\mathrm{yieldStress_k}$ is zero; otherwise $\mathrm{yieldStress_k}$ will be the difference between the equivalent stress and the yield stress.

The results are listed and can be compared in Table 4.5. With each problem, Table 4.5 shows the number of design variables, the average analysis time needed (i.e. objective function evaluation time), and the objective function value before optimization process.

The resulting objective function value and function calls used are listed to compare PSO and LM.

Table 4.5: Results of experiments to compare PSO and LM

	Initial State			PSO		LM	
	# of Design Variables	Average Analysis Time (s)	Initial Objective Function Value	Objective Function Value	Function Calls	Objective Function Value	Function Calls
Problem 1	9	0.0258	0.3888	0.009140	588	0.006273	190
				0.014292	477		
				0.011305	537		
				0.007729	1314		
				0.030219	534		
Problem 2	56	0.1491	1.4867	0.000387	169689	0.000574	3534
				0.000791	164540		
				0.000347	173679		
				0.000252	147212		
				0.000380	152361		
Problem 3	99	0.3503	2.3915	0.000385	303039	0.000918	4700
				0.001682	281358		
				0.000252	265716		
				0.000423	274857		
				0.000655	281061		

Multiple runs were performed for PSO since it is a stochastic algorithm. Since function evaluation time depends highly upon capacity of computer used and even with the same machine still fluctuates a lot, the performance of the algorithms are compared in terms of number of function calls and the resulting objective function value. Several result structure examples are shown in Figure 4.5~Figure 4.10.

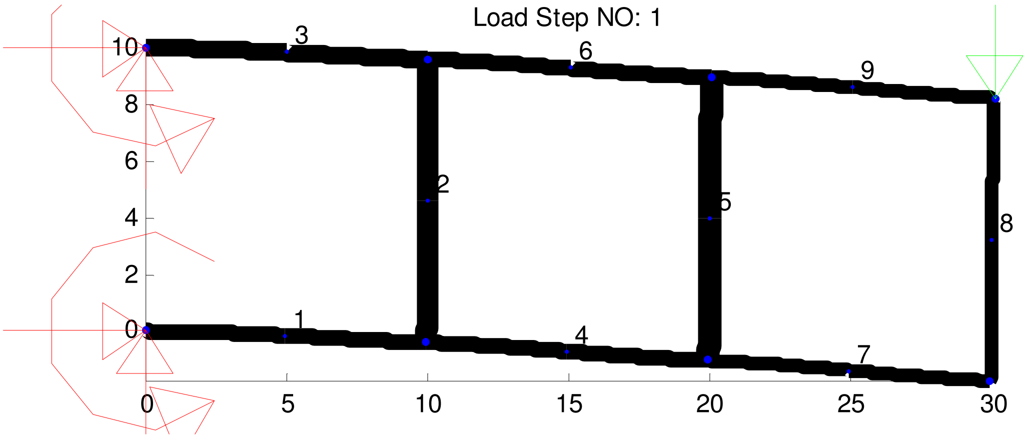


Figure 4.5: Result structure of problem 1 by LM

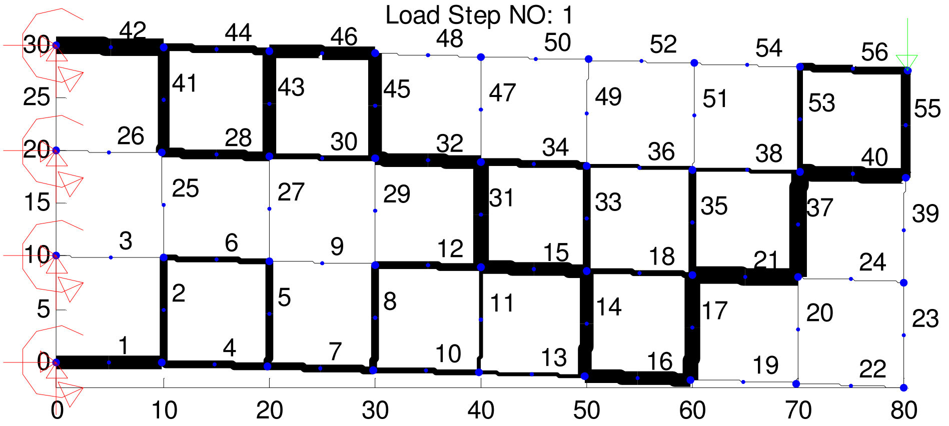


Figure 4.6: Result structure of problem 2 by LM

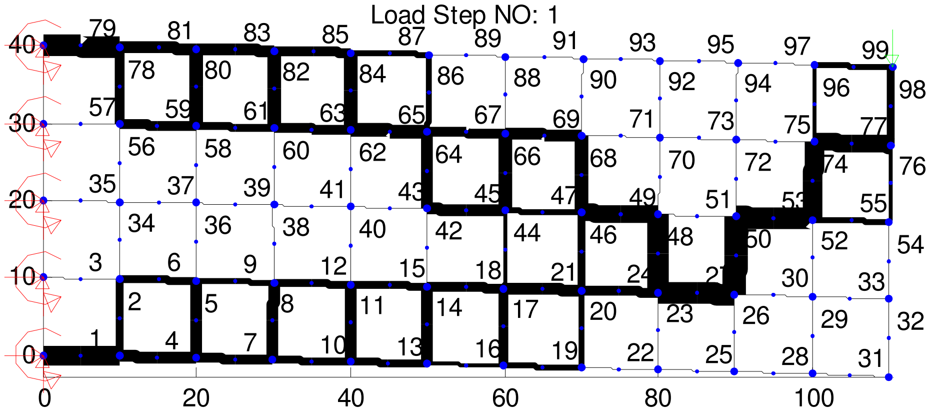


Figure 4.7: Result structure of problem 3 by LM

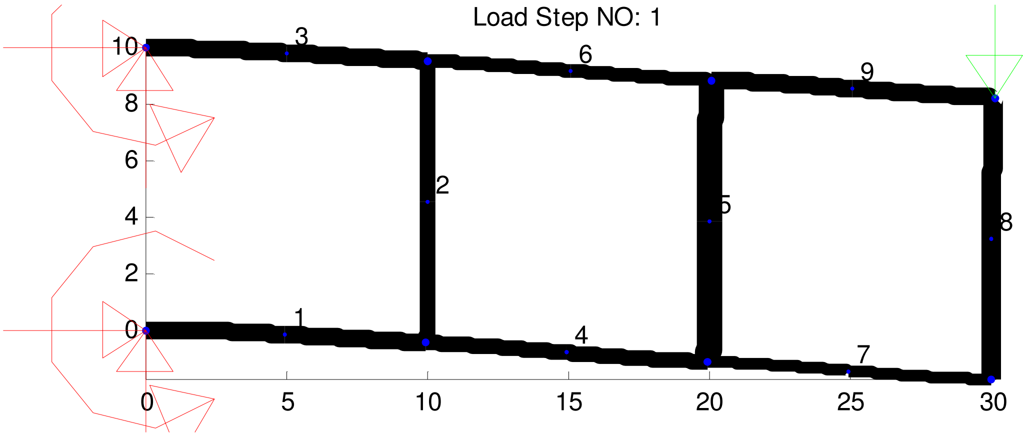


Figure 4.8: Result structure of problem 1 by PSO

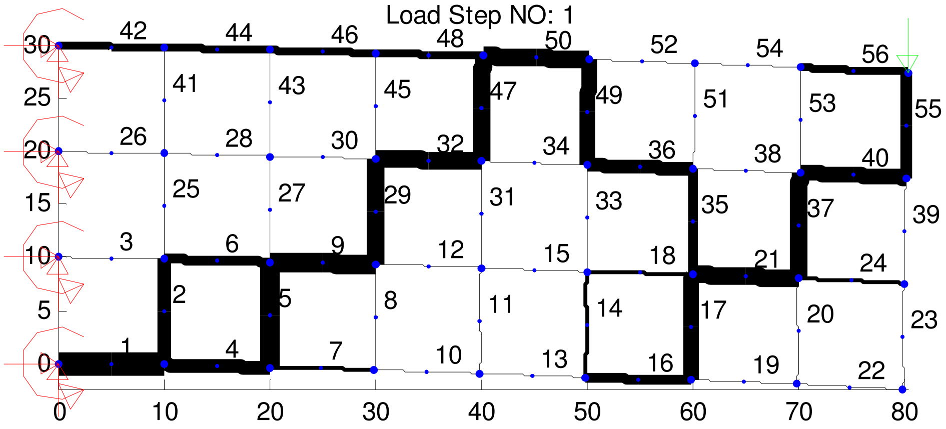


Figure 4.9: Result structure of problem 2 by PSO

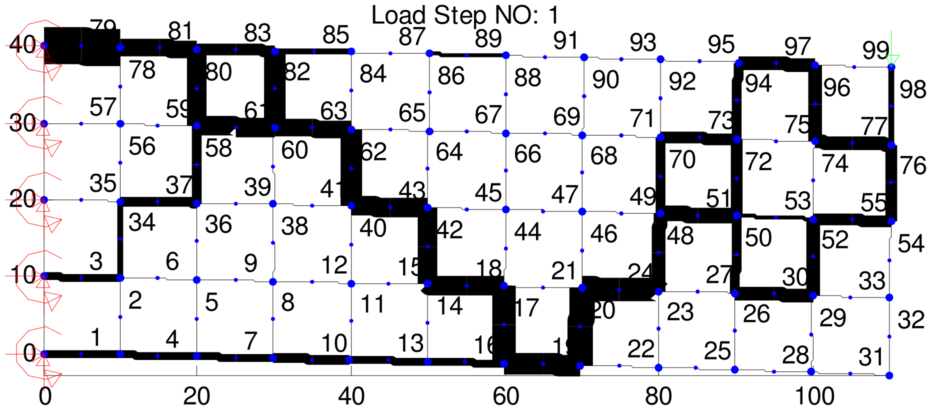


Figure 4.10: Result structure of problem 3 by PSO

The LM finds the solution faster than the PSO without the time consuming exploration and a thorough search of the design space. And that is also why the solution gained when dealing with larger scale problem is usually not the global optimum, but with smaller scale problem, it finds good solution fast.

In all the conducted experiments, the PSO for problem 2 and 3 was always terminated not due to convergence but due to no improvement for over 1/5 of maximum iterations. The setting can be changed to allow it to keep running, but from experience, with the convergence rate, it usually will end up not converged, but will rather exceed the maximum number of iterations with no or very little improvement. It is not very clear why this happens. It might be that some particles are “flying” in somewhere else of the space that don’t provide better function value and are converging slowly. Regardless of what the reason is, it seems that for large scale problems, PSO is getting better solution but uses a lot more time than LM.

4.3.1 *Pros and Cons*

With PSO and LM, It is hard to say which is better since PSO gives different result every time. But it seems that LM works better with smaller scale problems and PSO finds better solutions with large-scale problems because PSO provides a more thorough search of the design space. The struggle to choose an algorithm from these two is between the time required and better solution.

4.3.2 *More Efficient Algorithm*

Since LM works much faster but lacks thorough design space search, a new idea is then proposed. The idea is to utilize a strategy similar to the Monte Carlo Method, which includes some "randomness". From the previous result, we can see that effectiveness decreases with the increase of problem scale. But since LM requires a lot less function calls than PSO, running it several times might still be faster than running PSO just once. Thus, the approach of running it more times with different randomly generated starting points is conducted for problem 2 and 3 (larger scale problems).

The starting points were created using rand() in MATLAB which produces pseudo-random numbers between 0 and 1. By multiplying it with the difference between upper and lower bound and adding lower bound, the starting points are ensured to be within bounds. The results are in Table 4.6 below.

Table 4.6: Results of experiments to compare PSO with multiple runs of LM

	PSO		LM		
	Objective Function Value	Function Calls	Objective Function Value	Function Calls	Sum of Function Calls
Problem 2	0.000387	169689	0.000831	2280	***17955***
	0.000791	164540	0.000464	3021	

	0.000347	173679	***0.000197***	3648	
	0.000252	147212	0.000282	5985	
	0.000380	152361	0.000314	3021	
Problem 3	0.000385	303039	0.001156	5600	***32200***
	0.001682	281358	0.000386	5900	
	0.000252	265716	0.000285	7700	
	0.000423	274857	0.000854	6300	
	0.000655	281061	***0.000038***	6700	

From the results, we can see that out of five runs, LM found solution with a lot lower objective function value, and the sum of function calls for all 5 runs is still less than what PSO needs for one run by an order of magnitude.

It was discovered that with more runs starting from different starting point, LM would be a better choice for size optimization of struts. In this way, it includes a better global search and also more time efficient.

4.4 ***Summary***

In this chapter, PSO and LM are discussed and compared. From the results shown in Table 4.6, LM works more efficiently. The function evaluation time PSO requires is more than 10 times that of LM. Even given the same starting point, PSO will generate a different solution each time whereas by LM, a designer may explore the design space with multiple initial conditions and still use less computing time. Thus, we can conclude that using LM is more efficient than using PSO.

Chap.5 *Examples*

To examine the usability of the design method that is developed in the previous chapters, four examples are presented below. The first example is a relatively simple example that was already investigated in [4], so that results generated through the design method proposed in this thesis can be compared and validated. The second example is an extension of the first one. The third example is also duplicated from and validated against the literature [4, 51]. The last example is a wing design example which provides a compliant mechanism design problem as a display of the usage of the design method. All examples used the unit truss approach described in chapter two as the structural analysis method.

5.1 *A Simple 3D Cantilever Beam Example*

The first example was used in G. Graf's master thesis [4], in which he compares the optimization results to his method that completely removes the optimization process. But since this research focused on morphing structures, the same method cannot be utilized. Even so, the method proposed can be compared to the optimization results G. Graf studied in his work.

The example presented here is comprised of a cantilever beam. The dimension of the beam is: 50mm in length, 20mm in height, and 10mm in width as shown in Figure 5.1. The initial truss topology is shown in Figure 5.2. The material used in the example has an elastic modulus of 1960N/mm. It is fixed at one end and two 10N loads are applied at the other end on each corner of the beam tip. The objective is to minimize deflection while remaining a volume of 1600mm^3.

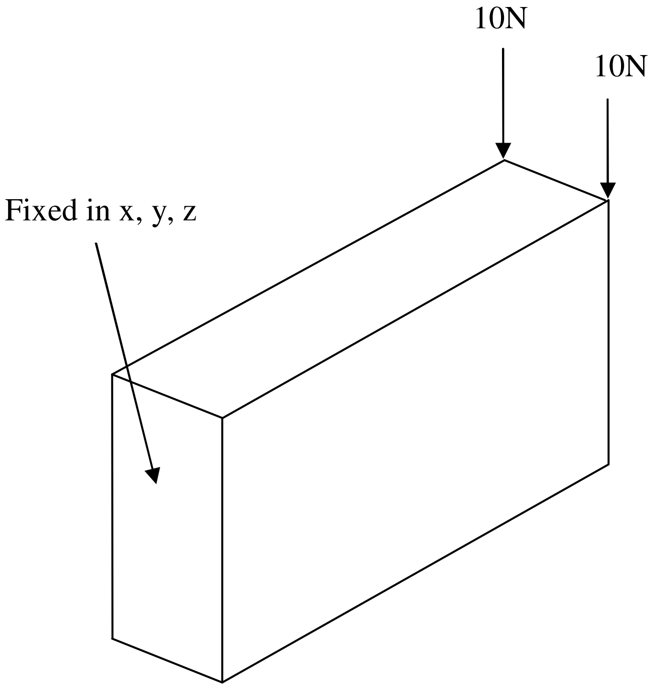


Figure 5.1: First 3D cantilever beam example

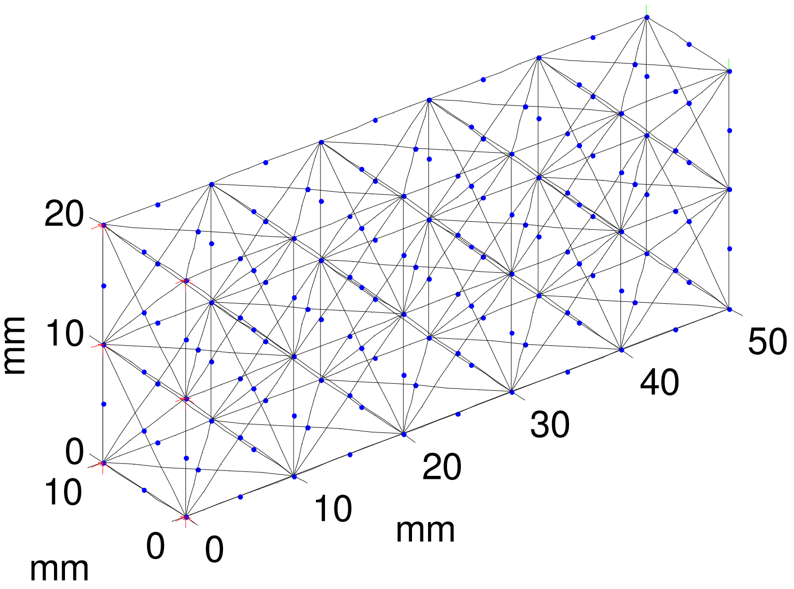


Figure 5.2: Base truss topology for first cantilever beam example

5.1.1 *Strut Approach*

The problem formulation of the optimization process is presented in Figure 5.3.

