

MULTILEVEL OPTIMIZATION OF LATTICE STRUCTURES OF STRUCTURES

By

Arric Ervin McLauchlan

A THESIS

Submitted to
Michigan State University
In Partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Master of Science

2018

ABSTRACT

MULTILEVEL OPTIMIZATION OF LATTICE STRUCTURES OF STRUCTURES

By

Arric Ervin McLauchlan

Recent advancements in additive manufacturing make it possible to design structures with complex and spatially varying microstructures, sometimes referred to as Structures of Structures (SoS). The detailed description of these systems can easily exceed available memory capacity, making it difficult to perform modeling, analysis and optimization using existing approaches. A multilevel strategy can be employed to reduce the required memory and computational expense of the analysis and design process.

In the current work, a multilevel optimization technique is developed that attempts to simplify the optimization of SoS. The target application is a lattice in which each system level lattice member is itself composed of a lattice microstructure. A generative geometry representation is used based on the research code "LatticeMaker," and various multi-level optimization representations are explored for structural and structural-thermal optimization problems.

ACKNOWLEDGMENT

First and foremost, I would like to thank my advisor Dr. Ron Averill for his advice and support throughout my graduate career. This project would not have been possible without his patient guidance. He has shaped my appreciation for a well-rounded education and training (both scientific and otherwise). I would also like to thank my thesis committee members Dr. Kalyanmoy Deb and Dr. Alejandro Diaz for their advice and contributions to this project.

In addition, I would like to thank my family for their continued support throughout my master's degree. My parents were very supportive of my academic ventures and for that I thank them. My father was a big part of why I entered the field of engineering and I could not have done it without him. My sister has always encouraged me to be ambitious and never settle on my dreams, and without her, I never would have had the courage to apply to graduate school. I'd like to thank my grandma for always believing I could do no wrong, and helping me out along the way. I'd like to thank my mother, who worked so hard to make sure I had the means to do whatever my attention turned to. And lastly, I would like to thank my wife for her unwavering support, patience, understanding, and help during this time in my life.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or of the U.S. Government.

TABLE OF CONTENTS

| | |
|---|-----|
| LIST OF TABLES | vi |
| LIST OF FIGURES | vii |
| CHAPTER 1: Introduction | 2 |
| 1.1 Background..... | 2 |
| 1.1.1 Additive Manufacturing..... | 4 |
| 1.1.2 Ultra High Definition Structures of Structures..... | 8 |
| 1.2 Literature Review of Additive Manufacturing | 12 |
| 1.3 Literature Review of Multilevel Optimization..... | 15 |
| 1.4 Objective of this Work..... | 21 |
| 1.5 Organization of the Thesis..... | 22 |
| CHAPTER 2: Optimization Details and Analysis Tools..... | 24 |
| 2.1 HEEDS Multidisciplinary Design Optimization..... | 24 |
| 2.2 Analysis Model and Optimization Process..... | 27 |
| 2.3 Analysis Model Validation | 32 |
| CHAPTER 3: Two-Step Serial Optimization..... | 37 |
| 3.1 Three-Member Structure Problem Statement | 37 |
| 3.2 Two-Step Serial Optimization Process Flow..... | 41 |
| 3.2.1 Two-Step Serial Optimization - Phase 1..... | 43 |
| 3.2.2 Two-Step Serial Optimization - Phase 2..... | 44 |
| 3.3 Problem Statement Decomposition for Two-Step Serial Optimization | 45 |
| 3.3.1 “Individual Only” Subsystem (Micro) Level Constraining Method | 46 |
| 3.3.2 “Magnitude Only” Micro Level Constraining Method | 47 |
| 3.3.3 “Combined” Micro Level Constraining Method..... | 48 |
| 3.3.4 Two-Step Serial Optimization Process Validation | 49 |
| 3.4 Three Member Structure Problem – Analysis and Results | 50 |
| CHAPTER 4: Multilevel Optimization..... | 54 |
| 4.1 Multilevel Optimization Process Flow..... | 54 |
| 4.2 Problem Statement for Multilevel Optimization | 57 |
| 4.2.1 Problem Statement Decomposition | 58 |
| 4.2.2 Reaction Constraint Methods | 59 |
| 4.3 Multilevel Optimization Results and Validation..... | 61 |
| CHAPTER 5: Optimization of a Transmission Tower | 63 |
| 5.1 Transmission Tower Problem Statement | 64 |
| 5.2 Multilevel Optimization Process Flow for Multiphysics Problems..... | 65 |
| 5.3 Multiphysics Problem Statement Decomposition for the Multilevel Optimization Method | 66 |
| 5.4 Analysis and Results | 68 |
| 5.5 Multilevel Optimization with Adaptive Tolerance Control..... | 72 |
| 5.6 Multilevel Optimization with Adaptive Constraint Scaling..... | 76 |
| CHAPTER 6: Analysis of Results and Drawn Conclusions | 82 |

| | | |
|--------------------|--|----|
| 6.1 | Conclusions..... | 82 |
| 6.1.1 | Two-Step Serial Optimization | 82 |
| 6.1.2 | Multilevel Optimization | 83 |
| 6.1.3 | Multilevel Optimization of a multi-physics problem | 84 |
| 6.2 | Future Work..... | 86 |
| 6.2.1 | Validation and Verification of the Modified ATC Equation | 86 |
| 6.2.2 | Average Mises Stress instead of Maximum Mises Stress | 86 |
| 6.2.3 | Sensitivity and Robustness..... | 87 |
| 6.2.4 | Homogenization in the Global Model..... | 87 |
| 6.2.5 | Handling Global Level Infeasibilities | 88 |
| 6.2.6 | Automation of the Multilevel Processes..... | 88 |
| 6.2.7 | Three Level Optimization Process..... | 88 |
| BIBLIOGRAPHY | | 90 |

LIST OF TABLES

| | |
|--|----|
| Table 1. LatticeMaker input file parameters, meaning, and type | 28 |
| Table 2. Lattice design variables as defined in HEEDS | 29 |
| Table 3. Validation Test problem parameters, definitions, and values | 33 |
| Table 4. Validation Test problem reaction forces and moments | 33 |
| Table 5. Validation Test's beam's cross-sectional properties | 34 |
| Table 6. Validation Test problem results and percent error relative to the analytical solution for the primary mode of failure | 36 |
| Table 7. Three-Member Structure problem parameters, definitions, and values | 38 |
| Table 8. Numerical tabulation of the Three Member Structure Two-Step Serial Optimization trial results | 51 |
| Table 9. Various material properties, environmental conditions, and load cases used in the Transmission Tower analysis | 64 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1. An example of (a) A SoS three member lattice in which each member is itself a lattice structure, (b) A homogenized system level model in which each lattice structure is replaced with an equivalent beam member, and (c) A subsystem level model in which the detailed microstructure of a single system level lattice member is represented explicitly. | 10 |
| Figure 2. Beers and Vanderplaats' [20] Multilevel Optimization flow diagram. | 17 |
| Figure 3. Diagram of the Validation Test problem used to verify an accurate analysis model is being used. | 33 |
| Figure 4. Validation Test's internal shear and moment diagrams. | 34 |
| Figure 5. Diagram of simple Three Member Structure problem that will be analyzed using a Two-Step Serial optimization process..... | 38 |
| Figure 6. Three-Member Structure geometric parameter definitions used to set up the optimization analysis model | 39 |
| Figure 7. Structure of Structures representation of the Three-Member Structure problem..... | 40 |
| Figure 8. Two-Step Serial Optimization process flow diagram | 42 |
| Figure 9. Homogenized representation of the global (system) level model. | 45 |
| Figure 10. Micro (subsystem) level Finite Element model of a single member of the Three Member Structure problem..... | 46 |
| Figure 11. Abaqus rendering of the optimized model using the "Individual Only" technique..... | 51 |
| Figure 12. Abaqus rendering of the optimized model using the "Magnitude Only" technique..... | 52 |
| Figure 13. Abaqus rendering of the optimized model using the "Combined" technique | 52 |
| Figure 14 . Comparison of the optimal solution using the "Individual Only", the "Magnitude Only", and the "Combined Constraints" techniques to the Pareto Front which was generated from an All-At-Once optimization study. | 53 |
| Figure 15. Multilevel Optimization Process Flow Diagram..... | 55 |
| Figure 16. Multilevel Optimization – Phase 1 Flow Diagram | 56 |
| Figure 17. Multilevel Optimization – Phase 2 Flow Diagram | 57 |
| Figure 18. Comparison of the "Individual Constraints" and "Combined Constraints" multilevel optimization techniques to the Pareto Front found from an All-At-Once optimization study..... | 62 |
| Figure 19. A diagram of the Transmission Tower model showing the convection region, the location of point loads and fluxes, and Region B (the SoS region)..... | 63 |

| | |
|---|----|
| Figure 20. Relationship between Volume and Stress for the unconstrained Transmission Tower problem. | 68 |
| Figure 21. Relationship between Iteration and Normalized Volume for the unconstrained Transmission Tower problem..... | 69 |
| Figure 22. Relationship between Iteration and Normalized Temperature for the unconstrained Transmission Tower Problem..... | 70 |
| Figure 23. Relationship between Volume and Stress for the constrained Transmission Tower Problem. | 70 |
| Figure 24. Relationship between Iteration and Stress for the constrained Transmission Tower Problem. | 71 |
| Figure 25. Relationship between Volume and Stress for the unconstrained Transmission Tower problem using Adaptive Tolerance Control | 75 |
| Figure 26. Relationship between Volume and Stress for the constrained Transmission Tower problem using Adaptive Tolerance Control. | 76 |
| Figure 27. The effects of Constraint Scaling as shown by the relationship between Volume and Stress for the constrained Transmission Tower Problem..... | 78 |
| Figure 28. The effects of Constraint Scaling with Adaptive Tolerance Control as shown by the relationship between Volume and Stress for the constrained Transmission Tower Problem. | 78 |
| Figure 29. The relationship between the “Degree of Feasibility” and the tolerance control factor, β , in the Modified ATC equation..... | 79 |
| Figure 30. Results of the Transmission Tower Problem using the Modified ATC equation and Constraint Scaling that show the relationship that Stress and Temperature have with Volume. Note that the results are not monotonic and cannot be followed from right to left as with most other figures in this work..... | 81 |
| Figure 31. Results of the Transmission Tower Problem using the Modified ATC equation and Constraint Scaling that show the progression of the study. This proves the results for this study are not monotonic..... | 81 |

CHAPTER 1: Introduction

The ability to quickly and efficiently manufacture quality products with minimal waste is of crucial importance in today's manufacturing industries. Additive Manufacturing (AM) enables these ideals of minimal waste and processing time. As a result, it has gained a considerable amount of intrigue from the manufacturing industry.

Advances in AM technologies are allowing the production of higher quality products with ever more complex designs, which has significantly aided its recent breakthrough as a serious contender into the manufacturing world. However, improvement in the technology itself is only the first step; soon AM technologies may surpass the capabilities of standard analysis and design techniques.

1.1 Background

Additive Manufacturing processes are Computer Numerical Controlled (CNC) processes that produce products directly from CAD files without needing the creation of detailed specifications. Because no detailed prints are required, designs that have freeform curvatures or are so complex that detailing is infeasible or impossible are able to be manufactured. Before a part is manufactured, however, it should be analyzed and optimized. This is usually performed using various engineering software packages.

There is a direct correlation between the complexity of a part and its file size. As a result, computational resources can quickly become exhausted. In order to facilitate AM's continued growth, new design and analysis methodologies must be developed that will accommodate the continued advancement of additive manufacturing processes while utilizing only the computational resources currently available.

The nature of Additive Manufacturing is such that it allows the placement of material in specific locations. One of the advantages that this provides over traditional subtractive manufacturing methods is that it unlocks the capability to control the microstructure of a design as well as the global level features. It is challenging to represent microstructure explicitly in a CAD model, in part because the CAD model size would become too large to manage. CAM software packages are available that allow some control over the microstructure prior to manufacturing, but this is usually defined in machine code, and is not geometrically represented. This means that any engineering analysis that is performed must utilize homogenized material properties to represent the microstructure. Due to the bonding variability that is inevitably present in all additively manufactured parts, accurate modeling through the use of homogenized materials can be very difficult, even without considering a part's microstructure. Adding the microstructure into the equation would further complicate the model and reduce the reliability of the analysis.

An alternative to using homogenized materials in engineering analyses is the manual modeling of the microstructure during the CAD phase of the design process. The issue with this is that it would be highly tedious and time-consuming. If parametric CAD modeling is used (as often required for optimization), the files size would be huge, causing any form of analysis to be extremely expensive. Furthermore, the number of variables and parameters would be very large. For most optimization algorithms, the number of evaluations required to find good designs is proportional to the number of variables. This renders optimization of all but the simplest of parts to be infeasible.

A Multilevel Optimization process could assist in reducing the computational expense of analyzing and synthesizing additively manufactured parts while maintaining the reliability of the simulation. Expanding on the work of Halepatali [1], a multilevel approach could be applied that models the global structure and its subsystems (or the microstructure) separately. The levels would be analyzed separately and would communicate through the application of boundary conditions at

subsystem interfaces. In this way, the file sizes would be smaller and the number of variables reduced allowing for an efficient optimization process to be feasible and facilitating the continued advancement of Additive Manufacturing.

1.1.1 Additive Manufacturing

The development of Additive Manufacturing processes began in 1987 with the invention of stereolithography. This was the precursor to a process called Vat Photopolymerization and was used simply as a way to decrease the cost of prototyping and aid in the creative phase of product design; thus, it was originally designated by the term Rapid Prototyping (RP) [2]. The ability to produce cheap prototypes greatly reduced the cost of prototyping (when compared to previous methods) as well as the time required to produce the prototype. This greatly reduced the cost of an iterative design process and allowed for a quicker design phase.

While the final product still required an expensive manufacturing process, the price at which the producer needed to sell the product in order to cover the development expenses was reduced. As the industry appeal of RP increased, more methods of rapid prototyping were developed and the current methods improved. The improvement of the technology allowed for higher quality outputs. Today, many final products can be directly produced from these processes, rendering the name “Rapid Prototyping” inadequate as it does not cover the entire range of possible uses.

The term Rapid Prototyping is still used in prototyping applications, however, the term Additive Manufacturing covers a wider spectrum of applications. Imagine the potential cost savings of a process that can produce high quality, high strength production-level parts with complex geometries and minimal waste material. This would revolutionize the manufacturing industry.

The ability of an AM process to produce production-level parts depends greatly on the part’s requirements (structural, electrical, surface quality, etc.). There are several forms of AM currently available. Powder Bed Fusion, Material Extrusion, Material Jetting, Binder Jetting, and Sheet

Lamination are just a few of the ever expanding arsenal of techniques that fall under the Additive Manufacturing umbrella [2]. Different processes have qualities that are better suited for certain applications, and the technology chosen needs to be based on the requirements of the product [2]. Today, Material Extrusion processes such as Fused Deposition Modeling (FDM) are the most commonly used AM processes. In Material Extrusion processes, a pressurized extrusion head extrudes a semi-liquefied material from a heated extrusion nozzle and deposits it on a build platform. The extrusion head follows a tool path that places the extruded material according to the part's cross-sectional geometry. Once a full layer is deposited, the build platform lowers, and the next layer is deposited on top of the previous layer [2].

Different AM processes are capable of utilizing different sets of materials. Until recently, only a few select materials could be used in each process, so material properties were an important consideration when choosing a manufacturing method for a given part. Now, there are several plastics that can be 3D printed, as well as some metals, ceramics, composites, and even biocompatible materials. There has also been some interest in the development of an AM process that can print multiple materials at once [3]. Material choice is still an important consideration, but there are many more options to choose from.

Every new addition to the family of additively manufacturable materials opens the door to new applications. AM has found its way into many industries ranging from Aerospace to Architecture, and from Biomedical Engineering to the Arts. NASA is currently utilizing Additive Manufacturing to develop tools and instruments for use in current and future space missions. They have even begun to look into the possibility of 3D printing food for use on long duration space missions [4]. The construction industry has adapted traditional AM process to extrude concrete. This allows a concrete structure with complex curvatures to be quickly produced without the use of intricate molds [5]. The advent of biocompatible printing has had a profound impact on the medical community. AM

processes can be used to create patient-specific prosthetics, hearing aids, and other medical devices. It can even be utilized to produce microscale scaffolds that can be used to aid the regeneration of bone and other tissues [6,7].

Improvements in the scale of AM processes are also of strong interest. The term “scale” as used in this context can refer to a few different concepts. It can refer to the production quantity of a process. This relates to the output volume of AM processes, as well as their speed and efficiency. Alternatively, it can also refer to the ability of a process to produce parts of extreme sizes: making products with very small features (microscale) or very large features (construction scale parts). The ability to combine these two scales in a single manufacturing process has unlocked an important capability of AM processes; the ability to control the microstructure of a product will expedite the transformation of AM from a prototyping method to a production quality manufacturing technique.

The “quality” of a product can refer to any number of measures, and is very much application specific. It can refer to surface quality, bond quality, or the reaction of a part under a given load. For example, a part that is designed for use in an aerodynamic analysis will likely require a very fine surface resolution, whereas an additively manufactured bracket will likely need to be very strong, stiff, and able to withstand a higher degree of stress. To produce a part with desirable qualities the processing parameters need to be carefully chosen. In Fused Deposition Modeling, parameters such as infill pattern, nozzle diameter, layer height, deposition speed, build platform temperature, and nozzle temperature are very important. Every AM process is different, and each one will have its own slightly different set of processing parameters that will affect the quality of the part produced. Despite the differences between the processes, most AM techniques will require a layer height and an infill pattern to be defined.

At this time, the steps involved in most AM processes is the same, and most follow relatively similar preprocessing steps. A three dimensional model is developed using CAD software. It is then

sliced to form cross-sectional layers. These layers are sent to the AM machine, where it adds material according to the layer geometry. The resolution of the model slicing is what controls the parameter known as the layer height. Layer height is a key parameter because it is directly related to the surface resolution of the product. A smaller layer height will produce a part with finer resolution and a smoother surface. On the other hand, it will greatly increase the processing time, and it may increase the amount of material used. For extrusion based processes such as FDM, a smaller layer height has also been shown to have degraded bending strength [8] but improved strength when subjected to axial loads [9,10]. Adaptive Slicing (AS) algorithms exist that adjust the layer height according to part geometry: sections with high curvature are given a smaller layer height to reduce the stair-stepping effect whereas areas that are relatively vertical can be given larger layer heights to reduce the processing time [8].

Rather than using the typical flat slices, a Curved Layer (CL) slicing algorithm can be utilized [8]. These algorithms will eliminate the stair-step effect entirely and, for certain AM processes, it may also present improved mechanical properties due to the continuity of the deposited material (known as “roads”). The drawback of these algorithms is that the infill pattern is generally restricted. Recently, new slicing algorithms have been developed that incorporate both Curved Layers and Adaptive Slicing (CLAS) [8]. These CLAS algorithms are able to capture the fine details of the CL algorithms while the Adaptive Slicing allows for the tool path to be less restricted. Freedom of the toolpath allows for the creation of more complex features and geometry-specific infill patterns, which can improve the bonding between layers.

Related to layer slicing, the orientation of the model during the building process can have significant effects on the mechanical properties of a part. For most extrusion based AM processes, delamination at the layer interface is the predominant mode of failure, unless precautions are taken to avoid tensile stresses in the build direction [11-14]. This of course depends on several parameters

such as the material, the process, and the part geometry, but in general, avoiding stress in the build direction will improve the mechanical properties of additively manufactured parts.

The orientation of the raster angle (the relative angle between the deposited road and the X-Y coordinates of the build platform) has also been shown to have an effect on the mechanical properties of FDM parts. Espin et al [11] suggest that, not only is it better to have a raster angle that arranges roads in the direction of axial loading, but orienting the part such that the number of layers is maximized is also mechanically beneficial. Raster angle is just one of the many infill parameters that can be controlled. Road width, number of shell layers, and air gap can all be controlled to produce parts with different mechanical qualities [12]. In this way, the infill (including layer height) of an additively manufactured part can be thought of as the defining microstructure.

In many commercial FDM machines, an algorithm is used to automatically calculate the tool path for the user. For example, MakerBot [15] utilizes proprietary software called MakerWare [15] to prepare a model prior to building. This software is the environment in which the user controls the part orientation and many other build parameters, including simplified infill parameters. The user simply has to specify a layer height, the desired number of shells, and an infill percentage. With this information, the software automatically develops the tool path. This is advantageous because it opens the door to Additive Manufacturing for many people who may not know how to program machine code. The drawback is that the user has very little control over the actual tool path and thus has very little control over the microstructure. If an infill percentage greater than 0% and less than 100% is specified, MakerWare will program a hexagonal tool path, making the inside of the part look like a honeycomb.

1.1.2 Ultra High Definition Structures of Structures

A Structure of Structures (SoS) is any object that has a defined microscale structure. In this sense, any product made from composite materials can be considered a Structure of Structures. Even

structures made from homogenous materials can be considered Structures of Structures, when the components themselves are composed of smaller structures. An example of this is networks of steel beams that compose larger lattice structures, which build on each other to define global features of a tower. This idea of beams building lattices which build structures is the concept behind Ultra High Definition Structures of Structures (UHD-SoS).

To illustrate the importance of a design and analysis methodology catered to UHD-SoS, imagine a simple additively manufactured component that could be modeled using a single primitive shape. Now imagine that MakerBot's honeycomb infill pattern was utilized, and say that the infill can be divided into 300 unit cells with no overlap, where each cell can be modeled using a few smaller primitive shapes – say 3. That single primitive component is now designed with a total of 900 primitive shapes. This is still feasible to design and has already been accomplished, but additive manufacturing makes it possible to take it step further: now imagine that each primitive is composed of only 50 yet smaller primitives. Now, this single primitive component is composed of 45,000 primitive shapes. With the ongoing effort to produce larger machines having finer resolutions, it is not a stretch to imagine the feasibility of producing a part whose design surpasses 10^5 primitive features - the largest number of primitives able to be modeled on most systems.

While UHD-SoS can be found in many different shapes, sizes, and dimensionalities, the scope of this work focuses on lattice-type global structures that can be modeled using one dimensional beam elements or, alternatively, annular cylindrical lattice structures that are composed of one dimensional beam elements. The global structure can, however, be one, two, or three dimensional. Because of all the elements and structures at different levels, it can be difficult to have a discussion about Structures of Structures of this type without utilizing a strict nomenclature for clarity. The next few paragraphs will outline the terminology and nomenclature utilized throughout this work. Unless otherwise mentioned, wherever these terms are referenced, the following definitions are implied.

As a major theme of this work is multilevel optimization, different analysis levels will often be discussed. In a two-level problem, we will have a macro (system) level and a micro (subsystem) level. The macro level model can be described by the members' coarse features and can either be homogenized so no fine details are included, or, it can be shown with full microstructure detail. The fully detailed model will sometimes be referred to as the "All-At-Once" (AAO) model, or the "Global" model. The micro level model's scope is of a single macro level member. Here the fine elemental details are included.

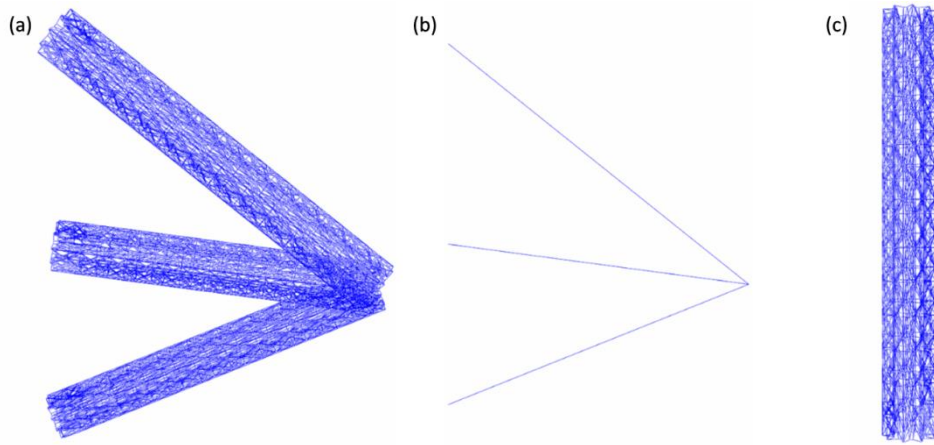


Figure 1. An example of (a) A SoS three member lattice in which each member is itself a lattice structure, (b) A homogenized system level model in which each lattice structure is replaced with an equivalent beam member, and (c) A subsystem level model in which the detailed microstructure of a single system level lattice member is represented explicitly.