Given: The ground truss and loading conditions stated in the original problem

Find: d_i: diameter of each strut. i = 1, 2, 3…number of struts

Minimize:

$$Z = W_u \times U^2 + (V - 1600)^2$$

Where U is the strain energy, $W_u = 100$ is a weighting value, and V is the volume

Figure 5.3: Problem formulation of 3D cantilever beam example strut approach

One of the results Graf presented was solved with the Levenburg-Marquardt method using strut diameters as design variables which is the same as the strut approach described in this thesis; thus, the parametric surface/solid approach is used to compare the results. The parameter that Graf used is shown in Table 5.1 below.

Table 5.1: Parameters used for first cantilever beam example

Termination tolerance on the function value	0.001
Termination tolerance on x	0.0001
Maximum Iterations	20
Initial Configuration	All struts 5mm

Graf's resulting truss structure using the strut approach is shown in Figure 5.4, which has a volume of 1601mm^3, strain energy of 3.3537Nmm, and a maximum tip displacement of 0.3354mm; it thus gives an objective function value of 1127.

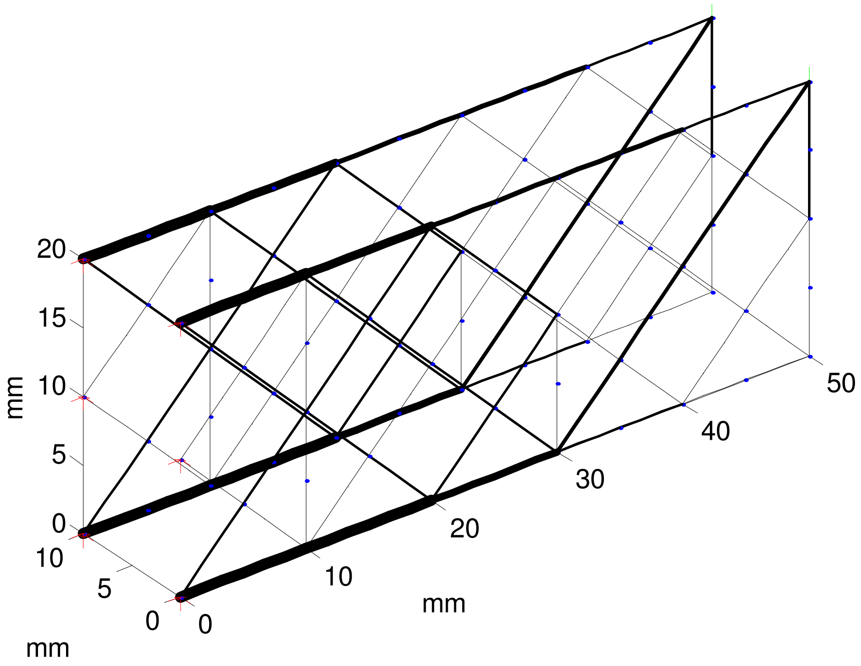


Figure 5.4: Graf's result for strut approach using LM/LSM

5.1.2 *Parametric Solid Approach*

Given: The ground truss and loading conditions stated in the original problem

Find: $\mathrm{d_{ijk}}$: control diameters. i = 1...5, j = 1, k = 1...2

Minimize:

$$Z = W_u \times U^2 + (V - 1600)^2$$

Where U is the strain energy, $W_u = 100$ is a weighting value, and V is the volume

Figure 5.5: Problem formulation of first 3D beam example parametric solid approach

Control diameters are set to be 5x1x2 which gives the parametric volume a degree of 4x0x1. In order that the results be comparable, the parameters used are the same as in Table 5.1. Optimal control diameters for all examples are provided in appendix II. Using the parametric solid approach with the same initial configuration as the starting point the optimization converged to the structure shown in Figure 5.6, which has a volume of 1608.4 mm^3, strain energy of 10.9053Nmm and an objective function value of 11962. The time used in total was 121.67 seconds, and the number of function calls was 259. The thinnest struts have a diameter of 0.6016mm and the thickest 1.2354mm. From Figure 5.6 we can see that the trend of truss diameter is similar to Graf's result shown in Figure 5.4. Thicker struts are required at the fixed end and the further away from the fixed end, the thinner the struts.

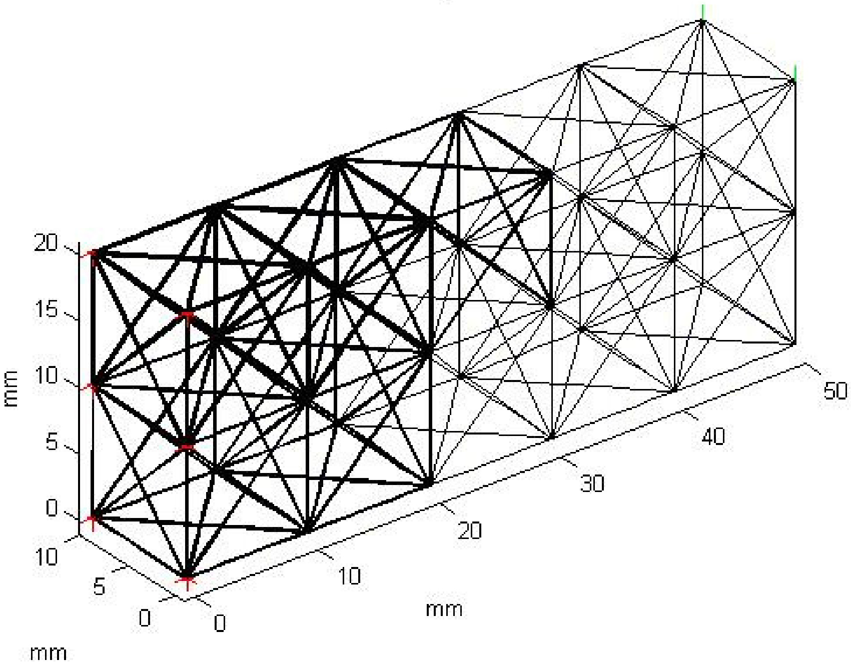


Figure 5.6: Parametric solid approach resulting structure

5.1.3 *Summary*

The unit cells used were 5, 1 and 2 in the x, y, z direction respectively. With such a small number of unit cells, the parametric solid approach doesn't perform as well as the strut approach. As can be seen from Figure 5.4, most of the struts in the middle of Graf's resulting structure are eliminated because they do not contribute to reduce the strain energy. With a volume value as an objective, eliminating unnecessary struts and enlarging other struts generated a better solution. Since the parametric solid approach gives the same diameter for all struts in the same unit cell, detailed design is not accessible and the volume cannot easily be reduced without increasing strain energy. But the parametric approach does greatly reduce the time needed comparing to other methods that employed optimization process. A brief summary is shown in Table 5.2 along with the results gained by Graf.

Table 5.2: Summary of first 3D cantilever beam example

Design Method	Volume (mm^3)	Maximum Tip Displacement (mm)	Strain Energy (Nmm)	Creation Time (s)
Identically Sized	1610	1.21	12.128	0.382
PSO	1603	0.5327	5.006	9359
LM/LSM Strut Approach	1601	0.3354	3.354	9283
LM/LSM Parametric Solid Approach	1608.4	1.0905	10.9053	121.67
Unit-Cell Library	1615	0.5547	5.547	1.639

The parametric approach has limitations when maintaining a certain volume is one of the goals. The fact that struts in the same unit cell have same size hinders its ability to eliminate unnecessary struts. The parametric approach also has a limited capacity when the design problem’s scale is relatively small. Like in this example problem, there is only one unit cell along y direction. As a result, there is no way to remove elements in the middle, since they are in the same unit cells as the struts on y=0 and y=10 plane where thicker struts are required to lower strain energy. If a problem contains more unit cells, this effect might be reduced. For example, if this design problem had three unit cells along y direction, the struts in the middle unit cells could have been eliminated. Conclusions can be made that the parametric approach is limited when the volume is one of the objectives and when the design problem’s scale is small, but this approach might perform better when the design problem is large.

5.2 *A more complicated 3D Cantilever Beam Example*

Extended from the previous example, a more complicated beam example is shown in Figure 5.7 which is 60mm in length, 30mm in height and 20mm in width. It is fixed at one end, and point forces are applied at the other end; three on the top and two at the side, each of 10N. Struts that connect fixed points are removed to reduce unnecessary variables that might affect the result. A starting truss topology is shown in Figure 5.8. The material used has an elastic modulus of 1960N/m. The objective is to minimize deflection while remaining a volume of $4400\mathrm{mm}^3$.

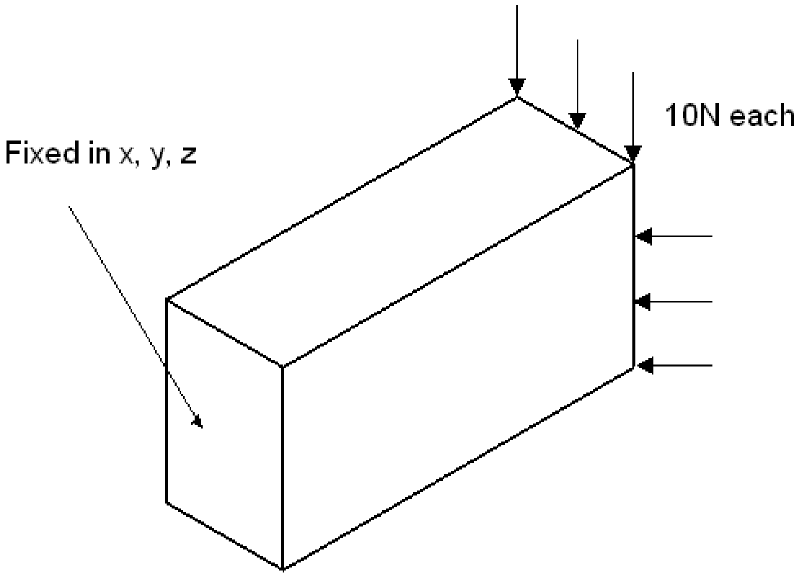


Figure 5.7: Second 3D cantilever beam example

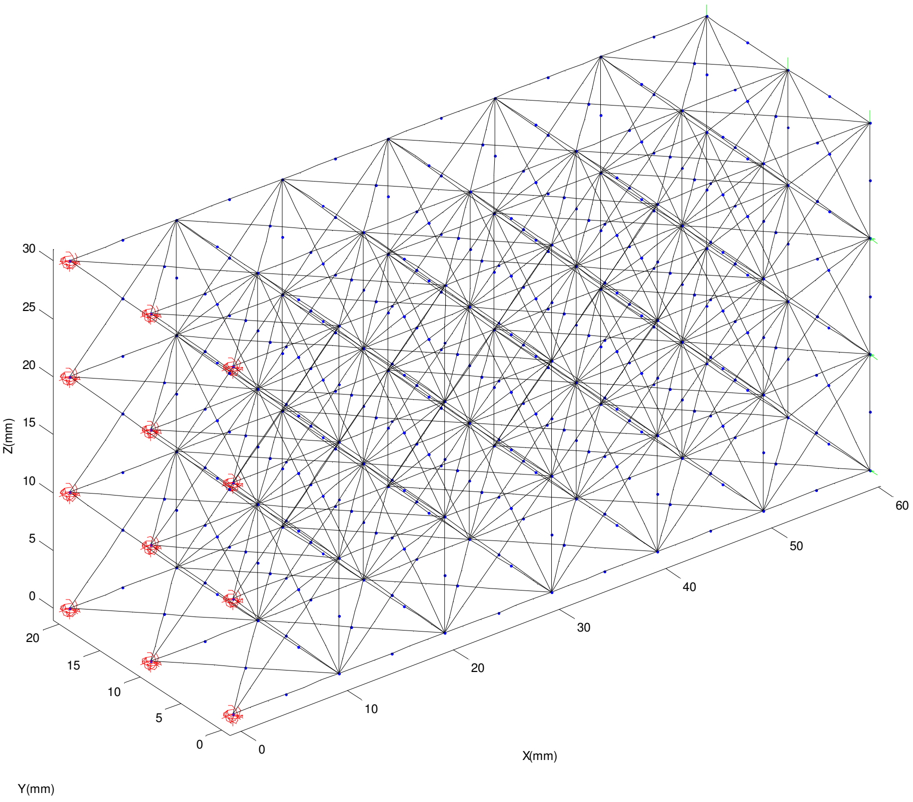


Figure 5.8: Base truss topology for second cantilever beam example

The relative stress distribution of this cantilever beam is shown in Figure 5.9. It was simulated using a solid beam model in ANSYS. Viewing from the fixed end, higher stress appears at the upper right and the lower left corner at the fixed end.

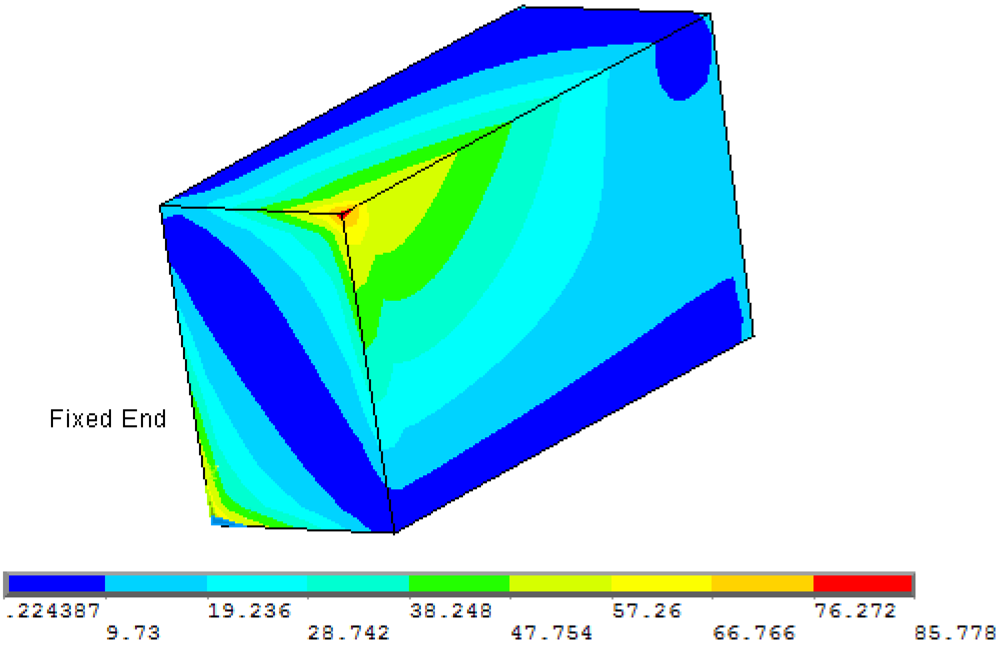


Figure 5.9: Relative stress distribution for second cantilever beam example

5.2.1 *Strut Approach*

The problem formulation of the optimization process is presented in Figure 5.10. The parameters used are shown in Table 5.3.

Given: The ground truss and loading conditions stated in the original problem

Find: d_i: diameter of each strut. i = 1, 2, 3…number of struts

Minimize:

$$Z = W_u \times U^2 + (V - 4400)^2$$

Where U is the strain energy, $W_u = 100$ is a weighting value, and V is the volume

Figure 5.10: Problem formulation for second cantilever beam example with strut approach

Table 5.3: Parameters used for second cantilever beam example

Termination tolerance on the function value	0.001
Termination tolerance on x	0.0001
Maximum Iterations	50
Initial Configuration	All struts 5mm

The resulting structure is shown in Figure 5.11 with a volume of 4622.9 mm^3, strain energy of 8.1963Nmm, max node displacement of 0.4mm, and an objective function value of 56384. It took a total number of 9940 function calls and 21 iterations to converge. Thicker struts were obtained at the two corners at the fixed end, which correspond to the stress distribution shown in Figure 5.9 where higher stress value appears.

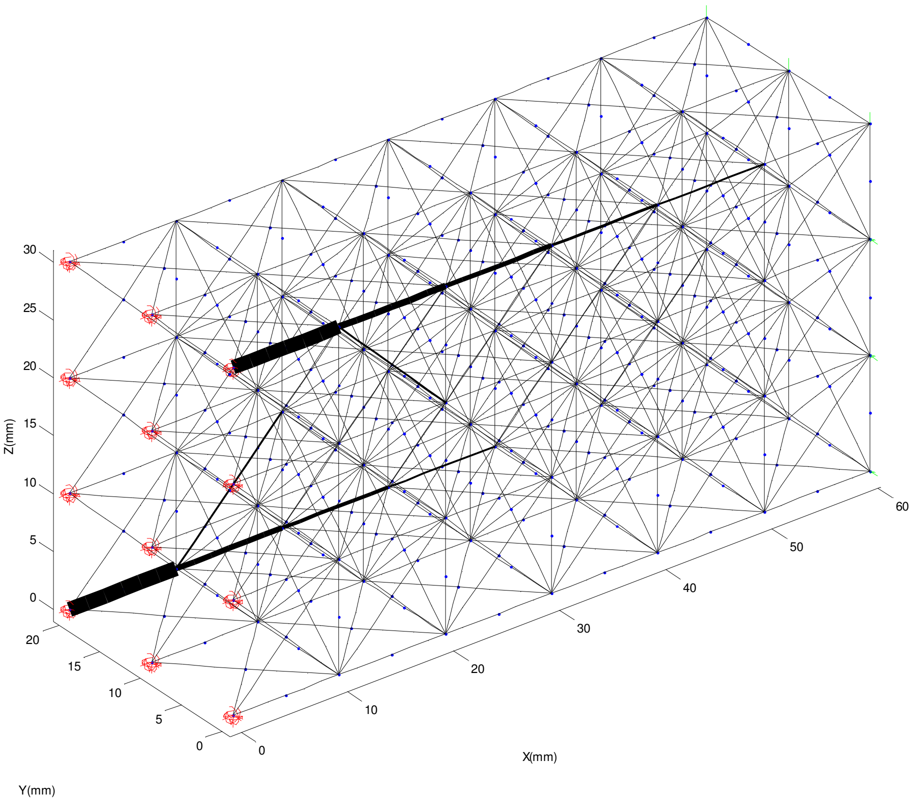


Figure 5.11: Resulting structure of second cantilever beam example using strut approach

5.2.2 *Parametric solid approach*

The problem formulation is shown in Figure 5.12. The control diameters are chosen to be 4x2x3 since it is only 2 unit cells long in the y and 3 in the z direction. The parametric volume thus has a degree of 3x1x2.

Given: The ground truss and loading conditions stated in the original problem

Find: d_{ijk}: control diameters. i = 1...4, j = 1...2, k =1...3.

Minimize:

$$Z = W_u \times U^2 + (V - 4400)^2$$

Where U is the strain energy, $W_u = 100$ is a weighting value, and V is the volume

Figure 5.12: Problem formulation for second cantilever beam example with parametric solid approach

The parameters used are the same as in Table 5.3. The resulting structure is displayed in Figure 5.13 which has a volume of 4412.6mm^3, strain energy of 23.3618Nmm, max node displacement of 1.1676mm and an objective function value of 54737. The total number of function evaluations is 1309. Although it finished at the last iteration, the 50th one, it did converge. This result is similar to the strut approach result in terms of the distribution of thicker struts and it also corresponds to the higher stress distribution in Figure 5.9. The unit cells at the upper right and the lower left have larger diameter value.

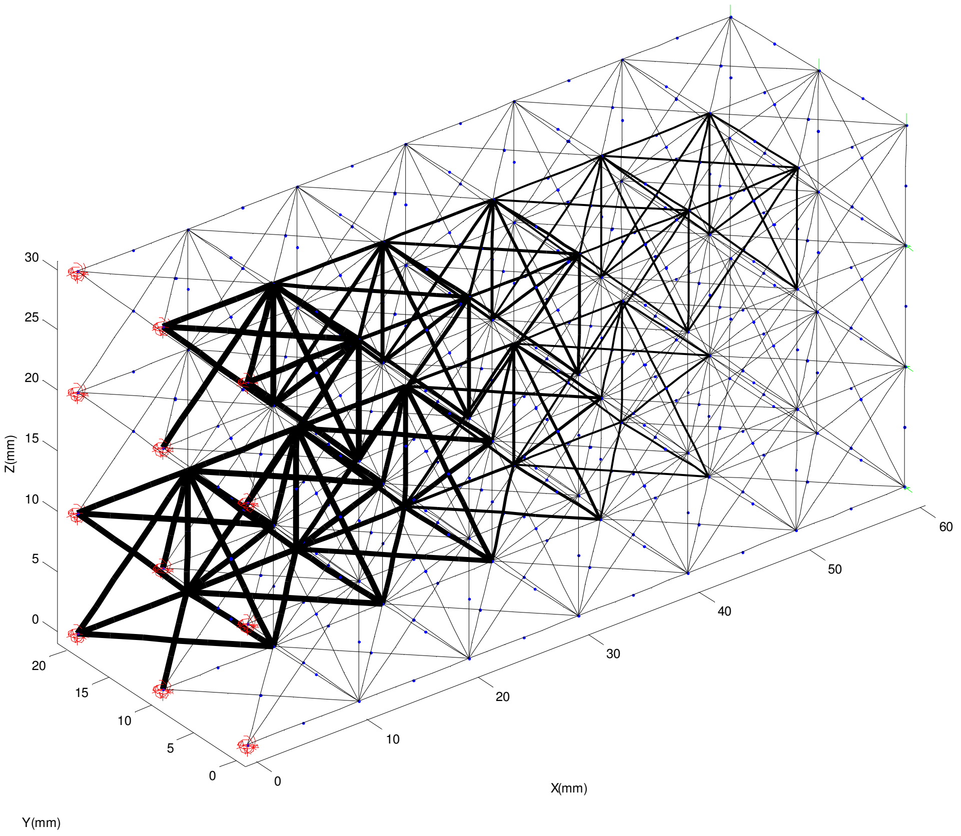


Figure 5.13: Resulting structure of second cantilever beam example using parametric solid approach

5.2.3 *An alternative assignment of parametric value*

Instead of assigning parametric values to unit cells, this alternative approach assigns u, v and w to the struts by the coordinates of the center of each strut. Because this example is composed of cuboids, the u, v and w direction perfectly correspond to the x, y and z direction respectively. Since the parametric values are in the range of 0 to 1, the u value of a strut is the strut's center node x coordinate divided by entire structure's total length in x direction. The same principle applies to calculating v and w by y and z coordinates. A 3x1x2 parametric volume was used.

The results are shown under two different objective function settings. The resulting structure of the second run is shown in Figure 5.14. Opposite to the original parametric approach, this new approach seems to work better when there is a stricter requirement on volume. The resulting strain energy and volume are both lower than that gained from the original parametric approach. The thicker struts at the two corners of fixed end correspond to the higher stress distribution shown in Figure 5.9, and the ones at the tip might be the result of the slightly increased stress value caused by the loads being applied directly to those struts. Comparing the results to original parametric approach, a conjecture may be that this new approach and the original one are complementary. When the volume constraint is stricter, this new approach might be a better choice.

Table 5.4: Results of alternative parametric approach

Run	1	2
Volume weight	1	100
Result volume	5034.8	4409.3
Strain Energy	17.7412	20.0825
FunCalls	2625	2623
ObjFunValue	4.34E+05	4.04E+04

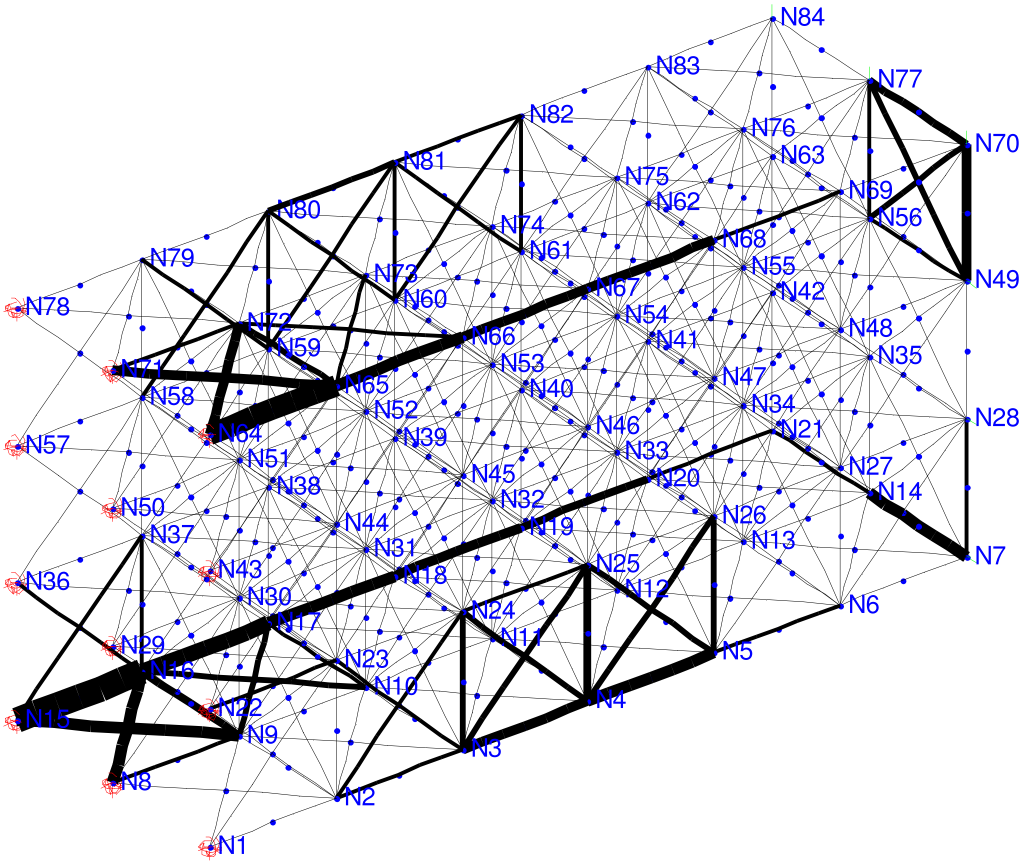


Figure 5.14: Alternative approach result structure for second beam example

5.2.4 *Summary*

When the problem is extended, the strength of the parametric solid approach is shown. Not only did it require fewer function calls, but also it found a solution that has lower objective function value. The reason the strut approach performed worse is not apparent; a rational assumption would be that it converged to a local minimum. Although the solution found by using the parametric solid approach doesn't have a smaller value of strain energy, which is a result of the objective function used, the volume is closer to the target and thus the objective function value is lower than that gained from using the strut approach. It is worth highlighting that the alternative diameter calculation method results in the lowest objective function value. Although it was dominated by the volume objective, the solution obtained is better than the original diameter calculation method both in strain energy and volume.

The parametric approach is particularly useful when the design problem contains more struts. A large number of struts renders a large design domain and raises difficulties in optimization for the strut approach. This fact corresponds to the original intension of using parametric approach, and the reduced design space can allow parametric approach to find better solutions.

Table 5.5: Summary of second cantilever beam example

Design Method	Volume (mm^3)	Strain Energy (Nmm)	Function Calls	Objective Function Value
Original	110805	1.1643	NA	$1.13x10^{10}$
LM/LSM Strut Approach	4622.9	8.1963	9940	$5.64x10^4$
LM/LSM Parametric Solid Approach	4412.6	23.3618	1309	$5.47x10^4$
Alternative parametric Approach	4409.3	20.0825	2623	$4.04x10^4$

5.3 *A 2D Beam Example*

This example was used both in Qi Xia and Yu Wang 's paper: "An optimization-based approach for the design of lightweight truss-like structures" and Graf's master's thesis: "Development of Specialized Base Primitives for Meso-Scale Conforming Truss Structures." The following statement was used in Qi Xia and Yu Wang's paper and quoted in Graf's.

> "The structure is loaded with a concentrated vertical force of P=200kN at the center of the top edge and is supported on two hinges at the bottom-right corner and the bottom-left corner. The design domain is a rectangle of size L=3m, H=1m. The beams of the structure has[sic] a circular cross-section with the diameter h_0=0.02m… …the upper bound of the

material volume [is] $\overline{V}$ =0.02m^3. The penalty parameter in topology optimization is p =2, and the lower bound of topology variables $\underline{\rho}$=0.04. "

The three different base truss topologies used are shown in Figure 5.15~Figure 5.17. The initial truss topologies used by Qi Xia and Graf are slightly different. Since the parametric surface approach requires quadrilateral cells, the base truss topology used for parametric surface approach will be the one shown in Figure 5.17. The goal is to minimize compliance while maintaining the material volume. Because they are essentially different design problems given different starting truss topologies, to better compare the results, the different starting configurations are compared in Table 5.6.