In defining the different model levels, the term "member" was used. A "member" can be thought of as a system level element that makes no assumptions about its subsystem level structure. It will often be preceded by the term "lattice". A "lattice member" generally refers to a complete lattice structure (so basically the subsystem model) as viewed from the system level, or a homogenized system level member that will be replaced with a lattice structure in the global model. Its scope will always be of a single macro level structural unit, but its level of detail can depend on context.

When referring to models, the term "element" will always be reserved for the smallest structural unit present in the subsystem level model. According to this definition, the micro level

model and the global model both consist solely of elements. The elements in the global model that cannot be grouped into lattice structures may be referred to individually as “non-lattice members” or “global elements”, depending on context and/or preference. With exceptions, the system level structural units will always be referred to as members.

When discussing the optimization of Structures of Structures, several terms related to optimization naturally arise. In both general and multilevel optimization, an “evaluation” refers to the assessment of a single design using the predefined analysis model. In general optimization, a set of evaluations that are executed, either sequentially or in parallel, for the purpose of retrieving an optimal solution as defined by an optimization statement, is generally called a “study”. This term does not carry over to multilevel optimization. In multilevel optimization, this definition better defines the term “iteration”. A set of iterations that are performed sequentially for the purpose of retrieving an optimal solution as defined by an optimization statement describes a multilevel optimization “study”.

In the body of this work and most sections leading up to it, this terminology will be followed as closely as possible. There are, however, a few situations where exceptions will be made. In particular, the term “element” may have a meaning other than the one outlined above. For example, it is difficult to discuss a finite element model, analysis, or results without occasionally using the term “element” regardless of the context (scale). Furthermore, outside the context of models, the term “element” can still refer to a representative unit within a vector or tensor.

In section 1.3, several multilevel optimization schemes will be introduced. Developers of multilevel schemes tend to adopt their own terminology, and different sources will often have different terms for the same or similar concepts. In the interest of presenting an accurate literary analysis, the terminology utilized by the referenced source will be temporarily adopted to avoid any errors that stem from incorrect or incomplete translations of terms from the discussed method to the terminology presented in this work.

1.2 Literature Review of Additive Manufacturing

Over the last decade or so, Additive Manufacturing has gained acceptance in a wide variety of industries and has solved many manufacturing issues. As a prelude to a special edition of *Annals of Biomedical Engineering*, Zadpoor and Jos [7] discuss several applications of AM in the biomedical engineering community, ranging from surgical tools and drug delivery mechanisms, to patient-specific orthotics and implants for cellular regrowth and prosthetics. Despite the vast appeal of AM to many industries, there are still several challenges that need to be addressed before it can be used as a widespread production level manufacturing process. Zadpoor and Jos address this topic by outlining two areas of AM that require improvement before its potential in biomedical engineering can be fully utilized. They explain that the limited number of biomaterials severely limits the number of acceptable applications (within the field). They also explain that the microstructure of biomaterials must be optimized and that the definition of “optimal” is very much application specific. When designing a microstructure, a multidisciplinary analytical technique that simultaneously accounts for mechanical, physical, and biological properties is needed.

Accordingly, the difficulties arising from the inaccurate predictive methods for determining effective material properties produced by additive manufacturing machines presents a major roadblock for AM. It is very difficult to perform engineering analyses on AM components because their effective properties are highly dependent on their processing parameters and are usually coupled to several processing parameters at once. A significant amount of research has been conducted that attempts to either model the effective properties or determine the degree of coupling between processing parameters ([3,8-14,16])

Recognizing the inadequacy of finite element material databases to model anisotropic AM parts, Domingo-Espin et al [11] explored the effects of build orientation on the ultimate strength of Polycarbonate test specimens produced by Fused Deposition Modeling. Build orientation’s effect on

stiffness was also investigated and compared to a simulated finite element model. The team determined that anisotropic models should be used to simulate parts exceeding the elastic limit of the build material and that build orientation is important when yield strength is exceeded.

Rayegani and Onwubolu [9] also attempt to determine the effect of part orientation on the mechanical properties of AM parts, but they use an entirely different method. The method they present involves using Design of Experiments (DOE) to analyze the effects of part orientation, raster angle, raster width (road width), and air gap on the ultimate tensile strength of FDM printed Acrylonitrile Butadiene Styrene (ABS) test specimens. The data collected from these tests is then used to train a GMDH (group method of data handling) network. Once completed, this network can be used to predict the mechanical properties of a test specimen based on its processing parameters. Then, using Differential Evolution, the GMDH model can be used to find the optimal processing parameters that maximizes the ultimate tensile strength.

In another use of DOE, Mohamed, Masood and Bhowmik [10] study the effects of layer thickness, air gap, raster angle, build orientation, road width, and number of contours (or “shells” as they are commonly referred) on the dynamic stiffness and dimensional accuracy of a FDM printed ABS test specimens. Rather than using an Artificial Neural Network as Rayegani and Onwubolu (2014) had, Mohamed et al use the DOE results to develop a response surface according to IV-Optimal methodology. Using the response surface, relationships between the processing parameters and the dynamic stiffness and dimensional accuracy were shown. Their results suggest that parameter modifications that improve dynamic stiffness tend to degrade dimensional accuracy. They also presented a very interesting relationship between layer thickness and dynamic stiffness: according to their model, an increase in a specimen’s layer thickness can either increase or decrease its dynamic stiffness, depending on the current layer height. This relationship could possibly explain the discrepancies found in literature regarding the effects of layer thickness.

Many more methods and models exist for predicting the effects of and optimizing AM processing parameters for FDM as well as other AM processes. Models developed for different AM processes may be more or less accurate than the models for FDM presented here. The accuracy of a model will depend greatly on the processing parameters for that AM process, as well as the part's geometry and desired qualities. The large and ever-increasing arsenal of nontoxic, extrudable materials and the utilization of an inductive heating element rather than use of a laser (which requires submersion in toxic gases) help make FDM one of the most popular AM processes. Unfortunately, it is also one of the more sensitive processes to processing parameters and part geometry. As a result, any model developed or optimization performed on FDM specimens regarding processing parameters will likely be sensitive to processing parameter fluctuations and will generally be specifically tailored to a part's function and geometry.

That is not to say that optimization for FDM is futile. Prediction of material properties and optimization of processing parameters in test specimens has been shown to be very effective when used for a particular design. Mohamed et al [10] even showed very reasonable results with different confirmation test geometries when compared to default printer settings (noting that the geometries were similar enough to the original test geometry such that they permitted the same stiffness test to be performed). Optimization can also be used in AM in other, less predictive ways.

Because AM processes create products directly from CAD models, extremely obscure or free-form shapes can be feasibly manufactured. This renders non-parametric modeling practices such as Topology Optimization incredibly useful. Zegard and Paulino [17] give a great example of this. After presenting a brief background discussion on Topology Optimization (TO), specifically detailing the Solid Isotropic Material with Penalization (SIMP) method, they discuss variations of the SIMP method using density filters, and present some issues that can be encountered when used on three dimensional problems. Through the use of a three dimensional cantilevered beam example problem,

they show how a linear density filter can cause unnatural thinning of members to occur. They then present a fix; the rapid decay of a cubic filter keeps material in the vicinity of a joint centralized rather than spreading it, unlike a linear filter, which generates densities in the joint region adding stiffness to the area and allowing it to carry more load [17]. The optimal design was then additively manufactured using either FDM or Selective Laser Sintering. Zegard and Paulino go on to discuss topology optimization problems with different length scales and how improving AM processes to make it possible to produce larger parts would be very beneficial.

Zhu et al [18] similarly paired Additive Manufacturing with Topology Optimization, but in an entirely different way. Touching closer to the issue that Zadpoor and Jos bring up in their prelude ([7]) in regards to the need for a method to optimize the microstructure of AM materials, Zhu's team develops a novel two-scale technique that exploits a combination of TO, stochastic sampling, and multimaterial printing to produce a part with a continuum of carefully optimized anisotropic microstructures. The technique starts by building a database of multi-material microstructures using stochastic search and continuous optimization to find and then extend the range of the material space boundaries. A smooth gamut is then fitted to the point cloud of tested microstructures to create a continuous representation of the material properties space. The macro structure is then optimized using TO. Rather than determining a binary material distribution as most standard topology algorithms such as SIMP, Zhu's customized topology algorithm determines the optimal material properties from the continuous gamut for any given cell. Once the TO is complete, each cell is replaced by the microstructure in the microstructure database that most closely resembles the continuous properties.

1.3 Literature Review of Multilevel Optimization

Multilevel optimization in general has been around for quite a while. In 1987, Beers and Vanderplaats [19] presented a method for solving complex systems by using a first order Taylor-

series approximation to linearize the objective and constraint functions at the “system” level, which can be decomposed into hierarchical “subsystems”. Highlights of their method include the use of “move limits”, which bound the design variables to a region defined in Equation (1) at both the system and subsystem levels, where X_i is the i th term of the vector contain the design variables, X , the U and L superscripts denote the upper and lower bounds of the i th term, and ML is the move limit.

$$\begin{aligned} X_i^U &= (1 + ML)X_i \\ X_i^L &= (1 - ML)X_i \end{aligned} \tag{1}$$

In their method, the subsystems are not linearized, but rather solved in their true form, with the exception of the system level constraints, which are passed to the subsystems in linearized form. They also include an operation that allows subsystem constraints to be adjusted based on their feasibility: violated constraints become less violated and unviolated constraints get closer to the constraint boundary. Another feature of their method is that the subsystem objective may or may not be included in the statement, but the ultimate goal of the subsystem problem is to find a feasible solution. Accordingly, the objective should only be allowed to influence the optimization slightly in order to avoid objective domination of the subsystem.

The basic procedure of Beers’ and Vanderplaats’ (1987) method is rather simple. It involves first evaluating the true system level and then passing the constraint gradients to the subsystems. The subsystem variable bounds, objectives, and constraints are modified as necessary, and then optimized independently. The optimal subsystem design variables, constraints, and constraint gradients are passed back up to the system where they are used to reconstruct the system, which is then reanalyzed. The system and subsystem constraint gradients are calculated for the linear system, which is then optimized. The optimal system level solution is passed to the true system model, where

it is analyzed. This process repeats until convergence, as shown in Figure 2 below. If a convex design space is correctly assumed, the system will converge to a Kuhn-Tucker optimal.

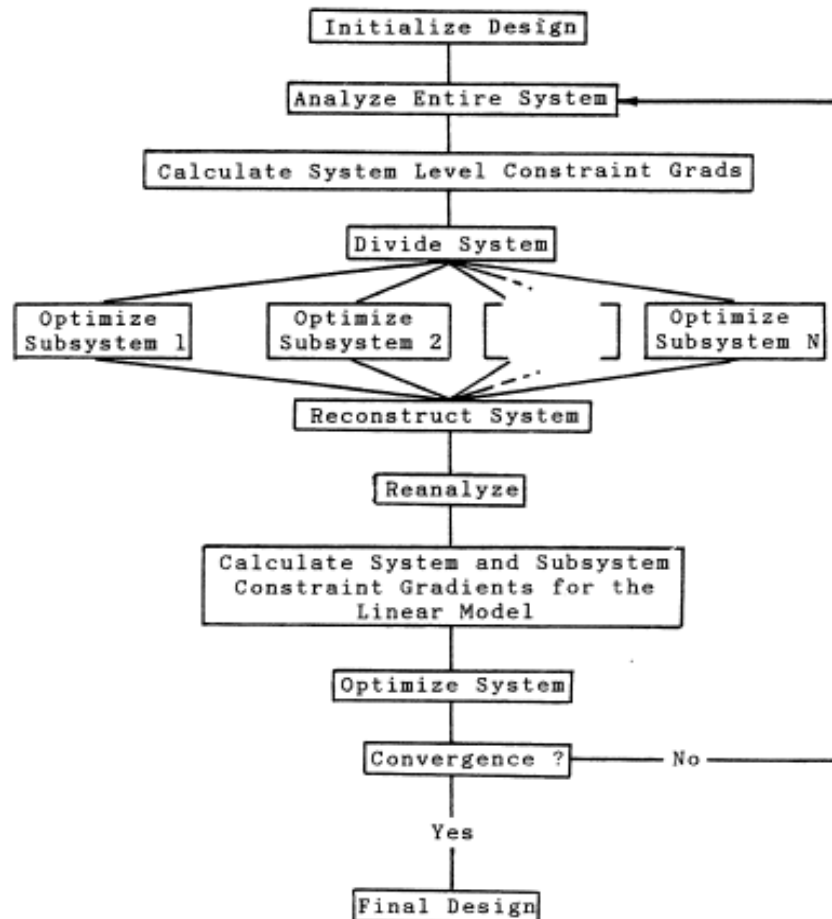


Figure 2. Beers and Vanderplaats' [19] Multilevel Optimization flow diagram.

To demonstrate this process, they provided an example “Portal Frame” test case, which consisted of three Aluminum alloy I-beam members connected to form a frame, which was subjected to an applied load, and whose deflection was constrained. Each I-beam had 6 shape parameters that could be independently modified, for a total of 18 independent design variables. They compare their multilevel scheme to a single level scheme and show that the multilevel scheme not only is valid, but converges (slightly) faster than the single level scheme.

Two years before Beer and Vanderplaats (1987) presented their linearized method, Sobieszczanski-Sobieski et al introduced a more generalized multilevel scheme [20]. Like Beers and Vanderplaats, Sobieszczanski-Sobieski's team decomposed a complex "structure" into a set of "substructures". Unlike Beers and Vanderplaats, Sobieszczanski-Sobieski's process took a step deeper and decomposed each substructure into its own set of substructures. The inclusion of the third level makes the logistics of data communication difficult, so Sobieszczanski-Sobieski adopted a naming scheme using indicial notation to refer to the specific substructures. To distinguish between two communicating (sub)structures, the "parent"- "daughter" relationship was used, where the "parent" is the higher-level (sub)structure and the "daughter" is the lower-level substructure.

Sobieszczanski-Sobieski's procedure starts at the most detailed level, however, the entire structure must be previously initialized from the top down, determining each level's stiffness and mass matrices, and passing them to all daughter substructures. Each substructure (with the exception of the most detailed) uses its parent's stiffness and mass matrices as constraints and uses its daughters stiffness and mass matrices as variables. The most detailed substructure's constraints follow the same scheme, but it has no daughter substructures, so the cross-sectional dimensions are used as design variables.

The objective of each substructure optimization is to minimize the amount of "Cumulative constraint" violation, which is defined in Equation (2), where ρ is a user-defined parameter and g_w^{ij} is the constraint function. Once an optimal solution has been found, the daughter passes the optimal variable values, constraint values, and sensitivity derivatives to its parent. The parent used this information to update its design variable bounds and objective equation, which is again, the cumulative constraint violation, but is modified by its daughter's cumulative constraint function and sensitivity.

$$C^{ij} \equiv KS(g_w^{ij}) = \frac{1}{\rho} \ln \left[\sum_w \exp(\rho g_w^{ij}) \right] \quad (2)$$

With everything set at this level, it is optimized and the results passed to its parent. This pattern continues until the highest level is reached. At this level, the objective function is simply the overall objective, and the constraints' boundary conditions of the overall structure are used instead of internal reaction forces. Once an optimal solution is found at this level, substructuring analysis is again performed for all daughter levels, and the process repeats until convergence.

To validate the procedure, Sobieszczanski-Sobieski's team performed an analysis of the same portal frame problem as Beers and Vanderplaats used, but with some minor modifications to facilitate a three-level analysis. Sobieszczanski-Sobieski reports that the method worked as well or better than expected, and the biggest issues encountered by the team were data communication issues associated with file names.

In a more recent development, Yao et al [21] present a multilevel optimization procedure that combines the advantages of two previously existing methods into a singular multi-stage process. An approach called multi-discipline-feasible (MDF) is combined with concurrent subspace optimization (CSSO) into a multi-stage and multilevel optimization procedure for solving multiple discipline problems (the method is referred to as MDF-CSSO). In the first stage, discipline specific surrogate models are independently generated using a Latin Hypercube DOE. MDF is then used to find the optimal design based on the surrogate models. The optimal design is then analyzed using the true system model and the surrogates are updated accordingly. MDF is again used and the system model is again checked. This is sequentially iterated through until convergence. The optimal design of the first stage is then used as the baseline in the second stage. In the second stage, the system level is decomposed and CSSO is used. For detailed information on CSSO, see [22]. For all disciplines in parallel, a discipline specialist optimizes a local subspace of the high-fidelity model, using the surrogates to estimate the state variables from non-local subspaces, which are held constant. Each

discipline's optimal solution is then analyzed using the true system model. All surrogate models are updated and CSSO is again performed. This is sequentially iterated until convergence, at which point an optimal solution has been found. Yao et al [21] also introduces a probabilistic version of their method for determining robust solutions, which updates response surface training sets accounting for local uncertainties. Using a problem involving the conceptual design of a satellite, both methods are validated.

All of the aforementioned multilevel techniques have been shown to be valid, and have their own benefits and drawbacks. A feature they all have in common is the use of sensitivity analyses; either of a substructure, subsystem, or surrogate model. Sensitivity analyses can be extremely powerful, and utilizing the information from them to limit or reduce the search space will generally accelerate convergence (in terms of iterations). They are also, however, somewhat computationally costly. For problems with just a few levels or a small number of variables, the benefits may outweigh the cost. But, as the number of levels increases, the total computational cost added to a multilevel method by the sensitivity analysis feature compounds. For problems with a very large number of levels, the benefits added to a multilevel method from a sensitivity analysis feature could become overshadowed by the additional cost associated with the sensitivity analyses.

As presented by Chase, Sidhu, and Averill [23], COMPOSE (Component Optimization within a System Environment) is similar to [19] and [20] in some aspects, but with a major difference: it is a direct iterative approach, meaning that sensitivity analyses are not needed. Rather than using sensitivity studies and gradients to update the boundary condition constraints, a stochastic weighting function is used. COMPOSE is particularly effective in multilevel problems where the optimization of a subsystem is highly dependent on its boundary conditions. COMPOSE was demonstrated in a crashworthiness problem in which they performed shape optimization on several structural components of a vehicle model that is subjected to a roof crush analysis. They showed that

COMPOSE consistently yielded feasible results that reduced the mass of the system while maintaining an acceptable reaction force. This was then compared to the same problem, but the subsystems were optimized independently and reassembled for a system analysis. This method did not reliably produce feasible results.

Goodman et al [24] utilized COMPOSE in a crashworthiness problem using shape optimization to design a truck's lower compartment rails. The team showed how COMPOSE can be used with collaborative independent agents on problems with enormous design spaces and computationally expensive analyses. It basically utilizes distributed computing with selective communication among agents to optimize different portions of the design space, which are modeled at multiple levels of detail. Again using shape optimization of an automotive rail, this was shown to reduce the computing time required to find a good design.

1.4 Objective of this Work

The development of UHD-SoS specific design and analysis processes aims to reduce the computational expense of creating UHD-SoS products and therefore increase their utilization in industry. The application of these processes to additively manufactured designs could improve the quality of AM products by providing precise control over the infill and microstructure without increasing the production cost. Optimization of these structures could be performed to yield parts with non-uniform microscale features that result in desirable nonlinear behaviors.

To that end, the optimization techniques outlined and reported in the next several chapters are aimed toward the development of an efficient optimization process to aid in the design and application of UHD-SoS. In addition to developing a better understanding of how multilevel optimization strategies can be applied to SoS, the current work aims to develop a technique that is scalable in terms of both computational cost and memory requirements, making it possible to design large scale SoS with complex microstructures at multiple levels.

1.5 Organization of the Thesis

Chapter 2 provides an insight into the optimization tools, analysis models, and other details of this work. It also contains a table of variables and variable settings that were used in every study. Each chapter will refer back to this table in order to avoid redundant information.

Chapter 3 provides an introduction to multilevel optimization by outlining the procedure for Two-Step Serial Optimization and illustrating its validity on a simple problem involving a 3 bar member.

Chapter 4 introduces a process for Multilevel optimization, in which the analysis levels communicate through boundary conditions and applied loads. Again the 3 bar member problem is used as an example. An investigation into different methods of constraining the microlevel analyses is explored.

Chapter 5 expands on the results of Chapter 3 by applying the most promising method of constraining to a three dimensional transmission tower problem. This chapter also explores the validity of this process on Multiphysics problems by adding a thermal load into the problem statement. In another study, the concept of Adaptive Tolerance Constraints is also introduced and explored. The results are then compared to original study.

Chapter 6 summarizes the performed studies and discusses the results. The results and data presented should be used with care. This work is meant only to show overall trends and act as a proof-of-concept.

Due to the unrefined nature of the LatticeMaker program during the span of this work, certain complexities existed that made parallelization difficult. In order to successfully parallelize the micro level studies, measures had to be taken that resulted in the occasional error design to be falsely reported. These false errors would likely have an effect (albeit a small effect) on the progression of

evaluated designs as determined by SHERPA. Because of this fact, exact recreation of the data presented in the following chapters is unlikely; however, the effect of these false errors should be minute enough that recreation of the qualitative observations and numerical trends presented will be unaffected. To further reduce the impact of the false errors, measures were taken that limited the allowable number of error designs: if an iteration contained a micro level optimization process that produced more than two false errors for any given member, that member's micro level optimization process was rerun.

CHAPTER 2: Optimization Details and Analysis Tools

To manage optimization studies and perform the optimization processes in the current studies, HEEDS Multidisciplinary Design Optimization software was utilized. In this environment, analyses can be defined and their execution automated with relatively little or no user interaction required.

2.1 HEEDS Multidisciplinary Design Optimization

An evaluation consists of a set of analyses that work together to predict the performance of a design. For example, an evaluation might consist of a CAD analysis, where a parametric model is modified, a meshing process, where the CAD model is preprocessed and meshed, and a Finite Element analysis, where the meshed model is analyzed. An optimization process contains specific information, such as execution commands, communication methods, performance function and normalization factors, and design variable and response definitions.

One of the benefits that HEEDS provides is analysis portals. A portal is an interface environment that provides a direct way to communicate with and tag the input and output files (referred to henceforth as “I/O files”) through a graphical user interface. A general portal exists that will present ASCII based I/O files as a spreadsheet where a cell’s content is determined by the delimitation of the I/O file according to a set of symbols. Alternatively, scripting can be used to parse the file for certain landmarks. For applications that do not use ASCII based I/O files as well as very common ASCII based applications, built-in scripts will parse the files, decode the content (if necessary), and present the data in a meaningful and intuitive way. This makes tagging the I/O files efficient.

The term “tagging” refers to the marking of an I/O file to define the location of a parameter of interest within the file. The tagging of design variables will set the value at the given location of the file and the tagging of responses will retrieve the value at the given location. In this way, the designs

can be communicated to the analysis model: as the optimization process defines new designs, the design variable values will be entered into the input file, the analysis will be executed, and the responses will be retrieved from the output file.

What drives an optimization process is an optimization algorithm. These algorithms are what determine the progression of design variable values based on the response history. A good optimization algorithm will thoroughly explore the design space (a space defined by every design variable's bounds) while also showing quick convergence to an optimal design. No algorithm exists that is "best" for all possible problems. Certain algorithms are better suited for certain classes of problems, and will tend to perform better on those problems than other types of algorithms. While this fact is steadfast, HEEDS can utilize a proprietary algorithm that performs well on many types of problems.