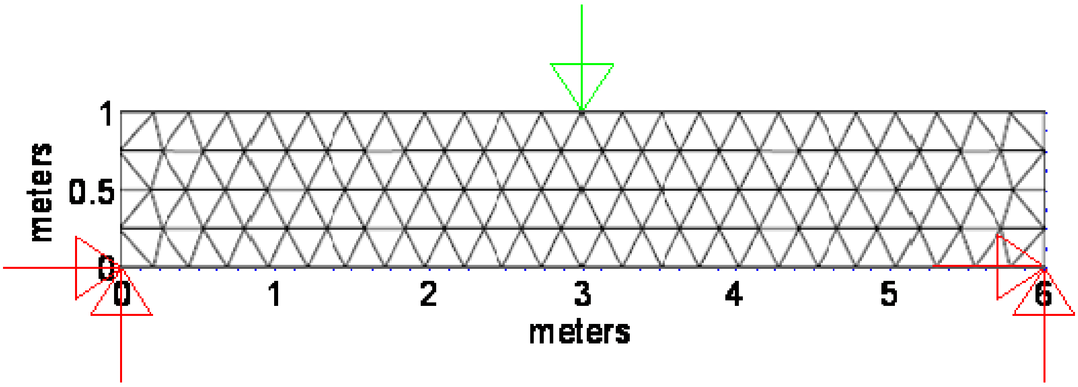


Figure 5.15: Base truss topology used by Qi Xia and Yu Wang

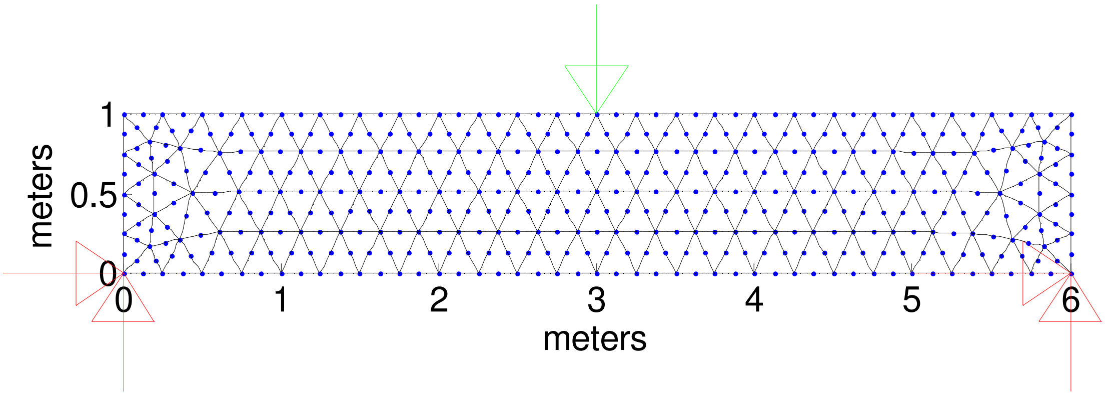


Figure 5.16: Base truss topology for 2D beam used by Graf

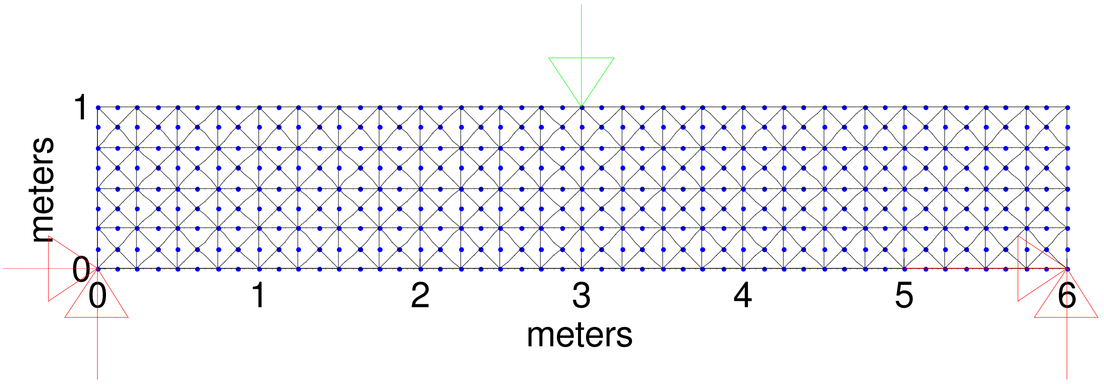


Figure 5.17: Base truss topology for 2D beam example parametric surface approach

Table 5.6: Comparison of starting configuration

	Compliance (Nm)	Volume (m^3)
Xia and Wang's Triangular Model	3325.96	0.0264
Graf's Triangular Model	3017.8	0.0271
Quadrilateral Model	2306.9	0.0386

5.3.1 *Strut Approach*

The problem formulation given in Graf's thesis is as below:

Given: The problem statement and ground structure for the second example

Find: ρ_e, a dummy size variable for each of the 328 struts

Satisfy: No specified constraints

Minimize:

$$\min_{\rho_e} f(x) = J^2 + w_v (V - 0.02)^2 + w_p {P_d}^2$$

Where: $D_i = 0.02 \times \rho_e^2$ are the truss strut diameters, penalized towards the minimum or maximum diameter,

$J = f^T u$ is the compliance,

$V = \sum r_k^2 l_k \pi$ is the truss structure volume, and

$P_d = \Sigma(D_k - 0.02)$ is a diameter penalty on all struts whose diameters are over 0.02m

w_v=10^{12} and w_p =10^8

The parameters that Graf used are shown in Table 5.7 and the resulting structure is shown in Figure 5.18. Implied in Graf's thesis, the optimization stopped due to reaching the number of maximum iterations which means it did not converge. Related data of this result is shown in Table 5.8.

Table 5.7: Parameters used for 2D beam example

Termination tolerance on the function value	0.01
Termination tolerance on x	0.001
Maximum Iterations	20
Initial Configuration	$\rho_e = 0.8$

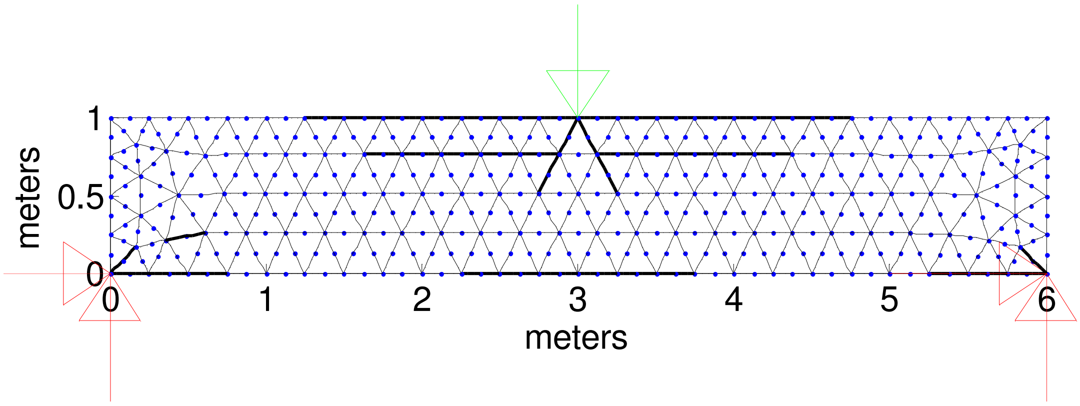


Figure 5.18: Graf's resulting structure for 2D beam example using strut approach

Table 5.8: Graf's optimization result for 2D beam example using strut approach

Optimization Time (seconds)	6990
Function Calls	6976
Final Objective Function Value	$1.19\text{x}10^7$
Iterations	20
Final Volume (m^3)	0.0201
Final Compliance (Nm)	3432.2

For the purpose of a more clear comparison, I chose to implement the strut approach using a quadrilateral model also. The resulting structure is shown in Figure 5.19. The strut size distribution is similar to Graf's result where thicker struts appear at the lower corners and the upper and lower middle part. As can be observed in Table 5.9, the resulting structure gives a very promising result in terms of the goal of this example.

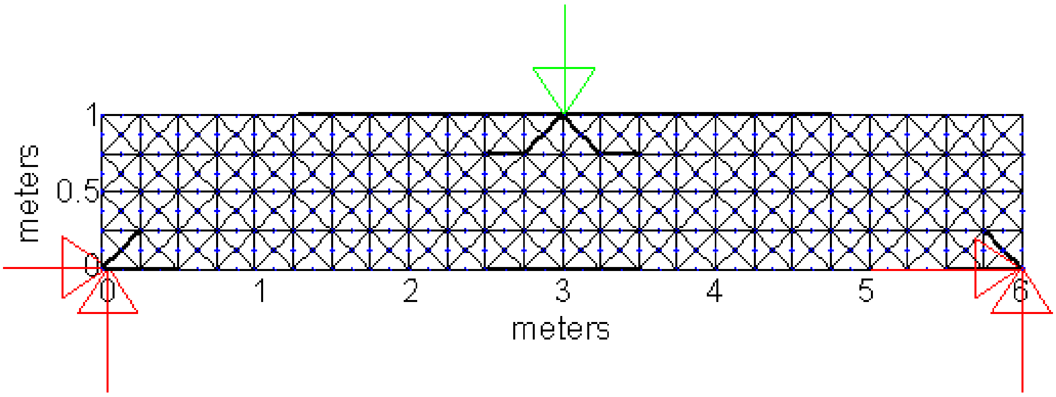


Figure 5.19: Resulting structure of 2D beam example using strut approach with quadrilateral model

Table 5.9: Result of 2D beam example using strut approach with quadrilateral model

Optimization Time (seconds)	1988
Function Calls	3316
Final Objective Function Value	$1.06\text{x}10^7$
Iterations	7
Final Volume (m^3)	0.02
Final Compliance (Nm)	3234

5.3.2 *Parametric Surface Approach*

The problem formulation is essentially the same, only the design variable is changed to control coefficient as has been used throughout this thesis. A set of 4x4 coefficients were used. The parameters used are the same as the ones shown in Table 5.7. The resulting truss structure is shown in Figure 5.20 which gives related data as shown in Table 5.10.

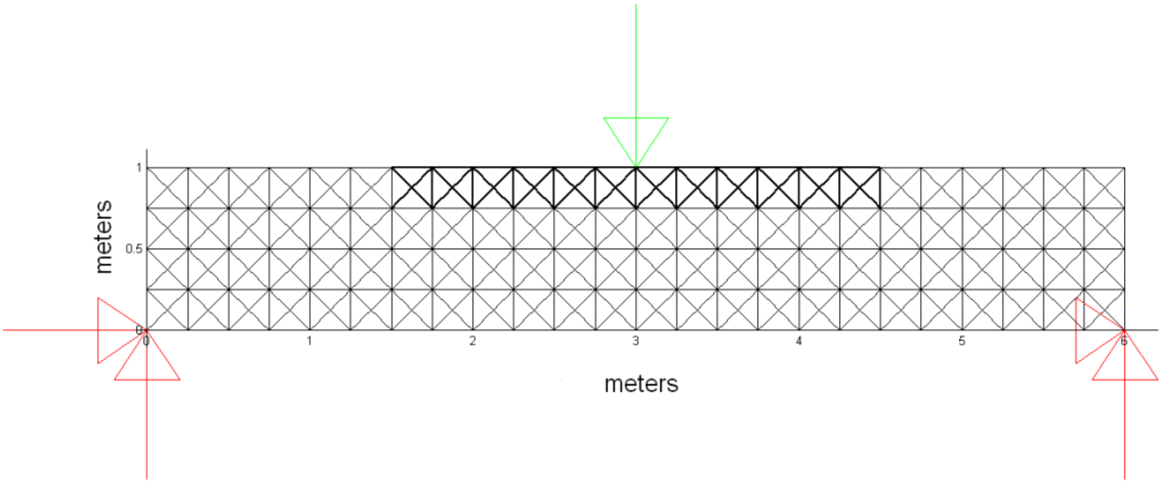


Figure 5.20: Resulting structure of 2D beam example using parametric surface approach

Table 5.10: Result of 2D beam example using parametric surface approach

Optimization Time (seconds)	94.31
Function Calls	132
Final Objective Function Value	1.35x10^7
Iterations	17
Final Volume (m^3)	0.0206
Final Compliance (Nm)	3619.1

As Graf stated in his thesis, since the volume cannot be explicitly constrained, it was added in the objective function as a penalty term. This caused the resulting objective function value to be fairly large. This objective function set up has limited the performance of the parametric surface approach. Comparing the results, it can be observed that the volume dominated the optima searching which is reasonable since it was added in the objective function with a very large weighting value. The limitation of the parametric surface approach was therefore shown. The reason that the parametric surface approach was outperformed is that when one strut is required by the loading condition to be thick to reduce compliance, other struts in the same unit cells have to be set as thick. From Table 5.6 we know that quadrilateral model started out with a larger volume. When optimization seeks to reduce the volume, compliance is sacrificed. This is especially obvious when comparing Figure 5.19 and Figure 5.20. The thicker struts in the lower corner of Figure 5.19 are not seen in Figure 5.20.

5.3.3 *Change of objective function*

With the limitation identified, a modified objective function is used. The weighting value on volume is changed from 10^{12} to 10^{6}. Also, the parametric surface used is of degree 7x3 with an 8x4 set of control diameters. Problem formulation will not be listed again since the only change is in the value of w_v.

The results are shown in Table 5.11, and the resulting structures in Figure 5.21. Both approaches converged within the given maximum number of iteration.

Table 5.11: Result of 2D cantilever beam with a modified objective function

	Original	Strut Approach	Parametric Surface Approach
Optimization Time (seconds)	NA	1939.7	496.05
Function Calls	NA	2076	511
Objective Function Value	$3.51\text{x}10^8$	$1.29\text{x}10^7$	$7.09\text{x}10^6$
Iterations	NA	4	14
Final Volume (m^3)	0.0386	0.0177	0.0311
Final Compliance (Nm)	2306.9	3535.7	2625.0

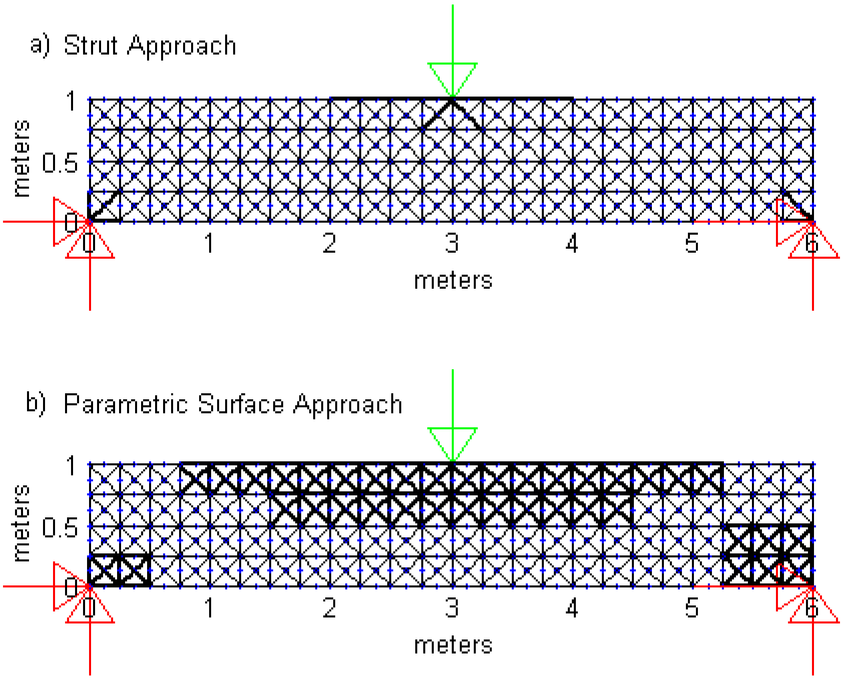


Figure 5.21: Resulting structures for 2D beam example with modified objective function.

5.3.4 *Summary*

A summary is presented in Table 5.12 including all the results from [4]. By reducing the weighting value of volume, the parametric surface approach was able to attain a lower objective function value than strut approach. This result supported the conjecture made in the previous section that the volume constraint restricted the parametric surface approach's performance. As illustrated in Figure 5.21, the struts at the lower corners were chosen to be thicker and thus give a lower objective function value. Strut approach obtained a higher objective function value. The same assumption is made that with the larger design domain, the result converged to a local optimum and gave a higher objective function value than parametric surface approach.

The result further proved the conclusions drawn in previous examples, that the volume objective might be a hindrance to the parametric approach and that parametric approach performs better with larger design problems.

Table 5.12: Summary of 2D beam example

	Method	Volume	Compliance	Time (s)
Xia and Wang's	Original	0.0264	3325.96	NA
	Optimized	0.02	3595.19	34.13
Graf's Triangular Model	Original	0.0271	3017.8	NA
	PSO	0.0196	4149.6	4754
	LM/LSM (strut approach)	0.0201	3432.2	6990
	Unit Cell library	0.0199	3400	1.70
Quadrilateral Model	Original	0.0386	2306.9	NA
	Strut approach	0.02	3234	1988
	Parametric Surface Approach	0.0206	3619.1	94.31
Quadrilateral Model with different objective function	Original	0.0386	2306.9	NA
	Strut Approach	0.0177	3535.7	1939.7
	Parametric Surface Approach	0.0311	2625	496.5

5.4 *Wing Example*

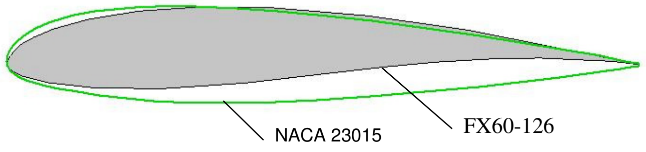


Figure 5.22: Airfoil cross section

So far, the previous examples have been structural problems. Here a compliant mechanism design example is presented. This wing design example is an extension of the example used in [8]. The purpose is to enable the wing cross section to morph from NACA23015 to FX60-126 (shown in Figure 5.22) when actuators are applied. The change of airfoil shape under different operational environments to meet different requirements can improve fuel efficiency. This example will be illustrated both in 2D and 3D. The unit truss approach, which is used for all the previous examples, is also used for structural analysis in this example. The problem formulation related to the ones presented in Figure 3.6, Figure 3.7, Figure 3.11 and Figure 3.12 is used. However, to investigate the shape-changing ability, only the deflection deviation from target is considered. By minimizing the difference between the actual deflection and the desired deflection of certain nodes, the desired shape can be attained.