SHERPA (Simultaneous Hybrid Exploration that is Robust, Progressive, and Adaptive) [25] is a proprietary hybrid and adaptive search strategy available within the HEEDS MDO software code. During a single parametric optimization study, SHERPA uses the components of multiple search methods simultaneously in a unique blended manner. This approach attempts to take advantage of the best attributes of each method. Attributes from a combination of global and local search methods are used, and each participating approach contains internal tuning parameters that are modified automatically during the search according to knowledge gained about the nature of the design space.

This evolving knowledge about the design space also determines when and to what extent each approach contributes to the search. In other words, SHERPA efficiently learns about the design space and adapts itself so as to effectively search many kinds of design spaces, even very complicated ones. SHERPA is a direct optimization algorithm in which all function evaluations are performed using the actual model, as opposed to using an approximate response surface model. SHERPA does

not require solution gradients to exist. The only parameter that must be specified by the user is the number of allowable evaluations, though other parameters can be tuned if desired.

Common to all optimization algorithms is the existence of a performance function, sometimes referred to as a cost function. A performance function is a way to rank the quality of a design in a quantitative way. This is the mechanism for determining the next design to try. Different algorithms utilize the performance function in different ways, and may have the performance function defined differently, but all algorithms must use some sort of ranking method to drive the design search. A very common performance function is given in Equation (3)

$$P = \sum_{i=1}^N \left((ObjSign_i)(ObjWeight_i) \frac{ObjResVal_i}{ObjNorm_i} \right) - \sum_{j=1}^M \left((ConWeight_j) \frac{ConViolation_j}{ConNorm_j} \right)^2 \quad (3)$$

where N represents the number of objective responses, and M represents the number of constraint responses. The *ConViolation* term is a value that will be zero if the constraint is feasible, otherwise the distance that the response is from the constraint limit. This could either be positive or negative, depending on the type of constraint. Notice that it is getting squared and that there is a negative in front – combined, these features force this term to either reduce the value of the performance function, or leave it unaffected. The *ObjSign* is a term that represents the type of optimization for that response – it will have a value of either 1 if the response is getting maximized or -1 if it is getting minimized. In this way, a better design will always have an increasing performance value.

The normalization terms in Equation (3) are very important. Normally, the Objective Normalization terms are taken to be the baseline value for the given response, and the Constraint Normalization terms are taken as the response limits. Normalization in this way eliminates biases caused from unit discrepancies, allowing all responses to affect the performance equally (or at least on the same order of magnitude). *ObjWeight* and *ConWeight* are weighting terms that control the

impact of a term on the performance function. Usually, *ObjWeight* is set to one and *ConWeight* is set to 10,000. This makes it very difficult for an infeasible design to perform better than a feasible design.

HEEDS utilizes a performance function that is similar to the one given in Equation (3), but is slightly different. Rather than only utilizing a linear term for the objectives, HEEDS also offers a quadratic option. Both linear and quadrature terms are available for constraints as well. In the interest of being as easy to use as possible, HEEDS has default values entered into the weighting factors and normalization terms. When evaluated, the default weighting values convert the quadratic performance function into the function defined by Equation (3). The default values for the normalization terms and weighting factors can be easily overwritten, allowing the user to configure the studies exactly as they please. This customization of the normalization terms is very important to the studies and experiments presented in the following chapters.

2.2 Analysis Model and Optimization Process

In Chapter 3, different constraint techniques are examined. A large part of these experiments involve an investigation into the performance function through different methods of objective and constraint normalization. As this defines an important part of the optimization process, this seems like a natural location to discuss the normalization techniques. However, the normalization methods depend very much on experimental context, so these methods will be discussed when they arise in later chapters. The exact analysis model is also dependent on experimental context as well; however, several similarities exist between every model used, which facilitates a discussion at this juncture.

The analysis models used throughout the scope of this work consisted of various combinations and arrangements of the same three analysis portals to perform variations of the same analyses. A general analysis portal was used to execute the LatticeMaker Analyses and then one or more Python portals acted as a bridge to transport the LatticeMaker data to a corresponding Abaqus portal for Finite Element analysis.

LatticeMaker [26] is a Java based applet that was developed by a research team from Georgia Institute of Technology. This program can generate a cylindrical lattice structure based on the values of several design variables entered in the input file. The structures created by LatticeMaker consist of a hexagonal mesh that is radially offset from a reinforcing triangular mesh. The size of the mesh as well as the offset distance are just a few of the variables contained within the input file. The input file is a simple text file and contains all of the input variables as well as the desired queries. LatticeMaker currently supports a total of six lattice parameters, which are defined in Table 1.

Table 1. LatticeMaker input file parameters, meaning, and type

| Parameter | Significance | Type |
|--------------------|--|-----------------|
| ringRad | Defines the outer radius of the lattice structure's base | Float or Double |
| ballRad | Defines the radius of the mesh beams at the base of the structure | Float or Double |
| numCells | Controls the number of hexagonal units around the structure | Integer |
| numRepeats | Controls the number of units along the axis of the structure | Integer |
| scale | Controls the taper of the structure as well as the taper of the mesh beams | Float or Double |
| reinforcementDepth | Defines the offset distance between the two meshes | Float or Double |

Because a design that has a larger ballRad than ringRad is feasible according to LatticeMaker but geometrically infeasible in actual applications, these designs should be avoided. A condition could be set up in HEEDS that treats designs of this nature as infeasible; however, this would yield a large number of infeasible designs. Alternatively, a new variable could be introduced that acts as a scale factor. In this way, ballRad could be treated as a dependent variable relying on this new variable and the ringRad rather than a continuous variable. A similar method could be used to enforce a geometrically feasible reinforcementDepth variable. Table 2 gives the LatticeMaker variable definitions used throughout this work. Notice the “scale” variable in Table 2 is described as a constant. This is because LatticeMaker requires it to be defined in the input file, so it must be included in the problem, though it will not have any impact on the optimization of the system.

Table 2. Lattice design variables as defined in HEEDS

| Variable | Type | Variable Definition | | |
|--------------------|----------------------|---|-------------|------------|
| | | Lower Bound | Upper Bound | Resolution |
| ringRad | Continuous | 2 | 750 | 749 |
| numCells | Continuous (Integer) | 2 | 20 | 19 |
| numRepeats | Continuous (Integer) | 2 | 20 | 19 |
| BallScale | Continuous | 0.05 | 0.95 | 91 |
| ReinforcementGap | Continuous | 0.05 | 0.95 | 91 |
| scale | Constant | 1 | | |
| ballRad | Dependent | ringRad*BallScale | | |
| reinforcementDepth | Dependent | $(\text{ringRad} - \text{ballRad}) * (1 - \text{ReinforcementGap})$ | | |

Two output files are generated from LatticeMaker, both containing the same information, but provided in a slightly different format. The basic content includes the coordinates for every node, as well as nodal connectivity information. In addition to coordinates, a radius is also defined for every node in the output files. Using this information the lattice structure can be easily assembled in any programming environment.

Because LatticeMaker is a research code, it is not yet entirely general or recommended for commercial use. The input file must be named “input.txt” and must be located in the same directory as the applet (“Lattice.jar”). Accordingly, the applet and input file must be copied into every analysis directory. This creates a large amount of unnecessary data communication that could be avoided if the program could be called from a static location. The only real consequence of this is that it may slow down the analysis slightly, and uses unnecessary memory space.

Similarly, the output files generated by LatticeMaker are always named “lattice1.txt” and “lattice2.txt” and they cannot be renamed. This complicates the communication stream in HEEDS because it cannot accept I/O files with the same name from different sources. To overcome this

obstacle, rather than using the LatticeMaker output file as the input file for the Python analysis (which is the usual practice with HEEDS), the Python package must retrieve the LatticeMaker output file from the LatticeMaker analysis directory. This requires an extra set of variables to be defined that would otherwise be unnecessary, which makes communication a bit more tedious and troubleshooting more complicated. Nonetheless, these can be overcome with some careful planning.

When LatticeMaker is used without the visualization tool, the output files will be generated, but the process will not end. This causes difficulties when using LatticeMaker as an analysis in an optimization study because a process will be started during an evaluation and will not end. The next evaluation will start a new process that will not end. This continues until the system runs out of memory. The work around for this was to utilize HEEDS' optional pre-analysis command to end the LatticeMaker (java) process at the beginning of the next Python analysis. Unfortunately, on Windows systems, the process id for a particular instance of LatticeMaker cannot be determined, so another method must be used. The solution used in this work was to end java processes based on CPU time using the command:

```
taskkill /fi "cputime ge 00:03:00" /im java* /f
```

This solution worked fairly well, but it was not perfect. Because parallelization was used, and multiple studies were performed at once, whenever a Python analysis commenced, there was the possibility that it would interrupt a LatticeMaker evaluation. In an attempt to alleviate this, every LatticeMaker analysis was looped several times, increasing the probability that it will complete without interruption. This worked well, but not perfectly: error designs due to Python interruptions did occur in most studies.

As previously mentioned, the Python analysis was always performed after the LatticeMaker analyses and bridged the gap between LatticeMaker and Abaqus. Based on information provided in

an input file, the Python analysis would retrieve lattice data and manipulate the geometry according to the data in a structure file. The program would translate, rotate, scale, and (depending on the desired purpose) combine lattice structures in order to replace predefined members in the structure input file. Because of its purpose of manipulating lattice structures to conform to a model, the Python package was dubbed “LatticeModeler”, paying homage to its LatticeMaker counterpart.

Some of the parameters defined in the input file include node and element group definitions, non-lattice element cross section shapes, material properties, boundary conditions, I/O file names, and analysis properties. One of the important analysis properties defines the analysis level. Acceptable values for this parameter are “Micro”, “Macro”, or “Global”. If the macro level option is chosen, LatticeModeler will not retrieve any lattice data. Rather, it will read the structure file and write an Abaqus input file for it, leaving the geometry unmodified (See Figure 1b). If the global option is used, the structure file will be read, then, for every defined lattice member, the corresponding LatticeMaker output file will be read, the data manipulated, and the macro level member replaced with the new lattice data. Once every lattice member has been inserted into the model, the Abaqus input file is written, containing every node and element defined in the model (See Figure 1a). If the micro level option is requested, only a single lattice will be read and transformed according to the corresponding structure file member. Rather than inserting the lattice data into the structure data, the lattice data itself will be written to the Abaqus input file (See Figure 1c).

Other important analysis properties defined in the LatticeModeler input file include: analysis type, analysis options, and element type. Supported analysis types are currently limited to Static and (uncoupled) Heat Transfer. Analysis options can include things such as “Perturbation” for static analyses, or “Steady State” for thermal analyses. Several element types are supported, but the most commonly used are Euler beams in space (B33) for static analyses and thermally active one dimensional two node continuum elements (DCC1D2) for heat transfer analyses.

There are many analysis-dependent properties that need to be defined in the input file as well. For instance, boundary conditions, element type, element cross-section, and node constraint types are all analysis-specific. A static analysis (within the scope of this work) could have either a point load applied to a node, or a set displacement. In either case, a BEAM-type Multi-Point Constraint (MPC) is used to distribute the load to the nodes circling the lattice tip or base. Because moments of inertia play a major role in structural analyses, the element cross-sections needed to be defined. LatticeModeler (and Abaqus) accept several options, but lattice elements were given a circular cross-section, while macro level elements and non-lattice global elements were defined with an annular cross-section.

In contrast, heat transfer analyses (within the scope of this work) could utilize thermal flux loads, convection, and temperature value boundary conditions. Because BEAM -type MPCs do not have a temperature degree of freedom, TIE-type MPCs are utilized for thermal analyses. Inertia is not applicable to uncoupled heat transfer problems, so the cross-sectional shape did not need to be defined. Only the cross-sectional area is required. However, the easiest way to define an area is to define a shape and automatically calculate the area, which is how LatticeModeler handles it.

2.3 Analysis Model Validation

To verify the accuracy of the optimization process and analysis models, a validation study was performed. This study used the optimization process set up in HEEDS to evaluate a single design for a simple cantilevered beam. The results presented by HEEDS were compared with the results shown in the Abaqus user interface, as well as an analytical solution using mechanics of materials concepts. A diagram of the study is given in Figure 3, followed by parameter definitions and values in Table 3.

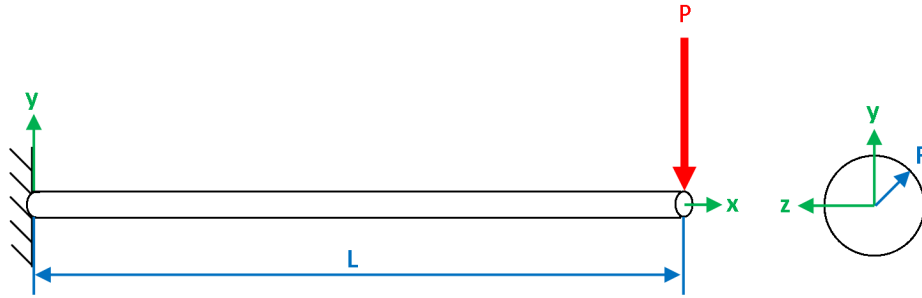


Figure 3. Diagram of the Validation Test problem used to verify an accurate analysis model is being used.

Table 3. Validation Test problem parameters, definitions, and values

| Symbol | Definition | Value |
|--------|----------------------------|-----------------------------------|
| R | Beam Radius (See Figure 3) | 0.3 (m) |
| L | Beam Length (See Figure 3) | 3 (m) |
| P | Point Load Force | $\langle 0, -500, 0 \rangle$ (kN) |

The numerical approach utilized a static Abaqus analysis with three Euler beam elements representing the domain of the beam. While it isn't needed in the analytical solution, the numerical solutions required a material to be defined, so the properties of stainless steel ($E = 190 \text{ GPa}$, $\nu = 0.3$) were used. It is expected that these results are identical, as they are getting extracted from the same output file. A comparison of these results to the analytical solution will validate (or invalidate) the model.

To solve the system by hand, a free body diagram was set up. Because this is a static system, equilibrium was utilized to determine the reaction forces and moments. With the reactions known (Table 4), the internal shears and moments could be calculated and plotted (Figure 4). Noticing that the maximum (magnitude of) shear and moment both occur at the base of the beam, it can be realized that the only stress state of interest is at this location ($x = 0$).

Table 4. Validation Test problem reaction forces and moments

| Reaction | Value |
|----------|--------------|
| F_x | 500 (kN) |
| F_y | 0 |
| M_z | 1,500 (kN·m) |

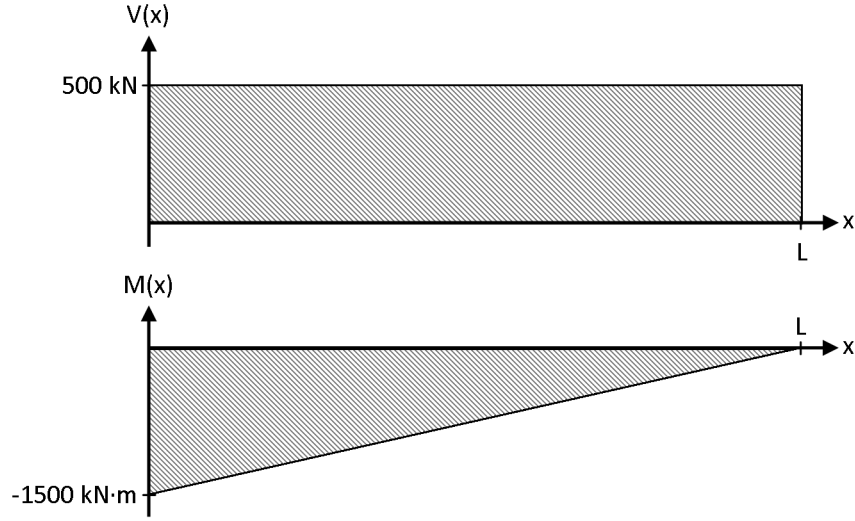


Figure 4. Validation Test's internal shear and moment diagrams.

Before the stress state can be determined, there are several geometric properties that must be calculated. The formulae and values for the cross-sectional area of the beam as well as the first and second moments of area are provided in Table 5. The First Area Moment of Inertia is calculated at the centroid of the beam, which is assumed to be the location of maximum shear stress.

Table 5. Validation Test's beam's cross-sectional properties

| Property | Formula | Value |
|-------------------------------|---------------------------|---|
| Area | $A = \pi R^2$ | $282.7 \times 10^{-3} \text{ (m}^2\text{)}$ |
| First Area Moment of Inertia | $Q = \frac{2}{3} R^3$ | $18.0 \times 10^{-3} \text{ (m}^3\text{)}$ |
| Second Area Moment of Inertia | $I = \frac{1}{4} \pi R^4$ | $6.362 \times 10^{-3} \text{ (m}^4\text{)}$ |

Then, knowing the maximum shear ($V_{Max} = 500 \text{ kN}$) and moment ($M_{Max} = -1,500 \text{ kN} \cdot \text{m}$) values, the stress components could be calculated and compared. At the top surface (which is assumed to be the location of maximum normal stress due to bending) of the beam:

$$\sigma_x = \frac{P_x}{A} - \frac{M_{Max} R}{I} = 70.74 \text{ MPa}$$

$$\tau_{xy} = \frac{V_{Max} Q}{2IR} = 0 \text{ MPa}$$

Because $Q = 0$ at the surface of the beam. At the centroid of the beam:

$$\sigma_x = \frac{P_x}{A} - \frac{M_{Max}R}{I} = 0 \text{ MPa}$$

$$\tau_{xy} = \frac{V_{Max}Q}{2IR} = 2.358 \text{ MPa}$$

Because $R = 0$ at the centroid of the beam. And because this is a plane stress scenario, $\sigma_y = 0$ for both cases. With the stress state at both locations now known, the principal stresses can be determined using Mohr's circle. Again taking advantage of the plane stress situation, the equation for Mohr's circle can be reduced:

$$\sigma_{1,2} = \frac{\sigma_x}{2} \pm \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \tau_{xy}^2} = \{70.74, 0\} \text{ MPa}$$

At the surface of the beam, and

$$\sigma_{1,2} = \frac{\sigma_x}{2} \pm \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \tau_{xy}^2} = \{2.358, -2.358\} \text{ MPa}$$

At the centroid.

Finally, with the principal stresses known, the Maximum Distortion Energy (Von Mises) theory for plane stress [Equation (4)] can be used to determine the Von Mises stress (also known as Mises stress) for both scenarios. In the case of bending stress, the Mises stress was found to be $\sigma_{Mises} = 70.74 \text{ MPa}$ and for the case of shear stress, the Mises stress was found to be $\sigma_{Mises} = 4.08 \text{ MPa}$. Clearly, the bending stress is the primary mode of failure.

$$\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2 = \sigma_{Mises}^2 \quad (4)$$

The analytically determined Mises stress for the primary failure mode is presented in Table 6, along with the stress found from the two numerical methods (HEEDS output and the Abaqus GUI). The numerical results are also compared to the analytical solution and the percent error is shown.

Table 6. Validation Test problem results and percent error relative to the analytical solution for the primary mode of failure

| Method | Mises Stress (MPa) | Error (%) |
|--------------|--------------------|-----------|
| Analytical | 70.74 | -- |
| Abaqus GUI | 70.74 | 0.0 |
| HEEDS Output | 68.08 | 3.760 |

The comparison between the analytical solution and the solution provided by Abaqus' user interface shows negligible discrepancy. Considering only three elements were used, this is a very accurate solution. The comparison of the solution extracted by HEEDS, however, shows a bit more discrepancy. An error of almost four percent is not terrible, but it is significant considering the extremely small error found from the GUI. Both methods extracted the response from the same output file, so why are they not identical?

It turns out that HEEDS extracts the raw data from the output file, whereas Abaqus' GUI post-processes it. The GUI extrapolates the elemental stresses to the nodes and then tunes the nodal stresses by averaging the stresses from connected elements. As the data in Table 6 shows, this tends to result in more accurate stress predications.

Now that the discrepancy between the numerical techniques has been reconciled and validated against an analytical technique, the analysis model can be confidently utilized, so long as the user understands that the data extracted by HEEDS represents elemental data. Because of this, there may be a small discrepancy if a solution is checked using Abaqus' user interface.

CHAPTER 3: Two-Step Serial Optimization

For relatively simple structures, a Two-Step Serial Optimization process may be employed. This process involves first optimizing a macroscale model composed of homogenized members. The reactions and resulting displacements of the optimal macroscale model are then collected and applied to each subsystem (micro) model. Finally, each micro model can be optimized individually. This process is fully defined in the following section. To illustrate the Two-Step Serial Optimization process, a simple three-member structure will be defined, decomposed, and analyzed in the subsequent sections. It is important to note that this is not a generalized method, and will only work on specific problems. It is only included here to act as base to expand from in the next chapters.

3.1 Three-Member Structure Problem Statement

As an initial example to demonstrate the Two-Step Serial Optimization process, a simple two dimensional structure will be analyzed. This structure initially consists of three tubular members welded at a point. The welded end has an applied point-load, and the non-welded ends of the tube members are subjected to either fixed or moving-pin joint boundary conditions, as shown in Figure 5, below. It is desired that the mass of the structure be minimized while maintaining the vertical deflection of the welded tip to be less than a given value. To control these responses, certain geometric parameters may be modified: The global structure's dimensions must remain unchanged, but the outside radius and the wall thickness of each tubular member can be modified independently.

In order to keep this problem as general as possible, the Young's Modulus is set to unity. To satisfy the energy bounds, a Poisson's ratio of 0.3 was used to calculate an appropriate Shear Modulus. This was then rounded to the nearest thousandth. A full list of problem parameters is given in Table 7, including global dimensions, material properties, and load definitions. Figure 6 is included to accompany Table 7 and provide a visual reference for the global geometric definitions.

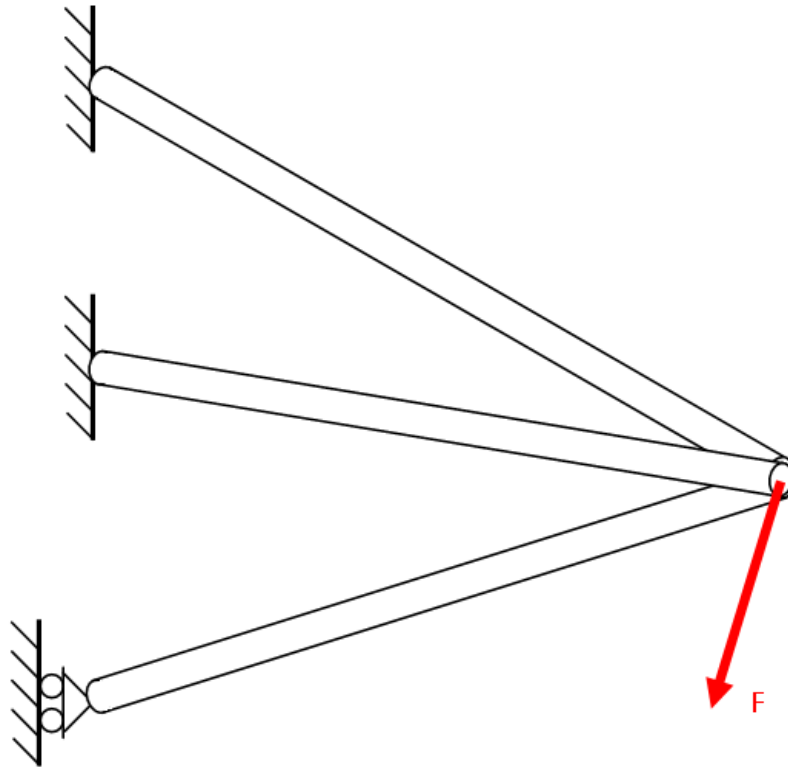


Figure 5. Diagram of simple Three Member Structure problem that will be analyzed using a Two-Step Serial optimization process.