5.4.1 *2D Airfoil*

A 2D example with 8x2 unit cells is explored. The initial truss topology is as shown in Figure 5.23 with 9 pairs of 30N forces applied. The goal is to find the proper strut diameter combination so that when forces are applied, the outer shape of the airfoil will change from NACA23015 to FX60-126. The outer struts have a constant diameter of 5mm.

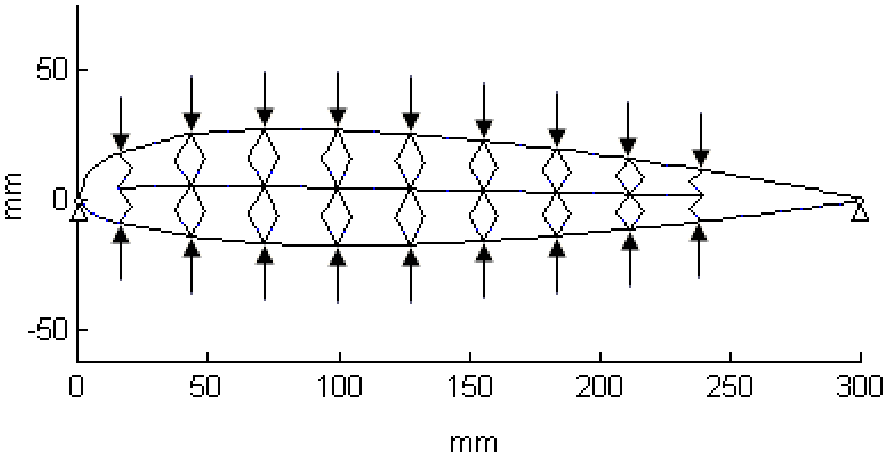


Figure 5.23: Initial truss topology of 2D airfoil example

Several different approaches are investigated, in addition to the strut and parametric surface approach, a “Unit Cell Approach” is also tested. This unit cell approach is very similar to the parametric surface approach in the way that it also takes each unit cell as a unit and struts that are considered to be in one unit have the same diameter. But, instead of controlling diameters across all unit cells, each unit cell itself has its own assigned diameter value. This way, the number of design variables is the same as number of unit cells.

The problem formulation is related to the ones given in Figure 3.6, Figure 3.7, Figure 3.11 and Figure 3.12. Because the previous examples demonstrated that with a volume objective, the parametric approach usually does not perform well, in order to understand the performance without that limitation, the volume objective is excluded. Also, it was found that the initial truss topology could not conform to the desired profile without yielding. Therefore, in order that the resulting deformed shapes can be clearly compared, the penalty function is also eliminated. This gives us the problem formulation in Figure 5.24 where Defl denotes the deflection. Subscriptions actual and target distinguish the resulting and desired deflection while norm denotes a term used to normalize difference between actual and desired deflection. The $\mathrm{Defl}_{\mathrm{norm}}$ was chosen to be a constant that is approximately the average of the desired deflection from initial profile to desired profile. The 18 points in interest correspond to the points on outer struts, same as the ones with forces applied on.

Given: The ground truss and loading conditions stated in the original problem

Find:

1. $\mathrm{d_i}$: diameter of each strut. i = 1...72 (number of struts)
2. $\mathrm{d_i}$: diameter of struts in each unit cell. i = 1...16 (number of unit cells)
3. $\mathrm{d_{ij}}$: control diameters. i = 1…3, j = 1...2 (3x2 control diameters)
4. $\mathrm{d_{ij}}$: control diameters. i = 1…4, j = 1...2 (4x2 control diameters)

Minimize:

$$\mathrm{Z} = \frac{1}{18}\sum_{\mathrm{i}=1}^{18}\left(\frac{\mathrm{Defl}_{\mathrm{i}_{\mathrm{actual}}} - \mathrm{Defl}_{\mathrm{i}_{\mathrm{target}}}}{\mathrm{Defl}_{\mathrm{norm}}}\right)^2$$

Figure 5.24: Problem formulation for 2D airfoil example

Figure 5.25 shows the 3x1 parametric surface (4x2 control diameters) used for this example. The relationship between the control diameters and the diameter value for each unit cells is demonstrated. Control diameters are shown in red dots whereas diameter values for each unit cell are shown in blue dots.

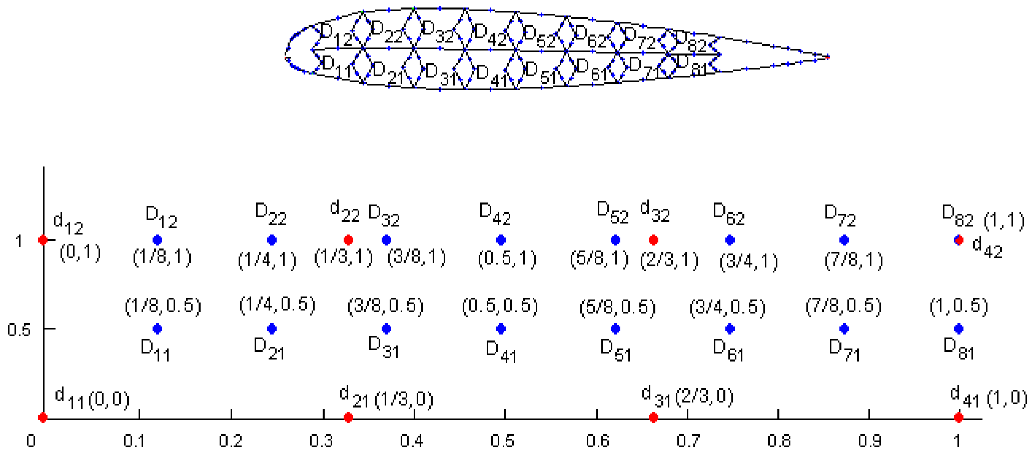


Figure 5.25: The 3x1 parametric surface used for 2D airfoil example 4x2 parametric approach

The different approaches used different design variables indicated numbers one through four under "Find." From one to four, they are: strut approach, unit cell approach, parametric surface approach with 3x2 coefficients and parametric surface approach with 4x2 coefficients respectively. The parameters used are presented in Table 5.13.

Table 5.13: Parameters used for 2D airfoil example

Termination tolerance on the function value	1e-6
Termination tolerance on x	1e-8
Maximum Iterations	300
Maximum Function Calls	30000
Initial Configuration	All struts 1mm

i Results

The results are shown in Table 5.14 and the resulting structures are displayed in Figure 5.26~Figure 5.30 where the target profile is portrayed by the red line. As shown in Figure 5.27, using the strut approach generated a fairly good result. The profile fits the target profile quite well, and the objective function value is as low as 0.0282. Some struts in the middle were chosen to be thicker to maintain the profile's position so that it won't deform like that in Figure 5.26. Just as expected, using more design variables gives a lower objective function value at the end, but it also requires more objective function evaluation. Comparing the unit cell approach with both the strut approach and the parametric surface approach, the limitation can be observed. Limitation was first encountered with the target profile being so narrow at the right end. With strut approach, diameters can be chosen so that some struts are thicker and some thinner, to maintain the profile shape. But when unit cells are used as design variables, the upper part is chosen to be thicker to maintain the shape, and the lower part is almost eliminated to obtain a narrow profile. With the unit cell approach, each adjacent unit cell can have its own diameter value without any restriction. But the parametric approach involves fitting, in this case, a 2nd or 3rd degree polynomial to the strut size (along x coordinate); therefore, the variation between neighboring unit cells is somewhat restricted. The strut size variation from Figure 5.28 does not change smoothly. This results in the parametric approaches getting a higher objective function value even than the unit cell approach.

Table 5.14: Results of 2D airfoil example

	Starting Topology	Strut Approach	Unit Cell Approach	3x2 Parametric Approach	4x2 Parametric Approach
Objective Function Value	5.4071	0.0282	0.4474	0.7323	0.6632
time (s)	NA	2521.6	431.4	12.3	25.8
Iteration	NA	300	218	12	21
Function Call	NA	22151	3834	100	209
Converge	NA	yes	yes	yes	yes

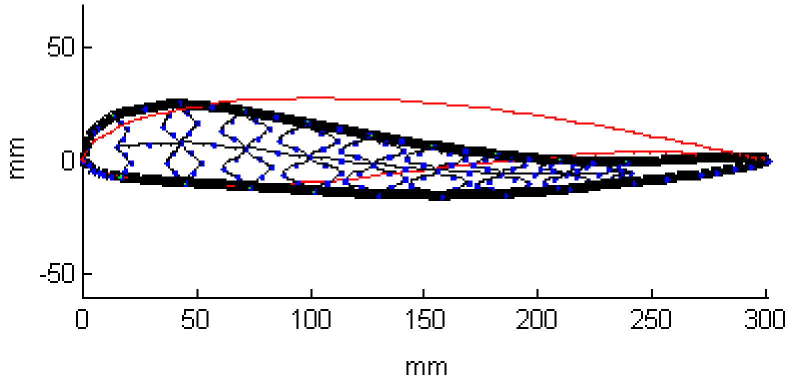


Figure 5.26: 2D airfoil starting truss topology

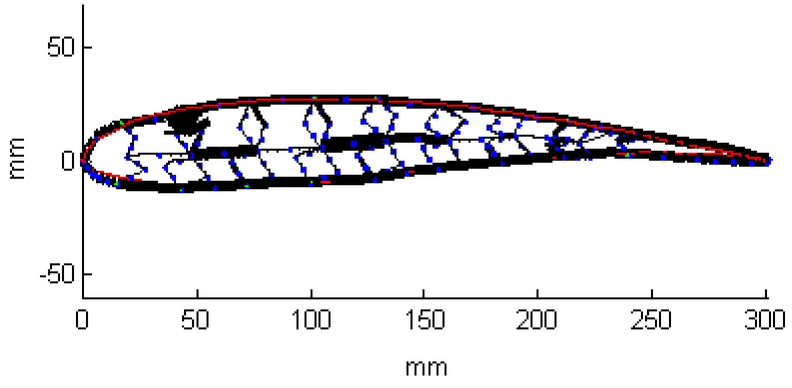


Figure 5.27: Resulting structure of 2D airfoil example using strut approach

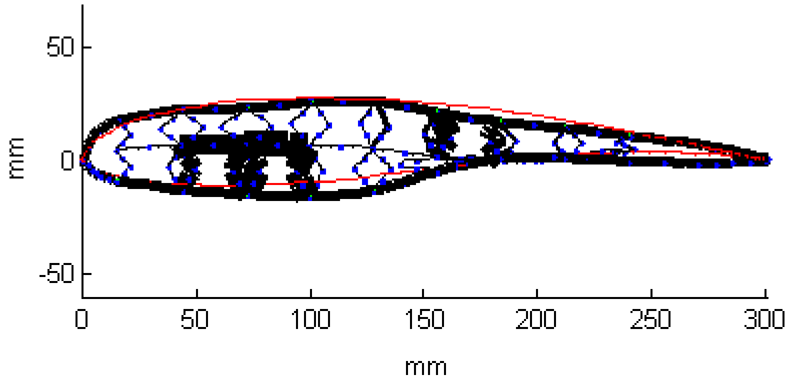


Figure 5.28: Resulting structure of 2D airfoil example using unit cell approach

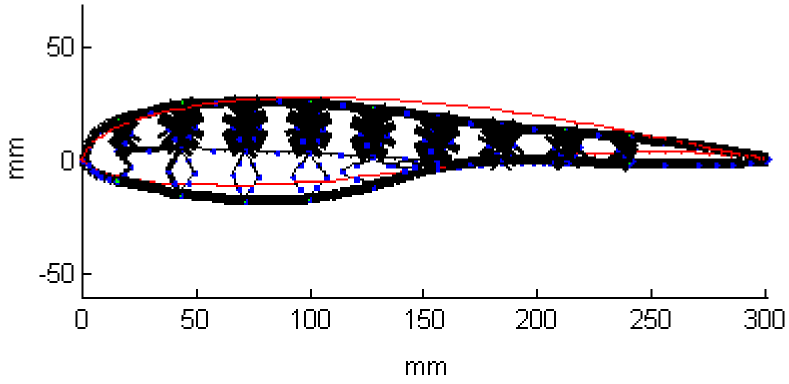


Figure 5.29: Resulting structure of 2D airfoil example using 3x2 parametric approach

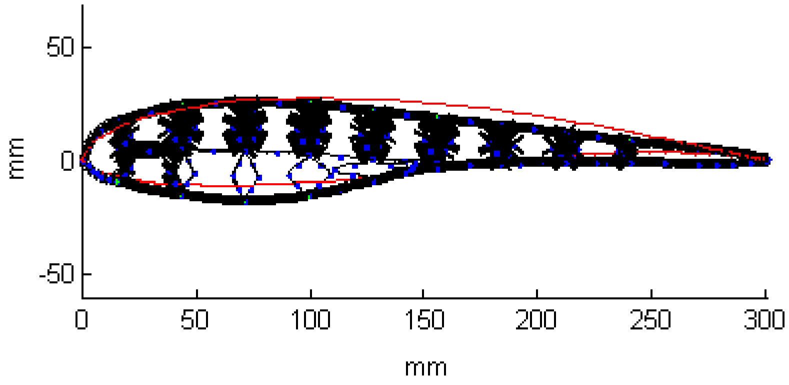


Figure 5.30: Resulting structure of 2D airfoil example using 4x2 parametric approach

The strut approach result shows that smooth variation of strut size is not always the case; as a result, the parametric approach does not perform as well with complaint mechanisms as with structural design problems.

5.4.2 *3D Wing*

The airfoil example is extended to a 3D problem using the following parameter settings [52] (Table 5.15):

Table 5.15: Parameters used to generate 3D wing

Span	Root Chord	Taper	Tip Chord	Sweep
(mm)	(mm)	(%)	(mm)	(deg)
501.8	203.2	77.6	157.6	34.3

The strut approach and the parametric solid approach are used to find the optimal solution. The number of unit cells is chosen to be $8 \times 3 \times 2$, giving the cross section shown in Figure 5.31 (in which size differs through the span.) And the base truss topology generated is presented in Figure 5.32. It has 594 elements. The outer struts are set as having constant diameter of 3mm; additionally the struts connect between cross-sections which don't affect the wing cross-section shape. Other struts' sizes, a total number of 432 diameters, are found by the optimization process. The problem formulation for this example is given in Figure 5.33.