Table 7. Three-Member Structure problem parameters, definitions, and values

| Symbol | Definition | Value |
|--------|------------------------------------|----------------------|
| L | Structure Dimension (See Figure 6) | 15 (m) |
| H_1 | Structure Dimension (See Figure 6) | 12 (m) |
| H_2 | Structure Dimension (See Figure 6) | 2 (m) |
| H_3 | Structure Dimension (See Figure 6) | 6 (m) |
| E | Young's Modulus | 1 (Pa) |
| G | Shear Modulus | 0.385 (Pa) |
| F | Point Load Force | $<-20, -50, 0>$ (kN) |

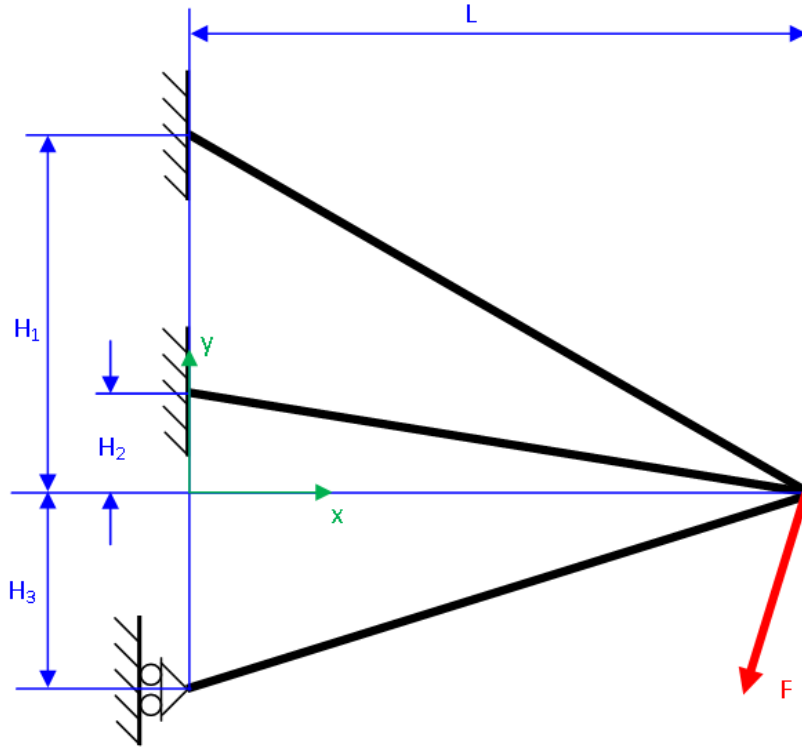


Figure 6. Three-Member Structure geometric parameter definitions used to set up the optimization analysis model

Using this information, a global optimization problem statement can be defined in Non-Linear Programming (NLP) form. As the global geometry is fixed, the lengths of each member are left unchanged. This causes a direct proportionality between the total mass of the structure and the total volume, which can be substituted into the NLP description as an objective. Thus, the NLP description becomes:

| | | | |
|-------------------|---|---------------------|-----|
| Objectives: | Minimize Volume | | |
| Constraints: | $U_y^{(4)} \leq U_{Max}$ | | |
| Design Variables: | $R_{Lower}^{(i)} \leq R^{(i)} \leq R_{Upper}^{(i)}$ | $i \in \{1, 2, 3\}$ | |
| | $t_{Lower}^{(i)} \leq t^{(i)} \leq t_{Upper}^{(i)}$ | $i \in \{1, 2, 3\}$ | (5) |

where U is the displacement vector, R is the outside radius, t is the wall thickness, and i denotes an index number symbolizing a given tubular member of the global structure. But what if the optimal solution found from this problem definition was still too massive?

A structure of structures concept could be used to replace the tubular members with lattice structures, as shown in Figure 7. This SoS model, would likely result in much lighter solutions with similar deflections. On the other hand, each global analysis of this structure would be very computationally expensive due to the large number of elements that define a lattice structure.

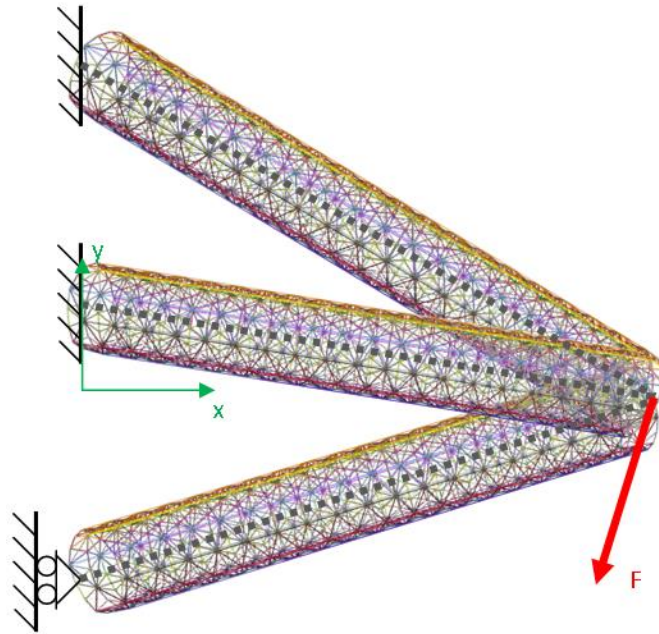


Figure 7. Structure of Structures representation of the Three-Member Structure problem

Furthermore, as each lattice within the structure has five independently controllable parameters (as shown in Table 2) defined in LatticeMaker for each member, the optimization of this structure would take place in a 15-dimension search space. This would slow down the rate of convergence, especially as the number of lattice members increases. If the optimal deflections could be determined, however, the structure could be decomposed into individual lattices and be optimized

independently and in parallel. This is the concept behind the Two-Step Serial Optimization process described in the following section.

3.2 Two-Step Serial Optimization Process Flow

Before the Two-Step Serial Optimization process is outlined, it must be made clear that this is a very specific process and is only valid for structural optimization problems described in Equations (5). In particular, there must be a constraint placed on displacement. With this understood, the procedure can now be outlined.

The process has two phases. The first phase involves the tubular model used in the original, non-SoS representation of the problem. The results of the first phase are then used as either prescribed conditions or constraints in the second phase, in which the global SoS model is decomposed and the subsystem models are optimized individually and in parallel. Figure 8, below, is a flow diagram of this process. In this figure, solid lines represent the process flow, and dashed lines represent the communication of information.

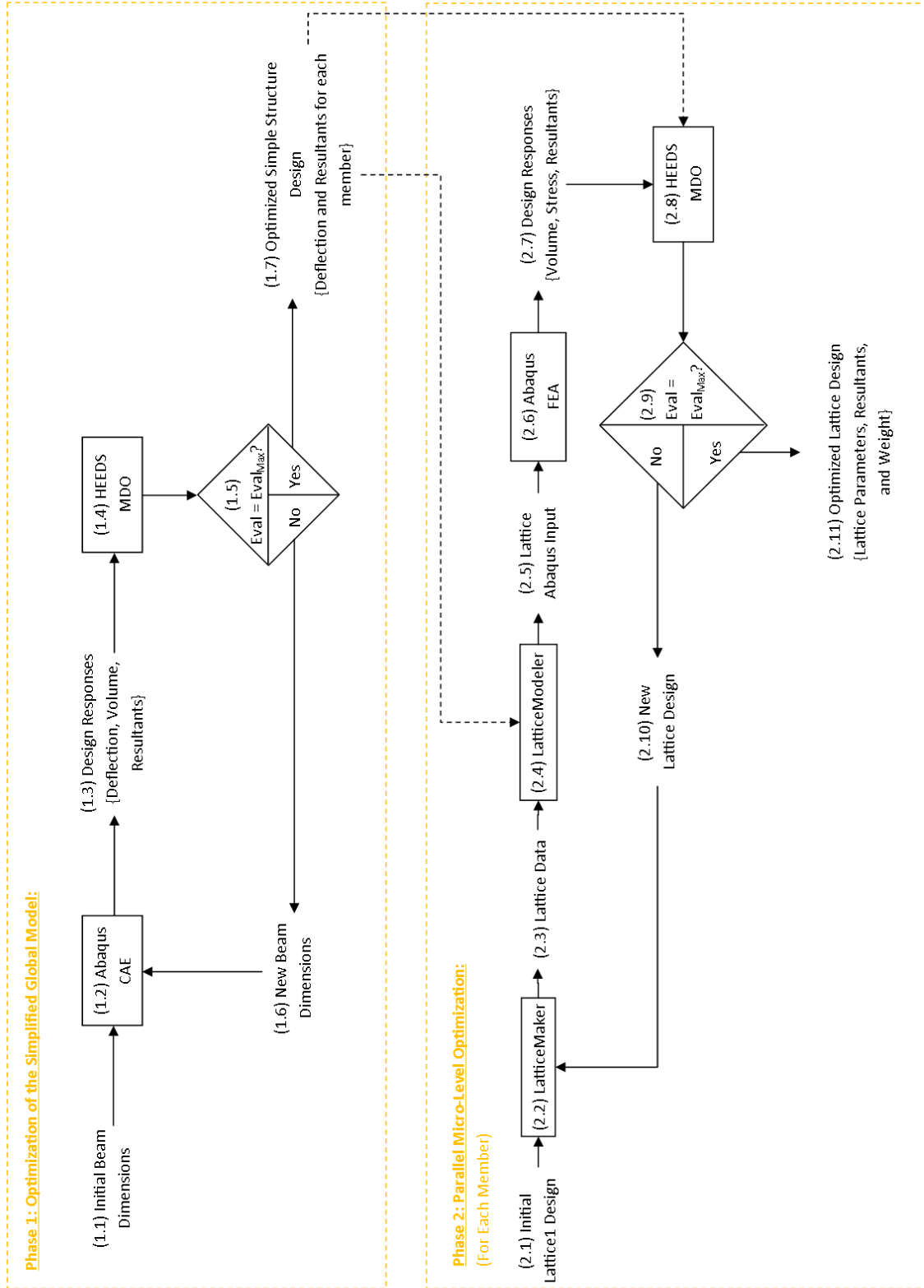


Figure 8. Two-Step Serial Optimization process flow diagram

3.2.1 Two-Step Serial Optimization - Phase 1

As shown in Figure 8, Phase 1 is rather simple, involving only a single analysis in the analysis model, which gets looped during the optimization study. The main goal of Phase 1 is to determine the optimal displacements for a prescribed volume using the simplified model. Because Abaqus allows for 1D beam elements that have simple cross-sectional shapes to be parameterized rather than geometrically defined, the macro level model can be used, significantly reducing the number of computations required. Because it is computationally inexpensive, the number of evaluations can be increased. Accordingly, 500 evaluations will be used in this stage, though this is far more evaluations than are needed for this problem. The objective for this phase is to minimize the deflection, while constraining the volume to be less than a certain amount:

$$\begin{array}{ll} \text{Objectives:} & \text{Minimize } Deflection \\ \text{Constraints:} & Volume \leq Volume_{Max} \\ \text{Design Variables:} & \text{(Problem Dependent)} \end{array} \quad (6)$$

While the problem defined by Equations (6) were used in this work, in theory, the Phase 1 problem statement could also resemble:

$$\begin{array}{ll} \text{Objectives:} & \text{Minimize } Volume \\ \text{Constraints:} & Deflection \leq Deflection_{Max} \\ \text{Design Variables:} & \text{(Problem Dependent)} \end{array}$$

It is important to note that in the two problem descriptions above (especially in the latter), *Deflection* is a scalar value that refers to the deflection of a particular point, and is a function of the displacements. The term “displacement” refers to a component of the vector (or the entire vector, depending on context) describing the position (in all 6 degrees of freedom) of a node relative to its original location. In some situations, “deflection” can refer to a particular element of a displacement vector, or it can refer to a function involving multiple elements that yields a scalar value.

3.2.2 Two-Step Serial Optimization - Phase 2

Once an optimal solution to the macro level problem has been determined, the relevant information is passed to the micro level models, thus initializing Phase 2. Specifically, the displacements are passed to LatticeModeler, which uses the information to define the prescribed displacement of the micro level model's MPC nodes in the Abaqus input file, as shown in Figure 10. The reactions are passed to HEEDS, which uses them to define the optimization constraints. The term “reactions” refers to the vector quantity detailing the forces and moments at the boundaries of the system that result from the prescribed condition. The objective of Phase 2 is to minimize the volume of each micro level model, given in Equations (7), where Λ_j is the reaction value in degree of freedom (DoF) j , and $\tilde{\Lambda}_j$ is the corresponding value of the Phase 1 reaction. The effect of this constraint is to maintain or exceed the stiffness of the system level (macro) model.

| | | | |
|-------------------|------------------------------------|----------------------------|-------|
| Objectives: | Minimize Volume | | |
| Constraints: | $\Lambda_j \geq \tilde{\Lambda}_j$ | $j \in \{1, 2, \dots, 6\}$ | |
| Design Variables: | (Problem Dependent) | | (7) |

It is important to note that the equation used in the constraint of Equations (7) assumes that $\tilde{\Lambda}_j$ is a positive value, however, this is not always true. For negative reaction values, the inequality sign is flipped. This nomenclature (and positive value assumption) will be held throughout this section, keeping in mind that the inequality is reversed for negative reactions.

According to typical constraint schemes, each constraint in Equations (7) would be normalized by its limit. This means there would be 6 independent normalization factors. After conducting a few pilot studies, it was found that there are often constraints with a value of zero (due to the unrestricted DoFs of Node 3). Rather than simply normalizing these constraints by unity, which is a common practice for zero constraints, they were normalized by unity and the reaction became double bounded, limiting the reaction to be between $\pm 10^{-6}$, regardless of the reaction's units.

Other than the non-zero constraints, all other normalization and weighting factors were left alone, in an attempt to make the method as general as possible. The default value of 150 evaluations was used with SHERPA as the optimization algorithm. The baseline variables were the same between all trials.

3.3 Problem Statement Decomposition for Two-Step Serial Optimization

As mentioned in the previous section, the system level (macro) model can be used in the first phase, due to the ability to define the cross section parametrically in Abaqus rather than geometrically using LatticeMaker. Accordingly, Figure 9 can be used as the model for Phase 1, for which the optimization problem statement is given by Equations (8). For illustration purposes, the constraint value in these equations was decided upon by using the volume of an arbitrary design near the center of the design space.

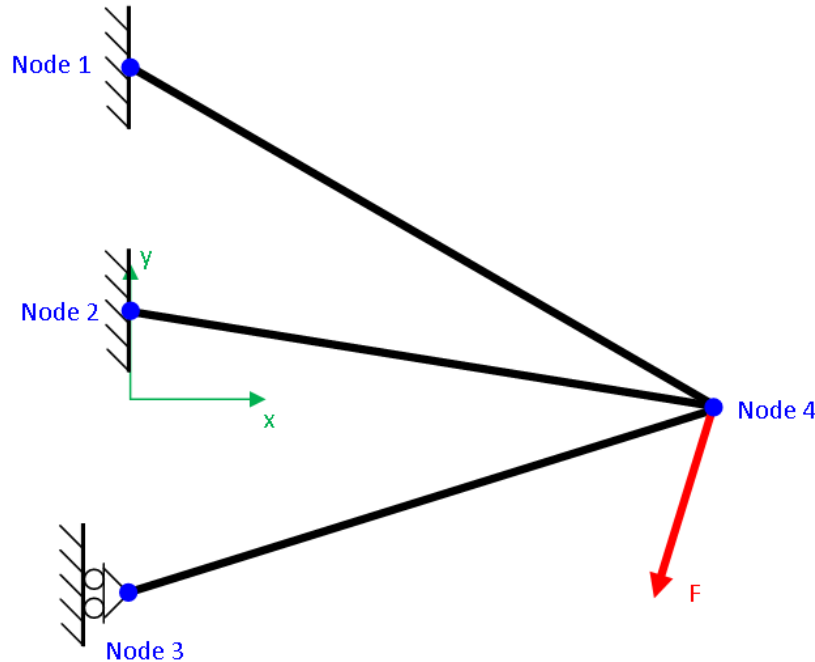


Figure 9. Homogenized representation of the global (system) level model.

This is a very simple model and can easily be optimized in HEEDS with the use of the Abaqus portals to tag the input file. The only thing that may not be obvious about this problem statement is the fact that the thickness cannot have a value greater than the radius. This can be avoided using a

number of methods; here, three new ratio variables were introduced, and the thickness variables became dependent variables, much as was done with the *ballRad* variable and the introduction of *BallScale* (see Table 2 in Section 2.2 for a review).

$$\begin{array}{llll}
 \text{Objectives:} & \text{Minimize } U_y^{(4)} & & \\
 \text{Constraints:} & \text{Volume} < 112 \times 10^9 & & \\
 \text{Design Variables:} & 1 \leq R^{(i)} \leq 750 & i \in \{1, 2, 3\} & \\
 & t^{(i)} = R^{(i)} \cdot \varepsilon^{(i)} & i \in \{1, 2, 3\} & \\
 & 0.1 \leq \varepsilon^{(i)} \leq 0.9 & i \in \{1, 2, 3\} & \quad (8)
 \end{array}$$

3.3.1 “Individual Only” Subsystem (Micro) Level Constraining Method

Several micro level optimization statements were tested during this exploration. However, every study had the same objective, design variables, and subsystem (micro) model (Figure 10). The only difference between these studies were the constraints. One constraining method was introduced in the previous section, and is referred to as the “Individual Only” method but is discussed in more detail here. The problem statement for this method is given in Equations (9), and was the first method to be explored.

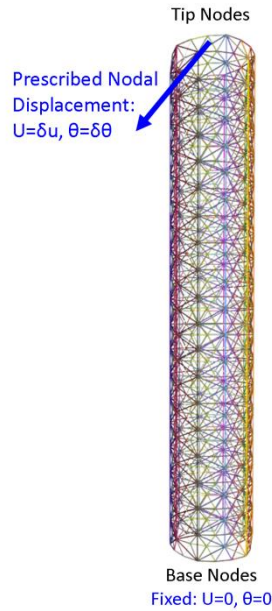


Figure 10. Micro (subsystem) level Finite Element model of a single member of the Three Member Structure problem

$$\begin{array}{ll}
\text{Objectives:} & \text{Minimize Volume} \\
\text{Constraints:} & \Lambda_j \geq \tilde{\Lambda}_j \quad j \in \{1, 2, \dots, 6\} \\
\text{Design Variables:} & \text{(See Table 2 in Section 2.2)}
\end{array} \quad (9)$$

The problem with this constraining method is that it yielded only infeasible designs. This is because the constraints are extremely restrictive: a design that would improve the feasibility of one constraint would likely be more infeasible for at least one other constraint. Just because the results are slightly infeasible does not necessarily mean they are bad results, though.

3.3.2 “Magnitude Only” Micro Level Constraining Method

In an attempt to find more feasible solutions, a different method of constraining was used. Rather than constraining each individual component of the response, as in Equations (9), only the total reaction force magnitude ($\tilde{\Lambda}_F$) and the total reaction moment ($\tilde{\Lambda}_M$) were constrained. The “total reaction force magnitude” is the square root of the sum of the squares of the first three reaction components. Similarly, the “total reaction moment magnitude” refers to the square root of the sum of the squares of the last three reaction components. These are shown in Equations (10) and (11), followed by the optimization problem statement for this method.

$$\tilde{\Lambda}_F = \sqrt{\sum_{j=1}^3 \tilde{\Lambda}_j^2} \quad (10)$$

$$\tilde{\Lambda}_M = \sqrt{\sum_{j=4}^6 \tilde{\Lambda}_j^2} \quad (11)$$

$$\begin{array}{ll}
\text{Objectives:} & \text{Minimize Volume} \\
\text{Constraints:} & \Lambda_F \geq \tilde{\Lambda}_F \\
& \Lambda_M \geq \tilde{\Lambda}_M \\
\text{Design Variables:} & \text{(See Table 2 in Section 2.2)}
\end{array} \quad (12)$$

While this “Magnitude Only” method accomplished the task of producing more feasible results, and greatly improved the solution found, it also had some undesirable characteristics. Specifically, because it does not directly reference any reaction component individually, it is able to

yield an optimal solution with reactions of the correct magnitude but in completely incorrect directions. This extreme deviation from the macro level model's reactions could make it difficult to predict the global model's behavior when reassembled. For a simple model such as this, it is not a great concern, but for larger, more complex problems, or more indeterminate problems, this could become a significant issue.

3.3.3 “Combined” Micro Level Constraining Method

Finally, the “Combined” method of constraining was analyzed. As the name suggests, this method combines the “Individual Only” constraints method in Equations (9) with the “Magnitude Only” constraints in Equations (12), but with an important modification. If a reaction had a value of zero, it was left out of the optimization problem. This is justifiable due the addition of the magnitude constraints, which can restrict the “zero” constraints to a small number based on the feasibility of the non-zero reaction component constraints. This problem formulation is shown in Equations (13), below.

$$\begin{array}{ll}
 \text{Objectives:} & \text{Minimize Volume} \\
 \text{Constraints:} & \Lambda_j \geq \tilde{\Lambda}_j \quad j \in \{1, 2, \dots, 6\} \text{ s.t. } \tilde{\Lambda}_j \neq 0 \\
 & \Lambda_F \geq \tilde{\Lambda}_F \\
 & \Lambda_M \geq \tilde{\Lambda}_M \\
 \text{Design Variables:} & \text{(See Table 2 in Section 2.2)} \quad \quad \quad \textbf{(13)}
 \end{array}$$

The magnitude constraints in this formulation acted as a sort of filter, reducing the amount of influence that each individual constraint had on the problem. While, in general, increasing the number of constraints in an optimization problem will increase the complexity and therefore decrease the number of feasible solutions found, the addition of the magnitude constraints to this problem did not have this general behavior. This is because the complexity was not increased by the addition of these constraints: if all of the individual constraints are satisfied, then the magnitude constraints must be satisfied as well, however, if some but not all of the individual constraints are

satisfied, then the magnitude constraint may or may not increase the severity of the penalty, depending on the amount of constraint violation. In this way, it guides the search towards more favorable designs. In fact, the “Combined” formulation yield approximately the same number of feasible solutions as the “Magnitude Only” formulation, with the advantage that the physical behavior of the recombined structure was more predictable.

3.3.4 Two-Step Serial Optimization Process Validation

The boundary conditions for the Three-Member Structure problem were not chosen arbitrarily. The free DoFs of node 3 were purposefully included in this model. Not only does this boundary condition ensure that the method is capable of handling a variety of configurations, it also plays a role in validating the results. The validity of a design can be intuitively determined by considering the fact that node 3 is a sliding pin joint. This means it is not capable of having a reaction opposite the direction of deflection, and therefore plays only a very small role in the determination of the deflection response. On the other hand, it plays an equal role in determining the volume of the structure. Accordingly, it is expected that the volume of this member would be as small as possible. If an “optimal” solution has design variables for this member that are towards the upper bounds, it is possible that this solution is not global optimal solution, but rather the study has found a local optima instead.

To further validate the results, a single level multi-objective optimization problem was set up using the global model. This problem attempted to minimize both the deflection and the total volume of the structure, as shown in Equations (14), where *ballRad* and *reinforcementDepth* are defined for each member as they are in Table 2 of Section 2.2. As these are competing objectives, a Pareto front developed over the course of the study. This Pareto front contains the set of optimal solutions to the global structure, given the allowable number of evaluations. With this set of optimal solutions to compare with, the performance of each constraining method could be determined. “Performance”, in

this context, does not refer to the use of the “performance function” shown in Equation (3). In this case, no function was set up to determine the performance, but rather the qualitative performance of each method was determined from the location of a method’s optimal solution (in the response space) relative to the Pareto front, and comparing it to the other methods.

$$\begin{array}{ll}
\text{Objectives:} & \text{Minimize Volume} \\
& \text{Minimize } U_y^{(Tip)} \\
\text{Constraints:} & \text{Volume} \leq 7.5 \times 10^9 \\
& U_y^{(Tip)} \leq 15 \\
\text{Design Variables:} & 1 \leq ringRad^{(i)} \leq 750 \quad i \in \{1, 2, 3\} \\
& 2 \leq numCells^{(i)} \leq 30 \quad i \in \{1, 2, 3\} \\
& 5 \leq numRepeats^{(i)} \leq 50 \quad i \in \{1, 2, 3\} \\
& 0.1 \leq BallScale^{(i)} \leq 0.75 \quad i \in \{1, 2, 3\} \\
& 0.1 \leq ReinforcementGap^{(i)} \leq 0.9 \quad i \in \{1, 2, 3\} \\
& ballRad^{(i)} \quad i \in \{1, 2, 3\} \\
& reinforcementDepth^{(i)} \quad i \in \{1, 2, 3\}
\end{array} \quad (14)$$

A multi-objective variant of SHERPA was utilized to conduct the search. This variant defaults to 500 evaluations and an archive size of 20. Because the results of this optimization problem are to be used as a validation measure, a thorough search is desired. Due of the high dimensionality of the search space (having 15 independent design variables), a thorough search requires significantly more evaluations than 500, so 1,000 evaluations were requested. Furthermore, a densely populated Pareto front is desirable, to reduce the chances of empty spaces in the Pareto front. Accordingly, the archive size was increased to 52.