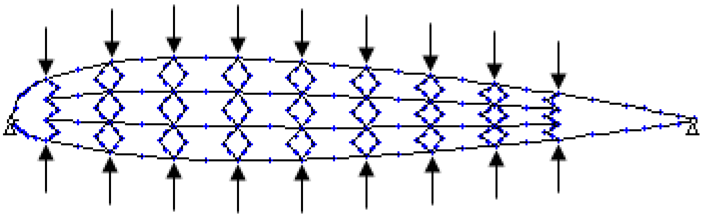


Figure 5.31: Structure of wing example at each cross section

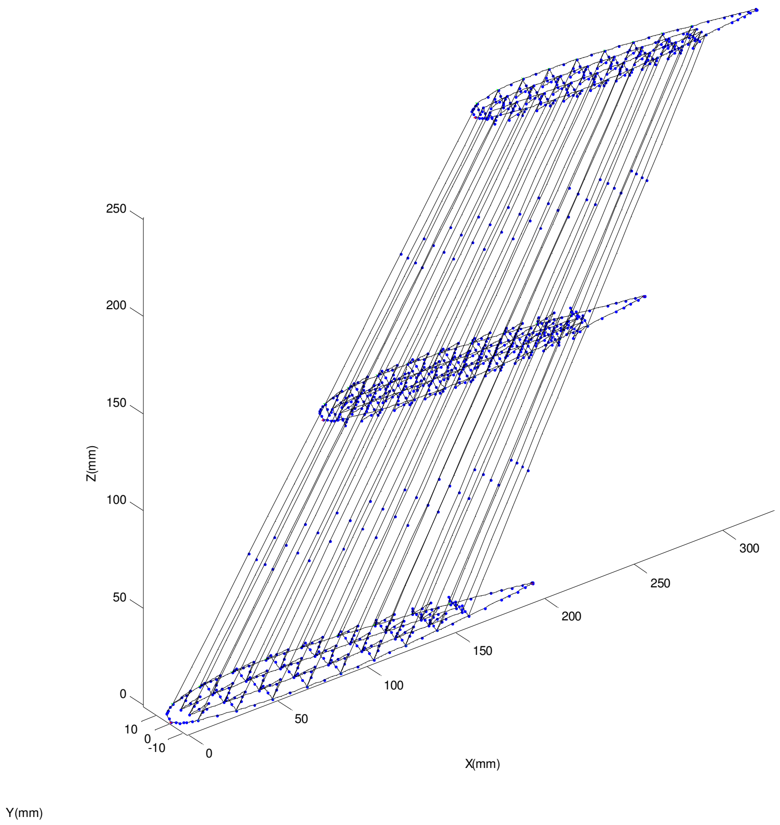


Figure 5.32: Base truss topology of wing example

Given: The ground truss and loading conditions stated in the original problem

Find:

1. $\mathrm{d_i}$: diameter of each strut. i = 1…432 (number of struts)
2. $\mathrm{d_{ijk}}$: control diameters. i = 1…4, j = 1…3, k = 1…2 (4x3x2 control diameters)

Minimize:

$$\mathrm{Z} = \frac{1}{54}\sum_{\mathrm{i=1}}^{18}\left(\frac{\mathrm{Defl_{i_{actual}}} - \mathrm{Defl_{i_{target}}}}{\mathrm{Defl_{norm}}}\right)^2$$

Figure 5.33: Problem formulation for 3D Wing example

The starting configuration has all other struts with diameters of 2mm which results in the objective function value of 0.6132. The difference between the starting profile and the target profile is shown in Figure 5.34 with red lines demonstrating the target. The parameters used are shown in Table 5.16.

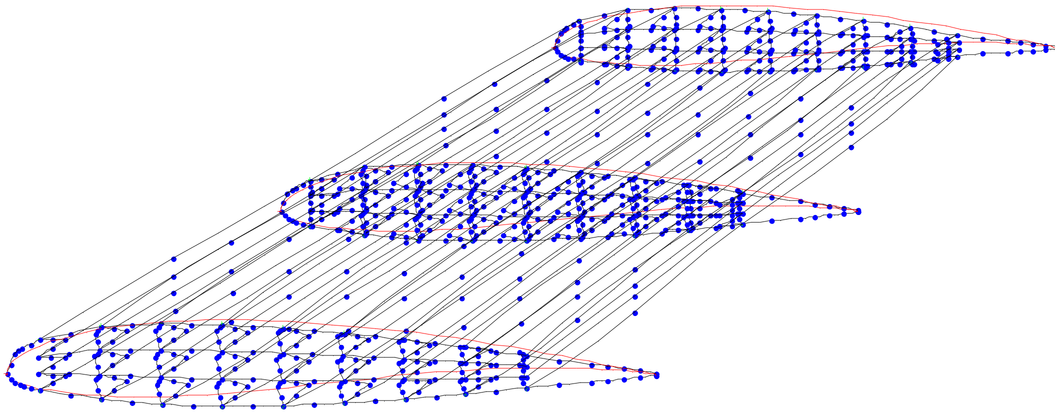


Figure 5.34: Starting truss topology profile comparing to target profile

Table 5.16: Parameters used for 3D wing example

Termination tolerance on the function value	1e-6
Termination tolerance on x	1e-8
Maximum Iterations	300
Initial Configuration	All struts 2mm

i Strut Approach

Optimization with strut diameters as design variables was unable to converge. The resulting structure has an objective function value as low as 0.0027. From Figure 5.35, the target shape is very much attained. The total number of function evaluation is 86724 which took 416751 seconds (almost 5 days).

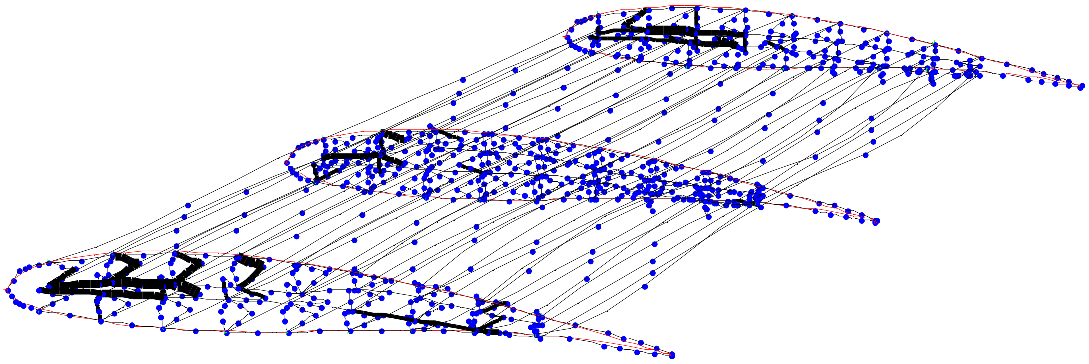


Figure 5.35: Resulting structure of 3D wing example using strut approach

ii Parametric Solid Approach

A set of 4x3x2 control diameters, parametric volume of 3x2x1 degree is used. With the parametric solid approach, the number of function evaluations is 3028, total time used is 16003, resulting in an objective function value of 0.121. The resulting truss structure is shown in Figure 5.36.

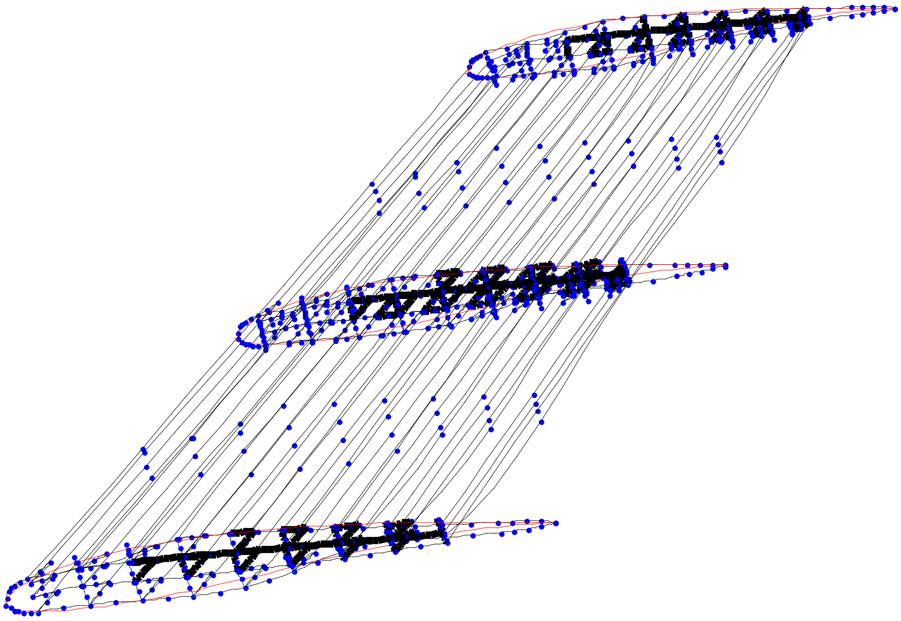


Figure 5.36: Resulting structure of 3D wing example using parametric solid approach

iii Summary

Although the strut approach still performed better in terms of the resulting objective function value, the time and function calls it required were more than 25 times that of parametric approach. For a relatively small problem like this, the time strut approach took to find a solution was fairly large. It could be imagined that with more realistic design that contains more than thousands of struts, the strut approach will not be able to handle in a practical manner.

Different objective functions were used for the 2D airfoil and the 3D wing so no direct comparison can be made. Intuitively the parametric approach did seem to obtain a better solution with the three unit cells along the y direction used in the 3D problem than the two used in the 2D problem. This again shows that the parametric approach might work better if the problem contains more unit cells. The strut size variation shown in Figure 5.35 might suggest that the parametric approach works better with structural design problems than with compliant mechanisms.

Table 5.17: Summary of 3D wing example

Methods	Objective Function Value	Function Calls	Time (s)
Original	0.6132	NA	NA
Strut Approach	0.0027	86724	416751
Parametric Solid Approach	0.121	3028	16003

5.5 *Optimization time*

To understand the relationship between the number of design variables and the number of function calls, multiple runs with same parameter settings for optimization and randomly generated starting points were conducted using the beam examples in the previous sections. The numbers of function calls used are listed in Table 5.18 for a clear comparison. Although the number of function calls varies with different starting points since some converge faster and some slower, generally, the ratio between the number of function calls is close to the same magnitude of the ratio between design variables.

A design with more struts will also require more time to analyze. Thus, the increase in the number of elements lengthens the structural optimization time for the strut approach by two-fold: one in the function calls required, the other in the time needed for each function evaluation. But for parametric approach, the number of design variables does not need to change with the expansion of design problem scale.

Table 5.18: Relationship between number of design variables and number of function calls

Examples	Method	Number of design variables	Number of function calls	Ratio (#funcCall/#DV)
First 3D Beam Example	Parametric Solid Approach	64	1249	19.5156
			1251	19.5469
			1477	23.0781
			1054	16.4688
			1054	16.4688
	Strut Approach	166	3689	22.2229
			3689	22.2229
			3690	22.2289
			3186	19.1928
			3689	22.2229
2D Beam Example	Parametric Solid Approach	16	272	17
			185	11.5625
			237	14.8125
			289	18.0625
			324	20.25
	Strut Approach	412	8279	20.0947
			6626	16.0825
			9106	22.1019
			6212	15.0777
			3317	8.0510

5.6 *Summary*

The advantage of using the parametric approach as opposed to the strut approach is in the optimization time. The reduction of design variables can surely reduce the time needed for optimization. Some degradation might arise due to the nature of the parametric approach. There are two possibilities. One is that since detailed design (tuning of single strut size) is not available which often reflects on the volume constraint. When volume is strictly required to have a certain value, the resulting structure might have a downgraded performance in other aspects. The other is that when the optimal design doesn’t have a smooth variation in strut diameter, the parametric approach will not be able to obtain it.

Through the examples presented, the two hypotheses are tested and/or validated. The two hypotheses are:

Hypothesis 1: A cellular structure that responds to given actuators by morphing to the desired shape can be designed by filling the initial shape with unit cells composed of struts to obtain a starting lattice structure, then through optimization to determine the thickness of each strut.

Hypothesis 2: The number of design variables can be reduced by changing the design variables from diameters of struts to a small number of coefficients that control the distribution of strut sizes across the structure using parametric surface/solid formulations.

The first hypothesis is proved to be valid through the wing example. In which a compliant mechanism is designed. Comparing the original structure and the optimized structure, all the results conformed more or less closer to the desired shape. The decrease of number of design variables and optimization time is apparent. The reduction of time is often significant. A more detailed and thorough discussion on hypothesis test and validation is presented in chapter six.

Chap.6 *Closure*

6.1 *Summary*

Advances in rapid manufacturing technologies have not only made cellular structures feasible, but also desirable. As the manufacturing cost decreases, the better mechanical properties and lighter weight that cellular structures possess further stand out to attention. To utilize the advantage they bring, design techniques have to keep up with the manufacturing ability. While manufacturing tools are able to build thousands to millions of lattice struts in a part, the design methods currently available are not able to solve such complex design problems. Since they are generally too computational costly when optimization is applied, this work focus on reducing the optimization time especially in designing compliant mechanisms. The research questions and hypotheses are identified:

- Research question 1: How can a cellular structure be designed so that it responds to certain actuators and morphs to a desired shape?
- Research question 2: How can the number of design variables be reduced without changing the nature of strut sizing design problems?
- Hypothesis 1: A cellular structure that responds to given actuators by morphing to the desired shape can be designed by filling the initial shape with

unit cells composed of struts to obtain a starting lattice structure, then through optimization to determine the thickness of each strut.

- Hypothesis 2: The number of design variables can be reduced by changing the design variables from diameters of struts to a small number of coefficients that control the distribution of strut sizes across the structure using parametric surface/solid formulations.

Chapter two contains a literature review. An overview of both stochastic and designed cellular structures is presented. It is identified that designed cellular structure has higher stiffness and strength. Several analysis methods are mentioned. The unit truss approach is chosen to render structural analysis in this work for its ability to closely model the behavior of trusses. A history of approaches taken to design compliant mechanism was also presented. Optimizations including topology optimization and size optimization are discussed. A research gap exists such that a large number of design variables is a main obstacle to the optimization process.

Chapter three detailed each step required by the design synthesis method proposed in this work. The generating of a starting truss topology with mapping unit cells is utilized. Given the assumption that optimized structure usually have smoothly varying changes in density/strut diameter, problem formulation that changes the design variable from diameter of all struts to a small number of coefficients is also introduced. This method is carried out by applying the concept of Bezier surface and control vertices. Instead of controlling the point location with control vertices, strut size across the structure is controlled by control diameters. These two steps formulate the design problem for optimization process.

Chapter four presents a comparison between PSO and LM/LSM. PSO is chosen for its global search ability and better convergence rate compared to GA. LM/LSM is also investigated since the compliant mechanism formulation uses the quadratic objective function that forms a least-squares formulation. It was discovered that LM/LSM converges a lot faster than PSO such that multiple runs of LM/LSM will still require less time than one run of PSO. Therefore, a conclusion was made that using LM/LSM is more efficient and stable than PSO, and global search can be included in LM/LSM by using multiple starting points.

Chapter five illustrates the use of proposed design synthesis method by several examples. First 3D example is duplicated from [4]. Results are compared with Graf's result. An extension example of the first one is also presented. It is shown that the parametric approach performs better when the number of unit cells is large. A 2D example is duplicated from [4, 51]. The limitations of the parametric approach are exposed and discussed. Finally a compliant mechanism example is presented. It is investigated both in 2D and in 3D. Results are compared and discussed.

6.2 *Conclusion*

In this thesis, a design synthesis method is proposed to design compliant cellular structures. As opposed to former methods that are impractical to handle design problem with more than thousands of elements, a new formulation to reduce the number of design variables is introduced. To validate the hypothesis, some things need to be tested.

For hypothesis #1:

- An initial truss topology can be successfully obtained by filling design space with unit cells composed of struts.
- Desired shape-change can be attained through optimization process

For hypothesis #2:

- Parametric surface/solid approach can reduce the number of design variables
- Optimization result is not degraded significantly through the operation

Several conclusions are drawn concerning the answering of research questions.

6.2.1 *Base truss topology*

The methodology for creating an initial truss topology is detailed in chapter three and proven to be accomplishable through all examples presented in chapter five. The means to mesh an input surface or space into quadrilaterals or hexahedrons has been studied and proven to be possible [38]. Once the mesh is established, primitives can be mapped into it as illustrated in Figure 3.2. Successful examples are shown in Figure 5.2, Figure 5.8,Figure 5.17, Figure 5.23, Figure 5.32. Thus, it can be conclude that a base truss topology can be obtained through the method proposed.

6.2.2 *Shape-Change Compliant Mechanism*

After the base truss topology is acquired, strut size can be tuned by the optimization process to find the best combination that would give the desired shape when actuator is applied. The problem formulation for this kind of size optimization is given in chapter three. The result of executing such optimization is demonstrated in the wing example in chapter five. From Figure 5.27~Figure 5.30 and Figure 5.35~Figure 5.36, it can be clearly observed that the outer shape of the resulting structures under applied forces do conform more or less to the desired shape. Therefore, it can be concluded that such operation is valid: the target shape can be obtained through optimization.