3.4 Three Member Structure Problem – Analysis and Results

The results of the studies outlined in Section 3.3 are presented in Table 8, below. In the first five rows of this table, three numbers are present in each cell. These numbers correlate to the corresponding parameter for the first member, second member, and third member respectively.

At first observations, this table appears to show that the “Magnitude Only” method is the most valid approach according to the intuition-based validity test described in the previous section, because it has the most regularly extreme variable values for the third member. However, by taking a look at the rendered designs (Figure 11 through Figure 13 on the next page) it becomes apparent that every solution is fairly similar, at least as far as the third member is concerned.

Table 8. Numerical tabulation of the Three Member Structure Two-Step Serial Optimization trial results

| Trial | Individual Only | Magnitude Only | Combined |
|-------------------------|------------------------|-----------------------|---------------------|
| ringRad | 335 | 394 | 443 |
| | 534 | 414 | 594 |
| | 32 | 1 | 5 |
| numCells | 5 | 2 | 2 |
| | 8 | 3 | 1 |
| | 9 | 1 | 5 |
| numRepeats | 2 | 1 | 9 |
| | 19 | 6 | 1 |
| | 1 | 1 | 19 |
| BallScale | 0.655 | 0.75 | 0.545 |
| | 0.23 | 0.53 | 0.545 |
| | 0.58 | 0.1 | 0.715 |
| ReinforcementGap | 0.22 | 0.27 | 0.295 |
| | 0.775 | 0.21 | 0.875 |
| | 0.25 | 0.795 | 0.64 |
| Total Volume | 1.236×10^9 | 6.291×10^8 | 4.867×10^8 |
| Deflection | 0.688 | 0.722 | 1.354 |

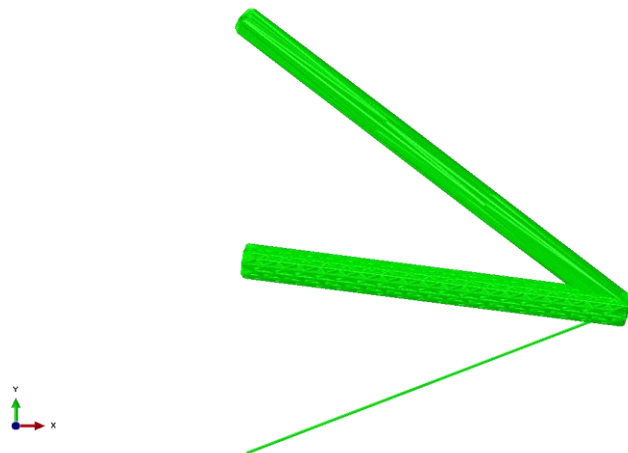


Figure 11. Abaqus rendering of the optimized model using the “Individual Only” technique

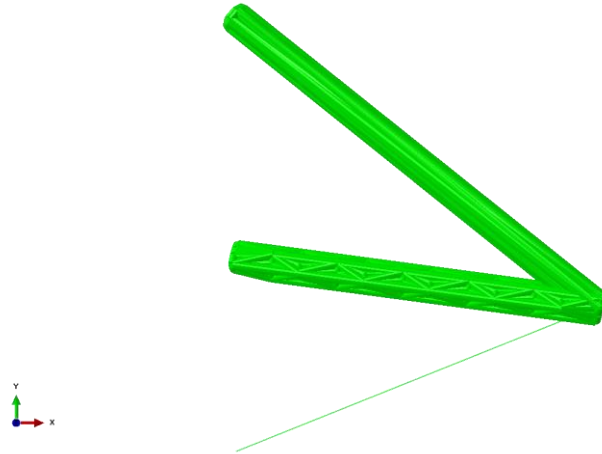


Figure 12. Abaqus rendering of the optimized model using the “Magnitude Only” technique

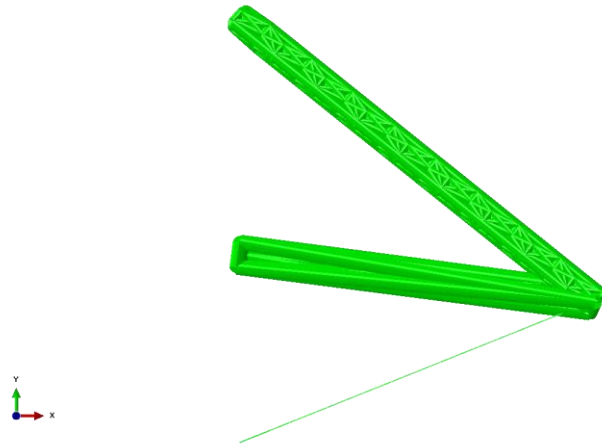


Figure 13. Abaqus rendering of the optimized model using the “Combined” technique

The next observation to be made comes from Figure 14, which shows the multi-objective validation study overlaid with the results from the three different constraint formulations. While the “Individual Only” method did not perform badly, it was outperformed by both the “Magnitude Only” and “Combined” methods. The “Magnitude Only” results seem to lie on the Pareto front, whereas the “Combined” method lies slightly behind the Pareto front. This shows that from a qualitative standpoint, the “Magnitude Only” technique performs better, but for the reasons outlined in Section 3.3.2, may not be as reliable as the “Combined” method, which also produced acceptable results.

Furthermore, because the objective of the subsystem level optimization problem was to minimize volume, quantitatively speaking, the “Combined” method clearly performed better.

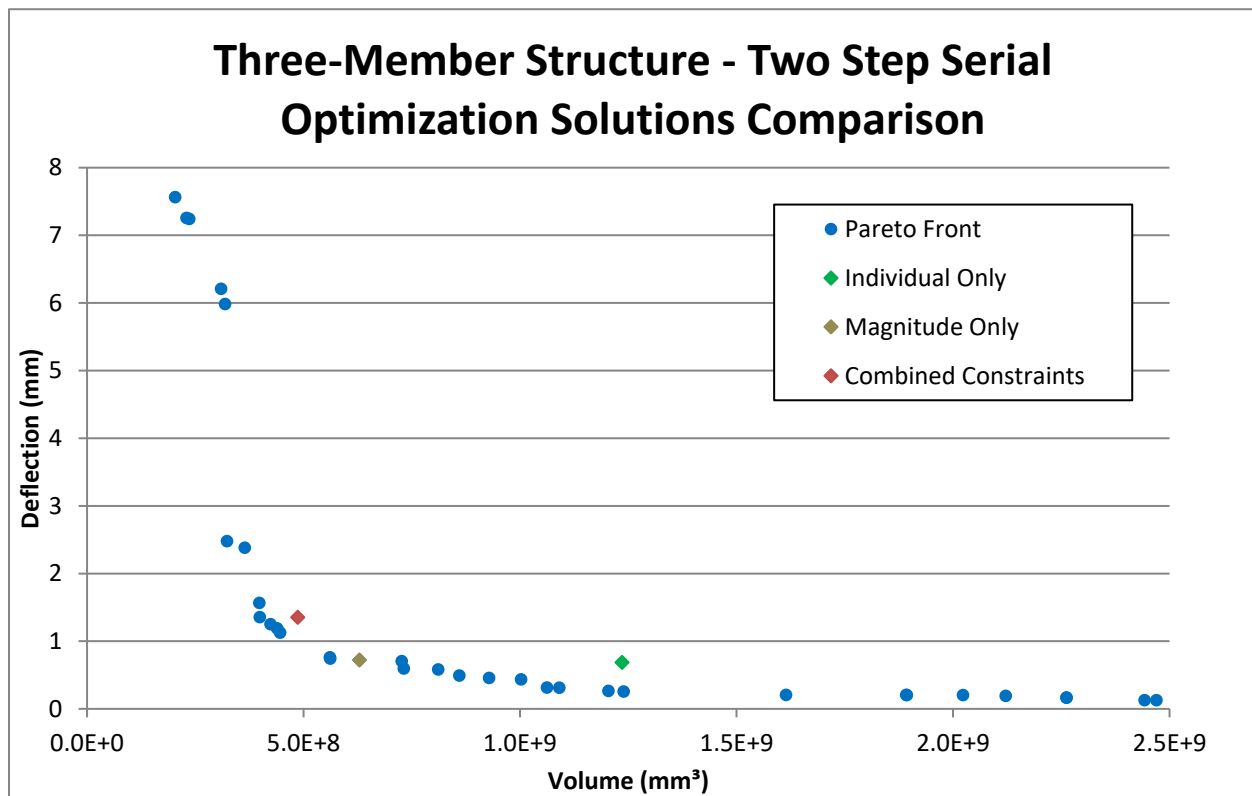


Figure 14 . Comparison of the optimal solution using the “Individual Only”, the “Magnitude Only”, and the “Combined Constraints” techniques to the Pareto Front which was generated from an All-At-Once optimization study.

CHAPTER 4: Multilevel Optimization

The process described in the previous chapter was shown to have reasonable results for all variations. It is important to keep in mind, however, that this is a very simple problem. For more complex problems with a higher degree of nonlinearity, it is likely that this process will not result in a good solution. This is because there is no feedback from the subsystem level back up to the system level. As the subsystem designs stray from the baseline, the reactions get outdated and may no longer be a good representation of the behavior of the global model. In an attempt to find a more generalized multilevel optimization process, a method was developed in which there is communication between the different levels. In order have a basis of comparison, the same example problem outlined in Chapter 2 will be examined using this multilevel method. This will allow the comparison of the methods to each other, and it will allow the Pareto front obtained by the multiobjective global study to be reused as a comparison to currently accepted practices.

4.1 Multilevel Optimization Process Flow

Like the Two-Step Serial process, the multilevel optimization process is also comprised of two phases. Unlike the Two-Step Serial process, the phases are repeated in the multilevel optimization process and communicate with each other. Also, in the current approach the system level (global) model is defined using LatticeMaker as opposed to using the annular beam model. The full process is shown in Figure 15 below. The two phases consist of a system (global) analysis (item 2 in Figure 15) and a subsystem (micro) level optimization process (item 4 in Figure 15).

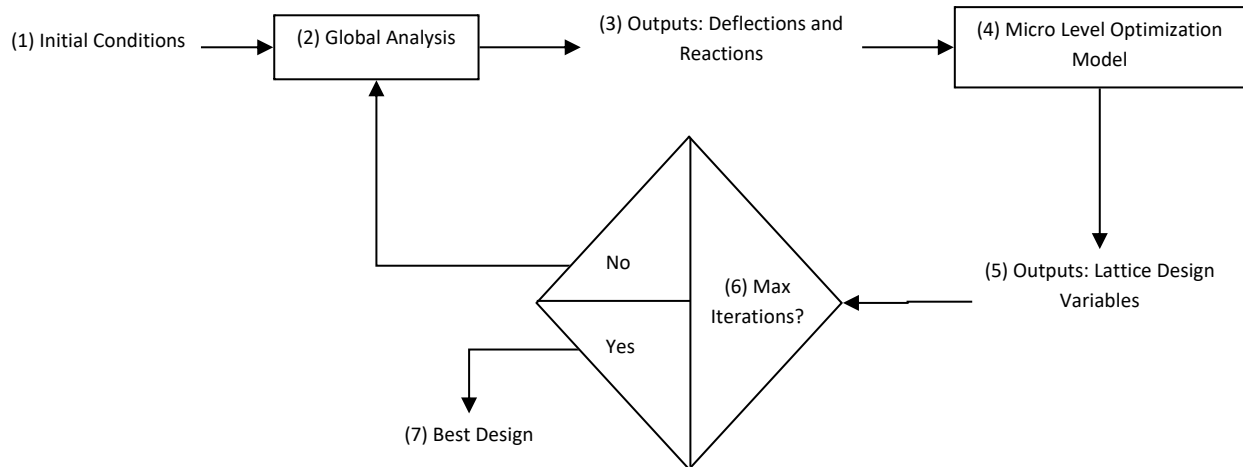


Figure 15. Multilevel Optimization Process Flow Diagram

The process starts with analyzing the initial conditions and baseline design variables in the global model. A single analysis is performed. The deflections and reactions are output from this model and fed into the micro level model as applied conditions and optimization constraints, respectively. SHERPA is used to optimize the micro level models and find the “optimal” design variables. These are then fed back into the global analysis for each member. The global analysis is again performed for the new design, and then the deflections and reactions are updated in the micro level analysis and the process repeats. In this way, the two levels communicate with each other and an optimal solution can be found after convergence of the global analysis.

Because automation of the communication between the levels has not yet been investigated, an Excel workbook was created to facilitate the transfer of results from one iteration to the next. The user would enter (via a copy-paste routine) the results from HEEDS into the workbook, which would then organize the information and arrange it in such a way that it could be easily copied and pasted back into the next HEEDS study. This was found to greatly increase the speed of and reduce the amount of error in the setup of iterations.

The first phase of the multilevel method is the system level (global) analysis of the design. This procedure is straight forward and easily automated in HEEDS. First, the LatticeMaker design variables are entered and then LatticeMaker is executed for each member, generating the required lattice geometry data. Then, LatticeModeler finds the geometry data and converts it into an Abaqus input file. Finally, Abaqus is executed and the deflection/reactions are output, for use in the second phase of the process, the micro level optimization.

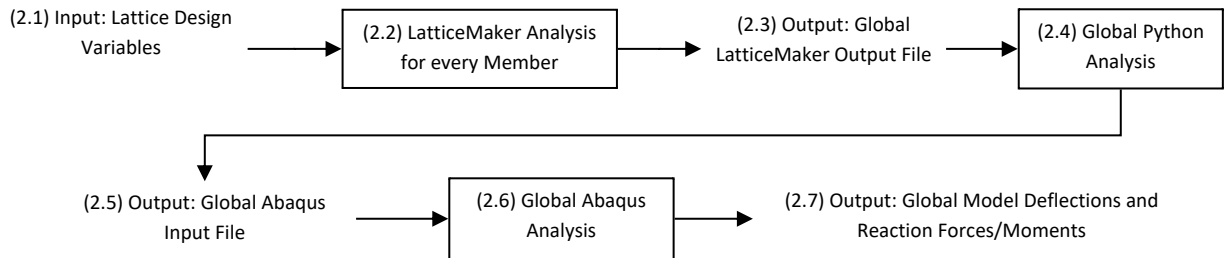


Figure 16. Multilevel Optimization – Phase 1 Flow Diagram

The second phase is the subsystem (micro) level optimization, and is performed independently for each member in question. As usual, this phase is much more intricate than the first phase, due to the optimization procedure. For each member, the baseline design for the given iteration must be analyzed to determine the micro level reactions. This step is only required if the member in question is not located at a boundary of the global model. This is because the internal reactions cannot easily be obtained from the global model analysis. For the current example, all members are located at the global model boundaries, so this step can be skipped because the reactions of the micro level must be identical to the reactions at the global level. The next step is to enter the reactions into HEEDS as optimization constraints. At this point, optimization of the analysis model can commence.

The micro level analysis model is quite similar to the global level analysis in that it consists of a LatticeMaker analysis, a LatticeModeler analysis, and an Abaqus Finite Element Analysis. The difference between the two models is that the micro model only uses a single member LatticeMaker analysis, and that the LatticeModeler analysis takes input from the global results. Furthermore,

whereas the global analysis is performed a single time per iteration, the micro level analysis is performed several times per iteration, as it is getting optimized for the given conditions.

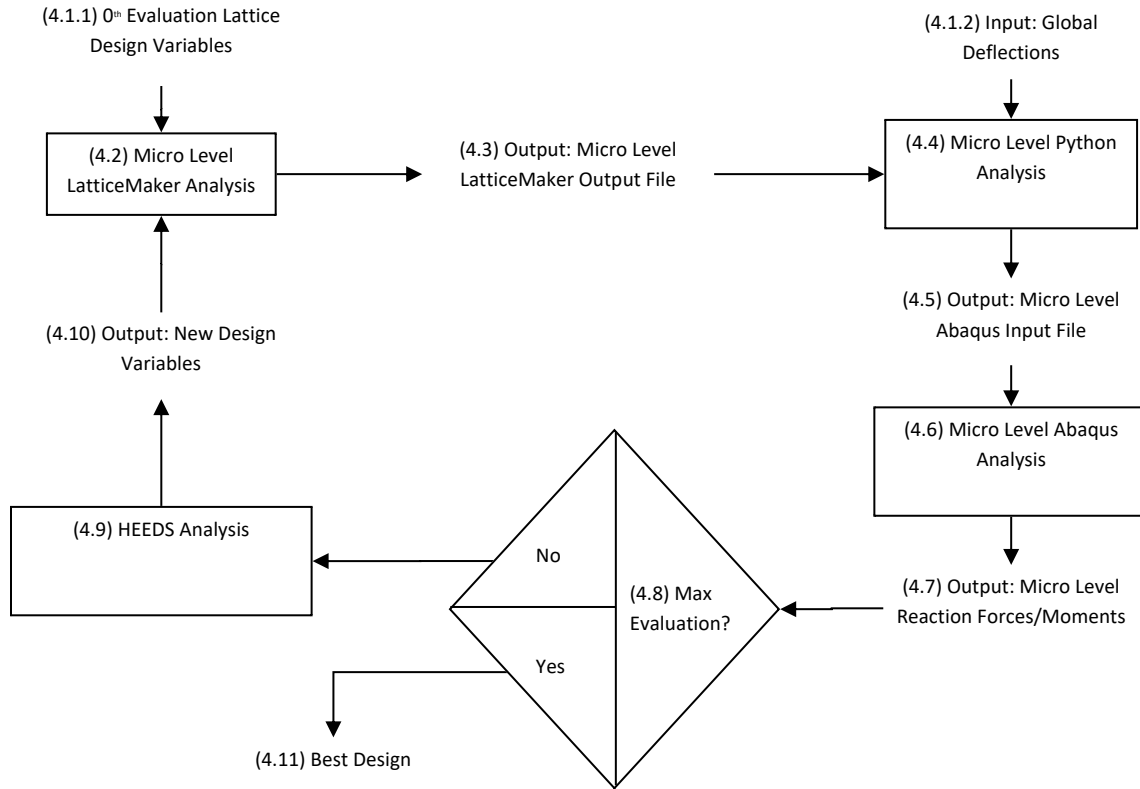


Figure 17. Multilevel Optimization - Phase 2 Flow Diagram

4.2 Problem Statement for Multilevel Optimization

To recap, the ultimate objective of the Three-Member Structure problem is to minimize the mass of the structure while maintaining a maximum tip deflection below a given value. In order to compare to the Pareto front that was previously generated, the global deflection constraint was set at a value of 15 millimeters.

| | | |
|-------------------|------------------------------|--------|
| Objectives: | Minimize Volume | (15) |
| Constraints: | $U_y^{(4)} < U_{Max}$ | |
| Design Variables: | (See Table 2 in Section 2.2) | |

4.2.1 Problem Statement Decomposition

The problem statement shown in Equations (15) was then decomposed into two related problem statements – one macro level problem statement, and one micro level problem statement. The goal of the decomposition is to accurately capture the objectives and constraints of the original problem statement, while reducing the complexity and thus reducing the number of evaluations required.

The macro level problem statement for the multilevel optimization method is a little bit different here than it was in the previous chapter. This is because there are no traditional variables that are getting changed at the macro level. Rather, the macro level is acting as more of a guiding check analysis, using the results of the micro level analysis as input variables. The NLP description resembles Equations (15), however, the design variables section is empty.

$$\begin{array}{ll} \text{Objectives:} & \text{Minimize Volume} \\ \text{Constraints:} & U_y^{(4)} < U_{Max} \\ \text{Design Variables:} & \text{None} \end{array} \quad (16)$$

The micro level problem statement is more traditional, having objectives, constraints, and design variables. The micro level objective is the same as the macro level objective – Minimize Volume. The constraints at the micro level, however, are not the same as the macro level. At the micro level, the ends of each lattice member are displaced by the amount found from the global analysis, and the reactions are constrained. The exact way that these reactions were constrained is the focus of this chapter and will be discussed in the next section. The design variables are the same LatticeMaker parameters as usual, and can be found in Table 2 in Section 2.2.

4.2.2 Reaction Constraint Methods

Because there are several reaction constraints in this optimization problem statement, it is possible that only a small number of feasible solutions will be found. As this is undesirable, a few methods were developed to increase the number of feasible solutions.

One method to increase the number of feasible solutions was to change the normalization scheme. The most common normalization scheme is to normalize the objective responses by the correlating baseline response and the constraints by the constraint boundaries. It is also common practice to normalize constraints with a limit of zero by unity. Stepping away from common practices, a normalization scheme was developed that attempts to mimic a real-world scenario.

In real world structures that are influenced by multiple loads of varying magnitudes, perturbations in the smaller loads tend to have less of an effect on the behavior of the structure than the effects of perturbations in the larger loads. To capture this behavior, the total magnitude of the reaction force was calculated, as well as the magnitude of the reaction moment, as shown in Equations (10) and (11) respectively. Then, all reaction force components are normalized by the magnitude of the total reaction force and all reaction moment components are normalized by the magnitude of the total reaction moment.

This normalization scheme encourages infeasible designs to prefer infeasibilities in the constraints corresponding to the low magnitude reactions. For instance, if a constraint with an extremely small limit is violated, and the constraint is normalized by its own limit, it will be divided by an extremely small number, resulting in an extremely large penalty. However, if all the constraints are normalized by the same value (which is representative of the impact each constraint has on the system), then the same constraint with the same constraint violation will have a much smaller penalty. An issue with this method is that if a reaction's component is found to be exactly zero, the optimization results will only produce infeasible designs, regardless of the tolerance. To overcome

this, a normalization technique was used in which the zero reaction components ignored the constraint equation given in Equation (7), and instead used a fixed tolerance of 1e-6. An alternative approach (not explored here) would be to set the tolerance equal to a small percentage of the corresponding (force or moment) magnitude.

As previously mentioned, the reactions at the lattice ends were used as constraints at the micro level. The baseline reactions for each iteration were determined, and then constraints were set up to bound the reactions within a certain range of the baseline. A parameter, λ , was used to constrain the reactions, as shown in Equations (17), below

$$\begin{array}{ll}
 \text{Objectives:} & \text{Minimize Volume} \\
 \text{Constraints:} & (1 - \lambda)\tilde{\Lambda}_j < \Lambda_j < (1 + \lambda)\tilde{\Lambda}_j \quad j \in \{1, 2, \dots, 6\} \\
 \text{Design Variables:} & \text{(Problem Dependent)}
 \end{array} \tag{ 17 }$$

where λ is a tolerance (similar to the “move limit” in [19] and [20]). This problem statement was used with a 5% tolerance, but yielded no feasible solutions. Accordingly, the value was increased to 10%, in which a few feasible solutions were found. This constraining method is referred to as the “Individual Only” constraint method for multilevel optimization.

Still, only a few feasible solutions were found, so a new normalization method was employed in which the zero constraints were ignored entirely. It was found that this would tend to produce more favorable results, however, this favorability could easily be eliminated by rotating the basis so that no zero reactions occur. Because of this dependency on orientation, the fixed normalization method for zero reactions was deemed more robust.

To make the non-zero constraint normalization method more robust, the magnitude constraints were added to the problem statement as shown in Equations (18), below. The addition of the magnitude constraints had a couple of important implications. First, they made up for the missing zero constraints by making sure the magnitude was conforming with the system level

responses. That, coupled with the inclusion of the non-zero constraints, ensured that the zero constraints must be met (or be close to being met). Secondly, these acted as a sort of filter, making designs that violate the magnitude constraints severely penalized (individual constraints must also be violated in addition to the magnitude constraints) whereas designs that only violate the individual reaction constraints were less severely penalized. This method of constraining is referred to as the “Combined Constraint” method.

$$\begin{array}{ll}
\text{Objectives:} & \text{Minimize Volume} \\
\text{Constraints:} & (1 - \lambda)\tilde{\Lambda}_j < \Lambda_j < (1 + \lambda)\tilde{\Lambda}_j \quad j \in \{1, 2, \dots, 6\} \text{ For } \Lambda_j \neq 0 \\
& (1 - \lambda)\tilde{\Lambda}_F < \Lambda_F < (1 + \lambda)\tilde{\Lambda}_F \\
& (1 - \lambda)\tilde{\Lambda}_M < \Lambda_M < (1 + \lambda)\tilde{\Lambda}_M \\
\text{Design Variables:} & (\text{See Table 2 in Section 2.2})
\end{array} \tag{18}$$

4.3 Multilevel Optimization Results and Validation

Again using the validation Pareto plot from the previous chapter, the two constraining methods can be compared and validated. The same baseline design was used for both studies. The data series in Figure 18 can be assumed chronological starting from the right and working to the left each iteration, as shown by the labels above each Individual Constraint data point. The data labels are included here for clarity, but will not be included in future figures, though the same assumption can be made unless otherwise noted. Knowing that the results monotonically improved, it can be observed that the Combined Constraints method converged to the Pareto Front at a much quicker rate – the results of the first iteration of 10% combined constraints method are in the same vicinity as the results of the 4th iteration of 10% Individual constraints method. Furthermore, after 10 iterations, the combined constraints method appears to be converging to a design that is located on the Pareto front whereas the individual constraints method appears to be converging to a local optima. This suggests that the combined constraint methods allows for more mobility and design exploration due to the more relaxed constraint scheme.

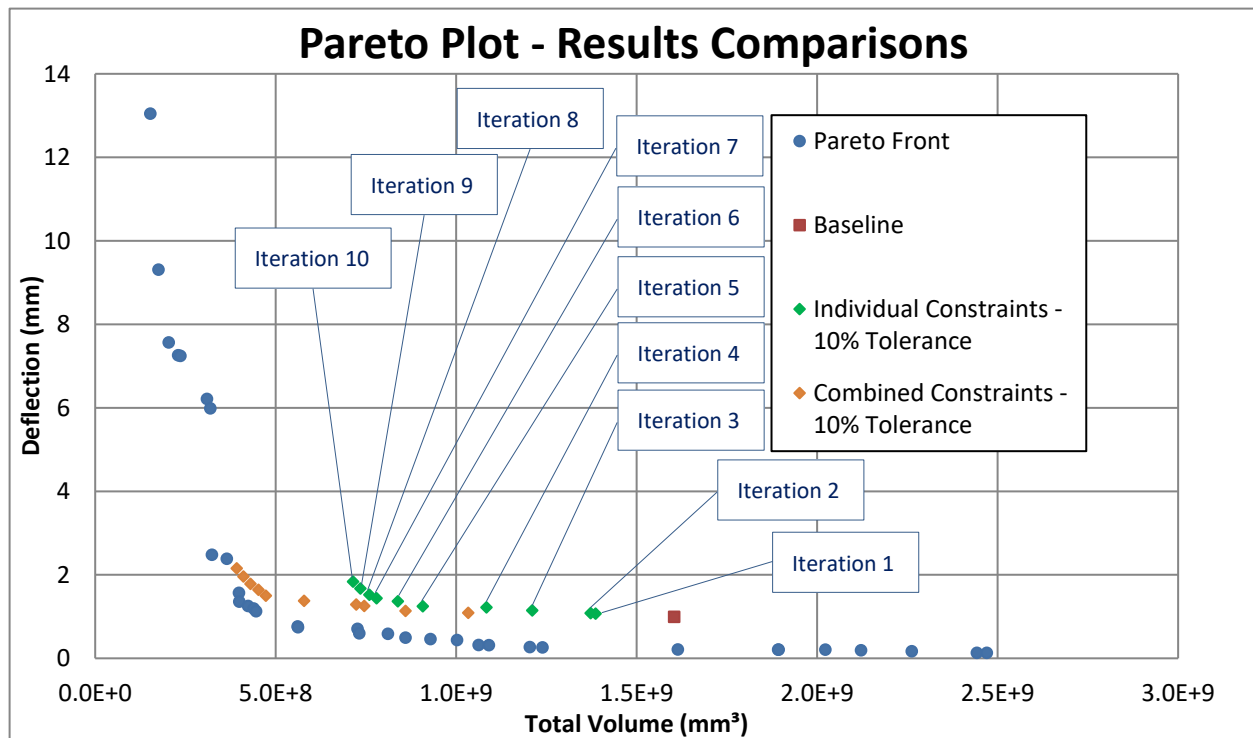


Figure 18. Comparison of the “Individual Constraints” and “Combined Constraints” multilevel optimization techniques to the Pareto Front found from an All-At-Once optimization study.

CHAPTER 5: Optimization of a Transmission Tower

In the previous chapters, a simple Three-Member structure was analyzed via a simple Static Structural Finite Element Analysis. Many times, however, an optimization problem cannot be simplified to a single static analysis. More often, multiple physical disciplines must be considered simultaneously. This is especially true for Additive Manufacturing, in which structural parts are frequently developed from the bonding of cross-sectional layers using thermal reactions. To optimize parts like this, a multi-physics multilevel optimization technique should be employed. Similar to additively manufactured parts, transmission towers will often require a multi-physics analysis approach. As these can easily be modeled using simple beam elements, this will be the focus of the next example.

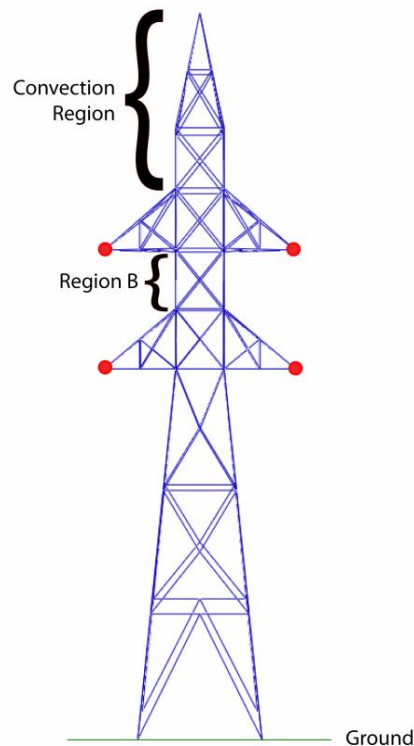


Figure 19. A diagram of the Transmission Tower model showing the convection region, the location of point loads and fluxes, and Region B (the SoS region).

5.1 Transmission Tower Problem Statement

As the geometry in the following analyses, a two-tiered transmission tower will be analyzed as shown in Figure 19, below. The base of the structure has several boundary conditions applied; the spatial degrees of freedom are all fixed to zero displacement and zero rotation, and a constant “ground temperature” is assumed and applied to the nodes. To simulate support wires, a point load is applied to horizontal members indicated with a red dot in Figure 18 in the negative z direction (towards the “ground”). Because transmission towers have high-voltage current running through them, a heat flux was also applied at the same locations. Convection conditions were applied to the transmission tower members above the second tier, where air velocities are highest. The environmental and material properties were held constant throughout this study, and are given in Table 9, below.

Table 9. Various material properties, environmental conditions, and load cases used in the Transmission Tower analysis.

| Material Properties | | Environmental Conditions | |
|-------------------------------|------------------------|--------------------------|-------|
| Reference Temperature | 20°C | Ambient Temperature | 20°C |
| Young’s Modulus | 190 GPa | Ground Temperature | 40°C |
| Poisson’s Ratio | 0.3 | Convection Coefficient | 9e-2 |
| Density | 8055 kg/m ³ | Load Cases | |
| Thermal Expansion Coefficient | 17.3e-6 m/(m*K) | Thermal Flux | 500 |
| Thermal Conductivity | 15.1e2 W/(m*K) | Static Force | 25 kN |
| Specific Heat | 480e2 J/(kg*K) | | |

As is common in design problems, optimization will be focused on only a small region, namely between the two tiers. The members in this region will be replaced by lattice structures, while all other members are considered to be conventional beam (non-lattice) members. There are a total of 16 members in this region, so to further reduce the number of evaluations, the rotational symmetry

of the region will be utilized. To that end, the members on only one side of the structure will be analyzed and optimized at the micro level, and the results will be applied to all of their doppelgänger members on the other sides of the structure. While the micro level boundary conditions for the doppelgänger members will be slightly different than the boundary conditions for the considered (optimized) members, they should be similar. Several methods of handling the difference between the doppelgänger's boundary conditions will be examined. The objective is to reduce the mass (Volume) of the structure while constraining the maximum temperature and the maximum stress in the region of focus to be below certain threshold values. This is shown in Equations (19). Unlike in the previous studies, the constraints shown in these equations will be applied at both the micro level and the global level.

$$\begin{array}{ll}
\text{Objectives:} & \text{Minimize Volume} \\
\text{Constraints:} & \sigma_{\text{RegionB}} \leq \sigma_{\text{Limit}} \\
& \Theta_{\text{RegionB}} \leq \Theta_{\text{Limit}} \\
\text{Design Variables:} & \text{(See Table 2 in Section 2.2)}
\end{array} \quad (19)$$

5.2 Multilevel Optimization Process Flow for Multiphysics Problems

The process flow for the Multiphysical Multilevel Optimization method is very similar to the process flow for the single physics version, though slightly more complicated due to the fact that it involves running analyses in parallel. It starts with the initialization of a baseline design at the global level. LatticeMaker is used to generate the lattice structure data for all lattice members, and then LatticeModeler is used to inject that data into the global model. Whereas, in previous studies, only one Abaqus input file was required, two files are now required – one for a static structural analysis and one for a static thermal analysis. Accordingly, two instances of LatticeModeler are run in parallel, one for each analysis type. Once both input files have been generated, two instances of Abaqus are run in parallel. The results from each analysis are collected in HEEDS and output as a list. This list can then be copied and pasted into an Excel spreadsheet for organization and storage.

The displacements and temperatures found for each node in the global analysis are used as prescribed conditions for the micro-level analyses. The analyses are each run once, in order to find the support reactions (forces, moments, and flux). These reactions are then used as constraints in the micro-level optimization, along with the Stress and Temperature constraint, that is passed down from the global analysis. Once a micro-level optimal solution has been found for each member, the designs are entered into the global analysis, and the process repeats. A total of 12 iterations are performed.

5.3 Multiphysics Problem Statement Decomposition for the Multilevel

Optimization Method

Similar to the previous chapter, the macro level analyses were not guided by an optimization loop at that level. Rather, the analyses acted as checking analyses that were used to update the actual nodal displacement (and temperature in this case) at the subsystem (micro). Accordingly, no variables were changed at the macro level. The NLP problem description resembles Equations (19), however, the design variables section is empty.

$$\begin{array}{ll}
 \text{Objectives:} & \text{Minimize Volume} \\
 \text{Constraints:} & \sigma_{\text{RegionB}} \leq \sigma_{\text{Limit}} \\
 & \Theta_{\text{RegionB}} \leq \Theta_{\text{Limit}} \\
 \text{Design Variables:} & \text{(See Table 2 in Section 2.2)}
 \end{array} \quad (19)$$

Using the information gathered from the results found in the previous chapter, the “Combined Constraint” method was chosen to analyze the multi-physics problem at the micro level. While the system level problem statement remains very similar to the system level statement in the previous chapter, a major modification was made to the subsystem level problem statement. In this study, the constraints from the system level were also applied to all members at the subsystem level. Another difference between this study and the previous one is that this problem has a higher

dimensionality. Because of this, the subsystem level studies were always found to have nonzero reactions for all degrees of freedom. This was an unexpected development, but did not seem to severely impact the study in any way. Thus, the micro level problem statement is given by Equations (20), below.

$$\begin{array}{ll}
\text{Objectives:} & \text{Minimize Volume} \\
\text{Constraints:} & \sigma_{\text{RegionB}} \leq \sigma_{\text{Limit}} \\
& \Theta_{\text{RegionB}} \leq \Theta_{\text{Limit}} \\
& (1 - \lambda)\tilde{\Lambda}_j < \Lambda_j < (1 + \lambda)\tilde{\Lambda}_j \quad j \in \{1, 2, \dots, 7\} \\
& (1 - \lambda)\tilde{\Lambda}_F < \Lambda_F < (1 + \lambda)\tilde{\Lambda}_F \\
& (1 - \lambda)\tilde{\Lambda}_M < \Lambda_M < (1 + \lambda)\tilde{\Lambda}_M \\
\text{Design Variables:} & \text{(See Table 2 in Section 2.2)}
\end{array} \tag{ 20 }$$

In the problem statement above, notice that the index for the first constraint ranges from one to seven. The introduction of the seventh degree of freedom is to account for the thermal impacts on the study. It is also important to understand the use of doppelgänger members in this study as they play an important role in the interpretation and behavior of the results.

5.4 Analysis and Results

As shown in Figure 20, below, the first set of studies that were analyzed all turned out to have inactive global constraints (Stress and Temperature), thus, this can be thought of as an unconstrained set of studies. These studies are the same in all aspects, except for the fact that they have different baseline designs. As would be expected from an unconstrained study, the results improved each iteration, as is shown in Figure 21 on the next page. Because Figure 21 shows the volume decreasing with each iteration for each study, the progression of a study can be followed in Figure 20 by following a dataset from right to left.

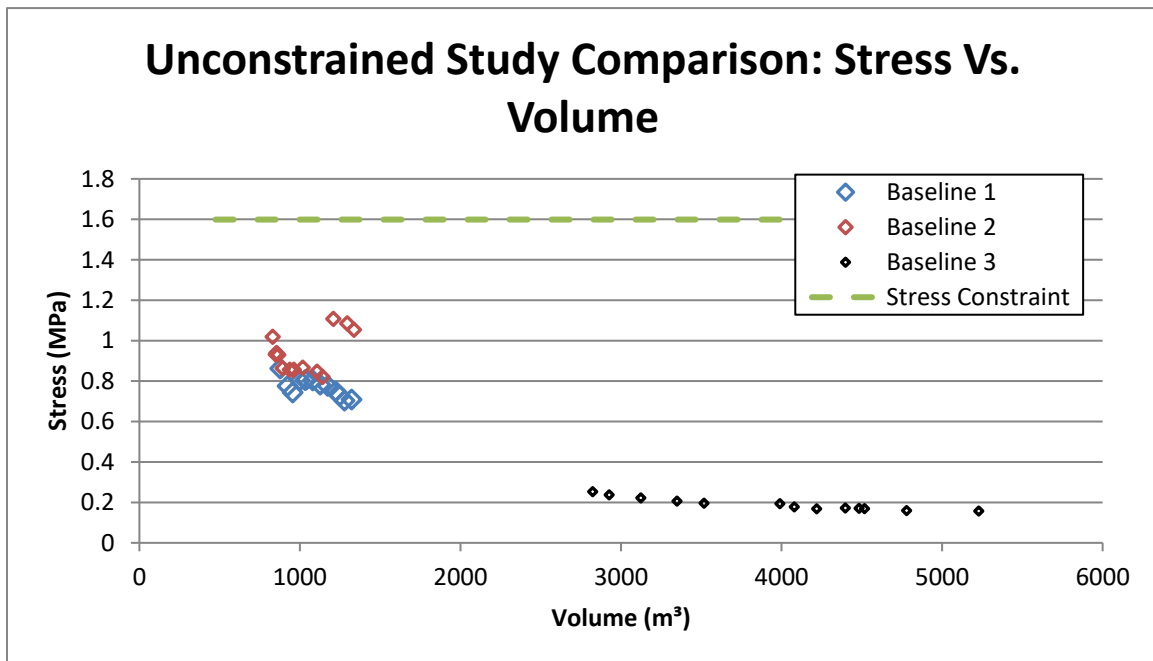


Figure 20. Relationship between Volume and Stress for the unconstrained Transmission Tower problem.

An unexpected result (and possible future improvement) of this study setup is shown in Figure 22, which shows the normalized progression of temperature. As would be expected, the temperature increases with iteration, however, the scale of the Normalized Temperature axis shows that the temperature changes are extremely minor. Not shown in this figure is the temperature constraint. This was intentionally excluded from the figure because it would require a much more significant scale to be utilized, which would significantly detract from the message that the figure is

showing, which is that the temperature is indeed increasing. Because this increase is so extremely minor and the temperature is so far from the constraint, the temperature constraint can be ignored for the sake of avoiding redundancies in the scope of this text.

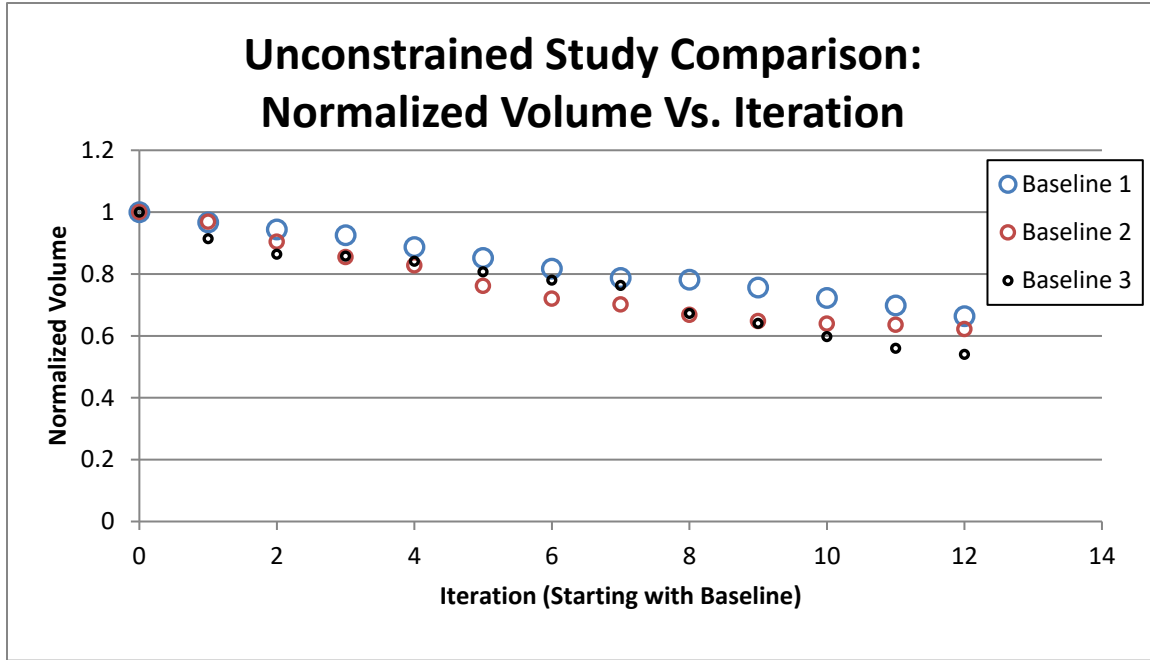


Figure 21. Relationship between Iteration and Normalized Volume for the unconstrained Transmission Tower problem.

While the results shown in Figure 20 and Figure 21 above depict an improving solution, and thus suggest the multilevel optimization process is working, the results aren't all that useful for illustrating the multilevel optimization strategy. The reason for this is that most practical optimization problems will have an active constraint. In an attempt to gain more useful data, a second set of studies were performed that were designed to result in active global constraints. For this set of studies, the baseline designs were the same, but the stress constraint was significantly lowered.

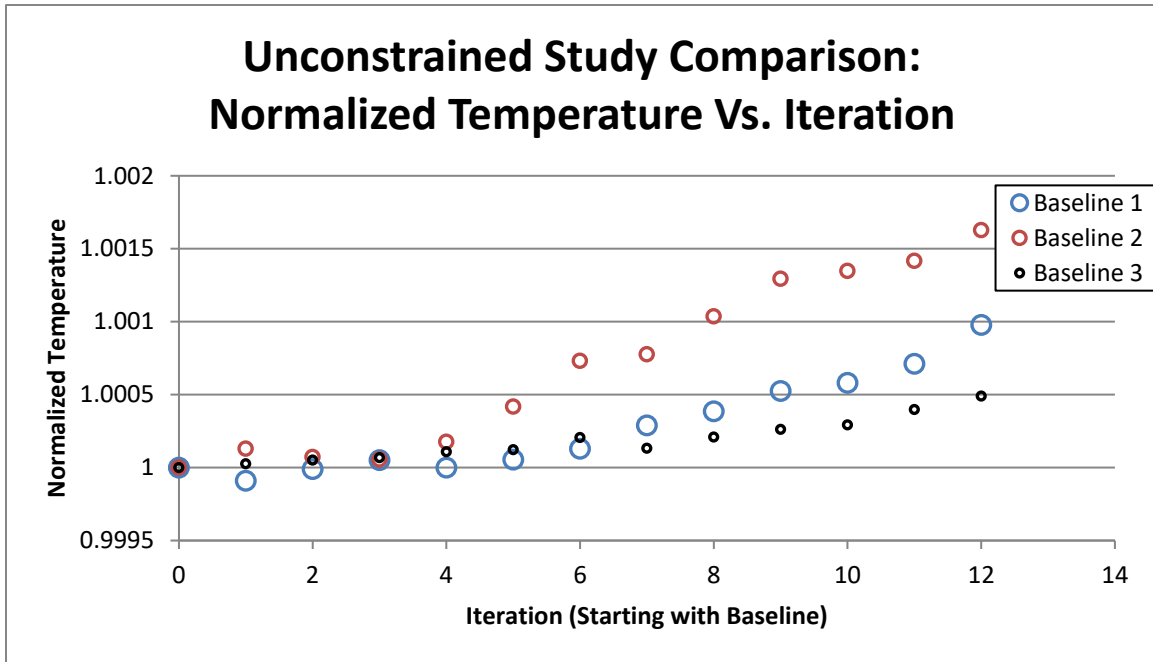


Figure 22. Relationship between Iteration and Normalized Temperature for the unconstrained Transmission Tower Problem.

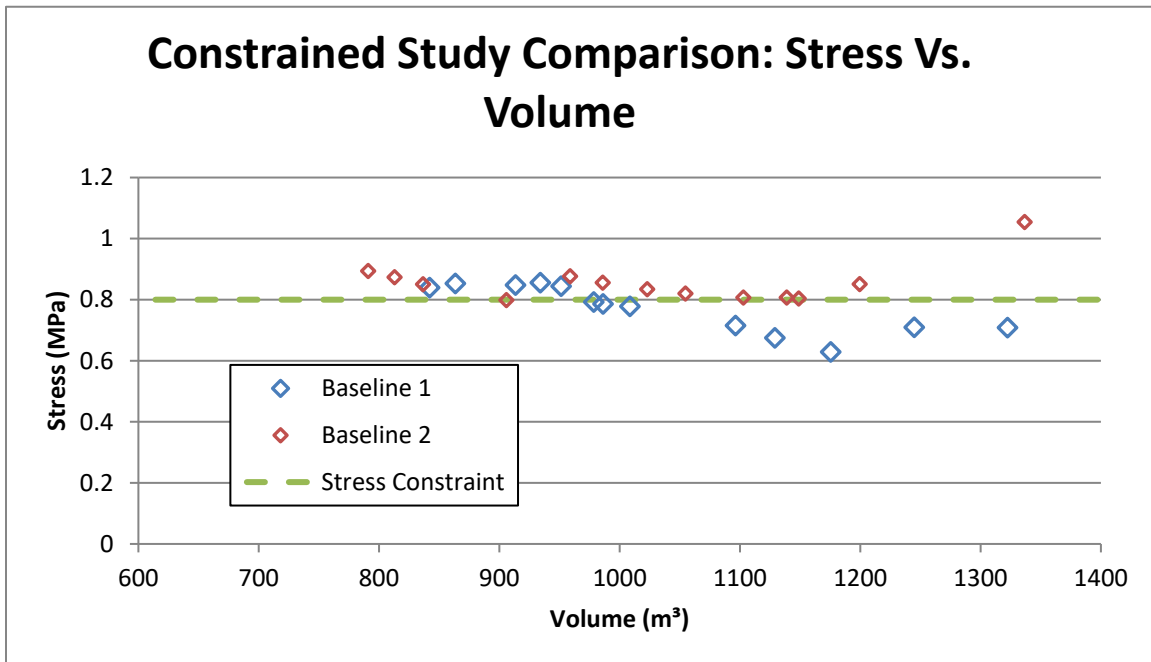


Figure 23. Relationship between Volume and Stress for the constrained Transmission Tower Problem.