6.2.3 *Number of design variables*

Although the number of design variables is user defined in the parametric approach, in principle, it should still be far less than the number of struts. Since there is usually more than one strut in a unit cell, the number of unit cells is less than the number of elements. The number of control diameters can be equal to or less than the number of unit cells. If there are two unit cells in the x direction, no matter what the optimal diameter value is for each, a linear variation is expected. From this extrapolation, the number of control diameters should always be less than the number of elements. From all examples in chapter five, the reduction of design variable is proven to greatly reduce computational cost.

6.2.4 *Parametric Surface/Solid Approach*

The parametric approach has two limitations. One is the fact that all struts in the same unit cells have same diameter. This affects the volume and compliant mechanism profile fitting. Usually an optimization formulation will include a volume restriction to reduce the material required. But as discussed in 5.1.3, without the ability to eliminate single elements rather than entire unit cells, it is often hard or sometimes impossible for parametric approach to reach other goals with restricted volume. But this might be solvable with the alternative diameter calculation method illustrated in 5.2.3, where the increase of volume weight actually improved its performance. The alternative diameter calculation method even outperformed both the strut approach and the original parametric approach. The results in Table 5.4 and Table 5.5 clearly showed the advantage of exploring alternative diameter calculation methods. Compared to the original diameter calculation method, better results are shown in both strain energy and volume obtained. The 2D airfoil result obtained through unit cell approach shown in Figure 5.28 demonstrates the effect of this limitation to shape conformation. In Figure 5.27 some struts in the middle were thickened to maintain the shape; but in Figure 5.28, to thicken the struts in the middle, several other struts in the same unit cells were also thickened, so the airfoil shape does not fit the target profile as well. This limitation has a reduced effect when a design problem is rather large. In principle, when the number of struts gets larger, the effect of single struts should become smaller. Therefore, it should apply that when the number of unit cells becomes large, effect cost by limitation relating to the fact that struts in one unit cell have same dimension value also diminishes. A good illustration of this will be the second cantilever beam example used in chapter five in which the parametric approach was proved to be effective and efficient.

The other limitation to consider is that the parametric approach generates structures with smoothly varying strut size. In a structural design problem, since the variation of struts is usually smooth, this limitation is less noticed. But in compliant mechanisms, as illustrated through the wing example in chapter five, the diameter of the struts actually does not vary smoothly as expected. This is especially clear in Figure 5.27. We can imagine that the distribution of strut diameter cannot easily be represented by polynomials.

From the example problems in chapter five, the parametric approach appears to be efficient with larger structural design problem. The functionality of resulting structure might not be as good for smaller problem or for problems with strict volume requirement. The parametric approach is more applicable to structural design problem than compliant mechanisms.

6.2.5 *Conclusions*

According to the results testing the four things previously addressed, conclusion can be made that the proposed hypothesis answered partially the research questions. Following conclusions can be drawn:

- An initial cellular structure can be obtained by mapped mesh approach. Then through setting the deflection difference between actual and desired shape for the nodes along surface in interest, shape-change can be attained through optimization process by finding optimal strut size combination.

- The parametric approach can successfully reduce the number of design variables thus the design space and computational time but some degradation of the functionality of resulting structure is to be expected.

6.3 *Contribution*

This work is intended to provide means to improve cellular lattice structure design synthesis methods for both structure and compliant mechanisms so that more realistic structure can be designed. The major contribution is to reduce the number of design variables which is formulated in chapter three. Along with this research, a comparison between PSO and LM is also conducted. Some major accomplishments are listed below.

6.3.1 *Level of optimization*

The introduction of the parametric approach has change the level of design variables from each strut to unit cell, even to across the entire structure. It might be applicable even to designing stochastic cellular structures. Essentially this approach controls the density throughout the whole structure. But instead of void by void, the number of values to find can be as small as a several coefficients. In utilizing to design of lattice structure, however, the stiffness and strength lattice structure provides can be maintained and optimization process is still applicable.

6.3.2 *Optimization methods*

The popularity of global search heuristics like GA and PSO is due to their ability to execute a global search, especially in lattice structure designs, when it usually involves a huge design space and many local optima. Under such condition, designers sought to have the optimization method to do a thorough search of the space and automatically generate the optimal result there is. Traditional mathematical programming (MP) methods were abandoned due to their nature of quickly converging to nearby relative optima. The comparison in chapter four proved that using traditional MP can be more efficient than global optimization. The difficulty of the high dependency on starting point can be overcome by repeating such computation with different starting points.

6.4 *Future Work*

Some areas identified to have potential improvement through further investigation are listed below.

6.4.1 *Optimization methods*

The use of LM did not allow specifying constraints. If constraints were to be added, it can only be implicitly integrated by the use of penalty function. Since the comparison results in chapter four have guided the direction back to traditional MP methods, further investigation of different optimization methods might have the potential of further improving the efficiency of optimization for lattice structure design.

6.4.2 *Topology and shape optimization*

The optimization conducted in this thesis is essentially pure size optimization. The base topology or geometry doesn't change during optimization process. If a initial truss topology, with the actuator applied, and other constraints specified using penalty function, the optimization method wouldn't be able to find a reasonable solution, and will end up either with a structure whose functionality is far from the required, or an unrealizable one. To include truss or shape optimization with parametric approach might be a valuable next step.

6.5 *Closure*

Accompanying the advances in rapid manufacturing technologies is the attention called to complex light weight structures from the automotive and aerospace industries [15]. The aerodynamic performance is highly relevant to the airfoil geometry as it influences the pressure distribution over the airfoil [10]. An airfoil with reconfigurable shape might be useful for different performance requirements under different operation environments. Current lattice structure design methods have proved to be too computationally impractical for design of a modest part [8]. This work focuses on making automated design synthesis methods a little more realizable. Hopefully computer-aided-design tools and rapid manufacturing technologies will both advance in the future to enable the realization of structures beyond the imagination today.

Appendix I: Results for Swarm Size Experiments

Dimension 10:

SwarmSize	Run	Iterations	ObjFunValue	EndReason	#Evaluation
10	1	419	93.782	GlobalMin	4190
10	2	286	97.377	GlobalMin	2860
10	3	166	86.017	GlobalMin	1660
10	4	241	86.514	GlobalMin	2410
10	5	239	97.29	GlobalMin	2390
10	6	139	95.858	GlobalMin	1390
10	7	21	99.56	GlobalMin	210
10	8	1480	97.451	GlobalMin	14800
10	9	1	70.121	GlobalMin	10
10	10	1	96.257	GlobalMin	10
20	1	102	85.914	GlobalMin	2040
20	2	309	93.609	GlobalMin	6180
20	3	1	98.486	GlobalMin	20
20	4	1	79.117	GlobalMin	20
20	5	5	93.392	GlobalMin	100
20	6	66	85.682	GlobalMin	1320
20	7	4	88.904	GlobalMin	80
20	8	2	83.414	GlobalMin	40
20	9	4	91.983	GlobalMin	80
20	10	1	96.82	GlobalMin	20
30	1	3	95.204	GlobalMin	90
30	2	232	86.619	GlobalMin	6960
30	3	5	91.104	GlobalMin	150
30	4	4	90.564	GlobalMin	120
30	5	2	93.602	GlobalMin	60
30	6	2	92.885	GlobalMin	60
30	7	220	95.824	GlobalMin	6600
30	8	266	96.539	GlobalMin	7980
30	9	2	89.009	GlobalMin	60
30	10	79	98.262	GlobalMin	2370
40	1	5	99.43	GlobalMin	200

40	2	30	89.941	GlobalMin	1200
40	3	29	96.585	GlobalMin	1160
40	4	3	87.418	GlobalMin	120
40	5	149	93.63	GlobalMin	5960
40	6	2	97.752	GlobalMin	80
40	7	132	99.973	GlobalMin	5280
40	8	1	68.235	GlobalMin	40
40	9	1	82.468	GlobalMin	40
40	10	186	73.79	GlobalMin	7440
50	1	73	97.673	GlobalMin	3650
50	2	26	94.959	GlobalMin	1300
50	3	1	67.376	GlobalMin	50
50	4	3	88.551	GlobalMin	150
50	5	2	98.66	GlobalMin	100
50	6	1	72.852	GlobalMin	50
50	7	1	82.242	GlobalMin	50
50	8	75	92.637	GlobalMin	3750
50	9	165	89.81	GlobalMin	8250
50	10	75	97.721	GlobalMin	3750

Dimension 20:

SwarmSize	Run	Iterations	ObjFunValue	EndReason	#Evaluation
10	1	4884	99.163	GlobalMin	48840
10	2	4738	99.596	GlobalMin	47380
10	3	8276	106.6	LocalMin	82760
10	4	6154	124.65	LocalMin	61540
10	5	4490	99.223	GlobalMin	44900
10	6	4472	97.397	GlobalMin	44720
10	7	6709	100.71	LocalMin	67090
10	8	9713	121.53	LocalMin	97130
10	9	5849	99.52	GlobalMin	58490
10	10	6299	122.59	LocalMin	62990
20	1	8768	99.802	GlobalMin	175360
20	2	6258	99.484	GlobalMin	125160
20	3	5421	99.989	GlobalMin	108420
20	4	4733	99.311	GlobalMin	94660

20	5	3682	99.906	GlobalMin	73640
20	6	5101	94.23	GlobalMin	102020
20	7	5154	99.469	GlobalMin	103080
20	8	4094	98.61	GlobalMin	81880
20	9	4002	99.802	GlobalMin	80040
20	10	3922	97.206	GlobalMin	78440
30	1	3656	99.987	GlobalMin	109680
30	2	3480	99.945	GlobalMin	104400
30	3	4372	99.883	GlobalMin	131160
30	4	3729	92.584	GlobalMin	111870
30	5	4244	94.163	GlobalMin	127320
30	6	3458	96.543	GlobalMin	103740
30	7	4551	99.615	GlobalMin	136530
30	8	4749	99.877	GlobalMin	142470
30	9	4130	99.895	GlobalMin	123900
30	10	5261	99.99	GlobalMin	157830
40	1	3999	94.1	GlobalMin	159960
40	2	4107	99.901	GlobalMin	164280
40	3	3743	96.957	GlobalMin	149720
40	4	4976	99.933	GlobalMin	199040
40	5	4280	98.051	GlobalMin	171200
40	6	3959	99.814	GlobalMin	158360
40	7	3858	96.851	GlobalMin	154320
40	8	2903	91.755	GlobalMin	116120
40	9	3968	98.929	GlobalMin	158720
40	10	3657	99.296	GlobalMin	146280
50	1	5442	99.533	GlobalMin	272100
50	2	4078	96.077	GlobalMin	203900
50	3	4341	99.592	GlobalMin	217050
50	4	3442	99.172	GlobalMin	172100
50	5	3452	99.947	GlobalMin	172600
50	6	3198	97.788	GlobalMin	159900
50	7	3079	98.552	GlobalMin	153950
50	8	4216	95.806	GlobalMin	210800
50	9	3462	97.68	GlobalMin	173100

50	10	3981	99.279	GlobalMin	199050

Dimension 30:

SwarmSize	Run	Iterations	ObjFunValue	EndReason	#Evaluation
10	1	6448	258.26	LocalMin	64480
10	2	7059	149.43	LocalMin	70590
10	3	8789	201.35	LocalMin	87890
10	4	7396	175.4	LocalMin	73960
10	5	6363	114.49	LocalMin	63630
10	6	6013	99.572	GlobalMin	60130
10	7	6575	122.61	LocalMin	65750
10	8	6969	213.35	LocalMin	69690
10	9	10001	223.15	ExceedMaxIter	100010
10	10	7602	193.11	LocalMin	76020
20	1	7141	99.037	GlobalMin	142820
20	2	4887	96.621	GlobalMin	97740
20	3	9009	162.39	LocalMin	180180
20	4	6628	99.825	GlobalMin	132560
20	5	6880	98.804	GlobalMin	137600
20	6	8672	218.39	LocalMin	173440
20	7	7127	211.36	LocalMin	142540
20	8	7140	145.48	LocalMin	142800
20	9	7904	123.52	LocalMin	158080
20	10	7009	155.35	LocalMin	140180
30	1	9368	107.6	LocalMin	281040
30	2	8342	174.4	LocalMin	250260
30	3	10001	186.41	ExceedMaxIter	300030
30	4	10001	127.5	ExceedMaxIter	300030
30	5	8308	162.46	LocalMin	249240
30	6	6579	93.818	GlobalMin	197370
30	7	9255	139.58	LocalMin	277650
30	8	5362	99.999	GlobalMin	160860
30	9	8329	194.44	LocalMin	249870
30	10	5521	97.999	GlobalMin	165630
40	1	6622	185.35	LocalMin	264880
40	2	10001	134.46	ExceedMaxIter	400040

40	3	7177	158.48	LocalMin	287080
40	4	4783	98.773	GlobalMin	191320
40	5	6433	99.946	GlobalMin	257320
40	6	10001	103.62	ExceedMaxIter	400040
40	7	9348	149.46	LocalMin	373920
40	8	10001	165.52	ExceedMaxIter	400040
40	9	8285	146.4	LocalMin	331400
40	10	7432	157.49	LocalMin	297280
50	1	9005	132.54	LocalMin	450250
50	2	5799	99.516	GlobalMin	289950
50	3	10001	175.32	ExceedMaxIter	500050
50	4	10001	107.6	ExceedMaxIter	500050
50	5	10001	126.57	ExceedMaxIter	500050
50	6	10001	156.56	ExceedMaxIter	500050
50	7	7008	99.06	GlobalMin	350400
50	8	5330	99.965	GlobalMin	266500
50	9	10001	119.47	ExceedMaxIter	500050
50	10	10001	132.61	ExceedMaxIter	500050

Dimension 40:

SwarmSize	Run	Iterations	ObjFunValue	EndReason	#Evaluation
10	1	7327	281.69	LocalMin	73270
10	2	7443	155.38	LocalMin	74430
10	3	8964	197.14	LocalMin	89640
10	4	6922	228.24	LocalMin	69220
10	5	6915	287.1	LocalMin	69150
10	6	7920	198.03	LocalMin	79200
10	7	6971	300.04	LocalMin	69710
10	8	7548	211.08	LocalMin	75480
10	9	6974	311.86	LocalMin	69740
10	10	7322	279.9	LocalMin	73220
20	1	10001	164.31	ExceedMaxIter	200020
20	2	10001	186.2	ExceedMaxIter	200020
20	3	7667	258.97	LocalMin	153340
20	4	10001	208.3	ExceedMaxIter	200020
20	5	8655	270.06	LocalMin	173100

20	6	10001	288.04	ExceedMaxIter	200020
20	7	8872	290.03	LocalMin	177440
20	8	10001	326.91	ExceedMaxIter	200020
20	9	10001	288.82	ExceedMaxIter	200020
20	10	8330	210.22	LocalMin	166600
30	1	10001	291.02	ExceedMaxIter	300030
30	2	10001	186.27	ExceedMaxIter	300030
30	3	7871	104.47	LocalMin	236130
30	4	10001	239.22	ExceedMaxIter	300030
30	5	10001	214.34	ExceedMaxIter	300030
30	6	10001	232.04	ExceedMaxIter	300030
30	7	10001	162.46	ExceedMaxIter	300030
30	8	7740	192.25	LocalMin	232200
30	9	8394	254.21	LocalMin	251820
30	10	8527	170.28	LocalMin	255810
40	1	10001	162.18	ExceedMaxIter	400040
40	2	10001	266.15	ExceedMaxIter	400040
40	3	8929	299.91	LocalMin	357160
40	4	10001	222.3	ExceedMaxIter	400040
40	5	10001	243.12	ExceedMaxIter	400040
40	6	7767	232.18	LocalMin	310680
40	7	7921	200.27	LocalMin	316840
40	8	10001	231.26	ExceedMaxIter	400040
40	9	10001	159.41	ExceedMaxIter	400040
40	10	10001	195.3	ExceedMaxIter	400040
50	1	7864	244.12	LocalMin	393200
50	2	9205	191.39	LocalMin	460250
50	3	9517	160.33	LocalMin	475850
50	4	10001	204.39	ExceedMaxIter	500050
50	5	10001	257.12	ExceedMaxIter	500050
50	6	10001	211.07	ExceedMaxIter	500050
50	7	10001	291.17	ExceedMaxIter	500050
50	8	10001	249.16	ExceedMaxIter	500050
50	9	10001	156.35	ExceedMaxIter	500050
50	10	10001	163.39	ExceedMaxIter	500050