The results in Figure 23 can again be followed chronologically from right to left. This shows that this multilevel optimization method is not very reliable for optimization problems like this - both

studies end on globally infeasible solutions after 12 iterations. 12 iterations may not be enough to fully judge this method. Baseline 3 was intentionally left out from this study as it would not have found a solution that had an active constraint, as is shown in Figure 20 and Figure 21. Figure 24 suggests that this method may find solutions that traverse the constraint boundary in a damped oscillatory fashion such that it would eventually converge on a feasible solution, but this has neither been demonstrated nor proved.

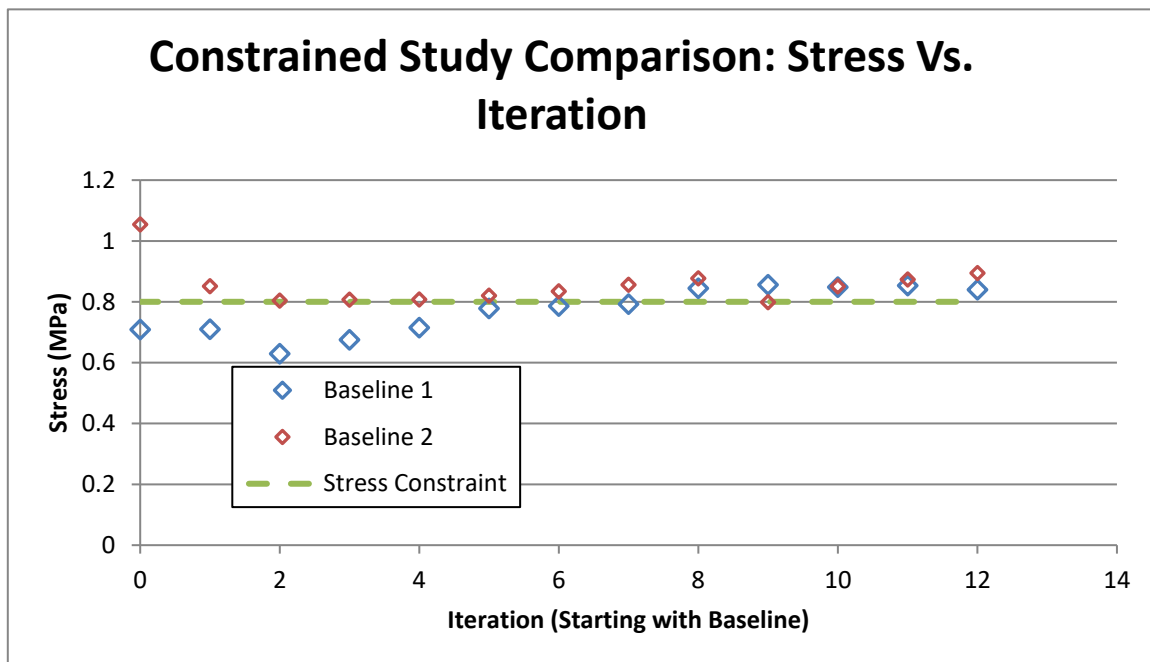


Figure 24. Relationship between Iteration and Stress for the constrained Transmission Tower Problem.

One question that must be asked is “why do the feasible designs at the micro level result in infeasible solutions at the global level?” One possibility is that changes in the micro level design result in different solutions (and boundary reactions) at the macro level, and this interaction needs multiple iterations to settle out. Another possible answer to this question is the use of doppelgänger members. The doppelgänger members in the global level have different boundary conditions than the original members that were analyzed and optimized at the micro level. This theory was confirmed when looking at the location of maximum stress in the global structure. The maximum stress was usually

found to be on a doppelgänger member. In an attempt to develop a better multilevel optimization process for this problem, perhaps there is a way to predict the difference in maximum stress between the analysis levels and account for that up front.

5.5 Multilevel Optimization with Adaptive Tolerance Control

As can be seen in the results provided in the previous section, the behavior of the doppelgänger members at the macro level can be significantly different than that predicted by the micro-analyzed members. A small amount of difference was expected, but the amount shown in the previous study is unacceptably unpredictable for use in an optimization study with active global constraints. These differences could very easily be the difference between a feasible micro-level analysis, and an infeasible global analysis using the same design variable values, as was shown in the previous section. A method was developed that changed the amount of tolerance given to the reaction forces and moments based on the relative location of the global constraints. The reasoning behind this is that with a tighter reaction tolerance, the result of the micro-level iteration will find a micro-analyzed member that behaves more similarly to the iteration's baseline and thus the deflection found in the global analysis will be more similar for all nodes. This means that the difference between the results and the baseline for the doppelgänger members should be reduced as well, making their behavior more predictable. To accomplish this, the tolerance, λ , is no longer treated as a constant, but rather a function of relative response location. The greater the difference between a design's response value and the global constraint value, the larger the value of λ . This allows the design to change more significantly when the constraints are very inactive and thus reduce the mass (improve the objectives) more rapidly. On the other hand, as the design approaches a global constraint, it needs to behave more predictably, so that an infeasible design is not found at the global level. Accordingly, the tolerance is tightened (value of λ decreased) so that the reactions cannot change as drastically. This method of controlling the tolerance is referred to as Adaptive Tolerance Control (ATC).

The exact form of ATC has gone through several iterations. Only the most recent iterations will be presented here. It works by having an upper and a lower tolerance range defined which will attempt to bound the possible tolerance values. Let these values be λ_{Max} and λ_{Min} respectively. Let the total number of constrained responses be n (this does not include responses used as objectives, i.e. volume) and the total number of allotted iterations be N . Furthermore, let i be an index variable representing the current response and j be an index variable representing the current iteration. Thus, i can take any integer value from 1 to n and j can take any integer value from 1 to N . Using these indices, let R_j^i represent the current response value and R_T^i be the constraint limit for that response. For the sake of normalization, a baseline is usually needed – let R_0^i represent the baseline value of the constrained responses. Using these variables, the responses' relative location to the constraint boundary (i.e., the constraint violation) can be normalized according to Equation (21).

$$\theta_j^i = \frac{R_T^i - R_j^i}{R_T^i - R_0^i} \quad (21)$$

Using the normalized response values, the change in response from the previous iteration can be found using Equation (22) and used to estimate the next iteration's responses by projecting the change onto the current response value as shown in Equation (23)

$$\Delta\theta^i = \theta_j^i - \theta_{j-1}^i \quad (22)$$

$$\theta_p^i = \theta^i + \Delta\theta^i \quad (23)$$

With these values calculated for each response, weighting factors can be found by multiplying the projected location by the response's relative change and comparing it to the sum of the same value for all other responses, as shown in Equation (24). Finally, the adjusted constraint tolerance can be calculated from Equation (25)

$$W^i = \frac{\theta_p^i \Delta \theta^i}{\sum_{k=1}^n \theta_p^k \Delta \theta^k} \quad (24)$$

$$\lambda_j = \lambda_{Min} + (\lambda_{Max} - \lambda_{Min}) \frac{N}{N + 2j} \sqrt{\sum_{k=1}^n (W^k \theta_p^k)^2} \quad (25)$$

Using this new variable tolerance, the optimization problem statement for the multilevel optimization problem with ATC can be written in the form of Equations (26). Notice that the index symbols here have changed from the previous problem statements in order to remain consistent with the ATC formulation.

| | | |
|-------------------|---|----------------------------|
| Objectives: | Minimize Volume | |
| Constraints: | $\sigma_{RegionB} \leq \sigma_{Limit}$ | |
| | $\Theta_{RegionB} \leq \Theta_{Limit}$ | |
| | $(1 - \lambda_j) \tilde{\Lambda}_i < \Lambda_i < (1 + \lambda_j) \tilde{\Lambda}_i$ | $i \in \{1, 2, \dots, 7\}$ |
| | $(1 - \lambda_j) \tilde{\Lambda}_F < \Lambda_F < (1 + \lambda_j) \tilde{\Lambda}_F$ | |
| | $(1 - \lambda_j) \tilde{\Lambda}_M < \Lambda_M < (1 + \lambda_j) \tilde{\Lambda}_M$ | |
| Design Variables: | (See Table 2 in Section 2.2) | (26) |

Using the same baseline designs that were used in the previous section, the same studies were performed, but using ATC to adjust the tolerance.

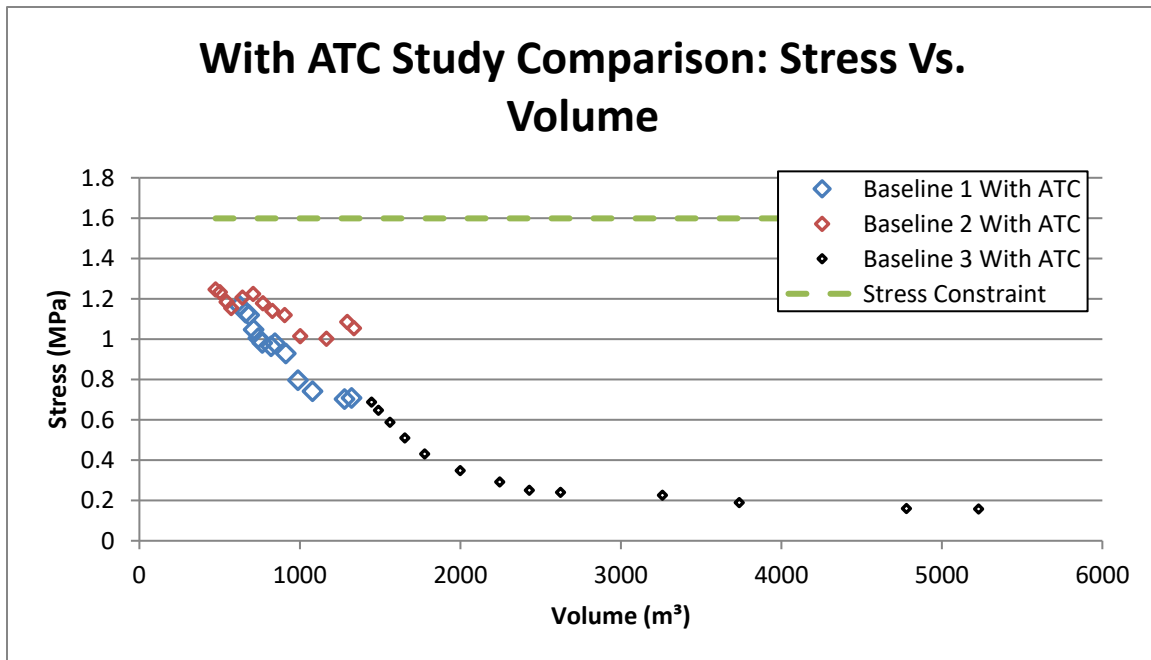


Figure 25. Relationship between Volume and Stress for the unconstrained Transmission Tower problem using Adaptive Tolerance Control

The data shows that for the unconstrained studies, the inclusion of ATC produces much better results.

From Figure 25, notice that the studies' volumes decrease much more rapidly than when compared to Figure 20. As with Figure 20, the progression of the study can be followed from right to left for a given data set. For the constrained studies; however, while the inclusion of ATC decreases the volume more than the exclusion of ATC, both studies conclude with infeasible results, as shown in Figure 26. This suggests that the current formulation of ATC does not serve its purpose. Preliminary results suggest that a newer version of ATC may produce more favorable results; however, this has not been extensively studied.

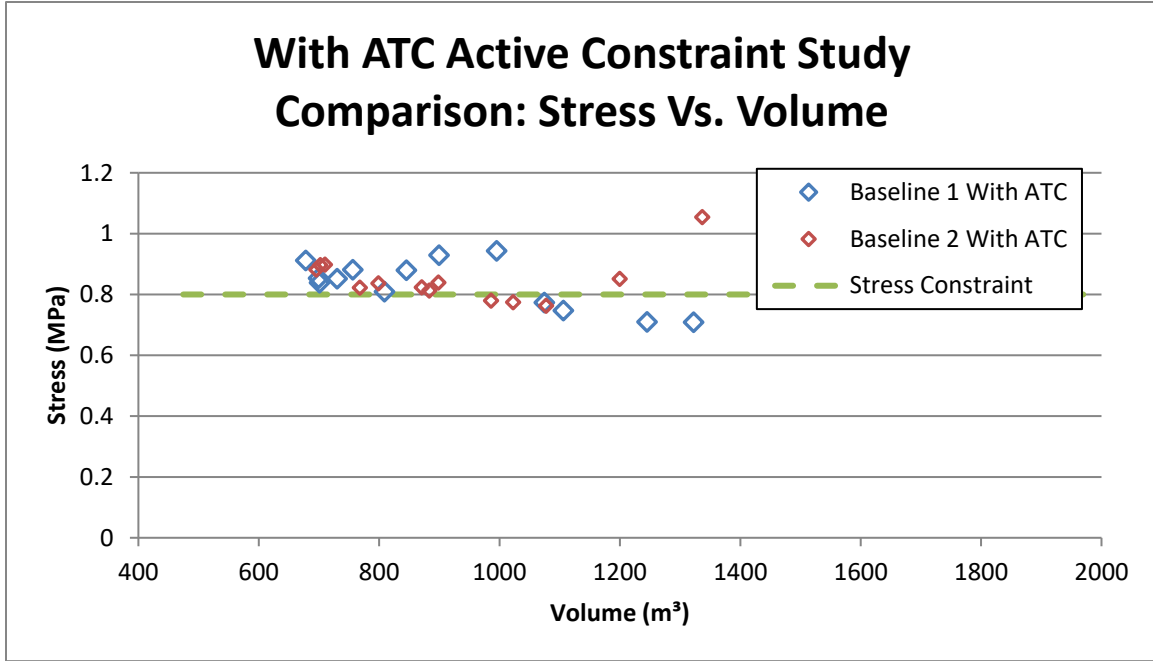


Figure 26. Relationship between Volume and Stress for the constrained Transmission Tower problem using Adaptive Tolerance Control.

5.6 Multilevel Optimization with Adaptive Constraint Scaling

Despite the improvements made by including ATC in the multilevel process, the use of doppelgänger members still caused infeasible results to be found at the macro level from feasible results at the micro level. To further improve the multilevel search process and encourage feasible macro-level results, the idea of scaling the global constraints at the micro level was introduced.

To determine the amount of constraint scaling required for a given response and a given iteration, a simple algorithm was developed that accounted for the amount of discrepancy between the maximum response value at the micro level study and the actual constraint limit at maximum response value found at the macro level. The algorithm also accounted for how close the study was to completion by keeping track of the current iteration number and comparing it to the allotted number of iterations.

The algorithm starts by calculating the maximum global-local response discrepancy for each response in the previous iteration. Then, for each lattice member, if the response value is greater than

the difference between the global constraint limit and the product of the response's global-local discrepancy and the global response value, then the constraint limit used in the micro-level iteration is reduced by a certain amount. If this condition is not met, however, then the discrepancy for that response is not a concerning factor for the next iteration, and that constraint value is left unmodified in the micro level iteration.

Keeping the definition of n , N , i , and j as defined in Section 5.5, and introducing M as the total number of lattice members and k as the index of the current member, the formulation of the Adaptive Constraint Scaling (ACS) can begin. In addition to these parameters, let G^i be the global constraint limit for the current response, R_G^i be the current global response value, g_k^i be the micro level constraint limit placed on the current response, and r_k^i be the current micro level response value. Then, the Global-Local Response Discrepancy can be calculated as shown in Equation (27) and the ACS algorithm defined, which was developed through trial and error.

$$\varepsilon^i = \frac{R_G^i - \max\{r_1^i, r_2^i, \dots, r_M^i\}}{R_G^i} \quad (27)$$

For k=1:M # For each lattice member...

For i=1:n # For every constrained response (temperature and stress)...

If $r_k^i \geq G^i - \varepsilon^i R_G^i$ # If a micro level response is greater than the Global Constraint Value after accounting for the response error/discrepancy...

$$g_k^i = G^i - \frac{r_k^i + \varepsilon^i R_G^i - G^i}{\text{erfc}(\varepsilon^i)} \left(\frac{2N + i}{3N} \right) \quad \# \text{ Adjust the micro level constraint value for that response}$$

Else

$$g_k^i = G^i \quad \# \text{ Otherwise, leave the micro level constraint value unmodified}$$

End

End

End

To analyze the impact of the adaptive Constraint Scaling, Baseline 1 with an active stress constraint was rerun twice more– Once with ATC and Adaptive Scaling (Figure 28), and once without

ATC (Figure 27). As these figures show, constraint scaling resulted in a feasible global solution when used without ATC; however, it did not result in feasible solutions when used in combination with ATC.

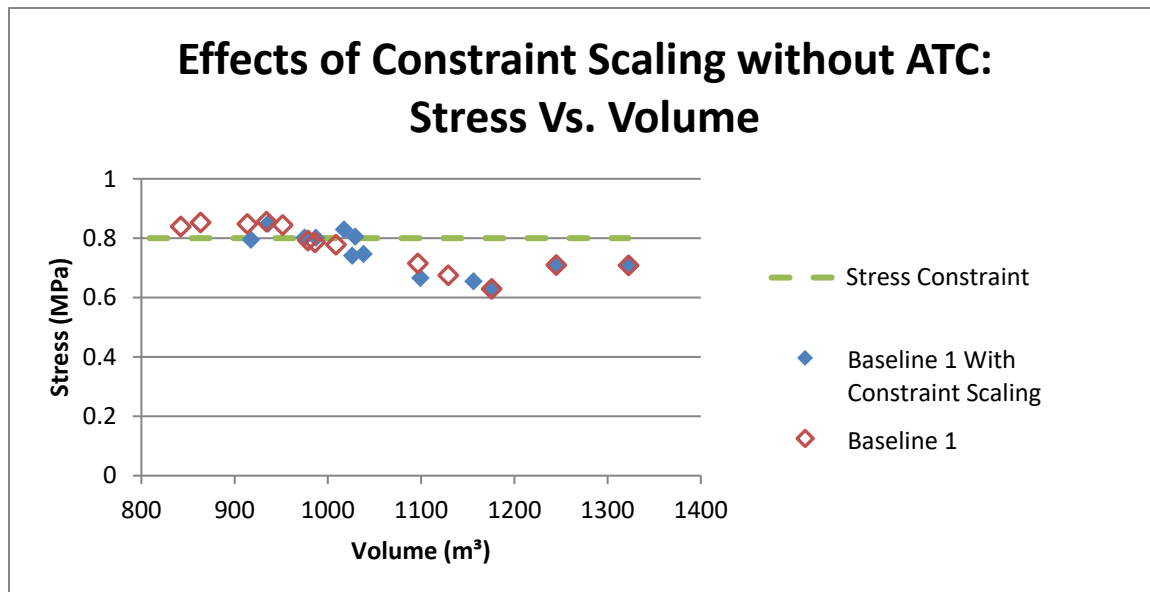


Figure 27. The effects of Constraint Scaling as shown by the relationship between Volume and Stress for the constrained Transmission Tower Problem.

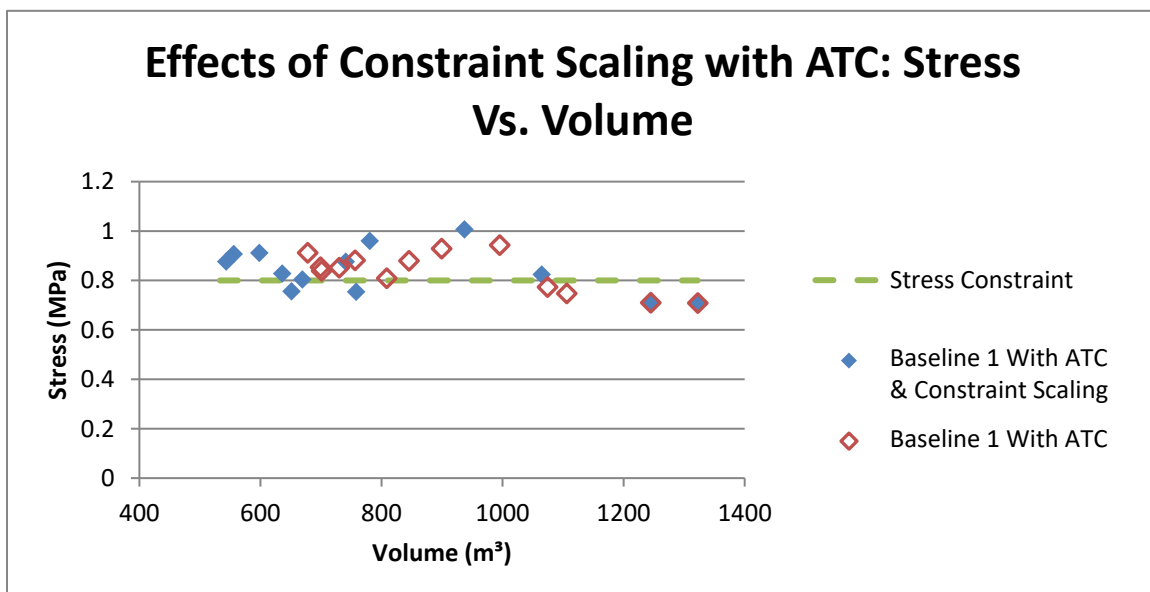


Figure 28. The effects of Constraint Scaling with Adaptive Tolerance Control as shown by the relationship between Volume and Stress for the constrained Transmission

Upon reviewing these results, it was found that the ATC equation was not behaving as intended. The equation used places too much emphasis on the change in responses and not enough

on the relative distance of a response from the constraint. Thus, a modified ATC equation was developed. This new equation more equally balanced the priorities of relative change in response and relative distance from the constraint boundaries and is of the form shown in (28)

$$\lambda_{j+1} = \left(\lambda_{Max} - \frac{\lambda_{Max} - \lambda_{Min}}{e^{\alpha_{j+1}}} \right) \beta_{j+1} + \lambda_{Min} (1 - \beta_{j+1}) \quad (28)$$

Where β is a variable that determines the amount of influence the tolerance definition has on the iterations tolerance, and is a function of ρ , which can be thought of as the “Degree of Feasibility” and is defined in (29), where W_j^k is the weighting factor for all constrained responses and Δv_j is the change in normalized volume. A value of $\rho = 0$ presents the situation in which all constraints are directly on the constraint boundary, a negative value of ρ shows an infeasible response and a positive value shows a feasible response. The relationship between ρ and β is shown in Figure 29 and is given by (31). Both Equation (28) and Equation (31) were developed using trial and error, trying to mimic the desired behavior.

$$\rho_j = \sum_{k=1}^n W_j^k \theta_P^k \quad (29)$$

$$W_j^i = \frac{\theta_P^i \frac{\Delta \theta_j^i}{\Delta v_j}}{\sum_{k=1}^n \theta_P^k \frac{\Delta \theta_j^k}{\Delta v_j}} \quad (30)$$

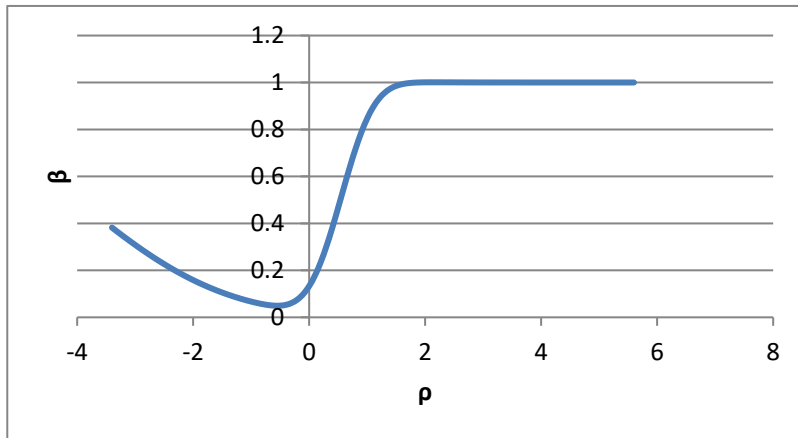


Figure 29. The relationship between the “Degree of Feasibility” and the tolerance control factor, β , in the Modified ATC equation.

$$\beta_{j+1} = \frac{1}{2} \left[\operatorname{erfc} \left(\sqrt{2} + \frac{\sqrt{2}}{4} \rho_j \right) + \operatorname{erfc} \left(\frac{\sqrt{3}}{2} - \frac{\pi}{2} \rho_j \right) \right] \quad (31)$$

The term in the denominator of (28) is used to normalize the tolerance range and control the impact of the range based on iteration number. Earlier iterations will be allowed to have a more allowance to span the range than the later iterations. This is controlled by the α parameter, which is given by (32). All other parameters have been previously defined somewhere within this text.

$$\alpha_{j+1} = \frac{N}{N + 2j} \sqrt{\sum_{k=1}^n (w^k \theta_p^k)^2} \quad (32)$$

Some problematic situations may arise when using the ATC equation defined in (28). This equation requires there to be a change in the design from iteration to iteration, so, if a better design is not found, this equation will fail (divide by zero error when calculating the weighting factor). Furthermore, this equation cannot be used for the first iteration because of the use of the change in response values. For the first iteration, it is recommended that the minimum tolerance value is used.

Using this modified version of the ATC equation with constraint scaling, Study 6 was rerun a final time. The results are shown in Figure 30; however, unlike the previous plots, the progression of this study can no longer be followed from right to left on this plot. Rather, Figure 31 should be referenced, which clearly shows that this method found a feasible solution. Furthermore, this figure also shows that the stress oscillates much closer to the constraint boundary, as was the objective of defining a new ATC equation. Clearly, more research is needed to prove the validity of this equation and method, but these results are very promising. Lastly, the modified ATC with ACS results in a feasible solution with a smaller volume than found with ACS alone.

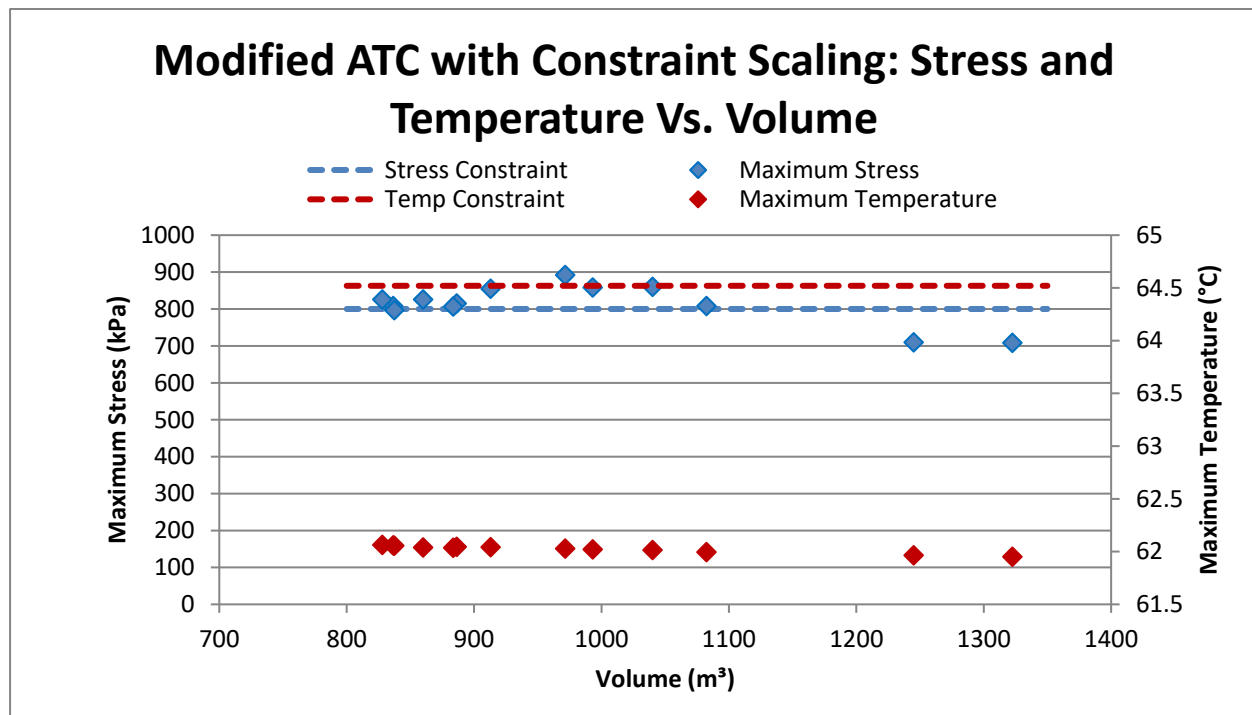


Figure 30. Results of the Transmission Tower Problem using the Modified ATC equation and Constraint Scaling that show the relationship that Stress and Temperature have with Volume. Note that the results are not monotonic and cannot be followed from right to left as with most other figures in this work.

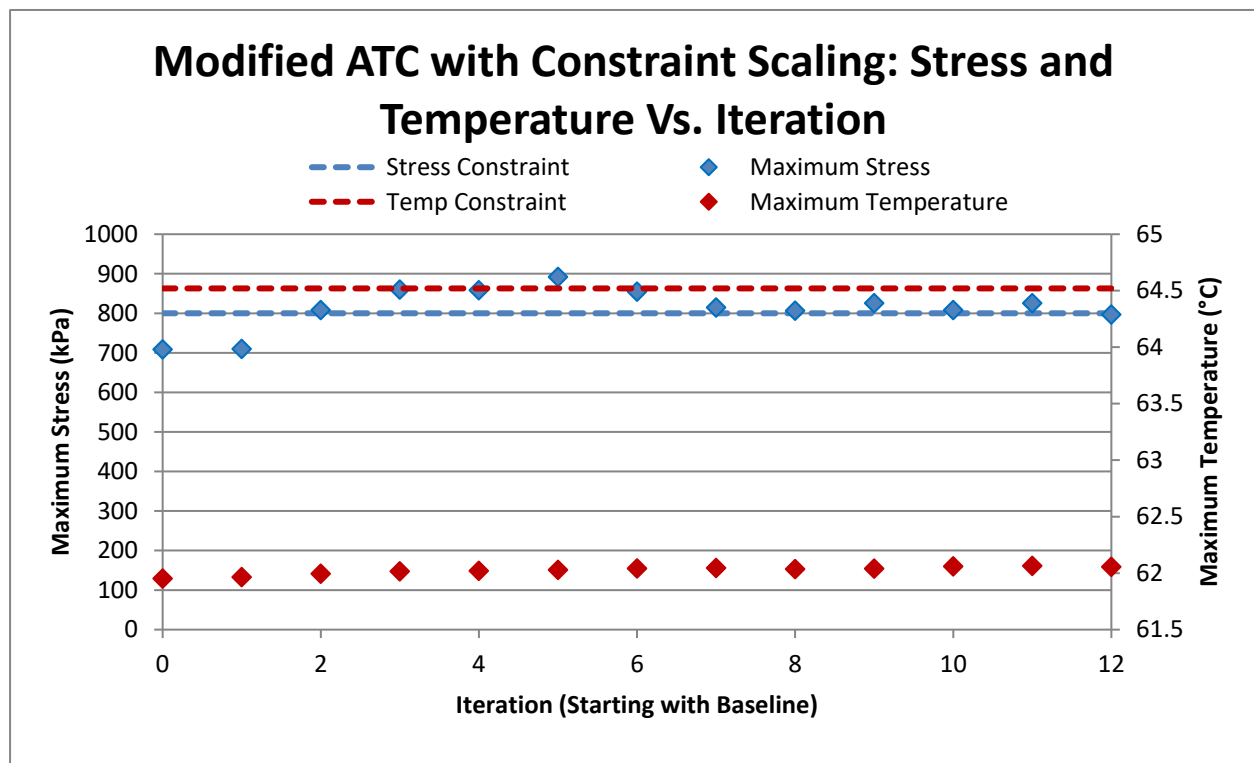


Figure 31. Results of the Transmission Tower Problem using the Modified ATC equation and Constraint Scaling that show the progression of the study. This proves the results for this

CHAPTER 6: Analysis of Results and Drawn Conclusions

6.1 Conclusions

In order to facilitate AM's continued growth, new design and analysis methodologies must be developed that will accommodate the continued advancement of Additive Manufacturing processes.

Significant research has been previously conducted into the effects of processing parameters of AM process on the structural capabilities of products. These studies show significant correlations, suggesting that microstructure is very important in the structural quality of an AM product. Limited computational resources prohibit the ability to model the microstructure in CAD or analyze it with FEA, thus limiting the effectiveness of optimization for Additive manufacturing processes. A Multilevel Optimization process could be the key to reducing the computational expense of analyzing additively manufactured parts while maintaining the reliability of the simulation.

6.1.1 Two-Step Serial Optimization

As a starting point, a Two-Step Serial Optimization Algorithm was developed, which involves first optimization a simplified macro-model, and then optimizing the micromodels of each member individually. This was shown using a two-dimensional three-member structure, with a built-in intuition check – One of the three members was a free-rolling member, which had very little influence on the rigidity of the structure, and thus should have very little mass. Three different variants of the algorithm were analyzed, which utilized different constraining techniques. The techniques differed by what reactionary forces/moments were used as constraints. Visually, the “Individual Only” constraints, the “Magnitude Only” constraints, and the “Combined” constraint results all agree with intuition, though the “Magnitude Only” and “Combined” constraints appear to reduce the mass significantly more.

As a process validation, the entire global model was optimized using a traditional multiobjective methods to find the Pareto Front, which was used to compare the 3 multilevel studies.

This showed that not only did the “Magnitude Only” and “Combined” constraints reduce the volume significantly more than the “Individual Only” study, they actually appear to be closer to the Pareto Front as well.

A couple of issues can be found with this study. First, this technique will only work for a very limited subset of problems. Any problem with a high degree of nonlinearity between the micro level and the macro level will not perform well with this technique. A more generalized algorithm should be developed that can handle a higher degree of nonlinearity. Secondly, this study does not contain a global constraint to limit the mass from continually increasing. Because of this, a truly “optimal” design will never actually be found, but rather, a noncompeting set of designs that resemble a Pareto front will be found, and the optimal must be manually chosen.

Because the Two-Step Serial technique described in Chapter 3 was highly specific to a certain type of problem, a multilevel optimization technique was developed. The problem statement used in the experiments outlined in Chapter 2 was again used for two reasons. First, using the same problem avoided the requirement of developing a new analysis model. Secondly, by using the same model, the same AAO validation data could be used, which was garnered from a high-fidelity optimization study. Because this study lacks any global constraints the results found cannot really be considered “optimal”, and thus is not a great problem to begin with. Still, conclusions can be made about the behavior of the different constraining techniques and methods.

6.1.2 Multilevel Optimization

A preliminary investigation showed that feasible solutions were very difficult to find. To improve the number of feasible designs returned in a given iteration, a couple of different techniques were employed. Two constraint normalization schemes were examined – the commonly used constraint boundary value scheme, as well as a scheme that attempted to mimic real-world

structural behavior. Furthermore, two constraint tolerance values were used – 5 and 10 percent deviation from the system-level response.

Regardless of the normalization scheme, the 5 percent tolerance studies failed to produce any feasible solutions, and thus the 5 percent value was deemed to be a poor choice for a tolerance value. Data was collected that compared the “Combined Constraints” method and the “Individual Only” method, each using a 10% constraint tolerance. This data showed that while both methods appeared to be moving in the correct direction, the “Combined Constraints” method converged to the Pareto Front much sooner.

One potential improvement to this study would be in the way that the zero-constraints were handled. With the exception of free boundaries, there are very few practical situations in which a boundary’s reactions would be zero and most of those that do find zero-boundaries can have their basis rotated such that the boundary component is no longer zero.

6.1.3 Multilevel Optimization of a multi-physics problem

In an attempt to solve a problem that is a bit more practical than those previously examined, a three-dimensional multi-physical application was found. In this study, a transmission tower was analyzed using both static and thermal analyses. Furthermore, this problem contained global constraints, which the previous studies were lacking. Though this study attempted to address problems present in the previous studies, it was not flawless. The biggest flaw was that it used rotational symmetry in an attempt to reduce the amount of calculations and studies required to optimize the system. This assumed that the boundary conditions experienced by a member on one side of the model would be the same in the corresponding member on the other sides of the model. It was recognized that this assumption was not completely correct prior to running the studies; however, the extent of which it was incorrect was not recognized until the study was underway.

Using the results of the previous chapters, the “Combined Constraints” method was used; however, there were some significant differences in this problem. As previously mentioned, this problem had system level constraints, which were also applied to sub-system level. These constraints were treated as normal constraints and were normalized by the constraint limits. Furthermore, due to the three-dimensional geometry, no nonzero reactions were found. This alleviated the concerns mentioned at the end of the previous section.

The first set of studies that were run had system level constraints that were large enough to be inactive in every iteration. Thus, the first set of studies can be thought of as unconstrained. As would be expected, the results improved each iteration, for each of these studies. While this is reassuring, it isn’t all that useful because most problems are not unconstrained.

After changing the system level constraints such that the studies are expected to act as constrained problems, the studies were rerun. This resulted in some very interesting behavior. While the subsystem level iterations always resulted in feasible solutions, when the same variable values were used in the system level analysis, the design was found to be infeasible. This was found to be the result of the poor assumption mentioned at the beginning of this section. Another possible cause for this error is that the interactions between the subsystems that occur at the system level were not accounted for at the subsystem level.

To try balance the rate of convergence with the unpredictability of the doppelgänger reactions, an algorithm was developed that attempted to control the tolerance of the reaction constraints based on the relative location to the constraint boundaries. This algorithm was referred to as ATC. For the unconstrained studies, this greatly increased convergence. For the constrained studies, this also improved the rate of objective improvement (minimize volume), however, it did not result in a feasible solution after 12 iterations.

Because the unpredictability of the doppelgänger members could not be controlled, the idea of anticipating the unpredictability was investigated. This involved scaling the system level constraints during the subsystem analyses based on the previous amount of system-level constraint violation. This was found to result in a feasible system level solution when used by itself and an infeasible system level solution when used in conjunction with ATC. During this set of studies, it was realized that ATC was not behaving as it was intended and was reformulated.

Using the reformulated ATC algorithm, the final study was rerun with the constraint scaling method. Not only does the stress vs. iteration plot shows much more of a damped oscillation than the previous version of ATC, which was the goal of the formulation in the first place, the final system level solution was feasible and had a smaller final volume than the study with ACS alone.

6.2 Future Work

There is clearly much more work that needs to be done to validate these conclusions. There were several issues found with the problem statements and models used. While these were attempted to be fixed in each consecutive problem, more issues were uncovered. The following sections list some of the issues with this work, as well as potential improvements that could be made.

6.2.1 Validation and Verification of the Modified ATC Equation

The latest form of the ATC equation showed very promising results, but much more work is needed to validate this. Only one study was performed which is not enough to make claims that it is a robust multilevel optimization technique.

6.2.2 Average Mises Stress instead of Maximum Mises Stress

While the use of maximum stress as the constraints in the studies examined in the scope of this work were more a realistic constraint to choose, due to the way the data was collected, perhaps using average stress would have been a more realistic representation of the stress experienced by

the structure. This is because the data collected was the raw nodal data extracted from the Abaqus output file, and no nodal averaging was performed. This means the “maximum” stress found from the analysis will always be less than the actual maximum stress experienced by the structure.

6.2.3 Sensitivity and Robustness

Sensitivity studies are useful at eliminating variables from a study that have a relatively small impact on a study. The downside of these is that they are really only helpful in the vicinity of an optima. Because each microlevel analysis is constrained to a 10% tolerance of the global reaction force/moment, each microlevel study doesn’t explore the global design space a significant amount. Accordingly, it may be beneficial to run a sensitivity study prior to each microlevel analysis and eliminate low-impact variables from each iteration, being sure to add them back in the next iteration.

Because processing parameters for additive manufacturing processes can be difficult to control, and because the microstructure is so dependent on these processing parameters, the actual microstructure of a part may vary from the expected microstructure. To account for this possible variance, it would be useful to include a robustness study in the scope of a multilevel optimization algorithm aimed at optimizing additively manufactured products.

6.2.4 Homogenization in the Global Model

Using the full non-homogenized system level model is a great way to develop and validate an algorithm, but once an algorithm is fully developed and adequately tested, finding a way to homogenize the system level model would potentially be a great way to further reduce the computations required and possibly reduce the duration of the optimization process. One potential method to homogenize the model would be to use SwiftComp to determine the effective engineering constants of a structure and build a simple model with custom materials.

6.2.5 Handling Global Level Infeasibilities

All of the studies investigated in the scope of this work had baselines that were found to have feasible system level responses. Some optimization algorithms require this to be the case while others work better with infeasible baselines. The baseline design has profound implications on the behavior of a study and ideally the algorithm used will account for that. Investigating how this algorithm behaves with infeasible baselines would be important to understand prior to using this in real-world applications.

6.2.6 Automation of the Multilevel Processes

Once a reliable algorithm is found, the communication between the micro and macro levels should be automated. This would greatly increase the efficiency (in terms of total time) of the entire study. Automating the communication was out of the scope of this work, but should be examined and considered moving forward as it would greatly increase the productivity. This increase does depend on the ability to rely on the error-free completion of the microstudies. The issues presented by the current versions of LatticeMaker would degrade the improvements made by automation, as uncaught errors would result in entire studies being invalid and needing to be rerun, rather than just a single iteration.

6.2.7 Three Level Optimization Process

The scope of this work is limited to two level optimization problems whereas the usefulness of a multilevel optimization algorithm really increases with its ability to work on problems with an indefinite number of levels. Accordingly, research should be conducted in which this algorithm is adapted to work on a more generalized three-level optimization problem. Once this is shown to work on a three level problem, it is not hard to extrapolate its usefulness to higher-level problems

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] P. Halepatali, Optimization of Sub Components within a Large System, East Lansing: Michigan State University, 2004.
- [2] I. Gibson, D. Rosen and B. Stucker, Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing, New York: Springer, 2015.
- [3] D. Espalin, J. A. Ramirez, F. Medina and R. Wicker, "Multi-material, multi-technology FDM: exploring build process variations," *Rapid Prototyping Journal*, vol. 20, no. 3, pp. 236-244, 2014.
- [4] J. Kietzmann, L. Pitt and P. Berthon, "Disruptions, decisions, and destinations: Enter the age of 3-D printing and additive manufacturing," *Business Horizons*, vol. 58, no. 2, pp. 209-215, March-April 2015.
- [5] S. Lim, R. Buswell, T. Le, S. Austin, A. Gibb and T. Thorpe, "Developments in construction-scale additive manufacturing processes," *Automation in Construction*, vol. 21, pp. 262-268, January 2012.
- [6] L. Podshivalov, C. M. Gomes, A. Zocca, J. Guenster, P. Bar-Yoseph and A. Fischer, "Design, Analysis and Additive Manufacturing of Porous Structures for Biocompatible Micro-Scale Scaffolds," *Procedia CIRP*, vol. 5, pp. 247-252, 2013.
- [7] A. A. Zadpoor and M. Jos, "Additive Manufacturing of Biomaterials, Tissues, and Organs," *Annals of Biomedical Engineering*, vol. 45, no. 1, pp. 1-11, 2017.
- [8] B. Huang and S. B. Singamneni, "Curved Layer Adaptive Slicing (CLAS) for fused deposition modelling," *Rapid Prototyping Journal*, vol. 21, no. 4, pp. 354-367, 2015.
- [9] F. Rayegani and G. C. Onwubolu, "Fused deposition modelling (FDM) process parameter prediction and optimization using group method for data handling (GMDH) and differential evolution (DE)," *International Journal of Advanced Manufacturing Technology*, 2014.
- [10] O. A. Mohamed, S. H. Masood and J. L. Bhowmik, "Experimental investigation for dynamic stiffness and dimensional accuracy of FDM manufactured part using IV-Optimal response surface design," *Rapid Prototyping Journal*, vol. 23, no. 4, pp. 736-749, 2017.
- [11] M. Domingo-Espin, J. M. Puigoriol-Forcada, A.-A. Garcia-Granada, J. Lluma, S. Borros and G. Reyes, "Mechanical property characterization and simulation of fused deposition modeling Polycarbonate parts," *Materials and Design*, vol. 83, pp. 670-677, 2015.

- [12] S.-H. Ahn, M. Montero, D. Odell, S. Roundy and P. K. Wright, "Anisotropic material properties of fused deposition modeling ABS," *Rapid Prototyping Journal*, vol. 8, no. 4, pp. 248-257, 2002.
- [13] C. M. Haid, "Characterizing Tensile Loading Responses of 3D Printed Samples," Massachusetts Institute of Technology, 2014.
- [14] O. S. Es-Said, J. Foyos, R. Noorani, M. Mendelson, R. Marloth and B. A. Pregger, "Effect of Layer Orientation on Mechanical Properties of Rapid Prototyped Samples," *Materials And Manufacturing*, vol. 15, no. 1, pp. 107-122, 2000.
- [15] "MakerBot," MakerBot, [Online]. Available: <https://www.makerbot.com/>.
- [16] S. Kapil, P. Joshi, H. V. Yagani, P. M. Kulkarni, R. Kumar and K. Karunakaran, "Optimal space filling for additive manufacturing," *Rapid Prototyping Journal*, vol. 22, no. 4, pp. 660-675, 2016.
- [17] T. Zegard and G. H. Paulino, "Bridging topology optimization and additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 53, no. 1, pp. 175-192, 2016.
- [18] B. Zhu, M. Skouras, D. Chen and W. Matusik, "Two-Scale Topology Optimization with Microstructures," *ACM Transactions on Graphics*, vol. 36, no. 5, 2017.
- [19] M. Beers and G. N. Vanderplaats, "A Linearization Method for Multilevel Optimization," in *Numerical Techniques for Engineering Analysis and Design*, Swansea, 1987.
- [20] J. Sobieszczanski-Sobieski, B. B. James and M. F. Riley, "Structural Optimization by Generalized, Multilevel Decomposition," Nasa Technical Memorandum, Hampton, 1985.
- [21] W. Yao, X. Chen, Q. Ouyang and M. van Tooren, "A surrogate based multistage-multilevel optimization procedure for multidisciplinary design optimization," *Structural and Multidisciplinary Optimization*, vol. 45, no. 4, pp. 559-574, 2012.
- [22] B. A. Wujek, J. E. Renaud, S. M. Batill and J. B. Brockman, "Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment," *Concurrent Engineering*, vol. 4, no. 1, pp. 361-377, 1996.
- [23] N. Chase, R. Sidhu and R. Averill, "A New Method for Efficient Global Optimization of Large Systems Using Sub-models," in *12th International LS-DYNA Users Conference*, Detroit, 2012.
- [24] E. D. Goodman, R. C. Averill and R. Sidhu, "Multi-Level Decomposition for Tractability in Structural Design Optimization," in *Evolutionary Computation in Practice. Studies in Computational Intelligence*, vol. 88, Berlin, Heidelberg, Springer, 2008, pp. 41-62.

- [25] Red Cedar Technology, "SHERPA - An Efficient and Robust Optimization/Search Algorithm," [Online]. Available: <http://www.redcedartech.com/pdfs/SHERPA.pdf>. [Accessed 9 October 2017].
- [26] *Internal research code developed at Georgia Tech..*