Dimension 50:

SwarmSize	Run	Iterations	ObjFunValue	EndReason	#Evaluation
10	1	7310	484.46	LocalMin	73100
10	2	7473	407.55	LocalMin	74730
10	3	7184	482.29	LocalMin	71840
10	4	8027	419.6	LocalMin	80270
10	5	8028	477.36	LocalMin	80280
10	6	8167	475.09	LocalMin	81670
10	7	7616	387.77	LocalMin	76160
10	8	7233	374.85	LocalMin	72330
10	9	7474	457.01	LocalMin	74740
10	10	7579	270	LocalMin	75790
20	1	9270	284.84	LocalMin	185400
20	2	8166	331.61	LocalMin	163320
20	3	8517	398.57	LocalMin	170340
20	4	9426	280.95	LocalMin	188520
20	5	7632	273.13	LocalMin	152640
20	6	9101	300.83	LocalMin	182020
20	7	8197	483.55	LocalMin	163940
20	8	8310	311.78	LocalMin	166200
20	9	8120	379.51	LocalMin	162400
20	10	8681	257.84	LocalMin	173620
30	1	10001	251.79	ExceedMaxIter	300030
30	2	7910	256.92	LocalMin	237300
30	3	8197	342.69	LocalMin	245910
30	4	8779	275.96	LocalMin	263370
30	5	10001	234.03	ExceedMaxIter	300030
30	6	9756	407.58	LocalMin	292680
30	7	10001	362.52	ExceedMaxIter	300030
30	8	8756	291.81	LocalMin	262680
30	9	10001	403.74	ExceedMaxIter	300030
30	10	10001	345.75	ExceedMaxIter	300030
40	1	10001	366.92	ExceedMaxIter	400040
40	2	7921	312.78	LocalMin	316840
40	3	8360	346.88	LocalMin	334400

40	4	10001	379.79	ExceedMaxIter	400040
40	5	10001	216.19	ExceedMaxIter	400040
40	6	10001	208.16	ExceedMaxIter	400040
40	7	10001	251.87	ExceedMaxIter	400040
40	8	10001	195.08	ExceedMaxIter	400040
40	9	10001	195.22	ExceedMaxIter	400040
40	10	10001	327.77	ExceedMaxIter	400040
50	1	10001	320.94	ExceedMaxIter	500050
50	2	10001	246.82	ExceedMaxIter	500050
50	3	10001	312.98	ExceedMaxIter	500050
50	4	10001	374.88	ExceedMaxIter	500050
50	5	8879	253.22	LocalMin	443950
50	6	9385	356.83	LocalMin	469250
50	7	8993	267.07	LocalMin	449650
50	8	10001	254.92	ExceedMaxIter	500050
50	9	9665	276.1	LocalMin	483250
50	10	10001	319.88	ExceedMaxIter	500050

Appendix II: Optimal Control Diameters

Example 1: A Simple 3D Cantilever Beam Example (5x1x2, degree 4x0x1)

w=0	w=1
1.6640	1.3411
0.4684	1.0490
0.8740	1.5658
1.1074	0.5614
0.2540	0.9491

Example 2: A more complicated 3D Cantilever Beam Example (4x2x3, degree 3x1x2)

1. Original diameter calculation:

w=0		w=0.5		w=1	
0.0845	3.4541	-1.0447	0.9096	3.7294	-0.0728
-0.7369	2.8787	-1.2204	0.9678	3.3957	-0.1407
-0.2578	2.0806	-0.1795	1.4435	1.9967	0.0744
-0.3481	2.3608	2.1663	-0.6106	2.8339	-0.9612

2. Alternative diameter calculation:

   a. Run 1: volume weight = 1

w=0		w=0.5		w=1	
0.1831	4.6340	-4.0117	-3.0061	4.4944	-0.1225
-3.1168	0.9512	-0.5577	-1.3784	-0.5614	-2.8594
-4.9677	4.2698	-1.2034	-2.5006	6.7366	-1.7476
3.2625	0.4588	-2.3940	-1.0318	-2.7968	-0.5719

   b. Run 2: volume weight = 100

w=0		w=0.5		w=1	
-0.1875	4.0797	-4.2605	-3.3274	4.2988	0.3104
-1.3268	1.5314	-0.6325	-1.0407	-0.5441	-3.0206
-6.1191	3.2469	-0.6978	-2.2034	6.1091	-1.4460
3.1454	0.8861	-2.1392	-1.3002	-2.5693	-0.4623

Example 3: A 2D Beam Example (8x4, degree 7x3)

1. Volume weight = 1E12

0.8739	0.8659	0.8067	0.7583
0.8249	0.8205	0.8012	0.8301
0.8075	0.8049	0.8044	0.9458
0.8128	0.8106	0.8106	1.0427
0.8140	0.8118	0.8116	1.0593
0.8056	0.8032	0.8058	0.9826
0.8025	0.7995	0.7985	0.8643
0.8905	0.8803	0.8076	0.7518

2. Volume weight = 1E6

0.9882	1.0111	0.9052	0.8768
0.9137	0.9483	0.9110	0.9703
0.8827	0.9213	0.9184	1.0179
0.8908	0.9306	0.9302	1.0040
0.8928	0.9326	0.9322	0.9948
0.8789	0.9180	0.9213	1.0125
0.8739	0.9106	0.9056	0.9892
1.0078	1.0385	0.9272	0.9041

Example 4: Wing Example

1. 2D Airfoil

    a. 3x2 (degree 2x1)

0.4961	11.1736
-11.2834	9.3639
-10.5202	12.6484

    b. 4x2 (degree 3x1)

7.5737	14.6379
-12.5459	5.9587
-10.6598	13.8882
-8.2662	9.9924

2. 3D Wing (4x3x2, degree 3x2x1)

w=0			w=1		
-3.1977	19.5948	-11.0302	-8.1426	-0.2802	-36.2320
-32.7911	16.9532	10.9386	-9.3527	46.2800	4.8441
-13.6197	24.7190	4.7716	-38.0863	20.6545	4.2578
-14.7758	13.9220	12.2821	-23.0327	18.4039	30.1703

References:

1. Sherman, L.M., *Additive Manufacturing — New Capabilities for Rapid Prototypes And Production Parts.* 2009, Plastics Technology
2. Giannatsis, J. and V. Dedoussis, *Additive fabrication technologies applied to medicine and health care: a review* The International Journal of Advanced Manufacturing Technology, 2009. **40**(1-2): p. 116-127.
3. eXpedio. *http://lisoong.trustpass.alibaba.com/*.
4. Graf, G., *Development of Specialized Base Primitives for Meso-Scale Conforming Truss Structures*, in *Mechanical Engineering*. 2009, Georgia Institute of Technology: Atlanta, GA.
5. Bak, D., *Rapid prototyping or rapid production? 3D printing processes move industry towards the latter.* Assembly Automation, 2003. **23**(4): p. 340-345.
6. Gibson, L.J. and M.F. Ashby, *Cellular Solids: Structure and Properties*. 1997, Cambridge, UK: Cambridge University Press.
7. Hayes, A.M. and A. Wang, *Mechanics of Linear Cellular Alloys.* Mechanics of materials, 2004. **36**: p. p. 691-713.
8. Chu, C., G. Graf, and D.W. Rosen, *Design for Additive Manufacturing of Cellular Structures*, in *Computer-Aided Design and Applications*. 2008.
9. Chu, C., et al., *A comparison of synthesis methods for cellular structures with application to additive manufacturing.* Submitted to Rapid Prototyping Journal, 2008.
10. Wang, H.V. and D.W. Rosen. *An Automated Design Synthesis Method for Compliant Mechanisms with Application to Morphing Wings*. in *ASME Mechanisms and Robotics Conference*. 2006. Philadelphia, PA.
11. Evans, A.G., N.A. J. W. Hutchinson, and M.F. Ashby, *The Topological Design of Multifunctional Cellular Matels.* Progress in Materials Science, 2001. **46**(3-4): p. 309-327.
12. Sypeck, D.J. and H.N.G. Wadley, *Cellular Metals Truss Core Sandwich Structures.* Journal of Materials Research, 2001. **16**(3): p. 890-897.
13. Deshpande, V.S., N.A. Fleck, and M.F. Ashby, *Effective properties of the octet-truss lattice material.* Journal of the Mechanics and Physics of Solids, 2001. **49**(8): p. 1747-1769.
14. Ashby, M.F., et al., *Metal Foams: A Design Guide*. 2000, Woburn, MA: Butterworth-Heinemann.
15. Johnston, S.R., et al., *Analysis of Mesostructure Unit Cells Comprised of Octet-truss Structures*, in *Solid Freeform Fabrication*. 2006. p. 421-423.

16. Wang, A. and D.L. McDowell, *Yield surfaces of various periodic metal honeycombs at intermediate relative density.* International Journal of Plasticity, 2005. **21**(2): p. 285-320.

17. Wang, H.V., *A unit cell approach for lightweight structure and compliant mechanism*, in *School of Mechanical Engineering*. 2005, Georgia Institute of Technology: Atlanta, GA.

18. Y.B., F. and O. R.W., *Nonlinear Elasticity: Theory and Applications*. 2001: Cambridge University Press.

19. Kota, S., et al., *Design of Compliant Mechanisms: Applications to MEMS.* Analog Integrated Circuits and Signal Processing, 2001. **29**(1-2): p. 7-15.

20. Lu, K.J. and S. Kota, *Design of Compliant Mechanisms for Morphing Structural Shapes.* Journal of Intelligent Material Systems and Structures, 2003. **14**(6): p. 379-391.

21. Lu, K.J. and S. Kota, *Compliant Mechanism Synthesis for Shape-Change Applications: Preliminary Results.*

22. Ananthasuresh, G.K., *A New Design Paradigm for Micro-Electro-Mechanical Systems and Investigations on the Compliant Mechanism Synthesis*. 1994, University of Michigan: Ann Arbor, MI.

23. Ananthasuresh, G.K., S. Kota, and Y. Gianchandani. *Systematic Synthesis of Microcompliant Mechanisms-Preliminary Results*. in *Third National Conference on Applied Mechanisms and Robotics*. 1993. Cincinnati, Ohio.

24. Sigmund, O. *Some Inverse Problems in Topology Design of Materials and Mechanisms*. in *IUTAM Symposium on Optimization of Mechanical Systems*. 1995. Stuttgart, Germany.

25. Larsen, U.D., O. Sigmund, and S. Bouwstra. *Design and Fabrication of Compliant Micromechanisms and Structures with Negative Poisson's Ratio*. in *IEEE Ninth Annual International Workshop on Micro Electro Mechanical Systems, An Investigation of Micro Sturctures, Sensors, Actuators, Machines and Systems*. 1996. San Diego, CA.

26. Frecker, M.I., et al., *Topological Syntesis of Compliant Mechanisms Using Multi-Criteria Optimization.* Journal of Mechanical Design, 1997. **119**(2): p. 238-245.

27. Kirsch, U., *Structural Optimization: Fundamentals and Applications*. 1993: Springer-Verlag Berlin and Heidelberg GmbH & Co. K.

28. Rozvany, G.I.N., *Topology Optimization in Structural Mechanics*. 2003: Springer.

29. Tai, K., G.Y. Cui, and R. Tapabrata, *Design Synthesis of Path Generating Compliant Mechanisms by Evolutionary Optimization of Topology and Shape.* ASME J. Mechanical Design, 2002. **124**: p. 492-500.

30. Joo, J., S. Kota, and N. Kikuchi, *Nonlinear Synthesis of Compliant Mechanisms: Topology Design*, in *ASME Design Engineering Technical Conferences*. 2001.

31. Hassani, B. and E. Hinton, *Homogenization and structural topology optimization: theory, practice and software*. 1999: Springer.

32. Yang, R.J. and C.H. Chuang, *Optimal Topology Design Using Linear Programming.* Computers & Structures, 1994. **52**(2): p. 265-275.

33. Burns, S.A., *Recent Advances in Optimal Structural Design*. 2002: American Society of Civil Engineers.

34. Fourie, P.C. and A.A. Groenwold, *The Particle Swarm Optimization Algorithm in Size and Shape Optimization.* Structural and Multidisciplinary Optimization, 2002. **23**: p. 259-267.

35. Saxena, A. and G.K. Ananthasuresh. *Towards the design of compliant continuum topologies with geometric nonlinearity*. in *ASME Design Engineering Technical Conferences*. 1999. Las Vegas, Nevada, USA.

36. Pedersen, C.B.W., T. Buhl, and O. Sigmund, *Topology synthesis of large-displacement compliant mechanisms.* International Journal for Numerical Methods in Engineering, 2001. **50**(12): p. 2683 - 2705.

37. Reddy, J.N., *An Introduction to The Finite Element Method*. 2005, Tata McGraw-Hill Publishing Company Limited.

38. Engelbrecht, S., ***DESIGN OF MESO_SCALE CELLULAR STRUCTURE FOR RAPID MANUFACTURING***, in *Mechanical Engineering*. 2009, Georgia Institute of Tehcnology: Atlanta.

39. Seepersad, C.C., et al., *Robust Design of Cellular Materials with Topological and Dimensional Imperfections. ASME J. Mechanical Design*, 2006. **128**: p. 1285.

40. Rosen, D.W. and B. Sager, *Use of Parameter Estimation For Stereolithography Surface Finish Improvement. Rapid Prototyping Journal*, 2008. **14**(4): p. 213-220.

41. Graf, G., et al., *Synthesis Methods for Lightweight Lattice Structures* in *ASME CIE Conference*. 2009.

42. Smith, A.E. and D.W. Coit, *Penalty Functions*, in *Handbook of Evolutionary Computation*, T. Baeck, D. Fogel, and Z. Michalewicz, Editors. 1995, Oxford University Press and Institute of Physics Publishing.

43. Zeid, I., *Mastering CAD/CAM*. 2005, McGraw-Hill. p. 268-271.

44. Zheng, Y.-L., et al. *On the convergence analysis and parameter selection in particle swarm optimization*. in *Second International Conference on Machine Learning and Cybernetics*. 2003.

45. Zhang, L.-p., Y. Huan-jun, and H. Shang-zu, *Optimal Choice of Parameters for Particle Swarm Optimization.* Journal of Zhejiang University SCIENCE, 2004.

46. Zhang, L.-p.Y.H.-j. and H. Shang-zu, *Optimal Choice of Parameters for Particle Swarm Optimization.* Journal of Zhejiang University SCIENCE, 2004.

47. Trelea, I.C., *The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection.* Information Processing Letters, 2003. **85**(6): p. 317-325

48. El-Gallad, A., et al., *Enhancing the particle swarm optimizer via proper parameters selection.* Electrical and Computer Engineering, 2002. **2**: p. 792-797.

49. Carlisle, A. and G. Dozier. *An Off-The-Shelf PSO*. in *the Particle Swarm Optimization Workshop*. 2001.

50. MATLAB, *Optimization Toolbox Help Document.*

51. Xia, Q. and M.Y. Wang, *An optimization-based approach for the design of lightweight truss-like structures.* Submitted to the Journal of Computer-Aided Design, 2008.

52. Adrian, T.C.H., *DESIGN AND PERFORMANCE EVALUATION STUDY OF A PROTOTYPE OF A TACTICAL UNMANNED AERIAL VEHICLE* 2007, NAVAL POSTGRADUATE SCHOOL.